



**ULBRA** **CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"  
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

*Diego Oliveira Rodrigues*

**Desenvolvimento de um modelo para recomendação de especialistas em  
ambientes sociais virtuais**

Palmas - TO

2013



# **CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"  
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Diego Oliveira Rodrigues

## **Desenvolvimento de um modelo para recomendação de especialistas em ambientes sociais virtuais**

Projeto de pesquisa apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação do Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Msc. Edeilson Milhomem Silva

Palmas - TO

2013



# **CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"  
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Diego Oliveira Rodrigues

## **Desenvolvimento de um modelo para recomendação de especialistas em ambientes sociais virtuais**

Projeto de pesquisa apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação do Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Msc. Edeilson Milhomem Silva.

Aprovada em \_\_\_\_\_ de 2013.

### **BANCA EXAMINADORA**

---

Prof. Msc. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas

---

Prof. Msc. Jackson Gomes de Souza

Centro Universitário Luterano de Palmas

---

Prof. Msc. Edeilson Milhomem da Silva

Centro Universitário Luterano de Palmas

Palmas - TO

2013

## **AGRADECIMENTOS**

A Deus, por ter me dado a graça de alcançar esse objetivo tão almejado.

A meus pais, por me proporcionarem educação de qualidade, por me oferecerem carinho e compreensão sempre que deles precisei.

A meus irmãos, pela força constante.

Ao professor e amigo Edeilson da Silva Milhomem, pela disposição em me orientar e em chamar a atenção sempre que necessário.

Aos professores Parcilene Fernandes de Brito e Jackson Souza que de alguma forma contribuíram para a concretização deste trabalho.

A todos os meus amigos e colegas pela ajuda mútua e momentos de descontração durante o período da construção do trabalho e toda a graduação.

Depois de 20 anos na escola  
Não é difícil aprender  
Todas as manhas do seu jogo sujo  
Não é assim que tem que ser  
Vamos fazer nosso dever de casa  
E aí então vocês vão ver  
Suas crianças derrubando reis  
Fazer comédia no cinema com as suas leis



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>3</b>
<b>2</b>	<b>AMBIENTES SOCIAIS VIRTUAIS .....</b>	<b>8</b>
<b>3</b>	<b>SISTEMAS DE RECOMENDAÇÃO DE ESPECIALISTAS.....</b>	<b>11</b>
<b>3.1</b>	<b>Identificação de Especialidade.....</b>	<b>13</b>
<b>3.1.1</b>	<b>Recuperação de Informação aplicada a SREs .....</b>	<b>15</b>
<b>3.1.2</b>	<b>Análise de Redes Sociais aplicada em SREs .....</b>	<b>18</b>
<b>3.1.3</b>	<b>Modelos de Ontologias aplicados em SREs.....</b>	<b>22</b>
<b>3.2</b>	<b>Seleção de Especialistas .....</b>	<b>32</b>
<b>4</b>	<b>METODOLOGIA.....</b>	<b>38</b>
<b>4.1</b>	<b>Git.....</b>	<b>38</b>
<b>4.2</b>	<b>Visão Geral .....</b>	<b>40</b>
<b>5</b>	<b>O MODELO.....</b>	<b>42</b>
<b>5.1</b>	<b>Especialidade Demonstrada.....</b>	<b>43</b>
<b>5.2</b>	<b>Especialidade Sugerida e Refutada.....</b>	<b>45</b>
<b>5.3</b>	<b>Especialidade Informada .....</b>	<b>47</b>
<b>5.4</b>	<b>Especialidade Inferida.....</b>	<b>50</b>
<b>5.5</b>	<b>Integração das especialidades .....</b>	<b>53</b>
<b>6</b>	<b>API .....</b>	<b>55</b>
<b>6.1</b>	<b>Manipulação dos Dados.....</b>	<b>56</b>
<b>6.2</b>	<b>Arquitetura dos Dados de Entrada e Saída.....</b>	<b>2</b>
<b>6.3</b>	<b>API de Inferência .....</b>	<b>8</b>

<b>6.4</b>	<b>Exemplo Prático .....</b>	<b>12</b>
<b>7</b>	<b>EXPERIMENTO .....</b>	<b>Erro! Indicador não definido.</b>
<b>8</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>22</b>
<b>9</b>	<b>REFERÊNCIAS .....</b>	<b>24</b>

## 1 INTRODUÇÃO

O excesso de informação disponível em ambientes virtuais originou um problema denominado *information overload*. Este excesso de informações dificulta que os usuários destes ambientes encontrem as informações mais relevantes quando necessitam. Uma maneira eficiente de solucionar o problema de *information overload* é fornecer aos usuários destes ambientes virtuais serviços de informação eficientes (O'Donovan & Smyth, 2005), que facilitem o acesso deles à informação.

A fim de prover tais serviços de informação eficientes surgem técnicas de recuperação da informação e mecanismos de busca que as implementam. Estas técnicas permitem, por exemplo, que seja analisado e classificado (*ranking*) um grande número de informações, facilitando o acesso dos usuários a informações valiosas (O'Sullivan, Wilson, & Smyth, 2002). Atualmente, diferentes softwares, que utilizam estas técnicas, fornecem informações classificadas apenas por meio de listas de resultados armazenados estaticamente (*i. e.* listas de conteúdos indexados), como o Apache Lucene (Apache Lucene, 2011), e o Sphinx (Sphinx Technologies Inc., 2007).

Estas classificações estáticas de informações são, muitas vezes, ineficientes, pois não são personalizadas aos problemas dos usuários (BRUSILOVSKY, 2001). Por exemplo, um indivíduo, ao pesquisar por soluções de problemas de configuração de servidores, pode encontrar a solução para um erro igual ao que ele obteve em seu servidor, no entanto as causas do erro podem ser diferentes, o que deve implicar em diferentes soluções. Neste caso seria necessária uma maior quantidade de conhecimentos sobre o assunto em questão, para reconhecer as causas do erro e implementar uma solução, ou ao menos melhorar o processo de busca por uma. O fornecimento de informações, neste último caso, não é proativo, de modo que o usuário deve conhecer muito bem o problema com qual está se lidando para encontrar informações que possam ajudá-lo.

Uma das técnicas utilizadas para fornecer serviços de informação personalizada e proativa aos usuários da internet é através de sistemas de recomendação. Estes sistemas são programas que apresentam aos usuários um conjunto de itens do seu interesse (Boim, Kaplan, Milo, & Rubinfeld, 2010). Tais sistemas já provaram ser uma importante resposta para o problema de *information overload*, uma vez que fornecem serviços de informação mais personalizados (Melville, Mooney, & Nagarajan, 2002) (O'Donovan & Smyth, 2005), se



comparados<sup>1</sup> a sistemas de busca, uma vez que fornecem um conjunto personalizado de informações, isto é, informações consideradas adequadas para um indivíduo de acordo com seu perfil. Este perfil do indivíduo pode ser formado de diferentes maneiras, como por exemplo, através da análise dos dados de navegação dele (*i. e.* sites visitados), ou através da análise das transações feitas por ele (*i. e.* itens adquiridos, como livros ou ferramentas). Neste sentido, sistemas de recomendação são alternativas que surgiram para substituir determinadas tarefas que antes eram executadas por sistemas de busca, para amenizar o problema de *information overload*.

Mesmo na presença de um conjunto personalizado de informações podem ser encontradas dificuldades para resolver problemas complexos, principalmente problemas que requerem determinado nível de experiência para serem resolvidos (*e. g.* problemas muito relacionados com algum conceito bem específico, como configuração de bancos de dados distribuídos em ambiente Unix). Nestes casos uma alternativa é encontrar um indivíduo com habilidades específicas, que possua esta experiência, para ajudar a resolver estes problemas (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011) (Casciaro & Lobo, 2005). Estes indivíduos podem auxiliar com a solução de problemas deste tipo, pois adquiriram conhecimentos tácitos que os permite, por exemplo, reconhecer mais facilmente as causas dos problemas. Este indivíduo com habilidades específicas pode ser considerado um especialista de domínio – ou simplesmente especialista –, por conta dos conhecimentos por ele adquiridos a respeito de um tema específico.

Entende-se por especialista um indivíduo que possua um nível apropriado de conhecimento, habilidade, ou proficiência, em determinada área (Fazel-Zarandi & Fox, Constructing expert profiles over time for skills management and expert finding, 2011), neste caso diz-se que o indivíduo é especialista, ou possui *expertise*, nesta área. A *expertise* é uma característica difícil de mensurar e de nível variável (Earl, 2001), que reflete o nível de conhecimento de um especialista em uma área. Esta dificuldade de avaliação da *expertise* é devida a vários fatores, como por exemplo, a quantidade de informação disponível sobre as habilidades de cada indivíduo; a frequência com que cada indivíduo muda sua área de atuação, e com isso seu nível de *expertise* em determinadas áreas; e também a falta de um

---

<sup>1</sup>Atualmente os sistemas de busca têm observado a possibilidade de melhoria dos resultados das pesquisas com informações personalizadas, no entanto este tipo de análise foge ao escopo deste trabalho.

padrão definido para mensurar a *expertise*, existem diferentes abordagens de medida, no entanto não há um consenso sobre qual é a mais eficaz.

Uma maneira de auxiliar na solução de problemas altamente específicos então seria através do compartilhamento do conhecimento dos especialistas. No entanto não é uma tarefa trivial realizar este tipo de compartilhamento, uma vez que o conhecimento tácito é de difícil documentação ou formalização (Silva, Costa, Schmitz, & Meira, 2011). Uma das dificuldades de formalização deste tipo de conhecimento é porque este se trata de um conhecimento altamente pessoal e não é facilmente compartilhado entre diferentes indivíduos (Morgan, 2008). Caso o conhecimento tácito pudesse ser formalizado facilmente, poder-se-ia compartilhá-lo no formato de documentos com outros indivíduos. No entanto, dada a dificuldade de formalização deste tipo de conhecimento, uma alternativa para compartilhá-lo é através da recomendação do próprio indivíduo, como especialista. Este tipo de recomendação de indivíduos especialistas se tornou um problema conhecido na literatura como recomendação de especialistas.

Uma das dificuldades do processo de recomendação de especialistas é encontrar maneiras de analisar e validar o nível de *expertise* dos indivíduos (Maybury, 2006). Isto pois, diferentes abordagens de validação de *expertise* podem ser utilizadas em diferentes ambientes, e o processo de recomendação deve ser capaz de empregar a melhor abordagem dada uma situação. Outra dificuldade para recomendar especialistas é a difícil identificação de especialistas proativos, isto é, dispostos a colaborar com seus conhecimentos (Fazel-Zarandi & Fox, Constructing expert profiles over time for skills management and expert finding, 2011). Esta última dificuldade está relacionada à falta de alternativas para identificar as intenções dos especialistas (*e. g.* querem colaborar, ou atrapalhar os membros de uma comunidade). Fatores como, por exemplo, a frequência de contribuição de um especialista com a comunidade são avaliados para analisar as intenções de colaboração dos especialistas, no entanto informações como estas podem ser insuficientes em determinadas circunstâncias.

Neste sentido, a tarefa de recomendar especialistas, segundo (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011), pode ser dividida em duas partes: (1) a identificação de áreas de especialidade dos especialistas e seus níveis de *expertise*; e (2) a identificação do melhor especialista a ser recomendado em determinada situação. As principais pesquisas na área de recomendação de especialistas focam a solução destes dois problemas. Além disso, diferentes ambientes requerem diferentes maneiras de desenvolver a tarefa de recomendação de

especialistas. Estas variações do processo de recomendação podem ocorrer por causa das informações disponíveis no ambiente (*e. g.* existem informações em formato de texto plano apenas, ou informações numéricas); por causa do tipo de especialista procurado (*e. g.* um especialista em filmes pode ser um pessoa que assistiu a vários filmes, e conhece várias histórias, ou um indivíduo que tem um ótimo senso crítico, mesmo não tendo assistido a tantos filmes); ou ainda por causa de outros fatores. Assim existem diversos trabalhos na literatura que objetivam o desenvolvimento de sistemas de recomendação de especialistas, em diferentes contextos. Alguns trabalhos relacionados à área de sistemas de recomendação de especialistas são apresentados na seção 3.

Mais especificamente, a necessidade de identificação de especialistas tende a surgir entre grupos de pessoas que se relacionam, como: empresas (Silva, Costa, Schmitz, & Meira, 2011), comunidades acadêmicas (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011) ou ambientes sociais virtuais (ASVs) (*e. g.* redes sociais web) (Rockenbach, Recomendação de especialistas na plataforma de gestão de conhecimento, 2012). Esta tendência deve-se ao fato de que é uma prática comum dos indivíduos procurarem por informações entre pessoas de grupos que se relacionam (Silva, Costa, Schmitz, & Meira, 2011).

As redes sociais web (RSWs), são ambientes virtuais de interesse das organizações modernas, pois proporcionam ambientes informais e livres onde ocorre a colaboração interpessoal (Silva, Rodrigues, Souza, Salgado, & Meira, 2012). Além disso, tais ambientes representam a estrutura básica da sociedade, através de pessoas e seus relacionamentos (Silva, Rodrigues, Souza, Salgado, & Meira, 2012) (Pimentel & Fuks, 2011). Ademais, uma vez que não exista uma RSWs explicitamente disponível pode existir uma rede social virtual, na qual pessoas influenciam outras não intencionalmente (Lin, Xie, Li, Huang, & Li, 2012) (Song, Tseng, Lin, & Sun, 2006). Neste sentido, ASVs são um interessante tipo de domínio para a aplicação de SREs.

Dada à importância dos ASVs, como as RSWs, para a colaboração e consequentemente produção do conhecimento, o presente trabalho objetiva desenvolver um modelo de recomendação de especialistas para tais ambientes virtuais. Outro fator que motiva o desenvolvimento deste modelo de recomendação é a necessidade de compartilhamento do conhecimento tácito nestes ambientes virtuais, de forma a facilitar e estimular o trabalho em equipe.

Esse desenvolvimento visa desenvolver, e aplicar em um ambiente real, um modelo genérico para recomendação de especialistas em ambientes sociais virtuais para responder a questões como: Quais informações disponíveis em ambientes sociais virtuais são relevantes para a identificação e recomendação de especialistas? E quais outras informações poderiam ser agregadas às disponíveis em ambientes sociais virtuais, a fim de melhorar o processo de recomendação de especialistas? Principalmente através da consideração de que: (1) as informações produzidas e compartilhadas pelos usuários em ambientes sociais virtuais são relevantes para o processo de inferência de especialidade; (2) características provenientes destes ambientes sociais, tais como distância social (*e. g.* amigo ou pessoa mais distante) e contexto (*e. g.* usuários que participam de uma mesma comunidade), influenciam a recomendação de especialistas; (3) informações sobre publicações e conteúdos dos usuários disponíveis em ambientes virtuais, não apenas em ambiente sociais, podem melhorar o processo de recomendação de especialistas; e que (4) SREs devem estimular a colaboração em ambientes sociais virtuais.

O restante do trabalho se organiza da seguinte forma: as seções 2 e 3 apresentam o referencial teórico utilizado como base de conhecimento para o desenvolvimento do projeto. A seção 2 apresenta informações sobre as principais características dos Ambientes Sociais Virtuais; já a seção 3 apresenta os fundamentos e principais abordagens da pesquisa relacionada à recomendação de especialistas, dividida em duas grandes áreas, identificação 3.1 e seleção 3.2 de especialistas. A seção 4 apresenta a metodologia que será utilizada para desenvolver o projeto, a seção 5 mostra o cronograma de criação e execução deste mesmo projeto. O modelo de identificação de especialistas é apresentado na seção 5, onde são mostradas as equações utilizadas para calcular o nível de expertise dos candidatos à especialistas. A seguir, a seção 6 apresenta os detalhes de desenvolvimento e funcionamento da API que implementa o modelo. Por fim são apresentadas as considerações finais e as referências bibliográficas.

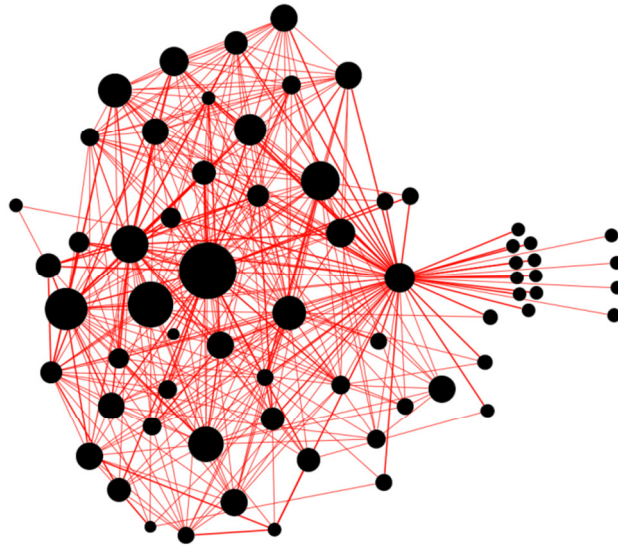
## 2 AMBIENTES SOCIAIS VIRTUAIS

“Redes são naturalmente o foco central da computação ligada à internet” (Douglass, 2010, p. 4). Tais redes, que conectam indivíduos geograficamente dispersos, têm afetado a maneira de relacionamento entre as pessoas. Esta alteração na maneira de relacionamento é evidenciada principalmente pelo surgimento de novos tipos de comunidades *on-line*, como fóruns de discussão (4Chan, 2013) (Imasters, 2013), comunidades de perguntas e respostas (StackOverflow, 2013) (Yahoo Answers, 2013), *Wikis* (Wikipedia, 2013) (Wiktionary, 2013), redes sociais web (Facebook, 2013) (Google+, 2013) dentre outros.

Tais ambientes virtuais tem em comum a característica de possuir redes sociais. Estas redes sociais podem surgir nos ambientes virtuais em sua forma explícita, como o caso das RSWs, ou implícita, nas quais pessoas influenciam as outras sem intenção, como nos *Wikis*. Estas redes sociais são formadas por um conjunto de nós – pessoas, organizações ou outras entidades sociais – conectados por um conjunto de relacionamentos – amizade, parentesco ou troca de informações (Wasserman & Faust, 1994) (Rogers & Kincaid, 1981) (Social Network Analysis: Theory and Applications). Estas relações são estabelecidas com o intuito de adquirir conhecimento e executar alguma tarefa, ou até mesmo lazer e a diversão dos membros destas redes sociais. Segundo Silva (2009) a interação dos indivíduos nesses ambientes virtuais cria uma tendência natural à produção de conhecimento. No entanto, este conhecimento nem sempre é criado de forma organizada, de modo que prover acesso a esta informação pode não ser uma tarefa trivial.

Outra característica destas redes de relacionamento é sua estrutura de grafo frequentemente complexa (Social Network Analysis: Theory and Applications). A Figura 1 apresenta um exemplo destas estruturas. No contexto de redes sociais, a complexidade da estrutura diz respeito à existência de uma grande quantidade de relacionamentos entre os membros da rede. Nessa estrutura, cada indivíduo, representado como um nó do grafo, fornece informações como, por exemplo, as informações sociais, armazenadas nos relacionamentos entre estes nós (*e. g.* confiança ou tipo do relacionamento: amigo, filho). A análise destas informações disponíveis em ambientes sociais virtuais se tornou uma área de estudo conhecida como Análise de Redes Sociais, que analisa as redes sociais de diferentes perspectivas (Social Network Analysis: Theory and Applications). Estas perspectivas vão

desde a rede social como um todo, até partes destas redes, analisando até mesmo a estrutura dos relacionamentos dos membros da rede e seus comportamentos.

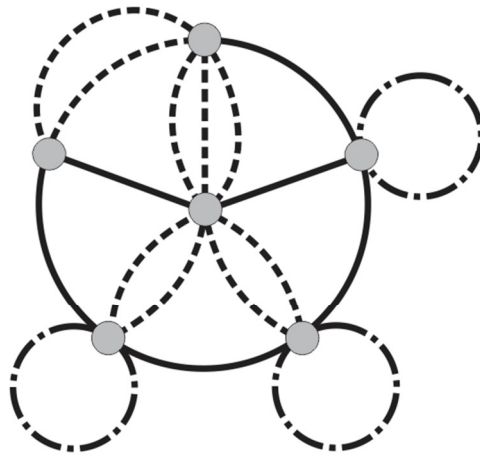


**Figura 1 - Grafo de relacionamentos da rede social de gestão do conhecimento Conecta**

Na Figura 1 é apresentado um grafo de relacionamentos de uma rede social extraído de uma instância da plataforma Konnen (Conecta). Neste caso, cada nó representa um usuário e as arestas que os conectam representam seus relacionamentos. Além disso, o tamanho dos nós representa o quão bem conectado está este nó, isto é, a facilidade deste nó em atingir qualquer outro membro da rede percorrendo um menor caminho no grafo. Um nó pode atingir qualquer outro nó de duas maneiras: (1) através de uma conexão direta, de modo que basta passar por uma aresta para navegar de um nó a outro; e/ou (2) através de conexões indiretas, em que é necessário navegar por uma cadeia de arestas, passando por mais de um nó, para alcançar outro nó no grafo.

Como apresentado anteriormente, cada uma das arestas na Figura 1 pode conter informações como, por exemplo, o tipo do relacionamento entre os indivíduos (e. g. namorado-namorada, chefe-empregado). Além disso, podem existir outras redes de relacionamentos com outros tipos de relacionamentos, como relacionamentos de confiança ou relacionamento de correspondência (*i. e.* troca de mensagens) que não estão explícitos neste grafo. A união destas várias redes de relacionamentos pode compor uma rede social multidimensional, ou multicamadas, que disponibiliza uma maior quantidade de informações para análise. Estas redes multidimensionais permitem a identificação de relacionamentos de

diferentes maneiras, e devem ser representadas como multigrafos<sup>2</sup>. A Figura 2 apresenta um exemplo de multigrafo.



**Figura 2 - Exemplo de multigrafo (Wikipedia, 2013), com alterações**

No caso ilustrado pela Figura 2, arestas com diferentes traçados poderiam identificar diferentes origens dos relacionamentos (*e. g.* relacionamento de confiança ou de correspondência), em uma rede social multidimensional. As informações sociais e outras informações disponíveis em ASVs como o apresentado na Figura 1 podem auxiliar na execução de uma série de tarefas, como identificação de relacionamentos implícitos, análise de opiniões de grupos de indivíduos, recomendação de produtos ou conteúdos baseado no perfil das pessoas ou mesmo na identificação de especialidades dos indivíduos.

Como exposto anteriormente, estes ASV são atuam como ferramentas que promovem a interação, e o compartilhamento de informação, entre seus membros. Uma maneira de estimular o compartilhamento de informação nestes ambientes seria identificar e apontar indivíduos que possuam informações de interesse para outros indivíduos no ASV. Neste caso, o compartilhamento desta informação poderia auxiliar no desempenho de alguma tarefa, solução de problema, ou mesmo na passagem de conhecimento entre indivíduos. Uma forma de realizar esta identificação de indivíduos é através de sistemas de recomendação de especialistas, assunto abordado na seção 3.

---

<sup>2</sup> Um grafo no qual podem existir múltiplas arestas conectando dois nós. Além disso, algumas das definições destes grafos descrevem a existência de identificações individuais para cada aresta, não só a partir dos seus pesos e nós aos quais estão conectadas. Este tipo de identificação de arestas é conveniente para o modelo de rede social multidimensional proposto neste trabalho.

### 3 SISTEMAS DE RECOMENDAÇÃO DE ESPECIALISTAS

Em ambientes virtuais, assim como no mundo real, frequentemente existem situações nas quais há a necessidade de conhecimentos específicos sobre determinado assunto. Por exemplo, para solucionar um problema no motor de um automóvel é necessário um conhecimento específico sobre mecânica, de modo semelhante, para realizar uma alteração de configuração em um sistema de controle de órbita de satélites são necessários diferentes tipos de conhecimento, sobre cálculo de órbitas, mecatrônica, sobre como operar o sistema e etc. Em algumas destas situações a informação pode não estar formalizada em documentos, porém existir apenas na mente de um indivíduo, o conhecimento tácito. Neste caso, esta informação apenas pode ser fornecida por algum especialista que possua conhecimento sobre o assunto em questão.

Para resolver o problema de localização de conhecimento tácito, usuários de ambientes virtuais podem, por exemplo, procurar por um conhecido em sua rede de contatos que tenha domínio sobre o assunto. Porém, como apresentado por (Lin, Ehrlich, Griffiths-Fisher, & Desforges, *SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence*, 2008) estas redes nem sempre são suficientemente grandes e diversificadas de forma a possibilitar a identificação de indivíduos com habilidades específicas. Uma alternativa a esta abordagem seria o envio massivo de mensagens em busca de indivíduos especialistas. No entanto esta tarefa poderia ter um alto custo de execução para o usuário, tanto para o envio de mensagens em massa quanto em tempo de espera por respostas.

Neste contexto de falta de alternativas para a identificação de especialistas surgem os Sistemas de Recomendação de Especialistas (ou SREs) (Silva, Costa, Schmitz, & Meira, 2011), que são sistemas que auxiliam o processo de identificação de indivíduos com habilidades específicas em algum domínio (Maybury, 2006). Estes sistemas são utilizados por:

1. indivíduos, para, por exemplo, localizar especialistas que possam ajudar na solução de algum problema ou trabalhar em alguma equipe; ou
2. organizações, para tarefas como identificar potenciais funcionários ou especialistas de domínio dentro da própria organização que ainda não foram identificados, ou não trabalhem em suas áreas de especialidades.



Os SREs podem ser utilizados como alternativa para ferramentas de gerenciamento de conhecimento organizacional tácito (Rockenbach, Recomendação de especialistas na plataforma de gestão de conhecimento, 2012). Um motivo para utilização destes sistemas é o fato de que soluções de gerenciamento de conhecimento organizacional eficientes não devem manter apenas conhecimentos contidos em documentos, mas também o conhecimento das pessoas (Yimam-seid & Kobsa, 2002). Isto pois estes sistemas de gerenciamento de conhecimento organizacional servem para criação, coleta, organização e disseminação do conhecimento organizacional (Alavi & Smith, 1999).

Além dos motivos previamente apresentados, (Alavi & Smith, 1999) apresentam mais cinco motivos que levam um indivíduo a preferir buscar informações de um especialista ao invés de um documento, que são:

1. informações não documentadas: uma vez que o conhecimento tácito é de difícil documentação é interessante ter acesso a este conhecimento de outras maneiras;
2. informações mais especificadas: ao encontrar um documento com determinada informação, uma pessoa pode se deparar com o fato de a informação não estar bem explicada. Neste caso o indivíduo pode perguntar por mais explicações para o especialista, o que não ocorre com o documento;
3. experiência de outrem: a experiência dos especialistas em uma área em questão pode ajudar a diminuir o tempo gasto na resolução de algum problema ou coleta de alguma informação. Neste sentido os especialistas podem agir como filtros de informação para informações úteis;
4. ausência da necessidade de interpretação: usuários frequentemente encontram fragmentos descontextualizados da informação procurada, de modo que eles necessitam de utilizar tempo para interpretar esta informação. Este tipo de problema pode ser contornado quando a informação é fornecida por um especialista;
5. necessidade de socialização: usuário preferem contato humano a contato com um documento para ajudar no desenvolvimento de alguma tarefa, ou na aprendizagem de algum conteúdo.

Naturalmente, cada uma destas motivações para procurar por um especialista pode surgir em diferentes contextos. Além desta variação na motivação, a mudança do contexto também pode alterar o tipo de especialista a ser encontrado (*e. g.* um especialista com maior conhecimento, ou com maior disponibilidade). Neste caso, os sistemas de recomendação de

especialistas devem solucionar dois tipos de problemas (MCDONALD; ACKERMAN, 1998) (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011): (1) a identificação da especialidade (3.1) e do nível de especialidade dos indivíduos; e (2) a seleção de especialistas (3.2), que deve considerar a pró-atividade e as condições do especialista para fornecer informações em diferentes contextos (Huang, Contractor, & Yao, 2008) (Datta, Yong, & Ventresque, 2011) (Gollapalli, Mitra, & Giles, Ranking authors in digital libraries, 2011). A pró-atividade do especialista deve refletir a probabilidade de ele ajudar, com seus conhecimentos, o desenvolvimento de alguma tarefa; e suas condições devem informar, por exemplo, se o especialista é um usuário ativo em um sistema ou não.

### 3.1 Identificação de Especialidade

O processo de identificação de especialidade consiste na localização dos indivíduos que possuem conhecimentos específicos por meio de evidências implícitas ou explícitas (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011) (MCDONALD; ACKERMAN, 1998). Este processo de identificação de especialistas é um importante e difícil início para a solução de diferentes problemas (*e. g.* localização e compartilhamento de informação) (MCDONALD; ACKERMAN, 1998). Segundo Maybury (2006) o processo de identificação de especialistas é difícil por uma série de razões, dentre elas:

- a quantidade de informações disponíveis que não possuem nenhuma relevância para a identificação de especialidades;
- o primeiro especialista encontrado pode não ser o mais indicado para suprir determinada necessidade;
- o escasso acesso a informações passadas sobre os candidatos a especialistas;
- políticas de privacidade que limitam o compartilhamento de informações sobre a *performance* dos especialistas em determinadas atividades – por exemplo, uma empresa pode não querer divulgar o trabalho desenvolvido por um de seus empregados por ter medo que alguma concorrente possa lhe fazer propostas de serviço melhores;
- não existem consensos que estabeleçam critérios ou qualificações necessários para a obtenção de determinado grau de especialidade;
- especialistas mudam frequentemente suas áreas de atuação – principalmente por se envolverem em diferentes projetos que requerem novas habilidades –, requerendo

abordagens de identificação de especialistas dinâmicas que se mantêm constantemente atualizadas;

Estas e outras razões tornam a identificação de especialistas em uma tarefa não trivial (Maybury, 2006). Neste sentido, surgem na literatura diferentes abordagens para a solução de problemas relacionados à representação do conhecimento de candidatos a especialistas e identificação de especialidades nestes candidatos.

A tarefa inicial em processos de identificação de especialidades de indivíduos é a criação de seus perfis de conhecimento. Estes perfis devem representar os conhecimentos relacionados aos usuários e permitir que sejam realizadas análises sobre estas informações (*e. g.* análises semânticas, análise estatísticas, análises de similaridade). A criação destes perfis de conhecimentos em SREs utiliza informações provenientes de ambientes virtuais, como: conteúdos criados pelos usuários (Datta, Yong, & Ventresque, 2011), atividades on-line dos usuários (Bozzon, Brambilla, Ceri, Silvestri, & Vesci, 2013), ou reputação e autoridade destes usuários (Liu, Chen, Kao, & Wang, 2013).

Para criar estes perfis de conhecimento de usuários, os SREs utilizam uma grande variedade de fontes de informação disponíveis nos ambientes virtuais, como: informações sobre os candidatos a especialistas disponíveis em suas *homepages*, ou no Wikipédia (Wikipedia, 2013); conteúdos publicados ou marcados por usuários, como: publicações científicas ou posts em blogs ou redes sociais web; e até mesmo analisando informações não formalmente documentadas, como e-mails e diálogos dos candidatos a especialistas. Estas informações são utilizadas para apresentar soluções ao problema de identificar especialistas e equipes de especialistas, bem como o conhecimento que eles possuem (Maybury, 2006). No entanto, para identificar estes especialistas ainda é necessária a realização de diferentes análises sobre estes dados.

Para analisar as informações representadas nos perfis de conhecimento dos usuários, são empregadas técnicas derivadas de áreas como a Recuperação da Informação (*e. g.* modelo vetorial e modelos probabilísticos); ou técnicas provenientes da Análise de Redes Sociais, que partem da premissa de que o conhecimento de um especialista pode ser evidenciado por meio da análise dos conteúdos relacionados aos indivíduos com os quais ele se relaciona. Além disso, comumente são utilizados modelos ontológicos que permitem a adição de análises semânticas ao processo de identificação de especialistas. Metodologias como estas permitem que os SREs analisem dados implícitos e explícitos para identificar áreas de conhecimento de

especialistas, e também o nível de especialidade de cada especialista nestas áreas (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011) (Yimam-seid & Kobsa, 2002). Neste sentido, as seções 3.1.1, 3.1.2 e 3.1.3 apresentam respectivamente exemplos de abordagens utilizadas na exploração destas metodologias em diferentes SREs existentes na literatura.

### 3.1.1 Recuperação de Informação aplicada a SREs

Uma técnica comumente utilizada para a identificação de especialidade de um indivíduo é o Modelo Vetorial, que realiza a contagem das frequências de termos relacionados ao indivíduo e cria representações vetoriais dos objetos analisados. Esta técnica é baseada na metodologia TF-IDF<sup>3</sup>. Este método consiste na inferência de especialidade de um indivíduo baseado na frequência dos termos relacionados a ele. Assim, quanto mais um termo está relacionado a um especialista, maior seu nível de especialidade em determinada área correlata ao termo.

O modelo TF-IDF representa a relevância de um termo relacionado a um objeto, ou especialista neste caso, contando duas frequências. A primeira frequência (*tf*) determina qual é o melhor conjunto de termos para representar um objeto (Baeza-Yates & Ribeiro-Neto). Neste caso são contadas as quantidades de ocorrência de cada termo relacionado a um objeto e definido um valor normalizado de frequência, conforme ( 1 ).

$$tf = \frac{freq_{i,j}}{\max_l freq_{l,j}} \quad ( 1 )$$

*j* - índice de um objeto *o<sub>j</sub>*;

*i* - índice de um termo *k<sub>i</sub>* relacionado a um objeto *o<sub>j</sub>*;

*freq<sub>i,j</sub>* - frequência do termo *i*, relacionada ao objeto *o<sub>j</sub>*;

*max<sub>l</sub>freq<sub>l,j</sub>* - maior frequência de um termo *k<sub>l</sub>* em um objeto *o<sub>j</sub>*.

A segunda frequência (*idf*) determina quais termos melhor diferenciam um objeto dentre um conjunto de objetos semelhantes (Baeza-Yates & Ribeiro-Neto). A frequência com a qual os termos se repetem em todos os objetos reduz a sua capacidade de representar. Neste caso, o número total de objetos existentes é dividido pelo número de objetos onde um termo ocorre, este valor também é normalizado, conforme ( 2 ).

---

<sup>3</sup>*TermFrequency – InverseDocumentFrequency*

$$idf = \log \frac{N}{n_i} \quad (2)$$

*N* – número total de objetos considerados na análise;

*n<sub>i</sub>* – número de documentos no qual ocorrem o termo *k<sub>i</sub>*;

Um SRE que utiliza a abordagem TF-IDF é o proposto por (Riahi, Zolaktaf, Shafiei, & Milios, 2012). Este sistema é uma proposta para identificação de especialistas em comunidades de perguntas e respostas. O sistema se concentra em encontrar uma lista de especialistas classificados conforme a probabilidade de responder a uma determinada pergunta da comunidade. O sistema analisa a especialidade dos indivíduos a partir de representações de seus interesses (*i. e.* seus perfis de conhecimento), utilizando duas abordagens: baseado em palavras e baseado em tópicos. Para a abordagem baseada em palavras o sistema utiliza a contagem de termos sobre o histórico de respostas do candidato a especialista na comunidade. Para a abordagem baseada em tópicos o sistema utiliza um algoritmo de segmentação de tópicos para determinar áreas nas quais o candidato a especialista tem interesse.

Outro trabalho que utiliza o modelo TF-IDF para identificação de especialidades de indivíduos é o proposto por (Demartini, 2007). Este trabalho propõe um SRE para criação de perfis de especialidade dos usuários do Wikipédia (Wikipedia, 2013), com base nas informações de artigos editados por estes usuários. O sistema coleta as informações listadas na página de perfil dos usuários do Wikipédia (*i. e.* os artigos listados na página do usuário) e utiliza o algoritmo TF-IDF sobre os conteúdos coletados. Além desta abordagem, o SRE proposto por (Demartini, 2007) também utiliza abordagens baseadas na conexão entre os usuários, como a rede de citações e a similaridade dos perfis dos usuários.

Outra técnica utilizada em recuperação da informação, principalmente para comparação entre objetos, é o Modelo Vetorial (Baeza-Yates & Ribeiro-Neto). Este modelo cria representações dos objetos em formato de vetores em um espaço vetorial. Cada dimensão destes vetores representa uma característica do objeto e o valor associado a esta dimensão representa a intensidade desta característica. Por exemplo, as notas de um aluno de Ciência da Computação poderiam ser representadas por um vetor, como  $\vec{N}_1 = 6\vec{a} + 8\vec{b} + 9\vec{c}$ , neste caso poder-se-ia especificar cada uma das dimensões do vetor (*e. g.*  $\vec{a}$  – Lógica de Predicados,  $\vec{b}$  –

Algoritmos II,  $\vec{c}$  – Engenharia de Software) para que os números fizessem sentido para um observador. Este vetor, por sua vez, poderia ser comparado a outros vetores por meio do cálculo do cosseno entre os vetores.

O cosseno é uma medida de correlação entre dois vetores calculada através do produto escalar entre eles no espaço vetorial (e. g.  $\vec{N}_i \cdot \vec{N}_j$ ). Este produto escalar é definido como o somatório dos produtos dos valores associados a cada uma das dimensões dos dois vetores ( 3 ), ou ainda como o produto dos módulos dos vetores e o valor do cosseno do ângulo formado entre eles ( 4 ). Quanto maior o cosseno entre os vetores (*i. e.* mais próximo de um), mais parecidas são as informações por eles representadas.

$$\vec{n}_i \cdot \vec{n}_j = \sum_{\forall_k \dim(k, \vec{n}_i) \vee \dim(k, \vec{n}_j)} n_{i,k} * n_{j,k} \quad (3)$$

$$\vec{n}_i \cdot \vec{n}_j = |\vec{n}_i| |\vec{n}_j| \cos \theta \quad (4)$$

Neste caso,  $\theta$  é o ângulo formado entre os dois vetores. Quanto maior o ângulo formado entre os vetores, mais diferentes são as direções para as quais eles apontam e, conseqüentemente, menor o valor do cosseno entre eles. Por outro lado, quanto menor este ângulo, maior é o valor do cosseno, logo mais similares são os objetos representados pelos vetores.

Um exemplo de SRE que utiliza o Modelo Vetorial é o SWEETS (Silva E. M., SWEETS: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais, 2009). Este SRE foi desenvolvido em duas versões, uma pró-ativa, que recomenda relacionamentos para os usuários com base em seus conhecimento e interesses, apresentada a seguir; e outra que atua mediante requisições, identificando especialistas em determinadas áreas, apresentada na seção 3.1.3. A primeira versão utiliza o modelo vetorial para recomendação de relacionamentos, enquanto que a segunda utiliza um modelo de ontologia simples para calcular o nível de especialidade de indivíduos em determinadas habilidades. Este algoritmo pode ser implantado em qualquer ambiente que disponibilize as ferramentas básicas para o surgimento de uma *folksonomia*, isto é exista um mecanismo que permita que os usuários classifiquem os objetos disponíveis no sistema com palavras-chave (*i. e.* processo de *tagging*).

Este sistema foi desenvolvido para facilitar o processo de localização e compartilhamento do conhecimento tácito em uma organização.

A primeira versão do algoritmo SWEETS utiliza filtragem baseada em conteúdo para identificar usuários que possuem conhecimentos e interesses em comum. A informação sobre os conhecimentos é extraída a partir do perfil de escrita dos usuários, de modo que os conteúdos por ele publicados dentro de um ambiente virtual organizacional seriam uma possível fonte de informações para o mecanismo de filtragem. Uma vez identificados usuários com perfis semelhantes, o SRE pode recomendar um possível relacionamento entre estes usuários. Para identificar o grau de similaridade entre os usuários, a primeira versão do SWEETS utiliza o Modelo Espaço Vetorial. Neste modelo, os principais termos associados aos usuários, bem como suas frequências de ocorrência, são utilizados para criar vetores que representam os usuários. Em seguida, é utilizado o algoritmo do cosseno para calcular a similaridade entre os usuários representados pelos vetores.

Além da inferência de especialidade utilizando técnicas de recuperação da informação, existem outros sistemas que exploram, por exemplo, as informações sociais dos indivíduos. Tais sistemas utilizam técnicas de análise de redes sociais sobre informações de usuários de ASVs para identificar padrões, como comunidades relacionadas a um tópico, ou interações entre usuários que possam revelar alguma informação útil para o processo de identificação de especialidade. A seção 3.1.2 apresenta alguns sistemas que utilizam este tipo de abordagem.

### 3.1.2 Análise de Redes Sociais aplicada em SREs

Análise de Redes Sociais surgiu como uma técnica empregada pela sociologia moderna para analisar a sociedade e foi adotada por diversas outras áreas científicas, como antropologia, psicologia, ciências da informação, dentre outras (Social Network Analysis: Theory and Applications). Como mencionado anteriormente, redes sociais são estruturas de relacionamentos entre entidades sociais (*e. g.* pessoas, organizações). A Análise de Redes Sociais é a área de pesquisa que busca identificar padrões nestas redes baseados nos relacionamentos de seus membros e componentes (Park, 2003).

A Análise de Redes Sociais consiste em uma série de métodos matemáticos que permitem mapear e identificar padrões em redes sociais. Como estas redes sociais possuem intrinsecamente a estrutura de grafos, algumas técnicas da teoria dos grafos (*e. g.* métricas) foram especializadas para a análise de redes sociais. Em suma, as principais métricas

utilizadas para analisar uma rede social, segundo (Social Network Analysis: Theory and Applications), são:

- a) **betweenness** – a distância que um nó se encontra dos demais nós da rede. Esta medida reflete o número de pessoas as quais um nó está conectado indiretamente, através de suas conexões diretas;
- b) **bridge** – medida sobre as arestas que conectam diferentes setores de um grafo. Uma aresta é uma *bridge* quando excluí-la implica em dividir o grafo em diferentes componentes;
- c) **centrality** – esta métrica indica o poder social de um nó, isto é, o quão bem conectado este nó está, e conseqüentemente sua influência em espalhar uma informação. A métrica *betweenness* também é um tipo de medida de *centrality*;
- d) **centralization** – reflete a forma como estão distribuídos os relacionamentos em uma rede social. Uma rede centralizada possui a maior parte de suas conexões em uma pequena porção de nós, enquanto que uma rede descentralizada possui uma pequena diferença entre a quantidade de conexões de cada nó;
- e) **closeness** – o grau de proximidade de um indivíduo em relação a todos os outros indivíduos da rede. Esta métrica reflete a habilidade de um nó em acessar alguma informação proveniente de rumores na rede social;
- f) **clustering coefficient** – mede a probabilidade da existência de relacionamentos em comum entre dois nós. Um coeficiente alto indica a existência de pequenos grupos bem relacionados;
- g) **cohesion** – reflete a frequência com a qual os nós estão conectados diretamente entre si em pequenos grupos. Estas estruturas com alto índice de conexões diretas em pequenos grupos são chamadas de círculos sociais;
- h) **degree** – o número de relacionamentos de um nó na rede. Pode ser explorado com *in-degree (prestige)* ou *out-degree (radiality)* (*i. e.* número de arestas que chegam a um nó e que saem dele);
- i) **density** – reflete a probabilidade de um nó qualquer estar conectado a outro na rede;
- j) **modularity** – assemelha-se ao *clustering coefficient* e indica a existência de grupos, como *clusters*, no grafo;
- k) **Pagerank** – algoritmo de análise de conexões em um grafo que fornece pesos numéricos aos nós de acordo com estas conexões. O peso do nó representa sua relevância no grafo;
- l) **path length** – a distância entre um par de nós na rede;



- m) *reach* – reflete a habilidade dos membros da rede em alcançar quaisquer outros membros;
- n) *structural cohesion* – o número mínimo de membros (ou nós) que devem ser excluídos de um grupo para desconectá-lo do restante da rede;
- o) *structural equivalence* – refere-se a nós que possuem o mesmo conjunto de relacionamentos na rede;

As informações contidas nas redes de relacionamentos dos usuários são comumente utilizadas em SREs, e as técnicas de análise de redes sociais podem auxiliar na utilização destas informações. Muitos destes SREs são desenvolvidos para RSWs ou outros tipos de ambientes sociais virtuais (ASVs). Sistemas que são desenvolvidos para estes ambientes contam com a vantagem de explorar as informações contidas nos relacionamentos interpessoais, além de outras informações sociais, como conteúdos postados da RSW, informações de perfil e reputação dentre outras. Desta forma, os SREs concebidos para operarem sobre RSWs utilizam várias técnicas provenientes da área de análise de redes sociais.

Diferentes SREs na literatura utilizam conjuntos específicos de informações sociais, como: publicações, mensagens e atividades dos usuários, que podem ser analisadas para identificação de especialidades; avaliações de usuários, que podem apontar indivíduos com especialidade em determinados assuntos; relacionamentos entre usuários, nos quais as interações e o tempo de duração do relacionamento podem apontar áreas nas quais os usuários possuem especialidades; as estruturas das comunidades formadas no ambiente social, que apontam indivíduos essenciais que conectam grupos de especialistas em determinados tópicos; redes de co-autorias, que indicam relacionamentos entre indivíduos e o contexto destes relacionamentos, dentre outras informações.

Um exemplo de SRE que utiliza as informações sociais é o proposto por (Fazel-Zarandi & Fox, Constructing expert profiles over time for skills management and expert finding, 2011) (Fazel-Zarandi & Fox, An Ontology for Skill and Competency Management, 2012), no qual a especialidade de um indivíduo pode ser medida de quatro formas: (1) especialidade demonstrada, a partir de, por exemplo, atividades realizadas pelo usuário; (2) especialidade desconhecida, especialidade informada pelo próprio indivíduo; (3) especialidade sugerida, especialidade informada, por exemplo, por membros de uma comunidade; e (4) especialidade refutada, que também pode ser extraída a partir das opiniões dos membros de uma comunidade, por exemplo.

A primeira métrica dos trabalhos de Fazel-Zarandi e Fox infere a especialidade demonstrada por um indivíduo. Para inferir este valor o sistema considera que os candidatos a especialistas devem ser observados e testados (através de atividades). O indivíduo possui o nível de especialidade correspondente às atividades que ele conseguiu solucionar corretamente. Para inferir este nível de especialidade, o sistema assume que um indivíduo que consegue resolver uma atividade  $a$ , que está em um nível  $l$  das habilidades, deve ser capaz de resolver todas as subatividades  $a_l$  de  $a$ , de níveis iguais ou inferiores. Esta afirmativa pode ser formalizada no predicado expresso em ( 5 ).

$$\forall_{s,l,a} \text{enablingsuite}(s,l,a) \equiv (\forall_{a_1} \text{enables}(s,l,a_1) \supset \text{subactivity}(a_1,a)) \quad (5)$$

Nestes trabalhos, a inferência das métricas (2), (3) e (4), do trabalho de Fazel-Zarandi e Fox, ocorre a partir das declarações dos membros de uma rede social – explícitas ou implícitas – sobre as habilidades dos candidatos a especialistas. Os autores consideram uma declaração como um predicado do tipo  $\text{declare}(r, p, s, l)$ , desta forma um usuário  $r$  pode declarar que outro  $p$  possui um nível  $l$  na habilidade  $s$ . Estas declarações podem ser positivas ou negativas, de modo que possam incrementar ou decrementar o nível de habilidade a ser sugerido pelo sistema para o candidato a especialista. Além disso, o sistema de recomendação apenas considera declarações sobre habilidades  $s$  nas quais o declarante  $r$  possui algum nível mínimo de proficiência.

Outro SRE que utiliza a estrutura das redes sociais para inferência de especialidade é o *SmallBlue* (Lin, Cao, Liu, Papadimitriou, Sun, & Yan, 2009) (Lin, Ehrlich, Griffiths-Fisher, & Desforges, SmallBlue: People Mining for Expertise Search, 2008). Este SRE analisa as redes sociais dos especialistas a fim de localizar estruturas de comunidades relacionadas a determinados tópicos, e assim encontrar quais especialistas estão no centro (*centrality* (3.1.2.a)) da estrutura destas comunidades e quais atuam como conexões entre estas comunidades. Este sistema localiza especialistas e comunidades utilizando técnicas de *Data Mining*, Recuperação da Informação e Inteligência Artificial sobre um grande conjunto (*e. g.* e-mails, palavras-chave de buscas) de informações que depende da contribuição dos indivíduos (*i. e.* as pessoas devem liberar o acesso destas informações, uma vez que podem ser consideradas confidenciais).

As informações sociais do indivíduo podem colaborar com o processo de identificação de especialidade dos indivíduos, por exemplo, as informações de seus relacionamentos, ou grupos dos quais faz parte, pode indicar que o indivíduo seja membro de uma comunidade de pesquisa sobre algum tema específico. Outra maneira de melhorar a inferência de especialidade é flexibilizar a maneira de reconhecer as habilidades (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011), isto é, reconhecer similaridades entre áreas de conhecimento afins e inferir o nível de especialidade de indivíduos em áreas correlatas às que eles atuam. Uma maneira de flexibilizar o processo de identificação de especialidade é através da utilização de modelos de ontologias que adicionam contexto ao processo de inferência, tornando possível, por exemplo, a identificação de especialidades relacionadas. Neste sentido a seção 3.1.3 apresenta alguns sistemas que utilizam este tipo de metodologia.

### 3.1.3 Modelos de Ontologias aplicados em SREs

Uma das tarefas de um sistema de recomendação de especialistas é identificar especialidades em indivíduos, por exemplo, a partir de contextos aos quais estes indivíduos estão relacionados. Neste sentido, entender estes contextos, bem como os conceitos envolvidos neles, fornece condições de realizar melhores análises das informações disponíveis (*e. g.* conteúdos publicados, mensagens), e conseqüentemente uma melhor identificação de especialistas. Para garantir este entendimento de conceitos e contextos podem ser utilizadas técnicas de descrição semântica das informações disponíveis para análise. Este tipo de descrição permite que sejam realizadas análises quantitativas (*e. g.* identificar conceitos em conteúdos mesmo quando estes não são apresentados de forma explícita) e qualitativas (*e. g.* identificar o nível de apresentação dos conceitos nos conteúdos – superficial, aprofundado) das informações (FAZEL-ZARANDI; FOX, [s.d.]).

Para criar essas descrições semânticas podem ser empregadas ontologias de domínio, que são vocabulários que permitem a formalização de conceitos de forma que estes possam ser processados por computadores. Estes vocabulários permitem a descrição do relacionamento entre os conceitos de um domínio, ou contexto, bem como a especificação das regras deste domínio. As subseções Ontologia e Ontologia Simples apresentam algumas definições destes vocabulários e alguns exemplos de SREs que utilizam técnicas que aplicam ontologias na identificação de especialistas.

## **Ontologia**

Ontologias são modelos que representam um entendimento comum, formalizado e compartilhado por um grupo de indivíduos, sobre determinada teoria de um domínio (Borst, Akkermans, & Top, 1997) (Gruber, 1993). O grupo de indivíduos, no neste caso, trata-se de um conjunto de especialistas de domínio. Já a característica de formalização destes modelos serve para garantir que ela seja interpretada por computadores, pois deve ser estabelecida de acordo com alguma linguagem formal (*e. g.* RDF, OWL). Uma vez que é necessário um entendimento de um grupo de especialistas de domínio sobre um conhecimento a ser formalizado em uma ontologia, não há ontologia sem uma comunidade que a suporte (Mika & Akkermans, *Towards a New Synthesis of Ontology Technology And Knowledge*, 2004). Neste caso, a comunidade de suporte à ontologia é formada por membros especialistas em um domínio, e é responsável por explicitar a realidade do domínio com um conjunto de restrições e conexões sobre o vocabulário utilizado pela comunidade (Mika, *Ontologies are us: A unified model of social networks and semantics*, 2007).

Desta forma, ontologias podem ser interpretadas como um vocabulário específico, utilizado para descrever uma realidade, e um conjunto de regras explícitas a respeito do significado das palavras e suas relações em um domínio (Guarino, 1998). Estas ontologias são utilizadas para adicionar significado às informações por elas representadas, de modo que o computador consiga realizar análises semânticas destas informações. Frequentemente ontologias são representadas como grafos direcionados, nos quais os nós representam conceitos e as arestas as relações entre estes conceitos (Giunchiglia & Zaihrayeu, 2007). Os conceitos são definidos como um conjunto de entidades, que são representados lexicalmente por palavras em linguagem natural. A estrutura básica das ontologias é uma taxonomia na qual os relacionamentos entre os termos podem ser definidos como regras (*e. g.* é-um, é-parte-de, está-em). A Figura 3 apresenta a estrutura de uma ontologia representada como grafo.

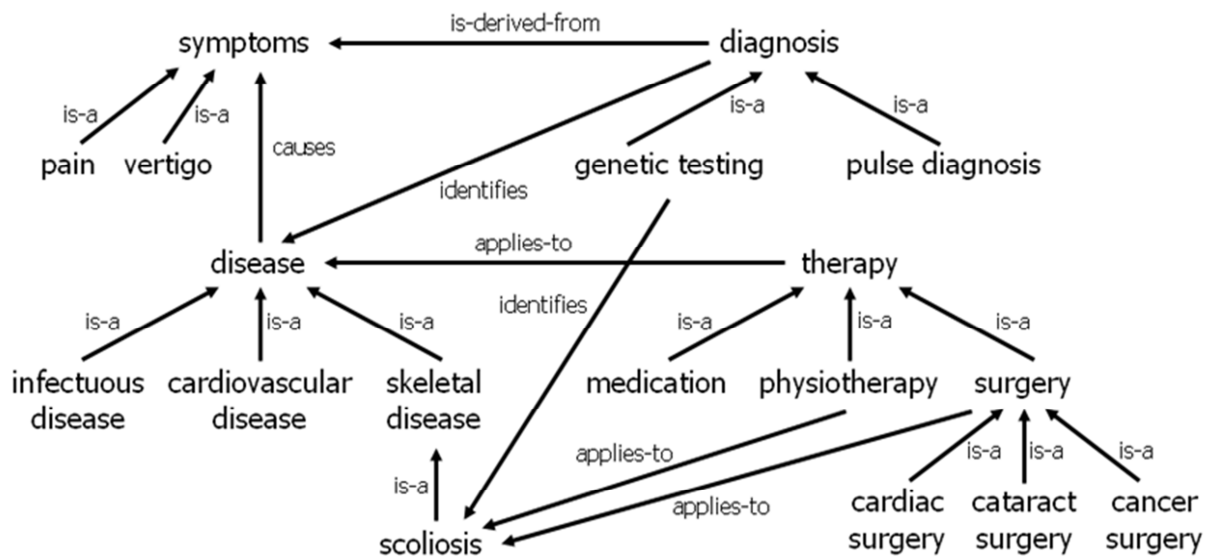


Figura 3 - Ontologia representada no formato de grafo (Reimer, 2004).

A ontologia apresentada na Figura 3 foi derivada de uma hierarquia de conceitos com a adição de novos relacionamentos. O conceito de hierarquia de conceitos será explicada *a posteriori*. Neste caso, a Figura 3 mostra como podem ocorrer os relacionamentos em uma estrutura de ontologia. Além dos rótulos, as arestas podem possuir inerentes a si algumas propriedades, como regras e restrições (*e. g.* uma fisioterapia é aplicada em pessoas vivas; um humano normal possui, no máximo, dois olhos). O conjunto destes objetos, bem como suas relações, regras, restrições, axiomas e etc. que podem ser representados nos sistemas formam o domínio da ontologia. O vocabulário desta ontologia deve conter, além do formalismo destes objetos, as relações entre eles. A Listagem 1 apresenta um exemplo de definição de ontologia para um objeto.

```
(define-instance UM (number)
  "Um é o elemento neutro da multiplicação."
  :axiom-def (forall ?x
    (=> (number ?x)
      (and (= (* ?x UM) ?x) (= (* UM ?x) ?x))))
  := 1)
```

Listagem 1 - Definição de objeto em ontologia (GRUBER, 1993, adaptado)

Na Listagem 1 é definida a instância UM, a marcação (number) identifica a classe a qual pertence a instância UM. Neste caso, o axioma de definição (axiom-def) de UM define que, para todo número  $x$ , a multiplicação de  $x$  e UM, ou UM e  $x$  resulta em  $x$ , ou ainda conforme (6). Em (6) é apresentada outra forma de representação da Listagem 1, porém utilizando linguagem matemática.

$$\forall_x \text{number}(x) \rightarrow (UM * x = x) \wedge (x * UM = x) \quad (6)$$

A classe número (*number*) deve ter sido definida em outra parte da ontologia, de modo que possa ser utilizada na definição da instância UM. A marcação  $:= 1$  informa que a palavra-chave que representa a instância UM é “1”.

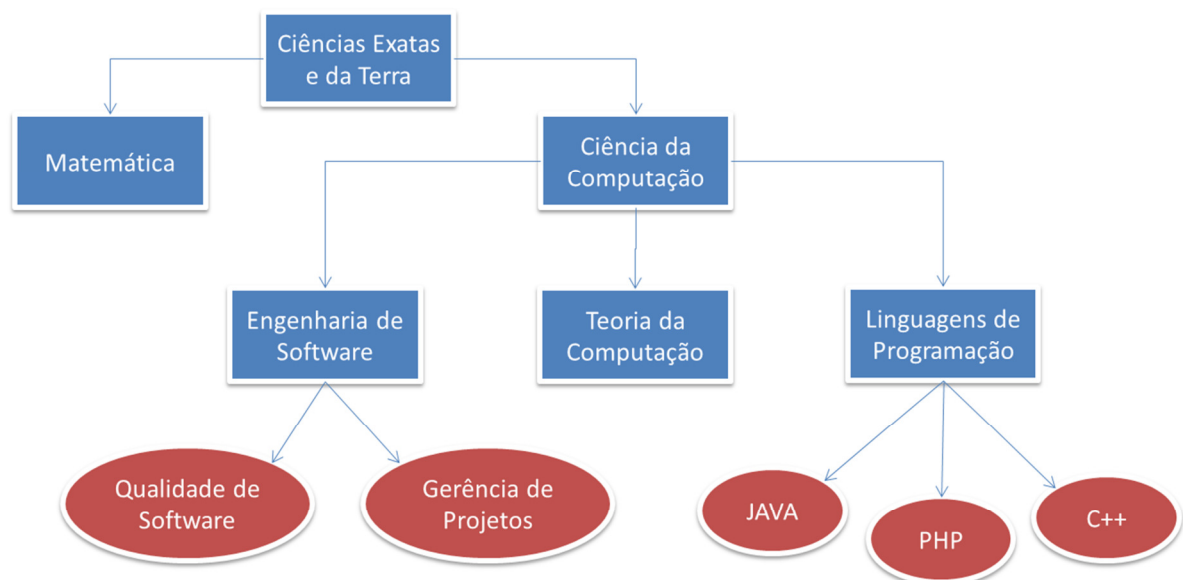
### ***SRE que aplica Ontologia***

O trabalho de (Mochol, Wache, & Nixon, Improving the accuracy of job search with semantic techniques, 2007) apresenta um protótipo de portal de empregos que utiliza ontologias de vagas de emprego e perfis de candidatos a estas vagas para recomendação de empregos. Em seu trabalho, os autores buscam amenizar o problema de sobrecarga de informação gerada pela grande quantidade de portais de emprego existentes. Segundo (Mochol, Wache, & Nixon, Improving the accuracy of job search with semantic techniques, 2007), esta grande quantidade de portais dificultou o processo de localização de vagas de empregos pelos candidatos a estas vagas, pois torna difícil a criação de uma visão geral das vagas relevantes a estes candidatos. Para amenizar este problema, são utilizadas tecnologias de web semântica nos domínios de recrutamento e gerenciamento de habilidades.

A análise semântica no portal de empregos é utilizada para flexibilizar as consultas dos usuários. Estas consultas são criadas a partir de seus perfis cadastrados no portal de empregos. Assim, candidatos devem cadastrar suas informações no portal de empregos a fim de que lhes sejam recomendadas vagas de trabalho. Uma vez cadastrados os candidatos, o portal cria uma ontologia para representação de recursos humanos (*i. e. human resources ontology, HR-ontology*) a partir das informações cadastradas pelos usuários. Esta ontologia, além de representar as informações dos usuários deve subsidiar a pesquisa semântica de vagas de emprego. Os modelos utilizados para definir a *HR-ontology* são derivados de um vocabulário comum utilizado no cenário da web semântica para descrever ocupações, setores industriais e habilidades de trabalho (Mochol, Paslaru, & Simperl, Bontas. Practical Guidelines for Building Semantic eRecruitment Applications, 2006). De modo semelhante, as vagas de emprego devem ser representadas na ontologia, a fim de que seja suportada a pesquisa semântica.

A pesquisa semântica é uma técnica que combina descrições feitas com um vocabulário controlado e um conhecimento sobre determinado domínio de aplicação (Mochol, Wache, & Nixon, Improving the accuracy of job search with semantic techniques, 2007). A partir da ontologia criada com as informações dos candidatos e das vagas de emprego, é gerada uma lista dos melhores candidatos para cada vaga. Para criar esta lista, são agrupados os perfis dos candidatos e as descrições das vagas em “clusters temáticos”. Por exemplo, são criados clusters a partir das informações sobre habilidades dos candidatos, cargos desejados, salários almejados dentre outras. Estes clusters são gerados comparando dados como habilidades, categorias de trabalho, cargos, salários e outras informações. Cada um dos clusters de descrições de vagas é comparado com os clusters de perfis de candidatos do mesmo tema, de forma a determinar a similaridade entre o perfil do candidato e a descrição da vaga. Para o cálculo de similaridade, é utilizada a correspondência semântica dos conceitos da descrição da vaga e do candidato; e a similaridade taxonômica, determinada pela distância entre os conceitos, extraída de suas posições na hierarquia de conceitos.

Um exemplo de hierarquia de conceitos é apresentado na Figura 4. Este exemplo foi criado a partir da divisão de áreas do conhecimento estabelecida pela CAPES (CAPES, 2013). Neste caso são apresentadas algumas áreas de conhecimento e suas subáreas (quadros); e alguns dos tópicos que podem surgir dentro destas áreas (elipses).



**Figura 4 – Hierarquia de conceitos criada a partir das áreas de conhecimento**

Neste caso, uma maneira de computar a distância entre conceitos é através da quantização do número de saltos necessários para navegar de um conceito a outro na

hierarquia. Por exemplo, a distância entre “C++” e “JAVA” é dois, enquanto que a distância entre “C++” e “Gerência de Projetos” é quatro. Quanto maior a distância entre os conceitos, menor é a relação entre eles. Outras metodologias podem ser utilizadas para calcular a distância entre conceitos na hierarquia, por exemplo, contabilizar saltos com pesos diferentes (*e. g.* saltos entre quadros poderiam ter peso maior que saltos entre elipses).

Outras maneiras de definição de ontologias podem surgir além das definições explícitas da realidade de um domínio definidas pelos membros de uma comunidade, ou especialistas de domínio. Por exemplo, as ontologias simples, que se originam, por exemplo, de um processo de análise de palavras-chave relacionadas a objetos disponíveis em um ambiente virtual. As ontologias simples são o objeto de discussão da próxima seção.

### **Ontologia simples (*lightweight ontology*)**

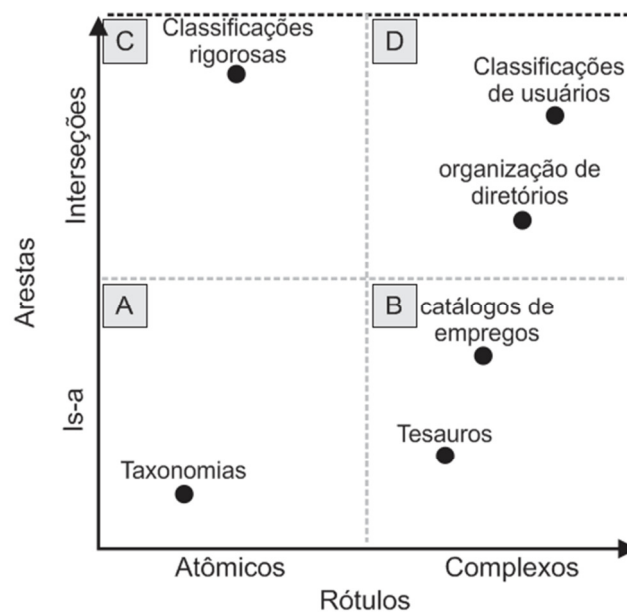
Diferentes autores propõem modelos nos quais a semântica, representada pelas ontologias, pode surgir naturalmente das interações de um grande número de indivíduos (Mika, *Ontologies are us: A unified model of social networks and semantics*, 2007) (Giunchiglia & Zaihrayeu, 2007) (Aberer, et al., 2004). Este tipo de modelo permitiria uma maneira mais fácil de criar e gerenciar ontologias em ambientes complexos, como a web, uma vez que não haveria a necessidade de um esforço para formalizar os conceitos de determinado domínio, esta formalização surgiria espontaneamente da interação entre os indivíduos. A existência de um grupo de indivíduos conectados por uma implícita rede social, cooperando na produção e organização de conteúdos de acordo com uma ontologia auto estabelecida entre os membros do grupo conduziria o processo de surgimento natural desta semântica (Mika, *Ontologies are us: A unified model of social networks and semantics*, 2007). Esta ontologia auto estabelecida é produto de discussões, definições e estabelecimento de conexões e regras sobre o vocabulário da comunidade assim que surgem as necessidades de interação entre seus membros.

No entanto, para aproveitar o surgimento natural desta semântica existe a necessidade de modelos que permitam a explicação deste processo e sua formalização (Mika, *Ontologies are us: A unified model of social networks and semantics*, 2007). Neste sentido, existem diferentes pesquisas na literatura sobre como estabelecer uma arquitetura semântica para este processo de surgimento de ontologias. Uma das arquiteturas existentes para formalizar este processo de surgimento natural da semântica mediante as interações de um grupo de indivíduos é a ontologia de pesos leves, (ou ainda ontologia simples ou *lightweight*



*ontologies*). Este processo utiliza como base as estruturas de taxonomias, que servem naturalmente como pontos de partida para definição de ontologias e podem ser concebidas como um tipo destas ontologias simples (Mika & Akkermans, Towards a New Synthesis of Ontology Technology And Knowledge, 2004). Ontologias simples consistem em ontologias que utilizam apenas a estrutura básica de taxonomias (Giunchiglia & Zaihrayeu, 2007).

Giunchiglia e Zaihrayeu (2007) definem uma estrutura de ontologia simples como sendo formada por uma tripla  $\langle N, E, C \rangle$ .  $N$  é um conjunto finito de nós relacionados pelo conjunto de arestas  $E$  – a tupla  $\langle N, E \rangle$  consiste em um grafo. Além disso,  $C$  é um conjunto finito de conceitos formalizados em uma linguagem formal  $F$ , de modo que para cada nó em  $N$  existe um, e apenas um, conceito em  $C$  que o represente. São exemplos destas ontologias simples: taxonomias, catálogos de empregos, tesouros (dicionário de ideias afins) e organização de diretórios em pastas ou na web. De acordo com a utilidade da ontologia simples ela pode ser classificada conforme a Figura 5.



**Figura 5 – Tipos de rótulos e arestas em ontologias simples (Giunchiglia & Zaihrayeu, 2007)**

A Figura 5 resume as características que diferem entre ontologias simples. De acordo com a complexidade das entidades é possível separar estas ontologias em dois grupos: com rótulos simples que representam conceitos atômicos (e. g. espécies de animais) e com rótulos complexos que comportam, por exemplo, locuções substantivas (e. g. temas de discussão de eventos científicos, como “Sistemas de estímulo a colaboração em redes sociais web”). Além disso, estas ontologias simples também podem ser divididas quanto ao tipo de arestas, que podem ser simplesmente arestas de especificações, caracterizadas por relacionamentos do tipo

“é-um” (e. g. um humano é um mamífero, que por sua vez é um animal) ou arestas representam interseções de conceitos, de forma que um nó pai especifica o contexto de um nó filho em uma sentença (e. g. as frases: “Castelo na Inglaterra” e “Castelo de Areia” conduzem o leitor a diferentes entendimentos da entidade “castelo”).

Em um ambiente de difícil controle como a web, este tipo de representação de ontologias simples tem recebido atenção. Esta atenção é devida ao fato de que uma ontologia simples pode abranger vários domínios (Silva E. M., SWEETS: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais, 2009), diferente de uma ontologia de domínio, que apenas trata de um único domínio. Uma ontologia simples pode surgir, por exemplo, de uma folksonomia, pois ambas compartilham a mesma estrutura taxonômica básica (Giunchiglia & Zaihrayeu, 2007).

Folksonomias são sistemas que permitem uma classificação colaborativa de objetos de acordo com marcações (*i. e. tags*, termos, palavras-chave) definidas por usuários em sistemas de informação (Peters & Becker, 2009) (Meira, Silva, Costa, & Jucá, 2011). Este processo de classificação pode ser utilizado para representação e indexação de conhecimentos de uma comunidade. O processo de categorização de objetos, com *tags*, a partir da visão individual de usuários em uma comunidade, cria uma classificação destes objetos que origina a folksonomia. Como vários usuários podem categorizar os mesmos objetos, surge uma classificação livre e coletiva que representa um conhecimento coletivo compartilhado entre os membros da comunidade (Meira, Silva, Costa, & Jucá, 2011), como caracterizado nas ontologias.

Existem diferentes formas de criação de ontologias simples, por exemplo, uma ontologia simples pode ser gerada a partir de uma comunidade que atribui *tags* a um conjunto de *links* de web sites, como por exemplo, no Delicious (Delicious, 2013), um serviço que permite que usuários descrevam e compartilhem seus *link* com *tags* (*i. e.* palavras-chave). A ontologia, neste caso, surge a partir da relação existente entre estas *tags*, de modo que o peso entre estas *tags* pode ser definido de acordo com sua co-ocorrência em diferentes objetos. A Figura 6 apresenta uma estrutura de grafo criada a partir das co-ocorrências das *tags* do Delicious em diferentes *links*.

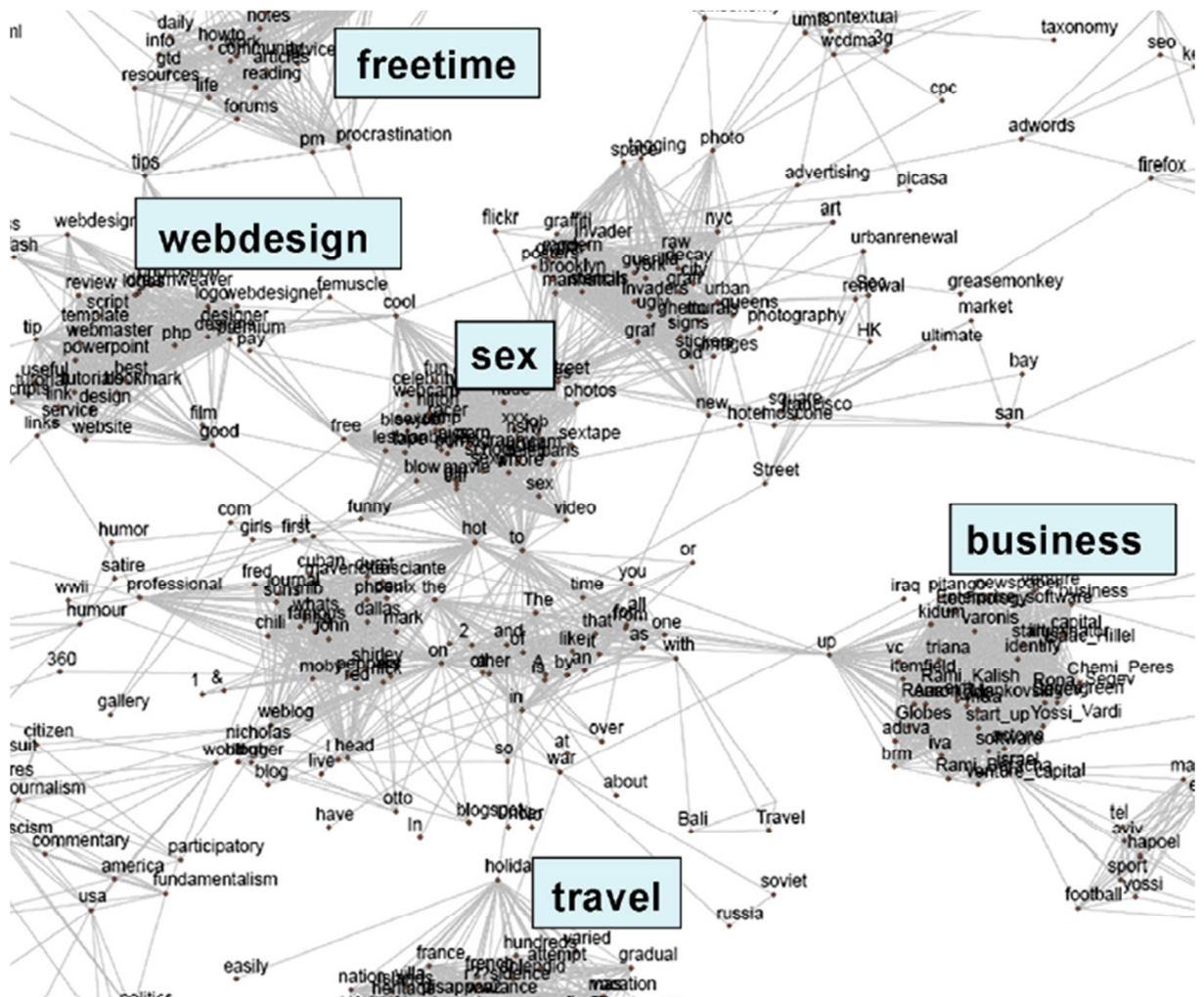
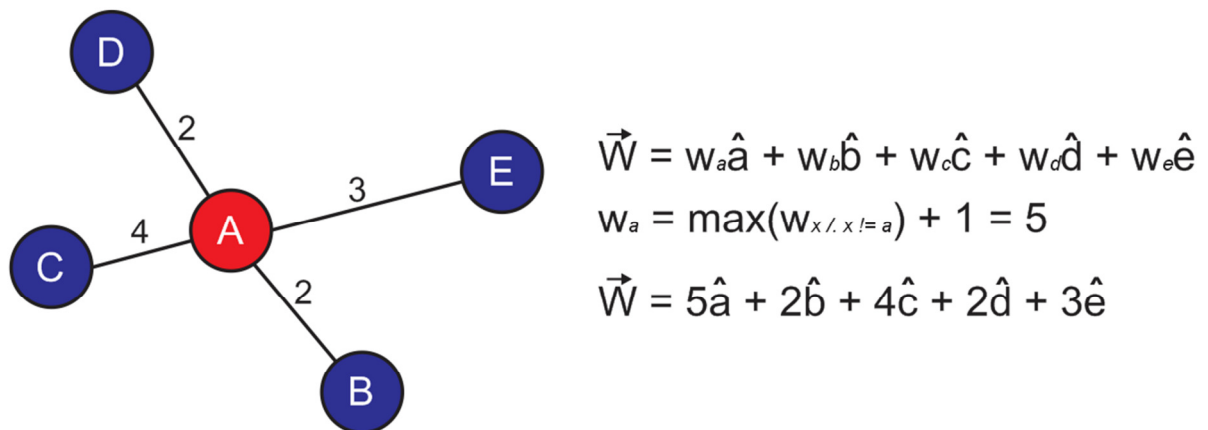


Figura 6 - Grafo gerado a partir das associações de co-ocorrências de *tags* no Delicious (Mika, Ontologies are us: A unified model of social networks and semantics, 2007)

Na Figura 6 os termos mais internos aos *clusters* são os termos especializados nos domínios dos quais fazem parte (e. g. *XXX* no cluster *sex*, *template* no cluster *webdesign*), enquanto que os termos entre os clusters são os termos mais generalistas (e. g. *other*, *time*, *you*). Análises como esta podem mostrar os interesses dos usuários do serviço Delicious e a maneira com a qual estes interesses estão relacionados. Por exemplo, através da análise do grafo pode ser notada uma rápida conexão entre os clusters *freetime* e *sex*, no entanto para navegar do cluster *business* para o cluster *sex* seria necessária uma maior quantidade de saltos entre os nós do grafo. Estes comportamentos nos relacionamentos entre as *tags* mostram a proximidade ou distância entre diferentes temas no contexto da comunidade do Delicious, no entanto em diferentes comunidades estas relações podem variar.

### *SWEETS : SRE baseado em Lightweight Ontology*

A segunda versão do SWEETS (Silva E. M., SWEETS: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais, 2009) – previamente citado na seção 3.1.1 – é um exemplo de SRE que utiliza uma ontologia simples para quantificar as especialidades dos usuários. É nesta versão que as ferramentas para surgimento de uma folksonomia são necessárias, pois o SRE cria a ontologia simples através da co-ocorrência das *tags* em relação aos objetos representados no ambiente virtual. Nesta ontologia, as *tags* que co-ocorrem em um mesmo objeto estão relacionadas; além disso, a quantidade de objetos nos quais as mesmas *tags* co-ocorrem define o peso do relacionamento entre elas. Após a definição da ontologia simples, são criadas representações vetoriais para cada um dos conceitos presentes na ontologia. Para criar a representação vetorial de um conceito são utilizados os pesos dos relacionamentos com os demais conceitos, conforme a Figura 7.



**Figura 7 - Representação vetorial de um conceito A - (Silva E. M., SWEETS: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais, 2009) adaptado.**

Como apresentado na Figura 7, os pesos dos relacionamentos são utilizados para criar a representação vetorial de cada conceito. De modo semelhante são criadas representações vetoriais para os usuários do ambiente virtual, no entanto é considerada a frequência dos termos em relação aos usuários, e não em relação à ontologia. Por fim, o algoritmo cosseno é novamente utilizado para calcular a similaridade entre os vetores dos conceitos e os vetores dos usuários.

Após o processo de identificação de especialista ainda resta a tarefa de determinar quais os melhores candidatos a suprirem determinada necessidade. Esta tarefa é conhecida na literatura como seleção de especialistas. Modelos de seleção de especialistas combinam técnicas e informações semelhantes aos modelos de identificação de especialistas (e. g. técnicas de recuperação da informação e análise de redes sociais), no entanto com um objetivo

diferente, analisar as motivações e capacidades dos especialistas em contribuir com seu conhecimento. A tarefa de seleção de especialistas é uma etapa fundamental para os SREs, uma vez que permite o melhor aproveitamento dos resultados da análise de especialidades, neste sentido a seção 3.2 apresenta abordagens utilizadas neste processo de seleção.

### 3.2 Seleção de Especialistas

Os SREs, além de identificarem especialistas, ainda podem determinar o melhor especialista para determinada situação. Por exemplo, procurar por uma resposta rápida, ou procurar por um possível membro de equipe deve necessitar de diferentes abordagens. Uma resposta rápida pode ser fornecida por um indivíduo que esteja ativo (*i. e. on-line*) em um ambiente virtual, por outro lado, um membro de equipe deve ser não apenas uma pessoa com conhecimentos específicos, mas também uma pessoa confiável. Com o propósito de encontrar os especialistas propícios para cada situação, SREs utilizam dados como: similaridade entre usuários, estatísticas de redes sociais e motivações de usuários. Estas, e outras informações, fornecem subsídios para execução do processo de seleção de especialistas. O processo de seleção de especialistas consiste na metodologia de classificação, ou *ranking*, do conjunto de especialistas identificados, de acordo com regras específicas definidas conforme o contexto da recomendação, ou do SRE. Este processo de seleção é utilizado principalmente para melhorar o aproveitamento dos resultados em um processo de recomendação de especialistas.

Para melhorar sua eficiência, comumente mecanismos de pesquisa têm utilizado pesquisas com resultados personalizados, de forma a facilitar o processo de localização da informação. A seleção de especialistas é o processo da recomendação de especialistas que possibilita a personalização do conjunto de resultados obtidos (*i. e. classificação dos especialistas*) (Fazel-Zarandi, Devlin, Huang, & Contractor, 2011). Desta forma, a utilização de mecanismos de seleção de especialistas é fundamental para o processo de recomendação de especialistas, uma vez que esta adiciona a possibilidade da personalização de resultados.

Apenas identificar as especialidades de indivíduos pode não ser suficiente para a transferência do conhecimento tácito, pois ao recomendar um especialista a um indivíduo ainda resta a tarefa de contatar o especialista e requisitar sua ajuda. No entanto o especialista pode não estar disponível a prestar ajuda (*e. g. falta de tempo, conhecimento insuficiente*); ou mesmo o indivíduo pode não querer solicitar ajuda de um especialista em específico (*e. g. falta de confiança*).

Na literatura, existem diferentes metodologias de seleção de especialistas, uma vez que a maior parte dos SREs utiliza este processo. Para classificar estes especialistas SREs utilizam sistemas de classificação baseados, por exemplo, na: similaridade entre perfis de usuários (Gollapalli, Mitra, & Giles, Similar researcher search in academic environments, 2012); relevância (*i. e.* um usuário conhecido e que atua geralmente como fonte de informações – pode ser analisado com a *centrality* (3.1.2.a)) em uma rede social) (Bozzon, Brambilla, Ceri, Silvestri, & Vesci, 2013), e reputação dos especialistas (Liu, Chen, Kao, & Wang, 2013), dentre outras metodologias. Nos parágrafos seguintes serão apresentadas abordagens de exploração de cada uma destas metodologias de seleção de especialistas. Foi dada ênfase adicional às metodologias supracitadas por estarem relacionadas às metodologias que serão utilizadas no desenvolvimento do projeto.

Gollapalli, Mitra e Giles (2012) propõem uma metodologia para classificação de especialistas voltada para comunidades acadêmicas. De acordo com o modelo proposto é possível utilizar a similaridade entre pesquisadores, baseado em seus perfis de especialidades, para selecionar especialistas em áreas semelhantes ou relacionadas. O modelo proposto por Gollapalli *et. al.* (2012) explora diferentes técnicas para a análise de similaridade entre os especialistas a partir da representação do perfil do pesquisador, como:

- representação com vetores com os termos contidos nos conteúdos associados com o pesquisador, neste caso a comparação dos perfis dos pesquisadores utiliza técnicas como *tf* e *idf*, e o algoritmo cosseno;
- representação em termos de distribuições de probabilidades, onde o algoritmo de divergência de Kullback-Leiber (Shalizi, 2006) é utilizado para computar a similaridade das distribuições de probabilidades;
- representação em uma distribuição de tópicos, neste caso, como pesquisadores geralmente atuam em diferentes áreas é calculada uma distribuição de especialidades entre estas áreas de atuação dos pesquisadores. A similaridade entre os perfis é calculada pela probabilidade de desenvolvimento de um perfil, por exemplo  $s_1$ , a partir de um outro  $s_2$ ;
- representação como subespaços, esta abordagem é similar a abordagem vetorial, no entanto são utilizadas matrizes para representar os perfis dos pesquisadores. A similaridade entre os perfis é calculada utilizando matrizes de densidade definidas com técnicas *tf-idf*.

As informações para modelagem dos perfis de especialidades, e posterior classificação dos especialistas provêm de suas publicações e páginas pessoais acadêmicas. Segundo os autores, o problema de seleção de especialistas em áreas similares ainda não foi explorado na literatura, de forma que este problema é modelado em seu trabalho. Trabalhos similares ao processo de recomendação de pesquisadores em áreas semelhantes são encontrados em pesquisas sobre predição de coautores (CHEN *et al.*, 2011) (HUANG; GILES; NEUROSCI, 2008) (NEWMAN, 2004) (NEWMAN, 2001), porém a recomendação de pesquisadores encontra indivíduos mesmo quando eles estiverem afastados em uma rede de coautoria (GOLLAPALLI; MITRA; GILES, 2012).

O trabalho apresentado por Gollapalli, Mitra e Giles (2012) apresenta diferentes técnicas de representação de perfis de indivíduos e mostra metodologias de comparação destes perfis. Neste caso, os perfis mais similares são classificados como mais adequados pelo processo de seleção. Os modelos de representação escolhidos pelos autores deriva dos modelos de representação comumente utilizados em problemas de procura por especialistas, como representação vetorial dos conteúdos de autoria de um usuário e distribuições de probabilidades sobre os tópicos nos quais os autores trabalharam (GOLLAPALLI; MITRA; GILES, 2012).

Outro trabalho, apresentado por Bozzon *et al.* (2013) mostra uma metodologia de seleção do conjunto de pessoas mais bem informadas para solução de uma necessidade. A metodologia é proposta para atuar em uma rede social web (RSW), de modo que considera informações disponíveis nesta rede (*e. g.* perfis, atividades dos usuários) como fonte de evidências para a identificação e seleção de especialistas. Neste caso, o trabalho apresenta uma série de experimentos a fim de demonstrar que a análise de atividades sociais, relacionamentos e conteúdos compartilhados em uma rede social podem melhorar a efetividade de um SRE. Estes experimentos são realizados a partir de informações extraídas das RSWs Twitter (Twitter, 2013), Facebook (Facebook, 2013) e LinkedIn (LinkedIn, 2013).

Bozzon *et al.* (2013) definem uma abordagem de seleção de especialistas em redes sociais web que classifica os membros de um grupo social de acordo com seus conhecimentos a respeito de determinado tema. Os autores propõem o uso de informações do comportamento dos usuários na RSW, geradas através de suas atividades sociais, para suprir o problema de limitação da quantidade de informações fornecidas nos perfis dos usuários destas redes. Para classificar os membros dos grupos sociais, são consultadas as informações relacionadas a cada

um destes membros e, com o auxílio de técnicas de processamento de texto como o modelo vetorial e os fatores *tf-idf*, são atribuídos *scores* de relevância aos usuários. As informações sociais são organizadas em grupos com diferentes pesos. Cada grupo é definido conforme a distância das informações em relação ao usuário.

As informações utilizadas podem estar relacionadas aos usuários de forma direta ou indireta no modelo proposto por Bozzon *et. al.* (2013). Informações diretamente relacionadas aos usuários são as informações criadas (*e. g.* atualizações de status no Facebook) ou marcadas (*e. g.* likes no Facebook) pelos usuários. Enquanto que informações indiretamente relacionadas aos usuários são aquelas disponibilizadas em grupos dos quais fazem parte os usuários (*e. g.* posts em páginas do Facebook). De acordo com esses tipos de relacionamentos entre os usuários analisados (*i. e.* usuários alvos da análise) e as fontes de informações a eles relacionadas, os autores definem a distância entre o usuário e as informações a ele relacionadas conforme a Tabela 1.

**Tabela 1 – Conteúdos e suas respectivas distâncias do usuário alvo da análise**

<b>Distância</b>	<b>Fontes de Informação</b>
<b>0</b>	Perfil do usuário alvo.
<b>1</b>	Conteúdos criados ou marcados pelo usuário alvo; Grupos dos quais o usuário alvo faz parte; Perfis de usuários seguidos pelo usuário alvo.
<b>2</b>	Conteúdos compartilhados em grupos dos quais o usuário alvo participa; Conteúdos criados ou marcados pelos usuários seguidos pelo usuário alvo; Grupos dos usuários seguidos pelo usuário alvo; Perfis de usuários seguidos por usuários seguidos pelo usuário alvo.

De acordo com as distâncias definidas na Tabela 1 o modelo proposto por Bozzon *et. al.* (2013) infere pesos para relacionar a especialidade evidenciada em um conteúdo e o candidato a especialista. O valor proveniente destes pesos e das especialidades evidenciadas é utilizado para classificar os usuários. Este trabalho apresenta diferentes maneiras de explorar o comportamento de usuários de RSWs. A principal contribuição deste trabalho é identificar, através de experimentos, algumas das informações relevantes a serem analisadas em RSWs. As principais constatações evidenciadas no trabalho são: (1) informações de perfis de usuários são menos efetivas que as informações contidas em conteúdos criados ou marcados pelos



usuários; (2) conteúdos produzidos por outros usuários contribuem para a melhoria da precisão da inferência de especialidades; (3) o Twitter é aparentemente a RSW mais eficiente para identificação de especialidades, quando comparado com Facebook e LinkedIn; (4) O Twitter é mais eficiente para identificação de especialidades nas áreas como computação, ciência, esportes e tecnologia, enquanto que o Facebook se destaca em áreas como locais, músicas e filmes; (5) O LinkedIn – surpreendentemente – apresenta-se como a menos eficiente das redes sociais, quando comparado com Twitter e Facebook, para a tarefa de identificação de especialidades em todos os domínios.

Liu *et al.* (2013) propõem um modelo de recomendação de especialistas para comunidades de perguntas e respostas que identifica especialistas de acordo com categorias de questões a serem respondidas. Neste modelo são consideradas informações sobre a relevância, autoridade e reputação do usuários dada uma categoria de especialidade. A relevância do usuário representa o conhecimento do usuário sobre a categoria em questão; enquanto que a análise da autoridade do usuário (*i. e.* nível de *centrality* dos usuários 3.1.2.c)) deriva de uma análise dos relacionamentos gerados a partir das perguntas e respostas na comunidade; por fim a reputação do usuário é extraída das informações de melhores respostas fornecidas pelo usuário, que são votadas por outros usuários da comunidade. O modelo proposto foi avaliado em um conjunto de dados extraídos da comunidade de perguntas e respostas do Yahoo (Yahoo Answers, 2013) em Taiwan. Para construir os perfis de usuários e fornecer suas classificações, o modelo proposto por Liu *et al.* (2013) utiliza uma abordagem híbrida que combina as informações supracitadas.

O perfil de conhecimento dos usuários é construído de acordo com perguntas e respostas previamente realizadas na comunidade. Uma vez de posse destas informações é possível construir vetores de representação do perfil de conhecimento dos usuários. Para criar estes vetores de representação dos perfis de conhecimento dos usuários, primeiramente são definidos vetores para cada par  $\langle \text{pergunta}, \text{resposta} \rangle$  dos usuários. A importância da especialidade evidenciada no par é influenciada por três fatores: a quantidade de votos recebida pela respostas – usuários podem votar na melhor resposta; a avaliação da questão – realizada pelo autor da pergunta; e o fator cronológico – quanto mais recente for a resposta, mais relevante ela é para a construção do perfil de conhecimento do usuário. Para estimar o conhecimento do usuário em determinada categoria, são ponderadas todas as respostas do usuário a perguntas desta categoria (*i. e.* média ponderada), onde os pesos são definidos pelos três fatores mencionados.

Para recomendar os usuários, os autores propõem uma abordagem híbrida que combina as informações de conhecimentos, reputação e autoridade do usuários para criar uma lista de especialistas a serem recomendados. A fim de mensurar o conhecimento dos usuários em relação a uma pergunta da comunidade é comparado o perfil de conhecimento do usuário na categoria da pergunta com um perfil de conhecimento criado para a própria pergunta. O *score* de conhecimento do usuário em relação à pergunta é calculado através da similaridade – algoritmo cosseno – entre os vetores que representam estes dois perfis de conhecimento. O cálculo da reputação dos usuários é dado pela relação entre a quantidade de melhores respostas – os usuários da comunidade podem determinar a melhor resposta para uma pergunta – fornecidas pelo usuário e a quantidade total de respostas fornecidas por ele. Por fim, para calcular a autoridade de um usuário em uma categoria, é criado um grafo direcionado de representação das perguntas e respostas da comunidade. Cada pergunta é um nó origem, e as respostas fornecidas a esta pergunta são nós destino, os nós neste caso representam os autores das perguntas e respostas. Os *scores* de autoridade dos usuários são definidos através de algoritmos de análise de links (*i. e.* conexões em grafos), neste caso são utilizados os algoritmos como o *PageRank* (Brin & Page, 1998), e o HITS (Kleinberg, 1999).

O trabalho proposto auxilia na identificação de especialistas para responder determinadas perguntas, e promove o compartilhamento da informação através da redução da espera por especialistas que possam responder estas perguntas (LIU *et al.*, 2013). No trabalho ainda são apresentadas diferentes abordagens de construção de perfis de especialistas: (1) a partir do modelo vetorial e dos termos relacionados com os usuários; (2) a partir da qualidade das respostas oferecidas pelos usuários; e (3) a partir da consideração das categorias das perguntas na comunidade. Uma das contribuições principais do trabalho é a constatação de que a reputação dos usuários pode melhorar a qualidade da construção de perfis de conhecimento em problemas de recomendação de especialistas.

As diferentes metodologias de identificação e seleção de especialistas apresentadas neste trabalho devem contribuir no desenvolvimento de um modelo genérico para recomendação de especialistas. Este modelo se caracteriza como genérico no sentido de poder ser aplicado em qualquer ambiente que seja capaz de fornecer entradas e interpretar as saídas em padrões pré-estabelecidos, independente das fontes dessas informações. Desta forma, os trabalhos apresentados nas seções anteriores visam à criação de uma base de conhecimentos que auxilie na definição deste modelo. Neste sentido, a seção 4 apresenta a metodologia que será utilizada no desenvolvimento do modelo de recomendação de especialistas.

## 4 METODOLOGIA

O presente trabalho é um projeto para o desenvolvimento de um modelo genérico de recomendação de especialistas. Este modelo atuará sobre informações fornecidas de acordo com um padrão pré-estabelecido para identificar e selecionar especialistas de domínio. Além disso, este modelo será desenvolvido com base nas métricas de especialidades definidas por Fazel-Zarandi *et al.* (2011) e Fazel-Zarandi e Fox (2011), que são: (1) especialidade demonstrada, (2) especialidade desconhecida, (3) especialidade sugerida e (4) especialidade refutada.

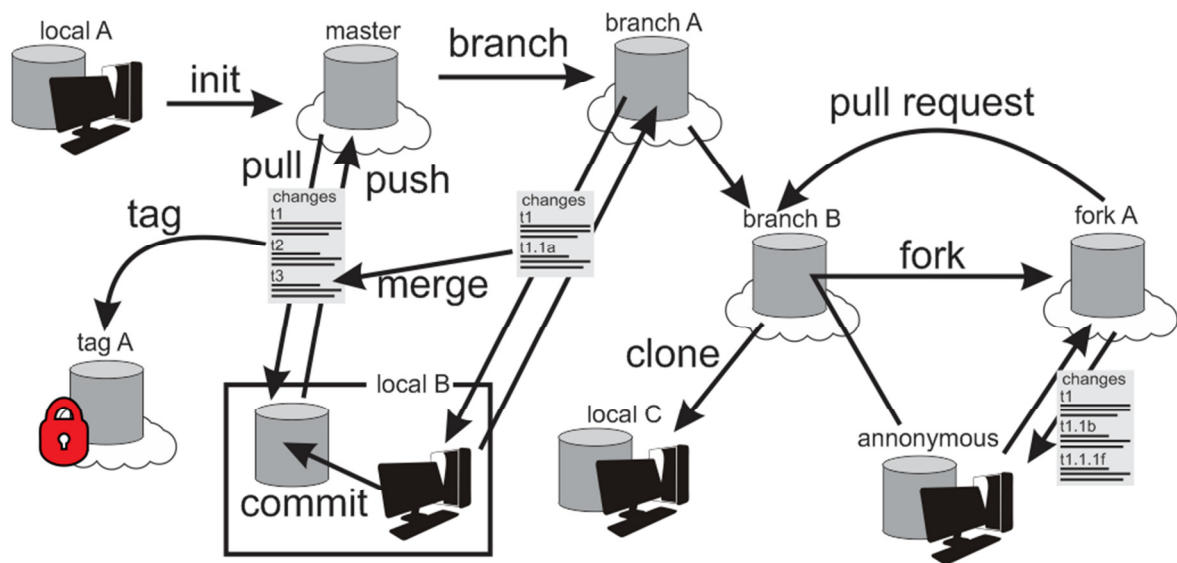
O modelo de recomendação de especialistas será implementado utilizando a linguagem de programação JAVA, uma linguagem de programação livre que permite maior flexibilidade de execução quanto a sistemas operacionais. Este modelo será desenvolvido de forma modular, de modo que colaboradores futuros possam facilmente integrar novos módulos a este projeto. Para o versionamento do código desenvolvido será utilizado o protocolo GIT de versionamento distribuído de códigos fonte (Git, 2013). A seção 4.1 apresenta os detalhes sobre o protocolo GIT. A seguir, a seção 4.2 apresenta uma visão geral da metodologia empregada no desenvolvimento do modelo e da API que o implementa.

### 4.1 Git

Git é uma ferramenta de controle de versionamento de códigos fonte (Chacon, 2009). Este sistema de versionamento é utilizado em desenvolvimentos desde projetos de código livre, como o Gephi (Gephi, 2013), ou até mesmo no desenvolvimento de softwares por grandes corporações como o Google (Google, 2013). Segundo Chacon (2009), sistemas de controle de versão servem para salvar as mudanças de um arquivo, ou conjunto de arquivos, de modo que se possa recuperar qualquer estágio deste arquivo. O controle destas mudanças aumenta a segurança do desenvolvimento do software, pois permite que sejam desfeitas quaisquer alterações – problemáticas, ou não – no código, além de permitir a análise de mudanças no projeto com o passar do tempo.

O GIT é um sistema de versionamento distribuído que permite a colaboração de equipes geograficamente dispersas. Em resumo, esta ferramenta permite a desenvolvedores o acompanhamento de projetos de software através de ações como: envios de alterações no

código fonte, recuperações de versões, liberação de versões estáveis, controle de soluções paralelas para um mesmo software (*branches*) dentre outras operações. Os códigos do projeto devem ser armazenados em um repositório que implemente a interface GIT e atenda as requisições dos clientes. Assim deve existir uma interface básica que estabelece a comunicação entre os clientes e o repositório, além disso, esta interface deve observar a característica de desenvolvimento distribuído do software. De fato, o GIT estabelece uma interface padrão para o versionamento de código fonte, as instruções mais comuns do protocolo são apresentadas na Figura 8.



**Figura 8 - Instruções mais comuns do protocolo GIT**

Para inicializar um repositório GIT é utilizada a instrução `init`. Esta instrução é executada sobre uma pasta local que contenha o projeto a ser versionado (i. e. projeto do qual deseja-se salvar o histórico de alterações do arquivo). Durante a execução deste comando devem ser especificadas informações como o endereço do repositório *master* e o protocolo de comunicação com o repositório que funcionará sob o GIT (e. g. SSH, HTTPS). Após a criação do repositório, cada desenvolvedor pode enviar e requisitar o código fonte do repositório através das instruções `push` e `pull`, respectivamente. Antes de enviar as alterações para o repositório distribuído, o desenvolvedor deve salvá-las em um repositório local, através da instrução `commit`.

O GIT ainda disponibiliza a funcionalidade de criação de um repositório que herda todas as características de um estado definido de outro repositório. Desta forma é possível criar soluções paralelas para um problema do software e, posteriormente, mesclar estas soluções no projeto principal. Estes repositórios paralelos podem ser criados com a instrução

`branch`, e mesclados através do comando `merge`. Além dos *branches* os desenvolvedores podem copiar um estado de um repositório e defini-lo como uma versão estável do software, a partir do comando `tag`. Os repositórios definidos como *tags* não devem mais ser alterados, pois são versões estáveis e – teoricamente – prontas para distribuição. Todos os tipos de repositórios podem ser copiados em uma versão local através do comando `clone`.

Por fim, outra funcionalidade interessante do GIT é a possibilidade de desenvolvimento em *forks*. Estes *forks* são repositórios paralelos semelhantes aos *branches*, no entanto podem ser criados mesmo por desenvolvedores que não façam parte da equipe original do projeto. Desta forma, se um projeto de código livre é disponibilizado publicamente, um desenvolvedor autônomo pode criar um repositório paralelo, através do comando `fork`, realizar alterações, e posteriormente solicitar que suas alterações sejam integradas ao repositório original, com a funcionalidade de `pull request`. Ao realizar um *pull request* o desenvolvedor autônomo submete duas alterações à inspeção da equipe original do projeto, que pode julgar se é conveniente ou não a integração das alterações. Este tipo de funcionalidade é interessante para projetos que são abertos para colaboração espontânea de desenvolvedores, como o Gephi (Gephi, 2013), uma ferramenta para análise de grafos que será apresentada na seção **Erro! Fonte de referência não encontrada.**

## 4.2 Visão Geral

O desenvolvimento do projeto será dividido nas seguintes fases:

- Definição das métricas utilizadas para inferência de especialidade:
  - estudo e discussão dos principais trabalhos influenciadores do modelo;
  - seleção das métricas que serão utilizadas;
- Definição da estrutura de implementação do modelo:
  - definição dos pacotes do modelo;
  - definição das classes do modelo;
  - criação da estrutura de classes para a implementação;
- Organização do desenvolvimento:
  - definição da ordem de dependências entre pacotes e classes;
  - elaboração do cronograma de desenvolvimento do modelo;
- Desenvolvimento do modelo de recomendação de especialistas (NetBeans) (está fase será executada em iterações com as seguintes etapas):

- implementação das classes previamente definidas, com versionamento (GIT);
- elaboração dos casos de testes do modelo;
- implementação das classes de testes;
- testes e correções das classes;

## 5 O MODELO

O trabalho de Fazel-Zarandi & Fox (2011) apresenta quatro tipos de especialidades dos indivíduos. Estas especialidades podem estar expressas de forma **Demonstrada**, de modo que o indivíduo executou alguma tarefa ou tem condições de provar que possui determinada especialidade; **Sugerida**, neste caso a especialidade do indivíduo foi apontada por outro indivíduo, ou grupo de indivíduos; **Desconhecida**, quando um indivíduo afirma possuir uma especialidade, porém não existem formas de comprová-la; e **Refutada**, no caso de haver provas que permitam concluir que o indivíduo não possui determinada especialidade. Estas quatro formas de manifestação das especialidades dos indivíduos foram utilizadas como base para a proposta de um modelo de recomendação de especialistas.

Além destes quatro tipos de especialidades, propostos por Fazel-Zarandi & Fox (2011), o presente trabalho sugere mais um estado de especialidades de indivíduos. Neste caso, acredita-se que determinada especialidade de um indivíduo pode ser inferida a partir de outras de suas habilidades. Por exemplo, existe uma probabilidade de que um indivíduo que possua *expertise* em projetos de bancos de dados e sistemas de bancos de dados distribuídos, possua também conhecimentos de sistemas gerenciadores de bancos de dados, como o MySQL. Mesmo na ausência de provas da existência deste conhecimento, ou até mesmo de indicações de que ele exista, esta probabilidade não deve ser desconsiderada. Uma forma de identificar este tipo de especialidades é através da análise do perfil de especialidades dos indivíduos, caso alguma área de especialidade do indivíduo seja identificada, diz-se que a especialidade foi **Inferida**.

O presente trabalho propõe um modelo que visa à identificação de especialidades de indivíduos baseados nestes cinco tipos de especialidades. Cada um destes cinco tipos indica ora uma fonte diferente de informações (*e. g.* informações sobre especialidades sugeridas são informadas por indivíduos em um ambiente social, enquanto que informações de especialidades demonstradas necessitam de provas); ora formas diferentes de analisar as mesmas informações (*e. g.* uma análise diferenciada das provas das especialidades demonstradas pode originar informações sobre especialidades inferidas). A partir destes cinco tipos de especialidades foram definidas fontes de informações e heurísticas que às analisam com o propósito de identificar especialidades em indivíduos em ambientes sociais virtuais. As seções 5.1, 5.2, 5.3 e 5.4 apresentam os cinco tipos de especialidades analisadas no modelo

proposto. Por fim, a seção 5.5 apresenta como esses tipos de especialidades são unificados para a obtenção de um valor geral de especialidade dos usuários.

### 5.1 Especialidade Demonstrada

Uma especialidade de um indivíduo pode ser considerada demonstrada quando existem provas de que o indivíduo realmente possui tal especialidade. No presente modelo, foi considerado que uma forma de provar uma especialidade é através de produções científicas relacionadas ao domínio da referida especialidade. Desta forma, a existência de publicações científicas, de autoria de um indivíduo e relacionadas a um domínio, indica que este indivíduo possui certo nível de *expertise* neste domínio. Neste caso, o nível de *expertise* deve ser refletido pela quantidade (*e. g.* número de artigos publicados) e pela relevância das publicações (*e. g.* conceito QUALIS da CAPES sobre o veículo de publicação) dos indivíduos. Em ( 7 ) é apresentado como é avaliada uma especialidade demonstrada de um usuário  $u$  em um domínio  $d$ .

$$D(u, d) = \sum_{\forall p \in P(u, d)} \frac{pw(p)}{|a(p)|} \quad (7)$$

$u$  – Usuário alvo da análise de especialidades

$d$  – Domínio que se deseja verificar a existência de especialidade no usuário  $u$

$P(u, d)$  – Conjunto de publicações do usuário  $u$  no domínio  $d$

$p$  – Uma publicação, neste caso pertencente a  $P(u, d)$

$pw(p)$  – Peso da publicação  $p$  (relacionado à relevância da publicação)

$a(p)$  – Conjunto de autores da publicação  $p$

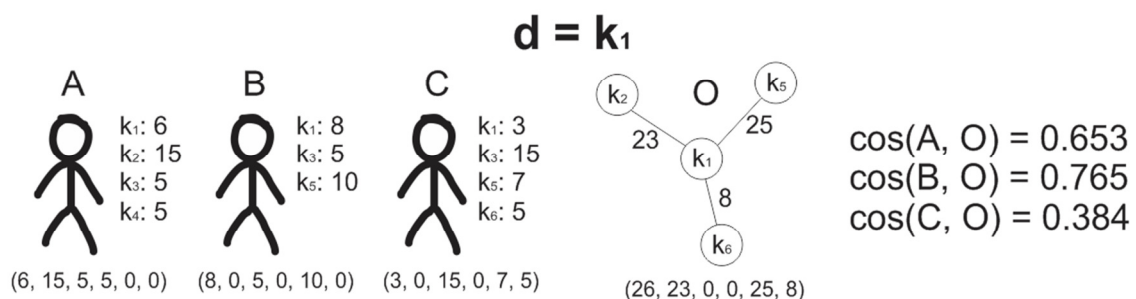
Em ( 7 ) é considerado que o usuário  $u$  é coautor de publicações em um domínio  $d$ . A partir da quantidade destas publicações, a quantidade de colaboradores delas e suas relevâncias, é determinado o valor de  $D(u, d)$ , que reflete o nível de *expertise* do usuário  $u$  no domínio  $d$ . Dado o caráter colaborativo das publicações científicas, durante o desenvolvimento de ( 7 ) surgiu o questionamento sobre como mensurar o quanto cada autor conhece sobre cada domínio. A primeira alternativa utilizada foi dividir igualmente o



peso da publicação entre os seus autores. Porém, esta divisão pode não ser a melhor solução para o problema, pois os autores podem não possuir conhecimento iguais sobre os temas abordados na publicação. Neste caso foram criadas duas hipóteses de solução para o problema:

- i) o questionamento sobre o quanto cada autor conhece sobre os domínios abordados na publicação não representa problema para a análise de especialidades;
- ii) utilizar uma ontologia de pesos leves para determinar qual autor deve possuir um peso maior para cada publicação.

Na primeira hipótese, acredita-se que o fato de serem avaliadas diferentes publicações de cada autor supriria a necessidade de medir os conhecimentos de cada autor por domínio na publicação. Isto pois, o autor que possuísse maior número de publicações em determinado domínio conseguiria atingir maiores valores de  $D(u, d)$ . Na segunda hipótese é considerada a necessidade de descobrir o peso de autoria para cada autor da publicação. Para isso seria utilizada uma ontologia de pesos leves que relaciona os termos que representam o domínio com outros termos presentes na ontologia. Neste caso, a partir da relação entre os termos da ontologia determinar-se-ia um conjunto de termos relevantes ao domínio avaliado. Em seguida, iria-se analisar a correlação deste conjunto de termos e os termos presentes em uma nuvem de *tags* relacionada a cada usuário. Quanto maior a relação entre estes dois conjuntos de termos, maior o peso do autor na publicação. Esta análise de correspondência de termos seria executada com o algoritmo cosseno, sendo que os pesos das palavras da ontologia seriam os pesos das arestas que as ligam com o termo que representa o domínio. A Figura 9 apresenta este processo de forma visual.



**Figura 9 - Cálculo da relevância dos autores em um domínio  $d$**

No caso apresentado na Figura 9 o domínio avaliado é  $k_1$ . Ao procurar por termos relacionados a ele na ontologia de pesos leves foi identificado que  $k_1$  possui relações com os termos  $k_2$ ,  $k_5$  e  $k_8$ , com os pesos apresentados na Figura 9. A partir dos pesos destes

relacionamentos na ontologia e dos pesos dos termos associadas a cada autor, através de suas nuvens de *tags* são criados vetores para comparação e determinação da relevância de autoria de cada autor no domínio. Através do algoritmo do cosseno foi determinado que o autor que possui maior relevância de autoria é o *B*. Note que o peso, no vetor criado a partir da ontologia, para o termo que representa o domínio ( $k_1$ ) é igual ao maior de todos os outros pesos, adicionado de uma unidade. Não se sabe ao certo a relevância do peso principal ( $k_1$ , neste caso) em relação aos outros ( $k_n$ ) no vetor construído a partir da ontologia, no entanto sabe-se que ele deve ser maior que os outros. Uma vez que fossem utilizados estes valores de relevância de autoria ( 7 ) sofreria uma alteração, resultando em ( 8 ).

$$D(u, d) = ar(u, d) \sum_{\forall p \in P(u, d)} \frac{pw(p)}{|a(p)|} \quad ( 8 )$$

*ar(u, d) - relevância de autoria do usuário u no domínio d, calculado através do algoritmo do cosseno.*

É possível notar em ( 8 ) a ausência da divisão igualitária dos pesos de autoria de cada publicação pelo peso de relevância determinado a partir do algoritmo do cosseno. Não é conhecido qual das duas hipóteses apresentadas resultará em melhores resultados para a análise, no entanto a segunda hipótese apresenta um maior custo computacional, uma vez que deve analisar um maior número de elementos e de forma mais complexa que a primeira. Futuramente almeja-se testar as duas hipóteses a fim de obterem-se conclusões mais precisas.

## 5.2 Especialidade Sugerida e Refutada

Para que um indivíduo possua uma especialidade sugerida é necessário que haja sugestões de especialidades feitas por outros indivíduos do ambiente social. Desta forma, caso um indivíduo *A* sugira que outro *B* possua *expertise* em um domínio, esta sugestão deve ser refletida no valor geral de *expertise* de *B* naquele domínio. O peso referente a esta sugestão deve estar relacionado à relevância (e. g. reputação) do usuário *A* no ambiente social. Em ( 9 ) é apresentado como é obtida a *expertise* sugerida de um usuário em um domínio.

$$S(u, d) = \sum_{s \in S(u)} r(u_s) \quad ( 9 )$$

*s - sugestão*

$S(u)$  – Sugestões de habilidades para o usuário  $u$

$r(u)$  – relevância do usuário  $u$ .

$u_a$  – Usuário que realizou a avaliação  $a$

Em ( 9 ) são somados os valores de relevância de todos os usuários  $u_s$  que sugeriram que o usuário  $u$  possuía *expertise* no domínio  $d$ . Esta soma garante que quanto mais sugestões de especialidades existam, mais evidente seja que o usuário  $u$  possua especialidade no domínio  $d$ , e maior o valor de  $S(u, d)$ . A relevância de um usuário em um ambiente social pode ser determinada de várias formas. Por padrão, o modelo proposto neste trabalho considera como relevância do usuário a sua reputação no ambiente e sua maturidade. Estes valores de reputação e maturidade do usuário são obtidos com o algoritmo T-SWEETS (SILVA et al., 2012).

Durante a concepção da ( 9 ) foi questionado o quanto a popularidade do usuário  $u$  poderia afetar a análise de reputação. Isto pois, um usuário mais popular poderia receber muitas sugestões de especialidades o que faria que ele possuísse um valor alto de  $S(u, d)$  que não reflete bem sua *expertise*. Além disso, este aumento do valor de  $S(u, d)$  de um usuário prejudica a avaliação de especialidades sugeridas de outros usuários no ambiente, estes usuários ficariam com valores baixos de especialidades sugeridas se comparados com a do usuário popular. No entanto, a quantidade de sugestões de especialidades também é um indicador do nível de *expertise* de um usuário, de modo que este fator não pode ser ignorado na análise de especialidades dos indivíduos.

Desta forma, o problema tem duas condições que devem ser observadas: (1) A quantidade de sugestões sobre um mesmo domínio deve ser refletida na especialidade informada, quanto mais sugestões de um determinado domínio, mais evidente é a especialidade do usuário; (2) Esta quantidade deve amenizar os efeitos da popularidade dos usuários, isto pois, essa popularidade pode fazer com que a quantidade de sugestões de especialidades não reflita a especialidade real do indivíduo, isso pode prejudicar usuários menos populares que possuirão os valores de especialidade sugerida mais baixo.

Uma abordagem interessante é a empregada em jogos do tipo RPG para a solução de um problema semelhante, conceder experiência aos *players* para que eles possam subir seus níveis (*levels*) no jogo. Para que os *players* não distanciem muito rapidamente o *level* entre si, a experiência necessária para trocar de *level* cresce em uma escala, por exemplo, exponencial.

Neste caso, o nível de especialidade sugerida dos usuários deve crescer de forma cada vez mais lenta, assim a escala dos valores de  $S(u, d)$  pode ser alterada para uma escala logarítmica, de forma a evitar que os valores de especialidade sugerida dos usuários populares cresça demasiadamente rápido. Neste caso, a ( 9 ) passaria por uma alteração, e resultaria em ( 10 ).

$$S(u, d) = \log \left( 1 + \sum_{s \in S(u)} r(u_s) \right) \quad ( 10 )$$

Neste caso, a base 10 do logaritmo foi escolhida ao acaso, qualquer outra base produziria efeito semelhante ao desejado, no entanto é necessário um estudo mais profundo sobre qual seria a melhor base, dado que pode variar de acordo com o ambiente de implantação do modelo. Dentro do argumento da função  $\log$  é adicionada uma unidade ao somatório das relevâncias dos usuários, isto pois  $\log_k 1 = 0$ , a origem desejada para a escala de especialidades sugeridas.

A especialidade refutada do usuário é calculada da mesma maneira que a especialidade sugerida. As considerações e possibilidades feitas para o desenvolvimento da equação também são as mesmas. A única diferença entre as análises da especialidade sugerida e da especialidade refutada é quanto ao peso utilizado no processo de unificação dos tipos de especialidade. Neste caso, o peso da especialidade refutada deve ser um valor negativo, pois deve refletir uma diminuição da *expertise* de  $u$  em algum domínio  $d$ .

### 5.3 Especialidade Informada

Uma especialidade é informada quando o próprio indivíduo afirma possuir esta especialidade. Desta forma, o indivíduo  $A$  alega ter conhecimentos sobre o domínio  $d$ . Neste caso, como o próprio indivíduo informa que possui a especialidade, para que essa métrica reflita a *expertise* do usuário  $A$  é necessário considerar um peso para a opinião do próprio indivíduo. Assim, quando o indivíduo alega possuir especialidade no domínio  $d$ , significa, para o modelo, que ele possui *expertise* naquele domínio igual ao valor do peso da sua opinião. Em ( 11 ) é apresentado como é obtido o valor de especialidade informada dos indivíduos.

$$I(u, d) = \begin{cases} w_l, & \text{caso } d \in D_{u,l} \\ 0, & \text{caso contrario} \end{cases} \quad (11)$$

$u$  – usuário

$d$  – domínio

$w_l$  – peso para as habilidades informadas pelo usuário

$D_{u,l}$  – domínios informados ( $l$ ) pelo usuário  $u$  como suas áreas de especialidades

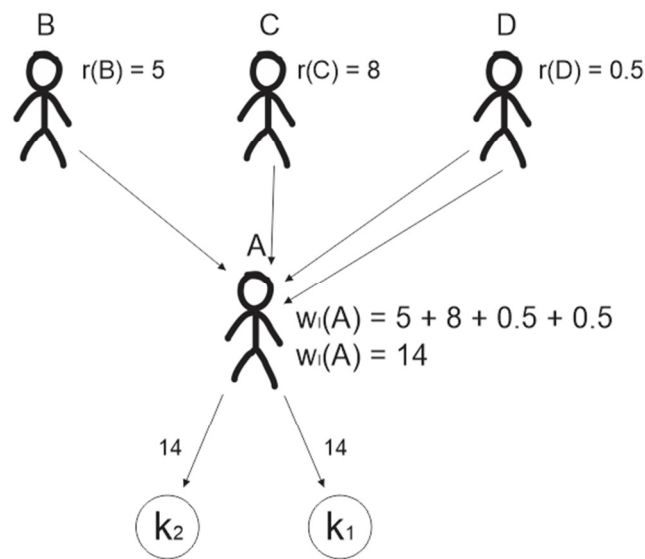
Em ( 11 ) é considerado um peso padrão ( $w_l$ ) que seria utilizado para dar valor às especialidades informadas de todos os usuários. Neste caso, o peso é fornecido integralmente para qualquer usuário  $u$  que informar possuir especialidade em um domínio  $d$ . Caso o usuário não informe que possui especialidade em algum domínio, o valor da especialidade informada é zero.

Ainda em ( 11 ), o peso padrão oferecido sempre que o usuário informa possuir uma especialidade pode motivá-lo a informar um grande número domínios, mesmo quando ele não possui tanta *expertise* no domínio em questão, até mesmo como uma forma de burlar a análise, aumentando o seu valor de  $I(u, d)$  e conseqüentemente o valor total de *expertise* no domínio  $d$ . Desta forma, os usuários que utilizassem mais a ferramenta de informar especialidades, seja ela qual for, possuiriam os valores de suas especialidades mais elevados.

Uma forma de resolver este problema é através do uso de uma abordagem que permita que cada vez que um usuário informar que possui uma especialidade, ele receba um valor adequado de  $I(u, d)$  para o domínio daquela especialidade. Uma forma de estimar o valor adequado a ser recebido por cada especialidade informada pelo próprio usuário é utilizar uma abordagem semelhante à utilizada no algoritmo PageRank (BRIN; PAGE, 2012). O algoritmo PageRank mapeia a web como um grafo de citações, no qual cada página é um nó e cada aresta uma citação (*link*) de um nó para outro. De acordo com a quantidade de nós que apontam para um nó,  $A$  por exemplo, e com o valor de PageRank de cada um destes nós, é obtido o valor de PageRank do nó  $A$ .

Para aplicar uma abordagem semelhante à do PageRank, o peso total que cada usuário pode atribuir a suas próprias especialidades deve depender de um fator externo ao próprio usuário, por exemplo as sugestões de especialidades que ele recebe. Estas sugestões,

conforme apresentado na seção 5.2, já consideram a relevância do usuário na rede. Neste caso, o valor total de que um usuário poderia receber de  $I(u, d)$  seria definido por estas sugestões, e conseqüentemente pela quantidade e relevância dos autores destas sugestões. Assim, de acordo com as sugestões de especialidades recebidas pelo usuário  $u$ , poder-se-ia obter um valor  $w_I(u)$ , que seria o valor total de especialidades que o próprio usuário poderia conceder a si próprio. Este valor seria dividido entre as especialidades que ele informar que possui *expertise*. A Figura 10 apresenta de forma visual o funcionamento deste processo.



**Figura 10 – Modelo visual para determinação do valor de  $I(u, d)$**

$w_I(u)$  – peso para as habilidades informadas pelo usuário

$k_i$  – termo que representa um domínio de especialidade

Na Figura 10, os usuários B, C e D realizarão sugestões de especialidades do usuário A. De acordo com a quantidade e a relevância dos usuários B, C e D foi determinado o peso total de  $I(u, d)$  que A poderia receber, denotado por  $w_I(u)$ . Este valor é distribuído igualmente entre os domínios ( $k_1$  e  $k_2$ ) que o usuário A informou possuir *expertise*. A EQ apresenta a formalização do exposto na Figura 10.

$$I(u, d) = \begin{cases} r(u), & \text{caso } d \in D_{u,I} \\ 0, & \text{caso contrario} \end{cases} \quad (12)$$

$u$  – usuário

$d$  – domínio

$r(u)$  – peso para as habilidades informadas pelo usuário

Como o peso é distribuído pela quantidade de especialidades, caso o usuário informe apenas um domínio ele poderia ficar com muitos pontos neste domínio. Porém, para que isso não ocorra, o peso total de especialidades informadas é ditado por outra variável que depende do usuário  $w_I(u)$ , porém não depende de suas interações, mas das interações de outros usuários do ambiente. A princípio as especialidades sugeridas serão utilizadas para ditar o peso de  $w_I(u)$ , isto pois a quantidade de especialidades sugeridas por outros usuários do ambiente social, bem como a relevância destes usuários, reflete a relevância do próprio usuário. Desta forma o algoritmo concede a estes usuários, considerados mais relevantes, mais crédito para informarem suas próprias especialidades.

#### 5.4 Especialidade Inferida

Uma especialidade é inferida quando não existem indícios de que um indivíduo a possua, mas existe uma forte probabilidade de que possua. Neste caso, acredita-se que determinada especialidade de um indivíduo pode ser inferida a partir de outras de suas especialidades. Por exemplo, existe uma probabilidade de que um indivíduo que possua *expertise* em projetos de bancos de dados e sistemas de bancos de dados distribuídos, possua também conhecimentos de sistemas gerenciadores de bancos de dados, como o MySQL. Mesmo na ausência de provas da existência deste conhecimento, ou até mesmo de indicações de que ele exista, esta probabilidade não deve ser desconsiderada. Uma forma de identificar este tipo de especialidades é através da análise do perfil de especialidades dos indivíduos, caso alguma área de especialidade do indivíduo seja identificada nesta análise, diz-se que a especialidade foi **Inferida**.

A análise das especialidades inferidas utiliza como base de informações as outras especialidades do usuário. Neste caso, o valor de especialidade obtido para cada domínio (representado por uma palavra-chave) é submetido a uma análise via ontologia de pesos leves. O valor obtido com uma especialidade,  $a$  por exemplo, é multiplicado pelos valores das arestas que saem do nó que representa a especialidade  $a$ . Estas arestas representam o quão relacionados são dois domínios, logo o valor dessa relação deve refletir o quanto o usuário conhece sobre aquela área relacionada, ou possui *expertise* nela. Para cada domínio  $b$  que possui relação com o domínio em questão  $a$ , a especialidade inferida de  $b$  é dada em ( 13 ).

$$I'(u,a,b) = E'(u,a) * R'(a,b) \quad ( 13 )$$

$u$  – Usuário analisado

$a, b$  – domínios representados por palavras-chave

$E'(u, a)$  – Especialidade do usuário  $u$  no domínio  $a$ , obtida através de algum dos tipos de especialidades anteriormente citados.

$R'(a, b)$  – valor normalizado do relacionamento entre o domínio  $a$  e o  $b$  na ontologia de pesos leves.

O peso normalizado do relacionamento entre os domínios  $a$  e  $b$  apresentado em ( 13 ) pertence ao intervalo  $[0,1]$ . Neste caso, como ele é multiplicado pelo valor de especialidades obtido em outra área relacionada, o valor resultante deve representar a *expertise* inferida do usuário nesta área relacionada. O processo de inferência de especialidades acontece com todos os termos nos quais o usuário analisado apresentou indícios de possuir *expertise*. Além disso, cada um destes termos é comparado com todos os seus termos relacionados na ontologia de pesos leves. A Figura 11 apresenta um exemplo do funcionamento deste processo.

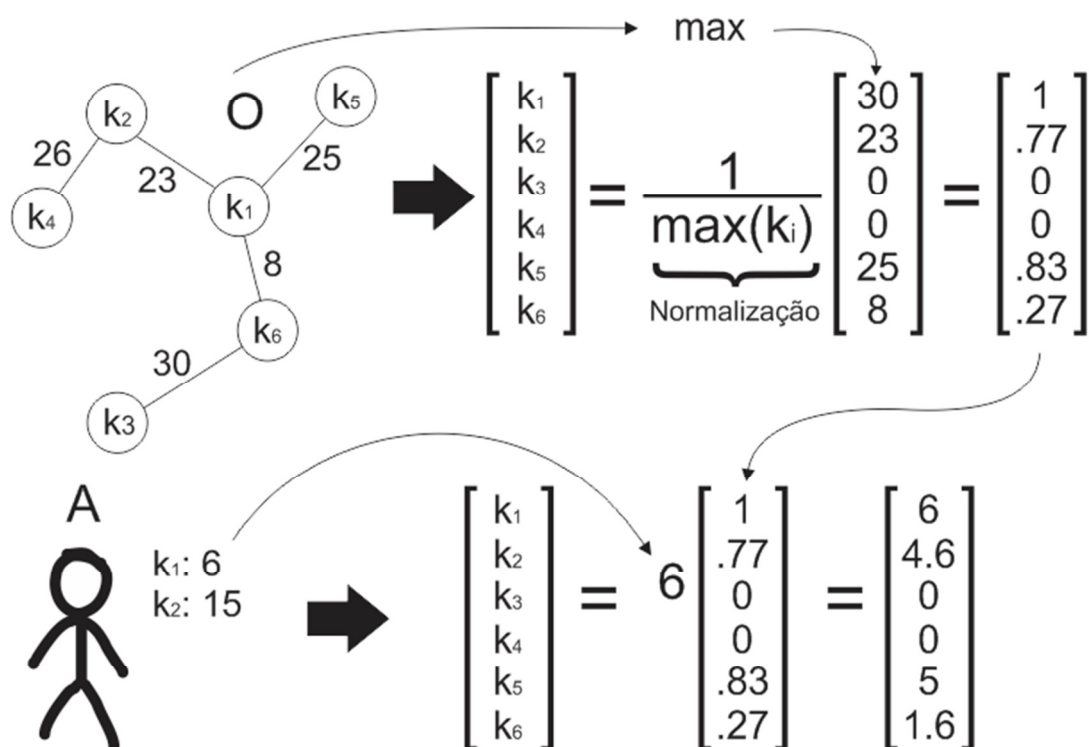


Figura 11 - Análise de especialidades inferidas a partir do termo  $k_1$

Na Figura 11 foi analisada a especialidade representada pela palavra-chave  $k_1$ . Neste caso, é criada uma matriz com os pesos dos relacionamentos entre as palavras-chave na



ontologia de pesos leves. O próprio termo  $k_l$  fica com o peso máximo de um relacionamento na ontologia. Em seguida, a matriz obtida a partir da ontologia é normalizada para que seja multiplicada pelo valor de especialidade do usuário no domínio  $k_l$ . O valor de especialidade para cada um dos termos relacionados a  $k_l$  é dado pelo seu respectivo valor na matriz resultante destas operações. A ( 14 ) resume o processo apresentado na Figura 11.

$$I'(u, d) = E'(u, d) * [R(d)]_{|D|,1} \quad ( 14 )$$

*u – Usuário analisado*

*d – Domínios representados por palavras-chave*

*E'(u, d) – Especialidade do usuário u no domínio d, obtida através de algum dos tipos de especialidades anteriormente citados.*

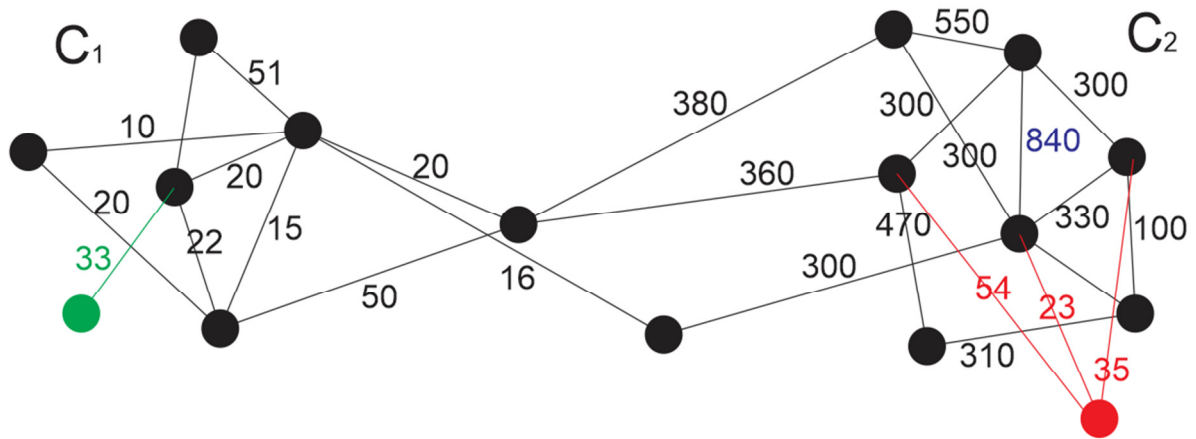
*D – Conjunto dos termos presentes na ontologia de pesos leves.*

*[R(d)]<sub>|D|,1</sub> – Matriz com os pesos normalizados de relacionamentos entre todos os termos da ontologia e o termo d.*

Como apresentado em ( 14 ), o valor de especialidade do usuário  $u$  no domínio  $d$  é multiplicado por uma matriz que contém o peso dos relacionamentos entre o domínio  $d$  e os demais domínios da ontologia de pesos leves. As dimensões dessa matriz são  $|D| \times 1$ , isto pois, é necessário um valor de relacionamento para cada termo contido na ontologia. Assim a matriz possui tantas linhas quantos termos existentes na ontologia, e apenas uma coluna. A multiplicação do valor de especialidade do usuário  $u$  no domínio  $d$  pelos valores normalizados dos relacionamentos entre  $d$  e os demais domínios deve refletir o quanto o usuário  $u$  possui *expertise* nestes domínios relacionados.

Como foi apresentado na Figura 11 os pesos dos relacionamentos entre os termos são normalizados com o maior peso existente na ontologia. Isto é feito para que termos que tenham baixos relacionamentos, mas estejam em uma região do grafo onde isso ocorre frequentemente, não sejam encarados como altamente relacionados. Porém, outras discussões podem questionar se este é mesmo um método válido, uma vez que uma ontologia de pesos leves é uma estrutura muito complexa que pode originar clusters de relacionamentos entre termos, principalmente termos relacionados a uma mesma área do conhecimento, e dentro de cada um destes clusters o comportamento dos relacionamentos pode ser diferente, ora pesos

muito altos, ora muito baixos. Esse tipo de comportamento junto a normalização dos termos utilizando o maior valor encontrado na ontologia pode prejudicar a análise. A Figura 12 apresenta exemplos destes dois casos mencionados.



**Figura 12 - Clusters na ontologia de pesos leves**

Na Figura 12 é possível notar que os pesos dos relacionamentos no cluster 1 são significativamente menores que os presentes no cluster 2. No entanto, todos os relacionamentos seriam normalizados com o maior peso presente na rede (840 em azul). Mesmo o termo com o relacionamento mais significativo no cluster 1 (33 em verde) seria transformado em um valor baixo. Uma alternativa para esse problema seria normalizar os termos a partir dos termos apenas na vizinhança do termo analisado. Porém, isto poderia resultar em um problema, pois termos que não estiverem bem relacionados poderiam ser beneficiados pela região onde estão presentes. Por exemplo, o termo em vermelho possui pesos baixos para o cluster 2, porém, caso sejam normalizados com os termos de sua vizinhança facilmente atingirão pesos altos, pois os termos da vizinhança também são baixos. Esses relacionamentos pouco significantes iriam competir com os relacionamentos mais relevantes do cluster (como o de 550), pois estes podem estar próximos a outros relacionamentos relevantes. Uma análise aprofundada sobre as características de cada um dos clusters poderia ser realizada para solucionar este tipo de problema, porém tal análise foge ao escopo deste trabalho e fica como um possível estudo futuro.

### 5.5 Integração das especialidades

Para encontrar o valor de *expertise* de um usuário  $u$  em um domínio  $d$  deve-se calcular uma média ponderada dos valores de cada um dos tipos de especialidades. De acordo com o

ambiente no qual se deseja avaliar a especialidade dos usuários, cada um dos tipos de especialidades possuirá diferente relevância no processo de análise. O peso de cada um destes tipos de especialidade é algo ainda não mensurado, de forma que ainda são necessários testes para determinar o melhor valor para estes pesos. Em ( 15 ) é apresentado a média ponderada dos tipos de especialidades.

$$E(u, d) = \frac{\alpha D(u, d) + \beta S(u, d) + \gamma R(u, d) + \omega I(u, d) + \sigma I'(u, d)}{\alpha + \beta + \gamma + \omega + \sigma} \quad ( 15 )$$

*D(u, d) – Especialidade demonstrada do usuário u no domínio d*

*S(u, d) – Especialidade sugerida do usuário u no domínio d*

*R(u, d) – Especialidade refutada do usuário u no domínio d*

*I(u, d) – Especialidade informada pelo usuário u no domínio d*

*I'(u, d) – Especialidade inferida do usuário u no domínio d*

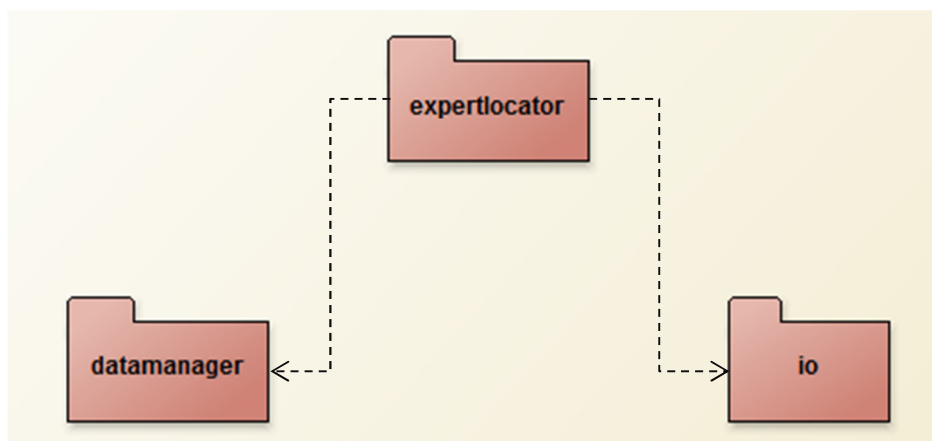
*$\alpha, \beta, \gamma, \omega, \sigma$  – Pesos das especialidades*

Em ( 15 ) é apresentada a média ponderada dos valores de especialidades do usuário *u* no domínio *d*. Neste caso, como não são conhecidos os pesos ideais para cada tipo de especialidade, para os testes iniciais serão considerados pesos unitários. Futuramente almeja-se um estudo aprofundado sobre formas de estabelecer estes de acordo com o ambiente de implantação do modelo.

## 6 API

O presente trabalho propõe um modelo que visa à identificação de especialidades de indivíduos baseados nos cinco tipos de especialidades apresentadas na seção 5. Cada tipo indica ora uma fonte diferente de informações (*e. g.* informações sobre habilidades informadas são informadas pelo próprio indivíduo, enquanto que informações de especialidades demonstradas necessitam de provas); ora formas diferentes de analisar as mesmas informações (*e. g.* uma análise diferenciada das provas das especialidades demonstradas pode originar informações sobre especialidades inferidas). A partir destes cinco tipos de especialidades foram definidas fontes de informações e heurísticas que às analisam com o propósito de identificar especialidades em indivíduos em ambientes sociais virtuais.

O conjunto de classes da API é dividido em três pacotes: (1) `datamanager`, que contém as classes que representam os objetos a serem analisados; (2) o pacote `io`, que realiza o acesso aos dados nos arquivos de entrada e os salva em arquivos de saída; e (3) o pacote `expertlocator`, que contém o arquivo executável e as classes que implementam a análise e o uso das equações apresentadas no modelo (seção 5). A Figura 13 apresenta os três pacotes.

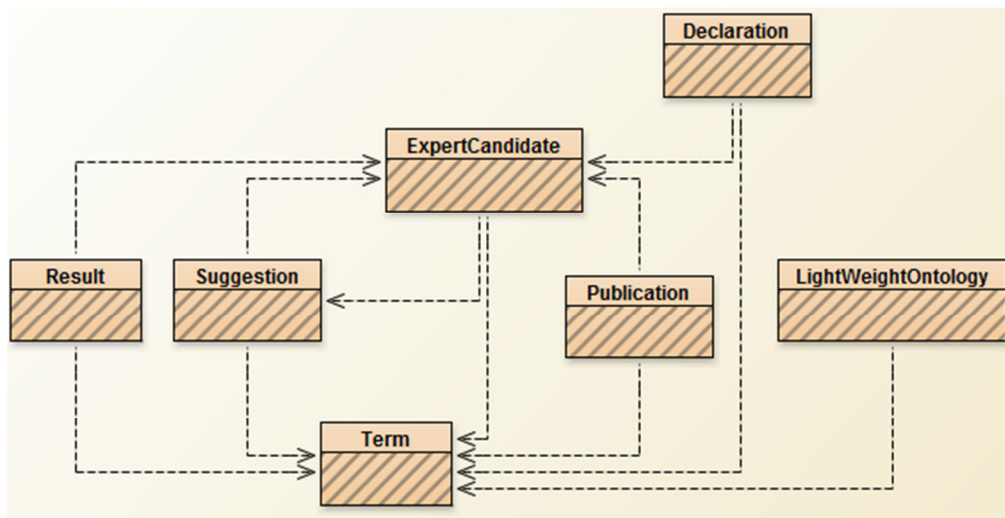


**Figura 13 - Pacotes da API**

A Figura 13 apresenta a dependência existente entre os pacotes que compõe a API. Neste caso, o pacote `expertlocator` acessa as classes contidas no pacote `datamanager` para realizar os processos de análise dos dados. Por sua vez, as classes do pacote `datamanager` utilizam as classes do pacote `io` para leitura dos dados. Ao final do processo de análise, as classes do pacote `expertlocator` utilizam as classes do pacote `io` para salvar os resultados obtidos em arquivos.

## 6.1 Manipulação dos Dados

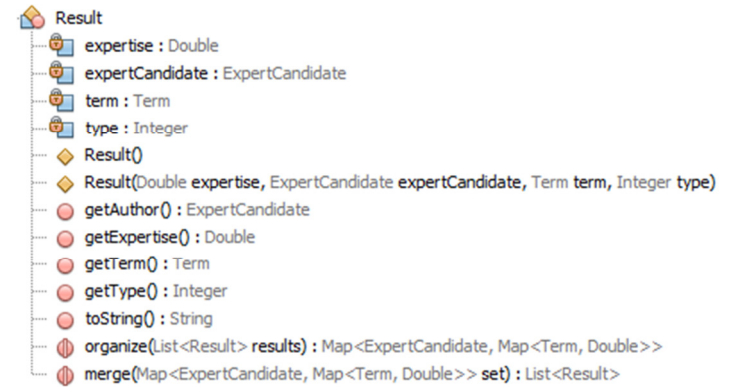
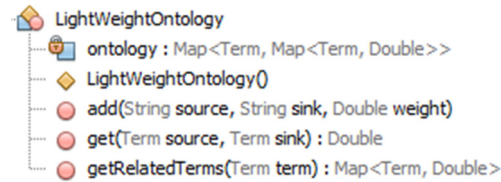
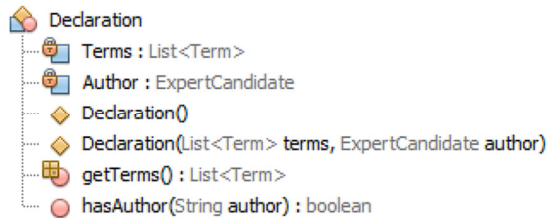
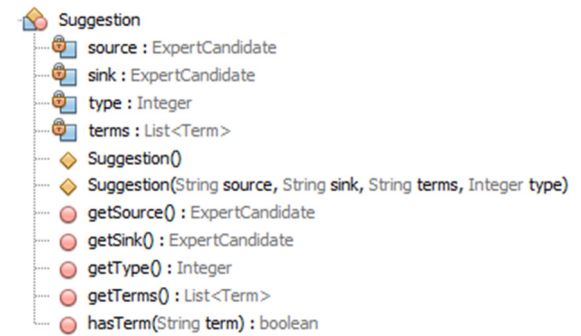
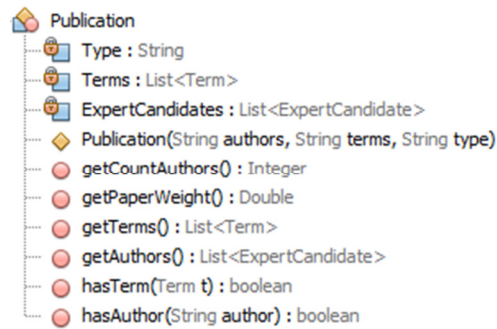
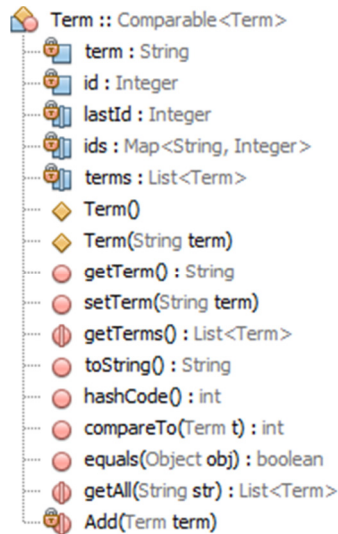
Para analisar estas informações, é necessário estabelecer os padrões sobre o qual elas serão submetidas para a análise. Neste sentido, foi elaborado um modelo de classes que apresenta as informações necessárias para analisar às especialidades dos indivíduos. A Figura 14 apresenta este modelo de classes.



**Figura 14 - Modelo de classes simplificado apresentando as classes que representam as informações durante a análise**

A Figura 14 apresenta o conjunto de informações necessárias para a análise de especialidades dos membros de um ambiente social virtual. A classe `Term` é responsável por representar as palavras-chave que representam os domínios de conhecimento dos especialistas. Essa classe é utilizada por todas as outras classes durante o processo de inferência de especialidades. Para a análise de Especialidades Demonstradas são consideradas publicações dos candidatos à especialista. A classe `Publication` representa as informações sobre essas publicações. Tais publicações podem ser de qualquer origem (*e. g.* Artigos Científicos, Posts em Blogs), basta que sejam disponibilizadas as palavras-chave. Para representar os usuários alvo da análise é utilizada a classe `ExpertCandidate`. Cada um deles possui uma lista de termos relacionados e uma lista de sugestões recebidas. Essas sugestões são utilizadas para calcular as especialidades Sugeridas e Refutadas e são representadas pela classe `Suggestion`. De forma semelhante, a classe `Declarations` é utilizada para representar as declarações de um candidato a especialista sobre especialidades que ele possui. Por fim, a classe `LightWeightOntology` é utilizada para representar a

ontologia de pesos leves para análise da Especialidade Inferida; e a classe `Result` representa um resultado da análise, contemplando o nome do especialista, sua *expertise* e a palavra-chave que representa o domínio daquela *expertise*. A seguir, a Figura 15 apresenta as classes, suas propriedades e seus métodos.



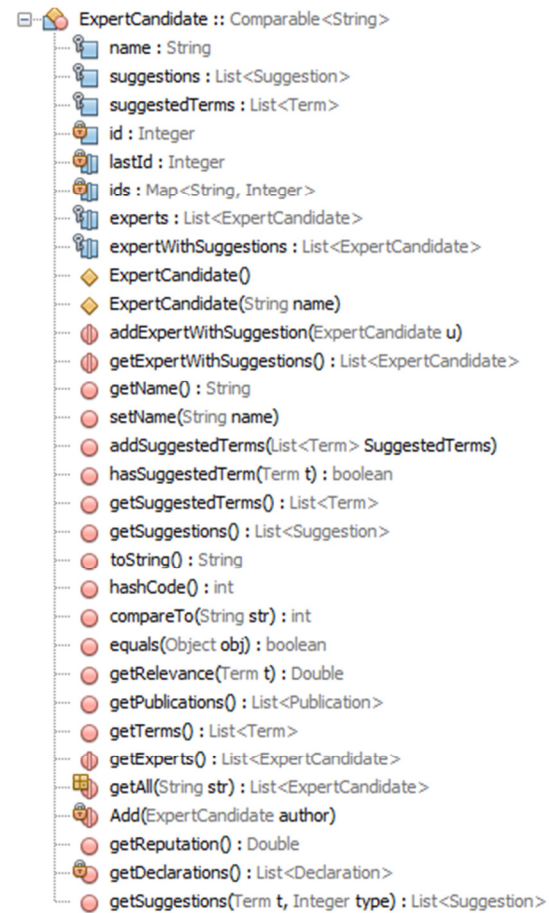


Figura 15 - Propriedades e métodos das classes de representação dos dados durante a análise de especialidades



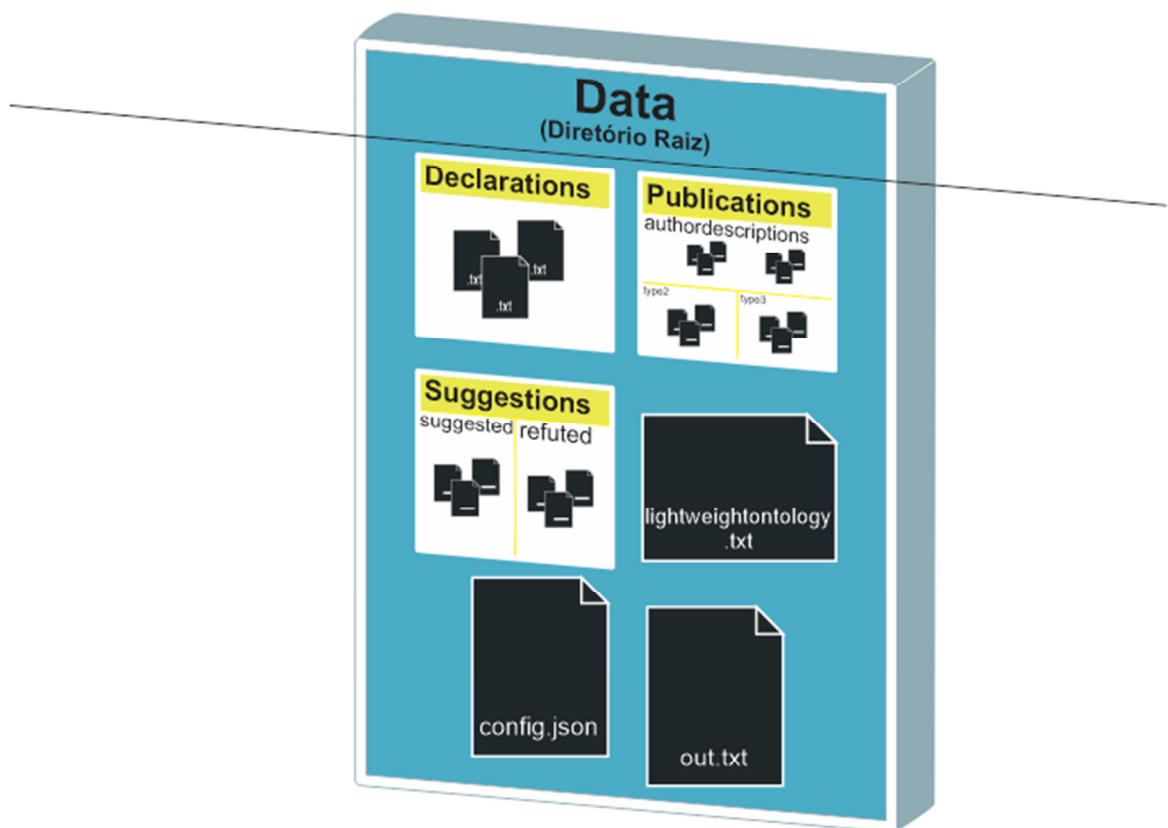
Como pode ser observado na Figura 15, as classes `Term` e `ExpertCandidate` implementam uma série de métodos que permitem a comparação de instâncias destas classes. Esta abordagem foi utilizada para possibilitar o uso das bibliotecas de listas e mapas já disponíveis no JAVA. Por exemplo, as funções `ExpertCandidate::hashCode()` e `Term::hashCode()`, que são pela classe `HashMap` do JAVA ao definir uma posição de um elemento em um mapa. Os códigos *hash* dessas funções são definidos a partir de identificadores únicos atribuídos às instâncias criadas de cada uma das classes. As funções `equals()` e `compareTo()` nessas classes tem funções semelhantes, porém atuam em outras classes, como a `ArrayList`. Ainda na classe `Term` existe uma lista estática de termos, essa lista comporta todos os termos que estão em uso durante a análise.

Na classe `Publication`, existem métodos que permitem acessar as principais propriedades das publicações, como por exemplo, o método `Publication::getPublicationWeight()`, que acessa o peso da publicação de acordo com o seu tipo. Já na classe `Suggestion` são definidos dois candidatos a especialistas, o primeiro `source` que criou uma sugestão de especialidade ao segundo `sink`. Essa sugestão pode conter um ou mais termos, listado na lista de termos `terms`. A classe `Declaration` contém apenas o candidato a especialista e a lista de termos informada por ele, estas informações são suficientes para representar as declarações do usuário sobre suas especialidades.

A classe `LightWeightOntology` é responsável por garantir o acesso às informações contidas na ontologia de pesos leves fornecida como entrada à análise de especialidades. Através desta classe é possível controlar um objeto que represente a ontologia com a possibilidade de adição e leitura de relacionamentos na ontologia (i. e. `add()` e `get()`). Além disso, o método `getRelatedTerms()` retorna um mapa dos termos relacionados a um termo específico e os pesos desses relacionamentos. Por fim, a classe `Result` deve carregar informações sobre as expertises dos candidatos à especialistas. Cada instância da classe, nesse sentido, possui uma propriedade `ExpertCandidate`, uma `Term` e uma `expertise (Integer)`. Ao fim da análise, cada uma das instâncias dessa classe que sobrem se tornarão linhas em um arquivo de saída. Em todas as classes foram implementados *getters* e *setters* das propriedades quando estes se fizeram necessárias. A documentação completa do código fonte foi criada em formato JAVADOC e disponibilizado junto ao código fonte.

## 6.2 Arquitetura dos Dados de Entrada e Saída

Por padrão, a API tenta fazer a leitura dos dados de entrada e fornecer as saídas conforme determinada configuração. Esta configuração poderá ser alterada por meio de um arquivo de configurações apresentado a seguir. A arquitetura padrão dos dados de entrada divide os arquivos de entrada em categorias diferentes, cujos arquivos estão armazenados dentro de diretórios diferentes dentro de um diretório base comum. A Figura 16 mostra esta organização inicial.



**Figura 16 - Organização inicial dos arquivos de entrada para leitura da API**

Como apresentado na Figura 16, o diretório `data` é o diretório base comum no qual estarão contidos os demais diretórios com os dados de entrada. No diretório `declarations`, devem ser armazenadas as auto-declarações de especialidades dos candidatos a especialistas. Estes dados serão utilizados para calcular o nível de especialidade informada do candidato a especialista. Já no diretório `publications` serão armazenadas as

publicações que podem comprovar alguma especialidade dos candidatos a especialista. Essas publicações são separadas por tipos de forma que cada um destes tipos de publicação possui um peso diferente, conforme descrito na seção 5.1. Cada tipo de publicação é indicado no sistema de arquivos por um subdiretório dentro de `publications`. No exemplo da Figura 16 apenas existe um tipo de publicação, indicado pelo diretório `authordescriptions`. O último subdiretório contido dentro da pasta `data` é o qual compreende as sugestões de usuários do ambiente social virtual quanto às especialidades de outros, o diretório `declarations`. Essas sugestões podem ser positivas, e serem utilizadas no cálculo de especialidades sugeridas (`suggested`); ou ainda podem ser negativas, e serem utilizadas para refutar as especialidades dos candidatos a especialistas (`refuted`).

Além dos subdiretórios, dentro da pasta `data`, na Figura 16 são exibidos alguns arquivos `*.txt`. Os arquivos alojados diretamente dentro da pasta `data` são: (1) `lightweightontology.txt`, que contém as informações dos relacionamentos entre os termos para criação de uma ontologia simples, utilizada para o cálculo de especialidade inferida; (2) `config.json`, onde estão localizadas as configurações de pesos e locais de arquivos a serem utilizados no processo de inferência de especialidades, arquivo cuja as nuances serão apresentadas a posteriori; e (3) `out.txt`, que conterá, após o processo de inferência de especialidade, as saídas com os níveis de especialidade dos candidatos analisados em seus respectivos domínios. Além destes arquivos, nos subdiretórios do diretório `data` estão localizados outros arquivos `*.txt`, cuja os nomes são irrelevantes ao funcionamento da API. Tais arquivos contêm os dados de entrada para o funcionamento do processo de inferência de especialidades.

Cada um dos arquivos dentro do diretório `declarations` representa uma declaração de um candidato a especialista sobre suas especialidades. Cada um desses arquivos pode conter um conjunto de palavras-chave, que representam diferentes domínios de especialidades do candidato, e o nome do candidato, utilizado para identifica-lo. A existência de mais de um arquivo com declarações do mesmo autor e mesmas palavras chave não interfere no processo de análise, elas serão somadas às demais especialidades durante a análise. A seguir na Listagem 2 é apresentado um exemplo do formato esperado para os arquivos do diretório `declarations`.

```

Parcilene Fernandes
Ontologias;Psicologia;Análise de Sentimentos

```

### **Listagem 2 - Exemplo de arquivo de dados de entrada no diretório `declarations`**

Na Listagem 2 é apresentado um arquivo composto por duas linhas. Na primeira linha deve ser informado o nome do candidato à especialista. Em seguida na segunda linha, são informadas as palavras-chave que representam os domínios de especialidades informados na declaração do candidato sobre suas especialidades. Estas palavras-chave devem estar separadas por “;” (ponto e virgula), sem espaçamentos. Demais linhas no arquivo serão desconsideradas.

Dentro de cada um dos subdiretórios de `publications` estão armazenados arquivos que representam as publicações que podem comprovar alguma especialidade do candidato a especialista. Esses arquivos devem possuir a mesma estrutura que os arquivos de declarações, isto é, compostos por uma linha com o nome do candidato e outra de palavras-chave. As mesmas regras de composição e utilização do arquivo de declarações se aplicam a estes arquivos de publicações.

Dentro do diretório `suggestions` estão contidos os arquivos que refletem as sugestões de usuários do ambiente social à candidatos a especialistas. Neste caso é necessária uma informação adicional no arquivo de entradas, deve ser adicionada informação sobre o autor da sugestão de especialidade. Com essa adição o formato dos arquivos de sugestões passa a ser o apresentado na Listagem 3.

```

Edeilson Milhomem da Silva
Jackson Gomes
Inteligência Artificial;Processamento de Imagens

```

### **Listagem 3 – Exemplo de arquivo de dados de entrada do diretório `suggestions`**

Na Listagem 3, o arquivo é composto por três linhas. A primeira linha informa o autor da sugestão; a segunda o candidato a especialista que recebeu a sugestão; e a terceira contém a lista de palavras-chave que representam os domínios de especialidade da sugestão. A lista de palavras-chave deve estar no mesmo formato das contidas nos outros arquivos nos subdiretórios de `data`, isto é, separadas por “;”. Como as sugestões podem ser positivas ou

negativas, elas podem ser armazenadas em um dos dois diretórios: `suggested` ou `refuted`, porém o formato dos arquivos em ambos os diretórios é o mesmo.

O arquivo `config.json` é onde serão definidas as configurações para o correto funcionamento da API. As configurações são divididas basicamente em dois tipos: (1) configurações de locais, que contém os locais nos quais ficarão armazenados os arquivos utilizados pela API; e (2) configurações de pesos, onde ficam informados os pesos utilizados durante o processo de inferência de especialidades. Para as configurações de locais de arquivos, é definido previamente um diretório comum no qual ficarão armazenados os demais arquivos (seria o diretório `data` no exemplo da Figura 16). Quanto aos pesos são referentes aos pesos de ponderação do resultado final da análise, descritos na seção 5.5 e os pesos dos tipos de publicações utilizadas pelo processo de análise de especialidades demonstradas (seção 5.1). A Listagem 4 apresenta um exemplo de arquivo de configuração.

```

1  {
2      locations: {
3          root: "C:\data\",
4          in:{
5              publications: "publications\",
6              declarations: "declarations\",
7              suggestions: "suggestions\",
8              lwontology: "lightweightontology.txt"
9          },
10         out: {
11             showed: "showedresults.txt",
12             declared: "declaredresults.txt",
13             suggested: "suggestedresults.txt",
14             refuted: "refutedresults.txt",
15             main: "out.txt"
16         }
17     },
18     weights: {
19         main: {
20             suggested: 1.0,
21             refuted: -1.0,
22             showed: 1.0,
23             declared: 1.0,
24             inferred: 1.0
25         },
26         publications: {
27             authordescriptions: 1.0
28         }
29     }
30 }

```

#### Listagem 4 - Exemplo de arquivo de configuração

O arquivo de configurações segue o padrão JSON para atribuição de seus valores. Como apresentado na Listagem 4, o arquivo de configuração é dividido em duas partes, uma para caminhos de arquivos e diretórios (`locations[2]`) e outra para pesos (`weights[18]`). Dentro da propriedade `locations` existem as propriedades: `root[3]`, que indica o diretório base no qual ficarão armazenadas os arquivos; `in[4]`, que indica os caminhos de arquivos de entrada; e `out[10]`, que indica os caminhos dos arquivos de saída. Na propriedade `out` não é necessário que sejam informados os valores das propriedades `showed[11]`, `declared[12]`, `suggested[13]` e `refuted[14]`, caso não se deseje utilizar as saídas parciais do algoritmo. Para utilizar a saída geral deve ser informada apenas a propriedade `main[15]`. As sub-propriedades de `weights` são: `main[19]`, que comporta os pesos utilizados para a ponderação dos tipos de especialidades para construção da saída

completa da API; e `publications` [26], que contém os pesos das publicações de acordo com seus tipos. Os tipos das publicações devem possuir o mesmo nome dos subdiretórios de `publications` na Figura 16.

O arquivo `lightweightontology.txt`, apresentado na Figura 16, é o arquivo que reflete os relacionamentos da ontologia de pesos leves utilizada para estender a análise de especialidades. Este único arquivo deve conter informações sobre todos os relacionamentos entre os termos da ontologia, bem como os pesos destes relacionamentos. O arquivo fica localizado no diretório raiz no qual estão armazenados os demais arquivos, a Listagem 5 apresenta um exemplo desse arquivo.

```
Redes Sociais;Ambientes Sociais Virtuais;5.0  
Recomendação de Especialistas;Ontologias;2.0
```

#### **Listagem 5 - Exemplo de arquivo de entrada da ontologia de pesos leves**

No arquivo `lightweightontology.txt`, cada linha representa um relacionamento entre dois termos na ontologia simples e informa também o peso desse relacionamento. Como apresentado na Listagem 5, a linha é composta por duas palavras-chave e um valor numérico, todos separados por “;” (ponto e virgula) sem espaçamentos. Este arquivo é lido durante o processo de análise de especialidades para a construção da ontologia simples que subsidia o cálculo de especialidades inferidas.

O último arquivo apresentado na Figura 16 é o `out.txt`, este é o arquivo no qual serão salvas as informações sobre as especialidades dos usuários e seus respectivos domínios. Cada linha do arquivo carrega informações sobre o especialista, o domínio e um valor de *expertise* daquele especialista naquele domínio. A Listagem 6 apresenta um exemplo deste arquivo construído a partir da execução da API sobre as entradas previamente apresentadas (um arquivo de declaração, um arquivo de sugestão positiva, a ontologia de pesos leves e as configurações).

```

parcilene fernandes;ontologias;1.0
parcilene fernandes;psicologia;1.0
parcilene fernandes;analise de sentimentos;1.0
parcilene fernandes;recomendacao de especialistas;0.4
jackson gomes;inteligencia artificial;1.0
jackson gomes;processamento de imagens;1.0

```

#### **Listagem 6 - Exemplo de arquivo de saída**

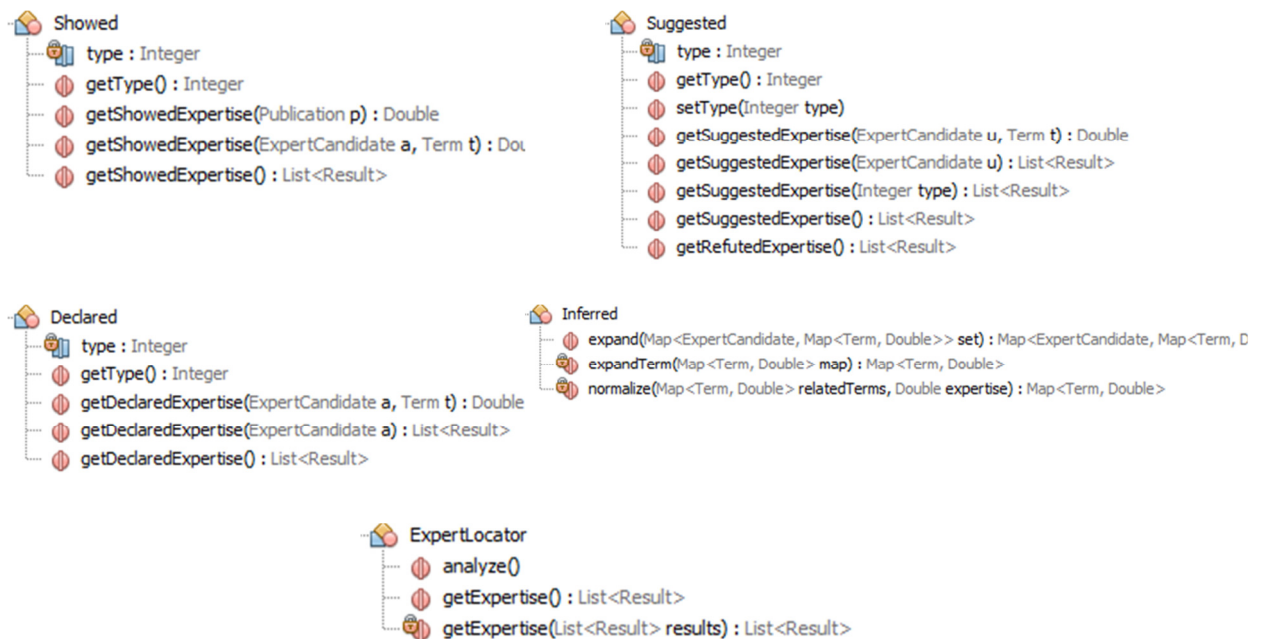
É possível notar na Listagem 6 que o formato é semelhante ao utilizado no arquivo da ontologia simples. No entanto, em lugar de duas palavras-chave existem o nome de um especialista e uma palavra-chave; por fim é mantido um valor numérico que reflete a *expertise* do especialista no domínio representado pela palavra-chave. Os arquivos de saída parciais, que podem ser utilizados caso necessário possuem este mesmo formato.

Todo o acesso aos arquivos de entrada e saída é realizado pelas classes do pacote IO. As classes servem para: (1) leitura específica de algum dos tipos de dados (`ReadDeclarations`, `ReadLightWeightOntology`, `ReadPublications` e `ReadSuggestions`), classes que mantém variáveis estáticas com todos os dados a serem utilizados em determinada parte da análise na memória; (2) uma classe para escrita dos arquivos de saída (`WriteResults`), parciais e total; (3) uma classe para realizar efetivamente o acesso aos arquivos (IO); e (4) uma classe para manipulação do arquivo de configuração (`Config`).

### **6.3 API de Inferência**

Para realizar o processo de cálculo de especialidades são utilizadas as classes do pacote `expertlocator`. Nesse pacote existe uma classe para cada um dos tipos de especialidades propostos no modelo (seção 5). A Figura 17 apresenta os métodos e propriedades dessas classes.





**Figura 17 - Métodos e propriedades das classes de cálculo de especialidades dos usuários**

Como apresentado na Figura 17, as classes que realizam o processo de inferência de especialidade são cinco: (1) *Showed*, que calcula as especialidades demonstradas; (2) *Suggested*, que mantém os processos de inferência de especialidades sugeridas e refutadas; (3) *Declared*, que controla o processo de análise das especialidades informadas; (4) *Inferred* que realiza o processo de expansão de especialidades dos especialistas por meio da inferência de novas especialidades a eles; e (5) *ExpertLocator*, que encapsula a chamada dos métodos das outras classes.

A classe *Showed* disponibiliza três métodos *getShownExpertise*, o primeiro deles sem parâmetros na assinatura é a chama mais geral do método, seu retorno é uma lista de resultados com as especialidades demonstradas de todo os candidatos a especialistas nos dados de entrada. O cálculo de especialidade é realizado em função de um candidato a especialista e um termo, neste sentido esses dois paramatros são utilizados para realizar várias chamadas ao segundo método *getShownExpertise(ExpertCandidate, Term)* na chamada genérica de *getShownExpertise()*. No método *getShownExpertise(ExpertCandidate, Term)* são realizadas chamadas para cada publicação do candidato a especialista que possui o termo especificado por parâmetro para o método *getShownExpertise(Publication)*. Esse método que utiliza como parâmetro as publicações é a chamada mais específica do método *getShownExpertise*, e é responsável por retorna o valor de expertise que deve ser atribuído ao candidato pela

produção daquela publicação conforme definido na seção 5.1  $\left(\frac{pw(p)}{|a(p)|}\right)$ . Esse valor é somado para todas as publicações do candidato que possuem o termo especificado, resultando na especialidade demonstrada do autor no domínio representado pelo termo. Esse valor, por sua vez é retornado para a chama mais geral do método `getShowedExpertise` para ser transformado em uma instância de resultado.

No processo de análise de especialidades que acontece na classe `Suggested` a relação entre os métodos é semelhante à descrita para a classe `Showed`. O processo de análise também é dividido em três etapas por ordem de especificidade. Porém, a classe `Suggested` realiza a análise tanto da especialidade sugerida quanto da refutada, o que torna necessário a existência de dois métodos que encapsulam a chamada do método mais genérico, apenas alternando um parâmetro que indica qual das duas análises está a ser realizada (`getSuggesteExpertise()` e `getRefutedExpetrtrtise()`). A partir desse ponto a relação entre os métodos ocorre de forma semelhante, o método mais genérico (`getSuggesteExpertise(Integer)`) deve retornar uma lista com os resultados das especialidades sugeridas ou refutadas de todos os candidatos. As especialidades sugeridas e refutadas são analisadas para cada usuário por meio da chamada do método `getSuggesteExpertise(ExpertCandidate)`. Para cada um desses candidatos serão analisadas as sugestões recebidas para cada um dos termos que representam os domínios de especialidades. As sugestões de especialidades são feitas por usuários do ambiente social virtual, e seus pesos devem refletir a reputação desses usuários, conforme discutido na seção 5.2. Desta forma, a função `getSuggesteExpertise(ExpertCandidate, Term)` irá retornar o logaritmo da soma das reputações dos usuários que fizeram sugestões de especialidades representadas pelo termo passado como parâmetro ao candidato, de acordo com  $(10) \left(\log\left(1 + \sum_{s \in S(u)} r(u_s)\right)\right)$ .

De modo semelhante às duas primeiras classes, a classe `Declared` realiza a análise das especialidades informadas do usuário a partir de um método mais genérico e por meio de outros dois mais específicos. O primeiro método, `getDeclaredExpertise()` é a chamada mais genérica, e retorna uma lista de resultados com as especialidades dos especialistas e seus respectivos domínios. A tarefa de analisar as especialidades informadas dos usuários é realizada para cada candidato a especialista no método `getDeclaredExpertise(ExpertCandidate)`. Esse método percorrerá uma lista

com todos os termos sugeridos ao candidato, realizando chamadas ao método mais específico (`getDeclaredExpertise(ExpertCandidate, Term)`) que acessa a reputação do candidato. A partir destas informações é possível construir a lista de especialistas e suas especialidades informadas. O valor de especialidades informadas reflete a reputação do especialista que a informou, como definido na **Erro! Fonte de referência não encontrada.** ( $r(u)$ ).

A classe `Inferred` realiza a inferência de especialidades que não estão explicitamente declaradas e relacionadas aos candidatos a especialistas, mas provavelmente existem. Através do uso da ontologia simples é possível descobrir quais domínios estão fortemente relacionados e inferir uma possível especialidade de um candidato a partir dessa relação. Essa classe possui algumas peculiaridades em relação às demais descritas nesta seção. Esta classe deve operar sobre um conjunto de respostas previamente definido, logo apenas pode ser utilizada em combinação com algum dos outros métodos de análise de especialidades. A classe `Inferred` realiza primeiramente uma expansão, com os dois métodos `expand`, dos termos já descobertos do candidato a fim de que sejam descobertas novos domínios de especialidades para ele. Após a expansão, o peso dos relacionamentos entre as palavras-chaves que representam esses domínios são normalizados e utilizados para inferir o grau de especialidade do especialista em cada domínio inferido. O processo de normalização e inferência de especialidades ocorre de acordo com o descrito na seção 5.4.

Por fim, a classe `ExpertLocator` encapsula a chamada de todos os outros métodos de forma a facilitar sua utilização por completo. Uma simples chamada ao método `analyze` irá analisar todos os dados contidos nos diretórios de entrada e gerar um arquivo de saída com os níveis de especialidades dos candidatos. O método `analyze` carrega as configurações e chama o método `getExpertise()` que retorna a lista definitiva de resultados a serem salvos no arquivo de saída. Esse método por sua vez faz chamada aos métodos previamente descritos para montar a lista definitiva de resultados da análise. Por fim é chamado o método `getExpertise(List<Result>)` para realizar a soma ponderada de todas as especialidades dos candidatos.

#### 6.4 Exemplo Prático

A presente seção objetiva apresentar um exemplo numérico do processo de análise de especialidades. Para tanto será considerada uma quantidade extremamente pequena de informações de entrada se comparado com as quantidades normalmente contidas em ambientes sociais virtuais. Esta decisão de utilizar um pequeno conjunto de dados de entrada visa a simplificação dos cálculos realizados de forma a facilitar sua compreensão e acompanhamento. A Tabela 2 apresenta as informações utilizadas como entradas classificadas por tipos.

TIPO DO ITEM	CONTEÚDO DO ARQUIVO DO ITEM	ID
PUBLICAÇÕES	Edeilson Milhomem Análise de Confiança; Sistemas Colaborativos	$P_1$
	Edeilson Milhomem; Diego Oliveira Recomendação de Especialistas	$P_2$
DECLARAÇÕES	Parcilene Fernandes Ontologia; Psicologia; Análise de Sentimentos	$D_1$
	Edeilson Milhomem Recomendação de Especialistas; Redes Sociais	$D_2$
SUGESTÕES	Fábio Ricardo Edeilson Milhomem Psicologia	$S_1$
	Diego Oliveira Jackson Gomes Inteligência Artificial; Redes Sociais	$S_2$
REFUTAÇÕES	Parcilene Fernandes Edeilson Milhomem Psicologia	$S_3$

Tabela 2 - Dados a serem utilizados no exemplo numérico de funcionamento da API

A partir desses dados e das informações contidas no arquivo de configuração é possível começar o processo de análise. As informações de configuração utilizadas serão as mesmas apresentadas na Listagem 4, é importante destacar que todos os pesos possuem módulos unitários, mais informações de entrada serão apresentadas assim que se fizerem

necessárias ao exemplo. No caso da Tabela 2, as primeiras informações apresentadas são as de publicações. Cada uma das duas publicações será analisada para cada autor e cada uma de suas palavras-chave. Essas informações devem ser avaliadas de acordo com ( 8 ), conforme segue para a palavra-chave Recomendação de Especialistas na publicação  $P_2$  de Edeilson Milhomem:

$$D(u, d) = ar(u, d) \sum_{\forall p \in P(u, d)} \frac{pw(p)}{|a(p)|}$$

$$D(u, d) = ar(u, d) * \left( \frac{1}{1} + \frac{1}{2} \right) \quad (16)$$

A relevância do autor ( $ar(u, d)$ ) deve ser calculada em função dos termos relacionados a ele junto à ontologia de pesos leves. Para o exemplo serão considerados os seguintes conjuntos de termos relacionados aos candidatos a especialistas apresentados na Tabela 3.

<b>Candidato a Especialista</b>	<b>Palavras-chave</b>	<b>Peso</b>
Edeilson Milhomem	Recuperação da Informação	20
	Recomendação de Especialistas	10
	Redes Sociais	4
Diego Oliveira	Recomendação de Especialistas	3
	Análise de Confiança	10

**Tabela 3 - Palavras-chave exemplo relacionadas a candidatos a especialistas**

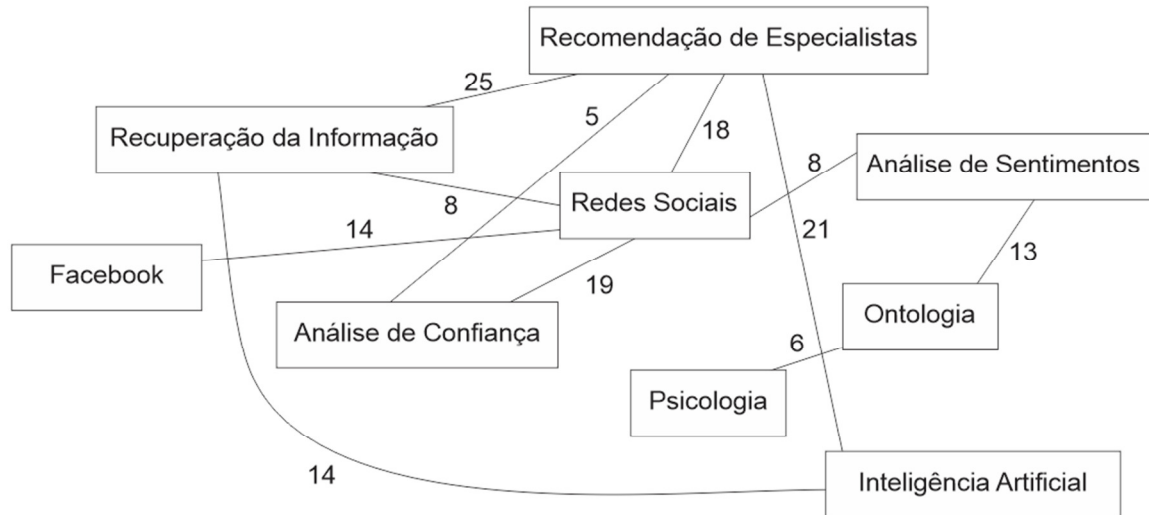
Além disso, a ontologia de entrada é dada pelo arquivo apresentado na Listagem 7.

```
Recuperação da Informação;Recomendação de Especialistas;25
Recomendação de Especialistas;Redes Sociais;18
Recuperação da Informação;Redes Sociais;8
Análise de Confiança;Redes Sociais;19
Análise de Confiança;Recomendação de Especialistas;5
Psicologia;Ontologia;6
Análise de Sentimentos;Ontologia;13
```

Redes Sociais;Facebook;14  
 Recomendação de Especialistas;Inteligência Artificial;21  
 Inteligência Artificial;Recuperação da Informação;14  
 Análise de Sentimentos;Redes Sociais;8

**Listagem 7 - Arquivo que contém os dados de construção da ontologia simples**

A estrutura representada na Listagem 7 também pode ser representada pela Figura 18.



**Figura 18 - Exemplo de ontologia simples**

Utilizando o algoritmo de inferência de relevância de candidatos a especialistas em domínios descrito na seção 5.1 é possível descobrir a relevância de Edeilson Milhomem na área de Recomendação de Especialistas dada a ontologia da Figura 18, conforme segue:

$$\vec{d} = 26\vec{r}_e + 25\vec{r}_i + 5\vec{a}_c + 18\vec{r}_s + 8\vec{i}_a$$

$$\vec{u} = 20\vec{r}_i + 10\vec{r}_e + 4\vec{r}_s \quad (17)$$

$$ar(u, d) = \cos \theta \cong 0,88$$

Substituindo o valor de ( 16 ), em ( 17 ), vem:

$$D(u, d) = 0,88 * \left( \frac{1}{1} + \frac{1}{2} \right) = 1,33$$

Os vetores  $\vec{d}$  e  $\vec{u}$  são respectivamente os vetores que representam os relacionamentos extraídos da ontologia simples e os termos relacionados ao Edeilson Milhomem. Dessa forma,  $\cos \theta$  é o cosseno do ângulo entre esses vetores. Além disso, os versores  $\vec{r}_e$ ,  $\vec{r}_i$ ,  $\vec{a}_c$ ,  $\vec{i}_a$  e  $\vec{r}_s$  representam os termos da ontologia Recomendação de Especialistas, Recuperação da

Informação, Análise de Confiança, Inteligência Artificial e Redes Sociais, respectivamente. Os mesmos passos podem ser repetidos para obtenção dos outros valores de especialidades demonstradas. Através da execução destes passos poder-se-ia obter o arquivo de saída parcial de especialidades demonstradas como o apresentado na Listagem 8.

```
edeilson milhomem;analise de confianca;0.28
edeilson milhomem;sistemas colaborativos;1
edeilson milhomem;recomendacao de especialistas;1.33
diego oliveira;recomendacao de especialistas;0.15
```

**Listagem 8 - Exemplo de arquivo de saída parcial de especialidades demonstradas**

O próximo tipo de especialidade a ser analisada é a especialidade informada. A especialidade informada deve refletir a reputação dos candidatos que as informaram. Neste caso, se fazem necessárias informações sobre a reputação dos usuários do ambiente social. Para este exemplo serão utilizados os valores de reputação apresentados na TABELA.

Usuário do Ambiente	Reputação
Parcilene Fernandes	0.8
Edeilson Milhomem	0.82
Fábio Ricardo	0.78
Diego Oliveira	0.6

**Tabela 4 - Exemplos de valores de reputação**

Com base nestas informações é possível calcular o nível de especialidades informadas, que deve ser igual à reputação dos candidatos à especialistas, de acordo com o exposto na seção 5.3. Nestas condições, o arquivo de saída parcial das especialidades informadas dos indivíduos teria o conteúdo apresentado na Listagem 9.

```
parcilene fernandes;ontologia;0.8
parcilene fernandes;psicologia;0.8
parcilene fernandes;analise de sentimentos;0.8
edeilson milhomem;redes sociais;0.82
edeilson milhomem;recomendação de especialistas;0.82
```

**Listagem 9 - Exemplo de arquivo de saída de análise de especialidades informadas**

Neste ponto, serão analisadas as especialidades sugeridas e refutadas dos usuários do ambiente social. Conforme discutido na seção 5.2, as especialidades sugeridas devem refletir a reputação dos usuários que realizaram as sugestões. Além disso, a popularidade dos usuários não pode interferir de maneira significativa no processo de análise. Dadas estas considerações, as especialidades sugeridas dos usuários devem ser calculadas de acordo com ( 10 ), conforme segue o exemplo utilizando o domínio psicologia para o candidato Edeilson Milhomem.

$$S(u, d) = \log \left( 1 + \sum_{s \in S(u)} r(u_s) \right) = \log(1 + 0,78) = 0,2 \quad (18)$$

O valor de reputação dos usuários do ambiente social virtual que fizeram sugestões de especialidades a Edeilson Milhomem são somados em ( 18 ). Neste caso, existe apenas um usuário, com reputação de 0,78. A repetição destes passos para os demais termos sugeridos e candidatos à especialistas resultaria no conteúdo de arquivo de saída parcial de especialidades sugeridas apresentado na Listagem 10.

```
edeilson milhomem;psicologia;0.25
jackson gomes;redes sociais;0.20
jackson gomes;inteligencia artificial;0.20
```

**Listagem 10 - Exemplo de arquivo de saída da análise de especialidades sugeridas**

O mesmo processo resultaria no arquivo de saída parcial da análise de especialidades refutadas, porém com o peso negativo característico deste tipo de análise, conforme discutido na seção 5.2. O conteúdo desse arquivo é apresentado na Listagem 11.

```
edeilson milhomem;psicologia;-0.26
```

**Listagem 11 - Exemplo de arquivo de saída da análise de especialidades refutadas**

Após a análise de todos os termos, o conjunto de termos obtido e seus pesos é utilizado no processo de inferência de novos possíveis domínios de especialidades dos candidatos à especialistas. Os valores das expertises já obtidas até este ponto são somados, já considerando seus pesos, para que então este conjunto seja expandido por meio da ontologia simples. A média ponderada de cada uma das, de acordo com ( 15 ), especialidades resultaria na lista de resultados apresentada na Listagem 12.

```
edeilson milhomem;recomendacao de especialistas;0.72
```



```

edeilson milhomem;sistemas colaborativos;0.33
edeilson milhomem;redes sociais;0.27
edeilson milhomem;psicologia;-0.00
diego oliveira;recomendacao de especialistas;0.05
parcilene fernandes;ontologia;0.27
parcilene fernandes;psicologia;0.27
parcilene fernandes;analise de sentimentos;0.27
jackson gomes;redes sociais;0.07
jackson gomes;inteligencia artificial;0.07

```

**Listagem 12 – Soma das expertises obtidas até o momento**

Uma vez de posse da lista com a soma dos valores de expertise, o próximo passo é utilizar ( 13 ) para expansão dos conjuntos, conforme segue em ( 19 ). Neste exemplo foi expandido o termo Inteligência Artificial do especialista Jackson Gomes.

$$I'(u, d) = E'(u, d) * [R(d)]_{|D|,1} \quad (19)$$

$$I'(u, d) = 0,07 * \left(\frac{1}{25}\right) * \begin{bmatrix} 25 \\ 21 \\ 14 \end{bmatrix} [\vec{i}a \quad \vec{r}e \quad \vec{r}i] = (0,07\vec{i}a + 0,06\vec{r}e + 0,04\vec{r}i)$$

No exemplo apresentado em ( 19 ) foi introduzida a notação vetorial para tornar mais claro o procedimento empregado. Os vetores são os mesmos utilizados na ( 17 ). Neste caso, o conjunto expandido de especialidades baseadas no especialista Jackson Gomes, para o termo Inteligência Artificial é apresentado na Tabela 5.

Domínio	Expertise
Inteligência Artificial	0,07
Recuperação da Informação	0,06
Recomendação de Especialistas	0,04

**Tabela 5 - Exemplo de conjunto expandido de especialidades**

A repetição destes procedimentos para obtenção de especialidades inferidas resultaria no conteúdo de arquivo parcial de saída da análise de especialidades inferidas apresentado na Listagem 13.

edeilson milhome;analise de confianca;0.09  
edeilson milhome;recomendacao de especialistas;0.02  
  
edeilson milhome;redes sociais;0.07  
  
edeilson milhome;sistemas colaborativos;0.33  
  
edeilson milhomem;recomendacao de especialistas;0.72  
edeilson milhomem;recuperacao da informacao;0.72  
edeilson milhomem;analise de confianca;0.14  
edeilson milhomem;inteligencia artificial;0.60  
  
edeilson milhomem;redes sociais;0.27  
edeilson milhomem;recuperacao da informacao;0.09  
edeilson milhomem;recomendacao de especialistas;0.20  
edeilson milhomem;analise de sentimentos;0.09  
edeilson milhomem;analise de confianca;0.21  
edeilson milhomem;facebook;0.15  
  
edeilson milhomem;psicologia;0.00  
edeilson milhomem;ontologia;0.00  
  
diego oliveira;recomendacao de especialistas;0.05  
diego oliveira;recuperacao da informacao;0.05  
diego oliveira;analise de confianca;0.01  
diego oliveira;redes sociais;0.04  
diego oliveira;inteligência artificial;0.04  
  
parcilene fernandes;ontologia;0.27  
parcilene fernandes;psicologia;0.06  
parcilene fernandes;analise de sentimentos;0.14

```

parcilene fernandes;psicologia;0.27
parcilene fernandes;ontologia;0.06

parcilene fernandes;analise de sentimentos;0.27
parcilene fernandes;redes sociais;0.09
parcilene fernandes;ontologia;0.14

jackson gomes;redes sociais;0.07
jackson gomes;analise de confiaca;0.05
jackson gomes;facebook;0.04
jackson gomes;redes informacao;0.02
jackson gomes;recomendacao de especialistas;0.05
jackson gomes;analise de sentimentos;0.02

jackson gomes;inteligencia artificial;0.07
jackson gomes;recomendacao de especialistas;0.05
jackson gomes;recuperacao da informacao;0.04

```

**Listagem 13 – Exemplo de arquivo de saída parcial da análise de especialidades inferidas**

No exemplo da Listagem 13, o arquivo foi dividido em blocos nos quais o primeiro item foi expandido para obtenção dos demais. No arquivo real tais divisões não existem. Após a obtenção das especialidades inferidas é possível obter a saída final do processo de análise de especialidades através da obtenção da média ponderada dos resultados obtidos em todos os processos de análise, de acordo com ( 15 ). Em ( 20 ) é apresentado o calculo para obtenção da expertise de Edeilson Milhomem no domínio Recomendação de Especialistas.

$$E(u, d) = \frac{\alpha D(u, d) + \beta S(u, d) + \gamma R(u, d) + \omega I(u, d) + \sigma I'(u, d)}{\alpha + \beta + \gamma + \omega + \sigma}$$

$$E(u, d) = \frac{\alpha 1,33 + \beta 0 + \gamma 0 + \omega 0,82 + \sigma 0,93}{\alpha + \beta + \gamma + \omega + \sigma}$$

( 20 )

*Como supracitado, os pesos tem módulos unitários, além disso, o peso  $\gamma$  é negativo, segue que:*

$$E(u, d) = \frac{1,33 + 0,82 + 0,93}{3} = 1,03$$

Por meio dos passos utilizados para obter a expertise de Edeilson Milhomem no domínio Recomendação de Especialistas é possível obter todos os resultados de expertise dos outros especialistas. O conteúdo do arquivo de saída final da análise, com esses valores, é apresentado na Listagem 14.

```
edeilson milhome;analise de confianca;0.24
edeilson milhome;recomendacao de especialistas;1.03
edeilson milhome;redes sociais;0.39
edeilson milhome;sistemas colaborativos;0.44
edeilson milhomem;recuperacao da informacao;0.27
edeilson milhomem;inteligencia artificial;0.2
edeilson milhomem;analise de sentimentos;0.03
edeilson milhomem;facebook;0.05
edeilson milhomem;psicologia;0.00
edeilson milhomem;ontologia;0.00
diego oliveira;recomendacao de especialistas;0.07
diego oliveira;recuperacao da informacao;0.02
diego oliveira;analise de confianca;0.00
diego oliveira;redes sociais;0.01
diego oliveira;inteligência artificial;0.01
parcilene fernandes;ontologia;0.42
parcilene fernandes;psicologia;0.38
parcilene fernandes;analise de sentimentos;0.40
parcilene fernandes;redes sociais;0.03
jackson gomes;redes sociais;0.09
jackson gomes;analise de confiaca;0.02
jackson gomes;facebook;0.01
jackson gomes;redes informacao;0.02
jackson gomes;recomendacao de especialistas;0.03
jackson gomes;analise de sentimentos;0.01
jackson gomes;inteligencia artificial;0.09
```

**Listagem 14 – Exemplo de arquivo de saída final da análise de especialistas**

A Listagem 14 apresenta os valores finais de expertise dos especialistas em seus respectivos domínios. Tais valores refletem a proficiência do indivíduo em determinado domínio. Os valores são fornecidos pela API em um arquivo \*.txt com o formato apresentado na Listagem 14.

## 7 CONSIDERAÇÕES FINAIS

Os avanços dos últimos anos nos sistemas de informação têm motivado organizações a enfatizar o processo de armazenar o conhecimento gerado com seus trabalhos (Yimam-seid & Kobsa, 2002). No entanto, nem todo conhecimento organizacional pode ser documentado e salvo nestes sistemas de informação, permanecendo armazenado apenas na forma de conhecimentos tácitos. Desta forma SREs tornam-se ferramentas interessantes para essas organizações, pois garantem acesso a esse conhecimento tácito dos colaboradores das organizações. Através de tais ferramentas, os SREs, pode-se localizar indivíduos com conhecimentos específicos em determinado domínio (*i. e.* especialistas). Estes especialistas, por sua vez, podem compartilhar o conhecimento que possuem com outros membros da organização.

Com o objetivo de facilitar o acesso ao conhecimento, principalmente em ambientes sociais virtuais, este trabalho apresentou um modelo para identificação de especialistas que explora informações comumente disponíveis em ambientes virtuais. Informações que podem ser obtidas de perfis de usuários de ambientes sociais virtuais, como declarações sobre suas especialidades; informações extraídas de interações sociais entre os membros desses ambientes, como as sugestões de especialidades e outras informações que podem surgir nos ASVs, ou em outros ambientes virtuais, como as publicações dos usuários que demonstram suas especialidades. Essas informações são analisadas e ponderadas para obtenção de um valor de *expertise* dos usuários de ASVs.

O modelo proposto realiza análises quantitativas e qualitativas das informações supracitadas de forma que elas possam refletir o nível de proficiência de determinado indivíduo, membro do ASV, em determinado domínio. Além dessas análises quantitativas e qualitativas, foi proposto um mecanismo de inferência de especialidades, que utiliza conceitos de ontologias simples para expandir o conjunto de domínios de especialidades de um especialista com outros domínios aos quais provavelmente ele possui alguma proficiência. O trabalho propôs que informações sobre a correlação entre os termos, contidas em uma ontologia simples, poderia direcionar a descoberta de novos domínios de especialidades para os especialistas. Algumas considerações foram feitas sobre formas de melhorar a análise à ontologia simples utilizando a exploração de informações de clusters entre os nós. Essa discussão será abordada com mais precisão em trabalhos futuros.

O modelo proposto ainda carece de validação em um ambiente real, pois todos os testes executados foram realizados em ambientes fictícios. Futuramente almeja-se a utilização do modelo para inferir os níveis de especialidades de autores de um portal<sup>4</sup> de notícias relacionados à temas de psicologia. Uma vez de posse dos resultados da análise de especialidades dos autores desse portal será feito uma análise da qualidade dos resultados através de questionários respondidos pelos editores e revisores do portal, que conhecem as especialidades dos autores. Através dessa metodologia espera-se comprovar a eficácia do modelo proposto e chegar a conclusões mais precisas sobre o comportamento dos pesos utilizados para ponderar as especialidades dos usuários.

Outra proposta a ser analisada futuramente é a aplicação do modelo em uma rede social acadêmica com o objetivo de recomendar relacionamentos entre acadêmicos e pesquisadores com áreas de pesquisa semelhantes, ou possibilitar a pesquisa por especialistas de domínios específicos. Ao implantar o modelo em tal ambiente almeja-se analisar qual seu impacto para a promoção da colaboração entre indivíduos dentro do ambiente. Esta análise seria realizada na rede social Conecta<sup>5</sup> que atualmente está em fase de desenvolvimento no CEULP/ULBRA, no entanto a rede foi desativada nos últimos meses, para fins de continuação de seu desenvolvimento, e não foi possível colher dados da mesma. Espera-se que o modelo atue como estimulante à colaboração no ambiente de forma que os usuários o utilizem mais para explicitar seus conhecimentos no ambiente facilitando dessa forma o processo de identificação desses conhecimentos.

---

<sup>4</sup> (En)cena <<http://ulbra-to.br/encena/>>

<sup>5</sup> Konnen <<http://conecta.ulbra-to.br>>

## 8 REFERÊNCIAS

- 4Chan. (2013). *4Chan*. Acesso em 2 de abril de 2013, disponível em 4Chan: <http://www.4chan.org/>
- Aberer, K., Cudr'e-Mauroux, P., Ouksel, A. M., Catarci, T., Hacid, M. S., Illarramendi, A., et al. (2004). *Emergent Semantics Principles and Issues*. Research Group of Distributed Information Systems (SID), University of Zaragoza, Zaragoza.
- Al-Ani, B., & Redmiles, D. (2009). Strangers We Trust? Findings of an Empirical Study of Distributed Development. *International Conference on Global Software Engineering* (pp. 121-130). Limerick, Ireland: IEEE Press.
- Alavi, M., & Smith, R. H. (fevereiro de 1999). KNOWLEDGE MANAGEMENT SYSTEMS: ISSUES, CHALLENGES, AND BENEFITS. *Communications of AIS*, 1, 7.1 - 7.37.
- Apache. (2013). *Apache Commons*. Acesso em 6 de junho de 2013, disponível em Apache Commons: <http://commons.apache.org/>
- Apache Lucene. (2011). *Apache Lucene*. Acesso em 16 de junho de 2013, disponível em Apache Lucene: <http://lucene.apache.org/>
- Apache Maven Project. (2002). *Apache Maven Project*. Acesso em 1 de junho de 2013, disponível em Apache Maven Project: <http://maven.apache.org/>
- Baeza-Yates, R., & Ribeiro-Neto, B. (s.d.). *Modern Information Retrieval*.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An Open Source Software for Exploring and Manipulating Networks. *International AAAI Conference on Weblogs and Social Media*.
- BibSonomy. (2013). *BibSonomy*. Acesso em 28 de maio de 2013, disponível em BibSonomy: <http://www.bibsonomy.org/>
- Boim, R., Kaplan, H., Milo, T., & Rubinfeld, R. (2010). Improved Recommendations via (More) Collaboration. *Proceedings of the 13th International Workshop on the Web and Databases* (pp. 12:1 - 12:6). Indianapolis, Indiana: ACM Press.
- Borst, P., Akkermans, h., & Top, J. L. (1997). Engineering ontologies. *Int. J. Hum.-Comput. Stud.*, pp. 365-406.



- Bozzon, A., Brambilla, M., Ceri, S., Silvestri, M., & Vesce, G. (2013). Choosing the right crowd: expert finding in social networks. *Proceedings of the 16th International Conference on Extending Database Technology* (pp. 637-648). New York, NY, USA: ACM Press.
- Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Journal Computer Networks and ISDN Systems*, pp. 107-117.
- BRUSILOVSKY, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2), 87-110.
- CAPES. (2013). *Capex: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*. Acesso em 08 de junho de 2013, disponível em Coordenação de Aperfeiçoamento de Pessoal de Nível Superior: <http://www.capes.gov.br/>
- Casciaro, T., & Lobo, M. S. (Junho de 2005). Competent Jerks, Lovable Fools, and the Formation of Social Networks. *Harvard Business Review*, pp. 2 - 8.
- Chacon, S. (2009). *Pro Git*. Berkely, CA, USA: Apress.
- Contractor, N. S., & Monge, P. R. (forthcoming). Using Multi-Theoretical Multi-Level (MTML) Models to Study Adversarial Networks. In: R. Breiger, & K. M. Carley (Eds.), *Summary of the NRC workshop on Social Network Modeling and Analysis*.
- Datta, A., Yong, J. T., & Ventresque, A. (2011). T-RecS: Team Recommendation System through Expertise and Cohesiveness. *Proceedings of International World Wide Web Conference Committee* (pp. 201-204). Hyderabad, India: ACM Press.
- Delicious. (2013). *Delicious*. Acesso em 10 de junho de 2013, disponível em Delicious: <https://delicious.com/>
- Demartini, G. (2007). Finding experts using wikipedia. *Proceedings of the Workshop on Finding Experts on the Web with Semantics (FEWS2007) at ISWC/ASWC2007*. Busan, South Korea: 33-41.
- Douglis, F. (Janeiro/Fevereiro de 2010). It's All About the (Social) Network. *Internet Computing*, 4 - 6.
- Duhon, B. (1998). It's all in our heads. *Proceedings of Informatiktage*, (pp. 8 - 13).
- Earl, M. (Verão de 2001). Knowledge management strategies: Toward a taxonomy. *Journal of Management Information Systems*, pp. 215-233.

- Facebook. (2013). *Facebook*. Acesso em 2 de abril de 2013, disponível em Facebook: <http://www.facebook.com/>
- Fazel-Zarandi, M., & Fox, M. S. (2011). Constructing expert profiles over time for skills management and expert finding. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies* (pp. 5:1-5:6). Graz, Austria: ACM Press.
- Fazel-Zarandi, M., & Fox, M. S. (2012). An Ontology for Skill and Competency Management. *Proceedings of the 7th International Conference on Formal Ontology in Information Systems*. Graz, Austria.
- Fazel-Zarandi, M., Devlin, H. J., Huang, Y., & Contractor, N. (2011). Expert recommendation based on social drivers, social network analysis, and semantic data representation. *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (pp. 41- 48). Chicago, Illinois: ACM Press.
- Gephi. (2013). *Gephi*. Acesso em junho de 11 de 2013, disponível em Gephi: <https://gephi.org/>
- Git. (2013). *Git --distributed-even-if-your-workflow-isnt*. Acesso em 1 de junho de 2013, disponível em Git-SCM: <http://git-scm.com/>
- Giunchiglia, F., & Zaihrayeu, I. (2007). *Lightweight Ontologies*. UNIVERSITY OF TRENTO: DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY, Trento (Italy).
- Gollapalli, S. D., Mitra, P., & Giles, C. L. (2011). Ranking authors in digital libraries. *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries* (pp. 251-254). New York, NY, USA: ACM Press.
- Gollapalli, S. D., Mitra, P., & Giles, C. L. (2012). Similar researcher search in academic environments. *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries* (pp. 167-170). New York, NY, USA: ACM Press.
- Google. (2013). *Google*. Acesso em 11 de junho de 2013, disponível em Google: <http://google.com.br/>
- Google+. (2013). *Google+*. Acesso em 2 de abril de 2013, disponível em Google+: <https://plus.google.com>

- Gruber, T. (junho de 1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, pp. 199-220.
- Guarino, N. (1998). *Formal Ontology and Information Systems*. (pp. 3-15). IOS Press.
- Huang, Y., Contractor, N., & Yao, Y. (2008). CI-KNOW: recommendation based on social networks. *Proceedings of the 2008 international conference on Digital government research* (pp. 375-376). Montreal, Canada: Digital Government Society of North America.
- Imasters. (2013). *Imasters*. Acesso em 2 de abril de 2013, disponível em Imasters: <http://forum.imasters.com.br/>
- Jacomy, M., Heymann, S., Venturini, T., & Bastian, M. (2011). *Force atlas 2, a graph layout algorithm for handy network visualization*.
- JUnit. (2013). *JUnit*. Acesso em 1 de junho de 2013, disponível em JUnit: <http://junit.org/>
- Kleinberg, J. M. (1999). Hubs, authorities, and communities. *ACM Comput. Surv.*, 31, p. 5.
- Lin, C., Xie, R., Li, L., Huang, Z., & Li, T. (2012). PRemiSE: personalized news recommendation via implicit social experts. *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1607 - 1611). Maui, Hawaii, USA: ACM Press.
- Lin, C.-Y., Cao, N., Liu, S. X., Papadimitriou, S., Sun, J., & Yan, X. (2009). SmallBlue Social Network Analysis for Expertise Search and Collective Intelligence. *IEEE 25th International Conference on Data Engineering*, (pp. 1483-1486).
- Lin, C.-Y., Ehrlich, K., Griffiths-Fisher, V., & Desforges, C. (2008). SmallBlue: People Mining for Expertise Search. *Multimedia at Work IEEE Computer Society*, 78-84.
- Lin, C.-Y., Ehrlich, K., Griffiths-Fisher, V., & Desforges, C. (janeiro - março de 2008). SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence. *IEEE MultiMedia*, 15(1), 78-84.
- LinkedIn. (2013). *LinkedIn*. Acesso em 9 de junho de 2013, disponível em LinkedIn: <http://linkedin.com/>
- Liu, D.-R., Chen, Y.-H., Kao, W.-C., & Wang, H.-W. (2013). Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Inf. Process. Manage.*, pp. 312-329.

- Maybury, M. T. (2006). *Expert finding systems*. Massachusetts, USA: MITRE Center for Integrated Intelligence Systems Bedford.
- Meira, S. R., Silva, E. M., Costa, R. A., & Jucá, P. M. (2011). Folksonomia. In: H. Fuks, & M. Pimentel, *Sistemas Colaborativos* (1 ed., Vol. único, pp. 221-229). Rio de Janeiro: Elsevier.
- Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (pp. 187-192). Edmonton, Canada: American Association for Artificial Intelligence.
- Mika, P. (março de 2007). Ontologies are us: A unified model of social networks and semantics. *Web Semant*, pp. 5 - 15.
- Mika, P., & Akkermans, H. (2004). *Towards a New Synthesis of Ontology Technology And Knowledge*. 2004) 317–345 DOI ONLINE 11.
- Ministério da Ciência Tecnologia e Inovação. (2013). *Plataforma Lattes*. Acesso em 1 de junho de 2013, disponível em Plataforma Lattes: <http://lattes.cnpq.br/>
- Mochol, M., Paslaru, E., & Simperl, B. (2006). Bontas. Practical Guidelines for Building Semantic eRecruitment Applications. In *International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies (AST)*.
- Mochol, M., Wache, H., & Nixon, L. (2007). Improving the accuracy of job search with semantic techniques. (W. Abramowicz, Ed.) *Business Information Systems*, 4439, 301-313.
- Morgan, K. K. (2008). Does Polanyi's Tacit Knowledge Dimension Exist? *Proceedings of Polanyi Society Conference* (pp. 1 - 35). Chicago, IL: Loyola University.
- O'Donovan, J., & Smyth, B. (2005). Trust in Recommender Systems. *International Conference on Intelligent User Interfaces* (pp. 167 - 174). San Diego, Californi, USA: ACM Press.
- Oliveira, V. P. (s.d.). *O conhecimento tácito na transferência de conhecimento científico: Mr. Data aprende a dançar*.

- O'Sullivan, D., Wilson, D., & Smyth, B. (2002). Improving Case-Based Recommendation. *6th European Conference, ECCBR 2002 Proceedings* (pp. 278-291). Aberdeen, Scotland, UK: Springer-Verlag Berlin Heidelberg.
- Park, H. W. (2003). Hyperlink network analysis: A new method for the study of social structure on the web. *Connections*, pp. 49-61.
- Perer, A., & Shneiderman, B. (setembro de 2006). Balancing Systematic and Flexible Exploration of Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 12, pp. 693-700.
- Peters, I., & Becker, P. (2009). *Folksonomies: indexing and retrieval in Web 2.0*. Berlin: De Gruyter/Saur.
- Pimentel, M., & Fuks, H. (2011). *Sistemas Colaborativos* (1 ed., Vol. único). Rio de Janeiro, RJ: Elsevier.
- Reimer, U. (2004). *Lightweight Ontologies (LWO) versus Full-Fledged Ontologies*. Acesso em 10 de junho de 2013, disponível em AI-One: <http://www.ai-one.com/tag/lightweight-ontology/>
- Riahi, F., Zolaktaf, Z., Shafiei, M., & Milios, E. E. (2012). Finding expert users in community question answering. *WWW (Companion Volume)* (pp. 791-798). ACM Press.
- Rockenbach, C. A. (2012). *Recomendação de especialistas na plataforma de gestão de conhecimento*. Palmas.
- Rockenbach, C. A., & Silva, E. M. (2012). Proposta de um Sistema de Recomendação de Especialistas aplicado a uma Rede Social Educacional. *Anais do XIV Encontro de Computação e Informática do Tocantins*, (pp. 132-141). Palmas, Brasil.
- Rogers, E. M., & Kincaid, D. L. (1981). *Communication networks: Toward a new paradigm for research*. New York: Free Press.
- Shalizi, C. (2006). Shannon Entropy and Kullback-Leibler Divergence. In: C. Shalizi, *Advanced Probability II* (pp. 189-196).
- Silva, E. M. (2009). *SWEETS: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais*. Redife.

- Silva, E. M., Costa, R. A., Schmitz, L. R., & Meira, S. R. (2011). SWEETS: um Sistema de Recomendação de Especialistas aplicado a uma plataforma de Gestão de Conhecimento. *Revista de Informática Teórica e Aplicada*, 18(1).
- Silva, E. M., Rodrigues, D. O., Souza, J. G., Salgado, A. C., & Meira, S. R. (2012). T-SWEETS: an alternative to the Stimulus Collaboration from trust Inference in Social Networks. *Proceedings of 2012 Brazilian Symposium on Collaborative Systems* (pp. 168 - 173). São Paulo, SP: IEEE Computer Society.
- Social Network Analysis: Theory and Applications*. (s.d.).
- Song, X., Tseng, B. L., Lin, C.-Y., & Sun, M.-T. (2006). Personalized recommendation driven by information flow. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 509 - 516). Seattle, Washington, USA: ACM Press.
- Souza, J., Brito, P., Sousa, C., Silva, E., Fagundes, F., Oliveira, F., et al. (2012). *Aprendizagem Organizacional Através de uma Rede de Gestão de Conhecimento*. COPEX, CEULP/ULBRA, Palmas.
- Sphinx Technologies Inc. (2007). *Sphinx: Open Source Search Server*. Acesso em 16 de junho de 2013, disponível em Sphinx: Open Source Search Server: <http://sphinxsearch.com/>
- StackOverflow. (2013). *StackOverflow*. Acesso em 2 de abril de 2013, disponível em StackOverflow: <http://stackoverflow.com/>
- Twitter. (2013). *Twitter*. Acesso em 9 de junho de 2013, disponível em Twitter: <https://twitter.com/>
- Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge, NY: Cambridge University Press.
- Wikipedia. (2013). *Multigraph*. Acesso em 20 de abril de 2013, disponível em Wikipedia: <http://en.wikipedia.org/wiki/Multigraph>
- Wikipedia. (2013). *Wikipedia*. Acesso em 2 de abril de 2013, disponível em Wikipedia: <http://www.wikipedia.org/>
- Wiktionary. (2013). *Wiktionary*. Acesso em 2 de abril de 2013, disponível em Wiktionary: <http://www.wiktionary.org/>

Yahoo Answers. (2013). *Yahoo Answers*. Acesso em 2 de abril de 2013, disponível em Yahoo Answers: <http://answers.yahoo.com/>

Yimam-seid, D., & Kobsa, A. (2002). Expert Finding Systems for Organizations: Problem and Domain Analysis and the DEMOIR Approach. *Journal of Organizational Computing and Electronic Commerce*, 13, pp. 1-24.