A study about graphs in the representation of knowledge in discussion forums in Software Engineering

Patrick Rodrigo Da Silva[1], Érica Ferreira de Souza[1], Glaúcia Braga e Silva[2], Katia Romero Felizardo[1], Giovani Volnei Meinerz[1]

[1]Academic Department of Computing
Federal University of Technology - Paraná (UTFPR)- Cornélio Procópio, PR, Brazil

[2]Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa (UFV) - Florestal, MG, Brazil

patricksap@hotmail.com, ericasouza@utfpr.edu.br, glaucia@ufv.br, giovanimeinerz@utfpr.edu.br, katiascannavino@utfpr.edu.br

Abstract. *In the social web paradigm, discussion forums are effective tools to facilitate the knowledge transfer among developers. However, manually finding useful information in discussions on a particular topic is a complex task, making it a major challenge for knowledge management. The objective of this study is to explore the representation of knowledge supported by graphs generated from discussion forums in the context of Software Engineering. Firstly, graphs were built considering the discussion topics of the Stack Overflow forum. Visual analysis as well as analysis of thematic relevance of the graphs were performed. Next, an evaluation of the graphs generated through interviews with software industry professionals was also conducted in order to obtain a practical view of the study conducted. Using graphs generated from discussion forums can help the software industry identify useful information and new trends.*

Resumo. *No paradigma da web social, os fóruns de discussão são ferramentas eficazes para facilitar o compartilhamento de conhecimento entre os desenvolvedores. Contudo, encontrar manualmente informações úteis em discussões sobre um determinado tema é uma tarefa complexa, tornando-se um grande desafio para a gestão do conhecimento. O objetivo deste trabalho é explorar a representação do conhecimento apoiada em grafos gerados a partir de fóruns de discussão no contexto da Engenharia de Software. Primeiramente, foram construídos grafos de conhecimento considerando os tópicos de discussão do fórum Stack Overflow. Foram realizadas análises visuais e de relevância temática dos grafos gerados. Em seguida, também foi realizada uma avaliação dos grafos gerados por meio de entrevistas com profissionais da indústria de software, a fim de obter uma visão prática do estudo realizado. O uso de grafos gerados a partir de fóruns de discussão pode ajudar a indústria de software a identificar informações úteis e novas tendências.*

1. Introdução

No desenvolvimento de software, um grande volume de informação é gerado e tem se tornado um componente de grande capital intelectual. A Gestão do Conhecimento (GC) pode desempenhar um importante papel para representar e compartilhar o conhecimento

gerado nas organizações (Silva et al. 2020). A transformação do conhecimento tácito em explícito para que esse possa ser compartilhado e difundido tem sido objeto de muito investimento, resultando em propostas metodológicas e tentativas práticas que focalizam a passagem do individual e pessoal para o coletivo/grupal (Nonaka and Krogh 2009).

O conhecimento tácito gerado a partir de uma discussão de grupo, por exemplo, pode ser um item de conhecimento valioso. O paradigma web social pode ser útil para compartilhar o conhecimento tácito através das tecnologias interativas e colaborativas, tais como redes sociais e fóruns de discussões (Abidi et al. 2009). Fóruns de discussão, em particular, são ferramentas eficazes para facilitar o compartilhamento de conhecimento tácito e informal entre desenvolvedores. Os fóruns de discussão tornam-se ferramentas importantes para a GC uma vez que o conhecimento útil pode ser gerado e capturado durante as discussões (Falbo et al. 2004). Além disso, por meio dos fóruns é possível fazer conversão do conhecimento tácito em explícito (Nonaka and Krogh 2009).

Os fóruns de discussão na área de desenvolvimento de software fornecem uma enorme quantidade de conteúdo valioso. A disponibilidade de uma grande quantidade de discussões de diferentes tópicos oferece amplas oportunidades para aquisição de conhecimento. No entanto, os fóruns na área de desenvolvimento de software contém muitos tópicos discutidos e encontrar manualmente informações úteis nas discussões de um determinado tópico é uma tarefa minuciosa (Gottipati et al. 2011), tornando-se um grande desafio para a GC. Dessa forma, sendo um fórum de discussão uma base com uma grande quantidade de dados textuais é possível que sejam realizadas análises visuais por meio da codificação desse conhecimento a fim de facilitar o entendimento. Algumas das abordagens existentes para codificar e representar o conhecimento são: mapas cognitivos, árvores de decisão, taxonomias de conhecimento e grafos (Dalkir 2005).

Os grafos, também conhecidos como grafos de conhecimento, representam os dados na forma de um grafo conectado (Paulheim 2016). A representação do conhecimento a partir de um grafo torna-se apropriada, pois as informações consideradas importantes são representadas por meio de conceitos e relações entre eles. Os vértices do grafo são os conceitos e as arestas representam a proximidade dos conceitos no texto (Azevedo 2011).

Considerando o contexto acima, o objetivo deste trabalho é explorar a representação do conhecimento apoiado em grafos gerados a partir de fóruns de discussão no contexto da Engenharia de Software. Para determinar qual fórum de discussão seria objeto deste estudo, um mapeamento sistemático foi conduzido em (Silva et al. 2020), e assim, foram construídos grafos de conhecimento considerando os tópicos de discussão do fórum *Stack Overflow*. Foram realizadas análises visuais e de relevância temática dos grafos gerados. A análise visual tem como objetivo identificar relações entre os conceitos do grafo que possam ser consideradas relevantes, já a análise de relevância temática mede a qualidade da comunicação e relevância da discussão em relação ao tópico selecionado (Azevedo et al. 2011). Por fim, também foi realizada uma avaliação dos grafos por meio de entrevistas com gestores de empresas desenvolvedoras de software, a fim de obter uma visão prática do estudo conduzido.

O restante deste artigo está estruturado da seguinte forma. A seção 2 apresenta os trabalhos relacionados. A seção 3 discute os passos para condução da pesquisa. Os resultados e discussões são apresentados na seção 4. Na seção 5 são apresentados impressões

de profissionais da área. Por último, as considerações finais são apresentadas na Seção 6.

2. Trabalhos Relacionados

Diversos trabalhos têm surgido com o objetivo de representar o conhecimento por meio de grafos considerando dados oriundos da internet, como os fóruns de discussão e repositórios de código fonte. Em (Liu et al. 2017), por exemplo, foi apresentado um novo método de roteamento de perguntas no *Stack Overflow*, baseado em grafo de conhecimento, denominado *Knowledge Graph Question Routing framework* (KGQR). De acordo com os autores, quando surge uma nova pergunta, o KGQR fornece uma lista classificada de usuários mais adequados para respondê-la, com base em um modelo treinado. Em (Zhao et al. 2019), foi projetada uma ferramenta chamada GitGraph, que recebe como entrada um repositório do Sistema de Controle de Versões Distribuído (GIT) e constrói automaticamente um grafo associado ao repositório. Resultados experimentais preliminares mostraram que o GitGraph pode gerar corretamente grafos de conhecimento para projetos GIT e foram úteis para os usuários compreenderem os projetos.

No trabalho (Azevedo 2011), foi desenvolvida a ferramenta MineraFórum. A ferramenta visa auxiliar os professores na análise qualitativa das contribuições textuais registradas por alunos em fóruns de discussões educacionais. No trabalho, foram utilizadas técnicas de mineração textual e grafos de conhecimento. A partir da análise das postagens, o docente pode identificar quais alunos redigiram contribuições textuais que contemplam conceitos relativos ao tema da discussão, e quais discentes não o fizeram. Já em (Neto and Silva 2018), é apresentada a ferramenta ColMiner. A ferramenta ColMiner é capaz de medir a qualidade da comunicação em um ambiente de rastreamento de problemas, por meio do cálculo da relevância temática dos comentários do assunto para o tema principal da discussão. Seu objetivo foi identificar falhas no processo de comunicação. A abordagem da análise de comunicação também utilizou algumas técnicas de mineração de texto baseadas no uso de grafos para representar o conteúdo textual dos comentários.

Assim como os trabalhos relacionados mencionados anteriormente, este trabalho tem como objetivo explorar a representação do conhecimento por meio de grafos gerados a partir de fóruns de discussão. No entanto, o escopo deste trabalho contempla discussões na área de Engenharia de Software no fórum do *Stack Overflow*, com o intuito de obter uma visão prática sobre a geração de grafos de conhecimento e a possível detecção de informação útil que possa apoiar gerentes de projetos de software em uma organização.

3. Metodologia para geração dos grafos de conhecimento

Nesta seção, apresenta-se a metodologia definida para criação de grafos de conhecimento, composta por três etapas, conforme apresentado na Figura 1. A seguir, são apresentadas as etapas, bem como suas respectivas atividades.

Etapa 1 - Preparação dos dados. A etapa de preparação dos dados tem como objetivo importar os dados textuais de um fórum de discussão. Para determinar qual fórum seria considerado no estudo, um mapeamento sistemático da literatura foi conduzido (Silva et al. 2020). De acordo com os resultados do mapeamento, os desenvolvedores de software tem usado extensivamente o *Stack Overflow* para compartilhar conhecimento sobre desenvolvimento de software. Dessa forma, o repositório definido para construção dos grafos de conhecimento foi o *Stack Overflow*. Já para a escolha dos tópicos a serem

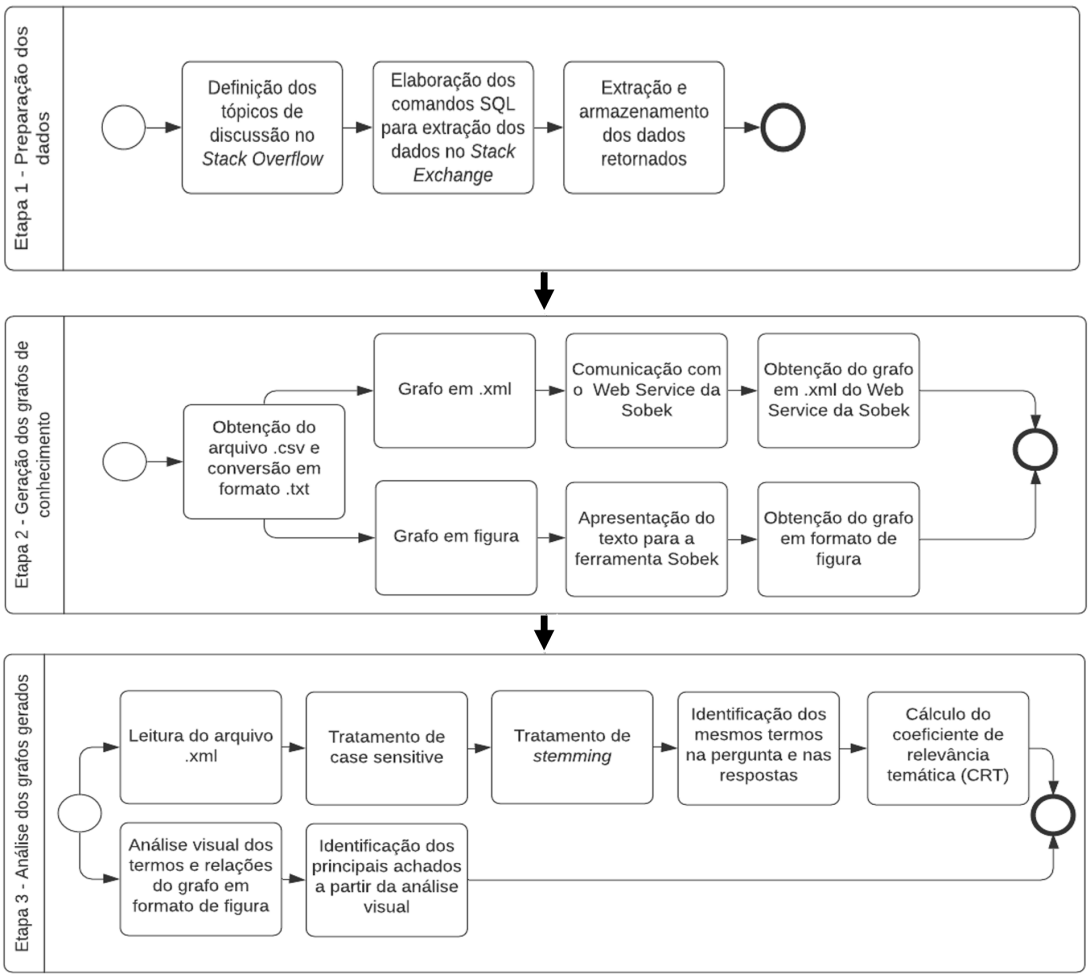

Figura 1. Etapas para criação do grafo de conhecimento

trabalhados foram considerados os seguintes parâmetros: aderência ao escopo de teste de software, devido ao conhecimento e à experiência dos autores deste artigo; e número de respostas registradas.

A segunda atividade é a extração dos dados textuais dos tópicos escolhidos no *Stack Overflow*. Cada tópico de discussão possui um identificador único (ID), por meio do qual é possível fazer pesquisas através de comandos SQL. Os comandos são executados no *Stack Exchange*[1], responsável por disponibilizar a consulta via comando SQL ao banco de dados do *Stack Overflow*. Vale destacar que duas consultas são necessárias; uma para a pergunta e outra para as respostas do tópico, pois no *Stack Overflow* são apresentadas em tabelas diferentes e relacionadas por um ID. Uma vez executadas as consultas SQL no *Stack Exchange*, o mesmo disponibiliza para download e armazenamento os dados em formato .CSV. Para geração dos grafos de conhecimento, foi necessário converter os arquivos .CSV para .txt e .xml.

Etapa 2 - Geração dos grafos de conhecimento. Para geração dos grafos, foi utilizada a Sobek[2], uma ferramenta de mineração textual que representa os dados encontrados em

[1]https://data.stackexchange.com/stackoverflow/query/new
[2]http://sobek.ufrgs.br/

formato de texto através de um grafo (Azevedo 2011). A ferramenta foi desenvolvida no Programa de Pós-Graduação em Informática na Educação, na Universidade Federal do Rio Grande do Sul (UFRGS). Foram utilizadas duas formas de representação de grafos utilizando a ferramenta Sobek: uma visual, por meio de uma figura; e uma textual, por meio de um arquivo .xml. A representação visual gera uma figura com o grafo de conhecimento contendo todo o conteúdo do tópico de discussão, a partir do qual é possível realizar uma análise visual dos nós (conceitos) e arestas (relacionamentos) do grafo. Já para a representação em arquivo .xml, utiliza-se o Web Service da ferramenta Sobek. A conversão do grafo em formato .xml torna-se necessária pois para ser realizado o cálculo de relevância temática é preciso ter acesso ao formato dos dados numéricos apresentados nos grafos, como, por exemplo, a quantidade de relacionamentos que um nó contém.

A Figura 2 apresenta um resumo do processo desde a extração dos dados até a geração dos grafos.

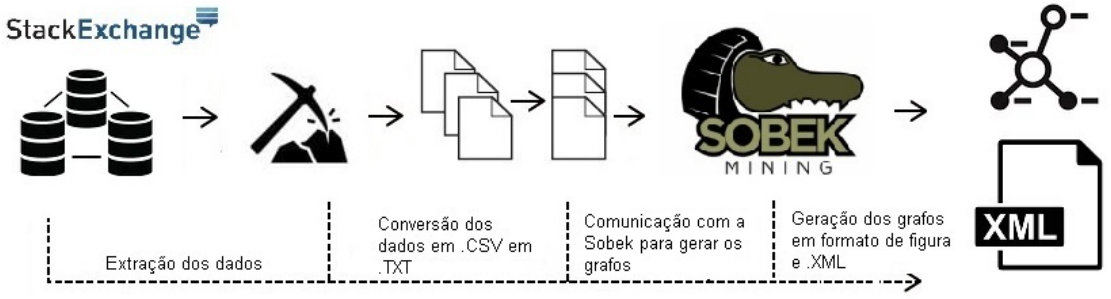

Figura 2. Processo da extração dos dados do tópico e geração dos grafos

Etapa 3 - Análise dos grafos. Uma vez gerados os grafos, tanto em formato de figura como em .xml, o objetivo desta etapa é a análise dos grafos. Duas análises foram consideradas neste estudo: análise visual e análise de relevância temática. A análise visual sobre a figura objetiva identificar conceitos e relações relevantes no grafo, como, por exemplo, novas tecnologias, *frameworks*, funcionalidades ou tendências. Já análise de relevância temática, foi conduzida a fim de medir a qualidade da comunicação e a relevância da discussão em relação ao tópico selecionado para análise. O termo "relevância temática", neste estudo, está sendo usado para representar a relevância das respostas em relação ao tema principal de uma discussão. A medição da relevância temática foi feita considerando o seguinte Cálculo da Relevância Temática (CRT) (Azevedo 2011):

$$CRT = NC + NA \tag{1}$$

Sendo, *NC* = número de conceitos relevantes utilizados no texto; e *NA* = número de relacionamentos entre os conceitos relevantes utilizados no texto. O valor mínimo para o CRT ser considerado relevante é 1. O valor mínimo indica que toda mensagem com valor de CRT maior ou igual a 1 é considerada relevante em relação ao tópico de discussão. O valor 1 é o menor número que pode ser calculado pela fórmula do CRT, situação alcançada quando *NC* = 1 e *NA* = 0. Os trabalhos (Azevedo 2011) e (Neto and Silva 2018) serviram como base para automatizar a realização do cálculo do CRT. Para este estudo foram criados diversos algoritmos em Python que apoiam todas as atividades da Etapa 3. Os algoritmos estão disponibilizados em *https://zenodo.org/records/10618628*.

Neste estudo, o algoritmo criado recebe os valores das tags (conceitos) retornadas no arquivo .xml. Em seguida, os valores da tags recebem tratamentos de *Case Sensitive* e *Stemming*. O tratamento de *Case Sensitive* foi aplicado para evitar que a mesma palavra que começa com maiúscula e outra com minúscula sejam consideradas diferentes. Já o tratamento de *Stemming*, foi realizado, pois, por muitas vezes o radical dos termos poderiam ser iguais. Uma vez efetuados os dois tratamentos nas tags, foram analisados os conceitos dos grafos (pergunta e resposta). Se um conceito da pergunta também constar na resposta, esse será contabilizado como conceito relevante. Do mesmo modo, se dois conceitos que constam na pergunta constarem também na resposta e tiverem um relacionamento entre si, esse relacionamento será contabilizado (Neto and Silva 2018). Esse processo de checagem (identificação e comparação) dos conceitos, também foi automatizado, ou seja, na checagem, leva-se em conta a comparação das listas de grafos das perguntas e das respostas do tópico a fim de identificar possíveis relações. Finalmente, o cálculo do CRT é realizado, conforme apresentado na Fórmula (1).

4. Resultados e Discussões

Para avaliar a metodologia definida neste estudo, três tópicos de discussão do *Stack Overflow* foram analisados considerando o escopo de "Teste de Software".

4.1. Tópico de Discussão 1

A Tabela 1 apresenta uma sumarização das principais características do primeiro tópico de discussão do *Stack Overflow* escolhido para análise.

Tabela 1. Características do Tópico de Discussão 1

Características	**Resultados**
Pergunta	*"What are unit tests, integration tests, smoke tests, and regression tests*?"
Tema	Teste de Unidade, Teste de Integração, Smoke e Teste de Regressão
Data de Criação	06 de fevereiro de 2009
Número de respostas	17
Número de visualizações	302.000
Número de caracteres	24.754
Número de palavras	4411
Número de frases	675
Número de parágrafos	242
SQL Criadas	Primeira extração: • Consulta do título e corpo do tópico: *select Title, Body FROM Posts WHERE Id=520064*; • Comentários feitos sobre o título: *select Text from Comments where PostId=520064*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=520064*; • Comentários feitos sobre as respostas: *select DISTINCT Text from Comments inner join Posts on PostId = ParentId WHERE ParentId=520064*; Segunda extração: • Consulta do título e corpo do tópico: *select Title,Body FROM Posts WHERE Id=520064*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=520064*;

A Figura 3 apresenta o grafo gerado para o Tópico de Discussão 1.

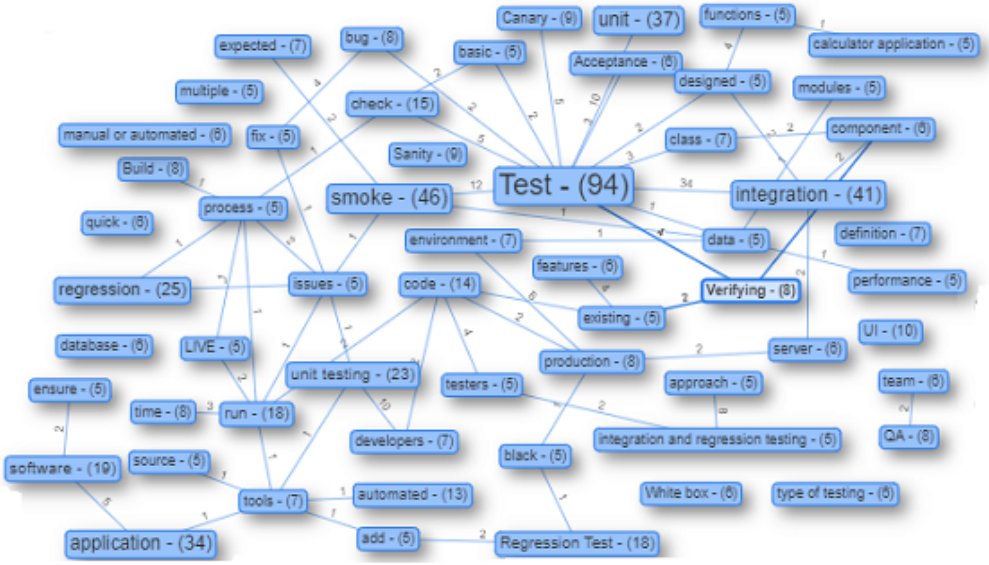

Figura 3. Grafo de conhecimento do Tópico de Discussão 1

(a) Análise Visual A seguir são apresentadas algumas interpretações e achados identificados no grafo do Tópico de Discussão 1 a partir de uma análise visual.

1. Relação entre os nós "*unit testing*" e "*developers*". Alinhado com o que é apresentado na literatura, a relação permite afirmar que o teste unitário é feito pelo próprio desenvolvedor quando testa pequenas partes de código fonte.
2. Relação entre os nós "*integration*", "*server*", "*designed*", "*component*" e "*test*". É possível inferir que o grafo gerado consegue demonstrar de forma visual relações que representam a finalidade do teste de integração, que é analisar o comportamento do software quando interage com outras aplicações ou processos.
3. Relação entre "*tools*"e "*automated*". A relação sugere que em vários momentos o assunto automação de teste entrou em discussão. Isso pode ser também notado em outras relações destacando técnicas de teste para automação.
4. Relação entre os nós "*canary*" e "*test*". A relação destaca um tipo de teste conhecido como "*canary test"* ou testes A/B. O teste canário consiste em habilitar novas funcionalidades apenas para um grupo controlado de usuários (os canários). De acordo com os respondentes do tópico, o teste canário é um teste automatizado e não destrutivo executado regularmente em um ambiente de produção, de forma que, se falhar, algo muito ruim acontecerá.
5. Relação dos nós "*smoke*" e "*test*". O *smoke test*, ou teste de fumaça, também está relacionado com a automação de teste. Esse tipo de teste compreende um conjunto não exaustivo de testes que visa garantir que as funções mais importantes funcionam. O resultado desse teste é usado para decidir se um software é estável o suficiente para prosseguir com testes adicionais.

(b) Relevância Temática A Tabela 2 apresenta os resultados do CRT para o Tópico de Discussão 1. Para o cálculo do CRT, foi gerado um grafo para cada resposta do tópico. Dessa forma, cada grafo gerado recebeu um ID, conforme apresentado na Tabela 2. Um cálculo do CRT é feito para cada grafo e depois uma média simples é apresentada.

Tabela 2. CRT para o Tópico de Discussão 1

ID	Quant. de Palavras	NC (N.º de Conceitos)	NA (N.º de Arestas)	CRT
1	263	4	1	5
2	63	2	0	2
3	41	1	0	1
4	191	2	1	3
5	106	4	1	5
6	173	4	1	5
7	118	3	1	4
8	171	2	1	3
9	64	4	3	7
10	59	1	0	1
11	179	4	3	7
12	148	2	0	2
13	664	2	0	2
14	109	2	1	3
15	28	1	0	1
16	56	1	0	1
17	83	2	1	3
Média	148	2	0,8	3

A seguir são apresentados as principais conclusões a partir da análise dos resultados do cálculo de CRT para o grafo do tópico de discussão 1.

1. O valor do CRT variou de 1 a 7, mostrando, assim, que todas as respostas continham relevância em comparação com a pergunta.
2. Quatro respostas do tópico (IDs 3, 10, 15 e 16) apresentaram valor igual a 1 para o CRT. Isso ocorreu, pois no grafo da resposta consta apenas um único conceito que existe no grafo da pergunta e sem nenhum relacionamento. A resposta 10, por exemplo, obteve valor de CRT igual a 1, pois o único conceito constante no grafo foi "*test*".
3. A resposta 13 contém o maior número de palavras (664 palavras) e obteve um valor de CRT igual a 2. Já a resposta 9, que contém 64 palavras, obteve o maior valor de CRT, sendo igual a 7. Isso mostra que não necessariamente uma resposta necessita conter muitas palavras para ter um valor de CRT alto.
4. Uma média simples dos resultados mostrou que a quantidade geral de palavras por resposta foi de 148, e o valor médio de CRT das respostas ficou em 3. Com isso pode-se concluir que as respostas do tópico têm uma boa relação com a pergunta, mostrando assim serem relevantes.

4.2. Tópico de Discussão 2

As principais características do segundo tópico de discussão analisado, encontram-se apresentadas na Tabela 3.

Tabela 3. Características do Tópico de Discussão 2

Características	**Resultado**
Pergunta	*"How to make junior programmers write tests?*"
Tema	Elaboração de testes por testadores iniciantes
Data de Criação	01 de Julho de 2015
Número de respostas	24
Número de visualizações	9.000

Número de caracteres	24.743
Número de palavras	4859
Número de frases	628
Número de parágrafos	162
SQL Criadas	Primeira extração: • Consulta do título e corpo do tópico: *select Title, Body FROM Posts WHERE Id=7252*; • Comentários feitos sobre o título: *select Text from Comments where PostId=7252*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=7252*; • Comentários feitos sobre as respostas: *select DISTINCT Text from Comments inner join Posts on PostId = ParentId WHERE ParentId=7252*; Segunda extração: • Consulta do título e corpo do tópico: *select Title,Body FROM Posts WHERE Id=7252*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=7252*;

A Figura 4 apresenta o grafo gerado para o Tópico de Discussão 2.

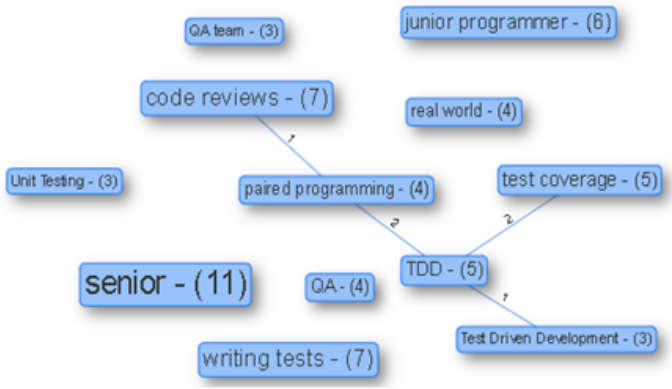

Figura 4. Grafo de conhecimento do Tópico de Discussão 2

(a) Análise Visual A seguir são apresentadas as principais interpretações e achados identificados a partir da análise visual do grafo do Tópico de Discussão 2.

1. Nó "*paired programming*". Programação em pares é uma prática utilizada pelo método ágil *eXtreme Programming*, e é indicada pelos respondentes como útil no aprendizado de testadores iniciantes.
2. Nó "*code reviews*". A revisão de código também é mencionada pelos respondentes como uma prática importante para aprendizado. Analisando as respontas dos respondentes, um deles mencionou que a adoção da práticas de revisão de código é muito eficiente quando aplicada com um intuito educativo e não punitivo.
3. Relação dos nós "TDD" e "*paired programming*". A prática *Test Driven Development* (TDD) tem como objetivo escrever os teste antes de desenvolver as funcionalidade do sistema. De acordo com o grafo gerado, TDD teve destaque em relação a testadores iniciantes. De acordo com um dos respondentes, a combinação entre TDD e programação em pares é desejável para um testador iniciante, pois melhora probabilidade de cobertura de teste.
4. Relação dos nós "TDD" e "*Test coverage*". A cobertura de teste é usada para medir o grau em que o código de um programa é executado (coberto) quando um determinado conjunto de testes é exercitado. A combinação de TDD e cobertura

de código também foi mencionada como boas práticas a serem utilizadas por testadores iniciantes.

(b) Relevância Temática A Tabela 4 apresenta os resultados do CRT para o Tópico de Discussão 2.

Tabela 4. CRT para o Tópico de Discussão 2

ID	Quant. de Palavras	NC (N.º de Conceitos)	NA (N.º de Arestas)	CRT
1	204	1	0	1
2	258	2	0	2
3	84	2	1	3
4	585	3	1	4
5	126	1	0	1
6	81	3	0	3
7	96	3	1	4
8	280	1	0	1
9	281	0	0	0
10	247	2	1	3
11	74	0	0	0
12	86	2	0	2
13	191	2	0	2
14	157	2	0	2
15	46	2	0	2
16	136	2	1	3
17	107	3	2	5
18	257	3	1	4
19	155	0	0	0
20	143	2	1	3
21	746	4	2	6
22	50	0	0	0
23	133	3	1	4
24	210	3	0	3
Média	188	1,91	0,5	2,41

As principais conclusões a partir da análise do CRT são apresentadas a seguir.

1. O valor de NC, variou de 0 a 4, e o valor de NA, ficou entre 0 e 2. O valor do CRT variou de 0 a 6. Das 24 respostas, apenas 3 tiveram o valor 1 (IDs 1, 5, 8), que é o mínimo para a resposta ser relevante.
2. Do total de 24 respostas, 4 tiverem o valor do CRT igual a 0 (IDs 9, 11, 19, 22), mostrando assim que nesses grafos não houve nenhum conceito nem aresta que constasse também no grafo da pergunta. Considerando o número total de respostas, o valor 4 foi um número considerável de respostas que não foram consideradas relevantes. A média geral do valor de CRT do tópico ficou em 2, mostrando que apesar de algumas respostas terem o resultado zero, a média geral teve um valor maior que 1 que é considerado o mínimo para ser relevante.
3. A resposta que obteve o maior valor de CRT foi a de ID 21, que teve o valor de 6. O número de palavras da resposta foi de 746 palavras.
4. As respostas que tiveram valor de CRT igual a zero, tiveram o número de palavras variando entre 50 e 281, por outro lado, a resposta que teve o menor número de palavras (ID 15), apresentou valor igual a 46, e obteve um valor de CRT igual a 2. Novamente, isso mostra que não necessariamente uma resposta com grande quantidade de palavras vai ter um número alto de CRT.
5. A média do valor de CRT foi de 2 e a média do número de palavras das respostas foi de 188.

4.3. Tópico de Discussão 3

A Tabela 5 apresenta as características do terceiro tópico de discussão do *Stack Overflow*.

Tabela 5. Características do Tópico de Discussão 3

Características	**Resultado**
Pergunta	*"A programming language designed to be testable*?"
Tema	É questionado no tópico se existe uma linguagem de programação que pode ser mais facilmente testada.
Data de Criação	28 de Outubro de 2014
Número de respostas	14
Número de visualizações	4.000
Número de caracteres	12.512
Número de palavras	2328
Número de frases	290
Número de parágrafos	74
SQL Criadas	Primeira extração: • Consulta do título e corpo do tópico: *select Title, Body FROM Posts WHERE Id=3371268*; • Comentários feitos sobre o título: *select Text from Comments where PostId=3371268*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=3371268*; • Comentários feitos sobre as respostas: *select DISTINCT Text from Comments inner join Posts on PostId = ParentId WHERE ParentId=3371268*; Segunda extração: • Consulta do título e corpo do tópico: *select Title,Body FROM Posts WHERE Id=3371268*; • Respostas: *select Title,Body FROM Posts WHERE ParentId=3371268*;

A Figura 5 apresenta o grafo gerado para o Tópico de Discussão 3.

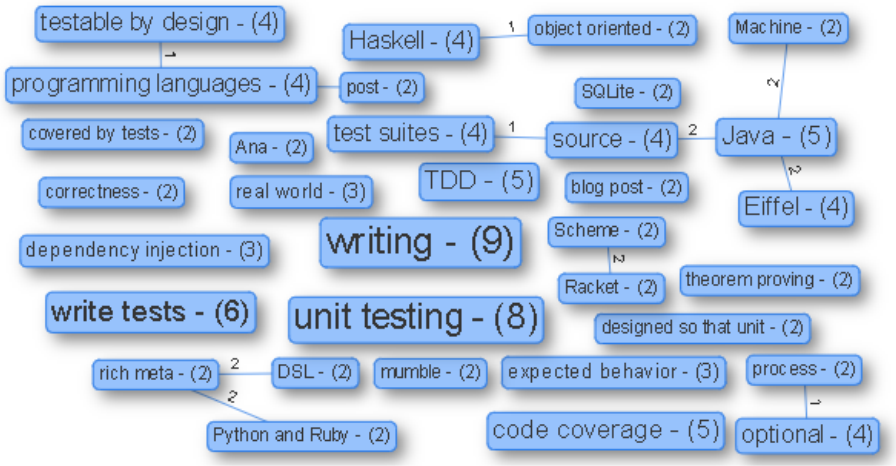

Figura 5. Grafo de conhecimento do Tópico de Discussão 3

(a) Análise Visual A seguir são apresentadas as principais interpretações identificadas no grafo a partir de uma análise visual.

1. Nó "*Eiffel*". A linguagem de programação *Eiffel* foi citada por um dos respondentes em relação a linguagens mais facilmente testáveis. De acordo com o res-

pondente, a linguagem encoraja o programador a colocar pré-condições e pós-condições em seu código, e elas são testadas.
2. Nó "*Racket*". De acordo com o respondente, se trata de uma linguagem de programação de pesquisa, e que na mesma, quando um teste falha, a linguagem informa não apenas a falha, mas também identifica em qual parte do código aconteceu a falha.
3. Nó "*Haskell*". De acordo com um dos respondentes, a *Haskell* é uma linguagem de programação puramente funcional, já que em tal linguagem as funções não têm efeitos colaterais e sempre produzirão os mesmos resultados com a mesma entrada, sendo assim, considerada uma linguagem com boas propriedades para condução de testes.
4. Nó "Ana". Onde o correto é "Anna", a linguagem é uma extensão da linguagem Ada e fornece anotações formais para todas as construções Ada. De acordo com um dos respondentes, a linguagem apoiou a elaboração de alguns testes, porém o mesmo não se aprofundou na explicação do porquê a linguagem apoiou testes. Segundo outro respondente, a linguagem Anna nunca passou da fase de pesquisa.
5. Outro achado foi a existência de linguagens mais comerciais como "Java", "Python" e "Ruby".
6. Um achado relevante para esse tópico foi a grande quantidade de nós com pouco ou nenhum relacionamento. Isso mostra que houve várias respostas interessantes, com bastante conteúdo, mas que não continham muitas relações entre si.

(b) Relevância Temática A Tabela 6 apresenta os resultados do CRT para o Tópico de Discussão 3, e a seguir são apresentadas as principais conclusões a partir da análise dos resultados do cálculo de CRT.

Tabela 6. CRT para o Tópico de Discussão 3

ID	Quant. de Palavras	NC (N.º de Conceitos)	NA (N.º de Arestas)	CRT
1	223	5	0	5
2	240	3	1	4
3	152	2	0	2
4	18	0	0	0
5	115	2	1	3
6	732	3	0	3
7	94	1	0	1
8	228	2	1	3
9	16	0	0	0
10	60	3	2	5
11	38	1	0	1
12	98	0	0	0
13	41	0	0	0
14	35	1	0	1
Média	149	1,64	0,35	3

1. O valor de NC variou entre 0 e 5, e o valor de NA, ficou entre 0 e 2. O valor de CRT variou entre 0 e 5.
2. Um achado que chamou a atenção deste tópico foi o grande número de grafos de respostas com valor de CRT igual a 0. Isso mostra que nesses grafos não houve nenhum conceito ou aresta que existisse também no grafo da pergunta. Foram 4 respostas com valor zero (IDs 4, 9, 12, 13).
3. A resposta que contém o menor número de palavras foi de 16 palavras (ID 9), obtendo um valor de CRT igual a 0. E a que obteve o maior valor de CRT, continha 223 palavras (ID 1).

4. A média geral de valor de CRT foi 2 e a média do número de palavras foi 149, mostrando assim que mesmo contendo um grande número de respostas com CRT igual a zero, na média geral, o valor de CRT ainda pode ser considerado relevante.

5. Visão prática de profissionais da área

Uma vez que os grafos foram criados e analisados, uma análise preliminar a partir da visão prática de profissionais da área também foi conduzida. Foram realizadas entrevistas semiestruturadas com dois profissionais de empresas de desenvolvimento de software. Os entrevistados possuem mais de 15 anos de experiência na indústria de software e com profundo conhecimento em teste de software. As entrevistas foram realizadas de forma síncrona por meio de videoconferência (via *Google Meet*) e gravadas. As perguntas realizadas na entrevista estão disponíveis em *https://zenodo.org/records/10618628*.

PRIMEIRA ENTREVISTA O primeiro profissional entrevistado é um diretor de tecnologia em uma empresa localizada na cidade de São Paulo-SP. Atualmente, é responsável por atividades de testes de software na empresa. A empresa tem como foco principal o desenvolvimento web, tendo como seguimento a criação de plataformas de incentivo de marketing para empresas. A empresa contém 106 funcionários distribuídos em todo o Brasil e adota metodologias ágeis em seus projetos.

Inicialmente, foram realizadas perguntas relacionadas às práticas de GC. De acordo com o entrevistado, a empresa utiliza um *Wiki* interno para registrar características dos produtos desenvolvidos e também uma prática de compartilhamento de erros. Também são utilizadas algumas ferramentas que permitem facilitar a troca e aquisição de conhecimento: *TEAMS*[3] e a *YAMMER*[4]. A ferramenta *TEAMS* permite a comunicação interna do time. Já a *YAMMER* permite desenvolver algumas práticas de GC. Por exemplo, existe uma função chamada "Quem sabe compartilha" que permite o time divulgar boas práticas e lições. O entrevistado também destacou que os desenvolvedores utilizam o fórum *Stack Overflow* para buscar conhecimento.

Na sequência, foi apresentado para o entrevistado o estudo conduzido neste artigo, e, em seguida, algumas perguntas foram feitas com intuito de coletar as impressões do profissional sobre uma possível utilização do uso de grafos na empresa. De acordo com o entrevistado, um grafo pode facilitar o entendimento de um tópico de discussão, pois mostra um resumo visual do tópico a ser analisado tornando possível inferir informações relevantes, visualizar comportamentos e mostrar tendências, como, práticas de teste mais utilizadas, por exemplo. Durante a entrevista, essa atividade de análise de conceitos nos grafos acabou sendo realizada junto com o entrevistado. O entrevistado também afirmou que utilizaria tais práticas na empresa. Ele mencionou que se o grafo mostrasse novas ferramentas, a chance dele fazer uma busca por estas depois, seria bem maior.

Por fim, ao ser questionado sobre uma possível desvantagem na utilização de grafos, o entrevistado apontou como desvantagem a possibiliade da fonte de dados ser viciada, inconsistente ou tendenciosa, pois não traria dados confiáveis para a análise.

SEGUNDA ENTREVISTA O segundo entrevistado assume o papel de *Product Owner* na empresa. A empresa fica localizada na cidade de Santo Antônio da Platina – PR e

[3]https://www.microsoft.com/pt-br/microsoft-teams/group-chat-software

[4]https://www.microsoft.com/pt-br/microsoft-365/yammer/yammer-overview

tem como principal produto um Sistema de Gestão Empresarial (*Enterprise Resource Planning* – ERP) para controle financeiro, estoque, vendas e área fiscal. A empresa adota práticas ágeis de desenvolvimento, principalmente relacionadas ao *framework Scrum*, e possui a certificação Melhoria de Processo do Software Brasileiro (MPS.BR).

Sobre as práticas de GC, o profissional respondeu que as mesmas não são aplicadas na empresa de forma explícita. De acordo com o entrevistado, o conhecimento adquirido fica com o indivíduo, e pode até ser compartilhado, porém de maneira informal durante os eventos de uma *Sprint*[5]. Além disso, a documentação da empresa hoje é considerada mínima. O entrevistado disse que a empresa possui um espaço no Wiki, mas não existe o hábito de registrar informações nesse espaço. Ele afirmou também que criar esse hábito será um grande desafio para a organização, e que entende como uma grande necessidade tornar explícito o conhecimento gerado.

Na próxima etapa da entrevista, foi apresentado para o entrevistado o estudo realizado neste artigo a fim de coletar impressões. De acordo com o entrevistado, um grafo facilitaria a interpretação de um tópico de discussão, pois auxiliaria na síntese e interpretação de tudo que é retornado em uma pesquisa online. O entrevistado mencionou que utilizaria grafos de conhecimento na empresa, inclusive ele destacou que o uso de tais práticas seriam vantajosas para mais áreas, além do desenvolvimento e teste de software, por exemplo, para os profissionais envolvidos com a área de suporte do software. Já em relação às desvantagens que o entrevistado visualiza, a resposta foi a possibilidade de o termo aparecer no grafo, mas não no contexto exatamente esperado. Por exemplo, uma técnica de desenvolvimento aparecer várias vezes no texto, tornar-se um conceito no grafo, mas não de forma positiva, e sim, uma técnica que não está sendo sugerida para ser utilizada no desenvolvimento. O grafo teria que ser inteligente o bastante para mostrar essa situação.

6. Conclusões

O objetivo geral deste trabalho foi explorar a representação do conhecimento apoiado em grafos gerados a partir de fóruns de discussão no contexto da Engenharia de Software. Considerando os resultados alcançados até o momento, é possível observar que os grafos de conhecimento podem auxiliar as empresas de desenvolvimento de software a identificarem informações úteis a partir de fóruns de discussão. Embora este estudo seja ainda preliminar, acredita-se que o uso de grafos possa auxiliar a indústria de software nas decisões de projeto, por exemplo, na identificação de novas tendências, ferramentas, *frameworks* ou práticas de desenvolvimento.

Algumas limitações foram encontradas neste trabalho. Na etapa de geração dos grafos, a principal limitação foi o processo manual de extração dos dados da base de dados do *Stack Overflow*. A criação de uma ferramenta para automatizar essa extração tornaria o processo mais rápido e menos trabalhoso. Já na condução das entrevistas, a principal limitação foi o número de profissionais entrevistados. Um número maior de entrevistas poderia melhorar a qualidade desta pesquisa; no entanto, dado o perfil dos entrevistados, acredita-se que as entrevistas realizadas trouxeram importantes contribuições para o estudo. Trabalhos futuros incluem a automatização do processo de captura dos dados do fórum de discussão através da criação de um *Web Crawler*; a análise de novos tópicos de

[5]Nome dado a um ciclo de desenvolvimento no Scrum.

discussão do *Stack Overflow*; e a realização de mais entrevistas com profissionais da área para a obtenção de uma visão mais ampla da aplicação da abordagem na indústria.

Referências

[Abidi et al. 2009] Abidi, S., Hussini, S., Sriraj, W., Thienthong, S., and Finley, A. (2009). Knowledge sharing for pediatric pain management via a web 2.0 framework. *Studies in health technology and informatics*, 150:287–91.

[Azevedo 2011] Azevedo, B. F. T. (2011). *Minerafórum: um recurso de apoio para análise qualitativa em fóruns de discussão.* PhD thesis, Universidade Federal do Rio Grande do Sul. Tese de Doutorado em Informática na Computação.

[Azevedo et al. 2011] Azevedo, B. F. T., Behar, P., and Reategui, E. (2011). Qualitative analysis of discussion forums. In *International Journal of Computer Information Systems and Industrial Management Applications.*, pages 671–678.

[Dalkir 2005] Dalkir, K. (2005). *Knowledge Management in Theory and Practice*. The MIT Press.

[Falbo et al. 2004] Falbo, R., Arantes, D., and Natali, A. (2004). Integrating knowledge management and groupware in a software development environment. volume 3336, pages 94–105.

[Gottipati et al. 2011] Gottipati, S., Lo, D., and Jiang, J. (2011). Finding relevant answers in software forums. In *26th IEEE/ACM International Conference on Automated Software Engineering*, page 323–332.

[Liu et al. 2017] Liu, Z., Li, K., and Qu, D. (2017). Knowledge graph based question routing for community question answering. In *Neural Information Processing*, pages 721–730, Cham. Springer International Publishing.

[Neto and Silva 2018] Neto, L. E. C. and Silva, G. B. e. (2018). Colminer: A tool to support communications management in an issue tracking environment. In *XIV Brazilian Symposium on Information Systems*, SBSI'18.

[Nonaka and Krogh 2009] Nonaka, I. and Krogh, G. (2009). Tacit knowledge and knowledge conversion: controversy and advancement in organizational knowledge creation theory. *Organization Science*, 30:635–652.

[Paulheim 2016] Paulheim, H. (2016). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508.

[Silva et al. 2020] Silva, P. R., Santos, V. ; Souza, E. F., Meinerz, G. V., Felizardo, K. R., and Vijaykumar, N. L. (2020). Extraction of useful information from unstructured data in software engineering: A systematic mapping. In *XXIII Ibero-American Conference on Software Engineering (CIBSE)*, pages 1–14.

[Zhao et al. 2019] Zhao, Y., Wang, H., Ma, L., Liu, Y., Li, L., and Grundy, J. (2019). Knowledge graphing git repositories: A preliminary study. In *International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 599–603.