INSTITUTO FEDERAL DE CIÊNCIA E TECNONOLOGIA DE SÃO PAULO
CÂMPUS SÃO JOÃO DA BOA VISTA

Filipe Navas Rodrigues

CONCEPÇÃO DE UMA FERRAMENTA DE APOIO À INDEXAÇÃO PARA BIBLIOTECÁRIOS

São João Da Boa Vista
2015

Filipe Navas Rodrigues

CONCEPÇÃO DE UMA FERRAMENTA DE APOIO À INDEXAÇÃO PARA BIBLIOTECÁRIOS

Trabalho de conclusão de curso apresentado ao Instituto Federal de São Paulo, como parte dos requisitos para a obtenção do grau de Tecnólogo em Sistemas para Internet.

Área de Concentração: Linguística Computacional

Orientadora: Profa. Dra. Rosana Ferrareto Lourenço Rodrigues
Coorientadores: Prof. Ms. Gustavo Aurélio Prieto e Ms. Maria Carolina Gonçalves

São João da Boa Vista

2015

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – IFSP

Rodrigues, Filipe Navas.
Concepção de uma ferramenta de apoio à indexação para bibliotecários. / Filipe Navas Rodrigues; orientadora Rosana Ferrareto Lourenço Rodrigues. São João da Boa Vista, 2015.

Trabalho de Conclusão de Curso, IFSP, 2015.

1. Mapas do Conhecimento. 2.Banco de Dados. 3.Grafos. 4. Indexação.

I. Concepção de uma ferramenta de apoio a indexação para catálogos bibliotecários.

AGRADECIMENTOS

Gostaria primeiramente de agradecer a Deus, por todas as oportunidades em minha vida.

A minha família, por todo o apoio e incentivo em minhas decisões.

A todos os meus amigos, que sempre estiveram comigo nessa jornada.

E a todos os professores e educadores, que mostraram o caminho, e àqueles que me ajudaram nesta jornada, em especial à Professora Rosana, ao Professor Gustavo, à Bibliotecária Maria Carolina, por toda a contribuição e apoio nesta pesquisa, que se insere em um contexto diferenciado dentro da informática e que, de alguma forma, vem quebrando algumas barreiras e mostrando o grande valor de ideias diversas, da inovação e do trabalho conjunto.

Ao Willian que iniciou esta proposta em seu excelente trabalho.

A todos os colegas envolvidos na pesquisa no IFSP – SBV e pelas inúmeras discussões da tarde.

A todo IFSP – SBV e seus profissionais e alunos, que com toda certeza foram um diferencial em minha vida e que me ofereceram diversas oportunidades.

Pelos ensinamentos da Professora Rosana e do Professor Gustavo, não só didáticos, mas por toda a experiência e conhecimento passado tanto em aulas quanto em conversas.

Agradeço a todos que fizeram de mim uma pessoa melhor nesse período. Sinto uma enorme gratidão por todos.

RESUMO

RODRIGUES, F. N. (2015). **Concepção de uma ferramenta de apoio à indexação para bibliotecários.** Trabalho de Conclusão de Curso - Instituto Federal de São Paulo, São João da Boa Vista, 2015.

O bibliotecário, como responsável pela indexação, tem conhecimento limitado acerca do vocabulário específico utilizado pelos especialistas em diferentes áreas. Desta forma, é importante uma ferramenta que propicie uma facilidade na organização e recuperação de assuntos para a indexação. Assim, esta pesquisa tem como objetivo propor e conceber uma ferramenta de apoio à indexação para bibliotecários. Este trabalho baseia-se na Linguística, que fornece o estudo de estruturas de organização semântica do conhecimento, fazendo esta pesquisa proveito dos Mapas do Conhecimento, na Ciência da Informação, que fornece o aporte necessário a respeito da indexação, e na Informática, que apresenta recursos e ferramentas tecnológicas necessárias para o desenvolvimento da solução proposta. Foi desenvolvido um sistema *web* em Java que utiliza o banco de dados em grafo Neo4j para armazenar os dados, que são nós e relacionamentos. Os nós representam os assuntos das obras de um acervo, e os relacionamentos as relações entre eles, que foram definidos nos princípios de definição e contiguidade da semântica lexical “é”, “é um”, e “está contido”. A ferramenta permite que o usuário possa incluir os nós (assuntos) e seus relacionamentos. Ainda permite a edição e deleção dos nós. Além disso, a ferramenta provê uma interface em que o usuário visualiza os assuntos e relacionamentos na forma de um grafo, desenvolvido com auxílio da biblioteca *JavaScript vis.js*. A pesquisa apresenta como resultado a ferramenta desenvolvida, que há de ser de grande valia para os bibliotecários. Além disso é notável a multidisciplinaridade deste estudo, que envolve várias áreas do conhecimento. Finalmente, este trabalho permite que outros pesquisadores expandam o projeto, como por exemplo definindo diferentes tipos de relacionamentos, e junto com bibliotecários criem uma fonte de recuperação de assuntos padrão.

Palavras-chaves: Mapas do Conhecimento. Banco de Dados. Grafos. Indexação.

ABSTRACT

RODRIGUES, F. N. (2015). **Conception of a tool to support the indexing process for librarians.** Course Conclusion Project – Instituto Federal de São Paulo, São João da Boa Vista, 2015.

Librarians, as responsible for indexing, have limited knowledge of the specific vocabulary used by specialists in different fields. Thus, a tool that provides an ease in organizing and retrieving subjects for indexing is important. Therefore, this research aims to propose and develop a support tool for indexing to be used by librarians. This work is based on Linguistics, which provides the study of semantic knowledge and organization structures, wherein this research has taken advantage of the Knowledge Maps, on the Information Science, which provides the input necessary regarding indexing, and the Computer Science that provides the resources and technological tools necessary for the development of the proposed solution. A web system in Java that uses the graph database Neo4j to store the data has been developed. The data consists of nodes and relationships. The nodes represent the subjects of the works of a collection, and relationships represent the relationships between them, which have been defined under the lexical semantics principles of definition and contiguity "*é* (is)", "*é uma (*it is one)" and "*está contido* (is contained)". The tool allows the user to include nodes (subjects) and their relationships. It also allows editing and deletion of nodes. In addition, the tool provides an interface where the user views the subjects and relationships as a graph, which was developed with the JavaScript library *vis.js*. The research has resulted in the developed tool, which shall be of great value to librarians. Also notable is the multidisciplinary approach of this study, which involves several areas of knowledge. Finally, this study allows other researchers to expand the project, such as adding different types of relationships, and together with librarians to create a standard source of storage and recovery of subjects.

Keywords: Knowledge Maps. Database. Graphs. Indexing.

LISTA DE FIGURAS

LISTA DE SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
AJAX	*Asynchronous JavaScript and XML*
BBL	*BrainBank Learning*
CALL	*Computer Assisted Language Learning*
CI	Ciência da Informação
CSS	*Cascading Style Sheets*
EAD	Ensino a distância
HCI	*Human Computer Interaction*
HTML	*HyperText Markup Language*
HTML5	*HyperText Markup Language 5*
HTTP	*Hypertext Transfer Protocol*
IA	Inteligência Artificial
IFSP	Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
IHC	Interação Humano-Computador
ISO	*International Organization for Standardization*
JSON	*JavaScript Object Notation*
JSP	*Java Server Pages*
JVM	*Java Virtual Machine*
MIT	*Massachusetts Institute of Technology*
NoSQL	*Not Only* SQL
NBR	Norma Brasileira
OLAP	*Online Analytical Processing*
OPAC	*Online Public Access Catalog*
PC	*Personal Computer* (Computador Pessoal)

PLN	Processamento Automático de Línguas Naturais
RDF	*Resource Description Framework*
SBV	São João da Boa Vista
SGBD	Sistema Gerenciador de Banco de Dados
SPARQL	*Protocol and RDF Query Language*
SQL	*Structured Query Language*
TIC	Tecnologias da Informação e Comunicação
TSI	Tecnologia em Sistemas para Internet
UNESCO	*United Nations Educational, Scientific and Cultural Organization*
UNISIST	*United Nations International Scientific Information System*
W3C	*World Wide Web Consortium*

SUMÁRIO

1 Introdução

Atualmente, há muitas aplicações computacionais que projetam atividades reais e ambientes físicos para o mundo virtual. Essa busca por criar um mecanismo que facilite a vida das pessoas, automatizando processos por meio de programas, é uma atividade que pode envolver várias áreas do conhecimento, como modelagem de *software* e análise de requisitos (Engenharia de Software), IHC (Interação Humano-Computador), programação e linguística, entre outras. Essa união de esforços interdisciplinares permite a viabilização de práticas que vão desde a identificação e análise de um problema à sua solução computadorizada, na busca por formas de apresentação de soluções ao usuário e de otimização da sua relação com o sistema.

Nesse contexto, desenvolve-se esta pesquisa, que visa a conceber uma ferramenta de apoio à indexação para bibliotecários, a partir de um trabalho iniciado por Silva (2013). Indexação é o processo de análise de assunto de um documento através da leitura documentária (SOUSA; FUJITA, 2014).

Dando continuidade a este trabalho, nesta etapa, será priorizado o aprimoramento do desenvolvimento da ferramenta de apoio à indexação, incluindo melhorias na visualização dos resultados de busca de assuntos bem como na entrada de dados.

1.1 Motivação

O estudo da recuperação da informação vem sendo conduzido, há vários anos, por diversos pesquisadores. Hoje é reconhecido como uma grande área onde o mundo acadêmico e profissional se encontram (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Nesse contexto, diversas organizações e instituições trabalham para lidar com a organização da informação textual e recuperação de seus significados. Como exemplo, podem ser citadas as bibliotecas, instituições que visam catalogar e indexar obras de um acervo, que geralmente, são objetos identificados a partir de elementos textuais.

Para facilitar sua organização, a biblioteca precisa de uma indexação eficaz, o que significa possibilitar que o usuário identifique em quais áreas se enquadra algum documento

ou obra do acervo. A falta de um sistema de indexação pode tornar o trabalho do bibliotecário demorado e ineficaz quanto à indexação e buscas no acervo, uma vez que as informações recuperadas podem trazer obras irrelevantes à busca ou pode ainda ser omissa quanto a resultados relevantes. Isso se deve ao fato de o bibliotecário, responsável pela catalogação e indexação de todas as obras, ter conhecimento limitado acerca de todas as áreas que o acervo abrange.

Mais especificamente, o catálogo da biblioteca do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), câmpus São João da Boa Vista (SBV) não apresenta um vocabulário controlado que mantenha a uniformidade da indexação e a recuperação das informações relacionadas à área de domínio da Informática, por exemplo. Um vocabulário controlado representa uma lista finita de termos (MCGUINNESS, 2003). Eles podem ser usados para consultas, e também podem estar categorizados em áreas ou domínios do conhecimento. Sua finalidade é fazer coincidir a linguagem do usuário com a do bibliotecário, ou seja, se a busca for feita pelo usuário a partir do assunto que não foi usado na indexação, o documento não será recuperado. Com o uso do vocabulário controlado, pretende-se diminuir a ocorrência desse fato ao se esclarecer uma incompatibilidade de terminologia que impeça que o documento/obra do acervo não seja recuperado. Um catálogo indexado a partir de uma ontologia – inventário de conceitos – criada a partir do vocabulário controlado viabiliza a apresentação do documento/obra que o usuário necessita.

Como solução para esse problema, uma pesquisa em Linguística Computacional tem sido realizada, utilizando-se do conceito de ontologias e de linguística de *corpus.* O *corpus* é constituído dos sumários dos livros da bibliografia básica do curso Tecnologia em Sistemas para Internet (TSI). Baseado nisso, foi desenvolvido um mapa conceitual implementado em uma estrutura de dados em grafos. No trabalho de Silva (2013), foi construído um mecanismo elementar de busca baseado em termos-chaves da área de informática, como, por exemplo, programação, redes, dados, software, entre outros. As relações semânticas estabelecidas para a recuperação de dados estão baseadas nos conceitos de definição – "é um" e de contiguidade – "é um tipo de". Por exemplo, C é uma linguagem de programação – "é um" e linguagem estruturada é um tipo de linguagem de programação – "é um tipo de" (SILVA, 2013, p. 66-67).

Silva (2013) justifica a opção por grafos, porque são estruturas que mantêm os conceitos das palavras e termos organizados de uma forma não hierárquica, ideal para a representação da ontologia em um mapa conceitual. O corpus da ontologia de Silva (2013)

valeu se dos sumários dos livros do curso TSI, na qual segundo o autor proporciona uma maior abrangência de termos em contraste com termos retirados apenas dos títulos das obras, proporcionando maiores possiblidades de relacionamento entre conceitos e recuperação de informações.

Nesta etapa de expansão do projeto, é considerado a concepção da ferramenta de indexação, na qual constituirá em um recurso de acesso aos termos (assuntos), podendo o bibliotecário realizar consultas, bem como inserir tais termos e seus relacionamentos.

1.2 Objetivos

O objetivo geral deste trabalho é conceber uma ferramenta de apoio à indexação para auxiliar o bibliotecário na organização do acervo, por assunto, em um catálogo, a partir de um sistema desenvolvido por Silva (2013).

Os objetivos específicos são:

- Permitir que o bibliotecário crie conceitos e relacionamentos dinamicamente;
- Permitir que o bibliotecário visualize os conceitos e relacionamentos em uma interface na forma de um grafo;

1.3 Organização deste trabalho

Este trabalho está divido em cinco capítulos.

No primeiro capítulo, são apresentados os objetivos, a motivação e a contextualização deste trabalho, bem com a organização da pesquisa.

O segundo capítulo consiste da pesquisa bibliográfica das áreas Linguística Computacional (ontologias, semântica e linguística de corpus), Ciência da informação (catalogação e indexação de bibliotecas) e Computação (estrutura de dados, grafos, banco de dados, IHC).

No terceiro capítulo, apresenta-se uma descrição da metodologia adotada para a obtenção dos objetivos propostos.

O quarto capítulo apresenta os resultados alcançados e, no quinto capítulo, observam-se as conclusões alcançadas e as perspectivas para trabalhos futuros.

2 Pesquisa Bibliográfica

Neste capítulo será apresentado o referencial teórico relevante a cada área em que esta pesquisa está inserida. Temas e termos inerentes à Linguística, à Computação e à Ciência da Informação serão abordados e suas relações com esta pesquisa serão demonstradas.

2.1 Tecnologia e Linguagem

Um diálogo entre uma pessoa e um computador em linguagem natural hoje é apenas uma suposição, mas, com os avanços tecnológicos em constante evolução e o desejo de melhorar a interação entre homens e máquinas, poderemos, em poucos anos, obter algo que torne isso realidade.

Tem sido crescente o número de trabalhos e aplicações que usam a linguagem natural para fazer a interação entre máquinas e pessoas, como comandos por voz, leitores de textos e *chatterbots* que interagem com humanos.

Para o desenvolvimento de pesquisas como essas, é comum buscar a contribuição da Linguística Computacional que, segundo Othero & Menuzzi (2005, p. 12), é "a área da linguística que se ocupa do tratamento computacional da linguagem para diversas finalidades práticas".

2.1.1 Linguística e Informática

Desde o surgimento de computadores até a popularização do computador pessoal (PC) a partir da década de 1980, grandes mudanças aconteceram. E elas são resultados do aumento do poder de processamento de um PC. Segundo a lei de Moore, os computadores dobram sua velocidade e complexidade a cada 18 meses (HAWKING, 2001, p. 167 apud OTHERO; MENUZZI, 2005, p. 15).

Essa evolução, ao longo dos anos, e o surgimento e aprimoramento de tecnologias ocasionam uma busca pela melhoria da interação entre máquinas e humanos. E essa grande tendência de humanizar o computador levou à criação de uma nova área de pesquisa, a HCI

(*Human Computer Interaction*), em português, Interação Humano-Computador (IHC). Essa área, segundo Carvalho (2000 apud OTHERO; MENUZZI, 2005, p. 18), é responsável pelos projetos de sistemas computadorizados de interação humana, sua implementação e avaliação. É também a área que estuda esses fenômenos, agrupando várias áreas, com o objetivo da excelência no campo das interfaces.

A interação entre homem e máquina e o constante aumento de informações também levaram ao surgimento de disciplinas sobre a recuperação da informação, tais como o Processamento de Linguagem Natural (PLN) e Mineração de Dados. No início, essas disciplinas surgiram da necessidade de buscar várias formas de conteúdo, como publicações científicas e registros bibliotecários, sendo na última década motivada principalmente pela explosão de conteúdo *online* na *World Wide Web* (MANNING; RAGHAVAN; SCHÜTZE, 2008).

Desta forma, pode se destacar na intersecção entre a Linguística e a Informática a Linguística Computacional, que de forma a contribuir para estudos que tornem o computador uma máquina mais inteligente, é uma das ciências que busca viabilizar uma interação em linguagem natural entre máquina e humanos. De acordo com Vieira & Lima (2001 apud OTHERO; MENUZZI, 2005, p. 22) é "a área de conhecimento que explora as relações entre linguística e informática, tornando possível a construção de sistemas com capacidade de reconhecer e produzir informação apresentada em linguagem natural".

A Linguística Computacional usa todos os conhecimentos da Linguística tradicional, como a sintaxe, a semântica, a fonética e a fonologia, a pragmática, a análise do discurso etc. Isso tudo é usado para o domínio do processamento das línguas naturais. Ela é dividida em duas subáreas:

- **Linguística de *Corpus***: responsável pelo armazenamento de amostras de linguagem natural, armazenados em bancos de dados, conhecido como "*corpora* eletrônico". Por exemplo, podem-se ter *corpora* de linguagem falada, *corpora* de linguagem escrita ou *corpora* específicos, como *corpora* de fala de crianças (OTHERO; MENUZZI, 2005, p. 23)[1].
- **Processamento de Linguagem Natural (PLN)**: responsável pela construção de programas capazes de processar, interpretar e gerar informação a partir de

[1] Veja, a esse respeito: SARDINHA, T. B. **Linguística de Corpus:** histórico e problemática. DELTA. São Paulo, v. 16, n. 2, p. 323-367, 2000.

> linguagem natural, ou de um *corpora*. Por exemplo, têm-se na PLN *software* como tradutores automáticos, *chatterbots*, *parsers* etc. (VIEIRA, 2002, p. 20 apud OTHERO; MENUZZI, 2005, p. 24).

Como se vê, a Linguística Computacional é a área do tratamento computacional da linguagem e das línguas naturais. Ela é uma área relativamente nova, tendo seus primeiros estudos a partir de 1950, em relação à linguística tradicional[2] (OTHERO; MENUZZI, 2005, p. 25).

Os estudos em Linguística Computacional cresceram a partir dos anos de 1950 e 1960, impulsionados pelo desenvolvimento de tradutores automáticos. De acordo com Grisham (1992 apud OTHERO; MENUZZI, 2005, p. 26),

> O potencial [dos computadores] para o processamento da linguagem natural foi reconhecido bem cedo no desenvolvimento de computadores, e trabalhos em linguística computacional – basicamente para tradução automática – começaram na década de 1950 em diversos centros de pesquisa. O rápido crescimento na área, no entanto, aconteceu principalmente a partir do final dos anos 1970.

O PLN está muito ligado a desenvolvimentos na área de inteligência artificial (IA) no desenvolvimento de *software* que utilizam linguagem natural para melhorar a interação entre homem e máquina. Isso é observado por McDonald & Yazdani (1990 apud OTHERO; MENUZZI, 2005, p.26): "a pesquisa em PLN pode proporcionar *insights* bastante úteis sobre processos e representações da linguagem na mente humana, apontando assim, para a verdadeira IA".

Podem-se verificar também outras áreas da linguística associadas à informática, como a fonética e a fonologia, em desenvolvimentos de aplicativos de PLN como reconhecedores e sintetizadores de fala, e sistemas de diálogos. Há hoje *software* capazes de digitar um texto falado pelo usuário ou executar ações a partir de comandos de voz, como o *Dragon NaturallySpeaking* 13[3].

---

[2] Esta teve seu surgimento considerado somente na chamada "linguística comparativo-histórico", no começo do século XIX, quando os gramáticos e filósofos gregos começaram seus primeiros estudos linguísticos há cerca de 2.400 anos. No Oriente, os estudos gramaticais datam de 2.500 anos, especialmente na Índia (OTHERO; MENUZZI, 2005, p. 25).

[3] Disponível em <http://www.nuance.com/for-individuals/by-product/dragon-for-pc/index.htm>. Acesso em 04/10/2015.

Na área da síntese de fala, há os programas capazes de produzir fala, dado um texto. Por exemplo, o programa *Talk it*[4] é capaz de ler em voz alta uma palavra digitada pelo usuário, bem como o popular tradutor *online* Google Tradutor[5]. Ainda na área de fonética e fonologia, podem-se destacar aplicativos capazes de permitir a interação entre ser humano e máquina por meio de diálogos em língua natural. Esses sistemas, ainda em desenvolvimento, têm grande potencial nas áreas de ensino à distância (EAD) e em programas de ensino de idioma (CALL – *Computer Assisted Language Learning*).

Esses tipos de aplicações podem, no futuro, melhorar a interação entre pessoas e máquinas e facilitar a inclusão de pessoas com deficiência, por meio de acesso a recursos disponibilizados pelos computadores.

No campo da sintaxe e da semântica, há sistemas que envolvem o entendimento ou a produção automática de frases. Esse é o exemplo do que faz um *chatterbot*, que é um *software* capaz de reconhecer frases digitadas pelo usuário e respondê-las. O primeiro *chatterbot* feito foi desenvolvido pelo pesquisador do *Massachusetts Institute of Technology* (MIT), Joseph Weinzenbaum, em 1966, e se chamava ELIZA[6]. Esse *chatterbot* é um programa de conversação, que se baseia em padrões para construir suas frases (OTHERO; MENUZZI, 2005, p. 31). Esse tipo de aplicativo vem sendo usado, hoje em dia, em atendimentos virtuais e em tutoriais educativos. Inclusive, existe uma competição anual entre *chatterbots*, criada em 1991 por Hugh Loebner, um filantropo norte-americano, em que vários *softwares* são testados baseados no Teste de Turing[7].

Na área de sintaxe e semântica, podem-se destacar também os programas de tradução automática, que foram os primeiros objetos de estudo da Linguística Computacional. Exemplos de programas de tradução automática são o *Power Translator*[8] e o *Systran Pro*[9],

---

[4] Disponível em <http://www.text2speech.com>. Acesso em 21/04/2013.

[5] Disponível em < https://translate.google.com/>. Acesso em 04/10/2015.

[6] Disponível em <http://nlp-addiction.com/eliza/>. Acesso em 08/10/2015.

[7] Turing foi um filósofo e matemático inglês que propôs em seu artigo "*Computing Machinery and Intelligence*", um teste que chamou de "jogo de imitação". No teste, um juiz se comunica com um interlocutor humano e o outro uma máquina, através de perguntas e respostas, tendo que julgar qual é humano e qual é máquina. Se um software for distinguido como humano deve ser atribuído a ele uma inteligência humana (HODGES, 2001, p. 45 apud OTHERO; MENUZZI, 2005, p. 34).

[8] Disponível em <http://www.lec.com/power-translator-software.asp>. Acesso em 22/04/2013.

[9] Disponível em <http://www.systransoft.com>. Acesso em 22/04/2013.

tradutores abrangentes, ou seja, sem restrições de léxico, gênero ou assunto do texto (OTHERO; MENUZZI, 2005, p. 37). Há também aplicativos como *parsers*, geradores de resumos, corretores ortográficos e gramaticais etc.

A Linguística Computacional traz importantes contribuições para esta pesquisa, tais como o entendimento de conceitos como ontologia, mapas do conhecimento e tesauros, e a relação entre eles. Esse entendimento foi essencial para que se verificasse a aderência da estrutura de mapa do conhecimento para esta pesquisa em particular.

2.2 O desafio da catalogação e indexação

No transcorrer da história, novas tecnologias foram sendo criadas e aperfeiçoadas. Dentre todas as áreas que sofreram modificações, pode-se notar também a Ciência da Informação (CI), que, segundo Sousa e Fujita (2012), tem como objeto de estudo a própria informação.

A forma de adquirir e passar informações tem evoluído desde o discurso oral, passando pela escrita, até os dias atuais, no formato eletrônico.

A Ciência da Informação tem como base a produção e uso de informação, levando a uma reflexão de práticas de questões como organização, representação e uso (SOUSA & FUJITA, 2012).

Como meios de organização de informações em uma biblioteca, serão destacados os catálogos e sua evolução, desde o formato manual de fichas ao formato *online*. Como esta pesquisa visa a concepção de uma ferramenta de indexação, que se insere dentro do contexto de catalogação, é importante observar as diferentes definições e mudanças adquiridas ao longo do tempo, que contribuíram para a evolução desse mecanismo de representação e recuperação da informação.

Em uma biblioteca, os catálogos auxiliam o usuário no acesso a documentos pela descrição temática, e/ou pela descrição física, e também direciona a localização física na estante.

Segundo Foskett (1973, p.164 apud SOUZA; FUJITA, 2012) “um catálogo de biblioteca destina-se a registrar o acervo da biblioteca [...]”. Seguindo ainda a definição, tem-se que “um catálogo é uma série ordenada de referências ou de inscrições que registram as peças de uma coleção” (GUINCHAT; MENOU, 1994, p. 197 apud SOUZA; FUJITA, 2012).

Quando sistematizada pela primeira vez, a catalogação tinha apenas que revelar os itens de uma coleção, uma vez que, devido à baixa quantidade de publicações, seu conteúdo podia ser conhecido (SHERA; EGAN, 1969, p. 11 apud SOUZA; FUJITA, 2012).

Mas, com o crescente número de publicações e os diversos meios de divulgação, os catálogos tiveram uma mudança no seu foco de uso, e, de acordo Martinho e Fujita (2011 apud SOUZA; FUJITA, 2012), passaram de mero depósito a valiosa ferramenta de recuperação de informações.

Os catálogos impressos são, por padrão, organizados pelas fichas de catalogação, que podem estar organizadas das seguintes formas, segundo Shera e Egan (1969, p. 15 apud SOUZA; FUJITA, 2012):

- Por autor;
- Por título;
- Pela forma física;
- Pelo período;
- Pelo lugar;
- Pelo idioma;
- Por características materiais e
- Por assunto.

Assim, estabelecendo alguns desses pontos para a catalogação dos documentos, serão formados os registros, que estarão ordenados e que poderão ser consultados, sendo possível estar disponível, de acordo com Ferraz (1991, p.91 apud SOUZA; FUJITA, 2012) “[...] a ideia do material a que se refere, sem necessidade de acesso físico a esse material”.

Os catálogos podem ser apresentados de várias formas, entre as quais se podem destacar:

- Manual: publicados em forma de livros;
- Impressa: apresentados em forma de listas;
- Automatizados: registrados em formas legíveis por computador.

É possível notar, de acordo com Ferraz (1991, p. 99 apud SOUZA; FUITA, 2012), que, desde o início da década de 1990, a evolução da informática iria influenciar o modo de catalogação.

> Os catálogos impressos permanecem até a virada do século, quando os catálogos de fichas tornaram-se mais comum; e desde a forma de fichas, a maioria das bibliotecas deste século o utilizam para o registro de suas coleções. Gradativamente, os catálogos eletrônicos vêm substituindo os catálogos em fichas [...].

Essa automatização para os catálogos eletrônicos, também denominados *online* ou OPAC (*Online Public Access Catalog*) (SOUZA; FUJITA, 2012) trouxeram grandes benefícios, principalmente na recuperação de informações, como é destacado por Araújo e Oliveira (2005, p.39 apud SOUZA; FUJITA, 2012) "os catálogos *online* oferecem várias vantagens no acesso à informação que os impressos não têm, como a rapidez na busca, uma maior possibilidade de padronização das informações etc.".

Os catálogos eletrônicos possuem todas as funções comuns das práticas bibliotecárias de busca, como consultas, empréstimo e processamento técnico, além de realizar pesquisas por autor, título e assunto, disponibilizando funções adicionais e com maior rapidez. Mesmo com melhorias das operações bibliotecárias, a função do tratamento de informações e da inserção dos dados no catálogo é responsabilidade do bibliotecário (SOUZA; FUJITA, 2012).

Uma função importante do bibliotecário está no tratamento das informações, em que ele deve compreender a prática de catalogação como indexação, atuando como indexador ao analisar o assunto, identificando e selecionando os conceitos que melhor representam o conteúdo do documento (FUJITA; RUBI; BOCATTO, 2009 apud SOUZA; FUJITA, 2012).

Ainda, segundo Sousa e Fujita (2014), a indexação é o processo de análise de assunto de um documento através da leitura documentária. Além disso, de acordo com os autores, há diversas normas que definem os princípios de indexação. Dentre delas, destacam-se os Princípios de Indexação do sistema UNISIST (*United Nations International Scientific Information System*), definido pela UNESCO (*United Nations Educational, Scientific and Cultural Organization*) (UNISIST, 1981). Este documento descreve que a indexação consiste de dois estágios, sendo eles o "estabelecimento dos conceitos tratados num documento, isto é, o assunto", e a "tradução dos conceitos nos termos da linguagem de indexação". Ainda no primeiro estágio da indexação, o de determinação do assunto, ele é divido em outras três etapas: compreensão do conteúdo do documento como um todo, identificação dos conceitos que representam esse conteúdo, e seleção dos conceitos válidos para a recuperação. Outra norma que define princípios para a indexação é a NBR 12676/1992 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 1992). Esta, por sua vez, descreve o processo de indexação em três estágios, quais sejam: "a) exame do documento e estabelecimento do

assunto de seu conteúdo; b) identificação dos conceitos presentes no assunto; c) tradução desses conceitos nos termos de uma linguagem de indexação". Ambas as normas definem que o primeiro passo na indexação de um documento é o estabelecimento dos assuntos. Com relação às técnicas para a extração dos assuntos, ambas as normas convergem para a ideia de examinar elementos do texto, como, por exemplo, o título, a introdução, ilustrações e tabelas entre outros, tendo na NBR 12676/1992 também o aspecto de considerar o sumário (UNISIST, 1981; ABNT, 1992).

Têm-se também, segundo Guedes e Dias (2010, apud ROWLEY, 2002, p. 169), diferentes formas de inventariar a descrição temática dos documentos:

- Linguagem controladas de indexação: os termos usados na indexação são bem definidos. Têm-se dois tipos, as linguagens alfabéticas de indexação (*e.g.* tesauros), e os sistemas de classificação (*i.e.* códigos ou notações);
- Linguagens naturais de indexação: qualquer termo no documento pode ser usado na indexação;
- Linguagens livres de indexação: não há limitação em relação a quais termos podem ser usados na indexação.

Como pode ser observado, a primeira forma é mais sistemática, no que se assemelha a essa pesquisa que tem como proposta conceber uma forma para que se crie uma fonte central de termos para uso na indexação.

Ainda, a indexação pode ser também orientada pelos diversos agentes executores da ação (GUEDES; DIAS, 2010 apud RAFFERTY; HIDDERLEY, 2002, p. 169):

- Indexação orientada por especialistas: a indexação é feita por intermediários, os especialistas;
- Indexação orientada pelo autor: a indexação é realizada pelo autor da obra/documento;
- Indexação orientada pelo usuário: a indexação apresenta um nível maior de interação com os agentes da comunidade, como os usuários.

A indexação orientada por especialistas, apesar de ser mais demorada, geralmente é também mais precisa. A proposta desta pesquisa mais se assemelha a essa, porém como os termos e seus relacionamentos ficarão à disposição para futura referência, em alguns momentos não será mais necessária a interferência de agentes intermediários.

Do ponto de vista da dificuldade do bibliotecário, como responsável pela indexação, de possuir o conhecimento específico necessário em todas as áreas que um acervo abrange, é notável o objetivo desta pesquisa de conceber um mecanismo de apoio na identificação de assuntos para a indexação de uma obra.

2.3 Ontologias, Tesauros e Mapas do Conhecimento

Ontologias, tesauros e mapas do conhecimento são termos e conceitos bastante utilizados no campo da linguística e da ciência da informação como formas de representar o conhecimento, mas possuem algumas diferenças que devem ser levadas em conta. Uma ontologia, de forma simplista, é um conjunto de tipos de objetos, suas propriedades e seus relacionamentos (CHANDRASEKARAN et al 1999); um tesauro é um inventário de termos e relacionamentos fixos (SALES; CAFÉ, 2009); e um mapa do conhecimento é usado tipicamente para navegação conceitual e visualização (MOREIRO; CUADRADO; MORATO, 2003). Além disso, eles apresentam diferentes origens e propósitos.

2.3.1 Ontologias

A Ciência da Informação sempre usou e desenvolveu mecanismos de representação do conhecimento. Ao longo do tempo, e com o constante desenvolvimento das TICs (Tecnologias da Informação e Comunicação), modelos clássicos e já estabelecidos precisam ser repensados, devido a novos processos de representação, organização, distribuição e recuperação das informações (RAMALHO, 2010, p.73).

Devido a essa necessidade, a TIC, junto com a Ciência da Computação, teve, ao longo do tempo, grandes desafios para a representação e recuperação do conhecimento. De acordo com Ramalho (2010, p.52), foi preciso criar tecnologias que descrevessem melhor os recursos de informações, o que deu origem às chamadas tecnologias semânticas.

As tecnologias semânticas são caracterizadas como linguagens que permitem passar de simples representações sintáticas para a descrição computacional de aspectos semânticos dos documentos, permitindo o uso de ontologias (RAMALHO, 2010, p.52).

De acordo com Carvalho e Carvalho (1975 apud RAMALHO, 2010, p. 52), “semântica é o estudo do significado das palavras considerado como o componente de sentido e de interpretação de sentenças e enunciados”.

O termo ontologia é utilizado em várias áreas, sendo bastante controversa sua definição devido a essa multidisciplinaridade (GUARINO, GIARETA, 1995 apud DI FELIPPO, 2008).

As ontologias também são caracterizadas por armazenar um conceito de forma lexicalizada, ou seja, expresso por uma ou mais palavras de uma língua (DI FELIPPO, 2008). Elas também são importantes porque "propiciam descrições concisas e desambiguas sobre conceitos e relacionamentos em um domínio de interesse" (LEÃO; REVOREDO; BAIÃO, 2011).

O consórcio W3C (*World Wide Web Consortium*), que é o órgão regulador de padrões para a Web, define ontologia como "a definição dos termos utilizados na descrição e na representação de uma área de conhecimento" e, de maneira simples, descreve que ontologias devem ter descrições para os seguintes conceitos (BREITMAN, 2010, p.30):

- Classes (ou "coisas") nos diferentes domínios de interesse;
- Relacionamento entre as classes e
- Propriedades (ou atributos) que as classes devem ter.

Ontologias também podem ser definidas com um rol de tipos de objetos, propriedades de objetos e dos relacionamentos entre eles em um domínio específico do conhecimento, tendo grande valor por transmitirem os conceitos independente dos termos do vocabulário do domínio em que está inserida (CHANDRASEKARAN et al 1999).

Segundo Staab (2006), o desenvolvimento de ontologias traz vantagens como fazer suposições explícitas em um domínio e compartilhar um entendimento consistente do significado das informações.

Conforme ilustrado na Figura 1, e de acordo com Staab (2006), uma ontologia apresenta um inventário de conceitos relacionados entre si. Além disso, nota-se que, em uma ontologia, regras podem ser criadas, a partir das quais é possível fazer inferências. Por exemplo, na figura pode ser visto "P -> *writes* -> D -> *is_about* -> T" (Pessoa P escreve Documento D que é sobre Tópico T), então infere-se que "P *knows* T" (Pessoa P sabe Tópico T).

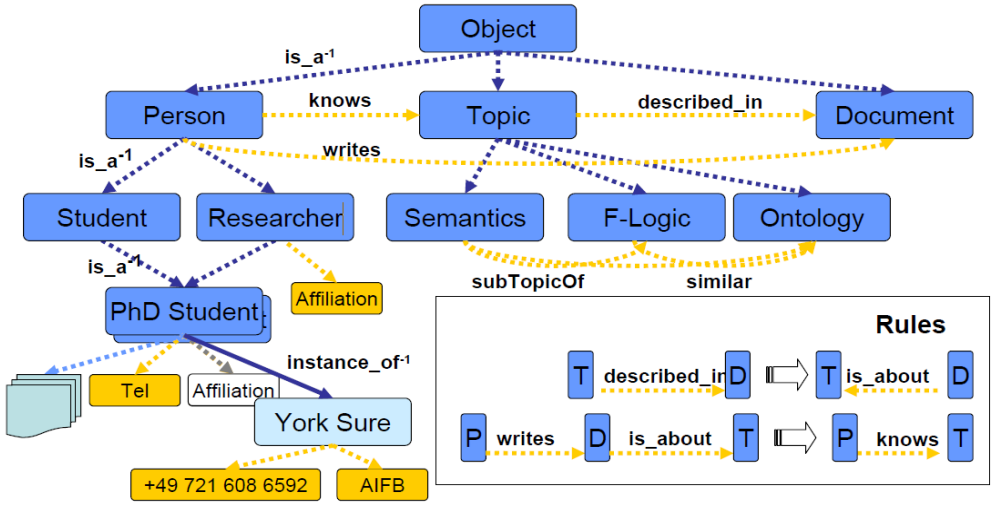

Figura 1 - Ontologia
(STAAB, 2006)

Em suma, ontologias são mecanismos poderosos de representação do conhecimento e suas relações semânticas e de inferência permitem interessantes formas de recuperação de informações.

2.3.2 Tesauro

O termo tesauro tem origem do grego e do latim e significa “tesouro”. Seu uso começou a ser conhecido em Londres, no ano de 1852, com a publicação do “*Thesaurus of English words and phrases*” de Mark Roget, que era definido pelo autor como um dicionário de palavras organizado pelo significado, e não por ordem alfabética (MOREIRA, 2003).

O termo sofreu diversas evoluções de significados ao longo do tempo, mas o mais comum na Ciência da Informação, de acordo com Moreira (2003, p.23), é que um tesauro constitui-se de uma lista de palavras, em que cada uma é seguida por outras relacionadas a ela.

Outra definição, segundo Currás (1995, p. 24 apud MOREIRA, 2003), é que tesauro envolve um vocabulário controlado dentro de um domínio, onde os termos são relacionados no momento do uso, com o fim de documentação.

Na Figura 2 pode ser visto a estrutura do tesauro segundo Staab (2006), onde é notável sua diferença em relação às ontologias no número de relacionamentos, e portanto, na possibilidade de descrição conceitual.

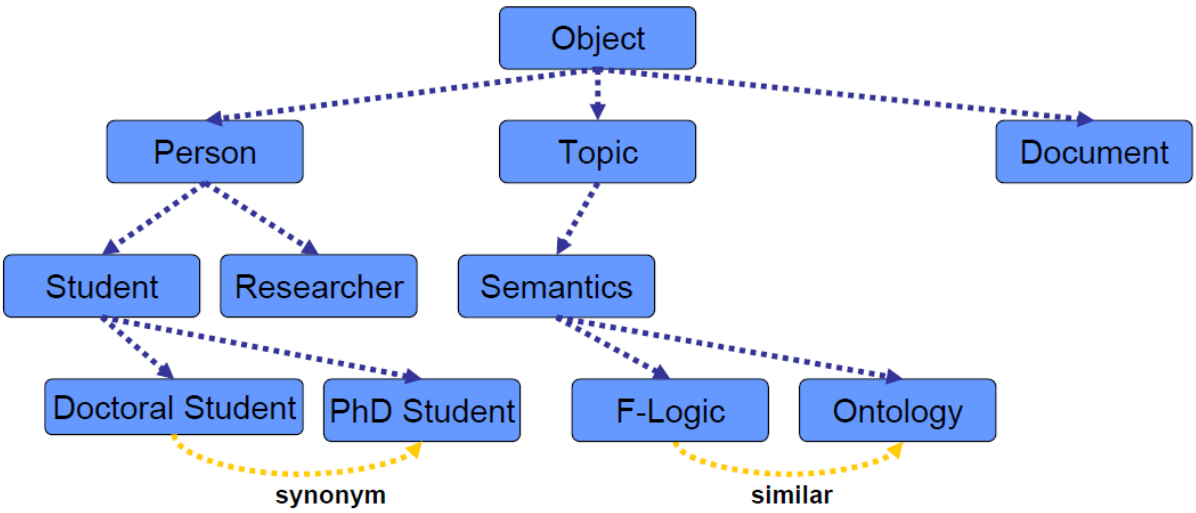

Figura 2 – Tesauro
(STAAB, 2006)

Essa característica dos tesauros faz com que auxiliem na área da documentação, indexação e na recuperação de informações. Ele pode ser usado no auxílio de consultas ou no momento de classificação por parte do indexador. Também tem a função de auxiliar na delimitação dos termos a serem usados, tanto para indexação quanto para a busca, bem como permitir a introdução de novos termos em sua estrutura (MOREIRA, p. 29, 2003).

Para Moreira (2003), os tesauros se diferem das ontologias por apresentarem diferentes origens e propósitos. O tesauro tem como objetivo auxiliar na indexação e na busca de documentos, ao passo que é a ontologia que descreve objetos e suas relações (MOREIRA, 2003).

Há, ainda, os vocabulários controlados, que também descrevem objetos, mas não descrevem as suas relações. Um exemplo de vocabulário controlado é o da USP[10], que apresenta, como uma de suas categorias, a ciência da computação. Há uma lista de termos que definem assuntos e/ou subáreas do domínio da Informática. Esse vocabulário pode ser utilizado por bibliotecários, mas ele não define relacionamentos, como na ontologia, o que torna a busca menos eficaz quanto à recuperação de informações específica, o que não permite um refinamento. Por exemplo, a área de Inteligência Artificial pode estar ligada à Linguística, a partir de uma relação "um tipo de" ou "é parte de/faz parte de", mas, em um vocabulário controlado, ela certamente estará caracterizada dentro da categoria Ciência da Computação, sem nenhum relacionamento com a Linguística. Caso alguém queira ver alguma área da Linguística que faça interface com a Informática, certamente, não encontrará, no

[10] Disponível em <http://143.107.73.99/Vocab/Sibix652.dll/ARV?Hier=CE610>. Acesso em 02/06/2013.

vocabulário controlado, Inteligência Artificial. Contudo, poderá encontrar esse conceito em uma ontologia.

2.3.3 Mapas do Conhecimento

Mapas do conhecimento têm por objetivo organizar elementos e notações utilizadas para estruturar a informação através de uma rede de relacionamentos semânticos. Além disso, eles permitem a navegação conceitual (MOREIRO; CUADRADO; MORATO, 2003).

De acordo com Pepper (2007), os mapas do conhecimento são a combinação entre modelos semânticos básicos e de associações de índices. Segundo os autores, com os conceitos de "tópico-ocorrência" e "tópico-associação", os mapas do conhecimento preenchem um espaço que há entre a representação do conhecimento e do gerenciamento da informação.

Os estudos sobre mapas do conhecimento começaram na década de 1990, quando o principal objetivo era o de organizar índices impressos (MOREIRO; CUADRADO; MORATO, 2003). Seu estudo e difusão levaram ao desenvolvimento de um padrão pela *International Organization for Standardization* (ISO) em 2003, a ISO 13250. Os mapas do conhecimento[11] se parecem muito com o formato RDF (*Resource Description Framework*), um modelo para representação de informações sobre recursos na *World Wide Web* desenvolvido pela W3C, agência internacional reguladora de padrões da Internet. Nesse sentido, um grupo de estudos chamado *Semantic Web Best Practices and Deployment Working Group* iniciou um trabalho para propor práticas de integração entre o modelo ISO 13250 e a especificação RDF (PEPPER *et al*, 2006). Esses estudos fomentam o desenvolvimento dos mapas do conhecimento, e ainda fortalecem sua importância como objetos de estudo da ciência da informação e da informática.

Segundo Pepper (2007), os mapas do conhecimento têm três estruturas básicas:

- **Tópico**: pode ser qualquer "coisa", independente se isso realmente existe ou tem qualquer outra característica específica. O autor ainda diz que se pode pensar o tópico como um assunto, ou além, uma ideia. O tópico é o que se refere ao objeto ou nó no mapa do conhecimento, que representa o assunto que

[11] Durante esta pesquisa foram encontrados diversos termos que representam o mesmo que Mapas do Conhecimento, como *Topic Maps* e *Conceptual Maps*. Sendo assim, nesta pesquisa será usado o termo Mapa(s) do Conhecimento.

está sendo referenciado. Os dois termos (tópico e assunto) ainda podem ser usados indistintamente. Tópicos podem também ter vários tipos e nomes (como nome base, nome de visualização, etc).

- **Associação**: é o relacionamento entre dois ou mais tópicos. Uma associação, assim como um tópico, também pode ter tipos.

- **Ocorrência**: um tópico pode estar ligado a um ou mais recursos informacionais que se considerar relevantes ao tópico. Esses recursos são as ocorrências de um tópico. Por exemplo, um artigo sobre um tópico ou uma imagem pode ser uma ocorrência, algum recurso informacional relevante àquele tópico.

Na Figura 3, pode-se visualizar a estrutura de um mapa do conhecimento, onde "*London*", "*UK*", e "*Europe*" são os tópicos, e as ocorrências estão na área descrita como "*Resource Space*". Os losangos representam as associações entre os tópicos.

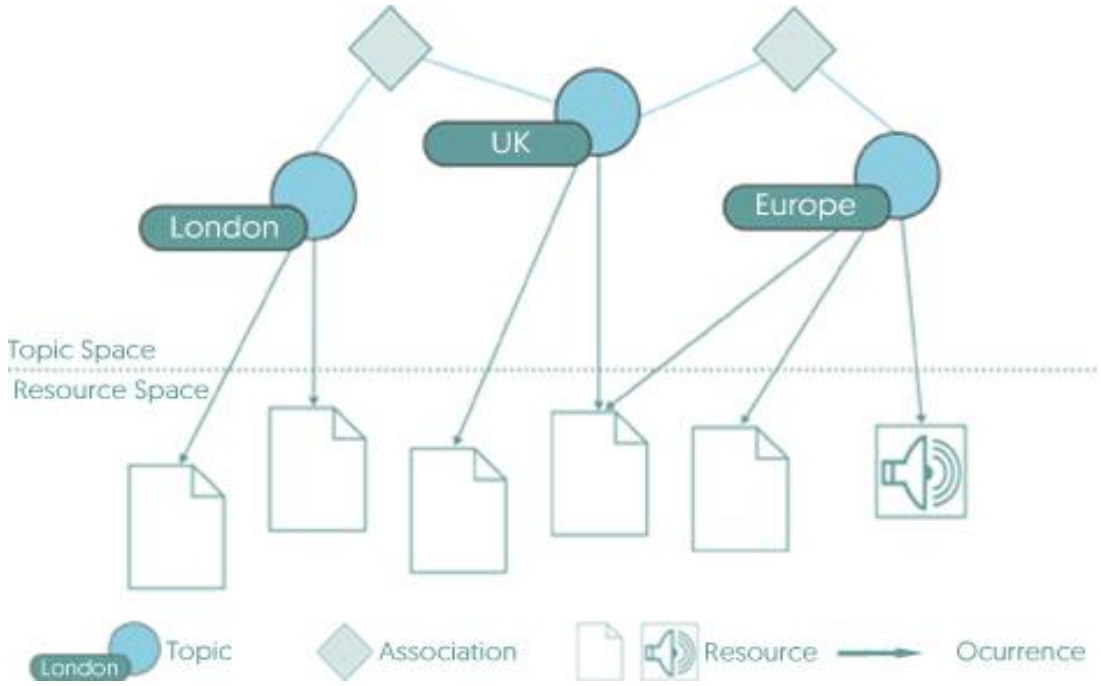

Figura 3 – Mapa do Conhecimento
(AHMED; MOORE, 2005)

2.3.4 Diferença entre ontologias, tesauros e mapas do conhecimento

Com a evolução dos mecanismos de representação da informação, novas tecnologias emergiram. É possível notar como estratégias que antes eram amplamente utilizadas não são mais suficientes, como consideram Dabrowski, Synak e Kruk (2009 apud RAMALHO, 2010, p. 73): o uso de palavras-chave não são suficientes para representar de forma significativa o conteúdo dos documentos, sendo então necessário o uso de descrições semânticas, que possibilitam descrever o modo como as pessoas consideram os conteúdos. Novos estudos se iniciaram nessa área, como a Web Semântica, que utiliza o conceito de ontologias e formatos de representação para criar um novo modo de recuperação de informações na Internet (BREITMAN, 2010).

Com base em Cunha (2000 apud RAMALHO 2010, p. 74), é possível verificar que meios de descrição dos registros e dos conteúdos informacionais não são mais suficientes para atender às novas necessidades. Na computação, isso pode ser observado com os novos modelos de bancos de dados que surgiram devido às mudanças na quantidade e na importância das informações.

Valendo-se das ilustrações de Staab (2006), na Figura 4, é possível identificar as principais diferenças entre os modelos de representação do conhecimento descritos neste trabalho. Os tesauros são estruturas que apresentam relacionamentos fixos, e possibilitam identificar similaridades e sinônimos. Os mapas conceituais já são mais avançados, porque têm um maior número de relacionamentos. E, por fim, as ontologias são estruturas com grande variedade e número de relacionamentos, e permitem a descrição de regras.

Como neste trabalho o objetivo é criar um mecanismo de apoio ao bibliotecário para se encontrar os assuntos para a indexação, estruturas como os mapas do conhecimento melhor se aderem a essa problemática. Os tesauros, por serem estruturas rígidas, não fornecem a flexibilidade de relações necessárias para construir tal solução. Por outro lado, as ontologias são estruturas mais complexas, que permitem o uso de técnicas de inferência mais avançadas, em que informações podem ser inferidas por meio das regras. Este trabalho limita-se apenas a habilitar o usuário, no caso o bibliotecário, a inserir as informações (assuntos e seus relacionamentos) e posteriormente visualizá-las e não tem por objetivo desenvolver uma busca “inteligente”, com o uso de inferência, por exemplo.

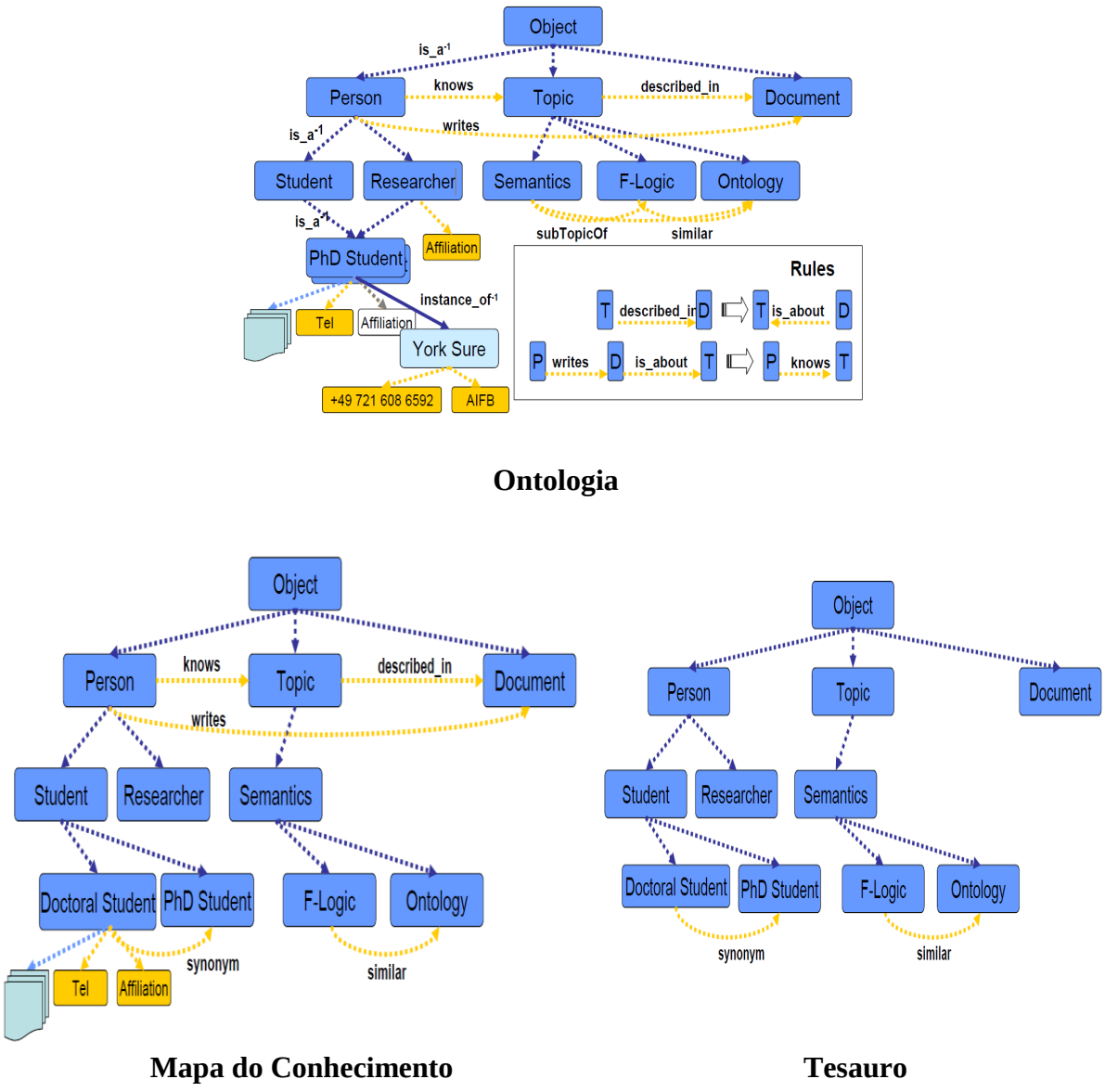

Figura 4 - Comparação das estruturas
Adaptado de Staab (2006)

O uso de mapas do conhecimento pode ser visto no trabalho de Lavik e Nordeng (2004), que consiste em usar mapas do conhecimento em instituições educacionais. Alunos podem criar uma conta individual no sistema *BrainBank Learning* (BBL) e, com o uso de ferramentas intuitivas, segundo os autores, criar tópicos e conectá-los em associações com nomes e/ou descrições a fim de aprenderem os conceitos e seus conteúdos, bem como eles se relacionam. Outro projeto que se beneficiou do uso de mapas do conhecimento foi o da cidade de Bergen, na Noruega. De acordo com Garshol (2007), o projeto se baseou em desenvolver

um novo portal (*online*) da cidade, para acesso dos cidadãos, utilizando-se o conceito de mapas do conhecimento. O projeto envolvia criar uma interface principal para a interação dos usuários e mover vários serviços para o ambiente *online*. Ainda segundo Garshol (2007), o mapa do conhecimento tinha 7.200 tópicos, 16.000 associações e 17.000 ocorrências, entre muitos outros artigos e serviços, e, mesmo assim, o portal obteve uma *performance* satisfatória.

2.4 Bancos de Dados

Como esta pesquisa visa à recuperação e ao tratamento de informações em um ambiente informatizado, é preciso, de alguma forma, armazená-las para que, posteriormente, possam ser acessadas, modificadas e recuperadas. Para isso, torna-se necessário o uso de um grande recurso da informática, os Bancos de Dados.

Bancos de dados são sistemas computadorizados que organizam e armazenam dados, análogos a um armário de arquivamento, mas implementados computacionalmente e utilizados para a manutenção de registros (DATE, 2003). Uma simples definição, segundo Date (2003), é que

> [...] um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar.

É importante notar que os dados estão armazenados para o uso por parte de algum usuário que fará operações sobre eles, como busca e atualização.

Há diversos tipos de bancos de dados, que surgiram, em sua grande maioria, a partir da segunda metade do século XX e estão evoluindo até hoje. Serão destacados os grupos mais conhecidos e utilizados, bem como um destaque maior na área de grafos e no banco de dados Neo4j, que será utilizado nesta pesquisa para modelar o mapa de conhecimento da ferramenta proposta.

Os bancos de dados em grafos surgiram por volta dos anos de 1960. Antes da supremacia dos bancos de dados relacionais, houve uma discussão na área de

desenvolvimento de *software* sobre qual modelo usar entre os bancos de dados hierárquicos[12] e bancos de dados em grafo, por vezes chamados de banco de dados de rede (POLLOCK, 2010, p. 102).

O modelo relacional surgiu no início dos anos 1970 e é utilizado pela maioria dos SGBDs (Sistemas Gerenciadores de Banco de Dados), como o SQL *Server*, Oracle, PostgreSQL, MySQL entre outros. Nesse modelo, os registros dos dados estão estruturados em linhas (tuplas) e colunas (atributos) organizadas em tabelas. Os dados podem estar relacionados pelas tabelas (BRITO, 2010). Esse modelo se baseia na organização de dados em conjuntos e foi proposto em 1970, por Ted Codd (POLLOCK, 2010).

Foi adicionado ao modelo relacional uma linguagem de definição, manipulação e consulta de dados chamada de SQL (*Structured Query Language*), que é uma linguagem declarativa inspirada na álgebra relacional com uma simplicidade e expressão que a tornaram líder junto com o modelo relacional (BRITO, 2010).

Os bancos de dados relacionais têm algumas vantagens que possibilitaram sua grande aceitação até os dias atuais, como a integridade, recuperação a falhas, a simplicidade da linguagem SQL entre outros (BRITO, 2010).

Mesmo com todas essas vantagens, o crescimento de novas aplicações e o grande volume de dados tornam questionável o uso do modelo relacional. Por exemplo, segundo Brito (2010), o Google atingiu um volume de dados de *petabytes*, o que equivale a 10^{15} *bytes*. Nesse cenário, o maior problema está em manter esse modelo com tamanho crescimento, ou seja, a escalabilidade do sistema.

É interessante notar também, conforme Miguel e Carneiro (2013), que, ao longo da história, o modelo relacional sofreu poucas modificações em relação à programação, que passou do paradigma da programação estruturada para a orientada a objetos até a programação orientada a aspectos. Segundo os autores, essa diferença resulta em modelos de dados não aderentes à estrutura do *software*.

Outros problemas começam a ser observados quando um sistema começa a ter muitos usuários, o que acaba prejudicando o desempenho da aplicação. Nessa situação, poderia ser tomado como solução aumentar o poder de processamento do servidor ou aumentar o número de servidores, ou até mesmo, escalar o banco, distribuindo-o em diversos computadores, o

[12] Esses modelos surgiram antes dos bancos de dados relacionais e modelam os dados com relações do tipo pai e filho, estruturados em um formato de árvore (POLLOCK, 2010, pg 100-101).

que não é uma tarefa simples. A partir dessa abordagem, feita por Brito (2010), justifica-se o fato de o foco começar a ser concentrado por soluções do tipo não relacionais.

Devido a essas limitações do modelo relacional, foram propostas novas soluções. A estrutura rígida desse modelo passou a ser questionada e as soluções propunham eliminar ou minimizar essa estruturação. Isso levou a soluções simples, como o gerenciamento de arquivos, que davam a impressão de uma volta no tempo (BRITO, 2010).

Mas essa perda de regras e rigidez trazia os benefícios de um alto nível de paralelismo e distribuição de sistemas, suprindo a necessidade de escalabilidade, além de ganhos em *performance* (BRITO, 2010; MIGUEL & CARNEIRO, 2013).

Em 1998, surgiu o termo NoSQL, em uma solução de banco de dados que não tinha uma interface SQL, mas ainda era baseado na estrutura relacional. Mas, a partir daí o termo passou a representar soluções que eram uma alternativa ao modelo relacional, e se tornou a abreviação de *Not Only* SQL (não apenas SQL) (BRITO, 2010).

A história dos bancos de dados NoSQL é recente e suas primeiras implementações surgiram na primeira década dos anos 2000.

Apesar das semelhanças entre esses bancos de dados, tais como a de serem livres de esquema, promover alta disponibilidade e maior escalabilidade, os bancos de dados NoSQL são agrupados em quatro diferentes categorias. São elas, segundo Brito (2010) e Miguel e Carneiro (2013):

- Sistemas baseados em armazenamento chave valor: há uma coleção de chaves únicas e de valores associados a essas chaves. É um exemplo o Amazon Dynamo, lançado em 2007.
- Sistemas orientados a documentos: as informações são armazenadas em documentos, que têm coleções de atributos e valores, sendo que um atributo pode ter um valor simples ou outro documento. O documento armazena informações em um formato específico, como o JSON (*JavaScript Object Notation*). São exemplos o Apache CouchDB, lançado em 2008 e o MongoDB lançado em 2009.
- Sistemas orientados a coluna: o paradigma é de orientação a colunas (atributos) e não mais a registros (linhas ou tuplas). É recomendado para ambientes analíticos (OLAP – *Online Analytical Processing*). São exemplos o Apache Cassandra, lançado em 2008 e o BigTable do Google, um dos primeiros bancos de dados NoSQL, lançado em 2004.

- Sistemas baseados em grafos: os dados são armazenados em nós de um grafo e os relacionamentos são representados pelas arestas do grafo. São exemplos o InfoGrid e o Neo4j.

Para Brito (2010), os principais critérios na escolha de um banco de dados estão relacionados ao escalonamento, consistência dos dados e disponibilidade. Em sua análise comparativa dessas características, observa-se que uma solução relacional leva vantagem em relação à consistência dos dados, além de ser mais madura devido a sua grande utilização desde os anos 1970, mas perde para o modelo NoSQL nos critérios de escalonamento e de disponibilidade.

As soluções surgiram e estão evoluindo ao longo do tempo, mas, como destacam Brito (2010) e Miguel e Carneiro (2013), é preciso antes ser feita uma análise do cenário do problema por parte dos envolvidos no projeto para a escolha da melhor opção.

Como destacado pelos autores, a escolha de um banco de dados deve observar o tipo de projeto. Nesta pesquisa, com a utilização do conceito de mapa do conhecimento, a melhor forma de fazê-lo é utilizando o conceito de grafos. Desta forma, foi escolhido um banco de dados baseado nessa estrutura, o Neo4j[13]. Não é do intuito desta pesquisa analisar uma grande quantidade de dados, como por exemplo é feito no *Big Data*[14]. A ferramenta proposta visa a atender à necessidade de inserção de dados de um ou mais domínios organizados semanticamente, de forma que eles possam ser recuperados pelo usuário.

2.4.1 Grafos, Banco de Dados em Grafo e o Neo4j

Um grafo é uma forma de representar um conjunto de objetos (vértices) que mantém um conjunto de relacionamentos entre dois vértices (arestas), ou seja, grafos representam relações entre pares de objetos de algum domínio. Isso pode ser representado de forma abstrata como um grafo G, onde os elementos de um conjunto V não vazio (os vértices) estão ligados por arestas (E) que relacionam pares de vértices (GOODRICH & TAMASSIA, 2013).

---

[13] Disponível em < http://neo4j.com/>. Acesso em 08/10/2015.

[14] *Big Data* é um termo usado para expressar que uma quantidade de informação é tão grande que ela não pode ser processada e analisada por meios e ferramentas tradicionais. Tal informação geralmente se apresenta de forma primitiva, semiestruturada ou sem nenhuma estruturação por exemplo (ZIKOPOULOS & EATON, 2011).

Acredita-se que o primeiro resultado a respeito da teoria dos grafos foi apresentado pelo matemático e físico suíço Leonhard Euler (1707-1783) em 1736, com o problema das “Sete Pontes de Konigsberg”, que consistia em uma ilha cercada por quatro regiões e sete pontes. Euler provou, usando a teoria dos grafos (Figura 5), que não era possível percorrer todas as pontes sem repetir nenhuma. Na representação, as regiões se tornaram os vértices e as pontes, as arestas (LUCCA, 2012, p. 19).

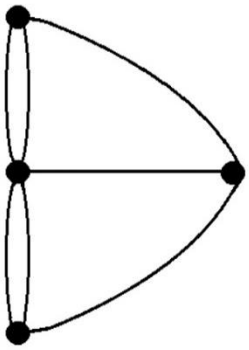

Figura 5 - Grafo Pontes de Konigsberg
Adaptado de Lucca (2012)

Grafos também foram utilizados para demonstrar a solução do problema proposto em 1853, por Francis Guthris, que consistia em saber qual o número mínimo de cores necessárias para pintar um mapa sem que dois municípios adjacentes tivessem a mesma cor. Nesse caso, as regiões são os vértices e as fronteiras, as arestas. Somente em 1976, essa pergunta foi respondida por uma demonstração em computador proposta por Kannet Appel e Wolfgang Haken, em que eles usaram teoremas matemáticos reduzindo um mapa maior em menores. Resolvendo os mapas menores, eles foram expandidos gradativamente até atingir o mapa original maior, e provou-se que qualquer mapa pode ser colorido com quatro cores, como mostrado na Figura 6, no mapa dos Estados Unidos da América (LUCCA, 2012).

Figura 6 - Grafo - Pintar Mapa com 4 cores
Fonte: Lucca (2012, p. 20)

Pode-se observar que o problema de colorir o mapa foi respondido computacionalmente somente em 1976, cerca de uma década após o surgimento dos bancos de dados em grafos, que, segundo Pollock (2010), ocorreu na década de 1960.

É interessante notar, nesse contexto histórico, que a resposta a um problema cotidiano se deu com o uso da teoria dos grafos. E que, a partir desse momento, diversos modelos de armazenamento de dados surgiram, e, como já foi abordado, o modelo relacional se tornou o mais utilizado.

Mas, mesmo com toda essa supremacia dos bancos de dados relacionais, o mundo mudou e a quantidade de informações e sua importância para a sociedade e as organizações aumentaram. Diante dessa nova perspectiva, o modelo relacional não mais supre as necessidades, e foi preciso investigar novas tecnologias e abordagens para enfrentar esse problema. No entanto, o mais interessante é que não foi preciso propor uma nova teoria para solucionar isso, mas sim encontrar na história algo que resolvesse o problema para implementá-lo computacionalmente. E isso foi o que aconteceu com os bancos de dados orientados a grafos. Eles utilizam os mesmos conceitos descritos há dois séculos, com novas tecnologias embutidas, mas representam fielmente os princípios básicos de um grafo, os nós (dados) e as arestas (relacionamentos).

O banco de dados em grafo Neo4j começou a ser desenvolvido em 2003 e foi distribuído publicamente em 2007. Ele é um banco NoSQL baseado em grafo *open-source*, ou seja, tem seu código aberto, e é implementado nas linguagens de programação Java e Scala (NEO4J, 2015a). Ele é um banco de dados que provê ACID (Atomicidade, Consistência, Isolamento e Durabilidade), suporte a clusterização, recuperação a falhas, cacheamento (*caching*) em memória para grafos, capacidade de armazenar bilhões de nós, e é escrito no topo da JVM (*Java Virtual Machine*) (NEO4J, 2015a).

Além disso, o Neo4j apresenta diferente edições (ou versões):

- *Community:* versão livre e de código-aberto que é de alta *performance* totalmente aderente a transações ACID;
- *Enterprise:* versão que apresenta as características anteriores da *Community* acrescidas de funcionalidades como escalabilidade, tolerância a falhas, alta-disponibilidade, *backups*, e monitoramento compreensivo.

O Neo4j é um banco de dados que implementa o Modelo de Grafo de Propriedades (*Property Graph Model*) (NEO4J, 2015a). Este modelo apresenta entidades conectadas,

chamadas de nós, que podem conter atributos do tipo chave-valor, conhecidos como propriedades (*properties*). Os nós podem ter rótulos (*labels*) que representam diferentes papéis dentro do domínio. Ainda segundo Neo4j (2015a), os relacionamentos desse modelo provêm conexões diretas, nomeadas semanticamente e de relevância entre dois nós. Um relacionamento ainda sempre apresenta uma direção (embora eles podem ser navegados independente de direção), um tipo, um nó de início e um nó de fim, além de poderem ter propriedades quantitativas como pesos, custos, distâncias entre outros (NEO4J, 2015a).

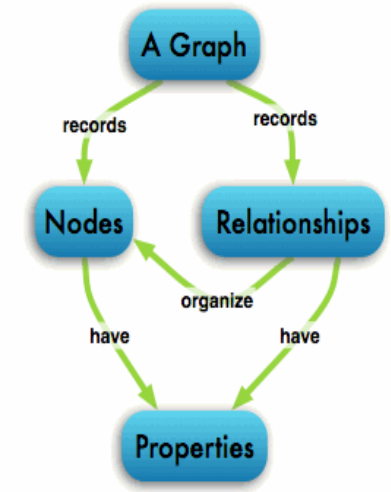

Figura 7 – Modelo de Grafo de Propriedades
(NEO4J, 2015a)

Outro conceito importante é o de que, em um grafo, um relacionamento sempre tem um nó de início e um de fim, não sendo possível deletar um nó sem deletar os relacionamentos ligados a ele, o que é chamado em inglês de *"No broken links*" (Sem ligação quebrada) (NEO4J, 2015a).

Alguns conceitos de relevância acerca do modelo de dados do Neo4j devem ser levados em conta. São eles, de acordo com Neo4j (2015b):

- Domínio: o contexto onde podem ser identificados os nós, *labels*, e relacionamentos do grafo;
- Nós: como apresentado antes, as entidades do grafo. Em um banco de grafos, é uma das unidades fundamentais do modelo. Pode ter propriedades e também *labels*;
- *Label* (rótulo): é um conceito nominal de um grafo, onde nós podem ser agrupados. Um nó não precisa necessariamente ter um *label*, e pode ser rotulado com um ou mais *labels*:

- Relacionamentos: como visto anteriormente, é também uma unidade fundamental do modelo de banco de dados em grafo que representa as relações entre os nós, podendo ter propriedades.

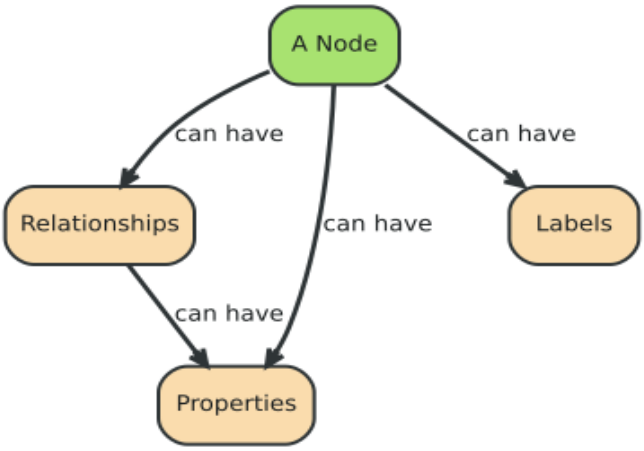

Figura 8 – Visão Geral do Nó
(NEO4J, 2015b)

Como é possível observar na Figura 8, um Nó ("*A Node*") pode ter ("*can have*") Relacionamentos ("*Relationships*"), Propriedades ("*Properties*"), e Rótulos ("*Labels*"). Um Relacionamento pode ter Propriedades ("*Properties*").

Um modelo de dados em grafo para o Neo4j pode ser visto na Figura 9, onde:

- **John, Sally, *Graph Database***: representam os nós do modelo;
- ***FRIEND_OF, HAS_READ***: representam os relacionamentos do modelo;
- ***Name, age, title, authors, since, on, rating***: representam propriedades do modelo;
- ***Person, Book***: representam as *labels* do modelo.

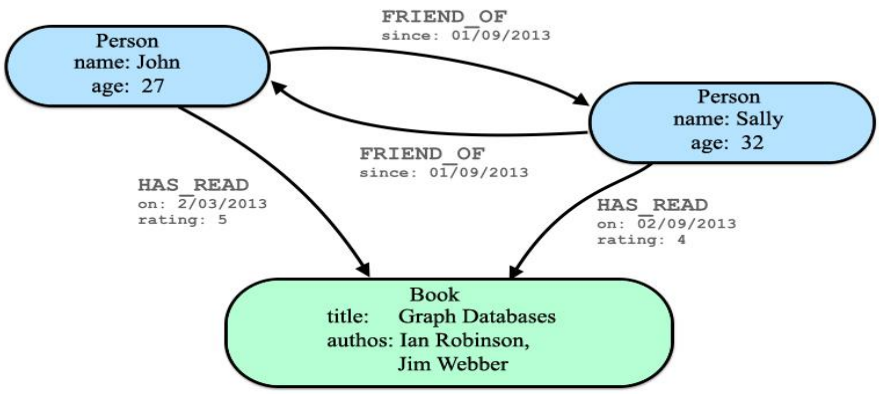

Figura 9 – Modelo de Dados em Grafo
Neo4j, (2015b)

Para efetivamente criar e consultar dados no Neo4j, deve ser usada a linguagem declarativa Cypher, que permite consultas e mudanças expressivas e eficientes em um banco de dados em grafo (NEO4J, 2015c). Segundo Neo4j (2015c), Cypher foi desenvolvida para ser uma linguagem de consulta que foca em simplicidade. É também baseada em termos do idioma Inglês e, por ser uma linguagem declarativa, concentra-se em expressar a clareza de "o quê" recuperar de um grafo e não em "como" recuperar dele. Foi inspirada em um número diferente de abordagens e algumas palavras-chaves e padrões foram inspirados no SQL e no SPARQL *(Protocol and RDF Query Language*).

Com relação à Figura 9, podem ser vistas algumas *queries* (comandos), utilizando-se a linguagem Cypher, segundo Neo4j (2015c):

- Criar um nó

  CREATE (john:Person { name: 'John', age: 27 })

  CREATE (sally:Person { name: 'Sally', age: 32 })

- Criar um relacionamento

  CREATE (sally)-[:FRIEND_OF { since: 1357718400 }]->(john)

- Buscar quando Sally e John se tornaram amigos

  MATCH (sally:Person { name: 'Sally' })

  MATCH (john:Person { name: 'John' })

  MATCH (sally)-[r:FRIEND_OF]-(john)

  RETURN r.since as friends_since

- Buscar a idade de John

  MATCH (john:Person { name: 'John' })

  RETURN john.age as john_age

Como demonstrado, o banco de dados Neo4j provê uma estrutura de grande aderência ao modelo de grafos de propriedades, e, com a utilização da linguagem declarativa Cypher, mudanças e consultas na base de dados podem ser feitas de forma intutiva e simples. Além disso, sua capacidade e estutura confiáveis permitem que diversas aplicações possam confiar seus dados em tal tipo de base.

O uso do Neo4j e do conceito de grafos para representar uma estrutura semântica, no caso uma ontologia, já é utilizado na pesquisa de Schriml et al (2011), intitulada *Disease Ontology: a backbone for disease semantic integration*, que é um grande banco de dados em que se relacionam, via integração semântica, os conceitos representados por termos que

designam doenças. A estrutura consiste em categorias mais gerais de doenças a partir das quais se podem visualizar outras que estão inseridas nelas. Na presente pesquisa, é proposto também representar relacionamentos do tipo, é um, parte de, composto de, é sinônimo de, entre outros. É possível ver que a pesquisa é bastante complexa, devido aos diversos tipos de relacionamentos.

Segundo Schriml et al (2011), o banco de dados em grafo Neo4j provê um mecanismo robusto e rápido de recuperação de nós (dados) e também eficaz para se percorrer um caminho entre nós e relacionamentos, o que seria complexo e necessitaria de vários *joins* (junções) em um banco de dados relacional. Esse tipo de projeto pode ser desenvolvido na plataforma web, utilizando tecnologias como HTML, CSS e *JavaScript* para a construção da interface com o usuário.

Outro trabalho que utilizou o banco de dados Neo4j foi o *Graph Database Application using Neo4j*, de Bungama, Maschietto e Mpinda (2015). O projeto consiste em simular uma rede ferroviária interconectada por várias estações. Cada conexão entre as estações (relacionamentos) possuem uma propriedade quantitativa de distância. O intuito da pesquisa é o de buscar o melhor caminho entre estações e saber quando uma estação é alcançável a partir de outra.

O banco de dados Neo4j também tem sido utilizado em aplicações comercias, como por exemplo pelo Walmart[15]. O grupo de comércio eletrônico da empresa no Brasil utilizou o Neo4j para criar uma ferramenta de recomendação em tempo real (NEO4J, 2015d). Essa ferramenta permite entender as preferências e comportamentos de clientes em uma grande base de dados, fazendo recomendações personalizadas em tempo real, como recomendações do tipo "*you may also like*" (você pode gostar também) (NEO4J, 2015d).

Nesta pesquisa, são utilizados os conceitos de grafo e do banco de dados em grafo Neo4j para implementar o mapa de conhecimento dos assuntos, utilizando-se como base as referências bibliográficas do curso de Tecnologia em Sistemas para Internet, a partir dos sumários das obras, levantado no trabalho de Silva (2013), a fim de atingir o objetivo de conceber a ferramenta de apoio a indexação.

Na Figura 10 é possível visualizar um exemplo do modelo de dados, desta pesquisa em particular, no banco de dados Neo4j. O conceito "Java" tem relacionamentos com outros

---

[15] Walmart é a maior rede de varejo no mundo, com mais de 245 milhões de clientes e mais de 11.500 lojas no mundo (WALMART, 2015). Mais informações em < http://corporate.walmart.com/our-story>.

três conceitos. Pode-se notar que Java então é uma “Linguagem de Programação”, é “Orientado a Objetos” e que contém “Classe”.

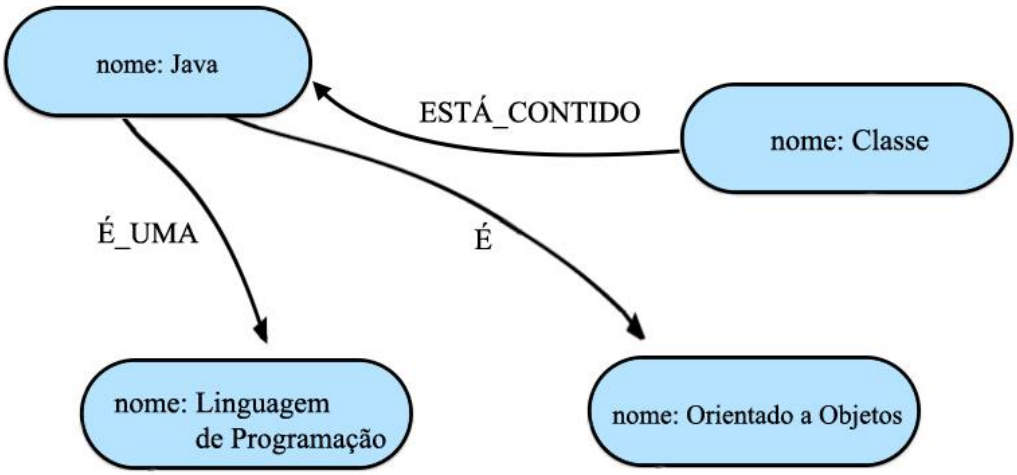

Figura 10 – Exemplo de Modelo de Dados da Ferramenta
Elaboração do autor

O conceito de grafos, bem como sua implementação no banco de dados Neo4j, se assemelha ao conceito dos Mapas do Conhecimento, no que condiz à sua estrutura. Sendo assim, o uso de tais conceitos e tecnologias emergem de forma conveniente na busca por uma solução ao desafio da indexação.

Desta forma, tendo sido estudado os conceitos relevantes à essa pesquisa, parte-se para a sua metodologia e execução, e em fim seus resultados e conclusões, apresentados nos capítulos 3, 4 e 5 respectivamente.

3 Metodologia

Esta pesquisa serve-se de diferentes conceitos e tecnologias a fim de conceber a ferramenta de apoio a indexação de assuntos. Sendo assim, dentro do domínio da Linguística, foi utilizado o conceito de Mapas do Conhecimento. Na Ciência da Informação, a indexação em si, e na Informática o uso do modelo de dados em grafo e um banco de dados em grafo, além da linguagem de programação Java.

A Figura 11 mostra o esquema seguido para a realização da pesquisa.

Figura 11 - Esquema da Metodologia
Elaboração do autor

A **Etapa 1** representa o levantamento do referencial teórico dos assuntos relevantes para esta pesquisa. Tal referencial encontra-se no Capítulo 2.

A **Etapa 2** consiste do levantamento de requisitos da problemática objetivada nessa pesquisa. Tal levantamento se deu em entrevistas com a especialista do domínio, no caso a bibliotecária do IFSP – SBV. Conforme as questões levantadas pela especialista foram definidos os objetivos desta pesquisa bem como sua justificativa, que se encontram no Capítulo 1.

Após a definição dos objetivos chega-se à **Etapa 3**. Esta etapa consiste em elaborar o projeto de desenvolvimento do objetivo proposto. Nela foram definidas as tecnologias a serem usadas e como integrá-las. Além disso foram concebidos diagramas e esquemas que melhor ilustram o sistema e proporcionam maior facilidade de entendimento para a próxima etapa, o desenvolvimento.

As tecnologias usadas no projeto incluem o banco de dados orientado a grafos, Neo4j, a tecnologia Java EE (*Java Enterprise Edition*)[16] e a linguagem de programação Java, como também o uso de um recurso do Java EE, o JSP (*Java Server Pages*). Também foram utilizadas as tecnologias HTML5[17] (*HyperText Markup Language*), CSS[18] (*Cascading Style Sheets*) e *JavaScript*[19], fazendo proveito das bibliotecas *JavaScript JQuery*[20] e vis.js[21] e do framework *Bootstrap*[22] para desenvolvimento da interface *web*. Neste trabalho, está sendo utilizada a versão *Community* 2.3.0 do Neo4j[23].

Na Figura 12 pode-se observar a arquitetura proposta para este trabalho.

---

[16] O *Java Enterprise Edition* é um conjunto de especificações publicadas e regulamentadas pela Oracle e pelo *Java Community Process* (JSRs), que fornecem toda a estrutura para o desenvolvimento de aplicações Web, com tecnologias nativas como o JSP e os Servlets (SAMPAIO, 2011).

[17] HTML5 é a nova geração do HTML, substituindo o HTML 4.01, XHTML 1.0 e XHTML 1.1. HTML5 provê novos recursos que são necessários para as aplicações web (PILGRIM, 2010).

[18] *Cascading Style Sheets* (CSS) é um mecanismo simples para adicionar estilo (por exemplo, fontes, cores, espaçamentos) aos documentos web (W3C, 2013).

[19] Javascript é uma linguagem de programação usada para construir páginas interativas. Ela roda no computador cliente (do visitante) e não requer *downloads* constantes de seu *website* (CHAPMAN, 2013).

[20] Disponível em <http:www.jquery.com>. Acesso em 01/10/2015.

[21] Disponível em <http://visjs.org/>. Acesso em 01/10/2015.

[22] Disponível em <http://getbootstrap.com/>. Acesso em 01/10/2015.

[23] Versão disponibilizada em 21/10/2015. Disponível em <http://neo4j.com/download/>. Acesso em 27/10/2015.

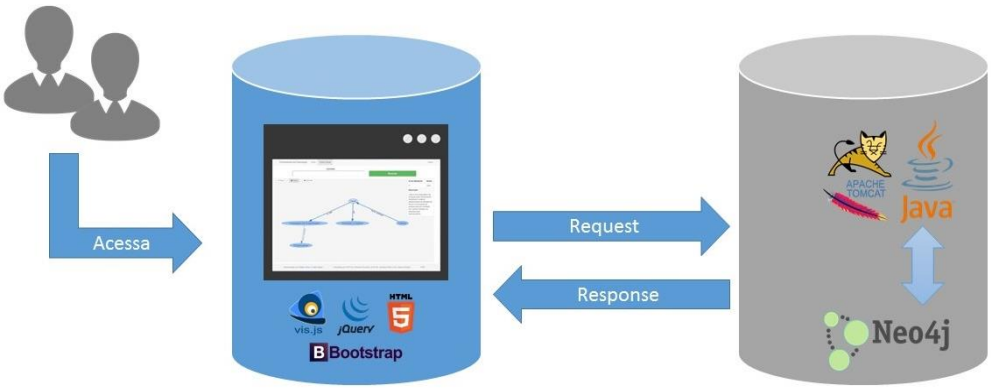

Figura 12 – Arquitetura do Sistema
Elaboração do autor

É possível observar que o usuário acessa o sistema via um navegador, que utiliza tecnologias web do lado cliente (HTML5, *JavaScript* e CSS). Ao fazer interações com o sistema, são enviadas requisições (*request*) para o servidor de aplicação, no caso desta pesquisa está sendo usado o *Apache Tomcat*[24], que se comunica com o banco de dados Neo4j, e então são retornadas respostas (*response*) para o cliente (usuário).

Além disso, foi proposto inicialmente o modelo de dados a ser usado no banco de dados em grafo, o Neo4j. Como a proposta desta pesquisa é auxiliar o bibliotecário na indexação, ou seja, na busca de assuntos para indexar um livro no catálogo, apenas os assuntos fazem parte do modelo de dados (e ficam de fora os livros, por exemplo). Além disso, é preciso definir as propriedades e os relacionamentos entre os dados, ou nós.

Na Figura 13, é possível observar o modelo de dados para essa pesquisa. “Assunto” refere-se ao *label*, que é um identificador que agrupa nós semelhantes. No caso desta pesquisa, têm-se apenas *labels* do tipo Assunto. “Nome” e “descricao” são as propriedades dos nós do modelo. Na implementação, omite-se os caracteres especiais e acentos. “Nome” é para o nome do assunto, como por exemplo Java ou Linguagem de Programação. “Descricao” é para uma breve descrição do assunto, que visa a facilitar a identificação dos assuntos pelo usuário.

Os relacionamentos estão expressos pelas linhas com setas nas pontas, que representam suas direções. Nesta pesquisa, foi delimitado o uso de três relacionamentos: EH_UMA (é uma), EH (é), e ESTA_CONTIDO (está contido). Tais relacionamentos têm como base os conceitos de definição (é um(a)), hiponímia (é um tipo de) e hiperonímia (está contido), que fazem parte da semântica lexical (PIETROFORTE & LOPES, 2007).

[24] Disponível em <http://tomcat.apache.org/>. Acesso em 01/10/2015.

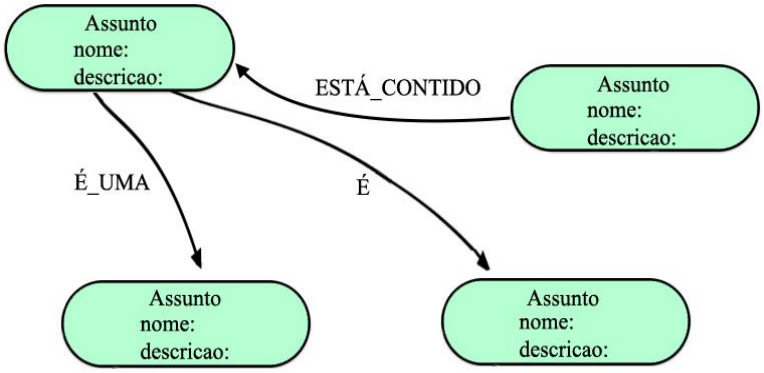

Figura 13 - Modelo de dados
Elaboração do autor

Na Figura 14, é possível ver o diagrama de classes da ferramenta. Foi utilizada a tecnologia de *Servlets* do Java para tratar as requisições HTTP (*Hypertext Transfer Protocol*) do navegador. Como o Neo4j é um banco de dados orientado a grafos, e não a objetos, foi proposto criar as classes “No” e “PercorreNo”, além da classe “Relacionamento”, que apenas define os tipos dos relacionamentos. Essas classes servem como uma abstração dos dados e permitem o trabalho entre o *Servlet* e a camada de conexão com o banco de dados, representada pelas classes “PercorreNoDao”, “NoDao” e “ConexaoBanco”.

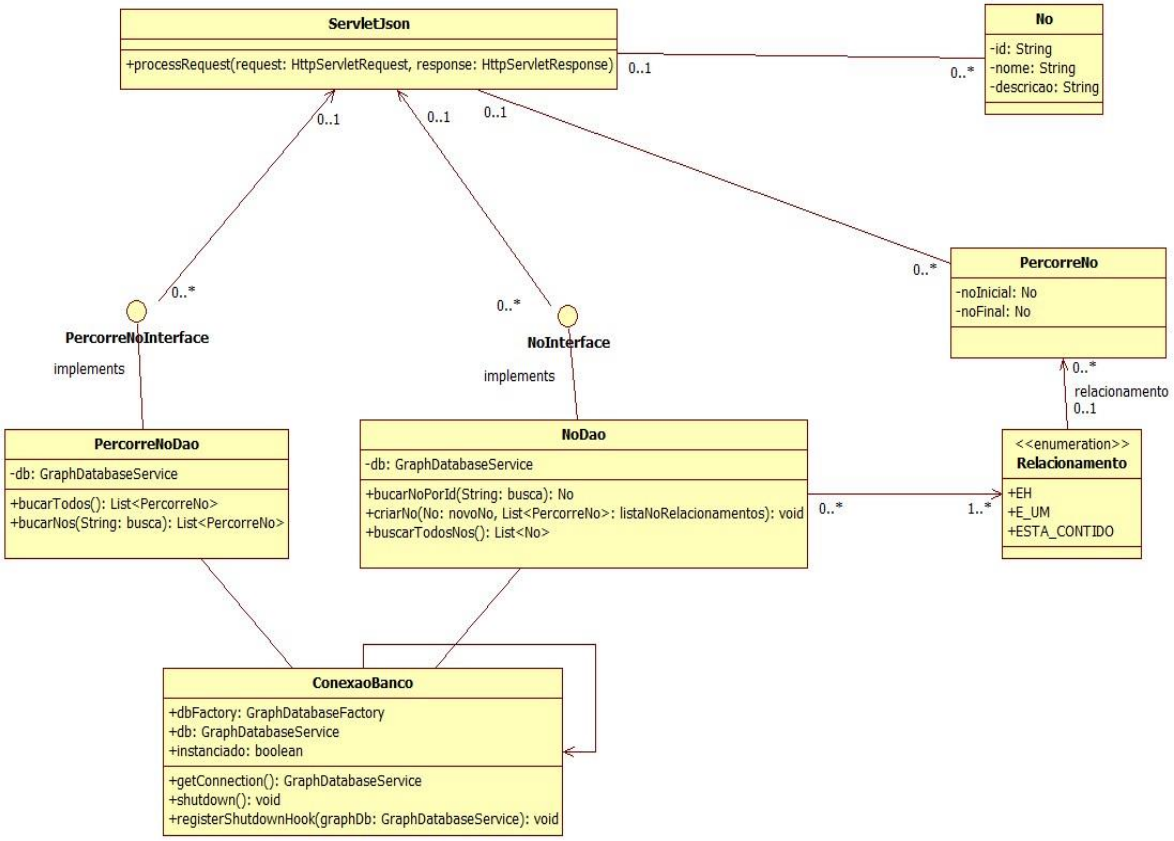

Figura 14 – Diagrama de Classes

Partindo desses princípios, foi desenvolvido o software, que consiste na **Etapa 4**.

Para o desenvolvimento foi utilizada a ferramenta NetBeans IDE[25]. A Figura 15 mostra a estrutura do projeto na ferramenta.

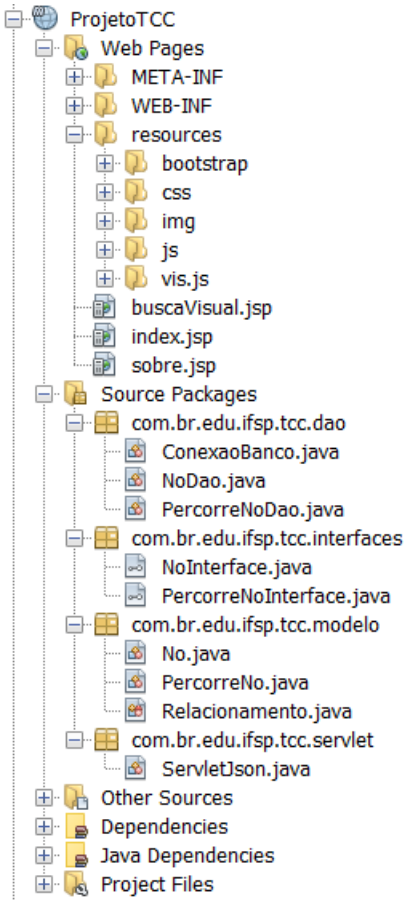

Figura 15 - Estrutura do projeto

As páginas JSP são interpretadas pelo servidor de aplicação e geram o resultado a ser exibido no navegador, onde ocorrem as interações. O usuário quando interage com o sistema, por exemplo, quando clica para exibir todos os nós, envia uma requisição na qual é processada pelo "ServletJson" que retorna um ou uma lista de objetos no formato JSON[26]. Este projeto utiliza a técnica AJAX (*Asynchronous JavaScript and XML*) que propicia requisições assíncronas, ou seja, apenas partes da página são atualizadas (RIORDAN, 2008). Quando os dados são recebidos pela requisição AJAX eles são montados utilizando-se da biblioteca vis.js para criar a representação gráfica do grafo.

---

[25] Disponível em <https://netbeans.org/>. Acesso em 01/11/2015.

[26] JSON (JavaScript Object Notation) é um formato de dados leve baseado em texto que é usado para troca de dados entre clientes e servidores (SRIPARASA, 2013).

Nesta pesquisa está sendo usada a conexão embarcada do Neo4j, Neo4j *Embedded*. A classe "ConexaoBanco" é responsável por prover a conexão com o banco de dados para uso na aplicação. As classes "NoDao" e "PercorreNoDao" são responsáveis pela execução de consultas e alterações no banco de dados. A Figura 16 mostra um fragmento da classe que recupera a conexão com o banco.

```
public final class ConexaoBanco {

    public static GraphDatabaseFactory dbFactory;
    public static GraphDatabaseService db;
    private static boolean instanciado = false;

    public static GraphDatabaseService getConnection(){

        try {

            if(!instanciado){

                dbFactory = new GraphDatabaseFactory();
                File file = new File("C:/Neo4jDB");
                db = dbFactory.newEmbeddedDatabase(file);
                registerShutdownHook( db );
                instanciado = true;

            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return(db);

    }
```

Figura 16 – Exemplo de conexão com o banco

Como pode ser visto, a conexão do modo embarcado é feita diretamente com o diretório onde ficam os arquivos de armazenamento dos dados, representado no objeto "*file*".

A Figura 17 mostra o método responsável por criar um novo nó no banco de dados. Neste exemplo é utilizada no projeto a biblioteca do Neo4j disponibilizada para desenvolvimento em Java, onde um nó é um objeto do tipo "*Node*". Tal objeto é criado a partir da conexão com o banco e então é atribuído a ele seu *label*, propriedades e relacionamentos.

```java
@Override
public void criarNo(No novoNo, List<PercorreNo> listaNoRelacionamentos){

    //Pega o Banco
    db = ConexaoBanco.getConnection();

    //Pega o ultimo no para incrementar o id
    No ultimoNo = buscarUltimoNo();
    String id;
    Node novoNoNeo4j;
    Node noFinalNeo4j;

    //Se o ultimoNo for vazio quer dizer q nao tem nada no BD.
    if(ultimoNo.getId() == null){
        id = "0";
    }else{
        //Cria o id a ser passado (adiciona 1 ao ultimo valor)
        //Retorna a conta como uma String
        id = String.valueOf(Integer.parseInt(ultimoNo.getId()) + 1);
    }

    try ( Transaction tx = db.beginTx() ){

        //Cria no no
        novoNoNeo4j = db.createNode();

        //Atribui as propriedades
        novoNoNeo4j.addLabel( DynamicLabel.label( "Assunto" ));
        novoNoNeo4j.setProperty( "id", id);
        novoNoNeo4j.setProperty( "nome", novoNo.getNome() );
        novoNoNeo4j.setProperty( "descricao", novoNo.getDescricao());

        //Se a lista nao estiver vazia vamos criar os relacionamentos
        if(listaNoRelacionamentos != null){

            //Cria os relacionamentos com os outros nos
            for (PercorreNo percorreNo : listaNoRelacionamentos) {

                //Busca o No (do Neo4j) pelo id
                noFinalNeo4j = buscarNoNeo4jPorId(percorreNo.getNoFinal().getId());

                //Cria os relacionamentos
                novoNoNeo4j.createRelationshipTo( noFinalNeo4j, percorreNo.getRelacionamento() );
            }
        }
        //Finaliza a transacao
        tx.success();
    }//fim do try
}
```

Figura 17 – Exemplo de código para criar um nó

A Figura 18 mostra o método que busca nós com um parâmetro. É possível observar a linguagem *Cypher* sendo usada na busca. Tal cláusula busca os relacionamentos direcionados entre dois nós na qual o nome de algum dos nós seja igual aquele passado no parâmetro, e retorna os nós e os relacionamentos. Como não é objetivo deste trabalho explicitar toda a implementação, apenas alguns pontos relevantes de código são apresentados[27].

[27] O código fonte dessa pesquisa encontra-se disponível em <https://github.com/FilipeNavas/ProjetoTCC>. Acesso em 06/11/2015.

```
@Override
public List<PercorreNo> buscarNos(String busca){


    //Pega o Banco
    db = ConexaoBanco.getConnection();

    List lista = new ArrayList();

    //cria um map chamado params para colocar chave/valor
   Map<String, Object> params = new HashMap<>();

   //coloca o valor busca na chave busca
   params.put( "busca", "(?i)" + busca);

    try ( Transaction transaction = db.beginTx();

    Result result = db.execute( "START n = node(*)"
                                + "MATCH (n)-[r]->(x)"
                                + "WHERE"
                                + " has(n.nome) and n.nome=~{busca} or"
                                + " has(x.nome) and x.nome=~{busca}"
                                + "return n, type(r), x", params) ){

        while ( result.hasNext() ){

            Map<String,Object> row = result.next();

            Node x = (Node) row.get("x");
            Node n = (Node) row.get("n");

            //Cria um objeto PercorreNo
            PercorreNo percorreNo = new PercorreNo();

            //Cria um objeto No para NoInicial
            No noInicial = new No();

            //Cria um objeto No para NoFinal
            No noFinal = new No();
```

Figura 18– Exemplo de código para busca

O desenvolvimento da interface *web* fez proveito de algumas bibliotecas *JavaScript* e CSS já mencionadas. Não foi levado em conta neste trabalho o uso de *layout* responsivo[28], apesar das tecnologias usadas e dos navegadores modernos já propiciarem algum tipo dessa funcionalidade nativamente.

[28] *Layout* responsivo (ou *design* responsivo) é uma forma de fazer com que websites sejam vistos e usados facilmente independentemente do tamanho da tela, desde celulares até computadores *desktop* (PETERSON, 2014).

Foi utilizada a biblioteca *JavaScript* vis.js para a criar a visualização dos conceitos e seus relacionamentos em forma de um grafo. Foi utilizada o componente *Networks* da biblioteca, que cria visualizações em forma de redes.

A Figura 19, mostra um fragmento da função que cria o grafo na tela. É possível ver que a função recebe como parâmetros os nós e relacionamentos, que então são atribuídos em *dataGraph*. É possível também configurar o *layout* do grafo e outras opções de exibição, representados na variável *options*.

```
//Função que cria o Grafo na tela. Parâmetros: nodes e edges.
function createGraph(nodes, edges){
    // cria uma network (rede)
    var container = document.getElementById('mynetwork');
    var dataGraph = {
        nodes: nodes,
        edges: edges
    };
    var options = {
        autoResize: true,
        height: '100%',
        width: '100%',
        locale: 'pt',
        interaction:{
            hover: true
        },
        //define o tipo de layout do grafo
        layout:{
            hierarchical: {
                sortMethod: 'directed',
                direction: direction,
                levelSeparation: 150
            }
        }
    };

    //Cria a network
    network = new vis.Network(container, dataGraph, options);
```

Figura 19 – Função JavaScript que cria o grafo

Já na Figura 20 é possível observar, de forma simplificada, como os dados são atribuídos para chamar a função que cria o grafo. São criadas variáveis do tipo *DataSet* da biblioteca vis.js, onde são atribuídos os nós (variável *nodes*) e os relacionamentos (variável *edges*). Então é chamada a função *createGraph* passando os nós e relacionamentos como parâmetros.

```
//NODES - CONCEITOS
var options = {};
var nodes = new vis.DataSet(options);
nodes.add([
    {id: val.noInicial.id, label: "'" + val.noInicial.nome + "'"}
]);

//EDGES - RELACIONAMENTOS
var edges = new vis.DataSet(options);
edges.add([
    {id: key , from: val.noInicial.id, to: val.noFinal.id,
     arrows: 'to', label: "'" + val.relacionamento + "'",
     font: {align: 'bottom', size: '10'}}
]);

//Chama a função que cria o grafo
createGraph(nodes, edges);
```

Figura 20 – Exemplo da atribuição de dados e chamada para criar o grafo

Finalizado o desenvolvimento partiu-se então para a **Etapa 5**, Entrega e Finalização. Esta é a etapa final da pesquisa e consiste em apresentar a ferramenta desenvolvida ao autor da problemática, no caso a bibliotecária do IFSP – SBV. Não é do escopo desta pesquisa analisar as reações do usuário, bem como conceitos como usabilidade e a praticidade da ferramenta.

A finalização consiste em apresentar os resultados e conclusões desta pesquisa, encontrados nos capítulos 4 e 5 respectivamente.

4 Resultados

Esta pesquisa, que se fundamentou na Linguística, na Ciência da Informação e na Informática, atingiu seu objetivo geral, porque concebeu-se a ferramenta de apoio à indexação proposta para auxílio do bibliotecário na organização do acervo.

Além disso, a ferramenta proposta na pesquisa fica à disposição do usuário para que ele próprio crie os conceitos e seus relacionamentos de forma dinâmica, através da própria ferramenta sem necessidade de apoio técnico constante.

A Figura 21 mostra a tela de inserção de dados. Nela, o usuário pode criar novos assuntos (ou nós) e relacioná-los com outros. Ainda é possível inserir a descrição do assunto e também relacioná-lo com os outros nós que já existem no sistema utilizando os relacionamentos possíveis, que são É, É UMA, e ESTÁ CONTIDO.

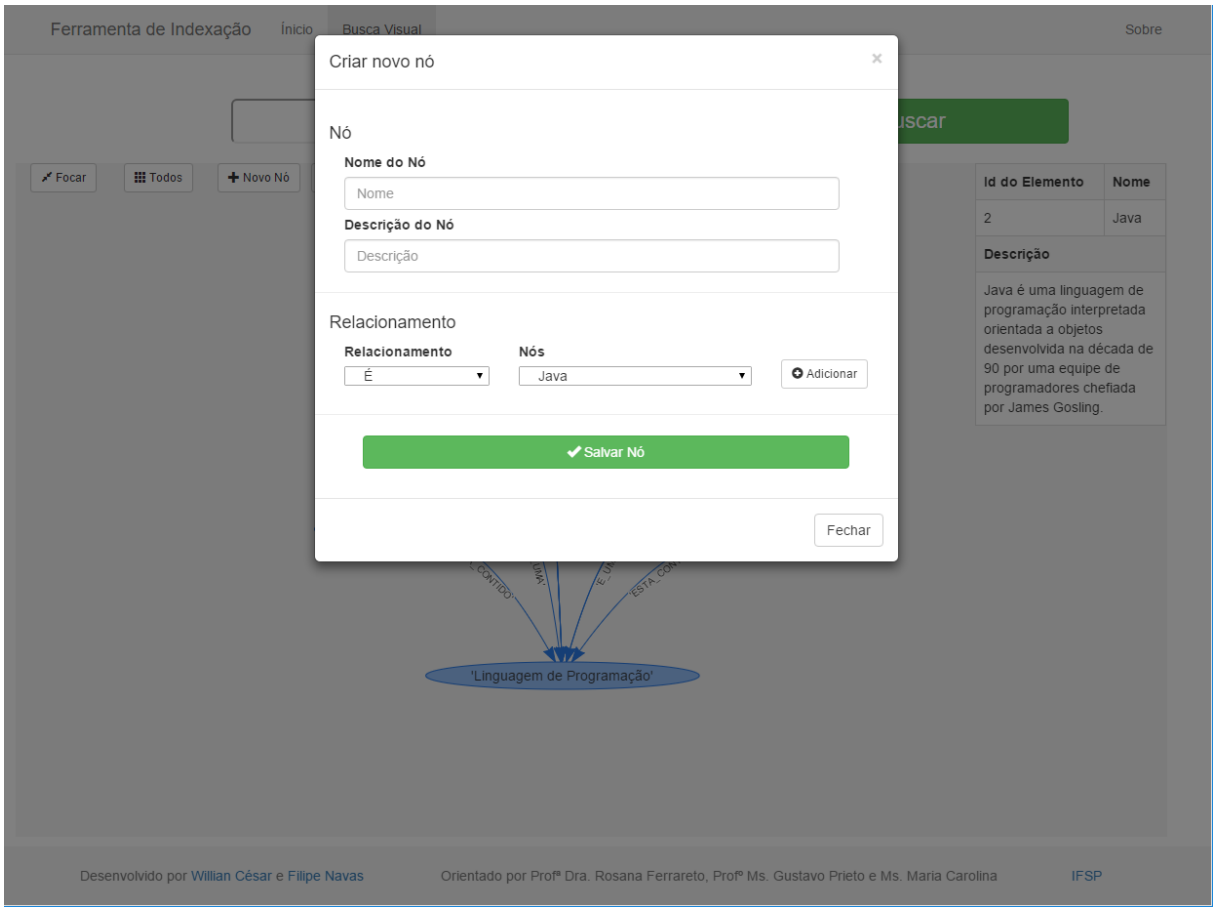

Figura 21 – Inserir nó

Além disso, o usuário pode editar nós já existentes, como pode ser visto na Figura 22. O usuário tem a opção de editar seu nome, descrição ou até mesmo adicionar novos relacionamentos. Ainda na mesma tela, o usuário tem a opção de deletar aquele nó, na qual irá deletar também seus relacionamentos.

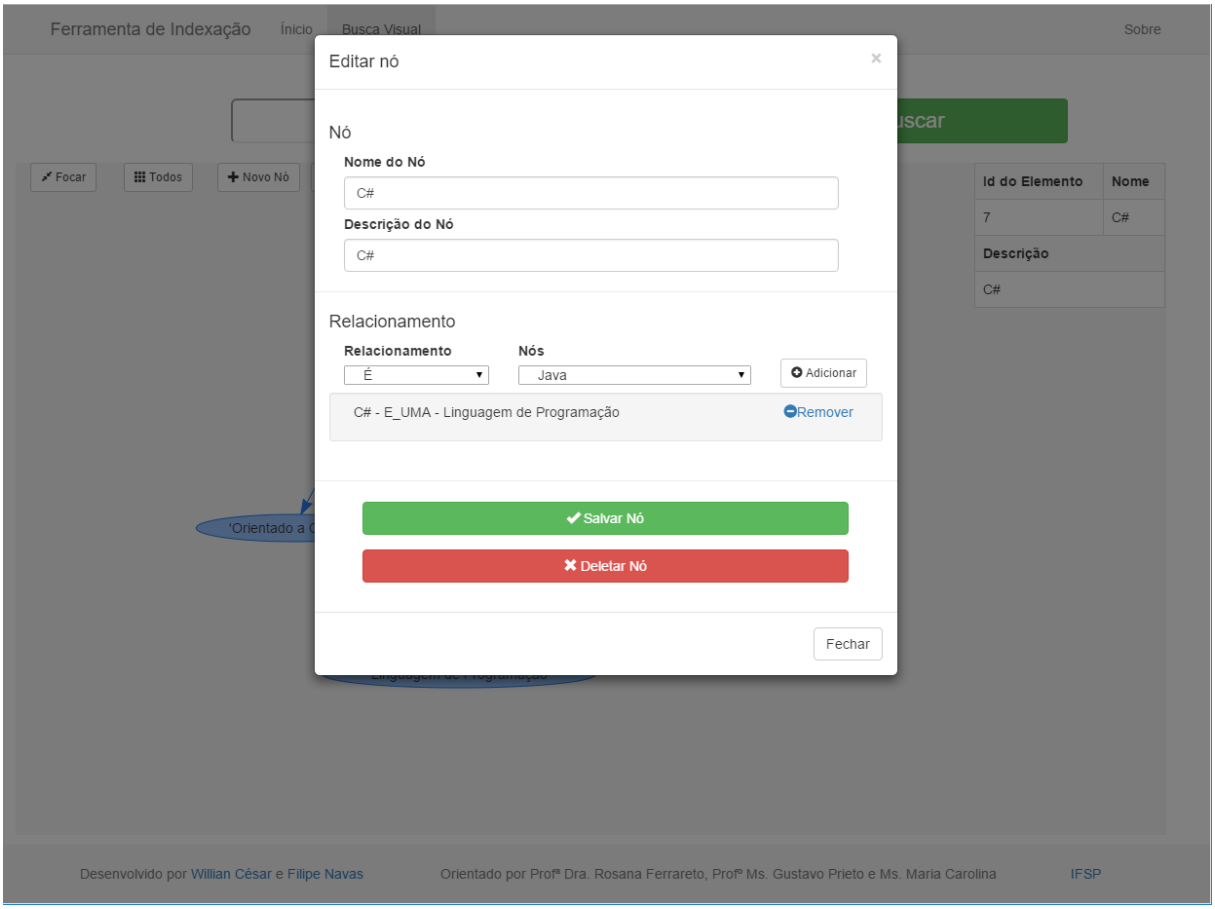

Figura 22 – Editar e deletar nó

Também foi atingido o objetivo de permitir ao usuário visualizar os conceitos e relacionamentos em uma interface na forma de grafo.

Na Figura 23, há um exemplo da funcionalidade de buscar todos os assuntos (nós) e relacionamentos. É possível observar também um quadro no canto superior direito em relação ao grafo que mostra detalhes do nó selecionado pelo usuário.

Na Figura 24, é possível ver um exemplo de uma busca. O usuário pode inserir o termo que deseja buscar no campo de texto e então clicar no botão “Buscar”. Dessa forma, o sistema irá buscar no banco de dados pelo nome dos elementos que coincidem com aquele termo informado pelo usuário e retornará na tela os resultados.

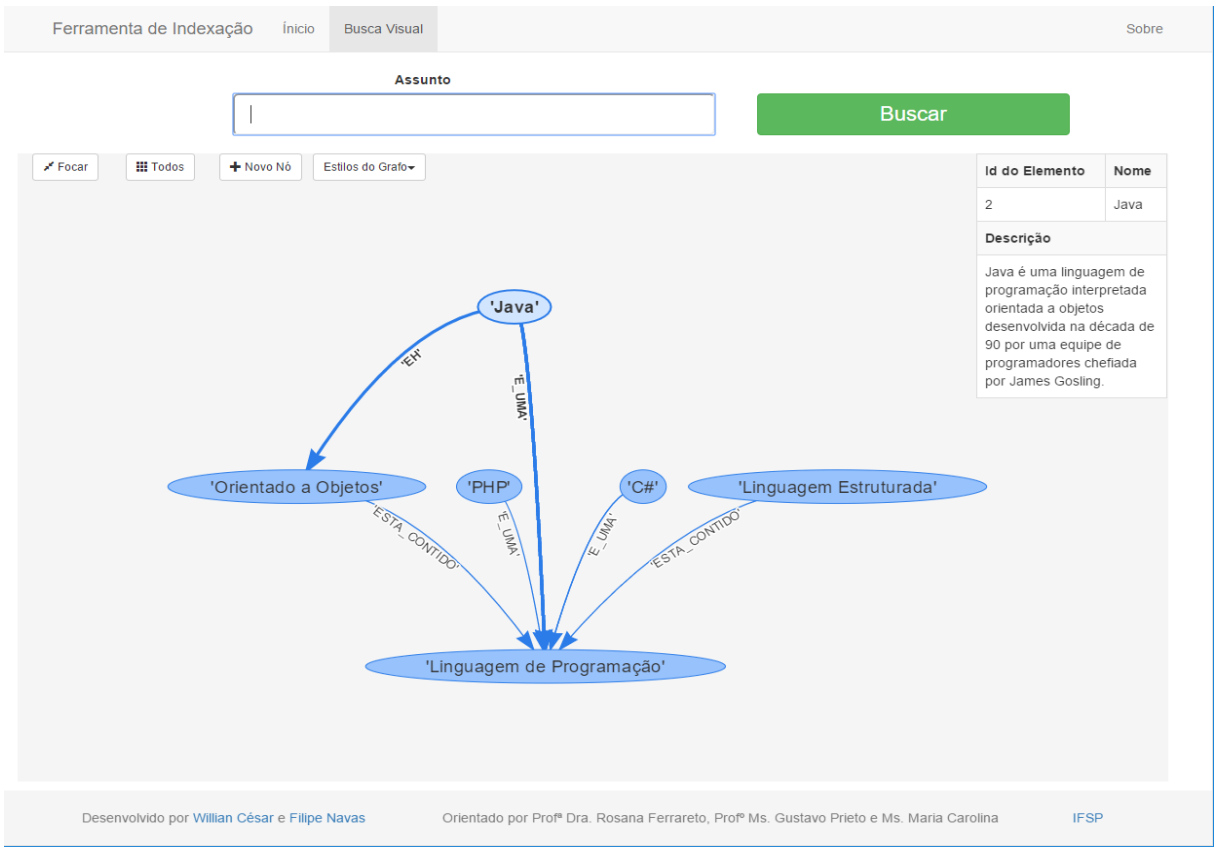

Figura 23 – Buscar todos os nós

Figura 24 – Buscar nó por nome

Os dados do sistema serão inseridos pelo usuário, que é responsável pela validade entre os relacionamentos, por exemplo se tal assunto pertence a tal área. Além disso, é de responsabilidade do usuário, quando preciso, requisitar auxílio de especialistas das áreas de obras a serem inseridas, para que se criem relacionamentos de acordo com o assunto.

A ferramenta pode ser acessada via um navegador de Internet (*web browser*), tendo que ser implantada em um servidor *web* com suporte à tecnologia Java, como por exemplo o *Apache Tomcat* utilizado nesta pesquisa. Quanto ao banco de dados Neo4j, é preciso apenas configurar na classe de conexão com o banco a pasta onde ficará os arquivos de armazenamento do banco, pois a ferramenta já inclui as bibliotecas necessárias para seu funcionamento.

Diante do exposto, está claro que a ferramenta pode ser de grande valia para os bibliotecários na organização e recuperação de assuntos para indexação.

5 Conclusões

Este trabalho atingiu seu objetivo ao conceber a ferramenta de apoio à indexação para bibliotecários, que visa facilitar o processo de identificação dos assuntos no momento da catalogação de alguma obra no acervo.

A fim de atingir tal objetivo, essa pesquisa valeu-se dos estudos da Linguística, identificando estruturas de organização semântica do conhecimento. Foi estudado também conceitos da Ciência da Informação, área de estudo geral onde está inserida a indexação. E, finalmente, foi estudado na Informática temas como os bancos de dados, a estrutura de dados em grafo e o banco de dados em grafo Neo4j.

A integração entre tais disciplinas levou à solução proposta. Da Linguística, tem-se o mapa do conhecimento, que é, nesta pesquisa, representado pelo grafo na Informática. A indexação é, em si, a fonte da problemática apresentada.

Sendo assim, é interessante notar a importância da multidisciplinaridade nas pesquisas aplicadas, algo que promove e gera inovações quando há uma abordagem diferenciada entre problemas de áreas distintas. Esta pesquisa, estando inserida em diferentes áreas, é em si um produto da inovação da multidisciplinaridade. De um problema da ciência da informação, foi levantada uma abordagem na linguística que foi então implementada na informática.

A ferramenta implementada nesta pesquisa pode ser agora utilizada por bibliotecários a fim de facilitar o trabalho de indexação. Além disso, por ser uma ferramenta dinâmica, em que o próprio usuário insere seus dados, é aberta a qualquer área ou sujeito que possa fazer proveito dos relacionamentos.

O uso de banco de dados em grafo, dentro das tecnologias NoSQL, é também notável para o fomento dessas novas tecnologias de armazenamento de dados.

Em um contexto geral, esta pesquisa se mostra de grande valia para profissionais das três áreas envolvidas, com discussões coerentes com temas atuais em relação à tecnologia e a representação da informação.

O autor desta pesquisa deixa como sugestões para trabalhos futuros a inclusão de diferentes relacionamentos, ou até mesmo a possibilidade de criar relacionamentos

dinamicamente. Além disso, podem ser incluídas melhorias nas formas de representação do grafo na tela, e também a visualização da listagem geral dos vocabulários controlados.

Também, pode ser proposto junto com bibliotecários especialistas na indexação, uma ferramenta que se adere a normas internacionais ou nacionais de indexação e que se torne uma fonte padrão de consultas para os profissionais do meio.

Por fim, pode ser disponibilizado as consultas dos assuntos por meio de serviços na Internet que possam ser utilizados por outras aplicações.

Referências

ABNT (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS). **NBR 12676:** Métodos para análise de documentos - determinação de seus assuntos e seleção de termos de indexação. Rio de Janeiro, 1992. 4 p.

AHMED, K.; MOORE, G. An Introduction to Topic Maps. 2005. Disponível em <https://msdn.microsoft.com/en-us/library/aa480048.aspx>. Acesso em: 08 out. 2015.

BREITMAN, K. K. **Web Semântica:** a Internet do Futuro. Rio de Janeiro. GEN, 2010.

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

BUNGAMA, P. A.; MASCHIETTO, L. G.; MPINDA, S. A. T. Graph Database application using Neo4j (Railroad Planner Simulation). In: International Journal of Engineering Research & Technology, Vol. 4 Issue 04. IJERT, 2015.

CARVALHO, P. S. de. Interação entre humanos e computadores – uma introdução. São Paulo: EDUC, 2000.

CHANDRASEKARAN, B.; JOSEPHSON, R.; BENJANMINS, V. What are ontologies, and why do we need them? **IEEE Intelligent Systems**. V. 14, n. 1, p. 20-25, Jan. 1999.

CHAPMAN, S. What is Javascript. Disponível em: <http://javascript.about.com/od/reference /p/javascript.htm>. Acesso em: 19 mai. 2010.

DI FELIPPO, A. **Ontologias linguísticas aplicadas ao processamento automático das línguas naturais:** o caso das redes wordnets. In: Magalhães, J. S.; Travaglia, L. C. (Orgs). Múltiplas perspectivas em Linguística. Uberlândia: Edufu, 2008. ISBN 978-85-7078-200-7.

GARSHOL, M. L. A Citizen's Portal for the city of Bergen. In: Scaling Topic Maps, Third International Conference on Topic Maps Research and Applications. Leipzig, Alemanha. TMRA, 2007.

GOODRICH, M.T.; TAMASSIA, R. **Estruturas de Dados & Algoritmos em Java.** 5. ed. Porto Alegre: Bookman, 2013.

GRISHAM, R. **Computational Linguistics:** an Introduction. Cambridge: Cambridge University Press, 1992.

GUEDES, R. de M. DIAS; E. J. W. **Indexação Social:** Abordagem Conceitual. In: **Revista ACB:** Biblioteconomia em Santa Catarina, Florianópolis, v.15, n.1, p. 39-53, jan./jun. 2010.

HAWKING, S. **Nosso futuro:** Jornada nas Estrelas? – Como a vida biológica e eletrônica continuarão evoluindo em complexidade a um ritmo sempre crescente, in HAWKING, S., O universo numa casca de noz. São Paulo: Mandarim, 2001.

HODGES, A. Turing: um filósofo da natureza. São Paulo: UNESP, 2001.

ZIKOPOULOS, P.; EATON, C. **Understanding Big Data:** Analytics for Enterprise Class Hadoop and Streaming Data. New York: McGraw Hill Professional, 2011.

LAVIK, S.; NORDENG, T. W. **Concept Maps:** Theory, Methodology, Technology. In: CMC: International Conference on Concept Mapping. Pamplona, Espanha. CMC, 2004.

LEÃO, F. REVOREDO, K. BAIÃO. F. Um framework para refinamento de ontologias através de técnicas de revisão de teorias. In: Congresso da Sociedade Brasileira de Computação, 31., 2011, Natal.

LUCCA, L. C. **GGraph:** Uma ferramenta para aplicações que envolvem grafos. São Carlos: USP, 2012.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **An Introduction to Information Retrieval**. Cambridge: Cambridge University Press, 2008.

MCDONALD, C.; YAZDANI, M. **Prolog Programming:** a Tutorial Introduction. Oxford: Blackwell Scientific Publications, 1990.

MCGUINNESS. D. L; Ontologies come of age. Universidade de Stanford. 2003. Disponível em: <http://www-ksl.stanford.edu/people/dlm/papers/ontologies-come-of-age-mit-press-%28withcitation%29.htm>. Acesso em: 28 Out. 2015.

MIGUEL, S. B.; CARNEIRO, F. C. F. Repositório de Dados Relacional ou NoSQL? **JAVA MAGAZINE**, n. 114., abr. 2013.

MOREIRA, A. **Tesauros e Ontologias:** estudo de definições presentes na literatura das áreas das Ciências da Computação e da Informação, utilizando-se o método analítico-sintético. Belo Horizonte. UFMG, 2003.

MOREIRO, J. A.; CUADRADO, S. S.; MORATO, J. Panorámica y tendencias en topic maps. Hipertext.net, núm. 1, 2003. Disponível em <http://ddd.uab.cat/pub/artpub/2011/88754/hipertext_a2011n9a2/topic_maps.html>. Acesso em: 07 out. 2015.

NEO4J. **What is a Graph Database?**. Neo Technology. 2015a. Disponível em: <http://neo4j.com/developer/graph-database/#property-graph>. Acesso em: 08 out. 2015.

NEO4J. **Graph Data Modeling Guidelines.** Neo Technology. 2015b. Disponível em: <http://neo4j.com/developer/guide-data-modeling/>. >. Acesso em: 08 out. 2015.

NEO4J. **What is Cypher?.** Neo Technology. 2015c. Disponível em: <http://neo4j.com/docs/stable/cypher-introduction.html/>. Acesso em: 08 out. 2015.

NEO4J. **Walmart Optimizes Customer Experience with Real-time Recommendations.** Neo Technology. 2015d. Disponível em: <http://info.neo4j.com/rs/neotechnology/images/neo4j-casestudy-walmart.pdf/>. Acesso em: 08 out. 2015.

OTHERO, G. A.; MENUZZI S. M. **Linguística Computacional:** teoria e prática. São Paulo. Parábola, 2005.

PEPPER, S. **The TAO of Topic Maps**. Ontopia. 2007. Disponível em <http://lingo.uib.no/trond/TopicMaps/Ontopia/Garshol%20-%20The%20TAO%20of%20Topic%20Maps.pdf>. Acesso em: 07 out. 2015.

PEPPER, S.; VITALI, F.; GARSHOL, L. M.; GESSA, N.; PRESUTTI. A Survey *of* RDF/Topic Maps *Interoperability* Proposals. W3C Working Group Note. Disponível em <http://www.w3.org/TR/rdftm-survey/>. Acesso em: 07 out. 2015.

PETERSON, C. **Learning Responsive Web Design:** A Beginner's Guide. Sebastopol: O'Reilly Media, 2014

PIETROFORTE, A. V. S.; LOPES, I. C. Semântica lexical. Em: FIORIN, J. L. (org.) **Introdução à linguística.** II. Princípios de análise. São Paulo: Contexto, 2007. p. 111-136.

PILGRIM, M. HTML5: Up and Running. Sebastopol, CA: O'Reilly Media, 2010.

POLLOCK, J. T. Web Semântica para Leigos. Rio de Janeiro: Alta Books, 2010.

RAFFERTY, P; HIDDERLEY, R. **Flickr and democratic indexing:** dialogic approaches to indexing. Aslib Proceedings, v. 59, Issue 4/5, 2007. p. 397-410.

RAMALHO, R. A. S. **Desenvolvimento e utilização de ontologias em Bibliotecas Digitais:** uma proposta de aplicação. Marília. UNESP, 2010.

RIORDAN, R. M. Head First Ajax. Sebastopol: O'Reilly Media, 2008.

ROWLEY, J. A biblioteca eletrônica. Brasília: Briquet de Lemos, 2002.

SALES, R. de; CAFE, L. Diferenças entre tesauros e ontologias. **Perspectivas em Ciência da Informação.** Belo Horizonte , v. 14, n. 1, p. 99-116, Apr. 2009 . Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-99362009000100008&lng=en&nrm=iso>. Acesso em: 06 out. 2015.

SAMPAIO, C. **Java Enterprise Edition 6**: Desenvolvendo aplicações coorporativas. Rio de Janeiro: Brasport, 2011.

SARDINHA, T. B. **Linguística de Corpus:** histórico e problemática. **DELTA**. São Paulo, v. 16, n. 2, p. 323-367, 2000.

SCHRIML, L. M.; ARZE, C.; NADENDLA, S.; CHANG, Y. W.; MAZAITIS, M.; FELIX, V.; FENG, G.; KIBBE, A. W. Disease Ontology: a backbone for disease semantic integration.Oxford: Oxford University Press, 2011.

SILVA, W. C. **Ontologia de Sistemas Para Internet:** proposta de um catálogo bibliotecário online baseado em banco de dados orientado a grafos. Trabalho de Conclusão de Curso - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. São João da Boa Vista, 2013.

SOUSA, P. B. de. FUJITA; M. S. L. **Do catálogo impresso ao on-line:** algumas considerações e desafios para o bibliotecário. In: **Revista ACB:** Biblioteconomia em Santa Catarina, Florianópolis, v.17, n.1, p. 59-75, jan./jun. 2012.

SOUSA, P. B. de. FUJITA; M. S. L. **ANÁLISE DE ASSUNTO NO PROCESSO DE INDEXAÇÃO:** um percurso entre teoria e norma. In: **Informação & Sociedade**: Estudos, João Pessoa, v.24, n.1, p. 19-34, jan./abr. 2014.

SRIPARASA, S. S. JavaScript and JSON Essentials. Birmingham: Packt Publishing, 2013.

STAAB, S. Ontologies and the Semantic Web. In: SMBM (Second International Symposium on Semantic Mining in Biomedicine). Jena, Alemanha. SMBM, 2006.

UNISIST. Princípios de indexação. Revista da Escola de Biblioteconomia da UFMG, Belo Horizonte, v.10, n.1, p.83-94, mar. 1981.

VIEIRA, R.; LIMA, V. L. S. **Linguística computacional:** princípios e aplicações. In: IX Escola de Informática da SBC-Sul. Passo Fundo, Maringá, São José. SBC-Sul, 2001.

VIEIRA, R. **Linguística computacional:** fazendo uso do conhecimento da língua. Entrelinhas, ano 2, n. 4, São Leopoldo: Unisinos, 2002.

WALMART. **Our Story.** Wal-Mart Stores, Inc. 2015. Disponível em: <http://corporate.walmart.com/our-story/>. Acesso em: 08 out. 2015.

W3C. **Cascading Style Sheets**. Disponível em <http://www.w3.org/Style/CSS/>. Acesso em: 09 jun. 2013.