

Uma Abordagem Baseada em Análises de Redes Sociais para Medir a Participação Colaborativa em Projetos de Software

Aislan Rodrigo de Almeida Leite¹, Gláucia Braga e Silva¹

¹Instituto de Ciências Exatas e Tecnológicas –
Universidade Federal de Viçosa (UFV) – *campus* Florestal
Florestal–MG–Brasil

{aislan.leite, glaucia}@ufv.br

Abstract. *Several tools are used in software development that record the information about team work. In general, these tools have different vocabularies, which makes it difficult to measure team collaboration. In this context, this paper presents a tool capable to integrate collaborative data from heterogeneous sources and performing the analysis of collaborative participation in software projects. The analyses are done after the application of commonly used metrics in social networks that evaluate the relevance of a node in relation to the network. For the evaluated software project, 87.5% of the most active users were present in the exchange of messages and the coordination of tasks; Other 80% were involved in both the message exchange and the shared edition of artifacts.*

Resumo. *No processo de construção de software, utilizam-se diversas ferramentas que registram as informações acerca do trabalho da equipe. Em geral, essas ferramentas possuem vocabulários diferentes, o que dificulta a análise sobre a colaboração da equipe. Nesse contexto, este trabalho apresenta uma ferramenta capaz de integrar dados colaborativos de fontes heterogêneas e analisar a participação colaborativa dos membros em projetos de software. As análises são feitas após a aplicação de métricas comumente utilizadas em redes sociais que avaliam a relevância de um nó em relação à rede. Para o projeto de software avaliado, 87.5% dos usuários mais ativos estiveram presentes na troca de mensagens e na coordenação de tarefas; Outros 80% estiveram envolvidos tanto na troca de mensagens quanto na edição compartilhada de artefatos.*

1. Introdução

Em um ambiente de desenvolvimento de software, equipes trabalham de forma colaborativa, acessando repositórios de informações compartilhadas, trocando informações entre si e executando tarefas de forma coordenada em prol da produção de artefatos de software. Nesse processo, a colaboração ocorre a todo momento (Schneider, 2011), podendo causar impactos negativos nos resultados do projeto, caso não seja conduzida de forma apropriada. Segundo Curtis et al. (1988), questões relacionadas à comunicação e à coordenação são os principais problemas no processo de desenvolvimento de software e devem ser tratados com cautela. Souza et al. (2009) ressaltam que a cooperação e a coordenação requeridas nesse processo têm se tornado tão importantes quanto os aspectos humanos envolvidos, com impactos diretos na construção de software em diferentes níveis segundo diferentes perspectivas. Desse modo, no que se refere aos aspectos sociais

do desenvolvimento de software, deve-se priorizar os relacionamentos que surgem entre os participantes de um projeto, ao invés dos atributos individuais dos participantes (Balieiro et al., 2007).

Considerando-se que os membros de uma equipe de software se conectam e estabelecem relações por meio das ferramentas de suporte, pode-se visualizar uma rede de colaboração que se comporta de forma similar a uma rede social (Balieiro et al., 2007). Nesse contexto, técnicas baseadas na análise de redes sociais (*social network analysis*) (Wasserman and Faust, 2007), (de Souza et al., 2005) têm sido adotadas para se avaliar aspectos relacionados à colaboração em desenvolvimento de software. Dentre as análises sobre a rede de colaboração, destacam-se aquelas que mostram a influência de um participante na rede. Essa influência pode ser obtida por meio de métricas quantitativas que calculam a centralidade do participante na rede (Rocha et al., 2016; Thung et al., 2013), ou seja, avaliam a influência do nó em de acordo com os outros nós da rede.

Em busca de soluções para analisar a influência de um membro enquanto colaborador em um projeto de software, este trabalho propõe uma ferramenta capaz de analisar a participação colaborativa dos membros da equipe. Para isso, utiliza-se métricas aplicáveis a redes sociais, com o intuito de quantificar a influência dos membros na rede em termos das atividades executadas (coordenação), das mensagens trocadas (comunicação) e dos artefatos produzidos em conjunto (cooperação) em um projeto de software.

Considerando-se que problemas relacionados à colaboração em um processo de software podem ocasionar atrasos e prejuízos financeiros, esta pesquisa mostra-se relevante, pois a análise da participação colaborativa dos membros pode fornecer indicadores úteis aos líderes de projetos de software para a correta distribuição de recursos e alocação de tarefas. Dentre esses indicadores pode-se citar: comunicadores mais ativos, membros mais produtivos e perfil do membro (se é mais efetivo na comunicação, coordenação e/ou cooperação).

O texto deste trabalho está organizado da seguinte forma: na seção 2, é apresentada a base conceitual para a compreensão da colaboração no contexto de projetos de software. Na seção 3, são discutidos os trabalhos relacionados. Na seção 4, apresenta-se o analisador de participação colaborativa proposto. Na seção 5, apresenta-se uma prova de conceito do analisador proposto. Por fim, a seção 6 traz as conclusões e trabalhos futuros.

2. Colaboração no contexto de Projetos de Software

Este trabalho aborda a colaboração em processo de software a partir do Modelo 3C, proposto originalmente por Ellis et al. (1991) e posteriormente adaptado por Fuks et al. (2007). Segundo esse modelo, para que a colaboração entre usuários ocorra, são necessárias atividades/ações de Comunicação, Coordenação e Cooperação entre eles. De acordo com Pimentel e Fuks (2011), a comunicação é caracterizada pela troca de mensagens entre pessoas; A coordenação caracteriza-se pelo gerenciamento de pessoas, atividades e recursos; E a cooperação refere-se à atuação conjunta num objeto de cooperação. A Figura 1 apresenta uma adaptação proposta por Gerosa (2006) para o Modelo 3C (Ellis et al., 1991).

Apesar da separação das tarefas para fins de análise, cada dimensão não ocorre de forma isolada, sendo realizadas de forma iterativa e contínua durante o trabalho em

Figura 1. Modelo 3C (Gerosa (2006), adaptado de Ellis et al. (1991))



grupo (Gerosa et al., 2005). As tarefas originam-se a partir dos compromissos assumidos na troca de mensagens entre os participantes, sendo gerenciadas pela coordenação e realizadas durante a cooperação. A percepção ocorre quando um participante obtém *feedback* de suas ações e *feedthrough* das ações da equipe. Assim, ao cooperar, o participante deve renegociar e tomar novas decisões, o que requer novas rodadas de comunicação e coordenação.

Acredita-se que cada uma das dimensões do Modelo 3C possa ser facilmente mapeada em um ambiente de software, uma vez que os membros realizam atividades de forma coordenada, se comunicando e editando artefatos de software de forma conjunta.

2.1. Fontes de Dados Colaborativos

Em um processo de software, a equipe utiliza várias ferramentas que auxiliam o trabalho, seja na comunicação entre membros, na coordenação de atividades ou no apoio à edição cooperativa de artefatos de software. Essas ferramentas, denominadas ferramentas CASE (*Computer-Aided Software Engineering*), são fundamentais para o trabalho em equipe e oferecem suporte a cada uma das dimensões do Modelo 3C. Sendo assim, os repositórios dessas ferramentas possuem as informações acerca do trabalho colaborativo realizado.

A Tabela 1 ilustra exemplos de dados colaborativos que podem ser obtidos de algumas ferramentas CASE, agrupados por dimensão do Modelo 3C.

Ao analisar os dados colaborativos disponíveis nos repositórios das ferramentas de suporte, observa-se que as informações possuem natureza heterogênea, uma vez que cada ferramenta possui seu vocabulário próprio. Essa natureza heterogênea faz com que sejam usados diferentes termos que possuem o mesmo significado, o que pode gerar dificuldades nas análises e interpretações dos dados. Por exemplo, *member* e *annotation* no MantisBT¹ e *user* e *comment* no Github², respectivamente, possuem o mesmo significado.

Dessa forma, para que os dados colaborativos sejam analisados de forma apropriada, é preciso, em um primeiro momento, lidar com o desafio dessa heterogeneidade.

¹Mantis Bug Tracker, é um popular sistema de código-aberto para rastreamento de defeitos em projetos de software

²Github, é uma comunidade de código-aberto bem estabelecida e com uma grande quantidade de repositórios hospedados: <http://github.com>

Tabela 1. Exemplos de dados colaborativos em projetos de software

Dimensão	Parâmetro	Fonte de Dados
Comunicação	E-mail	Gmail e Hotmail
	Chat	HipChat, Slack, WhatsApp e Skype
	Comentários em código-fonte	Códigos em diversas linguagens
Coordenação	Gestão de Workflow	Trello, Confluence, Jira e MantisBT
	Atribuição de atividades	Trello, Confluence, Jira e MantisBT
	Rastreamento de defeitos	Jira, Bugzilla e MantisBT
Cooperação	Edição colaborativa de códigos	Github, SVN e Bitbucket
	Edição colaborativa de documentos	Google Docs e MS Word Online
	Edição colaborativa de conteúdos	Wikis

Nesse contexto, podem ser utilizadas técnicas de integração semântica das informações, baseadas no uso de ontologias. Ontologia é uma especificação formal e explícita de uma conceitualização compartilhada, onde especificação formal quer dizer algo que é legível para os computadores Gruber (1993) ontologias podem ser utilizadas como uma interlíngua para mapear semanticamente conceitos usados por diferentes aplicações empresariais. Desse modo, o uso de ontologias objetiva estabelecer um vocabulário comum, que permita a integração de informações heterogêneas e tem sido utilizado em vários estudos para solucionar a integração semântica entre fontes heterogêneas e garantir a interoperabilidade entre aplicações (Izza, 2009).

No âmbito do uso de ontologias para integração semântica de dados heterogêneos, Calhau e Falbo (2008) apresentaram uma proposta de integração entre ferramentas de apoio à Gerência de Configuração de Software (GCS), sendo o Subversion e a ferramenta GCS-ODE. No domínio de integração semântica de ferramentas de GCS, Braga e Silva (2013) propôs uma ontologia do domínio da GCS para integração semântica entre dados colaborativos armazenados nos repositórios das ferramentas MantisBT e Subversion. De forma similar, de Souza (2014) propôs uma ontologia do domínio de testes de software, utilizada em um *framework* de gestão de conhecimento. De forma similar, este trabalho utilizará uma ontologia do domínio das colaborações em projetos de software para prover a integração semântica de dados colaborativos provenientes de diversas ferramentas de suporte ao trabalho colaborativo da equipe de software.

2.2. Rede de Colaboração em Projetos

Um projeto de desenvolvimento de software possui pessoas que trabalham de forma colaborativa em prol da produção de artefatos de software. Dessa maneira, pode-se derivar uma rede de colaboração dos envolvidos do projeto de software, sendo que essa rede pode ser analisada como uma rede social de desenvolvedores (Jiang et al., 2013).

Dentre as métricas utilizadas para analisar participações de indivíduos nas redes sociais, existem aquelas referentes ao cálculo da centralidade de um nó em uma rede: a Centralidade de Grau (Scott, 2000) e a Centralidade de Proximidade (Freeman, 1979).

De acordo com Scott (2000), a centralidade de grau para um nó é a sua quantidade de nós adjacentes, assim, reflete-se a ideia de um nó importante estar conectado com

muitos nós. Desta forma, em uma rede não-orientada a centralidade de grau do nó v_i é dada pelo seu grau, ou seja:

$$C_G(v_i) = k_i, \quad (1)$$

onde, v_i é o i -ésimo nó e k_i a quantidade de nós adjacentes a v_i .

Segundo Newman (2003) e Freeman (1979), a centralidade de proximidade pode ser uma métrica de influência que um determinado nó possui na rede, ou seja, ela reflete o fato de que um nó importante está próximo dos demais. A centralidade de proximidade de um nó v_i é obtida pelo inverso da soma das suas distâncias aos demais nós da rede, ou seja:

$$C_P(v_i) = \frac{1}{\sum_{v_j \in v, v_j \neq v_i} d(v_i, v_j)} \quad (2)$$

onde, v_i é o i -ésimo nó, v_j é atingível a partir de v_i e $\sum_{v_j \in v, v_j \neq v_i} d(v_i, v_j)$ é o somatório dos menores caminhos a partir de v_i para todos os seus nós atingíveis.

A escolha da centralidade de grau e de proximidade se deve ao fato de ambas quantificarem o nó em relação a rede. Dessa forma, é possível mensurar a influência que o nó possui na rede.

3. Trabalhos Relacionados

Do ponto de vista da análise de redes, Thung et al. (2013) modela o Github como rede social colaborativa descrevendo a estrutura da rede, a formação dos participantes e seus relacionamentos, para analisar a influência de desenvolvedores e projetos nos repositórios Github utilizando métricas, como a centralidade de proximidade. Aqui, o objetivo é mais específico, pois iremos analisar a participação colaborativa em projetos de software de acordo com a troca de mensagens, a coordenação de tarefas e a cooperação entre os participantes. Ainda no contexto de análise de redes no Github, Rocha et al. (2016) definiu que a influência da contribuição dos desenvolvedores por meio de linguagens de programação nos repositórios do Github, por meio da centralidade de grau. Entretanto, o trabalho não avaliou a colaboração entre os membros no repositório. Neste trabalho, a análise da colaboração será realizada por meio da aplicação das métricas de centralidades de proximidade e de grau em cada dimensão do Modelo 3C, com o intuito de avaliar a influência de cada desenvolvedor na rede como um todo e também em cada dimensão.

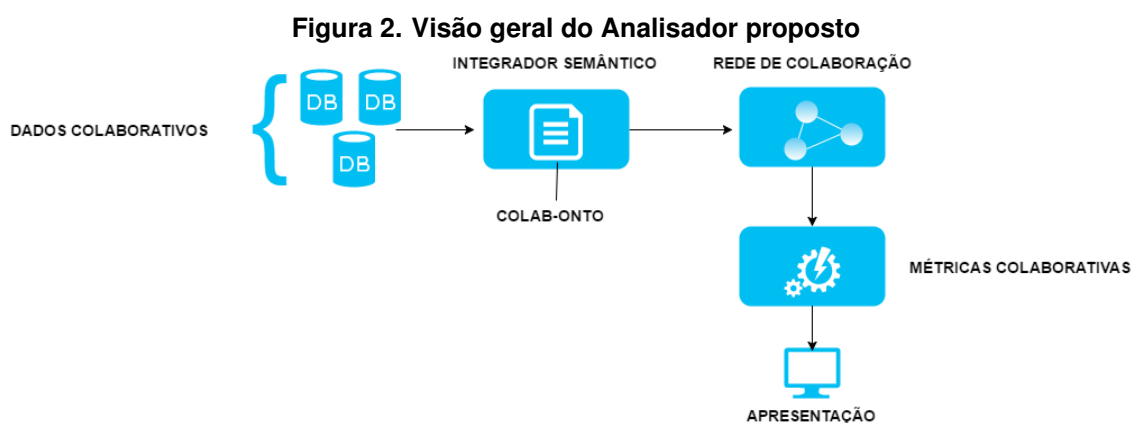
No contexto de ferramentas que realizam a extração de dados colaborativos para análise de informações em repositórios de software, da Silva et al. (2013) descrevem uma ferramenta para extração e mineração de repositórios de softwares, e apresentaram os participantes mais ativos em um projeto. Neste trabalho, a análise sobre os participantes mais ativos será feita com base em métricas comumente utilizadas em redes sociais. Já Bettenburg (2011), extraiu repositórios de software e apresentou o impacto da comunicação na colaboração em repositórios de software. Entretanto, como a carga dos dados não eram semanticamente integradas, a extração dos repositórios não poderia ser de fontes heterogêneas. Apesar de contextos diferentes, este trabalho contribui para essa discussão ao analisar a influência do desenvolvedor na rede de contribuintes do projeto e aplicar mé-

tricas para cada dimensão do Modelo 3C, através da carga dos dados colaborativos vindo de fontes heterogêneas.

4. Analisador de Participação Colaborativa em Projetos de Software

Esta seção é responsável por apresentar e detalhar o desenvolvimento de uma ferramenta capaz de realizar a análise da participação colaborativa de membros em projetos de software.

A Figura 2 apresenta uma visão geral da ferramenta proposta, composta de 3 elementos principais: um integrador semântico, responsável por integrar semanticamente os dados colaborativos; uma rede de colaboração, responsável por estruturar os dados em forma de rede e a aplicação de métricas para análise da participação colaborativa em projetos de software.



4.1. Carga e Integração Semântica dos Dados

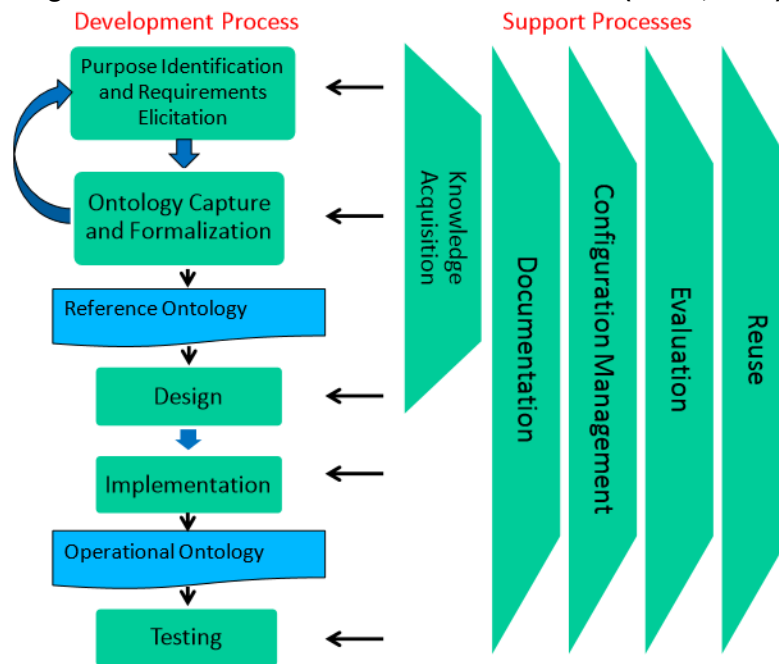
Para a obtenção dos dados, existe uma dependência tecnológica do repositório onde os dados estão armazenados e se existem ou não recursos fornecidos pelas próprias ferramentas para prover acesso aos dados (APIs, por exemplo).

Além das restrições tecnológicas, existe o desafio de se lidar com dados heterogêneos. Nesse contexto, este trabalho utiliza uma abordagem de integração semântica baseada em uma ontologia do domínio das colaborações em projetos de software, denominada Colab-Onto.

4.1.1. Colab-Onto

Esta seção apresenta a Colab-Onto, uma ontologia do domínio das colaborações. A Colab-Onto foi baseada em uma ontologia desenvolvida de acordo com o Modelo 3C, proposta inicialmente por Fuks et al. (2007), e a metodologia SABiO (*Systematic Approach for Building Ontologies*) (Falbo, 2014). Essa metodologia é composta por cinco fases: (i) Identificação de Propósito e Especificação de Requisitos; (ii) Captura e Formalização da Ontologia; (iii) Design; (iv) Implementação e (v) Testes, conforme ilustra a Figura 3.

Figura 3. Processo de desenvolvimento SABiO (Falbo, 2014)



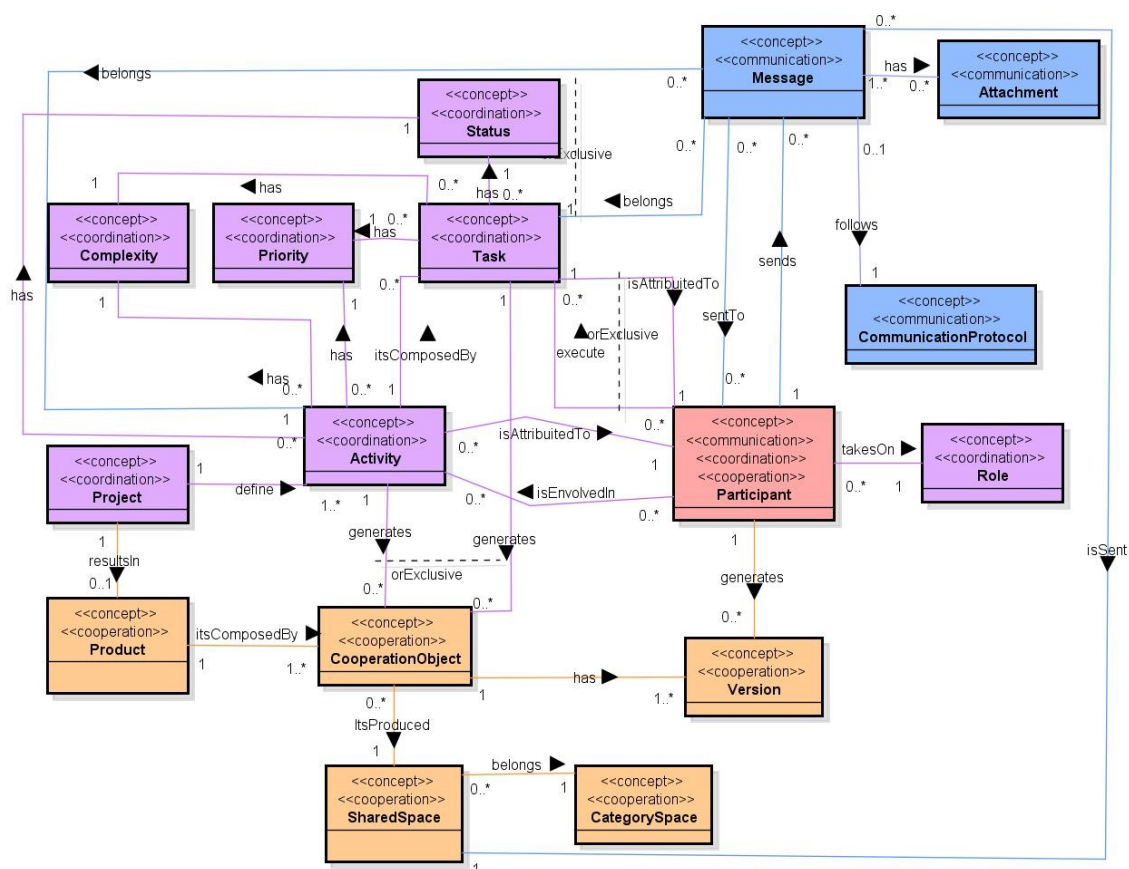
A fase de Identificação de Propósito e Especificação de Requisitos de uma ontologia possui como principal requisito a identificação dos requisitos funcionais e não-funcionais. Na fase de Captura e Formalização da Ontologia, é necessário descrever a competência da ontologia, por meio da modelagem conceitual. Para essa fase, utilizou-se a linguagem UML³ para representação dos conceitos, como classes, das propriedades, como atributos e das relações entre os conceitos, como associações entre classes. A fase de Design, uma versão operacional da ontologia deve ser produzida para que ela possa ser utilizada por alguma aplicação de computador. Já a fase de Implementação da ontologia, refere-se à codificação da mesma em uma linguagem de programação. Uma vez que a ontologia foi implementada, pode-se desenvolver casos de Testes para verificar e validar a ontologia dinamicamente.

Para facilitar a visualização de cada dimensão dos conceitos, a Figura 4 apresenta a Colab-Onto com conceitos de cada dimensão agrupados por um esquema de cores e com estereótipos contendo a dimensão para cada conceito.

Conforme pode ser observado na Figura 4, os conceitos na cor azul são aqueles que representam troca de mensagens entre os participantes (comunicação). Em laranja, são os conceitos relacionados à geração de objetos de cooperação em um espaço compartilhado (cooperação). Já os conceitos demarcados na cor roxa, são aqueles que modelam a coordenação das atividades colaborativas (coordenação). O Participante está diretamente envolvido nas três dimensões do Modelo 3C, ou seja, é um conceito central e por isso está destacado em cor avermelhada.

³Linguagem de Modelagem Unificada (do inglês, UML - *Unified Modeling Language*)

Figura 4. Representação da Colab-Onto



4.1.2. O Integrador Semântico

Para que a integração aconteça, demanda-se por um mapeamento entre os parâmetros a serem integrados e a Colab-Onto. O mapeamento busca obter uma correspondência entre os vocabulários das ferramentas CASE e a Colab-Onto. Dessa forma, esse mapeamento é feito com base na semelhança entre os parâmetros do recurso da ferramenta colaborativa e as definições do dicionário de termos da Colab-Onto. Entretanto, dependendo da origem dos dados, é possível que nem todos os conceitos da ontologia sejam mapeados, pois a Colab-Onto representa o domínio da colaboração como um todo e as ferramentas podem não fornecer apoio a todas as dimensões do 3C.

A Tabela 2 ilustra alguns parâmetros separados por dimensão que podem ser mapeados para Colab-Onto.

A partir da carga de dados e do mapeamento, são geradas instâncias da Colab-Onto correspondentes aos dados da(s) fonte(s) de dados colaborativos. Para ilustrar a instanciamento para a Colab-Onto, a Tabela 3 apresenta dados colaborativos fictícios separados por cada dimensão do Modelo 3C.

Tabela 2. Exemplo de parâmetros que podem ser mapeados para a Colab-Onto

Dimensão	Parâmetro	Conceito da Colab-Onto
Comunicação	Comment / Annotation / Message	Message
Coordenação	Issue / Project / Card	Activity
Cooperação	Files / Page Wiki / Google Doc	Cooperation Object
Comunicação, Coordenação e Cooperação	Member / User / Developer	Participant

4.2. Criação da Rede de Colaboração

Esta seção visa detalhar como a Rede de Colaboração foi criada, a partir das instâncias da Colab-Onto. Destaca-se que a rede é ponderada e não-direcionada, pois a colaboração não possui direção, ou seja, tanto o nó que envia quanto o nó que recebe colaboram no projeto de software da mesma forma. No entanto, as relações entre os nós possuem ponderações, já que a colaboração entre o par de nós pode ser quantificada conforme é realizada.

Devido à sua natureza e importância, os participantes foram considerados sendo os nós desta estrutura. Assim, a relação entre os participantes são as arestas. Cada aresta possui três diferentes tipos de pesos $\langle m, ac, oc \rangle$, que correspondem, respectivamente, ao número de atividades colaborativas, mensagens e objetos de cooperação, relacionados ao par de participantes. Dessa forma, os pesos das arestas representam as três dimensões do Modelo 3C.

Para cada par de participantes que trocarem mensagens ou estiverem envolvidos em uma mesma atividade colaborativa ou editarem um objeto de cooperação, o peso da aresta entre os dois participantes será incrementado. Assim, a matriz de adjacência é formada, como demonstrado na Tabela 4. Por exemplo, os pesos $\langle 3, 2, 1 \rangle$ na aresta entre os participantes P1 e P2, significam que P1 e P2 trocaram três mensagens, estiveram envolvidos em duas atividades colaborativas e editaram um objeto de cooperação. Por outro lado, se o par de participantes não trocar mensagens, não estiver envolvido em atividades colaborativas e não editar objetos de cooperação, eles não possuem relação colaborativa, logo não existe aresta.

Uma forma de representação gráfica de uma rede é considerá-la como um grafo (Figura 5). Esta representação possui as mesmas características computacionais da matriz de adjacência da Tabela 4.

4.3. Análise da Participação Colaborativa

Após a criação da rede de colaboração, procedeu-se com a aplicação das métricas sobre a rede para a análise da participação colaborativa dos membros de acordo com o Modelo 3C. As métricas aplicadas neste trabalho, são baseadas nas métricas Centralidade de Grau e Centralidade de Proximidade, adaptadas para o contexto da rede de colaboração em projetos de software.

A interface de apresentação dos resultados objetiva oferecer os resultados da aplicação das métricas sobre a rede de colaboração para o usuário do analisador de participação colaborativa em projetos de software. Essa interface, pode exibir os resultados em forma tabular, ranqueando os usuários por dimensão do Modelo 3C, de acordo cada

Tabela 3. Exemplo de uma instanciação da Colab-Onto

Conceito/Relação	Instância
Comunicação	
Participant	P1
	P2
Message	M1
	M2
	M3
send(Participant, Message)	send(P1, M1)
sendTo(Message, Participant)	M1 is sent to P2
send(Participant, Message)	send(P2, M2)
sendTo(Message, Participant)	M2 is sent to P1
send(Participant, Message)	send(P1, M3)
sendTo(Message, Participant)	M3 is sent to P2
Coordenação	
Activity	AT1
	AT2
belongs(Message, Activity)	M1 belongs AT1
	M2 belongs AT1
	M3 belongs AT2
Cooperação	
Cooperation Object	OC1
generates(Participant, CooperationObject)	P1 generates OC1

métrica de centralidade, ou, a partir do uso de gráficos. Entretanto, é importante que os dados sejam relacionados para que se permita a comparação dos mesmos, observando-se as intersecções entre as dimensões do 3C.

4.3.1. Métricas de Centralidade de Grau

No âmbito das colaborações em projetos de software, a Centralidade de Grau está relacionada à quantidade de vezes que um membro colaborou em relação aos demais. Assim, pode-se concluir que um membro influente na rede certamente estará conectado com vários outros membros.

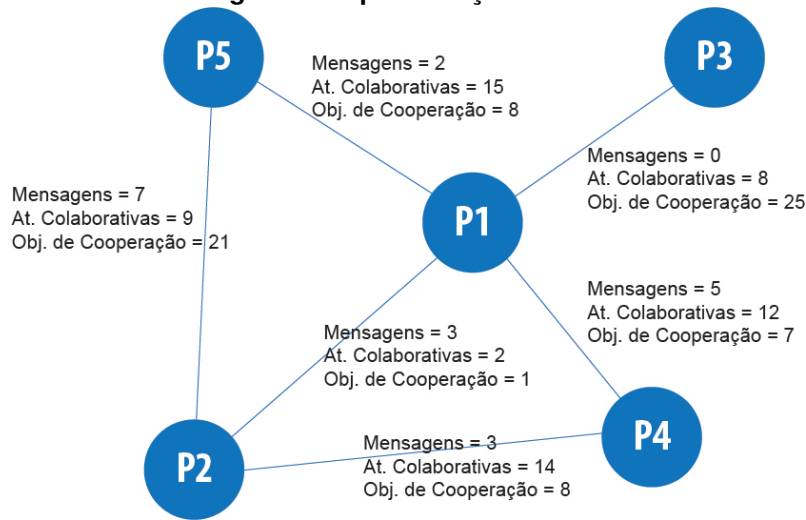
Neste trabalho, a métrica de Centralidade de Grau (C_G) foi definida em três contextos diferentes: mensagens (comunicação), atividades colaborativas (coordenação) e objetos de cooperação (cooperação). Assim, na Equação 3, a centralidade de grau de um participante p_i refere-se a todas as trocas de mensagens com outros participantes k_m . Dessa forma, um comunicador ativo certamente troca várias mensagens com outros participantes.

$$C_G(p_i) = k_m \quad (3)$$

Tabela 4. Exemplo de uma matriz de adjacência

	P1	P2	P3	P4	P5
P1	-	< 3, 2, 1 >	< 0, 8, 25 >	< 5, 12, 7 >	< 2, 15, 8 >
P2	< 3, 4, 1 >	-	< 4, 9, 2 >	< 3, 14, 8 >	< 7, 9, 21 >
P3	< 0, 8, 25 >	< 4, 9, 2 >	-	-	-
P4	< 5, 12, 7 >	< 3, 14, 8 >	-	-	-
P5	< 2, 15, 8 >	< 7, 9, 21 >	-	-	-

Figura 5. Representação da rede



Já na Equação 4, a centralidade de grau de um participante p_i , refere-se às atividades colaborativas em que ele estava envolvido com outros participantes k_{ac} . Assim, um participante que está envolvido em muitas atividades colaborativas certamente estará conectado com vários outros participantes por meio de atividades.

$$C_G(p_i) = k_{ac} \quad (4)$$

Por fim, a Equação 5 apresenta a centralidade de grau do participante p_i , que corresponde à quantidade de objetos de cooperação editados em comum com outros participantes (k_{oc}). Dessa forma, um participante que se destaca na cooperação provavelmente estará cooperando com vários outros membros.

$$C_G(p_i) = k_{oc} \quad (5)$$

4.3.2. Métricas de Centralidade de Proximidade

No contexto das colaborações em projetos de software, a Centralidade de Proximidade pode ser importante para analisar o quão determinado membro está colaborando junto com os outros membros da rede. Dessa forma, se um membro relaciona-se com vários outros membros do projeto, este estará numa posição central.

Neste trabalho, a métrica de centralidade de proximidade (C_P) foi definida em três contextos diferentes: mensagens (comunicação), atividades colaborativas (coordenação) e objetos de cooperação (cooperação). Assim, na Equação 6, a centralidade de proximidade na comunicação do i -ésimo participante (p_i), refere-se ao inverso da distância mínima das mensagens trocadas por todos os outros participantes atingíveis a partir do i -ésimo participante ($d(p_i, p_{j_m})$), isto é, se o i -ésimo participante (p_i) é um comunicador ativo, ele estará em uma posição central em relação aos outros comunicadores.

$$C_P(p_i) = \frac{1}{\sum_{p_{j_m} \in p_i} d(p_i, p_{j_m})}. \quad (6)$$

Para a coordenação, como mostra a Equação 7, a centralidade de proximidade do i -ésimo participante (p_i) se refere ao inverso da distância mínima das atividades colaborativas de todos os outros participantes atingíveis a partir de do i -ésimo participante (p_i), ($d(p_i, p_{j_{ac}})$), logo o i -ésimo participante (p_i) estará bem posicionado nessa dimensão se estiver próximo aos outros participantes envolvidos em atividades colaborativas.

$$C_P(p_i) = \frac{1}{\sum_{p_{j_{ac}} \in p_i} d(p_i, p_{j_{ac}})}. \quad (7)$$

Para a coordenação, como mostra a Equação 8, a centralidade de proximidade do i -ésimo participante (p_i), se refere ao inverso da distância mínima dos objetos de cooperação editados de todos os outros participantes atingíveis a partir do i -ésimo participante p_i , ($d(p_i, p_{j_{oc}})$), logo o i -ésimo participante (p_i) estará bem posicionado na dimensão da cooperação se estiver próximo de vários outros participantes que editaram os mesmos.

$$C_P(p_i) = \frac{1}{\sum_{p_{j_{oc}} \in p_i} d(p_i, p_{j_{oc}})}. \quad (8)$$

Dessa forma, com a aplicação das métricas de centralidade sobre a rede, é possível analisar a influência de membros mais ativos e quantificar a *participação colaborativa* de acordo com o Modelo 3C.

5. Prova de conceito

Esta seção apresenta uma prova de conceito para o analisador proposto. Para isso, escolheu-se a plataforma de controle de versão de arquivos Github, que possui apoio à Comunicação, à Coordenação e à Cooperação entre os desenvolvedores. A Comunicação pode ser observada na troca de comentários em um ambiente de rastreamento de erros. A Cooperação, por sua vez, pode ser definida como os arquivos editados em comum pelo grupo de usuários do repositório. Para a Coordenação, os *issues* e *pull requests* foram como atividades colaborativas. *Issue*, é o parâmetro no Github que os membros podem reportar problemas encontrados. Já *pull request*, é o parâmetro que os membros podem submeter melhorias.

5.1. Carga e Integração Semântica dos Dados

O mapeamento dos dados do Github para a Colab-Onto pode ser observado na Tabela 5. *Issues* e *pull requests* foram considerados atividades colaborativas e foram mapeados para

o conceito *Activity*. Os comentários de *issues* e de *pull requests* foram mapeados para o conceito *Message*. Os *commits files* possuem as mesmas propriedades do conceito *Cooperation Object*. Por fim, os *users* foram mapeados para *Participant*. Após o mapeamento para a ontologia é possível realizar a carga dos dados da ferramenta CASE.

Tabela 5. Mapeamento do Github para a Colab-Onto

Parâmetro do Github	Conceito da Colab-Onto	Dimensão do Modelo 3C
Issue / pull request	Activity	Coordenação
Issue comment / pull request comment	Message	Comunicação
Commit file	Cooperation Object	Cooperação
User	Participant	Comunicação, Coordenação e Cooperação

Para a extração dos dados colaborativos, o repositório escolhido foi o Battery Historian⁴, pois a quantidade de *issues*, *pull requests*, *commits* e comentários em *issues* e *pull request* foram considerados suficientes para análise. A carga dos dados do repositório ocorreu por meio da APF⁵ do Github e a quantidade total de dados extraídos é detalhada na Tabela 6.

Tabela 6. Quantidade total de dados extraídos

Parâmetro	Dados brutos	Dados integrados
Issue	35	Collaborative activities
Pull request	15	
Issue comment / pull request comment	112	Messages exchanged
Commit file	26	Cooperation objects
User	98	Participants

5.2. Criação da Rede

Após a carga dos dados, o Integrador Semântico gerou instâncias de classes da Colab-Onto para as propriedades mapeadas do Github. Assim, foi criada a rede de colaboração, a partir dos dados colaborativos semanticamente integrados.

Ressalta-se que a rede de colaboração obtida é formada por nós que contêm todos os usuários que colaboraram em *issues*, *pull requests* e *commits* no repositório, atuando em pelo menos uma dimensão do Modelo 3C.

5.3. Análise da Participação Colaborativa e Apresentação dos Resultados

Com a rede de colaboração criada, é possível a aplicação das métricas de centralidade para analisar se existe alguma influência dos usuários mais ativos entre as dimensões do Modelo 3C. Para a aplicação da centralidade de grau em cada dimensão, foram usadas

⁴Disponível em: <https://github.com/google/battery-historian>

⁵Disponível em: <http://api.github.com>

as equações definidas na Seção 4.3.1. Já a centralidade de proximidade foi calculada a partir das equações definidas na Seção 4.3.2. Para essa centralidade, o cálculo do caminho mínimo feito a partir do algoritmo de Dijkstra (Dijkstra, 1959).

Após a aplicação das métricas de centralidade, as tabelas da interface de apresentação são preenchidas. As tabelas da interface são separadas para cada dimensão do Modelo 3C, exibindo os usuários e seus respectivos *scores*. Como ilustram as Figuras 6, 7 e 8, as tabelas exibem os 10 primeiros colocados, segundo cada métrica e em cada dimensão do Modelo 3C. A Figura 6 apresenta os comunicadores mais ativos do repositório, já a Figura 7 exhibe os usuários mais ativos na dimensão da coordenação, enquanto a Figura 8 mostra os usuários que mais editaram artefatos de software em comum com outros usuários.

Figura 6. Recorte das medições para a Comunicação



Figura 7. Recorte das medições para a Coordenação

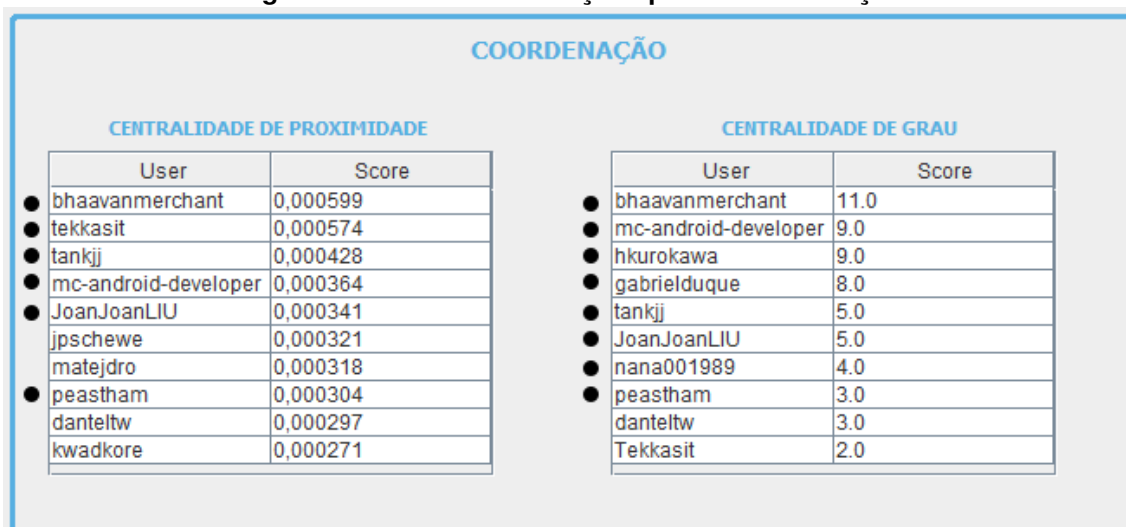
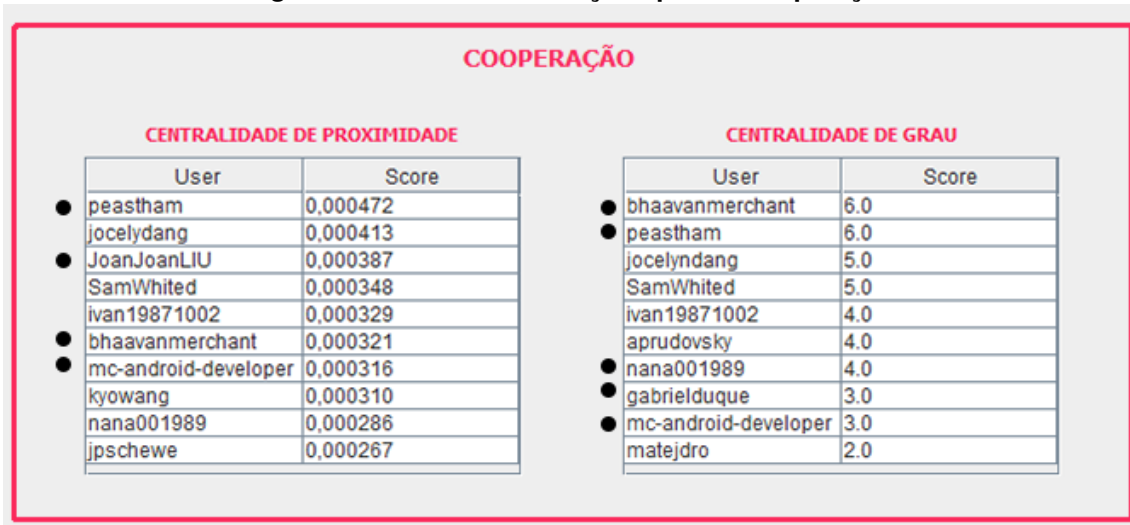


Figura 8. Recorte das medições para a Cooperação



Analisando as Figuras 6 e 7, observa-se que 87.5% dos usuários são comuns às duas dimensões. Esse resultado, se deve principalmente ao fato da Comunicação ter sido extraída a partir da Coordenação, isto é, os comentários de *issues* e *pull requests* são obtidos a partir dos *issues* e *pull requests*, respectivamente. Para a Figura 8, observa-se que cerca de 80% dos usuários mais ativos de acordo com a Cooperação são também aos usuários mais ativos nas dimensões da Comunicação e da Coordenação, como por exemplo, os usuários *peastham*, *JoanJoanLIU*, *bhaavanmerchant* e *gabrielduque*, entre outros. A Cooperação é obtida a partir dos arquivos editados e submetidos (*commit files* via *pull requests*, atividade colaborativa). Dessa forma, os usuários tendem a ser parcialmente comuns da Cooperação e Coordenação, já que os arquivos editados em comum são obtidos a partir *pull requests*. Entretanto, analisando a Comunicação entre os desenvolvedores, percebe-se que 57.5% também estão presentes na dimensão da Cooperação, como por exemplo, os usuários *bhaavanmerchant*, *peastham*, *mc-android-developer* e *gabrielduque*, entre outros. Assim, usuários mais ativos na dimensão da Comunicação também estão entre os mais ativos nas dimensões da Cooperação e Coordenação.

Os resultados obtidos indicam que usuários mais ativos tendem a ser ativos na Comunicação, Coordenação e Cooperação, pois a colaboração é cíclica (Gerosa, 2006), ou seja, usuários mais ativos ao comunicarem recebem *feedback* de suas ações e coordenam ações de outros usuários para cooperarem. Assim, os *scores* das Figuras 6, 7 e 8, é possível observar que vários usuários estão presentes nas três dimensões, como o *peastham*, *bhaavanmerchant*, *mc-android-developer*, *JoanJoanLIU*, *tankjj* e *gabrielduque*. Além disso, o usuário *bhaavanmerchant* apresentou o melhor *score* de acordo com a centralidade de grau nas três dimensões, somente na centralidade de proximidade da cooperação que o usuário ficou na 6ª posição. Dessa forma, é possível os resultados indicam que o *bhaavanmerchant* o colaborador mais ativo do repositório, seguido pelo *peastham*, *mc-android-developer*, *JoanJoanLIU*, *tankjj* e *gabrielduque*.

6. Conclusão

Este trabalho apresentou uma ferramenta capaz de integrar semanticamente dados colaborativos de fontes heterogêneas e analisar a participação colaborativa dos membros a partir da comunicação, coordenação e cooperação dos projetos de software. Para a integração semântica, utilizou-se uma ontologia do domínio da colaboração. Para a análise da participação, os dados colaborativos foram mapeados para essa ontologia e analisados utilizando métricas quantitativas de análise de redes sociais.

A análise sobre a rede foi feita através das métricas de centralidade de grau e centralidade de proximidade. Essas métricas analisam as relações entre os participantes do projeto e conseguem medir a influência da participação na comunicação, na coordenação e na cooperação. Para o repositório avaliado, 87.5% dos dez participantes com maiores *scores* estiveram presentes tanto na comunicação quanto na coordenação, ou seja, foram comunicadores ativos e estiveram envolvidos com várias atividades colaborativas. Outros 80% estiveram envolvidos tanto na comunicação quanto na cooperação.

Acredita-se que essa análise seja importante para que os líderes do projeto possam tomar decisões concisas ao identificar participantes com menor participação colaborativa durante o processo de construção de software, a fim de evitar problemas como atrasos e prejuízos financeiros.

Como trabalhos futuros planeja-se: investigar mecanismos de automatização da carga de dados de diversos repositórios de ferramentas CASE, já que o mapeamento para a Colab-Onto manualmente é custoso. Utilizar outras métricas de análises de redes sociais, para reforçar a influência dos participantes com relação às métricas de centralidade aplicadas neste trabalho e/ou gerar outras análises de influência de participação colaborativa. Além disso, técnicas estatísticas podem ser utilizadas a fim de correlacionar as dimensões do Modelo 3C.

Referências

- Balheiro, M., Junior, S. S., Pereira, L., and de Souza, C. (2007). Oss network: Um ambiente para estudo de comunidades de software livre usando redes sociais. pages 33–424. Experimental Software Engineering Latin America Workshop.
- Bettenburg, N. (2011). Mining development repositories to study the impact of collaboration on software systems. page 4. 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering.
- Braga e Silva, G. (2013). Colmeia, um analisador de colaborações baseado em métricas aplicáveis a informações semanticamente integradas em um ambiente de gcs. page 185. Instituto Tecnológico de Aeronáutica - ITA.
- Curtis, B., Krasner, H., and Iscoe, N. (1988). A field study of the software design process for large systems. pages 1268–1287. Communications of the ACM, ACM, New York, NY, USA.
- da Silva, J. T., Wiese, I. S., Steinmacher, I., and M, A. G. (2013). Minerall: Uma ferramenta para extração e mineração de dados de repositórios de software livre. page 6. 12th Free Software Workshop (WSL 2011).
- de Souza, C., Froelich, J., and Dourish, P. (2005). Seeking the source: Software source code as a social and technical artifact. pages 197–206. ACM Conference on Supporting Group Work.

- de Souza, E. F. (2014). Knowledge management applied to software testing: An ontology based framework. page 214. Tese (PhD in Computer Science) - Instituto Nacional de Pesquisas Espaciais - INPE.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. volume 1, pages 269–271. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Ellis, C. A., Gibbs, S. J., and Rein, G. L. (1991). Groupware: Some issues and experiences. Number 1, pages 39–58.
- Falbo, R. (2014). Sabio: Systematic approach for building ontologies. page 14.
- Freeman, L. (1979). Centrality in social networks: Conceptual clarification. pages 215–239. Social Networks.
- Fuks, H., Raposo, A. B., Gerosa, M. A., Pimentel, M., and Lucena, C. J. (2007). The 3c collaboration model. In *The encyclopedia of e-Collaboration*, pages 637–644. Information Science Reference.
- Gerosa, M. A. (2006). Desenvolvimento de groupware componentizado com base no modelo 3c de colaboração. pages 72–84. Tese (PhD in Computer Science) - Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO.
- Gerosa, M. A., Pimentel, M., D., F., C.G., B., A.B., R., Fuks, H., and C.J.P., L. (2005). Componentes baseados no modelo 3c para o desenvolvimento de ferramentas colaborativas. page 410. Anais do 5º Workshop de Desenvolvimento Baseado em Componentes - WDBC 2005.
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. volume 5, pages 199–220. Academic Press Ltd., London, UK, UK.
- Izza, S. (2009). Integration of industrial information systems: from syntactic to semantic integration approaches. pages 1–57. Enterprise Information Systems.
- Jiang, J., Zhang, L., and Li, L. (2013). Understanding project dissemination on a social coding site. page 12. 20th Working Conference on Reverse Engineering (WCRE).
- Newman, J. (2003). The structure and function of complex networks.
- Rocha, L. M. A., Silva, T. H. P., and Moro, M. M. (2016). Análise da contribuição para código entre repositórios do github. page 6. Simpósio Brasileiro de Banco de Dados - SBBD.
- Schneider, R. L. (2011). Um sistema de gerência cooperativa de configuração de software. page 117. Dissertação(Mestrado em Engenharia de Sistemas e Computação) - Universidade Federal do Rio de Janeiro - COPPE/UFRJ, Rio de Janeiro.
- Scott, J. (2000). Social network analysis. New York: SAGE Publications Ltd.
- Souza, C. R. B., Sharp, H., Singer, J., Cheng, L.-T., and Venolia, G. (2009). Guest editors' introduction: Cooperative and human aspects of software engineering. pages 17–19. IEEE Software.
- Thung, F., Bissyand, T. F., David, and Jiang, L. (2013). Network structure of social coding in github. page 4. 17th European Conference on Software Maintenance and Reengineering.
- Wasserman, S. and Faust, K. (2007). Social network analysis: Methods and applications. Cambridge, UK and New York: Cambridge University Press.