UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Hugo Guércio Fernandes

sSECO-Process: Avaliando a Dimensão Social em Ecossistemas de Software

Juiz de Fora

2018

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Hugo Guércio Fernandes

sSECO-Process: Avaliando a Dimensão Social em Ecossistemas de Software

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Victor Ströele de Andrade Menezes
Coorientador: José Maria Nazar David

Juiz de Fora
2018

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Fernandes, Hugo Guércio.
sSECO-Process: avaliando a dimensão social em ecossistemas de software / Hugo Guércio Fernandes. -- 2018.
135 f. : il.

Orientador: Victor Ströele de Andrade Menezes
Coorientador: José Maria Nazar David
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Ciência da Computação, 2018.

1. Ecossistemas de Software. 2. Redes Complexas. 3. Colaboração. I. Ströele de Andrade Menezes, Victor, orient. II. Maria Nazar David, José, coorient. III. Título.

Hugo Guércio Fernandes

sSECO-Process: Avaliando a Dimensão Social em Ecossistemas de Software

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 14 de Setembro de 2018.

BANCA EXAMINADORA

Prof. D.Sc. Victor Ströele de Andrade Menezes - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. José Maria Nazar David - Coorientador
Universidade Federal de Juiz de Fora

Profa. D.Sc. Regina Maria Maciel Braga
Universidade Federal de Juiz de Fora

Prof. D.Sc. Leonardo Guerreiro Azevedo
Universidade Federal do Estado do Rio de Janeiro

Resumo

A abordagem de Ecossistemas de Software(SECO) possui crescente adoção pelas comunidades de desenvolvimento e indústria de software. Esta abordagem pode gerar vantagens mas também pode agregar complexidade adicional na gestão dos recursos, que por consequência pode afetar a rede de produção de software. Ao observar SECOs, podemos vislumbrar três dimensões, sendo elas de negócios, técnicas e sociais. A dimensão social tem foco nos *stakeholders* e em como eles interagem com as outras dimensões. Diante disso, este trabalho apresenta o sSECO-Process, um processo para análise da dimensão social de Ecossistemas de Software, apoiado por técnicas de Redes Complexas, que permitem apresentar os relacionamentos existentes no SECO através de visualizações e da utilização de métricas pertinentes às redes. Este processo serviu como base para a análise de diferentes SECOs, e deu origem a tecnologias que apoiam a coleta, tratamento, armazenamento e processamento dos dados, que auxiliarão cientistas em novas pesquisas relacionadas ao desenvolvimento de software. Esta dissertação também apresenta medidas, que tem como objetivo mensurar o nível de colaboração dos usuários nos projetos, e, formas de relacionar os usuários quando não existem registros de interação entre eles Foi realizada uma avaliação preliminar, com dados reais e apoio de especialistas, com o intuito de avaliar o processo desenvolvido e, após a avaliação preliminar, o processo é estendido com o intuito de melhorar e complementar as atividades executadas durante a análise da dimensão social de SECOs.

Palavras-chave: Ecossistema de Software, Redes Complexas, Colaboração

Abstract

The Software Ecosystems approach (SECO) is increasingly adopted by development communities and software industry. This approach can provide advantages , but can also add additional complexity in resource management, which can consequently affect the software supply network. By observing SECOs, we can see through three dimensions, being business, technical and social. The social dimension focuses on stakeholders and how they interact with other dimensions. This paper presents sSECO-Process, a process for analyzing the social dimension of Software Ecosystems, supported by Complex Networks techniques, which allow the presentation of existing SECO relationships' through visualizations and use of metrics pertinent to complex networks. This process served as a basis for the analysis of different SECOs, and gave rise to technologies that support the collection, processing, storage and processing of data, which will aid scientists in new research related to software development. A preliminary evaluation, with real data and specialist support, is carried out to evaluate the process developed and, after the preliminary evaluation, the process is extended with purpose of improving and complementing the activities performed during the analysis of the social dimension of SECOs.

Keywords: Software Ecosystems, Complex Networks, Collaboration.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Ao estudar Ecosistemas de Software (SECO) percebe-se que os envolvidos se relacionam de diferentes maneiras, e essas relações afetam de maneira significativa todos os aspectos do ecossistema. É possível observar SECOs, por múltiplas perspectivas, e por isso existe uma divisão dessas perspectivas. (CAMPBELL & AHMED, 2010) apresenta uma divisão dessas perspectivas, dando uma visão tri-dimensional das dimensões que compõem o SECO, sendo elas de negócio, técnica e social. Essas dimensões estão presentes em estudos subsequentes BARBOSA *et al.* (2013); MANIKAS & HANSEN (2013), mostrando a aceitação da comunidade quanto a esta forma de segmentação. A literatura possui diferentes definições para os SECOs, e neste trabalho foi adotada a definição de LUNGU *et al.* (2010), que afirma que um SECO é uma coleção de projetos de software que são desenvolvidos e evoluem em conjunto em um mesmo ambiente.

BARBOSA *et al.* (2013) realizaram um mapeamento sistemático com uma perspectiva tridimensional, abordando as dimensões técnicas, de negócios e social. A dimensão técnica tem a plataforma como foco, e os estudos avaliados aparecem diretamente relacionados a evolução e arquitetura de software além de sistemas operacionais e linhas de produto de software. A dimensão de negócios tem o fluxo de conhecimento como foco, avaliando artefatos, recursos e informações através do negócio, com foco na inovação e planejamento estratégico, aparecendo diretamente nas atividades de modelagem, negócios e co-inovação. A dimensão social tem como foco os *stakeholders* de um SECO e, através do estudo das suas interações, pode auxiliar o SECO a melhorar o fluxo de conhecimento, o senso de comunidade, reconhecimento de pares para trabalho, aprendizado ou adoção de novas tecnologias.

Em MANIKAS (2016), uma revisão da literatura, realizada anteriormente (MANIKAS & HANSEN, 2013), foi atualizada com o objetivo de manter o conhecimento sobre a evolução dos estudos sobre ecossistemas. Assim, como BARBOSA *et al.* (2013), os artigos foram classificados em três categorias: negócios e gestão, engenharia de software e relacionamentos. Essa divisão deixa claro que os relacionamentos presentes em um SECO

são aspectos muito importantes que afetam o ecossistema, justificando a pesquisa para que os relacionamentos presentes no SECO possam de fato contribuir com seu objetivo.

Os ambientes que suportam os ecossistemas de software devem prover interações entre as entidades presentes no ecossistema que podem ser entre os atores envolvidos, entre os artefatos presentes no ambiente comum ou relacionando os atores e conhecimentos presentes no ecossistema (DOS SANTOS, 2016).

Com a maior popularidade dos SECOs, diferentes organizações iniciam a transição de outras abordagens de desenvolvimento para a abordagem SECO para que elas aproveitem as vantagens oferecidas pela estratégia. BARBOSA *et al.* (2013) tenta sumarizar algumas das vantagens encontradas na literatura nos artigos que compreendem o mapeamento sistemático realizado dando destaque ao sucesso, co-evolução, redução de custos e diversas outras vantagens.

De acordo com BOSCH (2009) abordagem de SECO pode ser tida como uma vertente da abordagem de *Software Product Lines* (SPL). O grande sucesso da abordagem SPL implica na contínua expansão da base de colaboradores, fazendo com que os limites organizacionais não sejam tão desejáveis. O aumento na base de colaboradores pode auxiliar a organização a complementar os graus de customização e eficiência na resposta às necessidades dos usuários da plataforma. A principal diferença entre as abordagem de SECO e SPL está nos limites que a organização estabelece para a participação dos usuários.

De acordo com BOSCH (2009) em uma organização típica de SPL que começa a expandir seus horizontes, para quebrar o limite intra-organizacional, e utilizar a abordagem de ecossistemas teríamos quatro níveis de desenvolvedores, sendo eles: internos, estratégicos, indiretos e independentes. Cada um desses níveis de desenvolvedores deve colaborar entre si para garantir que os usuários continuem tendo suas necessidades atendidas. A colaboração também deve existir entre diferentes níveis para manter todos os desenvolvedores alinhados com a proposta do ecossistema.

Independente dos níveis dos desenvolvedores que colaboram e das dificuldades existentes na colaboração é um fato que trabalho em grupo pode produzir resultados melhores quando comparados aos resultados oferecidos por atores trabalhando individu-

almente SCHULZE *et al.* (2016). Mesmo o melhor colaborador não consegue se sobrepor ao conhecimento coletivo, e por isso grupos de colaboradores são importantes. Para que exista um grupo, é necessário que um conjunto de pessoas possuam um objetivo comum, que estimule a interação entre os membros e possibilite a colaboração para criação de novos artefatos.

Neste contexto, a colaboração permite que o grupo combine diferentes capacidades, conhecimentos e perspectivas através da ótica de cada um dos seus membros. Desta forma, a pluralidade de características entre os membros do grupo enriquece a discussão de ideias, técnicas e abordagens para soluções teóricas e práticas. Esta combinação de características também é alterada de acordo com o tempo, onde os membros do grupo passam a identificar de maneira mais clara quais seus pontos fortes de contribuição para o grupo assim como quais os membros podem auxiliar, de maneira mais eficiente, nas diferentes atividades.

A evolução das comunidades de desenvolve anda em paralelo com a evolução dos sistemas de software. O estudo da evolução do desenvolvimento de software tem sido objeto de estudos da academia ao longo dos anos. Esse esforço é explicitado por trabalhos na literatura que tem como objetivo entender a evolução de software e que perduram por grandes períodos de tempo. Nesta linha, temos em destaque Lehman, que busca, a cinco décadas, formas de capturar e auxiliar novos cientistas nessa complexa atividade de estudo do software. Entre seus trabalhos mais notáveis, temos (LEHMAN, 1996; LEHMAN *et al.*, 1985, 1997) que apresentam leis de evolução de software. Uma das leis definidas pelos autores, é a lei do sistema de *feedback*, que teve sua primeira enunciação em 1974 mas apenas 22 anos depois foi declarada como uma lei, segundo os autores.

Também em destaque temos (RAMIL & LEHMAN, 2000), onde são propostos indicadores para identificar a evolução de software, através da observação da criação e edição de módulos e subsistemas. Em (PALOMBA *et al.*, 2017) as atividades de evolução de software estão em foco, e são observadas pela perspectiva social. A análise foi realizada através do estudo do débito social, identificado através de "*community smells*", que podem ter origens sócio-técnicas, como alta formalidade no processo ou por padrões de comportamento. A identificação do débito social também pode ter origens sociais,

como por exemplo, recorrência de membros do time agindo de maneira condescendente ou desistindo dos projetos. O relacionamento entre os débitos técnicos e sociais também é abordado. STOREY et al. (2014) utilizam informações em canais de mídia para auxílio no relacionamento entre aspectos técnicos e sociais, observando a utilização de tais canais a partir de 1968 até 2014.

O estudo dos aspectos evolutivos do desenvolvimento de software, mesmo que amplamente explorado, é uma fonte de motivação, visto que os softwares continuarão sua evolução, e, os interessados continuarão suas pesquisas para aumentar a compreensão do processo de desenvolvimento de software. A clareza cada vez maior da importância das pessoas no desenvolvimento de software também funciona como fator motivador, e com isso o relacionamento entre as pessoas e o desenvolvimento de software se alinha e funciona como incentivo.

Alinhados com a evolução do software e, da forma de construí-lo, temos a abordagem de SECO, que despertou interesse das comunidades envolvidas com software, que é refletida na adoção da abordagem pela comunidade e pela crescente quantidade de trabalhos na literatura referente aos SECOs. Tendo isso em vista, temos que a área, mesmo que ainda recente, tem grandes possibilidades de desenvolvimento e diferentes desafios para sua evolução. Esse crescimento no interesse da comunidade é refletido em uma série de estudos da literatura que tem como objetivo fornecer um panorama e auxiliar novos cientistas a contribuir com o campo de pesquisa (BARBOSA *et al.*, 2013; MANIKAS & HANSEN, 2013; MANIKAS, 2016; NASSERIFAR, 2016; ALVES *et al.*, 2017; DE LIMA FONTÃO *et al.*, 2015).

O presente trabalho, ao observar os ecossistemas através do tempo, pretende capturar como ocorre a evolução dos SECOs, e, a partir das medidas e indicadores identificados na literatura, fornecer subsídio aos colaboradores e mantenedores dos SECOs para tomada de decisão em relação ao acompanhamento dos colaboradores das comunidades que compõe os SECOs.

1.1 Problema

A abordagem de SECO visa proporcionar vantagem estratégica aos grupos que a adotam, expandindo limites organizacionais e dando ênfase no aspecto social da atividade de desenvolvimento de software. Além das vantagens oferecidas pela abordagem de ecossistemas, também temos aumento na complexidade de algumas atividades, como a gestão dos recursos humanos.

De acordo com IANSITI & RICHARDS (2006), um ecossistema de software é saudável caso seja produtivo para os atores em torno deles, sendo robusto e possibilitando criação de nichos. A dimensão social está diretamente relacionada à saúde de um ecossistema, pois os projetos que compõem ecossistemas de software só evoluem com contribuições constantes da comunidade. Caso os projetos tenham dificuldades em atrair e reter novos colaboradores a comunidade pode se enfraquecer levando à estagnação e dificuldades para a manutenção e evolução do SECO.

Desta forma, mecanismos que auxiliem o grupo no entendimento da sua evolução e dos participantes que fazem parte desta história se fazem necessários para subsidiar decisões estratégicas, como, por exemplo, ajustes nos níveis de abertura do ecossistema. Os níveis de abertura estão relacionados aos pré-requisitos para que um indivíduo possa colaborar com o SECO como, por exemplo, a aprovação de um membro do grupo ou a satisfação de algum critério de reputação.

Observando os ecossistemas pela perspectiva da dimensão social, temos que novos entrantes necessitam de um tempo de adaptação por não terem uma visão dos artefatos e dos outros colaboradores que participam do projeto. Neste contexto, surge o problema a ser tratado neste trabalho: *como facilitar a visibilidade dos aspectos evolutivos em termos de projeto, destacando os relacionamentos sociais existentes no ambiente comum do ecossistema, de forma que haja benefícios para os stakeholders.*

1.2 Questão de Pesquisa

A partir da motivação e problema apresentados, a questão de pesquisa deste trabalho é: *A concepção de um processo pode auxiliar cientistas a utilizar métricas disponíveis na*

literatura, em conjunto com técnicas de redes complexas, para estudar comunidades que compõem Ecossistemas de Software, auxiliando a análise e tomada de decisão em relação aos colaboradores e projetos que compõem o SECO?

1.3 Objetivos

O principal objetivo desta pesquisa é a proposição do sSECO-Process, um processo que auxilia a observação da dimensão social de Ecossistemas de Software. O processo deve ajudar quem o utilize a entender melhor os aspectos sociais presentes em um SECO, e para isso foi considerado o domínio de *Open Source Software* (OSS). Este domínio foi escolhido por possibilitar o estudo de maneira clara, visto que os dados estão disponíveis para aplicação do processo proposto e reprodução por outros cientistas sem dificuldades na obtenção dos dados. As análises foram realizadas com o auxílio de métricas da literatura e da perspectivas de redes complexas, utilizada com o intuito de destacar os relacionamentos sociais.

Para atingir esse objetivo principal emergem objetivos específicos. Esses objetivos compõem algumas etapas que podem auxiliar na conquista do objetivo principal. Além disso, fornecem metas específicas que, de forma conjunta, apontam para o resultado esperado. Dentre os objetivos específicos temos:

- Modelar redes complexas que explicitem as relações sociais existentes no ambiente compartilhado de um SECO.
- Capturar a evolução da dimensão social dos projetos que compõem um SECO.
- Identificar os principais colaboradores dos projetos que compõem um SECO.

Este trabalho também tem como objetivo avançar as pesquisas relacionadas aos Ecossistemas de Software.

1.4 Organização

Este capítulo apresentou a Introdução assim como problema, questão de pesquisa, objetivos, motivação, assim como a organização da dissertação. Capítulo 2 apresenta a fun-

damentação teórica que respalda as discussões nos capítulos seguintes. No Capítulo 3 o SECO-Process é estendido para aumentar a clareza das atividades, facilitando sua execução e adicionando novas atividades não compreendidas no processo inicial. No Capítulo 4 é realizada uma avaliação do SECO-Process. No Capítulo 5 são apresentados os principais trabalhos relacionados a esta dissertação identificados durante seu desenvolvimento, destacando características que os relacionam e diferenciam deste. Por fim, no Capítulo 6, são apresentadas as considerações finais do trabalho. A dissertação também apresenta um apêndice com a primeira versão do sSECO-Process e uma avaliação preliminar. A Figura 1.1 apresenta uma síntese da organização desta dissertação.

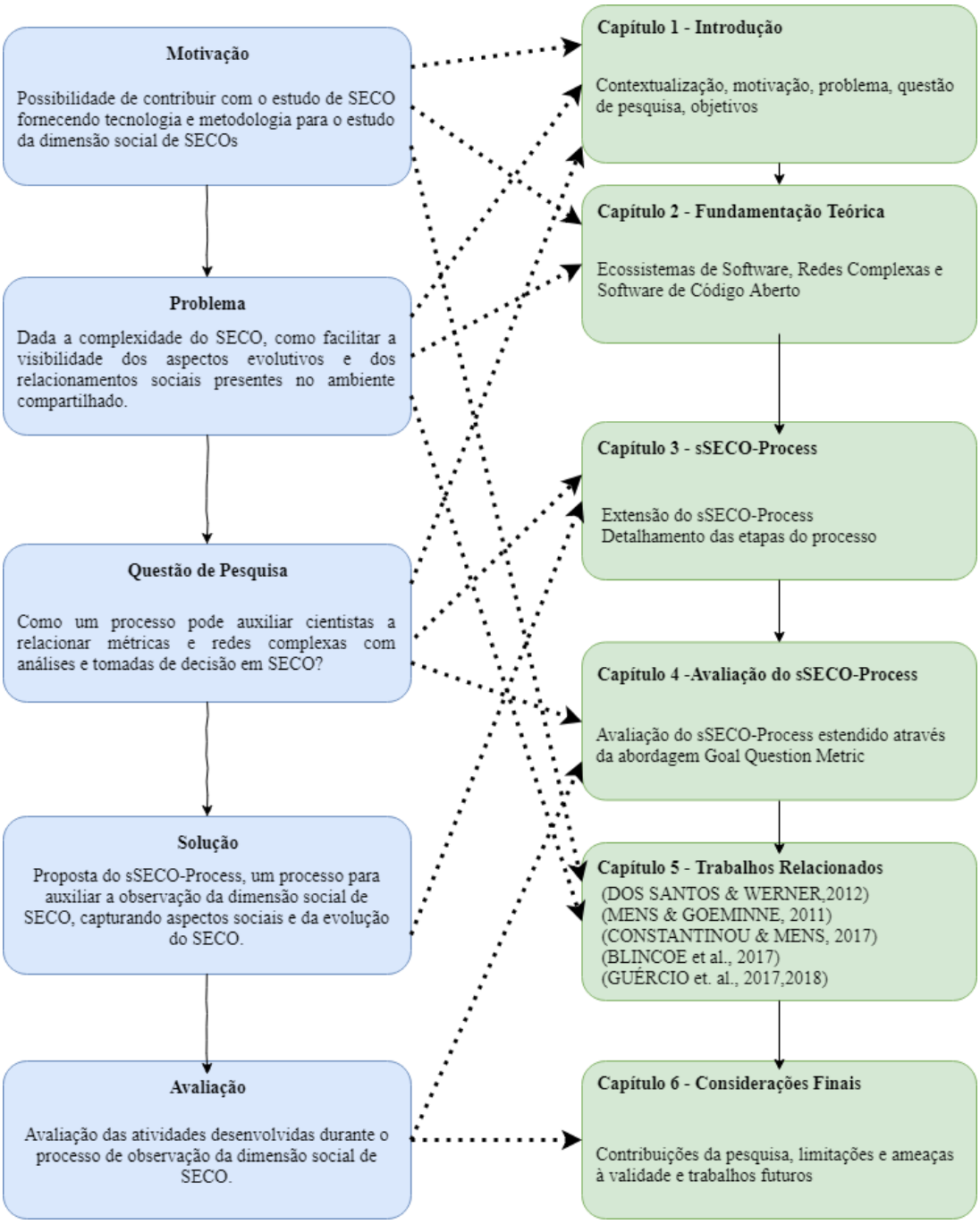


Figura 1.1: Organização da Dissertação.

2 Fundamentação Teórica

Neste capítulo, são apresentados os temas que compreendem a base teórica presente neste trabalho. O primeiro conceito apresentado é o de Ecossistemas de Software, onde são destacadas as motivações e origem da abordagem. Também são apresentadas algumas definições da literatura assim como fatores positivos proporcionados pela adoção dessa perspectiva, aplicações e o estado da arte. Em seguida, é discutido como os Softwares de Código Aberto(OSS) se relacionam com a perspectiva de SECO assim como benefícios e oportunidades, para a academia e organizações privadas. Por fim, os principais conceitos de Redes Complexas e Software de Código Aberto são apresentados por constituir parte relevante da solução apresentada neste trabalho seguida pelas considerações finais do capítulo.

2.1 Ecossistemas de Software

O desenvolvimento de software se molda às necessidades e oportunidades oferecidas pelo ambiente. Atualmente, as organizações estão seguindo modelos de comportamento que facilitam os relacionamentos entre diferentes companhias. Esta postura é um reflexo do modelo de inovações, visto que hoje em dia estas inovações são originadas através de diferentes participantes do mercado, fazendo com que essas companhias aumentem as dependências entre si e entrem em um estado de co-evolução através das inovações originadas dos seus relacionamentos BARBOSA *et al.* (2013). Um exemplo de SECO pode ser compreendido como as lojas de aplicativos para dispositivos móveis como a Play Store[1], que representa uma loja de aplicativos que são disponibilizados na plataforma, onde os aplicativos são produzidos por diferentes companhias que utilizam do ambiente compartilhado para distribuir suas aplicações.

Alguns fatores podem ter contribuído para esse direcionamento estratégico das empresas em aumentar a cooperação entre si. De acordo com BOSCH (2009), a crescente

[1]https://play.google.com/store

demanda de soluções customizadas e específicas força os produtores de software a recorrer a terceiros, como outras organizações desenvolvedoras de software ou desenvolvedores, para que eles contribuam e adicionem novas funcionalidades ao produto (VAN DEN BERK *et al.*, 2009).

O aumento de tecnologias que dão suporte ao desenvolvimento distribuído (GUÉRCIO *et al.*, 2017) também pode influenciar de maneira positiva o relacionamento entre as organizações e entidades que ultrapassam os limites organizacionais. Esse suporte à comunicação pode atuar como um estímulo, estreitando relações entre organizações, grupos e indivíduos que possuem objetivos alinhados ou complementares.

Nesse contexto emergem os Ecossitemas de Software (SECO), uma nova perspectiva inspirada nos ecossistemas naturais e de negócios. Essa perspectiva considera o software e sua dependência com componentes externos, seus colaboradores e as diferentes formas de interação entre os envolvidos. Nesta seção, algumas definições de SECO serão apresentadas assim como algumas de suas origens e suas principais características.

2.1.1 Definições

A abordagem de ecosssitemas é recente, e ainda não existe um consenso sobre qual a definição que representa melhor a abordagem. Desta forma, revisões recentes da literatura MANIKAS (2016); MANIKAS & HANSEN (2013); BARBOSA *et al.* (2013) tentam englobar e apresentar aspectos centrais e fundamentais da abordagem, dando norte a novos estudos. Na revisão da literatura realizada por MANIKAS & HANSEN (2013), uma de suas conclusões é a falta de um consenso sobre como definir um SECO. Essa conclusão surgiu da primeira questão de pesquisa formulada na revisão sistemática, que tem como objetivo capturar como o termo *software ecosystem* é definido. Como resultado, uma quantidade de trabalhos não define o SECO de maneira explícita (40 trabalhos), e alguns definem de acordo com suas próprias palavras (9 trabalhos). Outros trabalhos (48 trabalhos) utilizam uma das quatro definições a seguir.

MESSERSCHMITT & SZYPERSKI (2005), apresentaram a primeira definição de SECO na literatura no ano de 2005:

> "Traditionally, a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships." (MESSERSCHMITT & SZYPERSKI, 2005)

A definição mais citada na literatura foi apresentada em JANSEN *et al.* (2009), que apresentam o termo como:

> "We define a software ecosystem as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently under-pinned by a common technological platform or market and operate through the exchange of information, resources and artifacts." (JANSEN *et al.*, 2009)

BOSCH (2009) apresenta a segunda definição mais citada, sendo que em um trabalho posterior (BOSCH & BOSCH-SIJTSEMA, 2010) a definição foi ajustada. Neste trabalho, os autores afirmam que um SECO pode ser definido como:

> "software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs.". (BOSCH & BOSCH-SIJTSEMA, 2010)

Em (LUNGU *et al.*, 2010), uma definição diferente é apresentada, com um enfoque na parte técnica dos ecossistemas, onde os autores definem da seguinte forma:

> "A software ecosystem is a collection of software projects which are developed and evolve together in the same environment." (LUNGU *et al.*, 2010)

Como esperado, todas as definições possuem características comuns, mas observam a abordagem sobre diferentes perspectivas. As definições de LUNGU *et al.* (2010) e MESSERSCHMITT & SZYPERSKI (2005) possuem um caráter mais técnico, dando destaque aos projetos e produtos. JANSEN *et al.* (2009) e BOSCH & BOSCH-SIJTSEMA (2010) apresentam em suas definições os aspectos de negócios e sociais, deixando explícito que tais relacionamentos ultrapassam o nível técnico. Em destaque, todas as definições evidenciam a importância dos relacionamentos, mostrando que a conectividade entre as diversas perspectivas é ponto chave dos ecossistemas.

Neste trabalho, foi utilizada a definição de SECO proposta por LUNGU *et al.* (2010), que apresenta de maneira sucinta a forma de identificação do ecossistema, como

esse conjunto de projetos, relacionando os responsáveis pelo desenvolvimento e dando destaque à evolução no ambiente compartilhado.

Com o objetivo de consolidar as noções de SECO, BOSCH (2009) propõe uma taxonomia organizando os Ecossistemas através de duas dimensões. A primeira dimensão diz respeito a categorias em que os Ecossistemas estão reunidos através dos níveis de abstração em que o SECO existe. O autor atribui a esta dimensão três possíveis níveis, sendo eles: (i) sistema operacional: primeira categoria onde os SECOs foram explicitamente identificados e gerenciados. Essa categoria possui como característica a independência de domínio e ferramentas de apoio ao desenvolvimento no ecossistema para simplificar a adoção pela comunidade de desenvolvedores, que será responsável pelas customizações; o sucesso é definido pela quantidade de aplicações bem sucedidas do ecossistema assim como a quantidade de usuários; (ii) aplicações de software: como oposto da primeira categoria, esta é específica de domínio e, normalmente, tem início a partir de uma aplicação bem sucedida no ambiente de negócios sem o suporte que espera-se de um ecossistema. Esse início constrói uma larga base de clientes que possibilita o início deste tipo de ecossistema. Uma característica importante é a extensão das funcionalidades providas pela plataforma além da transparência para os consumidores da aplicação; o sucesso é simplificar as contribuições através das colaborações de terceiros, mantendo a facilidade para desenvolvimento, implantação e integração; e (iii) programação para usuário final: nesta categoria, a plataforma apoia os desenvolvedores que não necessariamente possuem conhecimentos sólidos na parte técnica, sendo baseados em uma linguagem específica de domínio, normalmente gráfica ou textual, tendo como características a construção de uma solução a partir de componentes pré-existentes, ao invés da criação de uma funcionalidade completamente nova; o sucesso desta categoria é definido pela quantidade de valor gerada pela plataforma e seus usuários para atender às necessidades específicas de cada um.

A segunda dimensão da taxonomia é referente à plataforma computacional em que o SECO está posicionado. Essa classificação é realizada de acordo com a infraestrutura, sendo dividida em (i) *desktop*, (ii) *web* e (iii) *mobile*.

2.1.2 Origens

O estudo de SECO foi originado da abordagem de Linhas de Produto de Software, ou *Software Product Lines* (SPL). Essa abordagem foi a mais bem sucedida em promover a reutilização de artefatos de software em um ambiente intra-organizacional (BOSCH, 2009). Essa abordagem possibilitou que as organizações aumentassem o valor de seus produtos através do aumento na variabilidade do produto oferecido com um custo de desenvolvimento baixo, visto que uma determinada funcionalidade poderia ser compartilhada entre diferentes consumidores.

Com o sucesso desta abordagem, o escopo dos envolvidos se torna cada vez maior atingindo uma grande porção da organização, mas as organizações perceberam que não existe um motivo que inviabilize essa expansão para fora dos limites intra-organizacionais. Ao decidir que o limite de contribuições é maior que o limite interno da organização, fazendo com que a plataforma esteja disponível fora dos limites organizacionais, a empresa realiza a transição da abordagem SPL para a abordagem de SECO.

A Figura 2.1 mostra quatro gerações de Reuso de Software (DOS SANTOS, 2016), sendo elas: (1) sistemas monolíticos: software desenvolvido através da integração de rotinas; (2) sistemas baseados em componentes: o aspecto técnico era amplamente explorado com a intenção de reutilização dos componentes; (3) linhas de produto: o aspecto de negócio começa a ser considerado e a abordagem de SPL se consolida; e (4) ecossistemas: o aspecto social ganha atenção e o cenário de desenvolvimento de código aberto assim como sistemas de sistemas e co-evolução.

Em (MENS & GROSJEAN, 2015), é dado destaque aos relacionamentos com ecossistemas biológicos e afirmam que a comparação com SECO pode auxiliar na produção de novas estratégias para aumentar a efetividade e resiliência. Um exemplo desta comparação é apresentado na Figura 2.2, que realiza a comparação entre um ecossistema biológico e um ecossistema de software visto a partir da sua perspectiva social.

De acordo com os autores, os ecossistemas de software podem ser comparados com ecossistemas biológicos em duas formas, sendo uma social e outra técnica. A comparação pela perspectiva técnica trata os componentes de software como as espécies biológicas em um ecossistema. O ecossistema compreende os componentes de hardware e software

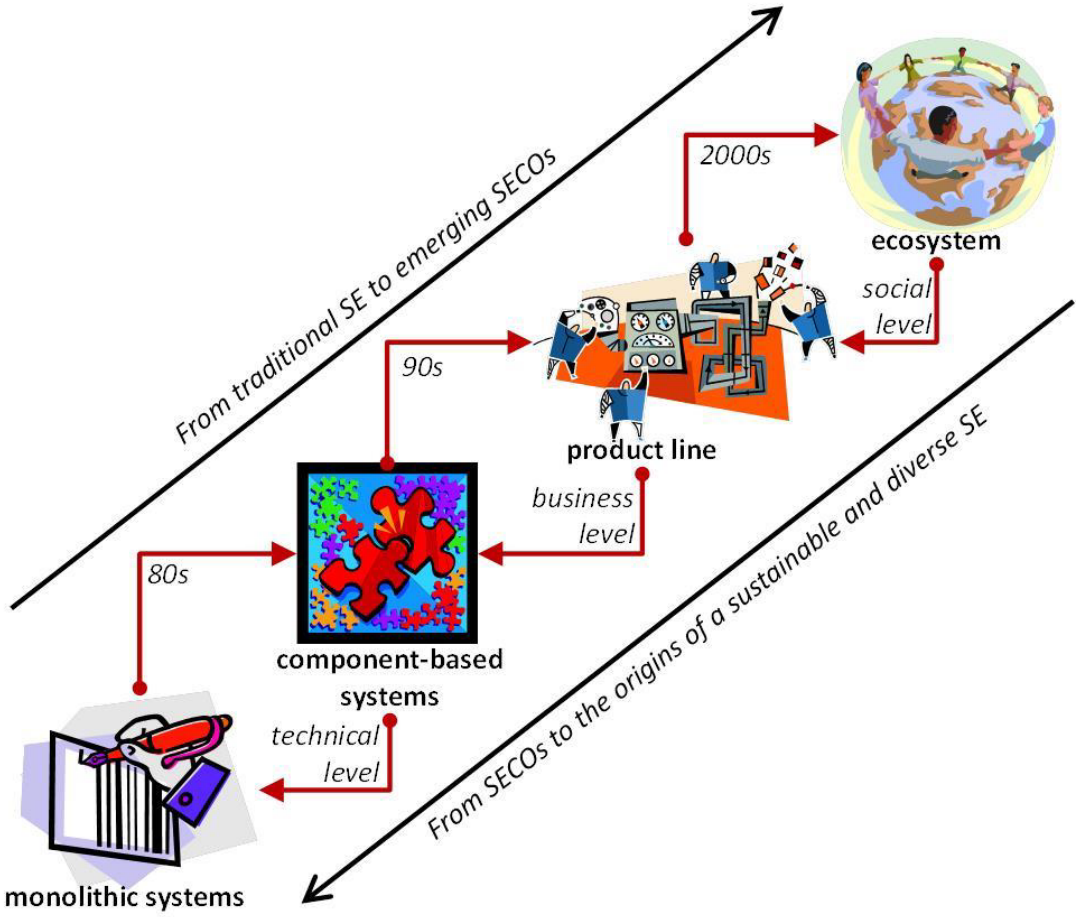


Figura 2.1: Trajetória de Ecossistemas através da Reutilização de Software em quatro gerações (DOS SANTOS, 2016).

necessários para desenvolvimento, teste, implantação e utilização, e, assim como em uma rede trófica[2], alguns componentes podem atuar como produtores que são consumidos por outros componentes de software.

A comparação a partir de uma perspectiva social faz a analogia das espécies com os indivíduos que contribuem no ecossistema. A partir dessa visão social, os produtores e consumidores são os envolvidos que interagem através do ambiente comum compartilhando o conjunto de recursos comuns, como testes, documentação, report de bugs e outros.

A questão de resiliência do ecossistema também tem uma forte analogia. Esta habilidade do ecossistema de se recuperar de situações que causam distúrbio no seu equilíbrio é fortemente relacionada a biodiversidade dos componentes do ecossistema. Considerando

[2]Uma rede trófica tem como objetivo representar as interações tróficas entre diferentes espécies. As relações tróficas são relações alimentares, entre os diferentes organismos presentes na rede. Uma rede trófica também pode ser entendida como a interconecção de diferentes cadeias alimentares das espécies em uma comunidade ecológica.

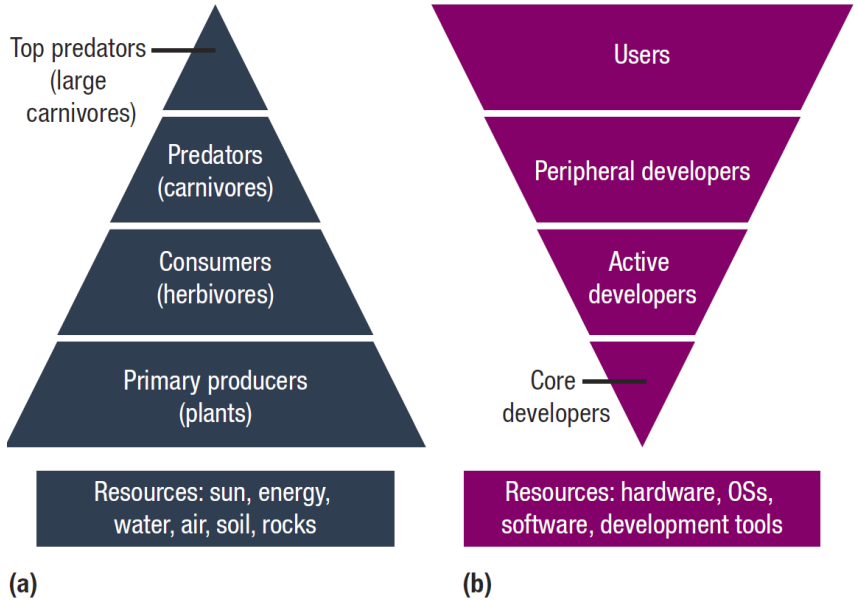


Figura 2.2: Comparação de ecossistemas. (a) Rede trófica de um ecossistema biológico. (b) Um ecossistema visto a partir da perspectiva social (MENS & GROSJEAN, 2015).

a perspectiva de SECO, essa alteração no equilíbrio pode ser causada por uma perda de recurso.

Desta forma, uma diversidade entre os desenvolvedores que colaboram é fator chave para auxiliar o ecossistema a superar os desafios, sejam eles na camada técnica, onde os desenvolvedores devem contar com um leque de tecnologias para dar suporte às necessidades do projeto, como na dimensão social, onde os desenvolvedores complementam suas capacidades para chegar no melhor resultado possível.

CAMPBELL & AHMED (2010) apresentam um modelo que delimita 3 dimensões presentes nos ecossistemas. Essas dimensões foram amplamente aceitas pela literatura e adotadas em trabalhos subsequentes. As dimensões dizem respeito aos conceitos técnicos, de negócios e sociais presentes nos SECO e seguem detalhadas a seguir:

- **Dimensão Técnica**: esta dimensão tem a plataforma de um SECO como foco. Se relaciona ao desenvolvimento da arquitetura de um SECO tendo como objetivo entender como a Engenharia de Software é aplicada na concepção, desenvolvimento e manutenção da plataforma (SANTOS & WERNER, 2011). Esta dimensão deve auxiliar na identificação da plataforma e contextualização dos papéis dos atores envolvidos. Em seguida, existem atividades relacionadas ao processo de abertura da plataforma. Por fim, essa dimensão possui atividades que envolvem elementos

para balancear a modularidade e transparência;

- **Dimensão Negócios**: esta dimensão tem como foco o fluxo de conhecimento, representados pelos artefatos, recursos e informações através de um negócio (estabelecimento de objetivos e planos de ação para projetos). O objetivo desta dimensão é entender como os mecanismos regulatórios e fatores naturais do processo de Engenharia de Software podem impactar um SECO. Além disso, possibilita a obtenção e manipulação de informações de sustentabilidade e diversidade como indicadores da saúde de um SECO. Em (SANTOS & WERNER, 2011) a dimensão de negócios é o foco e são apresentados passos para mapear conceitos, características e comportamentos de outros ecossistemas (naturais, de negócios e sociais) para o contexto SECO, analisar a sustentabilidade para manter os *hubs* e *niche players* e a comunidade que compõe o ecossistema e, por fim, atividades para analisar a diversidade de um SECO com o objetivo de manter diferentes aspectos técnicos e grupos de usuários;

- **Dimensão Social**: esta dimensão tem o foco nos *stakeholders* e suas interações, avaliando como eles integram, estendem e modificam um SECO. Esta dimensão tem como objetivo entender como as redes sociais são criadas, organizadas e como a manutenção de tais redes pode afetar a comunidade de um SECO. Em DOS SANTOS & WERNER (2012) exploram a dimensão social e apresentam atividades para modelagem dos relacionamentos entre atores e artefatos, caracterização dos elementos sociais, como canais de comunicação e perfis, e a análise dos aspectos sociais para utilizar as redes construídas previamente para identificar as interações, cálculo de reputações e possíveis recomendações para aumentar a eficiência na transferência do conhecimento.

A utilização da perspectiva de SECO pode trazer diferentes benefícios para as organizações e grupos que decidem adotar a abordagem. BARBOSA *et al.* (2013) enunciam em um mapeamento sistemático alguns dos benefícios identificados na literatura. Segundo os estudos identificados no mapeamento, o principal benefício da perspectiva de SECO é a promoção do sucesso do software e da co-evolução e inovação no nível

intraorganizacional, aumentando a atratividade para novos colaboradores. Além disso, também foram identificados como benefícios a redução dos custos de desenvolvimento e distribuição, apoio a cooperação e disseminação do conhecimento organizacional, auxílio na análise da arquitetura de software, entre outros.

2.1.3 Redes Complexas

Esta seção tem como objetivo apresentar conceitos básicos também abordados nesta dissertação, dando um referencial para posicionamento e acompanhamento das seções que seguem.

As redes complexas são amplamente utilizadas com o objetivo de auxiliar na compreensão das formas de relacionamento entre pessoas e diferentes entidades. Identificar os elementos mais influentes de um conjunto de membros é uma tarefa discutida em múltiplas pesquisas visto que estes nós podem auxiliar a aumentar a robustez em diferentes domínios.

Em (GUÉRCIO *et al.*, 2017), o domínio científico é explorado através da análise de uma rede de coautoria com o objetivo de identificar cientistas influentes em uma rede social científica. Os relacionamentos possuem pesos que consideram o tempo para penalizar interações antigas e utiliza a centralidade de *closeness* para identificar os autores com maior impacto de colaboração. A Figura 2.3 apresenta um exemplo de visualização de redes complexas, que representa uma rede de coautoria.

Os relacionamentos presentes nos SECOs podem auxiliar diferentes *stakeholders* a compreender melhor a rede de colaboradores que compõem o SECO. Com foco no domínio empresarial a identificação de colaboradores chave também é um ponto de interesse, visto que os envolvidos podem utilizar os resultados de pesquisa para obter vantagem competitiva. De forma geral, os elementos da rede complexa são identificados e os líderes destacados auxiliam na disseminação do conhecimento. Em (DI TOMMASO *et al.*, 2016) uma medida é proposta para capturar o nível de liderança de um usuário em redes sociais construídas a partir da análise do ambiente empresarial para auxiliar em decisões de avanços de carreira e com objetivo de auxiliar na gestão de pessoas.

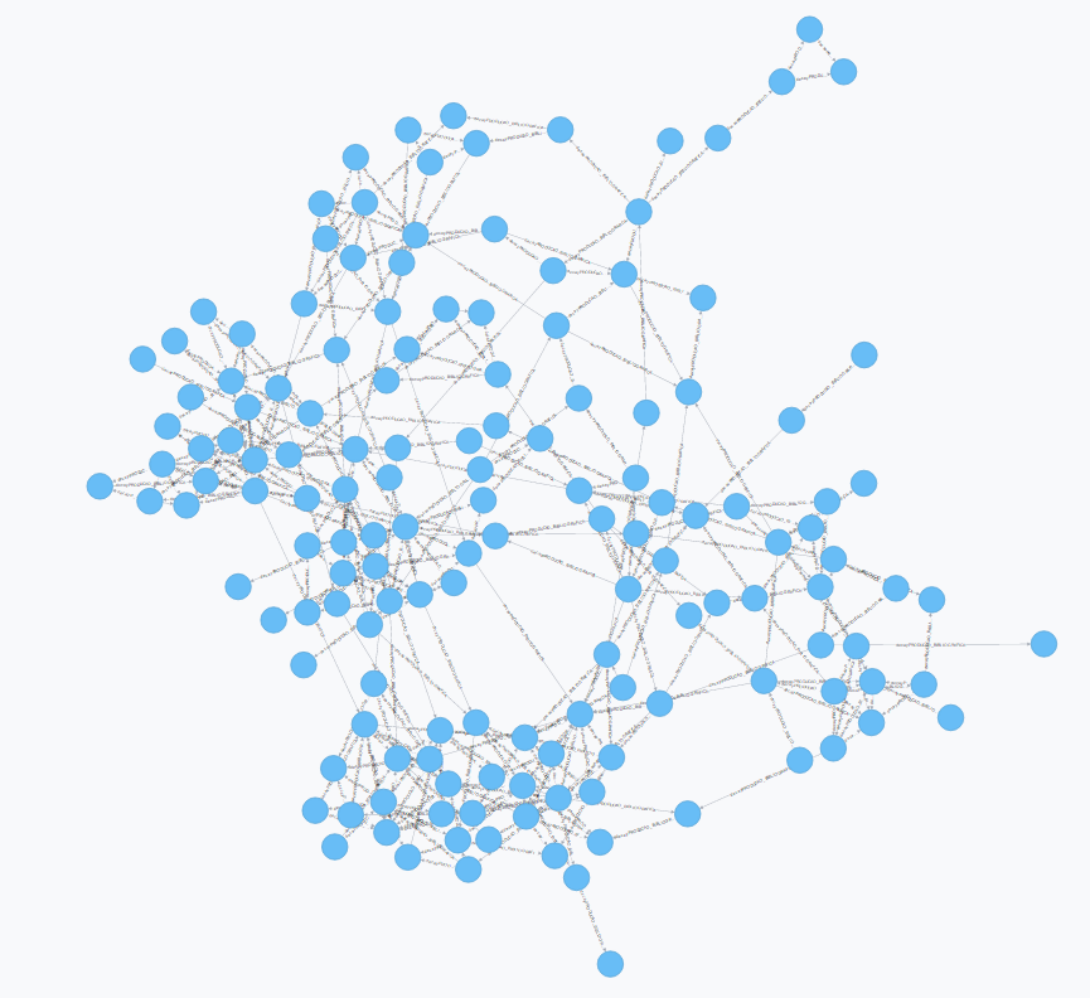


Figura 2.3: Rede de coautoria GUÉRCIO *et al.* (2017).

2.1.4 Softwares de Código Aberto

Os softwares de código aberto são uma abordagem de organizações ou grupos de desenvolvimento que decidem retirar os limites de acesso ao software, possibilitando que qualquer desenvolvedor interessado possa contribuir e entender o funcionamento do projeto. VAN DEN BERK *et al.* (2009), deixam claro que existem diferentes níveis de abertura a serem avaliados, e dentre os diferentes tipos os autores classificam como mais importantes as aberturas de padrões, formatos e de código fonte. A utilização de padrões tornam possível uma variedade de produtos interoperáveis e intercambiáveis, desenvolvidos por diferentes companhias, aumentando a satisfação dos usuários e a competição entre os envolvidos. A utilização de formatos padrões facilitam o armazenamento e transmissão de conhecimento, tendo como por exemplo os formatos HTML, XML ou JSON. A abordagem de Open Source possibilita o gerenciamento do desenvolvimento e distribuição de

software através de licenças específicas que garantem aos usuários o acesso necessário para colaborar.

A abordagem de código aberto e SECO são relacionadas, e NASSERIFAR (2016) realizou uma revisão sistemática da literatura sobre os OSSECO (*Open Source Software Ecosystem*). Dentre os ecossistemas identificados nos artigos selecionados, destacam-se o Eclipse, Gnome e MySQL. O MySQL por exemplo pode ser considerado um ecossistema maduro, com mais de 20 anos de atuação, que possibilita utilização de diferentes *plugins* e modelos de armazenamento que fazem com que a solução se ajuste aos moldes de quem a necessita, podendo ser customizada de acordo com suas necessidades. Ao fim do estudo, foram identificados nos diferentes trabalhos listas de definições, atores, modelos e desafios, e tal pesquisa pode auxiliar nos avanços relacionados a OSSECO.

2.2 Considerações Finais do Capítulo

O objetivo deste capítulo foi apresentar os conceitos base deste trabalho. Foram abordados os principais aspectos relativos aos Ecossistemas de Software, Redes Complexas e sobre softwares de código aberto.

3 sSECO-Process

Neste capítulo, a solução proposta nesta dissertação é estendida, com o objetivo de enriquecer o processo a partir das melhorias identificadas durante as reuniões com especialistas e da execução do estudo preliminar. Foi notada a necessidade de melhorias no processo para aumentar a clareza das atividades, assim como complementar atividades relevantes não consideradas no processo inicial.

Durante a execução do processo, dois papéis foram considerados, sendo eles de pesquisador e desenvolvedor. O usuário do processo com papel de pesquisador tem como responsabilidades orientar o processo, garantindo que os objetivos sejam estabelecidos e que as atividades sejam realizadas para que ao fim do processo o objetivo seja alcançado. O usuário com papel de desenvolvedor deve atuar nas atividades técnicas, apoiando durante a extração, tratamento dos dados e construção do ambiente de análise.

Este capítulo apresenta o processo melhorado, assim como os motivos das alterações em cada um dos pontos. O restante do capítulo está assim organizado: inicialmente uma visão geral do processo é apresentada, e as etapas foram divididas em seções que explicam as atividades de coleta, construção do ambiente de análise e da análise.

Figura 3.1: Visão global do processo.

A Figura 3.1 mostra de forma sucinta as macro atividades a serem realizadas durante a análise da dimensão social de SECOs a partir do processo melhorado. A primeira atividade compreende a obtenção dos dados, para que, a partir destes dados, um ambiente de análise seja construído. O ambiente de análise é construído a fim de possibilitar que os envolvidos, interessados em extrair conhecimento, possam utilizar os dados disponíveis de maneira eficiente e replicável.

Após a conclusão da construção do ambiente de análise é possível avaliar as comunidades e organizações, representadas pelos dados extraídos, para entender suas características. Neste trabalho, foram identificados elementos que possam auxiliar os envolvidos em suas atividades.

Por fim, são apresentadas oportunidades de análise para auxiliar em pesquisas futuras que desejem utilizar o processo proposto, avaliando novos cenários e enriquecendo a pesquisa aqui apresentada. As seções a seguir detalham os subprocessos da Figura 3.1, fornecendo abordagens e direcionamentos para que as atividades sejam executadas de maneira correta.

3.1 Coleta de Dados

A primeira etapa do processo principal compreende a obtenção dos dados, para as etapas posteriores. A questão de pesquisa deve ser considerada desde a primeira atividade, pois existem diversas fontes de informação disponíveis. Exemplificando, o cientista pode desejar entender sobre o desenvolvimento de software por corporações que desenvolvem software proprietário, fazendo com que possam existir restrições de inspeção do código fonte ou de identificação dos colaboradores. Desta forma, o cientista deve entrar em contato com as corporações que detém os direitos para solicitar acesso aos dados.

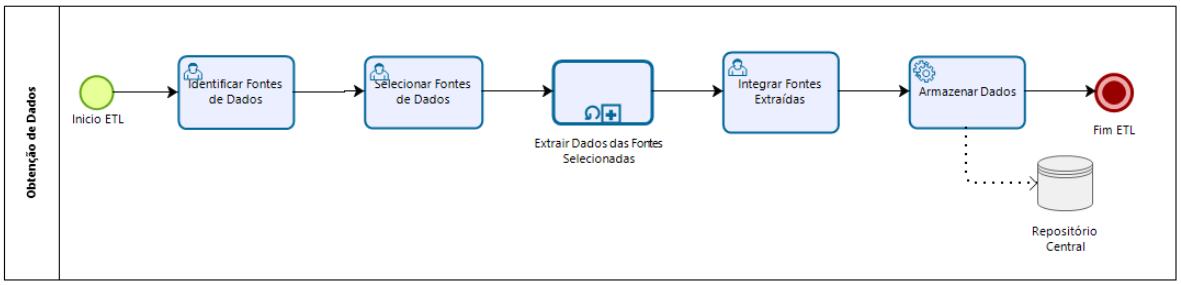


Figura 3.2: Subprocesso de obtenção dos dados.

O subprocesso de obtenção dos dados é apresentado na Figura 3.2, que apresenta as atividades a serem desenvolvidas pelos envolvidos. De forma resumida, as possíveis fontes de dados devem ser identificadas, para que a partir dessa lista de fontes sejam selecionadas as mais aderentes ao objetivo definido a priori pelos utilizadores do processo.

Após a seleção das fontes o processo de extração deve ser executada. Devido à

complexidade desta atividade, ela foi modelada em um subprocesso, que é apresentado na Figura 3.3. Após a conclusão da extração dos dados, uma atividade de integração é realizada para consolidar as informações das diferentes fontes de dados em um banco de dados centralizado.

As atividades estão detalhadas nas subseções a seguir, assim como as justificativas de melhorias nas atividades que foram alteradas.

3.1.1 Identificação e Seleção das Fontes de Dados

De acordo com a natureza livre e distribuída do desenvolvimento *open source*, diversas ferramentas, descritas como *software forges* são utilizadas para apoiar as atividades de desenvolvimento.

De acordo com RIEHLE *et al.* (2009), um *software forge* é uma plataforma web que integra as melhores ferramentas para colaboração em desenvolvimento de software. Um *software forge*, normalmente, possui duas visões. A primeira é referente aos desenvolvedores de software, que enxergam estas plataformas como um lugar para hospedar código fonte e outros arquivos pertinentes ao desenvolvimento, como documentação ou *bug reports*. A segunda visão é dos usuários, que as visualizam como um repositório de aplicações. A partir da identificação de recursos disponíveis os usuários coletam as aplicações ali armazenadas e as utilizam para auxiliar a atender suas necessidades.

A etapa de identificação das fontes de dados deve ser realizada com cautela, visto que caso uma fonte importante não seja capturada durante esta etapa, ela não poderá ser selecionada e a avaliação poderá ser prejudicada. Desta forma, durante esta etapa é desejável que seja realizada uma cautelosa busca na literatura para identificar as possíveis fontes estudadas pela comunidade. Ao realizar a identificação das fontes, também é necessário que a data das publicações avaliadas sejam consideradas, visto que alguns trabalhos podem utilizar fontes desatualizadas devido ao ritmo acelerado da evolução tecnológica.

Além da consulta à literatura, é interessante observar como a comunidade utiliza as ferramentas, correlacionando os resultados encontrados na literatura com o estado atual das tecnologias para auxílio ao desenvolvimento de software, visto que tais tecnologias

serão as candidatas a fornecer os dados capturados durante sua utilização.

Durante o estudo preliminar, diversas fontes foram identificadas, mas fez-se necessário selecionar quais seriam as mais relevantes. Desta forma, a atividade de seleção das fontes de dados foi adicionada deixando explícito que deve ser realizada uma seleção a partir do conjunto de fontes identificadas, visto que a identificação de uma fonte, não necessariamente implica que ela será aderente aos objetivos dos envolvidos no processo.

3.1.2 Extração dos Dados

Após a seleção das fontes de dados, faz-se necessário identificar as possíveis abordagens para extração de cada uma delas. As fontes de dados selecionadas devem possuir características comuns para possibilitar a integração dos dados. Entretanto, podem existir características específicas entre as fontes selecionadas que devem ser consideradas.

Essas especificidades entre as fontes podem ter diferentes origens, sendo, por exemplo, a diferença entre o foco de cada uma das fontes, fazendo com que algumas funcionalidades não sejam comuns. Além disso, diferentes fontes de dados podem ter políticas de publicidade distintas, fazendo com que alguns atributos não possam ser extraídos.

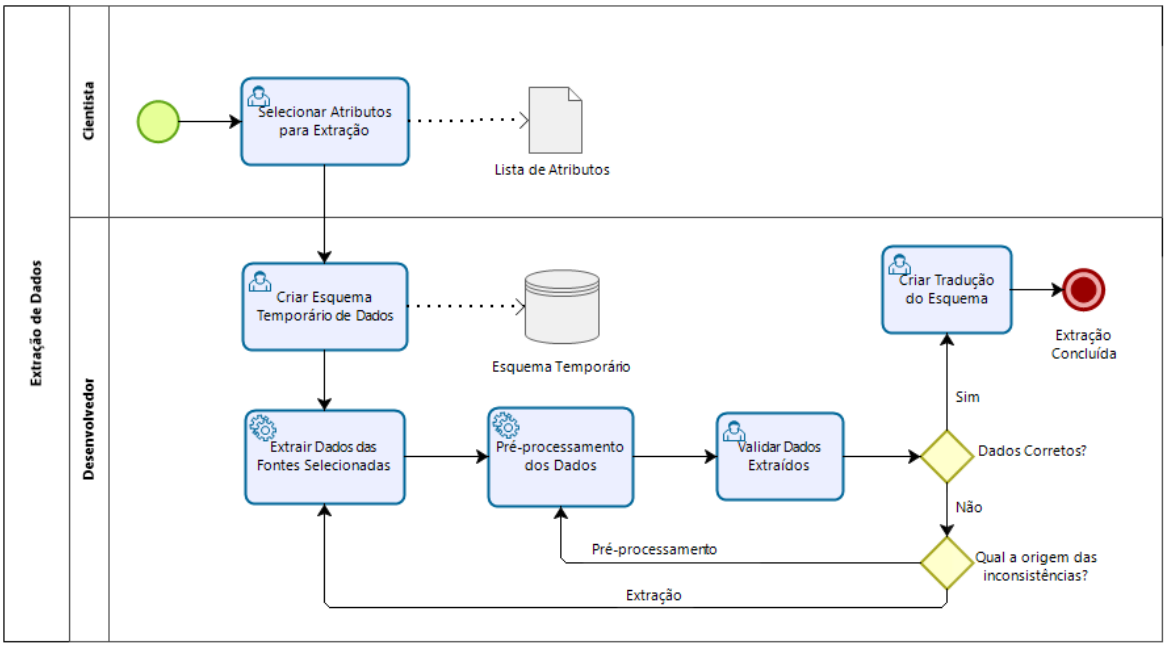


Figura 3.3: Subprocesso de extração dos dados.

A atividade de extração possui atividades para agentes com diferentes perfis. O

primeiro perfil, é o do cientista que deseja avaliar um SECO através do processo proposto nesta dissertação. O cientista deve avaliar as fontes de dados a fim de identificar atributos para a extração. Ao avaliar as fontes de dados, é interessante que ele observe os agentes envolvidos e modelados nos dados, assim como suas características e relacionamentos. Após a avaliação do cientista um documento é produzido contendo a lista de atributos a serem extraídos.

O segundo perfil envolvido é representado pelos desenvolvedores, que são responsáveis por construir as soluções técnicas que capturam os atributos selecionados. A primeira atividade do desenvolvedor é a de criação de um esquema temporário de dados. Este esquema deve ser criado para possibilitar o armazenamento temporário dos dados contidos nas fontes selecionadas. Após a criação do esquema temporário, o desenvolvedor deve avaliar os mecanismos de extração que possam capturar os elementos presentes no documento produzido pelo cientista na etapa anterior.

A avaliação das abordagens de extração é realizada pelo desenvolvedor, que avalia as soluções técnicas fornecidas por cada fonte de dados para capturar os elementos de observação. Tais elementos podem ser recuperados através de chamadas a APIs que possuam os *endpoints* aderentes às necessidades do cientista. Outras fontes de dados podem não ter interfaces que facilitem a extração dos dados ou tais interfaces podem ser incompletas. O desenvolvedor deve ser capaz de identificar maneiras alternativas de obtenção dos dados, como, por exemplo, a utilização de *web scrapers* para capturar informações presentes nas páginas das possíveis fontes de dados.

Após a identificação da abordagem de extração, uma base de dados temporária é instanciada a partir do esquema criado na atividade anterior. Esta base temporária é preenchida com os dados extraídos. Posteriormente à extração dos dados, um pré-processamento deve ser realizado com o objetivo de ajustar os dados, que muitas vezes possuem ruídos com diferentes origens e comportamentos. Nesta etapa, o desenvolvedor deve tratar casos como, por exemplo, dados ausentes, conversão de unidades e datas ou identificação de *outliers*. No fim do pré-processamento dos dados na base temporária, o desenvolvedor deve realizar uma validação para garantir que os dados foram extraídos corretamente e que o pré-processamento ocorreu de maneira correta.

Caso os dados não estejam corretos, o desenvolvedor deve identificar a origem das inconsistências e ajustar a abordagem de extração ou pré-processamento. Após as tratativas necessárias, o desenvolvedor, entendendo que os dados foram extraídos corretamente, deve realizar a atividade de tradução do esquema temporário. Esta tradução visa garantir que os dados sejam consolidados em uma base central, possibilitando que os cientistas possam gerar conhecimento a partir das múltiplas fontes de dados selecionadas.

Devido a alta complexidade da tarefa de extração dos dados, decidiu-se por transformar essa atividade, presente no estudo preliminar, em um subprocesso, com o intuito de apresentar mais detalhes sobre as tarefas a serem executadas. Além disso, a atividade de ajuste de dados, presente no processo preliminar, foi incorporada ao subprocesso de extração, visto que possíveis inconsistências podem estar atreladas a atividade de extração. Desta forma, a extração de dados só é dada como completa após a validação dos dados coletados. Além disto, foram inseridas atividades de criação de um esquema temporário e da sua tradução para o esquema do repositório central. Esta atividade está relacionada à integração de dados que se faz necessária quando se adotam múltiplas fontes de dados.

Após a execução do subprocesso de extração de dados, as atividades de integração dos dados a partir dos dados extraídos deve ser realizada. Essa integração é facilitada pelos *scripts* de tradução, criados durante o subprocesso de extração, para relacionar os dados extraídos com os campos correspondentes no repositório central. Após sua execução é possível consolidar os dados, dando fim a etapa de obtenção dos dados.

3.2 Construção do Ambiente de Análise

Após a etapa de extração, os dados estão consolidados em uma base central, independente da sua origem ou estratégia de coleta. A partir da base consolidada faz-se necessária a construção de um ambiente que proporcione ao cientista a possibilidade de analisar estes dados de maneira facilitada. Para tal, são necessários alguns ajustes para delimitar o escopo de observação e definição das possíveis redes a serem utilizadas durante a análise.

A Figura 3.4 apresenta o subprocesso de construção do ambiente, definindo atividades básicas a serem realizadas para que, ao fim deste subprocesso, o ambiente esteja pronto para análise do cientista.

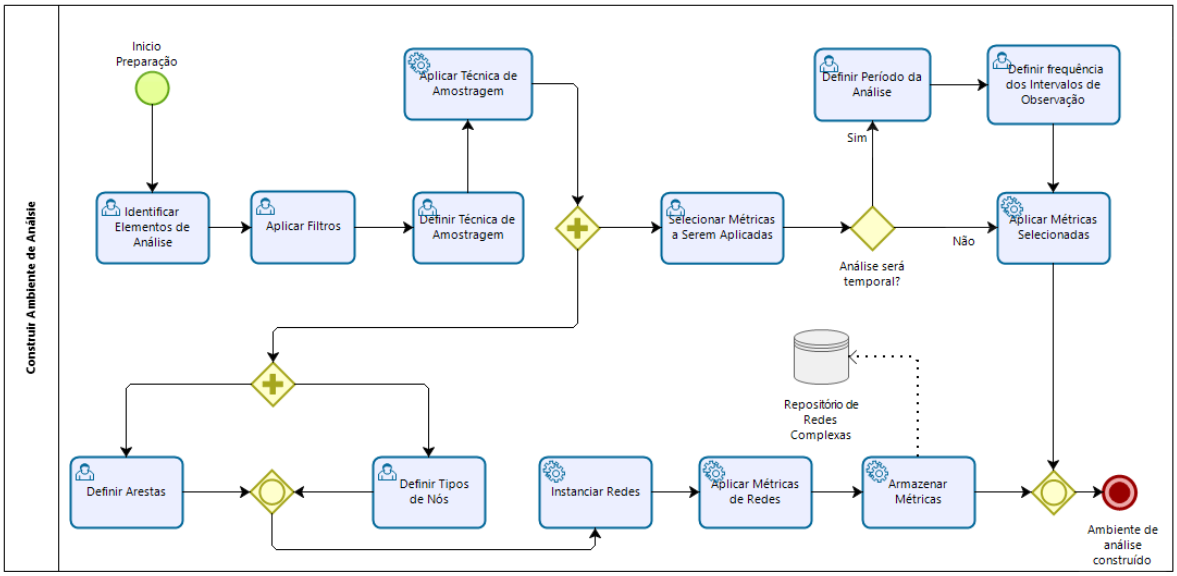


Figura 3.4: Construção do ambiente de análise.

A primeira atividade a ser realizada é a identificação dos elementos de análise presentes na base central. Estes elementos compreendem os agentes e artefatos envolvidos durante o desenvolvimento de software. Durante a identificação, o cientista deve ponderar quais são os elementos que o auxiliará nas respostas às suas questões de pesquisa. Utilizar-se de todos os elementos disponíveis pode não ser a melhor abordagem, visto que a complexidade da análise pode aumentar muito, sem trazer um resultado superior. Desta forma, esta atividade foi acrescentada como uma melhoria no processo.

De posse dos elementos de análise, o cientista deve aplicar filtros com o intuito de mitigar problemas de interpretação ou ameaças à validade das pesquisas conduzidas seguindo o processo. Estes filtros também podem auxiliar na delimitação do escopo desejado, limitando o período de observação ou os elementos a serem avaliados.

A definição dos filtros deve considerar as fontes de dados utilizadas, visto que diferentes fontes de informação podem ter características que prejudiquem a análise posterior. Durante esta etapa, a pesquisa à literatura faz-se necessária, visto que os trabalhos científicos, que prezam pelo rigor, identificam os possíveis perigos e ameaças à validade. A partir da consulta a literatura o cientista pode reunir o conhecimento compartilhado pela comunidade para garantir filtros de alto nível para sua pesquisa. Esta atividade foi decorrente de uma melhoria no processo, visto que a não execução de técnicas de filtragem pode comprometer análises posteriores, deteriorando os resultados.

Em seguida, a amostra deve ser definida, e, como uma melhoria no processo, essa tarefa foi dividida em duas atividades, sendo a primeira de definição da técnica de amostragem, seguida pela aplicação da técnica nos dados selecionados.

Concluída a aplicação da técnica de amostragem, três atividades podem ser iniciadas e executadas em paralelo. A primeira das atividades é a de seleção das métricas a serem aplicadas durante o estudo. Cada métrica deve ter um objetivo para avaliação, mantendo a complexidade da análise baixa e evitando dificuldades nas etapas posteriores. As outras atividades, seguintes a aplicação da técnica de amostragem, são referentes à utilização das redes nas etapas de análise. A utilização de redes complexas para analisar o aspecto social de um SECO tem grande valor, visto que as relações sociais existentes no SECO podem ser modeladas e analisadas através dessas redes. Desta forma, o cientista pode utilizar as abordagens existentes na literatura, aplicando estratégias de visualização e utilizando medidas relacionadas às redes complexas aplicadas ao seu cenário.

As arestas e nós devem ser definidos para possibilitar a modelagem e construção das redes. Para identificação dos possíveis nós, o cientista deve observar os elementos de análise identificados e as relações entre estes elementos (arestas). Faz-se importante considerar as características, dos elementos e relacionamentos, para viabilizar a construção de redes que sejam ricas em informação e com múltiplas oportunidades de análise.

Após a definição dos elementos básicos das redes, a instanciação das redes é viabilizada. As atividades de definição de arestas, nós e instanciação da rede visam detalhar a atividade de definição das redes complexas presentes no processo executado durante o estudo preliminar. Tais atividades foram subdivididas com o objetivo de destacar os elementos básicos ao se trabalhar com redes complexas.

Após a instanciação das redes complexas, a aplicação das métricas referentes a elas se torna possível. As medidas podem dar uma visão geral da rede, como por exemplo o diamêtro, quantidade de componentes conectados, grau médio ou comprimento do caminho médio entre dois nós. Também podem ser avaliadas medidas dos nós, como, por exemplo, o seu grau ou das diferentes medidas de centralidade, como *betweenness, closeness* ou *eigenvector* a título de exemplo. Após a aplicação das métricas elas devem ser armazenadas em um repositório de redes complexas com o objetivo de diminuir o

processamento para diferentes análises que utilizem as redes instanciadas.

Durante a construção do ambiente de análise, o cientista também deve considerar se a análise será temporal, a fim de capturar aspectos da evolução dos objetos de estudo. Caso a análise seja temporal, o cientista deve definir o período de análise, assim como a frequência entre os intervalos de observação, dividindo o período em segmentos de igual duração e possibilitando a observação do comportamento através de uma janela deslizante de tempo. As tarefas relacionadas a questões que relacionam o tempo foram adicionadas, visto que elas não estavam presentes no processo inicial. Além disso, foi adicionada uma decisão, já que a análise temporal não é obrigatória, atuando apenas de forma a enriquecer as análises a serem realizadas.

3.3 Análise

A etapa de análise representa a última macro atividade a ser realizada durante o processo descrito neste capítulo. A Figura 3.5 apresenta as atividades a serem realizadas para analisar a dimensão social de SECOs. É possível perceber uma grande diferença quando comparada às atividades de análise do processo executado no estudo preliminar.

As alterações nas atividades de análise foram realizadas pois esta é a etapa que gera mais valor durante a execução do processo. A partir desta análise o cientista pode auxiliar os envolvidos no processo de desenvolvimento colaborativo a serem mais eficientes, possibilitando melhorias na comunicação e indicando pontos de atenção para a comunidade que compõe o SECO. Sendo assim, os especialistas indicaram que esta etapa deveria ser enriquecida com o intuito de facilitar a execução do processo pelos cientistas.

A primeira atividade a ser realizada pelo cientista é a definição de um objetivo. A observação pode ser feita com diferentes propósitos, e cada um desses objetivos pode necessitar de diferentes elementos, métricas e abordagens. Esta atividade não estava presente no processo executado no estudo preliminar e foi acrescentada, pois diferentes objetivos levam a diferentes estratégias de análise.

As tarefas relacionadas às redes complexas estavam associadas a apenas uma atividade no processo executado no estudo preliminar. Elas foram subdivididas em outras atividades e em um subprocesso para detalhar as atividades a serem realizadas, possibi-

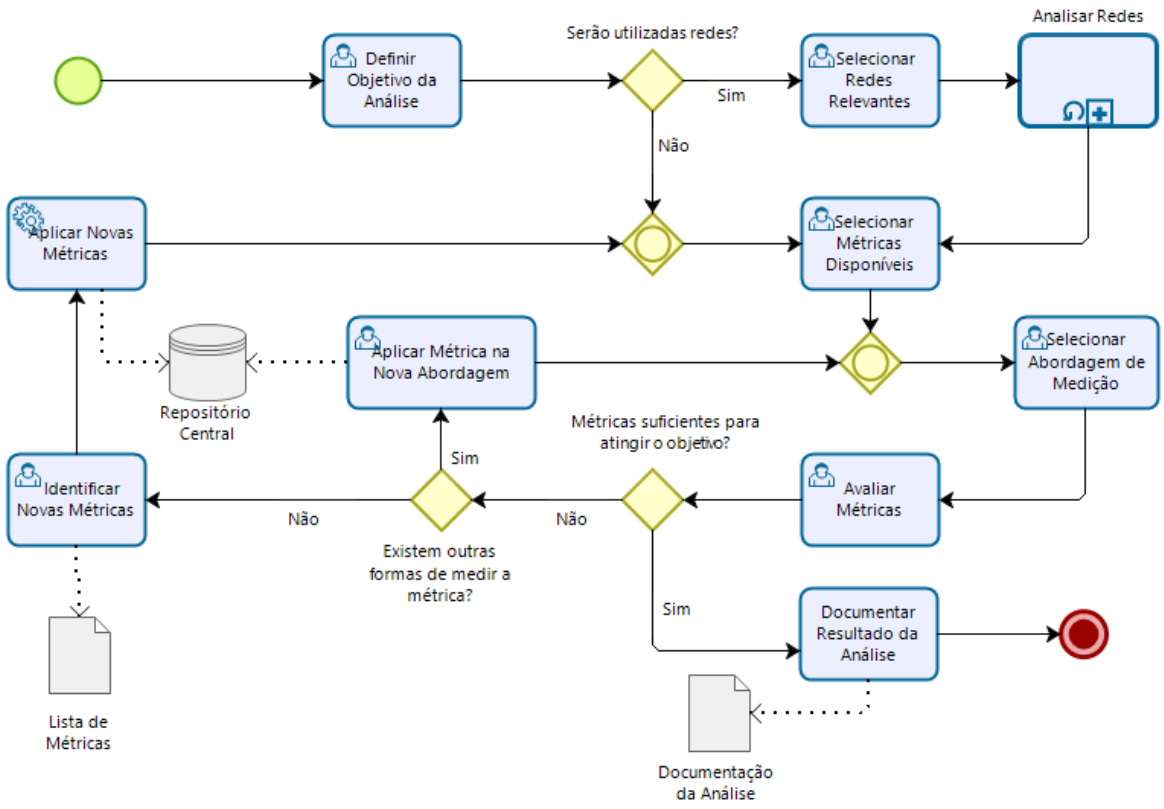


Figura 3.5: Atividades realizadas durante a etapa de análise.

litando a reutilização do processo por outros cientistas.

Após a definição do objetivo o cientista deve decidir sobre a utilização das redes complexas ou não. Caso o cientista decida por utilizar as redes complexas, deve-se identificar quais as redes complexas, modeladas durante a construção do ambiente de análise, são as mais aderentes aos seus objetivos. O processo de avaliação das redes complexas pode ser decomposto em diferentes atividades, desta forma os passos a serem realizados nesta análise foram separados em um subprocesso, apresentado na Figura 3.6 e detalhado na próxima subseção.

Após a decisão sob a utilização das redes, caso a resposta seja negativa o cientista inicia a seleção das métricas disponíveis, que serão utilizadas para identificar os elementos que merecem destaque e que sejam aderentes aos objetivos da pesquisa. Se o cientista tiver optado por utilizar as redes em sua avaliação, ele deve realizar a análise das redes para, em seguida, realizar a seleção das métricas.

Após a seleção das métricas, o cientista deve avaliar as possíveis abordagens para realizar a medição das métricas. Isto faz-se necessário visto que uma métrica pode ter di-

ferentes alternativas de medição, impactando em seus valores e nas observações realizadas a partir de cada métrica. Exemplificando, pode-se observar uma métrica que tenha como objetivo capturar a eficiência de uma determinada atividade. Existem diferentes formas de medir a eficiência, como a quantidade de recursos alocados, o tempo necessário para conclusão, a satisfação dos consumidores do artefato, a comparação com outras instâncias de realização da mesma atividade ou de várias outras maneiras.

Após a seleção das métricas e das maneiras de medir cada uma delas, o cientista deve avaliar os elementos de análise para cada métrica selecionada, e decidir se as métricas foram ou não suficientes para auxiliar a atingir o objetivo da análise. Caso as métricas não sejam suficientes, o cientista deve avaliar se existem outras formas de medir a métrica em questão, e caso exista ele deve aplicar a métrica a partir desta nova abordagem, consolidando seus valores no repositório central, e realizando uma nova avaliação da métrica.

Se o cientista não possuir novas formas de realizar a medição desta métrica, e ainda assim, ela não seja suficiente para alcançar o objetivo, o cientista deve realizar a identificação de novas métricas, aplicar as novas métricas a partir dos dados disponíveis e realizar novamente o processo de avaliação.

Quando as métricas se mostrarem suficientes para auxiliar na obtenção do objetivo, o cientista deve documentar o resultado da análise, assim como suas escolhas de métricas, redes e abordagens de medição, para possibilitar que a comunidade possa utilizar sua abordagem para realizar novas avaliações em conjuntos diferentes de dados ou de replicar o trabalho com outros propósitos.

A atividade de análise foi a que mais sofreu alterações com relação ao processo executado durante o estudo preliminar. Foram acrescentadas novas atividades e outras foram subdivididas. Além disso, foram acrescentadas decisões durante o subprocesso de análise para que o cientista possa refinar gradativamente o estudo e encontrar resultados mais interessantes, de acordo com seus objetivos.

3.3.1 Análise de Redes Complexas

A análise das redes complexas é uma atividade opcional durante a análise da dimensão social dos ecossistemas de software, visto que outros estudos avaliam a dimensão sem utilizar deste recurso, entretanto, esta adição na avaliação pode render bons resultados visto que a grande quantidade de relacionamentos entre as entidades pode ser modelada através das redes, que possuem diferentes métricas e características que podem ser utilizadas para obter mais conhecimento a partir dos dados selecionados. Estes relacionamentos modelados com o auxílio das redes complexas são profundamente aderentes a abordagem de SECO visto que os relacionamentos entre as unidades que compõe o ecossistema estão presentes nas definições identificadas na literatura.

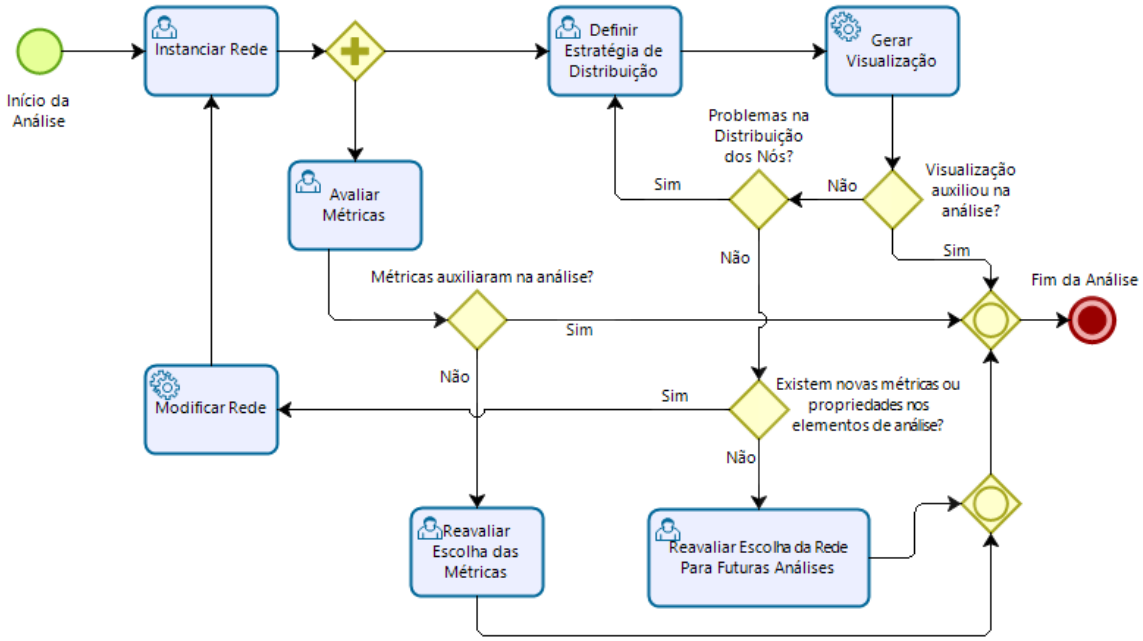


Figura 3.6: Subprocesso de análise de redes complexas.

A Figura 3.6 apresenta as atividades a serem realizadas na análise das redes. O primeiro passo é instanciar esta rede, a partir dos nós e relacionamentos selecionados em etapas anteriores. Esta instanciação pode ser realizada com o auxílio de diferentes aplicações, como graphviz[3] ou gephi[4], duas plataformas *open source* para auxílio na visualização e coleta de métricas de redes. Também é possível realizar a instanciação com

[3]http://graphviz.org/
[4]https://gephi.org/

outras ferramentas, como NodeXL[5] que cria templates e possibilita aos usuários utilizarem o Excel para instanciar e avaliar redes. Outra forma é a utilização de bancos em grafos, como o Neo4j[6].

Após a instanciação das redes, o cientista deve avaliar as métricas de rede e ponderar sobre a utilidade de cada uma durante o processo de análise. Além da avaliação das métricas, o cientista deve escolher a estratégia de distribuição dos nós para que seja possível gerar visualizações ricas. Diferentes estratégias devem ser avaliadas de acordo com o propósito da rede. Exemplificando, caso o cientista queira destacar os *authorities* e *hubs*[7] de uma rede, ele pode utilizar a estratégia Force Atlas, que tende a centralizar os nós com estes valores acentuados. Caso as redes possuam arestas com peso e não direcionadas, a estratégia OpenOrd pode ser utilizada para distinção de *clusters*.

Após a definição da estratégia de distribuição e com a visualização disponibilizada o cientitsta deve avaliar os resultados. Caso a rede não auxilie o cientista a aproximar-se dos seus objetivos, então ele deve reavaliar a escolha de distribuição dos nós. Caso não existam abordagens que auxiliem o cientista, ele deve verificar se existem novos elementos ou métricas que possam auxiliar nesta observação. Se existirem novos elementos ou características, o cientista deve realizar uma nova instanciação da rede, reiniciando o processo de avaliação.

Por fim, o cientista deve documentar a escolha das métricas e das redes para futuras análises, mostrando aos cientistas os caminhos a serem seguidos para utilização correta das redes com o intuito de atingir os objetivos esperados.

3.4 Considerações Finais do o Capítulo

Neste capítulo foi apresentado o sSECO-Process, que teve suas atividades detalhadas assim como novas atividades e o aumento na granularidade quando comparadas com o sSECO-Process apresentado no Apêndice A. Todas as etapas do processo foram estendi-

[5]https://archive.codeplex.com/?p=nodexl

[6]https://neo4j.com/

[7]O conceito de authorities e hubs é apresentado no algoritmo de análise de links desenvolvido por (KLEINBERG, 1999) que tinha como objetivo ranquear páginas web. De acordo o autor um bom hub representa um elemento conectado a diferentes elementos de uma rede, e um bom authority representa um nó conectado a vários hubs

das, acrescentando detalhes, sendo a etapa de análise a que sofreu mais mudanças. Esta etapa teve o acréscimo de decisões que o cientista deve tomar com o objetivo de refinar a análise. Além disso, a atividade de análise de redes foi decomposta em um subprocesso pela sua importância. O capítulo a seguir apresenta a instanciação e avaliação do sSECO-Process.

4 Avaliação do sSECO-Process

Neste capitulo é apresentada a avaliação da solução, através de um estudo seguindo a abordagem GQM, que tem como objetivo avaliar a viabilidade da solução proposta neste trabalho, apontando evidências que os objetivos definidos previamente foram atendidos e que a questão de pesquisa formulada no início deste trabalho foi respondida.

Na proposta apresentada, um processo para extração de conhecimento através da análise da dimensão social em ecossistemas de desenvolvimento de software foi definido. A avaliação foi dividida em duas partes, sendo a primeira composta de um estudo preliminar, que teve como propósito avaliar o processo. A segunda parte da avaliação é apresentada neste capítulo, e foi necessária, pois, a partir do estudo preliminar realizado, foram identificados pontos de melhoria e oportunidades de enriquecimento da solução.

Desta forma, o restante deste capítulo apresenta de maneira detalhada o planejamento da avaliação e a execução da solução apresentada anteriormente, culminando numa avaliação conduzida com o objetivo de avaliar o trabalho presente nesta dissertação.

4.1 Planejamento da Avaliação

A avaliação segue a abordagem GQM, utilizada também no estudo preliminar, com o propósito de auxiliar na resposta das questões de pesquisa e seguindo o arcabouço proposto por BASILI (1992). Tem-se como objetivo: **Analisar** o processo de extração de informação **com o objetivo** de gerar conhecimento através da análise de redes complexas **em relação** ao suporte no desenvolvimento global de software **do ponto de vista de** *stakeholders* **no contexto** de Ecossistemas de Software.

O arcabouço apresentado no estudo preliminar foi mantido visto que o objetivo de avaliar o processo de extração de informação, gerando conhecimento através de redes complexas se manteve, entretanto novas questões foram acrescentadas. As novas questões tem como objetivo explorar novas perspectivas visto que o processo apresentado na solução possibilita novas oportunidades de análise e exploração. Desta forma, novas métricas

também foram adicionadas com a finalidade de avaliar as questões adicionais.

Assim como no estudo preliminar, os objetivos, questões e métricas são apresentados por um grafo direcionado na Figura 4.1, que auxilia na organização e entendimento do fluxo dos objetivos até as métricas.

O detalhamento da nova questão e das métricas é apresentado a seguir. Mais detalhes sobre as questões e métricas presentes na primeira execução do GQM podem ser encontrados na Subseção A.2.1.

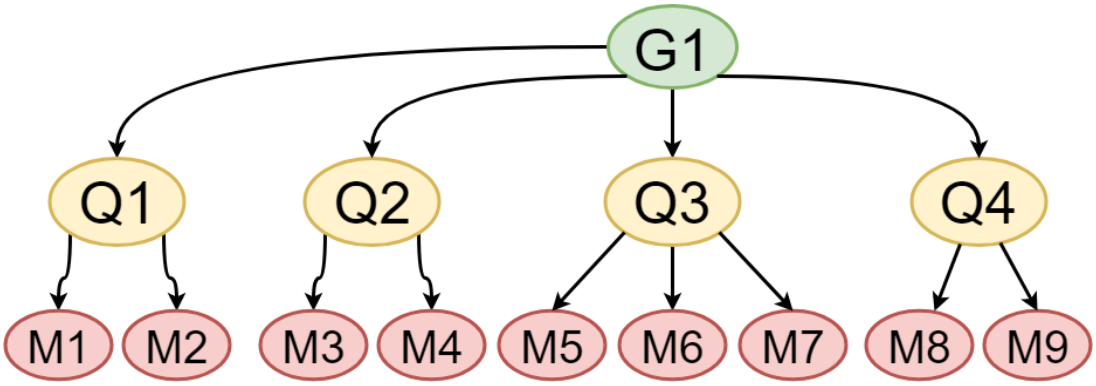


Figura 4.1: Grafo direcionado representando a abordagem GQM.

Questions

- **Q4:** Quais as relações existentes entre o conhecimento obtido através das medidas das entidades estudadas e das medidas obtidas através das redes complexas?

  Objetivo: verificar como as medidas disponíveis na literatura se relacionam com o conhecimento obtido através da análise das redes complexas modeladas.

Metrics

- **M8:** Correlação de Pearson.
- **M9:** Correlação de Spearman.

Os participantes durante a avaliação do GQM foram especialistas do domínio de SECO e de Desenvolvimento de Software Distribuído.

4.2 Avaliação

Esta seção apresenta a execução do sSECO-Process apresentado no capítulo anterior e está subdividida de acordo com as macro atividades apresentadas na Figura 3.1, detalhando cada uma das atividades desenvolvidas.

O conjunto de dados utilizado para a avaliação do sSECO-Process foi construído por BLINCOE *et al.* (2015), que apresentam uma estratégia de identificação de ecossistemas a partir dos dados do GitHub. De acordo com OSSHER *et al.* (2010), a identificação de dependências técnicas entre projetos é uma atividade custosa, principalmente quando realizada em larga escala, fazendo com que a identificação de tais dependências sejam um desafio para estudos que desejem realizar este tipo de análise em repositórios de desenvolvimento de software.

Dada a dificuldade na identificação dependências técnicas entre projetos, BLINCOE *et al.* (2015) utilizam referências cruzadas, especificadas pelos usuários em comentários dos projetos, como base da estratégia de identificação de ecossistemas. A Figura 4.2 apresenta as referências cruzadas nos projetos do GitHub(BLINCOE *et al.*, 2015).

Através de um contato com os autores, foi obtida a listagem dos projetos avaliados no estudo, assim como dos ecossistemas identificados. Para a avaliação apresentada neste capítulo foi utilizado o maior ecossistema identificado durante o estudo, que teve seus dados extraídos do *dataset* fornecido pelo projeto GHTorrent. Desta forma, a avaliação foi realizada com dados atualizados e através de uma técnica de extração de dados diferente do estudo preliminar.

O ecossistema em questão tem como projeto central o projeto joyent/node, que é um interpretador de código JavaScript com código aberto, que tem como foco utilizar javascript diretamente nos servidores. Durante o estudo realizado por BLINCOE *et al.* (2015) foram identificados 134 projetos que compreendem este ecossistema. Cada projeto identificado teve seus repositórios *fork* considerados, fazendo com que o *dataset* utilizado durante a avaliação, construído a partir da identificação das dependências entre projetos realizado em BLINCOE *et al.* (2015), tivesse um total de 65.379 repositórios e 239.559 usuários.

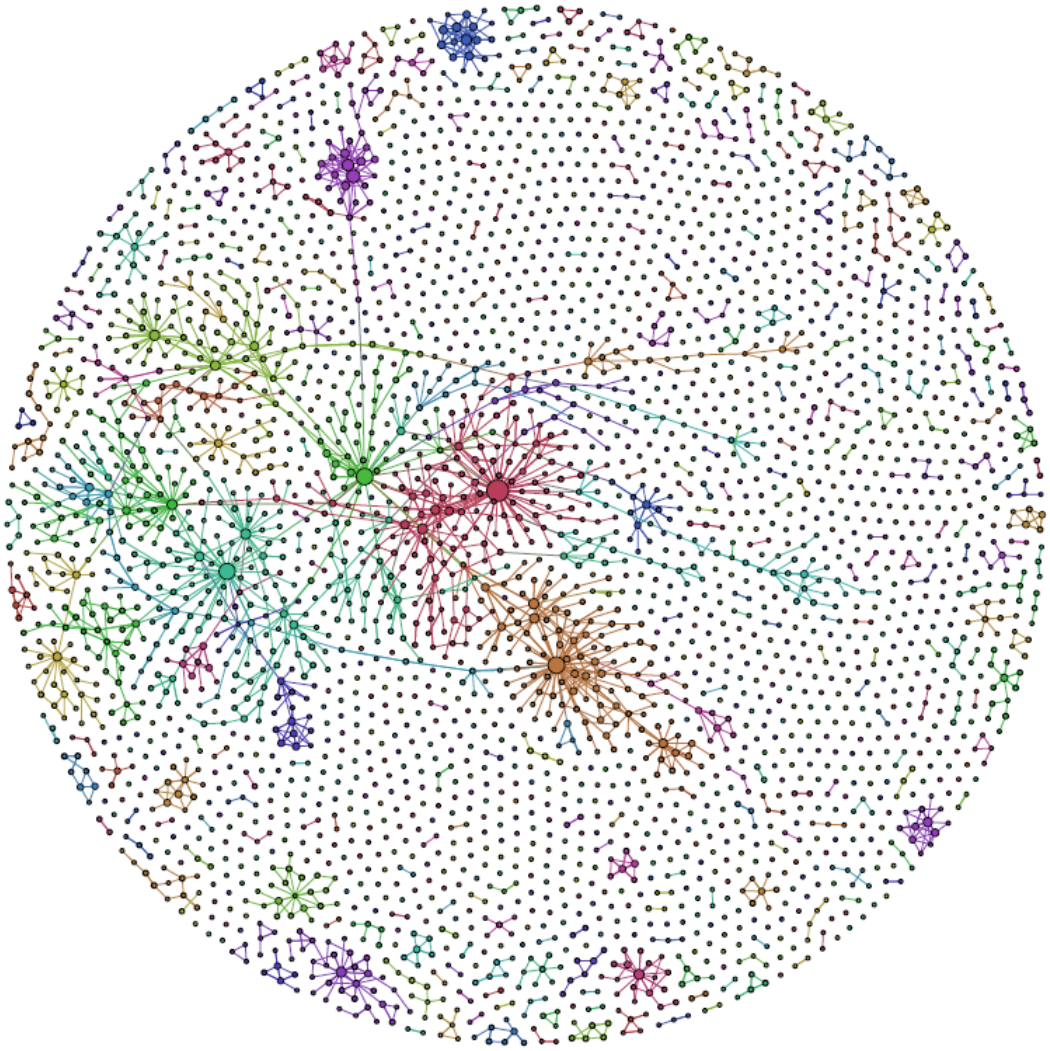


Figura 4.2: Projetos do GitHub com referências cruzadas. O maior componente conexo representa o subgrafo mais conectado, aparecendo no centro do grafo (BLINCOE *et al.*, 2015).

4.2.1 Coleta de Dados

Os códigos fonte dos projetos que compõe o SECO, identificado por (BLINCOE *et al.*, 2015), estão hospedados no Git através de infraestrutura própria ou no GitHub (espelhado também na sua infraestrutura própria).

A fonte de dados selecionada para extração foi o GitHub, por possibilitar que estudos com outros escopos sejam possíveis visto que o GitHub possui mais de 25 milhões de usuários e hospeda mais de 85 milhões de repositórios[8]. Além disso, esta foi a fonte utilizada por BLINCOE *et al.* (2015), possibilitando a reprodução no presente trabalho.

Dada a seleção da fonte de dados, o subprocesso de extração é detalhado a seguir.

[8]https://github.com/about/facts

Subprocesso de Extração de Dados

A extração de dados é uma etapa essencial para o sucesso das análises realizadas utilizando a solução. Na realização do estudo preliminar foi utilizada a forma de extração pela API fornecida pelo GitHub. Na execução desta avaliação a coleta dos dados foi realizada através do GHTorrent, que segundo COSENTINO *et al.* (2017) é a mais utilizada nos trabalhos que estudam software a partir do GitHub. Os projetos analisados na avaliação do sSECO-Process foram extraídos a partir da listagem de projetos que constituem um SECO, identificados por BLINCOE *et al.* (2015) a partir da análise de referências entre repositórios em comentários de usuários.

Sendo assim, a presente dissertação utiliza-se das duas formas mais populares de extração, que combinadas somam mais de 72,5% do conjunto de estratégias utilizadas para extração nos estudos avaliados por COSENTINO *et al.* (2017). Em novas execuções do processo, a abordagem de extração de dados pode ser escolhida, sendo que a coleta através da API disponibiliza dados mais atualizados. Entretanto, soluções de terceiros podem diminuir os problemas da extração através da API, como tratamento dos dados e limites de requisições. Nesta avaliação foi utilizado o *database dump* de novembro de 2017[9], disponibilizado pelo GHTorrent Project, que teve seu início em 2012(GOUSIOS & SPINELLIS, 2012).

Além do *dump* completo da base, o GHTorrent disponibiliza outras formas de acesso aos dados coletados do GitHub. Existe a possibilidade de se realizar queries a MongoDB e MySQL de maneira programática, através de *streaming* das entradas de MongoDB e MySQL, e, por fim, a possibilidade de se realizarem queries MySQL através de uma interface web. Os serviços são oferecidos através de tunelamento SSH, que faz o redirecionamento das requisições ao servidor com os dados coletados.

Essas abordagens possuem como ponto positivo a facilidade na utilização dos dados, visto que não existe custo para acesso aos serviços. Para obter acesso basta realizar uma alteração adicionando sua chave pública SSH ao arquivo correspondente no projeto e realizar um *pull-request*. Outro diferencial da abordagem de utilização dos serviços "ao vivo" do GHTorrent é a possibilidade de acessos em tempo reduzido, pois este acesso é

[9]http://ghtorrent.org/downloads.html

disponibilizado antes que os dados sejam consolidados. Essa abordagem pode ser útil para estudos que observem os dados mais recentes, mas dificulta a reprodução por outros cientistas por não se saber ao certo quais foram os dados avaliados. Dentre as diferentes oportunidades de acesso aos dados do projeto GHTorrent, foi escolhida a abordagem que utiliza o *dump*, por possibilitar a gestão dos recursos, visto que as abordagens *online* possuem limitações de acesso para garantir que todos os usuários possam usufruir dos dados disponíveis. Além disso, não existem garantias da qualidade dos dados, mesmo com os envolvidos no projeto trabalhando de maneira sistemática na identificação e solução de erros. Por fim, e mais importante, a utilização de uma abordagem diferente do *dump* disponibilizado dificulta a oportunidade de replicação dos estudos realizados no presente trabalho por outros cientistas.

O *dump* completo do banco, utilizado para recuperar os dados mais atualizados dos projetos identificados por BLINCOE *et al.* (2015), possui mais de 230 GB de dados. Os dados foram disponibilizados em arquivos csv divididos em 21 tabelas, que incluem dados sobre usuários, *commits*, projetos, comentários e outros, conforme pode ser visualizado no esquema da Figura B.1.

O processo de transferência dos dados presentes no arquivo não é um processo trivial, principalmente devido ao tamanho dos arquivos, o maior arquivo era correspondente a relação entre *commits* e projetos, totalizando 96,9 GB contendo 990.983.516 registros disponibilizados em uma arquivo csv.

Nesta abordagem, o Sistema Gerenciador de Banco de Dados escolhido foi o SQL Server por uma maior familiaridade com o uso, mas outras alternativas podem ser utilizadas sem nenhuma restrição. Assim, foi realizada uma transcrição para ajustar as diferenças entre a linguagem utilizada no SQL Server e do *dump* fornecido, que foi gerado a partir de uma base MySQL. Após a reconstrução do schema foi utilizado o MicrosoftSQL Server Integration Services (SSIS), uma plataforma para criar soluções de integração de dados, incluindo a extração, transformação e carregamento (ETL).

O projeto criado para tratamento e inserção dos dados do *dump* está disponível para reutilização, hospedado no GitHub[10], para auxiliar o processo de restauração dos

[10]https://github.com/hugoguercio/sSECO/

dados disponibilizados pelo projeto GHTorrent em pesquisas futuras. Com a abundância de dados, deve-se tomar cuidado sobre o quão confiáveis eles são. Como boa prática, cientistas identificam ameaças a validade dos seus estudos e das técnicas escolhidas e utilizadas(CAMPBELL & STANLEY, 2015). Para minimizar os perigos de estudo dos dados, faz-se necessário realizar um pré-processamento para encontrar inconsistências e evitar equívocos na interpretação(KALLIAMVAKOU *et al.*, 2016). Na Figura 4.3 é

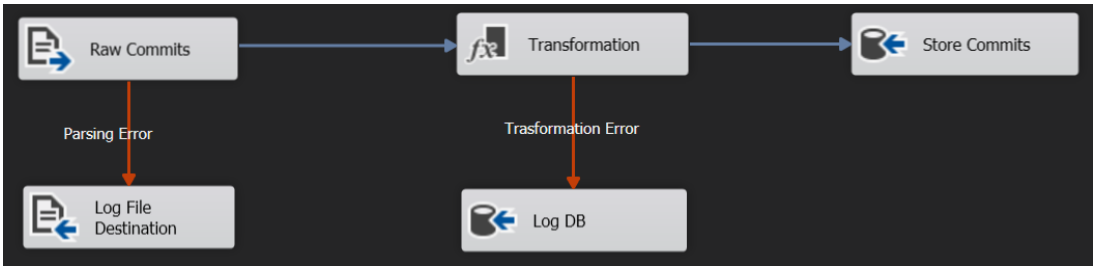


Figura 4.3: Exemplo de tarefa ETL.

apresentada a estratégia de extração, tratamento e armazenagem dos arquivos. A primeira tarefa tem como objetivo extrair os registros armazenados nos arquivos e realizar uma análise com o objetivo de identificar a quantidade de colunas. Caso um registro não seja analisado corretamente ele é transferido para um arquivo de log para análise posterior. Em seguida, a tarefa de transformação estrutura os dados nos formatos correspondentes de cada coluna e, em caso de falha, os registros com erro são armazenados em uma tabela de logs para uma análise posterior. Por fim, a tarefa de carregamento transfere os dados tratados para a base temporária criada anteriormente.

A grande quantidade de dados presentes no GitHub e sua disponibilidade, podem impactar na sua integridade, e, por isso, a atividade de validação dos dados extraídos e pré-processados é de suma importância. Esta validação é realizada na base temporária e, caso alguma inconsistência seja detectada faz-se necessário identificar a origem da inconsistência. De acordo com a origem, a etapa de pré-processamento, extração ou ambas devem ser ajustadas para fazer com que a próxima validação tenha sucesso ou encontre novas inconsistências, até que o arquivo seja verificado pelo desenvolvedor e validado pelo cientista. Com os dados validados, o subprocesso de extração das fontes selecionadas termina com a criação de um *script* que é responsável por traduzir o esquema temporário para o repositório central.

Encerrando a etapa de coleta de dados, é realizada a integração das diferentes fontes extraídas e armazenamento no repositório central, que é utilizado para construção do ambiente de análise. Ao fim da etapa de coleta, foram identificados 239.559 usuários, que colaboraram em 65.379 repositórios através de 349.849 *commits*. O *dataset* também contém informações de comentários, totalizando 641.168 distribuídos entre comentários em *commits*, *pull requests* e *issues*.

4.2.2 Construção do Ambiente de Análise

Esta seção tem como objetivo apresentar as atividades a serem executadas para que o ambiente de análise seja construído. A atividade de construção deste ambiente possui uma sequência de atividades que são descritas a seguir.

Identificação dos elementos de análise

Nesta etapa são apresentadas as estratégias utilizadas para auxiliar na caracterização das entidades envolvidas na avaliação e foi subdividida para apresentar cada um dos elementos de maneira detalhada.

Colaboradores

Existem diferentes atores envolvidos nas atividades de um SECO, sendo que cada um deles pode assumir diferentes papéis. No presente trabalho, os atores foram extraídos a partir dos dados presentes no GitHub, relacionados com os projetos que compõe o SECO identificado por BLINCOE *et al.* (2015). Todos os envolvidos em um determinado projeto são classificados como colaboradores deste projeto.

Em (HATA *et al.*, 2015) os colaboradores são divididos em duas categorias diferentes, sendo elas: (i)*coding contributors*: colaboradores que realizam modificações no código fonte do projeto; (ii)*discussion contributors*: colaboradores que participam das discussões em *issues*, *commits* ou *pull-requests* e não realizam modificações no código fonte do projeto. A partir desta classificação, os autores introduzem uma "pirâmide populacional de software" e avaliam as pirâmides construídas dividindo-as em diferentes tipos de acordo com sua forma. Os autores também realizam avaliações sobre a transição das pirâmides durante o tempo.

O trabalho de PADHYE *et al.* (2014) avalia as contribuições da comunidade em um projeto apenas de acordo com os *commits* realizados. Desta forma, os colaboradores que participam exclusivamente de discussões não foram considerados. Ao avaliar os desenvolvedores de um projeto, cada um deles foi classificado em diferentes categorias, sendo elas:

- ***Core***: Os desenvolvedores desta categoria são todos que possuem permissão de escrita em um determinado projeto, podendo realizar *commits* associados diretamente ao repositório base.

- **Externo**: Os desenvolvedores externos são todos que possuem *commits* associados a repositórios que são *forks* de um projeto. Além disso, para que o desenvolvedor seja considerado externo, pelo menos um *commit* realizado no *fork* do projeto deve ser aceito por um desenvolvedor *core* através de um *pull-request*.

- **Mutante**: Desenvolvedores que realizaram modificações no código base do projeto e que não possuem nenhum *commit* que foi associado ao repositório base.

Ao avaliar as categorias propostas, o motivo da não incorporação do commit para o código base não foi considerada pelo trabalho (PADHYE *et al.*, 2014). Desta forma são considerados desenvolvedores mutantes todos que realizam alterações, sem realizar distinção entre tentativas de colaboração com o projeto base.

Neste trabalho, é proposta uma alteração na classificação dos desenvolvedores mutantes, adicionando uma nova categoria de usuário, denominada "candidatos". Os desenvolvedores candidatos são desenvolvedores que possuem alterações no código-fonte de um projeto e que tiveram seus *pull-requests* recusados pelos desenvolvedores *core*. Essa nova categoria de desenvolvedor foi definida, pois a tentativa de contribuir com o projeto base deve ser avaliada, visto que ao iniciar um *pull-request* o desenvolvedor deixa de maneira explícita seu desejo em colaborar com a comunidade. A não aceitação da contribuição pode ter diferentes motivos. Em (GOUSIOS *et al.*, 2014) são avaliadas as principais razões para não aceitação de um *pull request*. Os motivos mais frequentes encontrados pelos autores são a existência de outro *pull request* que resolve o problema de uma maneira melhor, a implementação da funcionalidade estar incorreta, inexistente

ou fora dos padrões de projeto, e problemas na execução do processo da *pull request* no projeto.

Desta forma, no atual trabalho, são consideradas quatro categorias de desenvolvedores: core, externos, candidatos e mutantes; sendo mutantes apenas os desenvolvedores que não realizaram nenhuma tentativa de incorporação das suas modificações ao código fonte do projeto base.

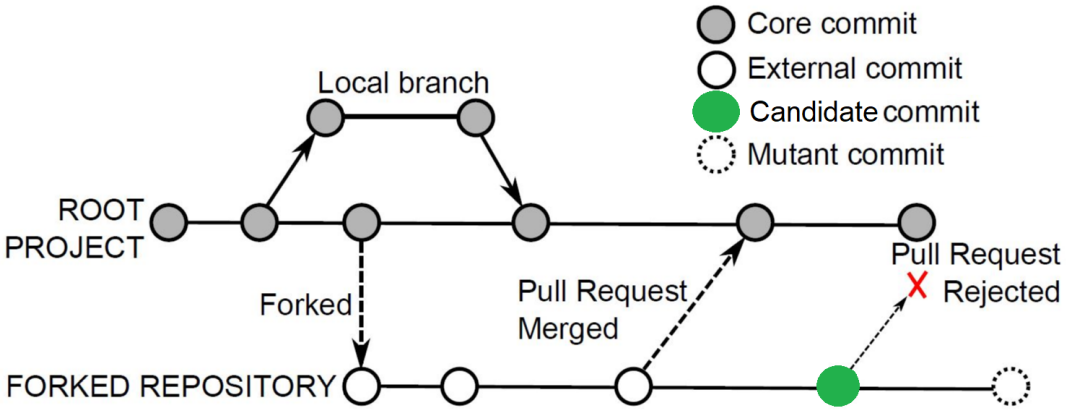


Figura 4.4: Linha do tempo simplificada de um repositório e um fork, assim como classificação dos commits. Adaptada de PADHYE *et al.* (2014)

A Figura 4.4 é uma adaptação do trabalho de PADHYE *et al.* (2014), com a adição do novo papel, mostrando a diferença entre os perfis de mutante e candidato.

Projetos

O modelo de desenvolvimento orientado a *pull requests* representa um novo método para colaboração no desenvolvimento distribuído de software. Neste modelo, nem todos os possíveis colaboradores tem permissão para fazer alterações no código fonte, ao invés disso os colaboradores criam um *fork* do repositório raiz e fazem suas alterações de maneira independente. No momento que o colaborador entende que suas alterações estão prontas para serem incorporadas no repositório principal ele inicia uma *pull request*. Este *pull request* será gerenciado pela equipe *core* do projeto e caso as alterações estejam de acordo com as diretrizes do projeto elas podem ser aceitas e incorporadas.

De acordo com esse modelo de desenvolvimento, os repositórios podem ser divididos em dois tipos, sendo repositórios base e repositórios *fork*. Toda atividade em repositórios *forks* são registradas de maneira separada.

Desta forma, foram tratados como projetos o conjunto de repositórios que com-

preende um repositório base e todos os repositórios que representam um *fork* do repositório base. Para contabilização das atividades de um projeto, todas as colaborações realizadas no código base, e todas as atividades realizadas em repositórios *fork* que foram incorporadas no repositório base foram consideradas.

KALLIAMVAKOU *et al.* (2016) destacam que os repositórios *fork* podem atuar de maneira independente do projeto, como, por exemplo, com customizações que não possuem intenção de incorporação no projeto base.

Aplicar Filtros

A literatura aponta que existem riscos, como em (HOWISON & CROWSTON, 2004) que avalia perigos e armadilhas na utilização do SourceForge[11], um dos primeiros sites que disponibilizava a hospedagem de códigos fontes, adotado de maneira consistente antes da adoção em larga escala do GitHub. Neste trabalho foi observado que os dados poderiam estar contaminados por sistemas de armazenamento anteriores, também foi identificado que os dados eram hospedados em estruturas auxiliares, tornando-os incompletos.

Um trabalho que segue na linha de avaliação de perigos na utilização de repositórios para entendimento da atividade de se desenvolver software é o de KALLIAMVAKOU *et al.* (2016), que avaliam de forma específica o GitHub e os perigos envolvidos na sua utilização para aquisição de conhecimento. Os autores identificaram 13 possíveis perigos, sendo eles relacionados aos projetos, *pull request*s, usuários e à plataforma.

Segundo as considerações de KALLIAMVAKOU *et al.* (2016), alguns filtros foram implementados com o objetivo de reduzir os perigos relacionados aos estudos a partir do Github.

Filtro 1: Considerar atividade nos repositórios base e forks.

O modelo de desenvolvimento baseado em *pull requests*, utilizado em diversos projetos do GitHub, é um novo método para colaboração em software desenvolvido de maneira distribuída. Neste modelo, os potenciais colaboradores de um projeto não necessariamente possuem permissão para realizar alterações nos arquivos do projeto. Para colaborarem com o projeto, os potenciais colaboradores criam um clone do repositório

[11] https://sourceforge.net/

alvo, tendo total acesso a todos os arquivos. Após as alterações realizadas no clone, o desenvolvedor pode, ou não, submeter uma solicitação de merge entre o seu *fork* e o repositório principal. Desta forma, as alterações realizadas em um projeto não necessariamente estarão contidas no repositório principal, entretanto tais contribuições também são relevantes, pois contam como atividade relacionada ao projeto.

Filtro 2: Seleção de projetos com atividade relevante.

No GitHub existe uma grande quantidade de projetos com pouca atividade. Isso pode ser observado pela quantidade de *commits* nos projetos, que possuem como mediana 6 *commits* conforme verificado por KALLIAMVAKOU *et al.* (2016). Os autores verificaram que 2.5% dos projetos mais ativos correspondem à quantidade de *commits* do restante dos outros projetos hospedados no GitHub. Para filtrar apenas os projetos com atividade relevante foram considerados os projetos com mais de 198 *commits* em todo o período de observação. Este valor foi escolhido pois a mediana dos valores dos projetos identificados teve o valor 9. Dados os 22 trimestres avaliados, determinou-se um projeto como ativo como a soma da mediana para cada um dos trimestres. Na literatura, não foram encontrados indicadores de atividade do projeto de forma explícita, mesmo que existam trabalhos com esse objetivo, como (TREUDE *et al.*, 2015) ou em (LI *et al.*, 2016; ONOUE *et al.*, 2009) que avaliam a atividade pela perspectiva dos desenvolvedores.

Filtro 3: Repositórios com atividade recente.

Com a grande quantidade de repositórios com pouca atividade, é consequência que vários projetos se tornem inativos. Muitos projetos podem ter atividade recente, mas observar apenas a atividade recente também não é suficiente visto que repositórios recém criados terão atividades recentes registradas, mas não necessariamente se tornarão projetos ativos.

De acordo com o *dataset* analisado, 1945 repositórios possuem ao menos um *commit* durante os últimos 12 meses, compreendidos entre 21/09/2016 e 21/09/2017, sendo que 74,55% destes repositórios foram criados neste intervalo de tempo. Observando os 55662 repositórios criados antes dos 12 meses avaliados, apenas 494 possuem atividade nos últimos 12 meses, o que mostra que grande parte dos repositórios se encontram inativos.

Para mensurar a atividade dos repositórios, foram comparadas as datas de criação

e do último *commit* registrado. A partir dessa avaliação, temos que 61,93% dos repositórios possuem apenas 1 dia de atividade, o que segundo KALLIAMVAKOU *et al.* (2016) pode sugerir que os repositórios foram utilizados para teste ou com o propósito de armazenamento. Observando a atividade dos repositórios, encontrou-se que 23,36% possuem até um mês de atividade, 13,05% chegam a seis meses e apenas 8,07% dos repositórios possuem atividade superior a um ano.

Com o objetivo de verificar se a atividade de um repositório possui relação com seu tipo (base/fork) foi realizada uma análise de variância (ANOVA) de um fator para testar a hipótese de que as médias de dias de atividade entre os dois tipos de repositórios são iguais.

O valor da probabilidade de significância para a ANOVA do tipo de repositório foi menor que 0,05, indicando que a média de dias ativos difere estatisticamente, mas o modelo não pode ser extrapolado para outros cenários, pois o valor R2 foi predito baixo (23,74%) indicando que o modelo é impreciso para novas observações.

Esta análise teve como objetivo identificar se mais um filtro era necessário, sendo este novo filtro com relação ao tipo de repositório ser fork ou base. Como o modelo não pôde ser extrapolado para outros cenários, essa nova estratégia de filtragem não foi utilizada.

A partir do filtro de atividade recente, foram considerados apenas projetos com pelo menos uma contribuição nos últimos seis meses de observação, totalizando 59 projetos.

Filtro 4: Projetos que não são individuais.

O GitHub possui diferentes funcionalidades que destacam o aspecto social[12] no desenolvimento de software, como: seguir outros usuários para receber notificações sobre suas atividades, observar repositórios para receber notificações sobre atualizações ou diversos canais para troca de mensagens, como em *pull requests* ou *issues*. Mesmo com diversas características sociais, vários repositórios possuem poucos *commiters*, indicando que são repositórios pessoais com finalidades de armazenamento ou experimentações diversas.

De acordo com o *dataset* avaliado, 82% dos repositórios possuem 1 *committer*,

[12]https://help.github.com/articles/be-social

12,93% possuem 3 ou menos. Isso mostra que grande parte dos projetos não exploram as características sociais presentes na plataforma, visto que grande parte dos repositórios são de uso individual.

Para observar os repositórios que possuem mais interações, foram considerados os projetos com mais de 10 colaboradores distintos, assim como em (YAMASHITA *et al.*, 2014) que o fez para reduzir ruídos nos resultados.

Filtro 5: quantidade de pull requests

A utilização de *pull-requests* é justificada apenas em projetos colaborativos, visto que os *pull-requests* são úteis somente nos casos onde múltiplos desenvolvedores realizam contribuições ao código fonte do projeto.

Dentre os 65379 repositórios do GitHub presentes no *dataset*, 3593 possuem múltiplos colaboradores, sendo que dentre esses apenas 1029 utilizaram o modelo de *pull-request* ao menos uma vez. Desta forma, apenas projetos com ao menos um *pull request* foram considerados.

Filtro 6: Identificação do status de pull requests

Após a avaliação de uma solicitação de um *pull request*, em caso de aceitação pelo colaborador responsável, existem diferentes maneiras de incorporar a colaboração realizada no código fonte do projeto, permitindo no mínimo três estratégias de merge, sendo elas:

- Usando o botão *Merge* através da interface web do GitHub.
- Usando git, realizando um merge entre o repositório raiz e a branch de um *pull request*
- Criando um *patch* textual entre o *pull request* e o repositório raiz para depois aplicá-lo ao master *branch*. Essa técnica também é conhecida como *commit squashing* e consiste na consolidação de mútliplos *commits* em um único antes do *merge* da *pull request*.

A quantidade de registros históricos sobre as ações de um determinado *pull request* é influenciada pela ténica utilizada, visto que nem todas são detectadas pelo GitHub. Sendo assim, contribuições que tenham sido aceitas pelos responsáveis do projeto podem

não ter histórico armazenado, fazendo-se necessárias técnicas para identificação do real situação de um *pull request*.

Os *pull requests* que foram aceitos fora do GitHub podem ser identificados através de conjuntos de heurísticas. GOUSIOS *et al.* (2014) apresentam um conjunto de quatro heurísticas para auxiliar na identificação dos *merges* não registrados explicitamente no *dataset*. A forma de identificação do *pull request* não é relevante para este trabalho, desta forma algumas herísticas foram unificadas com relação ao trabalho de GOUSIOS *et al.* (2014).

H_1 Ao menos um dos *commits* na *pull request* está disponível na *master branch* do projeto alvo.

H_2 O último comentário realizado antes do fechamento do *pull request* possui um valor verdadeiro ao ser avaliado pela expressão regular (Equação 4.1) que visa capturar os casos onde o fechamento de *issues* ou *pull requests* foram realizados através de conteúdos nas mensagens trocadas. Exemplificando, caso um *pull request* numerado 147, e a descrição ou mensagem contenha *merging#147*, assim que ele seja incorporado na *branch* principal o *pull request* será fechado.

$$(?:merg|appl|pull|push|integrat)(?:ing|i?ed) \tag{4.1}$$

A utilização das heurísticas foi importante para garantir que o filtro que considera os *pull requests* seja coerente com a realidade do projeto. Com as heurísticas aplicadas, 12 projetos foram incluídos na análise por possuírem apenas *pull requests* identificados através da heurística.

Tabela 4.1: Tabela relacionando filtros e a quantidade de projetos que satisfez cada um dos filtros.

Filtro	**Quantidade de Projetos**
2 - Com atividade relevante	76
3 - Ativos nos últimos 6 meses	59
4 - Mais de 10 colaboradores	134
5 - Utilização de *pull requests*	122

Após aplicar os filtros no *dataset* utilizado, foram identificados 40 projetos que atendem a todos os filtros. A Tabela 4.1 apresenta a quantidade de projetos que atendeu

a cada filtro de maneira específica. Em destaque, o filtro 1, que considera a quantidade de colaboradores distintos não retirou nenhum projeto do escopo. Este resultado era esperado visto que o *dataset* utilizado para análise representa um SECO, identificado em (BLINCOE *et al.*, 2015), que tem projeto central o projeto joyent/node.

Estratégia de Amostragem

Após a aplicação dos filtros, foi realizada a definição da técnica de amostragem. Esta etapa é muito importante, pois, se os dados amostrais não forem coletados de maneira apropriada, os dados podem se tornar completamente inúteis e mesmo com diferentes tratamentos estatísticos, não poderá ser salva TRIOLA (2006).

No presente estudo, foi utilizada a técnica de amostragem intencional, onde os elementos da amostra são selecionados a partir do conhecimento da população e do propósito do estudo.

Selecionar Métricas

Esta etapa tem como objetivo delimitar o escopo do ambiente de análise com relação às métricas existentes na literatura. Existem diferentes métricas disponíveis que podem auxiliar os cientistas a alcançarem seus objetivos, e essa atividade é realizada para que o cientista identifique as métricas que se relacionam com os objetivos da pesquisa. A seguir as métricas selecionadas para esta avaliação são apresentadas e foram divididas de acordo com a entidade que estão relacionadas.

Projetos

- **Magnet**: Esta métrica tem como objetivo capturar o quão "magnético" um determinado projeto é, onde o magnetismo do projeto é a proporção de colaboradores que contribuíram em um período de tempo p_i, mas que não colaboraram em um período anterior p_{i-1}, indicando novos usuários atraídos para o projeto.

  Esta métrica foi proposta no estudo realizado em YAMASHITA *et al.* (2014) e aprimorada em um trabalho posterior YAMASHITA *et al.* (2016) pelo mesmo grupo de autores e é tratada neste trabalho como MAG.

Nos dois trabalhos a métrica é apenas explicada através de exemplos. A partir das explicações, a métrica foi definida pela Equação 4.2, onde C indica um conjunto de colaboradores e p_i se refere ao período alvo e p_{i-1} um período anterior.

$$MAG = \frac{\left\|C_{p_i} - C_{p_{i-1}}\right\|}{\left\|C_{p_i} \bigcup C_{p_{i-1}}\right\|} \tag{4.2}$$

Desta forma, MAG é o resultado da fração entre a quantidade de novos colaboradores sobre o total de colaboradores nos dois períodos observados.

Caso o projeto não possua nenhum colaborador nos períodos observados, o valor de MAG será zero. Assim como no trabalho de referência que apresenta a métrica, projetos com menos de 10 colaboradores foram desconsiderados para reduzir ruídos no resultado. Este também foi o critério adotado no filtro 4, descrito anteriormente.

- **Sticky**: Assim como a métrica MAG, esta métrica também foi apresentada por YAMASHITA *et al.* (2014, 2016) e tem como objetivo capturar o quão "pegajoso" um determinado projeto é. Esta métrica tem como objetivo avaliar a quantidade de novos participantes no projeto.

  A partir da explicação fornecida pelos autores, foi elaborada a Equação 4.3, que foi utilizada para calcular *sticky* de um projeto, que é tratada neste trabalho como PEG, onde C indica um conjunto de colaboradores e p_i se refere ao período alvo e p_{i-1} um período anterior.

$$PEG = \frac{\left\|C_{p_i} - C_{p_{i-1}}\right\|}{\left\|C_{p_{i-1}}\right\|} \tag{4.3}$$

  Desta forma, PEG é o resultado da fração dos colaboradores que colaboraram no período p_i sobre o total de colaboradores do período anterior p_{i-1}.

  Caso p_i ou p_{i-1} compreendam o início do projeto, o valor de PEG será zero por não existir um período a ser utilizado como base de cálculo. Assim todos os colaboradores de p_i serão novos colaboradores visto que não existem colaboradores em p_{i-1}. Conforme os trabalhos de YAMASHITA *et al.* (2014, 2016), que apresentam a métrica, projetos com menos de 10 colaboradores foram desconsiderados para

reduzir ruídos no resultado. Este foi o mesmo critério adotado no filtro 4.

- **Popularidade**: Existem diferentes formas de se calcular a popularidade dos projetos. Ao avaliar os projetos hospedados no github, diversos trabalhos utilizam como base a quantidade de *stars* e *watchers*. Quando um usuário se torna um *watcher* de um projeto, ele é notificado sobre as conversas realizadas no ambiente do projeto. Quando um usuário marca um projeto como *starred*, o projeto ganha destaque e projetos similares podem ser recomendados no *feed* do usuário. O uso de *stars* é uma forma de mostrar a aprovação e reconhecimento para a comunidade, sendo realmente um grande indicador de popularidade.

  Os trabalhos de RASTOGI & NAGAPPAN (2016); BORGES *et al.* (2016) utilizam apenas a quantidade de *stars* de um projeto como indicador de popularidade. Em JARCZYK *et al.* (2014) a quantidade de *stars* também é a única variável utilizada para capturar a popularidade de um projeto, mas é realizada uma transformação logarítimica, $x' = log_{10}(x + 10)$ antes do seu uso como métrica de atratividade e popularidade.

  Em (AGGARWAL *et al.*, 2014) o número de estrelas também é utilizado para obtenção de um indicador da popularidade de um determinado projeto. Inspirados pelo trabalho de (DABBISH *et al.*, 2014) que realizou entrevistas profundas, com usuários centrais e periféricos do GitHub, com o objetivo de auxiliar no entendimento da transparência e colaboração em repositórios de software livre. Na seção de gestão de contribuições de código, múltiplos pesquisados afirmaram que lidavam com vários *pull requests* diariamente. Ao considerar isso, AGGARWAL *et al.* (2014) propõem que a popularidade de um projeto possui influência da quantidade de estrelas, *forks* e *pull requests*, de acordo com a equação Equação 4.4

$$POP_{p2} = (\#Stars) + (\#Forks) + (\#Pulls)^2 \tag{4.4}$$

  Segundo os autores, a medida valoriza a quantidade de *Pulls*, visto que elas representam a quantidade de colaborações bem sucedidas e aceitas pela comunidade. A quantidade de estrelas possui a mesma importância da quantidade de *Forks*, que

foram considerados por representar a intenção de modificar o código.

Usuários

- **Popularidade**: A popularidade de um usuário está associada ao reconhecimento de um colaborador nas comunidades em que ele está inserido. BADASHIAN *et al.* (2014) apresentam um indicador para representar a popularidade, POP_u, de um colaborador, definida pela Equação 4.5.

$$POP_u = \#followers + \#mentions \tag{4.5}$$

O indicador proposto em (BADASHIAN *et al.*, 2014) apresenta uma visão geral do reconhecimento do usuário a partir de toda base de usuários e projetos. Ao avaliar os projetos individualmente, a medida apresentada pode ser distorcida, visto que um colaborador muito popular pode realizar uma contribuição pontual em um projeto mais distante das comunidades que o tornaram popular.

Partindo desse pressuposto, que um indivíduo pode ter um reconhecimento maior nas comunidades em que se encontra muito ativo, uma nova medida de popularidade é proposta neste trabalho, com o objetivo de capturar o reconhecimento da comunidade com relação a um projeto específico. A nova medida é apresentada na Equação 4.6, referente à popularidade dos colaboradores por projeto, POP_{up} é calculada através do agregado da quantidade de *followers* F que um determinado colaborador u possui, sendo que esses *followers* devem ter alguma contribuição C no projeto p, com a quantidade de *mentions* MO em comentários relacionados do projeto.

$$POP_{up} = \left\| F_u \bigcap C_p \right\| + \left\| M_u \bigcap MO_p \right\| \tag{4.6}$$

- **DEV**: A métrica DEV tem como objetivo mensurar o grau de comprometimento dos colaboradores em atividades de desenvolvimento. Ela foi apresentada por BADASHIAN *et al.* (2014), onde os autores definem o indicador DEV, que é obtido através da soma de $log(commits)$ com a quantidade de *pull requests* iniciados e

tratados pelo desenvolvedor, de acordo com a Equação 4.7. Essas atividades foram combinadas visto que todas podem ocasionar mudanças no código fonte do projeto.

$$DEV = log(Commits) + \#PullRequets + \#PullRequestsHandled \quad (4.7)$$

Os valores de *commits* estão em escala logarítmica visto que *pull requests* são resultados de múltiplos *commits* e, segundo avaliação dos autores, o valor médio desta variável é dez vezes maior que o das outras métricas.

Definições de Análise Temporal

Nesta avaliação, a decisão foi de realizar uma análise temporal, com o objetivo de capturar os aspectos da evolução do ecossistema. Com o objetivo de delimitar o período de observação foi realizada uma avaliação com relação à quantidade de *commits* assim como o total de repositórios distintos em que esses *commits* foram registrados. Estes dados são apresentados de maneira detalhada na Tabela B.1 e através da Figura 4.5, que apresenta os valores em escala logarítmica.

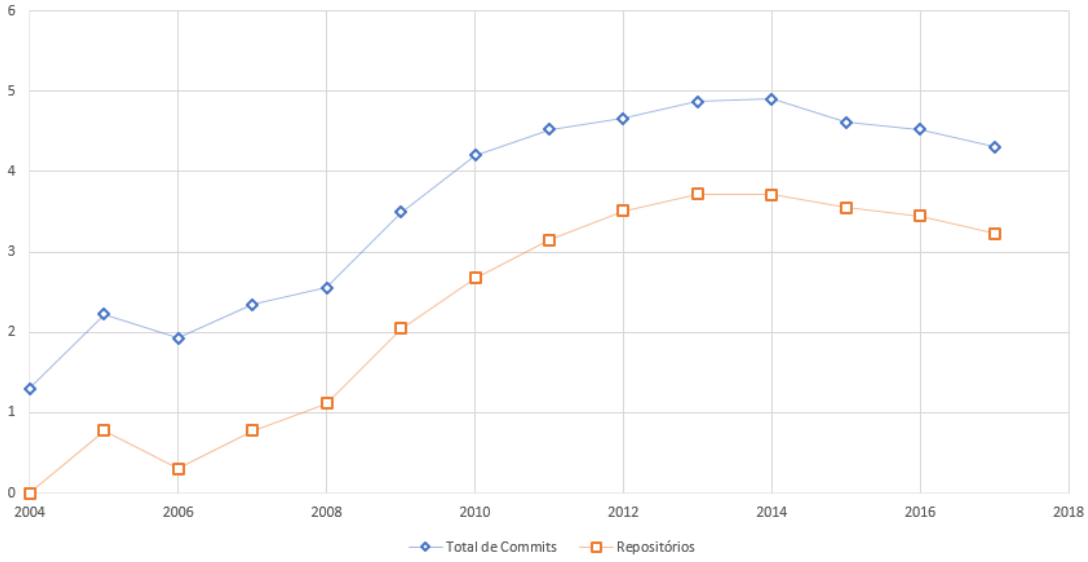


Figura 4.5: Total de *commits* e repositórios distintos através dos anos.

Após a avaliação do *dataset*, foi definido que o período avaliado seria a partir do início do ano de 2012 até o segundo trimestre de 2017. O período de observação foi fatiado em trimestres, assim como em (CONSTANTINOU & MENS, 2017), e no total foram analisados 22 trimestres.

Aplicações das Métricas Selecionadas

Após a seleção das métricas e da decisão se a análise será temporal ou não é possível aplicar as métricas selecionadas. É importante destacar que caso a análise seja temporal, as métricas devem ser aplicadas para cada um dos intervalos de tempo definidos, para que seja possível avaliar o comportamento das métricas com o passar do tempo, possibilitando uma análise evolutiva das métricas selecionadas.

As métricas calculadas nesta etapa foram disponibilizadas através do *dataset* do ambiente de análise disponibilizado através do GitHub[13].

Construção de Redes Complexas

A construção do ambiente para análise dos dados através de redes complexas foi condensada em um subprocesso com o objetivo de aumentar a legibilidade, agrupando as atividades básicas necessárias para concepção e construção das redes.

Definição dos Nós

Esta atividade visa identificar quais elementos de análise serão representados através das redes complexas. A partir da definição dos nós, começa-se o entendimento do que será avaliado através das redes. Uma forma de avaliar quais entidades podem ser representadas por redes é considerar as formas de mensurar seus relacionamentos. Durante a avaliação realizada, foram modelados como nós os projetos e colaboradores.

Os atributos que cada entidade possui também deve ser levado em consideração, visto que eles podem auxiliar em resultados mais refinados facilitando a análise. É possível, por exemplo, considerar métricas de popularidade para ajustar o tamanho dos nós em sua apresentação ou utilizar cores para representar diferentes papéis que as entidades possam assumir.

Definição das Arestas

A atividade de definição de arestas tem como objetivo representar os relacionamentos existentes entre os elementos de análise identificados. As arestas que representam os relacionamentos também podem conter atributos para complementar as análises. Dentre os atributos que uma aresta pode possuir, o de maior destaque é o peso. A modelagem

[13]https://github.com/hugoguercio/sSECO/

das arestas define qual o nível de refinamento que se deseja com uma rede complexa, assim como quais as análises serão possíveis após a modelagem.

Exemplificando, se existe um relacionamento entre dois projetos, isto representa a existência de colaboradores em comum entre estes projetos. Desta forma, caso dois projetos possuam um colaborador em comum eles seriam relacionados por esta aresta. Realizando a modelagem desta forma, é possível observar projetos que se relacionam a partir dos seus colaboradores, mas não é possível mensurar quais projetos estão mais densamente conectados. Outra forma de modelar o mesmo relacionamento é considerar como o peso da aresta a quantidade de colaboradores em comum entre os projetos. Desta forma, quanto mais colaboradores em comum, maior será o peso da aresta, possibilitando a avaliação de quais projetos possuem mais colaboradores em comum.

Tabela 4.2: Relacionamentos identificados entre os nós modelados

Entidades relacionadas	**Relacionamento existente**	**Peso**
Projeto - Projeto	Colaboradores Comuns	Quantidade de Colaboradores
Projeto - Usuário	Comentário	Quantidade de Comentários
Projeto - Usuário	*commit*	Quantidade de *commits*
Projeto - Usuário	Role	N/A
Projeto - Usuário	Contribuição	Medida
Projeto - Usuário	Watch	N/A
Projeto - Usuário	Fork	Quantidade de *forks*
Projeto - Usuário	Reporter	Quantidade de *issues* reportados
Usuário - Usuário	Follow	N/A
Usuário - Usuário	Relacionamento por Contribuição	Medida

A Tabela 4.2 apresenta, de maneira sintetizada, os relacionamentos existentes entre os nós modelados. Estes relacionamentos podem ser utilizados para instanciar diferentes redes complexas que foram identificadas a partir dos dados presentes no *dataset* avaliado. A primeira coluna apresenta os relacionamentos entre as entidades, mostrando relacionamentos entre diferentes entidades assim como entre entidades do mesmo tipo. Também são apresentadas formas de atribuir peso aos relacionamentos, mas é possível construir redes utilizando os relacionamentos descritos e mantendo um valor padrão para o peso.

Instanciação das redes

A atividade de instanciação das redes tem como objetivo construir o ambiente inicial para análise do cientista. Existem diferentes soluções que podem auxiliar na ins-

tanciação da rede, compreendendo desde pacotes adicionais para o Excel[14], passando por bases em grafos até softwares específicos para esta finalidade, como por exemplo Cytoscape[15] ou Gephi[16].

É importante que a escolha da ferramenta contemple as decisões anteriores, e caso a análise desejada contemple questões temporais é importante escolher uma ferramenta que possua funcionalidades de redes complexas dinâmicas. Durante a avaliação, a instanciação foi realizada através do Gephi, uma solução *open-source* que auxilia tanto na visualização quanto no cálculo de medidas e estratégias de distribuição dos nós.

Para instanciação, foram gerados dois arquivos[17] para importação na ferramenta, sendo eles uma lista de nós e uma listagem de arestas. Além das informações básicas dos nós, como seu ID e descrição, é possível utilizar os atributos calculados para cada um. A lista de colaboradores por exemplo, leva consigo seus atributos de popularidade e DEV, calculados para cada um dos intervalos de observação.

Dentre as redes instanciadas, o destaque é dado para as redes complexas que apresentam relacionamentos sociais de forma explícita. A primeira rede complexa com destaque é a rede de relacionamentos, criados através das interações dos usuários medidas pela Equação A.3, que define o relacionamento proposto por GUÉRCIO *et al.* (2018).

Aplicação das Métricas de Redes Complexas

A aplicação das métricas referentes às redes complexas é a última etapa. Após a instanciação das redes na ferramenta, devem ser medidas as métricas de redes complexas. A aplicação dessas métricas foram realizadas com o auxílio da ferramenta de instanciação, mas a partir dos dados armazenados no ambiente de análise é possível extrair medidas indisponíveis nas ferramentas. Os dados consolidados das métricas foram disponibilizados no GitHub.

Foram aplicadas diferentes medidas de centralidade, dentre elas as centralidades de *closeness*, *betweenness*, *eigenvector*, de grau e grau ponderado. Além das medidas de centralidade também foram calculados o coeficiente de agrupamento dos nós e a classe de modularidade.

[14]https://archive.codeplex.com/?p=nodexl
[15]http://www.cytoscape.org/
[16]https://gephi.org/
[17]https://github.com/hugoguercio/sSECO/

A classe de modularidade foi calculada com o objetivo de particionar os colaboradores em comunidades. A rede complexa utilizada para medir a classe de modularidade foi a rede de relacionamento por contribuição, considerando todo o período de tempo observado no *dataset* selecionado para avaliação do sSECO-Process. A modularidade da rede foi calculada de acordo com o algoritmo apresentado em BLONDEL *et al.* (2008). O parâmetro de resolução utilizado foi 2, e o resultado foram 11 comunidades, onde as 5 comunidades com menos colaboradores representam menos de 1% do total de colaboradores presentes no *dataset*.

Tabela 4.3: Classes de modularidade e quantidade de colaboradores. As cores das células são referentes às visualizações geradas, que consideram as classes de modularidade para ajudar no reconhecimento de comunidades nas redes complexas.

Classe de Modularidade	**Colaboradores**	**Percentual**
5	2246	28,9
0	2142	27,56
8	1295	16,66
2	1015	13,06
1	726	9,34
6	288	3,71
4	46	0,59
7	4	0,05
9	4	0,05
3	3	0,04
10	2	0,03

A Tabela 4.3 apresenta a quantidade de colaboradores classificados em cada uma das classes de modularidade. A partir da avaliação das classes de modularidade, é possível notar que 5 das 11 comunidades identificadas possuem menos de 1% do total de colaboradores presentes na rede complexa. As classes de modularidade dos 6 módulos com mais colaboradores foram avaliados com maiores detalhes. Os projetos de cada colaborador pertencente a estes módulos foram analisados com o objetivo de identificar projetos em comum entre os grupos.

Em todas as classes de modularidade foram identificados projetos que poucos participantes estavam relacionados, e por este motivo foi definido um limite inferior para análise dos projetos, sendo este igual a 10% da quantidade de colaboradores da menor classe de modularidade avaliada. Desta forma, apenas projetos com 29 ou mais colabora-

dores distintos em uma classe foram considerados.

Dentre os mais de 7.712 usuários considerados na análise, foram identificados 71 projetos principais, que se destacam no conjunto de classes de modularidade. Destes projetos, apenas 11 apareceram em múltiplas classes de modularidade, e eles relacionaram as classes 0, 5 e 8. Essa interseção também pode ser vista através das visualizações geradas, em especial na Figura 4.6, que será apresentada a seguir. A separação das classes de modularidade também pode ser vista na mesma figura, onde as classes 2, 1 e 6 estão separadas do restante da rede apresentada, mesmo existindo laços que as relacionem.

A partir desta avaliação, existem indícios que as classes de modularidade estão coerentes com a participação dos usuários nos projetos. A existência dessa sobreposição entre as classes de modularidade 5 e 8 apenas mostram que estas duas comunidades possuem uma forte relação, evidenciada por conjuntos de colaboradores classificados em cada uma delas que possuem interesses e contribuições em projetos comuns. Essas e outras análises estão detalhadas na Subseção 4.2.3.

4.2.3 Análise

Esta seção tem como objetivo apresentar as atividades a serem executadas durante a análise dos dados presentes no ambiente de análise. A sequência de atividades realizadas para análise são descritas a seguir.

Definir Objetivos

A primeira atividade realizada na análise é a definição dos objetivos de forma detalhada. Durante as etapas anteriores, os objetivos de maneira genérica já são considerados, pois impactam diretamente na obtenção e construção do ambiente de análise.

O principal objetivo durante a avaliação do sSECO-Process foi o de representar a dimensão social presente no ecossistema contido no *dataset* utilizado. Outros objetivos secundários foram estabelecidos, sendo os seguintes:

- Modelar as redes complexas existentes no SECO
- Instanciar as redes complexas que explicitem a dimensão social do SECO

- Capturar aspectos evolutivos do SECO
- Identificar os principais colaboradores dos projetos que compõem o SECO

Seleção das Redes Complexas

Durante a construção do ambiente de análise, diferentes redes complexas foram modeladas e sintetizadas na Tabela 4.2. Dentre as redes complexas modeladas existem diferentes relacionamentos, e foram selecionadas para análise as redes complexas que relacionam os usuários, pela maior aderência ao objetivo desta avaliação.

As outras redes complexas modeladas e presentes no ambiente de análise não foram selecionadas nesta avaliação, pois o foco do estudo é a dimensão social.

Análise de Redes Complexas

A partir da seleção das redes complexas, faz-se necessário analisar cada uma e verificar se elas podem auxiliar o cientista a atingir o(s) objetivo(s) estabelecido(s). A análise é realizada para cada uma das redes selecionadas na atividade anterior e, devido à quantidade de atividades, foi decomposta em um subprocesso. As atividades do subprocesso para as duas redes selecionadas nesta avaliação são descritas a seguir.

Instanciação da Rede Complexa

A instanciação das redes complexas é realizada com o auxílio de aplicações escolhidas pelo cientista. Como um dos objetivos da avaliação é de capturar aspectos evolutivos do SECO, a rede complexa modelada é do tipo dinâmica, onde suas características variam de acordo com o tempo. Para instanciação da rede dinâmica, foram exportados em formato gexf as redes complexas por trimestre, e estes arquivos foram combinados, fazendo com que ao final fosse possível instanciar uma rede que apresenta os dados de maneira sensível ao tempo.

Os arquivos das redes instanciadas, assim como as listas de nós, arestas e as métricas calculadas foram disponibilizadas no GitHub[18].

Definição das Estratégias de Distribuição

A estratégia de distribuição dos nós é uma atividade muito importante, por ter

[18]https://github.com/hugoguercio/sSECO/

um alto impacto na forma como os dados são apresentados para a análise visual. As estratégias de distribuição podem estar relacionadas a quantidade de elementos que se deseja apresentar ou à forma como eles se relacionam. A utilização de uma estratégia inadequada pode impactar negativamente nas análises, como, por exemplo, a estratégia OpenOrd(MARTIN *et al.*, 2011) que tem como objetivo destacar *clusters*, que se comporta bem para grafos com uma quantidade grande de nós e espera arestas direcionadas e sem peso. Caso esta estratégia seja utilizada para representar redes complexas que não sigam esses pré-requisitos, as análises posteriores não serão confiáveis.

Geração das Visualizações

Após a instanciação da rede e da seleção da estratégia de distribuição é possível gerar visualizações das redes complexas. Os passos anteriores são de extrema importância para que a visualização possa auxiliar o cientista a alcançar seus objetivos.

Figura 4.6: Force Atlas todos os trimestres.

Uma mesma rede instanciada pode ter diferentes visualizações, desta forma após a geração das visualizações pode ser necessário alterar a forma de distribuição, de apre-

sentação dos nós, parâmetros de visualização ou filtros, para que desta forma a visualização possa ser útil.

A Figura 4.6 apresenta uma visualização gerada para a rede que relaciona colaboradores através do relacionamento por contribuição. Nesta visualização, apenas 24% dos nós estão visíveis, pois um filtro foi aplicado para que colaboradores que contribuíram em apenas um trimestre fossem ocultados. A estratégia de distribuição Force Atlas 2 (JACOMY *et al.*, 2014) foi utilizada nesta visualização e a partir da sua aplicação é possível distinguir 2 conjuntos de colaboradores de maneira mais clara do restante da rede, sendo localizados nos cantos superiores da Figura 4.6 e representados com as cores laranja e preto. Estes dois grupos representam usuários que trabalharam em conjuntos de projetos similares nos mesmos intervalos de tempo.

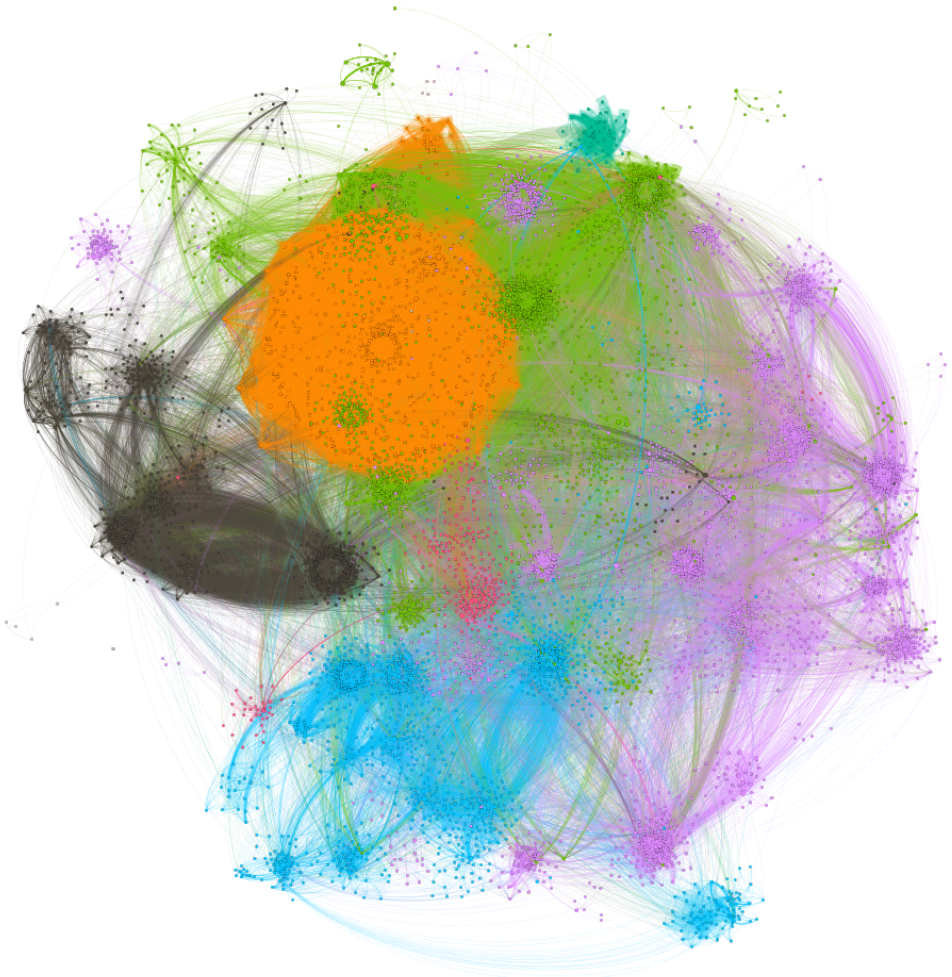


Figura 4.7: OpenOrd todos os trimestres.

Outra estratégia de distribuição foi aplicada sob a mesma rede complexa, desta vez a OpenOrd(MARTIN *et al.*, 2011). O filtro de apenas colaboradores que apareceram em dois ou mais trimestres foi mantido e o período temporal também é o mesmo. A visualização é apresentada na Figura 4.7. Através da distribuição aplicando o algoritmo OpenOrd, é possível identificar comunidades dentro de conjuntos de nós com a mesma classe de modularidade, apresentadas na Tabela 4.3.

A partir da comparação entre as duas visualizações, é possível notar como os parâmetros e algoritmos de distribuição influenciam no resultado final da visualização e, por este motivo, é importante que o cientista avalie o impacto das suas escolhas e faça os ajustes necessários para que a visualização da rede complexa possa auxiliá-lo. O subprocesso apresentado na Figura 3.6 contém os principais pontos para ajuste da visualização.

Reavaliação da análise de redes

Durante o subprocesso de análise de redes complexas, após a geração das visualizações e seleção das métricas, são realizadas decisões para avaliar as escolhas que guiaram a análise de redes complexas. Nesta etapa, é importante que o cientista avalie a maior quantidade de variáveis possíveis com o intuito de utilizar as medidas mais aderentes ao objetivo.

Exemplificando o ajuste, a partir das visualizações apresentadas na Figura 4.6 e Figura 4.7, a identificação dos colaboradores que possuem mais influência a nível global não é facilitada. As duas visualizações apresentam a rede social contida no ecossistema avaliado, mas não existe nenhum auxílio na identificação deste tipo de colaborador. A partir da avaliação das métricas calculadas previamente, foi selecionada a centralidade de *betweenness* para auxiliar na identificação dos colaboradores com maior impacto global, visto que esta medida considera a quantidade de caminhos mínimos entre os nós da rede complexa. Esta métrica pode atuar como um indicador dos colaboradores que são ativos em múltiplos projetos pois estes são os colaboradores que criam os laços entre diferentes comunidades.

A centralidade de *betweenness* foi escolhida por estar alinhada com o objetivo de identificar os principais colaboradores dos projetos que compõe o Ecossistema. Desta

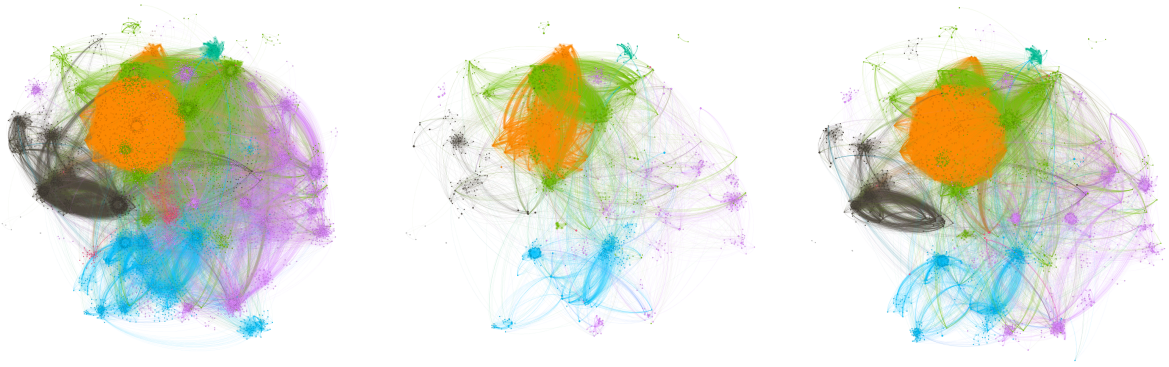
forma, a rede teve o tamanho dos nós ajustados de acordo com a centralidade de *betweenness*, e esta visualização é apresentada na Figura 4.8. Com o ajuste no tamanho dos nós, é possível identificar de maneira facilitada os colaboradores que possuem maior impacto na rede, pois os nós com esta centralidade alta tem tendências de possuir influência maior na rede visto que a informação passa por eles mais que por outros nós visto que eles são os responsáveis por facilitar a comunicação entre grupos distintos. Além disso, caso estes nós sejam removidos, a chance que o fluxo de comunicação da rede seja prejudicada é alto, como avaliado em GUÉRCIO *et al.* (2017), que remove colaboradores selecionados a partir de medidas de centralidade com o objetivo de identificar rupturas na rede complexa.

Figura 4.8: Distribuição OpenOrd, ajustando o tamanho dos nós de acordo com a centralidade de betweenness.

As visualizações também podem apoiar no reconhecimento da evolução de projetos, usuários e comunidades. A partir da observação dos diferentes períodos de tempo é

possível capturar aspectos evolutivos do ecossistema de maneira facilitada e ágil.

Na Figura 4.9 é possível notar como a representação dos dados considerando o período afeta na forma como é possível reconhecer o SECO e as comunidades que o compõe. Ao observar o ecossistema fatiado em cada um dos anos, é possível perceber como grupos de colaboradores entram e saem do ambiente compartilhado.



(a) Visualização acumulada do ecossistema, com distribuição OpenOrd

(b) Ano 1

(c) Ano 2

(d) Ano 3

(e) Ano 4

(f) Ano 5

Figura 4.9: Conjunto de figuras contendo usuários que colaboraram durante o período de observação, sendo apresentados entre os anos de 2012 e 2017. As visualizações utilizam a estratégia de distribuição OpenOrd.

Considerando a Figura 4.9(a), que contém os dados de todo o período, e a partir dela é possível identificar grupos inseridos nas classes de modularidade, que representam conjuntos de nós densamente conectados, apresentados na Tabela 4.3. Tomando como exemplo a classe de modularidade 2, representada pela cor mais escura, é possível perceber a existência de diferentes grupos. A Figura 4.9(b) tem um grupo de colaboradores da classe 2 que deu início no desenvolvimento, e na Figura 4.9(c), que representa o segundo ano, é possível perceber que dois grandes grupos de usuários começaram e mantiveram

uma forte interação neste ano, e no próximo, representado na Figura 4.9(d). No quarto ano da observação, representado na Figura 4.9(e) o grupo que iniciou o desenvolvimento não possui representatividade devido à falta de atividade no ano. No último período de observação, Figura 4.9(f), é possível notar que um dos grupos envolvidos já não possui atividade.

Esta observação pode ser realizada para cada uma das classes de modularidade, onde a classe 1, em laranja, teve uma grande atividade no primeiro ano, que seguiu uma crescente no segundo e terceiro ano, mas que no quarto ano teve uma diminuição significativa, e no ano 5 já não é possível encontrar grandes grupos de colaboradores.

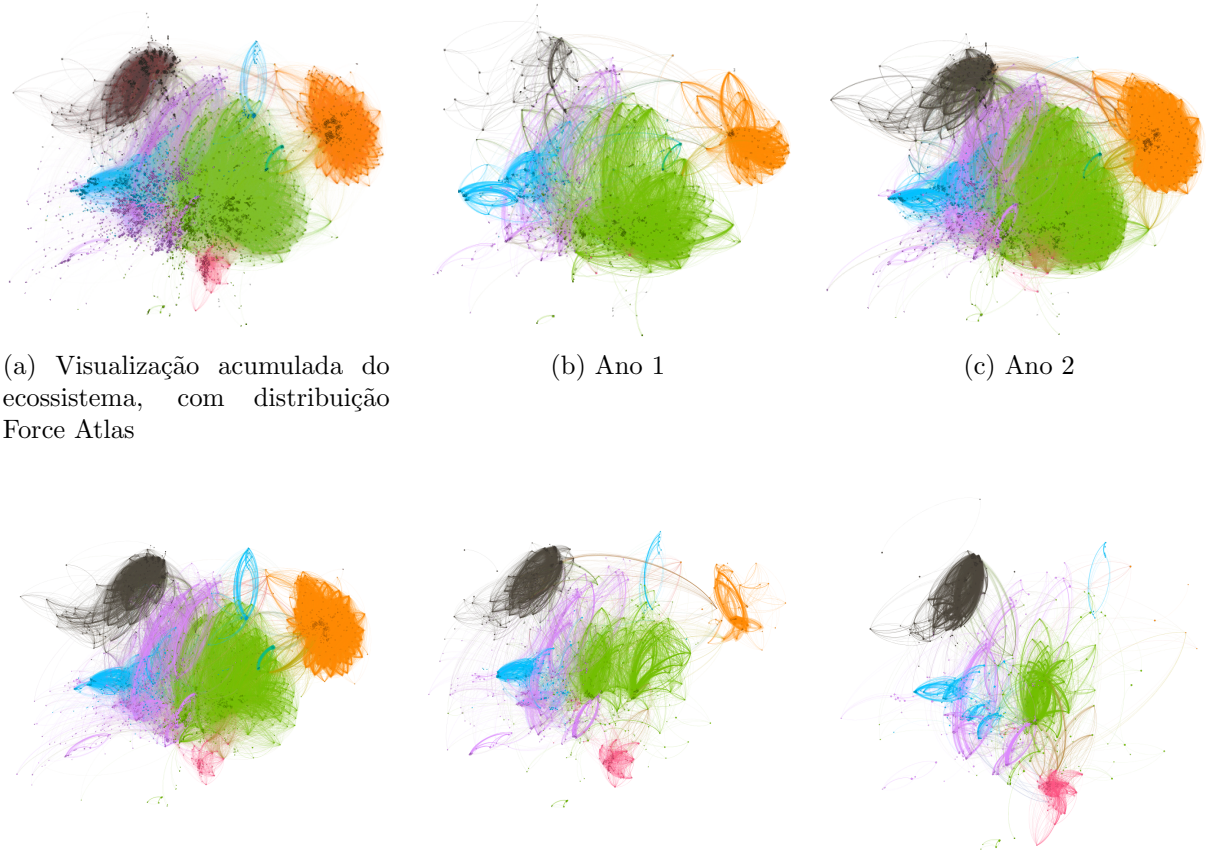
A Figura 4.10 é análoga a figura anterior, que dividiu 5 anos de atividades e tem o período completo de observação na primeira figura. A diferença entre os dois conjuntos de figuras é a estratégia de distribuição dos nós. A estratégia de distribuição tem grande impacto na análise, e isso pode ser visto ao se comparar as duas figuras.

A partir da análise pela distribuição OpenOrd, é mais fácil visualizar as comunidades existentes dentro de um conjunto de colaboradores pertencentes a uma mesma classe de modularidade. Nas Figura 4.10(e) e (f) que utiliza a estratégia de distribuição Force Atlas, não é possível identificar de maneira clara que existem dois subgrupos, e é ainda mais difícil perceber que entre estes anos um subgrupo praticamente cessou suas atividades. Ao observar as Figura 4.10 (e) e (f) que utiliza a estratégia de distribuição OpenOrd esta avaliação é muito facilitada.

Em contrapartida, a estratégia de distribuição Force Atlas dá um maior destaque às classes de modularidade, e nesta estratégia de distribuição é possível ver de maneira facilitada os colaboradores que fazem parte da classe de modularidade 6, apresentada na cor rosa, que teve o início das suas atividades no ano 3.

Cada uma das estratégias se aplica melhor a um determinado cenário e objetivo, e, por isso, o cientista deve realizar diferentes configurações que impactem nas visualizações para que elas possam ser úteis. No caso das estratégias selecionadas, o algoritmo OpenOrd facilita a identificação dos grupos de colaboradores pelo cientistas. De posse disso, é possível avaliar os grupos de usuários com uma granularidade maior visto que esta estratégia dá destaque aos agrupamentos. A estratégia Force Atlas, que foi a utilizada

para gerar o outro grupo de visualizações também apresenta uma distribuição coerente com as classes de modularidade, mesmo este atributo não sendo considerado durante a visualização. A vantagem da estratégia Force Atlas está em mostrar os *authorities*, que são elementos referenciados por diferentes nós da rede, dando destaque aos colaboradores que são referência dentro do ecossistema.



(a) Visualização acumulada do ecossistema, com distribuição Force Atlas

(b) Ano 1

(c) Ano 2

(d) Ano 3

(e) Ano 4

(f) Ano 5

Figura 4.10: Conjunto de figuras contendo usuários que colaboraram durante o período de observação, sendo apresentados entre os anos de 2012 e 2017. As visualizações utilizam a estratégia de distribuição Force Atlas.

Além das visualizações da rede que relaciona os usuários através da medida de relacionamento por contribuição(Equação A.3), também foram geradas visualizações para a outra rede social extraída a partir do *dataset* avaliado. A segunda rede social apresentada foi construída a partir das relacionamento entre os participantes através da atividade de seguir um participante, que é realizada com o objetivo de receber atualizações sobre as atualizações das pessoas que o usuário decidiu por seguir.

A Figura 4.11 apresenta a visualização gerada a partir da rede de *followers*. A rede construída não possui peso nas arestas e as arestas são direcionadas, onde a origem da

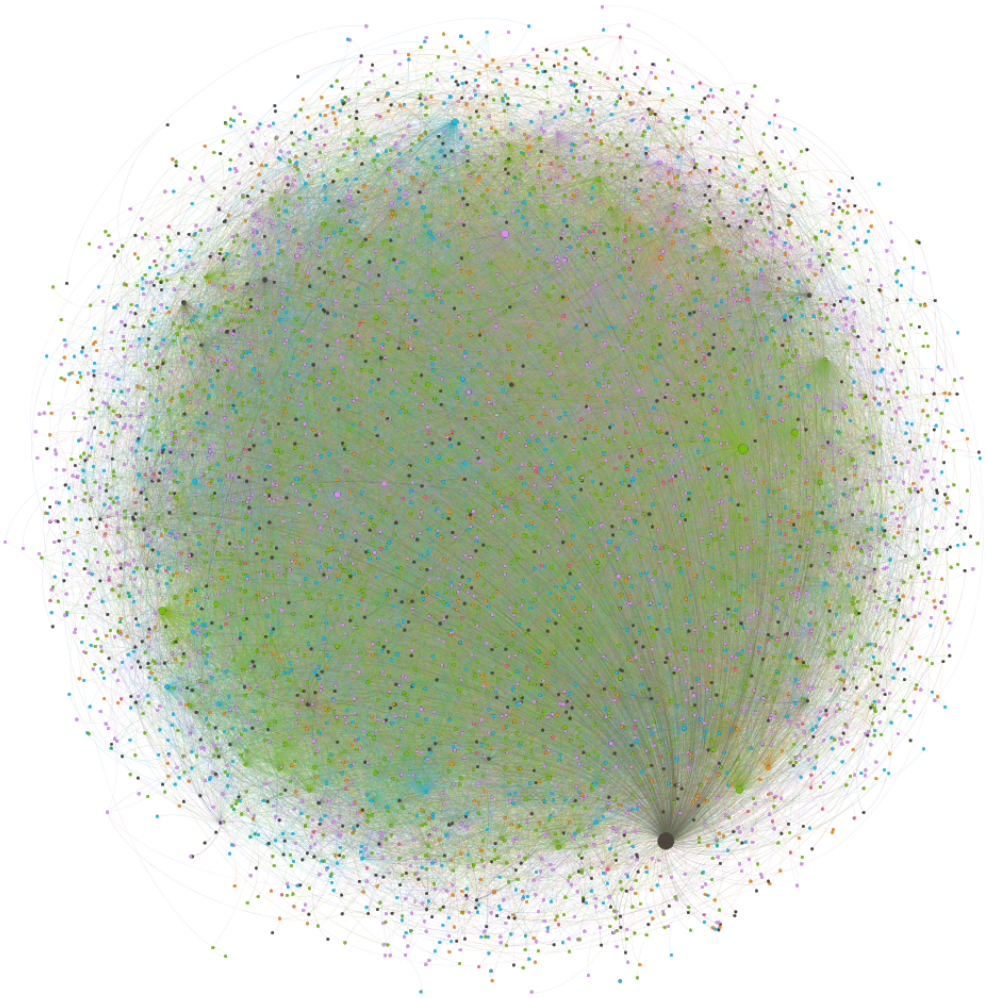


Figura 4.11: Rede de colaboradores se relacionando através de follows.

aresta é o usuário que decidiu por seguir outro colaborador. Os nós foram dimensionados de acordo com seu grau de entrada, para dar destaque aos colaboradores que possuem mais seguidores.

A classe de modularidade foi calculada para a rede de *followers* mas as classes de modularidade não foram representativas, gerando poucas classes com quantidades significativas de usuários e agrupando a grande maioria em uma classe de modularidade que compreendia mais de 90% dos nós. Desta forma, os usuários foram representados com as classes de modularidade geradas a partir da rede de relacionamento por contribuição.

Avaliação das métricas

Durante a atividade de análise, pode ser possível utilizar as redes complexas para gerar conhecimento mas não necessariamente este é um passo obrigatório. Após a definição dos objetivos e da decisão pela utilização das redes complexas, o cientista deve dar início nas atividades que utilizam as métricas disponíveis no ambiente de análise.

O cientista deve selecionar as métricas disponíveis no ambiente de análise que sejam aderentes aos seus objetivos, e após a seleção ele também deve selecionar qual a abordagem de medição mais apropriada. Uma mesma medida pode ter diferentes formas de ser medida, como por exemplo a medida de popularidade de um projeto, que pode considerar diferentes características, conforme descrito anteriormente na Seção 4.2.2

Dentre as métricas disponíveis, foram selecionadas as métricas de usuários para análise. Para avaliação da medida de popularidade por projeto, proposta neste trabalho e apresentada na Equação 4.6, foi necessária a instanciação de uma nova rede. Isto se dá ao fato da medida observar a popularidade do usuário em um determinado projeto, e as redes instanciadas consideram múltiplos projetos.

Desta forma, foram instanciadas 3 redes, referentes aos projetos node[19], marionette[20] e jekyll[21]. Esses projetos foram escolhidos pela quantidade de colaboradores que tiveram valores de popularidade. A Tabela 4.4 apresenta de maneira sintetizada os valores

Tabela 4.4: Avaliação da correlação de Pearson entre medidas extraídas das redes complexas e a popularidade de usuário por projeto.

Projeto / Medida	**Pagerank**	**Closeness**	**Betweenness**	**Degree**	**Weighted Degree**
Jekyll	0,900	0,457	0,748	0,500	0,902
Marionette	0,564	0,300	0,389	0,566	0,711
Node	0,506	0,317	0,467	0,395	0,507

encontrados. É possível perceber que para o projeto Jekyll, a medida de popularidade apresentou forte relação com as medidas de pagerank e de grau ponderado. O cálculo da medida de popularidade aqui avaliada depende do registro entre a interação entre os usuários, entretanto esta medida depende de registros de mensagens, que no *dataset* utilizado estavam incompletos por existir uma limitação de caracteres nas mensagens. Os

[19]https://github.com/nodejs/node
[20]https://github.com/marionettejs/backbone.marionette
[21]https://github.com/jekyll/jekyll

projetos são apresentados na tabela pela ordem de quantidade de usuários com a medida de popularidade maior que um, entretanto em nenhum dos projetos existiam mais de 10% de usuários com a popularidade maior que zero, indicando a necessidade de novos testes em conjuntos de dados mais completos.

Durante a avaliação, foi observado a categoria dos colaboradores, classificados entre mutantes, candidatos, externos e core. A descrição de cada um das categorias é apresentada na Seção 4.2.2. Seguindo a divisão temporal estabelecida durante a avaliação, que dividiu o período em trimestres, foram avaliadas as transições entre as categorias que os desenvolvedores foram classificados. No *dataset* avaliado, não ocorreu nenhuma transição de desenvolvedores externos para desenvolvedores core. Era esperado um valor baixo, visto que os colaboradores externos conseguem incorporar seu código no projeto através das solicitações de *pull requests*, entretanto nenhuma transição foi registrada.

Ao avaliar as transições entre as categorias de mutantes, candidatos e externos, notou-se que os candidatos que realizam a transição de mutante a externo realizam essa transição de forma rápida. Também se observou que a maioria dos desenvolvedores realiza a transição apenas entre uma categoria. A moda da quantidade de trimestres da transição entre as categorias de mutante até candidato e de candidato até externo foi de 1 trimestre. De forma específica, a transição de mutante a candidato teve 39% e de candidato até externo foi de 48%. Este valor pode ser um indicativo que para observar a mudança entre as categorias de desenvolvedores, deve-se aumentar a granularidade do período para que seja possível capturar de maneira mais precisa, principalmente a mudança de categoria entre mutante e candidato.

Ao diminuir a quantidade de dias de cada período observado, espera-se ser possível identificar, de maneira mais clara, a transição entre as categorias de mutante à candidato, visto que os desenvolvedores que realizaram esta transição, em sua maioria, não precisaram de mais de 2 trimestres. Também foram observados casos onde após longos períodos sem contribuições nos *forks*, alguns desenvolvedores retomaram o interesse nos projetos e realizaram novas colaborações. Isso pode ser observado nos casos onde existiu um hiato de participação do desenvolvedor. Em um dos casos, um desenvolvedor realizou *commits* em um fork, e após 4 anos foi realizadas novas modificações no código, que foram aceitas.

Tabela 4.5: Desenvolvedores que realizaram transições entre as categorias de mutante, candidate e externo no projeto octopress. Os valores das colunas mutante, candidato e externo representam o trimestre em que foi registrado o primeiro registro do desenvolvedor na devida categoria.

Desenvolvedor	**Mutante**	**Candidato**	**Externo**
A	6	7	8
B	7	9	10
C	5	6	
D	7	9	
E	1	2	
F	1	12	

A tabela Tabela 4.5 mostra a transição de alguns desenvolvedores no projeto octopress. Os desenvolvedores A e B, levaram, respectivamente, 3 e 4 trimestres, para concluir a transição entre as categorias de mutante e externos. Também é possível perceber que a transição de candidato até externo foi realizada pelos desenvolvedores A,B,C e E em um trimestre, indicando que após a solicitação do *pull request* não se levou muito tempo até a sua aceitação. Os desenvolvedores A e B tiveram seus *pull request*s avaliados de maneira específica. O desenvolvedor A, realizou 3 *pull request*s em 19 dias, todos os três resolvendo pequenos bugs relacionados a troca de nomenclatura

O desenvolvedor F também foi observado de maneira detalhada, visto que entre sua primeira tentativa de colaborar com o código e a aceitação na comunidade existiu um período de 3 anos. Seu primeiro *pull request* , realizado em 05/12/2011 foi fechado pelo próprio autor, por identificar erros em suas alterações. Em sua segunda tentativa de colaborar, em 22/05/2012 seu *pull request* teve uma sequência de 31 comentários com 9 participantes distintos. Inicialmente, colaboradores ativos do projeto não conseguiram reproduzir o erro, que em seguida teve report de outros membros da comunidade com o mesmo problema. Em seguida, o problema foi resolvido pelo responsável pelo repositório, fazendo com que seu *pull request* fosse rejeitado. Seu terceiro *pull request* foi realizado em 17/11/2014, que resolvia um bug reportado e foi aceito rapidamente.

A partir disso, é possível perceber que mesmo que as tentativas de contribuir com o projeto sejam rejeitadas, os desenvolvedores continuam acompanhando o projeto, indicando um alinhamento com seus objetivos. Caso uma nova oportunidade para contribuir com o projeto apareça, os desenvolvedores realizam novos *pull request*s.

4.2.4 Avaliação dos Resultados

De acordo com o planejamento da avaliação, os resultados serão analisados a partir da abordagem GQM, planejada no início deste capítulo. A primeira questão a ser respondida tem como objetivo verificar se o processo possui todas as atividades necessárias para extração, e foi avaliada pelas métricas **M1** e **M2**, que mensuram, respectivamente, a quantidade de atividades desenvolvidas durante o processo e a quantidade de atividades identificadas na revisão do processo por especialistas que não estavam contempladas no processo.

Após a revisão dos especialistas no sSECO-Process não foram identificados pontos de incerteza que pudessem motivar novas atividades não contempladas no processo. Vale-se destacar que o mesmo conjunto de especialistas avaliou as duas propostas apresentadas neste trabalho. Desta forma é provável que existam pontos de melhoria no processo, mas que não ficaram explícitos aos especialistas pela familiaridade com o processo a partir das reuniões periódicas. Sendo assim, a avaliação da **Q1** foi positiva visto que o processo auxiliou na identificação do conhecimento sem a necessidade de ajustes em tempo de execução.

Com o objetivo de responder a **Q2**, que verifica se os *stakeholders* podem ser auxiliados pelo uso de redes complexas na identificação dos desenvolvedores que mais contribuíram foram utilizadas duas métricas, onde a primeira (**M3** mede a quantidade de redes complexas modeladas e a segunda (**M4**) mede a quantidade de desenvolvedores que mais contribuem. Durante a execução da avaliação do sSECO-Process foram identificadas 9 redes complexas a partir dos dados, que são apresentadas na Tabela 4.2, onde duas dessas redes são redes sociais que relacionam usuários através de diferentes estratégias.

A **M4** foi medida da mesma forma do estudo preliminar, através da centralidade de *betweenness*. O cálculo das centralidades para responder a **Q2** foi realizado na rede de relacionamento por contribuição, considerando todo o período de observação. Dentre os 7771 colaboradores, 5835 possuem esta centralidade igual a zero, o que significa que eles não participam de nenhum caminho mínimo da rede construída e por este motivo foram avaliados os 1936 que fazem parte de algum caminho mínimo. Os valores de centralidade foram normalizados e apresentados na Figura 4.12.

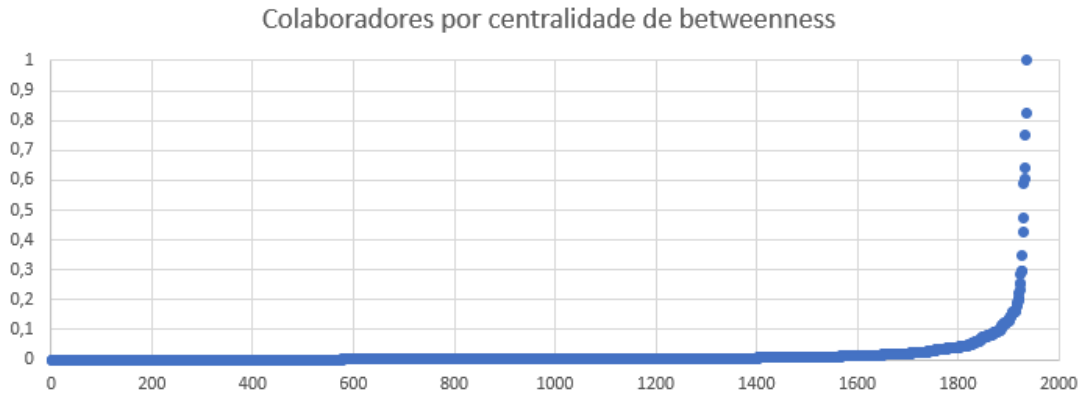


Figura 4.12: Colaboradores de acordo com a centralidade de betweenness normalizada.

Assim como no estudo preliminar, existe uma grande quantidade de colaboradores com centralidade baixa. A partir das centralidades normalizadas foi calculada a média, que foi de 0.0119 e o desvio padrão de 0.05247. No total, 92 colaboradores estão 1 desvio padrão acima da média e 45 estão 2 desvios padrões acima da média. Desta forma, como resultado da **M4**, temos 92 colaboradores que se destacaram em relação aos 7771 presentes na rede social construída a partir do *dataset* construído, que representa o SECO identificado em (BLINCOE *et al.*, 2015). Com os valores das métricas **M3** e **M4** disponíveis, foi possível verificar que a **Q2** foi atendida por auxiliar na identificação dos desenvolvedores que mais contribuíram com o SECO.

Com o objetivo de responder a **Q3**, que avalia a colaboração no ambiente compartilhado sob o aspecto evolutivo, foram estabelecidas três métricas. As métricas tem como objetivo avaliar cada um dos pilares da colaboração, que são a cooperação, coordenação e comunicação. Mais detalhes sobre o modelo de colaboração podem ser encontrados em (FUKS *et al.*, 2003), que apresenta o Modelo 3C. Em síntese, os autores afirmam que para colaborar, os indivíduos têm que trocar informações (se comunicar), organizar-se (se coordenar) e operar em conjunto num espaço compartilhado (cooperar). A partir da troca de informação gerada pela comunicação, são firmados compromissos que são gerenciados pela coordenação, que por sua vez organiza e dispõe as tarefas que são executadas na cooperação. Desta forma, a **M5** mede a quantidade de contribuições no ecossistema no tempo. Para medir esta métrica, foram considerados todos os registros de *pull requests* que foram abertos no período. Estes registros são um indicador da cooperação no espaço compartilhado visto que a contribuição é reflexo do esforço de um colaborador que pro-

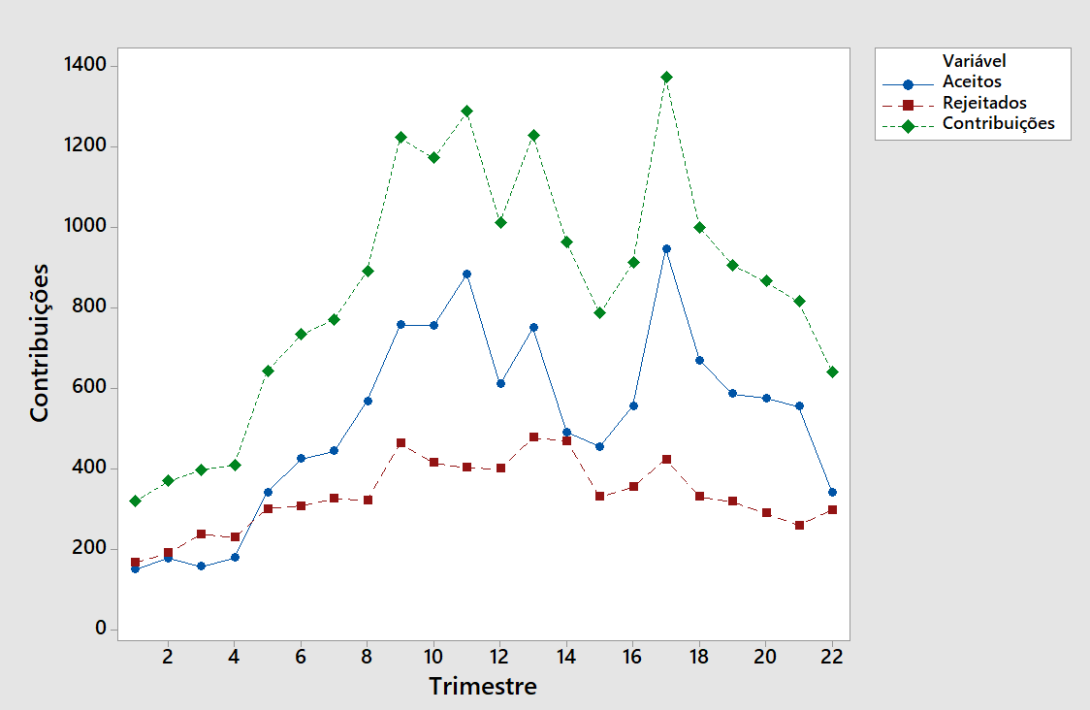


Figura 4.13: Contribuições realizadas no ecossistema através do tempo.

duz código e de outro colaborador que deve revisar e aceitar a solicitação de incorporação de código. A aceitação do *pull request* não é obrigatória para a **M5**, pois mesmo que o código não tenha sido incorporado o solicitante realizou uma tentativa de colaborar com o SECO.

A métrica que avalia a coordenação **M6** considera os *pull requests* rejeitados no tempo. A rejeição de um *pull request* significa que participantes da comunidade avaliaram a solicitação de colaboração e rejeitaram esta tentativa, seja por inadequação técnica, desalinhamento com os objetivos do grupo ou outros motivos. A Figura 4.13 apresenta as contribuições através do tempo no SECO analisado.

A última métrica a ser avaliada para responder a **Q3** é referente à comunicação dentro do SECO. Para isso foram identificados os comentários no ecossistema no tempo. Existem diferentes locais onde o colaborador pode comentar e se comunicar com o restante dos membros da comunidade. Os comentários em *issues* e nos *pull requests* foram utilizados para coleta dos dados da **M6**. A quantidade dos *commits* é apresentada na Figura 4.14, que utiliza escala logarítmica de base 10 para relacionar as duas quantidades de comentários, pois os comentários em *pull requests* são menos frequentes, onde a média de comentários em *issues* é de 23.126 e a média de comentários em *pull requests* é de

Tabela 4.6: Matriz dos valores de correlação de Pearson.

Correlação Pearson	Pagerank	Closeness	Betweenness	Degree	Weighted Degree
Dev	0,564	0,181	0,512	0,397	0,388
Pop	0,183	0,069	0,18	0,126	0,086

1.395. De acordo com as métricas referentes a **Q3** é possível observar que o SECO possui maior atividade entre os trimestres 8 e 18, que pode ser visualizado através da Figura 4.9, que apresenta os colaboradores ativos por ano.

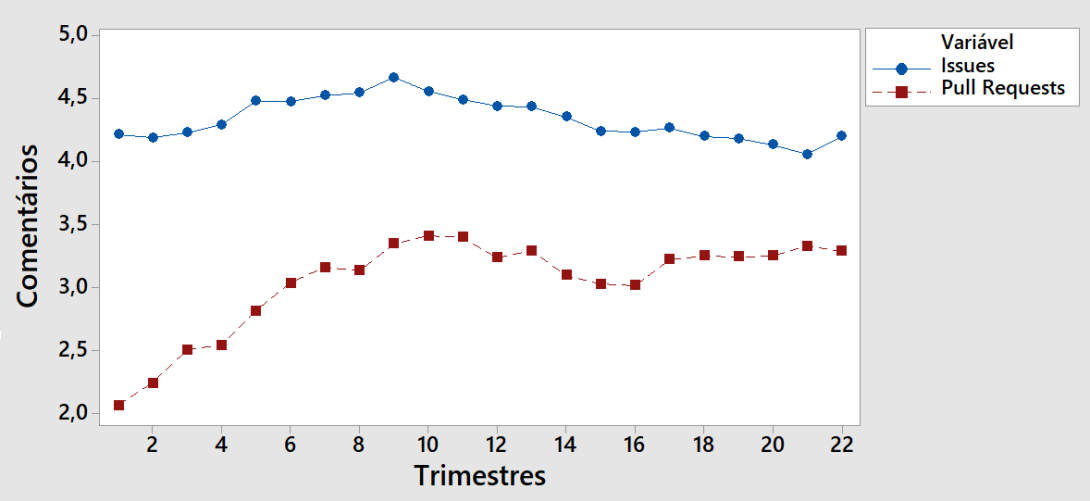


Figura 4.14: Comentários realizados no ecossistema através do tempo.

Para responder a **Q4**, que deseja verificar se as métricas disponíveis na literatura possuem correlação com as medidas obtidas através das redes complexas foram utilizadas duas métricas, **M8** e **M9**, sendo, respectivamente, as correlações de Pearson e Spearman. O coeficiente de correção mede o grau pelo qual duas variáveis tendem a mudar juntos. Além de descrever a força, o coeficiente também descreve a direção dessa relação, podendo variar de -1 até 1, onde -1 significa uma alta correlação decrescente e 1 significa uma alta correlação crescente, isto é, se uma variável cresce a outra cresce na mesma proporção.

A diferença entre as métricas **M8** e **M9** é que a correlação de pearson avalia uma relação linear entre duas variáveis contínuas e a correlação de Spearman avalia a relação monotônica, onde as duas variáveis tendem a mudar juntas, mas não necessariamente seguindo uma taxa constante.

A partir da avaliação das métricas referentes a **Q4**, notou-se que não existe forte correlação entre as métricas identificadas na literatura e as métricas extraídas da rede

Tabela 4.7: Matriz dos valores de correlação de Spearman.

Correlação Spearman	Pagerank	Closeness	Betweenness	Degree	Weighted Degree
Dev	0,378	0,220	0,541	0,289	0,304
Pop	0,232	0,198	0,338	0,181	0,139

complexa. A maior correlação encontrada foi entre as métricas DEV, que mensura o grau de comprometimento dos colaboradores em atividades de desenvolvimento, com a centralidade de *betweenness*, que avalia a quantidade de caminhos mínimos que passam por um determinado nó.

A métrica **M8** foi calculada, considerando a medida de popularidade de usuários por projeto, proposta neste trabalho. O resultado é apresentado na Tabela 4.4, que mostra que em determinados projetos existem relações fortes entre as centralidade de grau e pagerank à nova medida proposta. Este pode ser um indicativo da assertividade da medida em identificar colaboradores populares.

4.3 Limitações e ameaças à validade

Algumas limitações e ameaças à validade foram identificadas. A primeira ameaça é relacionada ao *dataset* utilizado durante a avaliação realizada no Capítulo 4, que considera que a técnica proposta por BLINCOE *et al.* (2015) é capaz de identificar SECOs de maneira consistente. Outra ameaça é com relação à quantidade de SECOs avaliados, totalizando dois, um tratado na avaliação preliminar e o outro na avaliação do sSECO-Process. Desta forma o estudo não pode ser generalizado, entretanto os dados estão disponíveis para serem continuados por outros projetos.

Com relação às dificuldades enfrentadas, temos a dificuldade em trabalhar com grandes volumes de dados. A utilização do *dataset* fornecido pelo GHTorrent foi vital para desenvolvimento do trabalho, mas o esforço computacional foi grande devido à quantidade de dados disponíveis.

Mesmo com a grande quantidade, outra limitação enfrentada foi a falta dos dados da mensagem, que impossibilitaram uma avaliação mais profunda das trocas de mensagem entre os usuários, como realizado em um estudo preliminar que capturava a semântica das

mensagens (GUÉRCIO *et al.*, 2016). Esta limitação acarretou outra ameaça à validade, visto que o método para identificação de ecossistemas proposto por BLINCOE *et al.* (2015) não pôde ser reproduzido por não conter as mensagens completas, e por isso não foi possível identificar novos projetos que entraram no ecossistema ou a relação entre os projetos do SECO.

5 Trabalhos Relacionados

Neste capítulo, são apresentados os trabalhos relacionados, organizados em dois grupos principais. Primeiro, são apresentados os trabalhos referentes aos Ecossistemas de Software e depois os trabalhos que utilizam Redes Complexas, entretanto alguns trabalhos podem ser classificados nos dois grupos. No fim deste capítulo, são definidas as características comuns para os grupos e é realizada uma comparação entre os trabalhos apresentados. Os trabalhos foram identificados após uma análise sobre os resultados da revisão realizada em (FRANCO-BEDOYA *et al.*, 2017), que disponibilizou os arquivos base. Além de auxiliar na identificação de trabalhos aderentes a esta pesquisa, a documentação possibilitou que os trabalhos selecionados indicassem novos trabalhos através de suas citações.

SANTOS & WERNER (2012) apresentam um *framework*, ReuseECOS '3+1', para auxiliar na gestão e desenvolvimento de Ecossistemas de Software, que utiliza a visão tridimensional apresentada por CAMPBELL & AHMED (2010), que divide os aspectos de negócio, sociais e sócio-técnicos presentes nos SECOs. A dimensão técnica explora decisões arquiteturais como escolha de padrões de desenvolvimento e visualização de código. A dimensão de negócios explora analogias com ecossistemas em outros domínios, como apresentado em (MENS & GROSJEAN, 2015) que relaciona ecossistemas ecológicos aos SECO, e, a partir do relacionamento entre diferentes ecossistemas, é possível obter modelos para classificar e avaliar SECOs. A dimensão social explora as redes complexas existentes no SECOs, sejam elas redes sociais, técnicas ou sócio-técnicas.

Além das três dimensões, o framework apresenta a visão de gerenciamento e construção, que explora os relacionamentos entre as dimensões anteriores para obter conhecimento sobre o SECO e seus impactos na engenharia de software. Em (DOS SANTOS & WERNER, 2012) o foco dos autores é a dimensão social, que tem como objetivo entender como as relações sociais se estabelecem, organizam e mantém a comunidade que compõe o SECO, considerando a aquisição, transmissão e troca de conhecimento entre as redes complexas.

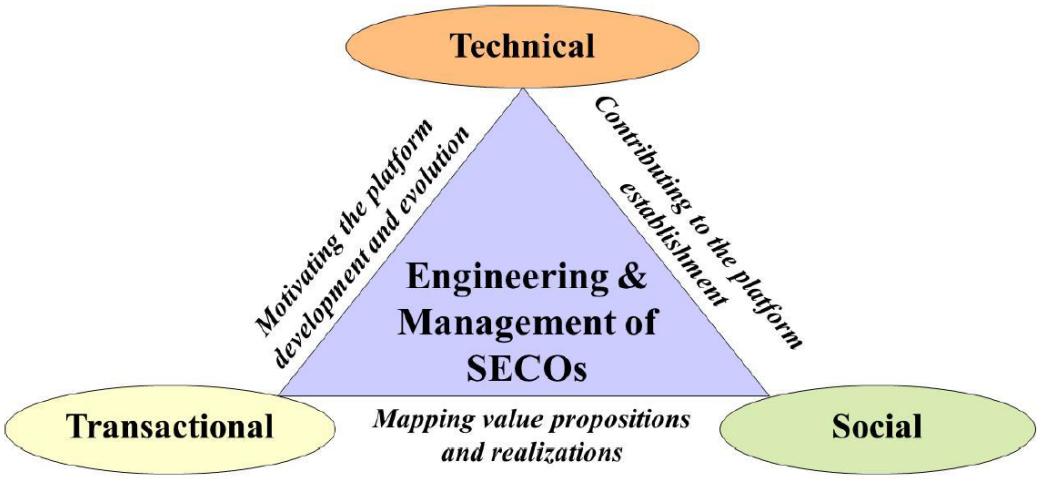


Figura 5.1: Visão geral do ReuseECOS '3+1' Framework (DOS SANTOS, 2016).

O trabalho de MENS & GOEMINNE (2011) também observa os aspectos sociais em SECO, mas neste estudo o escopo foi delimitado em OSSECO, e a comunidade do ecossistema GNOME foi analisada, com o objetivo de entender a comunidade de desenvolvedores, identificando as formas de trabalho dos desenvolvedores, como eles cooperam no ambiente comum além das formas de comunicação e disseminação do conhecimento. Os autores destacam que a comunicação é fator essencial em projetos de software, mas no cenário (*open source*) as barreiras de novos entrantes devem ser menores, para diminuir o esforço de envolvimento de novos colaboradores com o time de desenvolvimento, implicando em estruturas flexíveis que possam lidar com a chegada de novos entrantes assim como com a perda de colaboradores, visto que a rotatividade dos colaboradores é muito alta nas comunidades open source. MENS & GOEMINNE (2011) identificaram

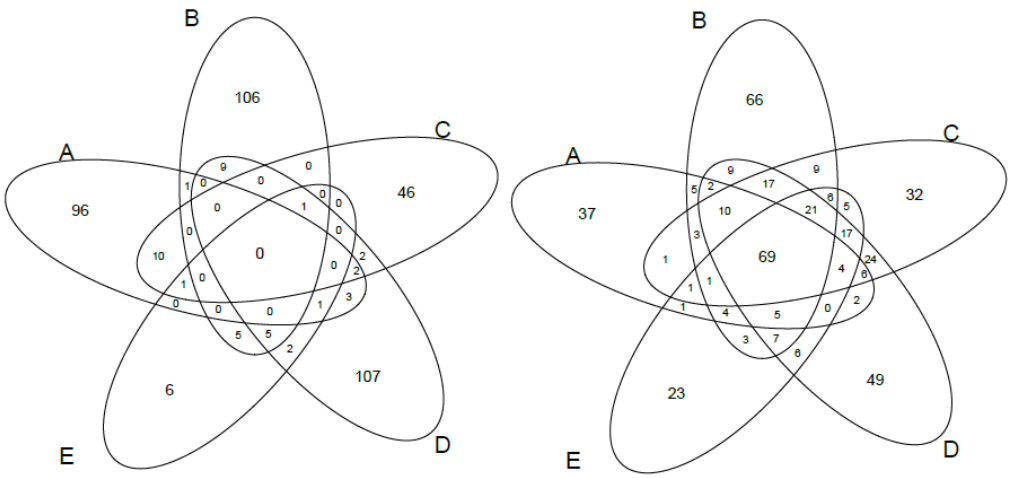
Tabela 5.1: Tipos de atividade e os tipos de arquivo correspondente. Apenas as atividades mais frequentes entre os projetos foram listadas MENS & GOEMINNE (2011).

Tipo de atividade	**Tipos de arquivo**
Desenvolvimento	*.c, *.h, *.cc, *.pl, *.java, *.s, *.ada, *.cpp, *.chh, *.py
Documentação de código	readme*, *changelog*, todo*, hacking*
Documentação	*.html, *.txt, *.ps, *.tex, *.sgml, *.pdf
Tradução	*.po, *.pot, *.mo, *.charset

tipos de atividades de desenvolvimento, associados aos tipos de arquivos que cada um dos colaboradores alterou. Essas atividades são apresentadas na Tabela 5.1, que relaciona os tipos de atividades com tipos de arquivo.

Os autores também avaliam as interseções entre as diferentes atividades que um

mesmo colaborador desenvolveu. Também foi observado que colaboradores que desenvolvem as atividades de desenvolvimento normalmente estão associados a apenas um projeto do ecossistema, enquanto colaboradores que atuam em atividades de tradução estão envolvidos em múltiplos projetos, conforme apresentado na Figura 5.2. Os autores concluíram que a atividade realizada pelo colaborador tem impacto na quantidade de projetos que ele está envolvido e acreditam que isto pode estar relacionado às características intrínsecas de cada atividade ou pela presença de ferramentas e/ou mecanismos que auxiliem a comunidade a distribuir e compartilhar seu trabalho, visto que no caso do GNOME existe um projeto responsável por realizar traduções entre os outros projetos que compõem o ecossistema.



(a) Intersection of coders (b) Intersection of translators

Figura 5.2: Interseção entre os autores do git que contribuíram em 5 projetos selecionados do GNOME (representados pelos conjuntos [A,E], onde (a) representa a interseção entre os projetos que colaboradores de desenvolvimento atuaram e (b) representa a interseção dos projetos onde tradutores atuaram MENS & GOEMINNE (2011).

Em (CONSTANTINOU & MENS, 2017), o ecossistema Ruby, com seu código hospedado no GitHub é avaliado a partir dos seus 9 anos de dados para capturar aspectos da sua evolução. Um ecossistema evolui a partir das suas alterações sociais e técnicas, e este estudo realiza análises sócio-técnicas com o objetivo de identificar mudanças no ecossistema, mas o foco dos autores são nas mudanças no ecossistema que o afetam de maneira permanente. A análise é realizada tanto através do código fonte, quanto das características dos colaboradores, avaliando a evolução sob a perspectiva dos projetos

base, dos *forks*(cópia) e do ecossistema de maneira completa, apresentando os resultados a nível de projeto, considerando os projetos base e *forks*, assim como indicado por KALLIAMVAKOU *et al.* (2016), e o outro resultado a nível de ecossistema.

Os dados do ecossistema Ruby, entre 2007 e 2016 foram avaliados, e o tempo foi dividido em trimestres. Para cada trimestre de atividade, os projetos, colaboradores e arquivos tiveram seus status definidos, como obsoletos, novos, ativos, renovados ou abandonados. Antes da análise dos dados, os autores aplicaram estratégias de filtragem, assim como indicado por KALLIAMVAKOU *et al.* (2016). As questões de pesquisa são referentes ao crescimento do ecossistema, evolução dos artefatos e evolução da comunidade. A partir da resposta das questões, os autores indicam que a observação dos resultados podem auxiliar a identificar riscos de sustentabilidade do SECO. Além disso, CONSTANTINOU & MENS (2017) afirmam que as observações sobre as mudanças que ocorrem com o tempo, combinadas com observações externas, podem aumentar a confiança na identificação de quedas de sustentabilidade do SECO.

Ainda na seção sobre ecossistemas, temos o trabalho de BLINCOE *et al.* (2015), que apresentam um novo método para identificação de ecossistemas. A identificação dos ecossistemas existentes no ambiente é relevante por auxiliar os colaboradores a compreender seus papéis e como suas atividades estão inseridas em um escopo que extrapola os projetos dos quais ele participa de maneira direta. Utilizando a definição de LUNGU *et al.* (2010), que definem um ecossistema de software como um conjunto de projetos que são desenvolvidos, e evoluem em conjunto em um mesmo ambiente, os autores utilizam tais dependências para relacionar os projetos, identificando ecossistemas através do novo método.

O método para identificar ecossistema é baseado na identificação das dependências técnicas entre projetos. As dependências são identificadas através de referências cruzadas, especificadas pelos usuários em comentários. Durante a identificação de dependências, foram reconhecidos dois tipos principais, sendo eles a dependência técnica direta entre dois projetos, que é o tipo mais comum. Esta dependência existe quando um *issue* em um determinado projeto depende de uma correção ou atualização em outro projeto. O outro tipo de dependência ocorre quando um par de projetos dependem de um terceiro, que não

é referenciado nas mensagens.

Após a utilização do método de Louvain, para detecção de comunidades, os autores identificaram conjuntos de projetos que são densamente conectados através de suas dependências técnicas, e estas comunidades representam os ecossistemas de software. No total foram identificados 43 ecossistemas no *dataset*, e o maior ecossistema foi utilizado neste trabalho, após contato com os autores, para servir como base para a avaliação realizada no Capítulo 4.

O trabalho de DOS SANTOS & WERNER (2012) tem como objetivo combinar a abordagem de redes complexas para auxiliar na compreensão de SECOs e de sua evolução. O trabalho analisa o impacto das redes sociais presentes em SECOs, propondo uma arquitetura sócio-técnica para auxiliar o ciclo de vida dos SECOs, baseados na utilização das redes sociais presentes no SECO em conjunto com ferramentas de mídias sociais, que podem aproximar pessoas com interesses comuns, aumentando o capital social do SECO e aproximando produtos, serviços e stakeholders.

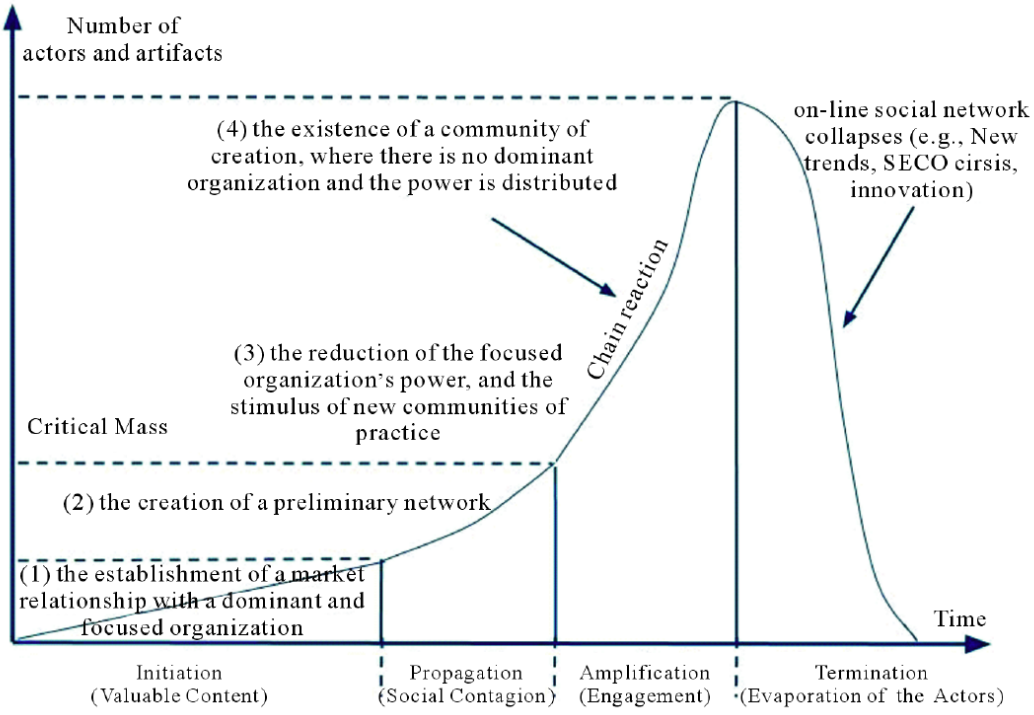


Figura 5.3: Relação entre uso de mídias sociais em SECOs e seu ciclo de vida(DOS SANTOS & WERNER, 2012).

A Figura 5.3 apresenta a proposta do ciclo de vida do uso de mídias sociais em conjunto com SECOs. A primeira etapa é a de iniciação, que representa a criação de

uma página que represente o ecossistema na mídia social. Desta forma, é estabelecida uma relação com os interessados, acrescentado valor ao relacionamento com clientes, distribuídores, vendedores e agentes externos. Em seguida, o estágio de propagação se inicia, com a possibilidade de criar novos laços com novos atores e artefatos. Neste estágio é criada uma rede preliminar, que serve de subsídio para o próximo estágio, que representa a amplificação. Durante a amplificação, a estrutura cresce de maneira rápida, onde deve se estabelecer uma estrutura capaz de organizar-se, visto que o crescimento da rede aumenta a dificuldade na gestão. O último estágio e o de término, normalmente pela saturação da ferramenta de mídia social ou pela substituição por uma nova ferramenta ou pelas novas tendências do mercado.

Em (GUÉRCIO *et al.*, 2017), é realizada uma análise para identificar os participantes que possuem destaque em uma rede social científica. Este trabalho representa um trabalho anterior à esta dissertação que estabeleceu bases para a utilização de redes complexas no domínio de SECO. A rede de social de co-autoria foi analisada com o objetivo de identificar os colaboradores que tem papel central na rede modelada. O relacionamento entre eles considerou o tempo, penalizando contribuições antigas. Durante a análise foram realizadas remoções de colaboradores da comunidade e após cada remoção foi avaliada a distância de colaboração entre participantes distintos. A rede de coautoria se comportou bem quando as remoções foram aleatórias, mas ao se remover os colaboradores de destaque, identificados através de medidas de centralidade, notou-se que a rede perdeu eficiência na transmissão da informação, e foi observado um ponto crítico onde a rede complexa foi dividida em dois componentes que não se comunicavam.

A partir da base estabelecida em GUÉRCIO *et al.* (2017), a estratégia de redes complexas foi aplicada para o domínio de Engenharia de Software, em específico na abordagem de SECO em GUÉRCIO *et al.* (2018). Este trabalho é o resultado da documentação do estudo preliminar, conduzido durante a concepção do sSECO-Process. O objetivo do trabalho foi o de modelar as redes sociais presentes no ambiente de software, e para isso foram utilizados dados do Eclipse SECO, coletados através da API fornecida pelo GitHub e contendo dados da atividade entre os anos de 2002 e 2017. A partir das contribuições nos projetos do SECO, foi construída uma rede social que utilizou uma nova

medida, do nível de contribuição do usuário em um projeto, para relacionar diferentes membros da comunidade. Mais detalhes podem ser encontrados na avaliação preliminar do sSECO-Process no Apêndice A.

5.1 Análise Comparativa

As soluções existentes apresentadas nos trabalhos relacionados são comparadas a um conjunto de características na Tabela 5.2. O trabalho (DOS SANTOS & WERNER, 2012) não é apresentado na tabela comparativa por ter o foco na parte teórica, fazendo com que as características observadas não se apliquem a ele. A escolha das características se baseou em características do sSECO-Process, como por exemplo sobre a utilização de métricas de redes complexas e das fontes de informação e estratégia de coleta.

- **Repositório de dados**, cujo objetivo descrever a fonte de informação, que relaciona as informações do SECO. As fontes avaliadas podem ser classificadas em código fonte, mensagens, outros ou não informado.

- **Fonte de informação**, representa a fonte de informação utilizada durante os estudos.

- **Forma de Coleta dos Dados**, que deve nortear a forma de utilização das fontes de dados.

- **Utilização de Métricas de Redes Complexas**, que objetiva descrever se a solução aborda métricas de redes complexas para auxílio no alcance dos objetivos estabelecidos.

- **Aplicação em Ecossistemas de Software**, que objetiva classificar os trabalhos que tratam de SECOs de maneira explícita. Nos casos onde se aplica é apresentado o SECO alvo do estudo.

- **Características de Evolução**, cujos objetivo é destacar trabalhos que tratam dos aspectos evolutivos, traçando visões que considerem o tempo durante as análises realizadas.

Tabela 5.2: Tabela comparativa dos trabalhos relacionados.

Trabalho / Característica	Dados avaliados	Fonte de Informação	Forma Coleta	Métricas de Redes Complexas	Aplicação em SECO	Evolução
(DOS SANTOS & WERNER, 2012)	Código Fonte	Não Informado	Não Informado	Não	Brechó	Sim
(MENS & GOEMINNE, 2011)	Não Informado	Não Informado	Não Informado	Não	GNOME	Parcial
(CONSTANTINOU & MENS, 2017)	Código Fonte	GitHub	GHTorrent	Não	Ruby	Sim
(BLINCOE *et al.*, 2015)	Mensagens	GitHub	GHTorrent	Sim	Múltiplos	Não

Os trabalhos apresentados neste capítulo se relacionam com esta dissertação mas existem pontos que os diferenciam deste trabalho. Em (DOS SANTOS & WERNER, 2012; MENS & GOEMINNE, 2011; CONSTANTINOU & MENS, 2017) não são utilizadas redes complexas para auxílio na análise dos relacionamentos presentes no SECOs. Os trabalhos de (DOS SANTOS & WERNER, 2012; CONSTANTINOU & MENS, 2017; BLINCOE *et al.*, 2015) não explicitam os papéis que os colaboradores do ecossistema possuem. (BLINCOE *et al.*, 2015) não realizaram suas análises considerando a evolução dos projetos e usuários analisados, e (MENS & GOEMINNE, 2011) o realizou de forma parcial, pois apenas um de seus três estudos considerou a evolução do SECO.

5.2 Considerações Finais do Capítulo

Neste capítulo foram apresentados trabalhos relacionados a Ecossistemas de Software que tem como foco a dimensão social, compreendendo trabalhos teóricos e com avaliações sobre conjuntos de dados reais. Alguns dos trabalhos utilizam características de redes complexas como métricas e visualizações. Tais trabalhos se relacionam com o sSECO-Process. O próximo capítulo apresenta as considerações finais desta dissertação.

6 Considerações Finais

Este capítulo apresenta as considerações finais sobre esta dissertação, argumentando as principais contribuições deste trabalho, suas limitações e oportunidades para trabalhos futuros.

Segundo DRESCH *et al.* (2014)(Apud Veiga, 2016) um projeto de pesquisa científica possui sete características fundamentais, que consistem na (I) criação de um artefato para (II) atender a um problema particular, (III) cuja utilidade deve ser explicitada através de uma avaliação apropriada de sua aplicabilidade e (IV) os resultados e contribuições da pesquisa devem ser compartilhados com os profissionais interessados e a academia. Para assegurar sua validade, (V) as investigações devem ser conduzidas com rigor e (VI) as possíveis formas de solução analisadas e, por fim, (VII) os resultados devem ser comunicados aos interessados.

Desta forma, esta pesquisa atende aos sete critérios fundamentais ao propor (I) o sSECO-Process para (II) identificar características de evolução do SECO, (III) avaliá-lo através da abordagem GQM e (IV) apresentar os resultados à comunidade acadêmica, através desta dissertação, dos recursos disponibilizados na internet e das publicações realizadas. Mantendo (V) o rigor metodológico durante seu desenvolvimento, (VI) analisando as soluções apresentadas no mesmo domínio e (VII) garantindo a publicação dos resultados.

6.1 Contribuições

As principais contribuições dessa pesquisa são:

- Desenvolvimento de um processo para obtenção de conhecimentos relacionados a dimensão social de SECO, a partir de repositórios de código fonte.
- Concepção de uma medida para avaliar o nível de contribuição de desenvolvedores em projetos de software, que penaliza contribuições antigas, descrita na Equação A.1.

- Concepção de uma medida para relacionar colaboradores de software a partir dos projetos que eles participam, sem a necessidade de interações diretas entre eles, descrita na Equação A.3.

- Concepção de uma medida que avalia a popularidade de colaboradores a nível de projeto Equação 4.6.

- Enriquecimento dos papéis identificados por PADHYE *et al.* (2014), acrescentando o papel de candidato, que representa um desenvolvedor com a intenção de colaborar mas que ainda não foi aceito pela comunidade.

- Criação de um modelo para importação de dados provenientes do projeto GHTorrent.

- Criação de uma base relacional que possibilita estudos que avaliem o software através do tempo, contendo além das estruturas, procedimentos para cálculo das métricas presentes na dissertação assim como procedimentos para criação de amostras

- Formação de um repositório de informações do desenvolvimento, possibilitando a replicabilidade dos estudos aqui conduzidos como extensão em trabalhos futuros.

6.2 Trabalhos Futuros

O desenvolvimento desta pesquisa levou a oportunidades que ainda estão em aberto, e podem ser exploradas em trabalhos futuros. Embora a avaliação tenha sido conduzida com rigor, novos estudos experimentais podem ser conduzidos. Como por exemplo, estudos de caso avaliando as visualizações e a consulta à comunidade OSSECO com o objetivo de avaliar o resultado da execução do processo.

A aplicação do processo por diferentes equipes também é uma oportunidade para trabalhos futuros, possibilitando novas melhorias no processo, que continuamente pode ser aprimorado para aumentar a clareza das atividades e seu nível de detalhamento. Também é uma oportunidade para trabalhos futuros generalizar o processo para sua aplicação em diferentes domínios.

Outra oportunidade de trabalhos futuros é referente à possibilidade de utilizar o conhecimento para gerar recomendações. A etapa de recomendação poderia ser acrescentada após a análise, onde os resultados obtidos são os mais refinados, e incorporados no processo seguindo o metamodelo proposto por SIMÕES *et al.* (2016).

Também é possível desenvolver uma ferramenta, que se integre com o ambiente de análise construído durante o processo, que gere visualizações e apresente as métricas para facilitar a obtenção do conhecimento. Por fim, uma outra oportunidade para trabalhos futuros é utilizar os *daily dumps* disponibilizados pelo GHTorrent com o objetivo de obter as mensagens trocadas de forma completa, possibilitando novas análises.

Referências

Aggarwal, K.; Hindle, A. ; Stroulia, E. **Co-evolution of project documentation and popularity within github**. In: Proceedings of the 11th Working Conference on Mining Software Repositories, p. 360–363. ACM, 2014.

Alves, C.; Oliveira, J. ; Jansen, S. Software ecosystem governance–a systematic literature review and research agenda. **Inf. Softw. Technol.(In submission)**, 2017.

Badashian, A. S.; Esteki, A.; Gholipour, A.; Hindle, A. ; Stroulia, E. **Involvement, contribution and influence in github and stack overflow**. In: Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, p. 19–33. IBM Corp., 2014.

Barbosa, O.; dos Santos, R.; Alves, C.; Werner, C. ; Jansen, S. A systematic mapping study on software ecosystems from a three- dimensional perspective. p. 59–81, 01 2013.

Basili, V. R. **Software modeling and measurement: the goal/question/metric paradigm**. Technical report, 1992.

Bastian, M.; Heymann, S.; Jacomy, M. ; others. Gephi: an open source software for exploring and manipulating networks. **Icwsm**, v.8, p. 361–362, 2009.

Blincoe, K.; Harrison, F. ; Damian, D. **Ecosystems in github and a method for ecosystem identification using reference coupling**. In: Proceedings of the 12th Working Conference on Mining Software Repositories, p. 202–207. IEEE Press, 2015.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R. ; Lefebvre, E. Fast unfolding of communities in large networks. **Journal of statistical mechanics: theory and experiment**, v.2008, n.10, p. P10008, 2008.

Borges, H.; Hora, A. ; Valente, M. T. **Understanding the factors that impact the popularity of github repositories**. In: Software Maintenance and Evolution (ICSME), 2016 IEEE International Conference on, p. 334–344. IEEE, 2016.

Bosch, J. **From software product lines to software ecosystems**. In: Proceedings of the 13th international software product line conference, p. 111–119. Carnegie Mellon University, 2009.

Bosch, J.; Bosch-Sijtsema, P. M. **Softwares product lines, global development and ecosystems: collaboration in software engineering**. In: Collaborative Software Engineering, p. 77–92. Springer, 2010.

Campbell, P. R.; Ahmed, F. **A three-dimensional view of software ecosystems**. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, p. 81–84. ACM, 2010.

Campbell, D. T.; Stanley, J. C. **Experimental and quasi-experimental designs for research**. Ravenio Books, 2015.

Constantinou, E.; Mens, T. **Socio-technical evolution of the ruby ecosystem in github**. In: Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on, p. 34–44. IEEE, 2017.

Cosentino, V.; Izquierdo, J. L. C. ; Cabot, J. A systematic mapping study of software development with github. **IEEE Access**, v.5, p. 7173–7192, 2017.

Dabbish, L.; Stuart, C.; Tsay, J. ; Herbsleb, J. **Social coding in github: transparency and collaboration in an open software repository**. In: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, p. 1277–1286. ACM, 2012.

de Lima Fontão, A.; dos Santos, R. P. ; Dias-Neto, A. C. **Mobile software ecosystem (mseco): a systematic mapping study**. In: Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, volume 2, p. 653–658. IEEE, 2015.

Di Tommaso, G.; Stilo, G. ; Velardi, P. **Detecting network leaders in enterprises**. In: Computer Supported Cooperative Work in Design (CSCWD), 2017 IEEE 21st International Conference on, p. 275–280. IEEE, 2017.

dos Santos, R. P.; Werner, C. M. L. **Treating social dimension in software ecosystems through reuseecos approach**. In: Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on, p. 1–6. IEEE, 2012.

Dos Santos, R. P.; Esteves, M. G. P.; Freitas, G. d. S. ; de Souza, J. M. Using social networks to support software ecosystems comprehension and evolution. **Social Networking**, v.3, n.02, p. 108, 2014.

dos Santos, R. P. **Managing and monitoring software ecosystem to support demand and solution analysis**. 2016. Tese de Doutorado - Universidade Federal do Rio de Janeiro.

Dresch, A.; Lacerda, D. P. ; Antunes Jr, J. A. V. **Design science research: A method for science and technology advancement**. Springer, 2014.

Franco-Bedoya, O.; Ameller, D.; Costal, D. ; Franch, X. Open source software ecosystems: A systematic mapping. **Information and Software Technology**, v.91, p. 160–185, 2017.

Fruchterman, T. M.; Reingold, E. M. Graph drawing by force-directed placement. **Software: Practice and experience**, v.21, n.11, p. 1129–1164, 1991.

Fuks, H.; Raposo, A. B.; Gerosa, M. A. ; Lucena, C. J. P. Do modelo de colaboração 3c à engenharia de groupware. **Simpósio Brasileiro de Sistemas Multimídia e Web–Webmidia**, p. 0–8, 2003.

Gousios, G.; Spinellis, D. **Ghtorrent: Github's data from a firehose**. In: Proceedings of the 9th IEEE Working Conference on Mining Software Repositories, p. 12–21. IEEE Press, 2012.

Gousios, G.; Pinzger, M. ; Deursen, A. v. **An exploratory study of the pull-based software development model**. In: Proceedings of the 36th International Conference on Software Engineering, p. 345–355. ACM, 2014.

Gousios, G.; Spinellis, D. **Mining software engineering data from github**. In:

Proceedings of the 39th International Conference on Software Engineering Companion, p. 501–502. IEEE Press, 2017.

Guércio, H.; Ströele, V.; David, J. M. N.; Braga, R. ; Campos, F. **Topological analysis in scientific social networks to identify influential researchers**. In: Computer Supported Cooperative Work in Design (CSCWD), 2017 IEEE 21st International Conference on, p. 287–292. IEEE, 2017.

Guércio, H.; Ströele, V.; David, J. M. N.; Braga, R. ; Campos, F. **Complex network analysis in software ecosystem: Studying eclipse community**. In: Computer Supported Cooperative Work in Design (CSCWD), 2018 IEEE 22nd International Conference on. IEEE, 2018.

Guércio, Hugo, S. V.; Campos, F. O uso de lnformações semânticas para recomendação de recursos educacionais usando grafo bipartido. **XX Conferência Internacional sobre Informática Educativa - TISE**, v.12, p. 177–187, 2016.

Hata, H.; Todo, T.; Onoue, S. ; Matsumoto, K. **Characteristics of sustainable oss projects: A theoretical and empirical study**. In: Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering, p. 15–21. IEEE Press, 2015.

Howison, J.; Crowston, K. **The perils and pitfalls of mining sourceforge**. In: Proceedings of the International Workshop on Mining Software Repositories (MSR 2004), p. 7–11. IET, 2004.

Iansiti, M.; Richards, G. L. The information technology ecosystem: Structure, health, and performance. **The Antitrust Bulletin**, v.51, n.1, p. 77–110, 2006.

Jacomy, M.; Venturini, T.; Heymann, S. ; Bastian, M. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. **PloS one**, v.9, n.6, p. e98679, 2014.

Jansen, S.; Finkelstein, A. ; Brinkkemper, S. **A sense of community: A research agenda for software ecosystems**. In: Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on, p. 187–190. IEEE, 2009.

Jarczyk, O.; Gruszka, B.; Jaroszewicz, S.; Bukowski, L. ; Wierzbicki, A. **Github projects. quality analysis of open-source software**. In: International Conference on Social Informatics, p. 80–94. Springer, 2014.

Kalliamvakou, E.; Gousios, G.; Blincoe, K.; Singer, L.; German, D. M. ; Damian, D. An in-depth study of the promises and perils of mining github. **Empirical Software Engineering**, v.21, n.5, p. 2035–2071, 2016.

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. **Journal of the ACM (JACM)**, v.46, n.5, p. 604–632, 1999.

Lehman, M. M.; Belady, L. A. **Program evolution: processes of software change**. Academic Press Professional, Inc., 1985.

Lehman, M. M. **Laws of software evolution revisited**. In: European Workshop on Software Process Technology, p. 108–124. Springer, 1996.

Lehman, M. M.; Ramil, J. F.; Wernick, P. D.; Perry, D. E. ; Turski, W. M. **Metrics and laws of software evolution-the nineties view**. In: Software metrics symposium, 1997. proceedings., fourth international, p. 20–32. IEEE, 1997.

Li, S.; Tsukiji, H. ; Takano, K. **Analysis of software developer activity on a distributed version control system**. In: 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), p. 701–707. IEEE, 2016.

Lungu, M.; Lanza, M.; Gîrba, T. ; Robbes, R. The small project observatory: Visualizing software ecosystems. **Science of Computer Programming**, v.75, n.4, p. 264–275, 2010.

Manikas, K.; Hansen, K. M. Software ecosystems–a systematic literature review. **Journal of Systems and Software**, v.86, n.5, p. 1294–1306, 2013.

Manikas, K. Revisiting software ecosystems research: A longitudinal literature study. **Journal of Systems and Software**, v.117, p. 84–103, 2016.

Martin, S.; Brown, W. M.; Klavans, R. ; Boyack, K. W. **Openord: An open-source toolbox for large graph layout.** In: Visualization and Data Analysis, p. 786806, 2011.

Mens, T.; Goeminne, M. **Analysing the evolution of social aspects of open source software ecosystems.** In: IWSECO@ ICSOB, p. 1–14, 2011.

Mens, T.; Grosjean, P. The ecology of software ecosystems. **Computer**, v.48, n.10, p. 85–87, 2015.

Messerschmitt, D. G.; Szyperski, C. ; others. Software ecosystem: understanding an indispensable technology and industry. **MIT Press Books**, v.1, 2005.

Nasserifar, J. **Open Source Software Ecosystem: A Systematic Literature Review**. Finland, 2016. Dissertação de Mestrado - University of Oulu.

Onoue, S.; Hata, H. ; Matsumoto, K.-i. **A study of the characteristics of developers' activities in github**. In: Software Engineering Conference (APSEC), 2013 20th Asia-Pacific, volume 2, p. 7–12. IEEE, 2013.

Ossher, J.; Bajracharya, S. ; Lopes, C. **Automated dependency resolution for open source software**. In: Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on, p. 130–140. IEEE, 2010.

Padhye, R.; Mani, S. ; Sinha, V. S. **A study of external community contribution to open-source projects on github**. In: Proceedings of the 11th Working Conference on Mining Software Repositories, p. 332–335. ACM, 2014.

Palomba, F.; Serebrenik, A. ; Zaidman, A. **Social debt analytics for improving the management of software evolution tasks**. In: 16th Edition of the BElgian-NEtherlands Software EVOLution Symposium, BENEVOL 2017. CEUR-WS. org, 2017.

Ramil, J. F.; Lehman, M. M. **Metrics of software evolution as effort predictors-a case study**. In: icsm, p. 163. IEEE, 2000.

Rastogi, A.; Nagappan, N. **Forking and the sustainability of the developer community participation–an empirical investigation on outcomes and reasons**.

In: Software Analysis, Evolution, and Reengineering (SANER), 2016 IEEE 23rd International Conference on, volume 1, p. 102–111. IEEE, 2016.

Riehle, D.; Ellenberger, J.; Menahem, T.; Mikhailovski, B.; Natchetoi, Y.; Naveh, B. ; Odenwald, T. Open collaboration within corporations using software forges. **IEEE software**, v.26, n.2, p. 52–58, 2009.

dos Santos, R. P.; Werner, C. M. L. **A proposal for software ecosystems engineering.** In: IWSECO@ ICSOB, p. 40–51, 2011.

dos Santos, R. P.; Werner, C. **Treating business dimension in software ecosystems**. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, p. 197–201. ACM, 2011.

dos Santos, R. P.; Werner, C. M. L. **Reuseecos: An approach to support global software development through software ecosystems**. In: Global Software Engineering Workshops (ICGSEW), 2012 IEEE Seventh International Conference on, p. 60–65. IEEE, 2012.

Schulze, C.; Newell, B. R. More heads choose better than one: Group decision making can eliminate probability matching. **Psychonomic bulletin & review**, v.23, n.3, p. 1–8, 2016.

Simões, L.; Almeida, R.; Campos, F.; Ströele, V.; David, J. M.; Braga, R. ; Guércio, H. Mmrecommender: Metamodelo de sistemas de recomendação aplicado a grupos educacionais. **XXI Congresso Internacional de Informática Educativa (TISE)**, p. 505–510, 11 2016.

Storey, M.-A.; Singer, L.; Cleary, B.; Figueira Filho, F. ; Zagalsky, A. **The (r) evolution of social media in software engineering**. In: Proceedings of the on Future of Software Engineering, p. 100–116. ACM, 2014.

Treude, C.; Figueira Filho, F. ; Kulesza, U. **Summarizing and measuring development activity**. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, p. 625–636. ACM, 2015.

Triola, M. F. **Elementary statistics**. Pearson/Addison-Wesley Reading, MA, 2006.

van den Berk, I.; Jansen, S. ; Luinenburg, L. **Software ecosystems: a software ecosystem strategy assessment model**. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, p. 127–134. ACM, 2010.

Yamashita, K.; McIntosh, S.; Kamei, Y. ; Ubayashi, N. **Magnet or sticky? an oss project-by-project typology**. In: MSR, 2014.

Yamashita, K.; Kamei, Y.; McIntosh, S.; Hassan, A. E. ; Ubayashi, N. Magnet or sticky? measuring project characteristics from the perspective of developer attraction and retention. **Journal of Information Processing**, v.24, n.2, p. 339–348, 2016.

A sSECO-Process Preliminar

Neste capítulo, a solução proposta nesta dissertação é apresentada, com o objetivo de esclarecer as etapas e atividades relevantes ao se estudar a dimensão social dos Ecossistemas de Software.

A.1 Definição do Processo

Os SECO possuem um ambiente para colaboração, e este é o ambiente que deve ser observado a fim de se aumentar o entendimento dos processos e atividades existentes nos ecossistemas. A natureza da abordagem de SECO, em essência, nos mostra que o ambiente compartilhado existente tenha recursos que possam reduzir as dificuldades no desenvolvimento de software geograficamente distribuído.

Neste cenário, ferramentas são necessárias para que os envolvidos possam colaborar e ter visibilidade das ações de cada um dos participantes e das necessidades do ecossistema. Tais ferramentas estão disponíveis e têm como objetivo auxiliar na comunicação, visibilidade, armazenamento e versionamento, dando meios para que os envolvidos possam colaborar e atingir os objetivos da comunidade. Além de fornecer meios para colaboração, essas ferramentas produzem uma grande quantidade de dados, que registram as ações, características e interações entre os envolvidos no ecossistema, além de relacioná-los com os artefatos presentes no ambiente compartilhado.

Com o objetivo de auxiliar o estudo da dimensão social dos SECOs, esta dissertação propõem um processo que auxilia os cientistas a identificar aspectos importantes, contando como um ponto de partida para a análise da perspectiva social de SECOs e extração de conhecimento a partir desta observação.

A Figura A.1 apresenta uma visão global do processo para auxílio do estudo da dimensão social de um SECO. O processo compreende as atividades de coleta e preparação assim como atividades para avaliar os dados que possibilitam a análise. Essas atividades estão descritas a seguir com maiores detalhes.

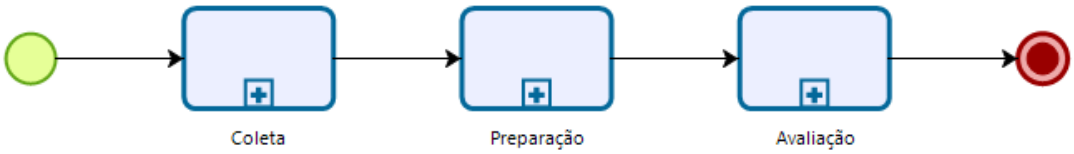


Figura A.1: Visão global do processo.

A atividade de coleta dos dados foi modelada em um subprocesso apresentado na Figura A.2. A primeira tarefa a ser realizada é a identificação das fontes de dados, que deve ser realizada pelo cientista a partir da observação da literatura e das diferentes comunidades de desenvolvimento. O cientista deve identificar quais as fontes de dados que podem conter elementos para análise do SECO. Após a identificação das fontes de

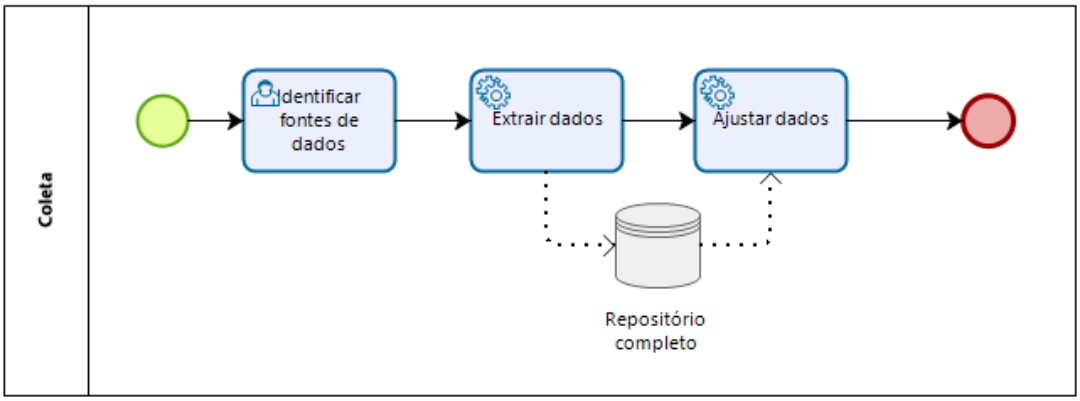


Figura A.2: Subprocesso de coleta.

dados, um desenvolvedor fica responsável por realizar a extração dos dados a partir das fontes identificadas na etapa anterior. Estes dados são armazenados em um repositório, que é tratado para retirar inconsistências nos dados. Esta atividade é realizada pelo desenvolvedor na tarefa de ajuste dos dados. Após o ajuste, os dados estão prontos para a próxima etapa do processo aqui definido.

Na Figura A.3 o subprocesso de preparação é descrito. As atividades a serem realizadas durante a preparação são realizadas a partir dos dados coletados na etapa anterior.

A primeira atividade é a definição da amostra a ser estudada. Esta amostra é definida pelo cientista a partir do seu objetivo. Caso ele deseje estudar uma parcela específica da população, esta é a etapa responsável por definir os elementos pertencentes a esta parcela, indentificado-os e delimitando o escopo da observação. Se ele desejar

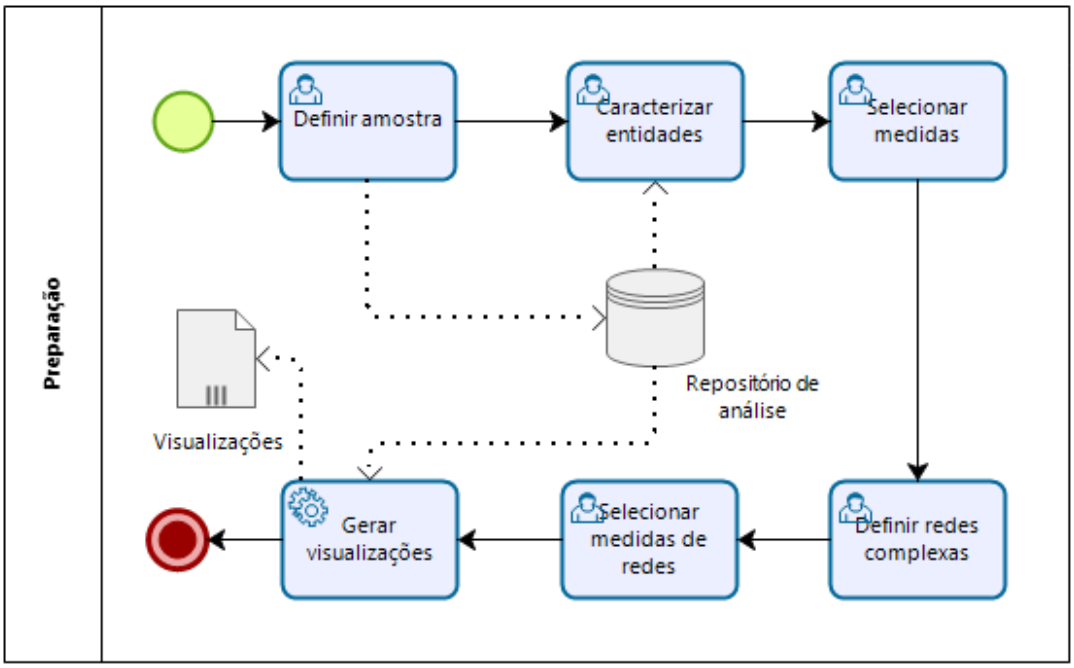


Figura A.3: Subprocesso de preparação.

estudar o SECO com uma perspectiva geral, mas sem a possibilidade de tratar e consumir todos os dados, ele pode optar por realizar uma amostragem com o propósito de reduzir a quantidade de dados a serem avaliados, economizando recursos e facilitando a manipulação dos dados. Após a definição da amostra o cientista fica responsável por caracterizar as entidades de acordo com suas características e em seguida, selecionar as medidas que o auxiliarão em suas análises.

Durante a preparação, o cientista identifica os elementos disponíveis no repositório de análise a fim de diferenciar os elementos cujas relações podem ser representadas e estudadas por redes complexas para auxílio durante a avaliação. Nesta etapa, o cientista identifica os nós e seus relacionamentos e após a definição das redes o cientista seleciona quais medidas, relacionadas às redes modeladas, serão utilizadas durante a avaliação. Por fim, o cientista gera visualizações das redes com o objetivo de possibilitar análises posteriores.

A etapa de análise é apresentada na Figura A.4, que mostra o subprocesso de avaliação. Nesta etapa, o cientista utiliza-se das visualizações produzidas anteriormente. O uso das visualizações possibilita que os cientistas identifiquem elementos importantes ou outras informações que tem como propósito auxiliá-lo a alcançar os objetivos propostos. Após a avaliação das redes, o cientista fica responsável por examinar as medidas dis-

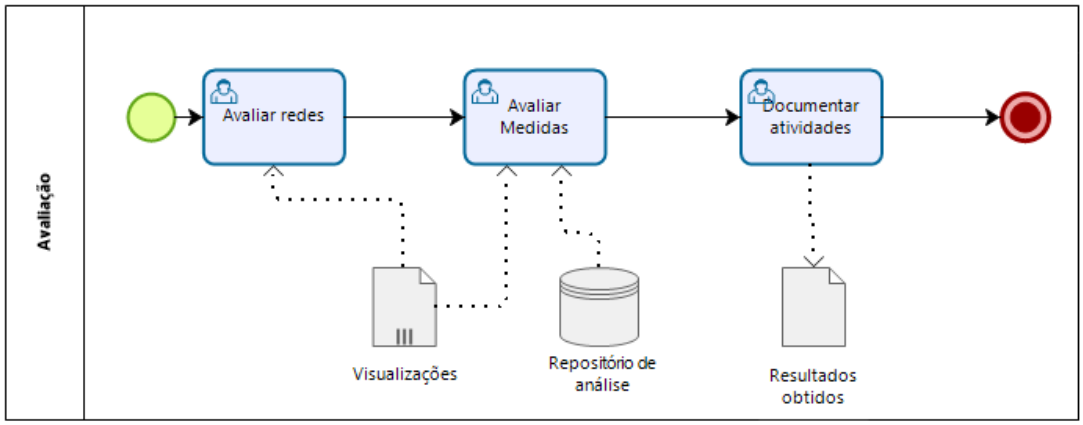


Figura A.4: Subprocesso de avaliação.

poníveis com o intuito de satisfazer as necessidades que o levaram a instanciar o processo de análise dos ecossistemas. Por fim, o cientista realiza a documentação das atividades com os resultados obtidos e lições aprendidas, a fim de indicar a outros cientistas, com necessidades semelhantes, quais são os caminhos que os levam a resultados mais positivos.

A.2 Avaliação Preliminar

Esta seção apresenta um estudo preliminar executado durante o desenvolvimento deste trabalho. Este estudo foi realizado com o intuito de avaliar o processo desenvolvido, para identificar os principais pontos do processo que poderiam ser melhorados a fim de enriquecer os resultados obtidos a partir da sua instanciação.

Para tal, o processo foi instanciado com o objetivo de colocar em prática as ações e atividades, e evidenciar os pontos fracos assim como suas forças. O objetivo é verificar a viabilidade do processo proposto através da análise de um ecossistema real de desenvolvimento de software.

Durante a execução do estudo preliminar, foram realizadas reuniões de acompanhamento com especialistas em desenvolvimento de software e cientistas com competências relacionadas com o estudo de SECO. Este acompanhamento foi realizado com o propósito de adequar o fluxo de execução das atividades, e identificar novas atividades a serem desenvolvidas.

O estudo preliminar utilizou dados do ecossistema Eclipse, disponíveis no GitHub e extraídos do através da sua API. Os dados extraídos foram tratados e sintetizados em

um conjunto de dados, que foi utilizado para aplicar métricas identificadas na literatura, relacionadas ao desenvolvimento de software e redes complexas. Por fim, foram geradas visualizações e realizadas análises com relação à evolução do SECO.

Ao fim do estudo preliminar foi realizada uma reavaliação das atividades de acordo com os resultados obtidos, com o objetivo de buscar melhorias no processo e auxiliar no amadurecimento do mesmo. Além disso, os resultados também foram documentados e disponibilizados para a comunidade científica (GUÉRCIO *et al.*, 2018).

A.2.1 Planejamento do Estudo Preliminar

O estudo preliminar teve como objetivo gerar conhecimento através da análise de redes complexas modeladas a partir de dados extraídos de um SECO. As redes complexas foram selecionadas como fonte de informação visto que através delas é possível modelar os relacionamentos presentes entre diferentes entidades. Desta forma, a representação das redes sociais se justifica por dar ênfase nas relações entre os participantes.

Para tal, foi seguido o modelo GQM (*Goal - Question - Metric*) que determina que devem ser definidos objetivos do estudo, seguidos pelas questões de pesquisa e métricas para avaliação das questões de pesquisa.

Seguindo o arcabouço proposto por BASILI (1992), tem-se como objetivo **Analisar** o processo de extração de informação **com o objetivo** de analisar redes complexas da dimensão social de SECO **em relação** ao suporte no desenvolvimento global de software **do ponto de vista de** *stakeholders* **no contexto** de Ecossistemas de Software.

Segundo BASILI (1992), o fluxo dos objetivos até as métricas no paradigma GQM pode ser visualizado através de um grafo direcionado, onde o fluxo se inicia no nó de objetivo, passa pelos nós que representam as questões, chegando nos nós de métrica. A Figura A.5 apresenta o fluxo desta avaliação.

Goal

- **G1:**Analisar um ecossistema a partir da sua dimensão social, modelando redes a partir dos dados disponíveis de um ecossistema e das relações sociais presentes, assim como os colaboradores de maior importância.

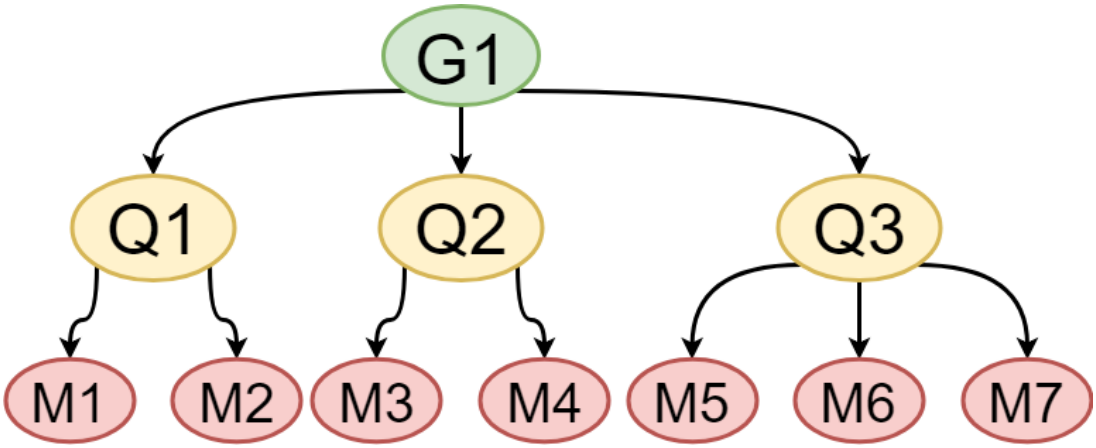


Figura A.5: Grafo direcionado representando a abordagem GQM.

Questions

- **Q1:** O uso do processo compreende as atividades necessárias para extração de informação?

  Objetivo: verificar se o processo possui todas as atividades necessárias para extração de informação capaz de gerar conhecimento explícito.

- **Q2:** O uso de redes complexas pode auxiliar os stakeholders na identificação dos desenvolvedores que mais contribuíram no desenvolvimento global de software?

  Objetivo: verificar se visualizações e métricas de redes complexas auxiliam na identificação dos desenvolvedores que mais contribuem sob a perspectiva dessas métricas.

- **Q3:** O processo auxilia na análise da colaboração em projetos de desenvolvimento de software?

  Objetivo: avaliar a colaboração através da análise da cooperação, comunicação e coordenação ao longo do tempo.

Metrics

- **M1:** Quantidade de atividades desenvolvidas durante a execução do processo.
- **M2:** Quantidade de atividades identificadas na revisão do processo por especialistas.
- **M3:** Quantidade de redes modeladas.
- **M4:** Quantidade de desenvolvedores identificados que mais contribuem

- **M5:** Quantidade de contribuições no ecossistema no tempo
- **M6:** Quantidade de comentários no ecossistema no tempo
- **M7:** Quantidade de contribuições rejeitadas no ecossistema no tempo

Métricas quantitativas foram selecionadas para apoiar a resposta das perguntas pois os resultados entregues são avaliados desta forma pelos *stakeholders*.

A.2.2 Execução do Estudo Preliminar

O estudo preliminar foi conduzido de acordo com o processo modelado neste capítulo. A Figura A.6 mostra na parte superior as atividades realizadas de acordo com o processo, e na parte inferior alguns dos resultados e decisões tomadas.

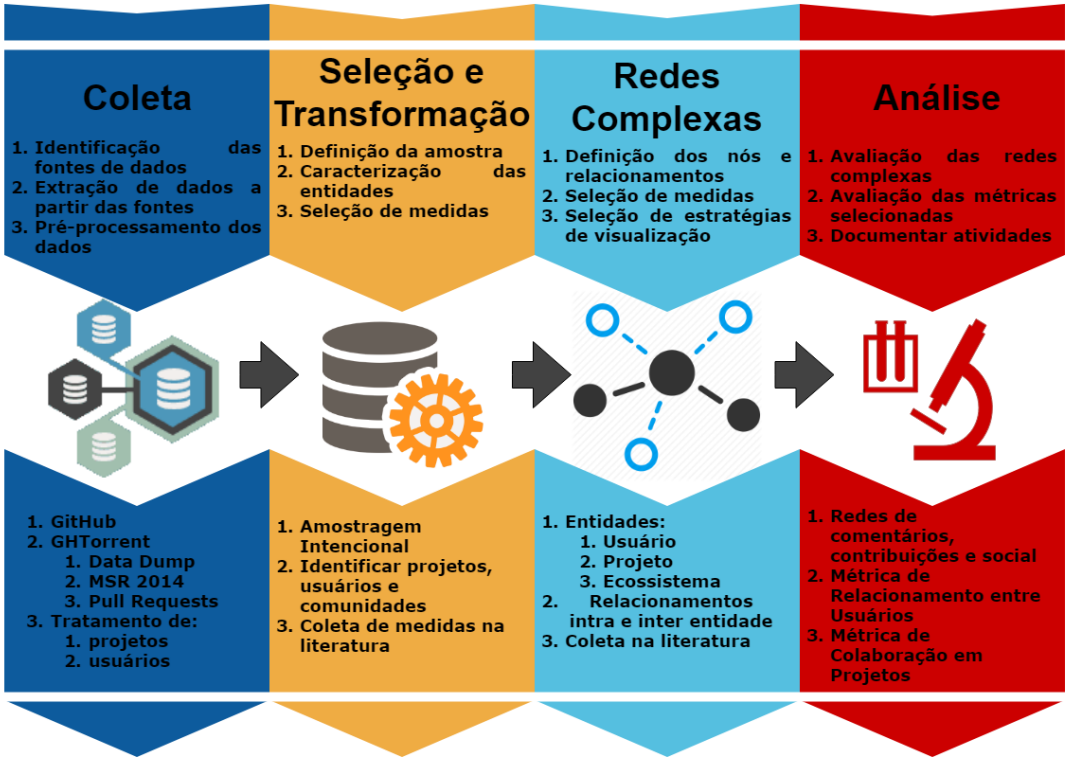


Figura A.6: Fluxo de atividades desenvolvidas durante o estudo preliminar.

Coleta

A primeira etapa que compreende o estudo preliminar é referente à atividade de coleta, que seguiu as três atividades apresentadas na Figura A.2. A atividade de identificação das fontes de dados teve como resultado o GitHub, pois ele representa a nova geração

dos *software forges* por combinar os aspectos clássicos de hospedagem e versionamento de código com características sociais (COSENTINO *et al.*, 2017). Essas características têm como objetivo facilitar a colaboração e interações sociais sobre os projetos presentes na plataforma com diferentes funcionalidades como *issue trackers* ou suporte a *pull-requests*. As novas funcionalidades foram aceitas pela comunidade, visto que a adoção do GitHub, como ferramenta para apoiar a colaboração em desenvolvimento, é crescente, sendo hoje a maior plataforma do segmento (GOUSIOUS & SPINELLIS, 2017).

COSENTINO *et al.* (2017) conduziu um mapeamento que reúne estudos sobre desenvolvimento de software através do GitHub. Neste mapeamento, ele identifica as abordagens utilizadas pela literatura para extração dos dados dividindo-as de acordo com a ferramenta utilizada. A partir desta estratégia de divisão, foram identificadas 6 abordagens para coleta dos dados, sendo elas: (1)GHTorrent (GOUSIOS & SPINELLIS, 2012), (2) GitHub Archive[22], (3) GitHub API[23], (4) manualmente, (5) outras ferramentas e (6) uma mistura das abordagens anteriores, sintetizadas pelos autores na Figura A.7.

De acordo com o mapeamento realizado, foi identificado que, para trabalhos que estudam software a partir dos dados presentes no Github, as abordagens da GitHub API e GHTorrent se destacam das demais. Avaliando os estudos que utilizam o GHTorrent como maneira de extração, foram identificadas as principais formas de utilização, sendo elas a carga completa dos dados coletados através de um *data dump*, a utilização de um *dataset* criado com o propósito de possibilitar estudos da conferência de mineração de repositórios de software de 2014, e um *dataset* que dá ênfase aos estudos que abordam *pull-requests*.

Durante o estudo preliminar, foi escolhida a abordagem de extração a partir da API fornecida pelo GitHub, visto que a possibilidade de consultar todos os possíveis elementos disponíveis na ferramenta apresentava indicativos de análises mais ricas.

As chamadas a API do GitHub foram realizadas através de *scripts* que foram executados a partir de NodeJS[24]. Os *scripts* estão disponíveis no GitHub[25] e foram utilizados para obtenção de dados de repositórios de um usuário arbitrário. O SECO selecionado

[22]https://www.githubarchive.org/
[23]https://developer.github.com/v3/
[24]https://nodejs.org/
[25]https://github.com/hugoguercio/sSECO/tree/master/NODE_SCRIPTS

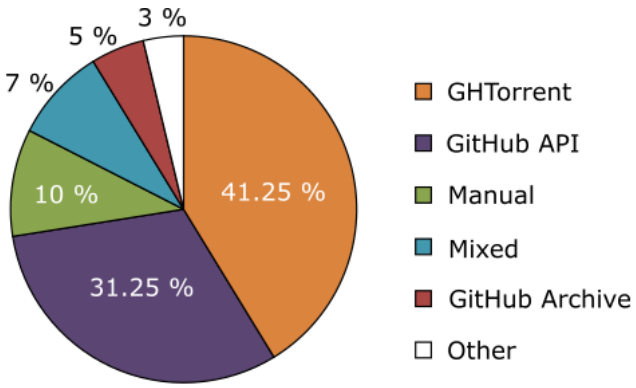


Figura A.7: Ferramentas utilizadas para coleta de dados do GitHub (COSENTINO *et al.*, 2017).

foi o da Eclipse Foundation[26], que representa um ecossistema saudável e com uma grande quantidade de projetos. Os dados foram coletados desde o início do ecossistema até a segunda semana de outubro de 2017 e armazenados em uma base relacional *Postgres*.

Esta abordagem possui benefícios como a aquisição de dados recentes em tempo real, mas foram encontrados pontos negativos pois nem todos os eventos retornaram os resultados esperados, como, por exemplo, a contagem de *stars* e *forks*. Outro ponto negativo são as restrições da quantidade de requisições, que limitam o processo de extração e tornam o processo muito demorado. Em requisições não autenticadas, o limite é de 60 por hora. Já para requisições autenticadas este limite sobe para 5000, entretanto, a quantidade de dados disponíveis é abundante, fazendo com que o processo de coleta se torne lento.

Para contornar os problemas identificados e garantir a consistência foram utilizadas duas abordagens. A primeira, que visa contornar os problemas nas respostas das requisições, foi a de utilizar um *scrapper*, um *software* que tem como objetivo extrair dados de páginas web, para atualizar as informações com base nos dados presentes na plataforma do eclipse através de navegação. Para isso, foi utilizado como base um *scrapper* disponível para utilização com fins educacionais, disponível no github[27].

Após as atividades de extração, foi realizado um pré-processamento dos dados, para identificar inconsistências nas informações coletadas. Durante o pré-processamento foram identificadas datas inconsistentes, como, por exemplo, registro dos anos de 1970 e

[26]https://github.com/eclipse
[27]https://github.com/nelsonic/github-scraper

2099, que foram removidos. A quantidade de estrelas de projetos também precisou ser ajustada visto que a informação respondida pela API era diferente da apresentada na página web. Desta forma, os dados foram atualizados a partir do *scrapper* utilizado na etapa anterior.

A execução da atividade de coleta teve como resultado dados gerados, do SECO Eclipse Foundation, entre junho de 2002 até a segunda semana de outubro de 2017, totalizando 499 projetos, mais de 2200 colaboradores e pouco mais de 286.000 *commits*. Também foram coletados 36.815 comentários.

Preparação

TRIOLA (2006) afirma que se os dados amostrais não forem coletados de maneira apropriada, os dados podem se tornar inúteis e, mesmo com diferentes tratamentos estatísticos, não poderá ser salvo. Desta forma, torna-se importante selecionar os indivíduos da amostra de uma maneira que eles sejam parte representativa da população.

De acordo com a abordagem de extração e as dificuldades encontradas, principalmente com a quantidade de requisições, a amostra foi definida de maneira não probabilística. Foi utilizada a técnica de amostragem intencional, onde os elementos da amostra são selecionados a partir do conhecimento da população e do propósito do estudo. Os elementos selecionados foram os associados ao ecossistema Eclipse.

As entidades extraídas foram caracterizadas em repositórios e usuários. Os repositórios foram subdivididos em duas categorias, sendo elas a de repositórios bases, que foram criados sem utilizar nenhum repositório como origem. Repositórios secundários, foram criados a partir de uma ramificação de algum outro repositório.

Também foram identificados usuários que são divididos pelo GitHub em dois tipos, sendo eles usuários ou organizações. Usuários do tipo organização representam contas compartilhadas, onde grupos de pessoas podem colaborar em vários projetos. Os usuários do tipo usuário caracterizam contas pessoais, que dão acesso a quantidades ilimitadas de repositórios e a possibilidade de interagir com outros projetos.

A partir da divisão do GitHub, foi realizada uma subdivisão dos usuários durante esse trabalho. Essa divisão considerou o tipo de contribuição de cada um com os

projetos. Desta forma, os usuários foram divididos entre os que participaram através de código e usuários que participaram através de comentários. Por último, os usuários foram separados através dos dados disponíveis no GitHub, em grupos referentes à quantidade de projetos distintos em que o usuário colaborou. No escopo deste estudo preliminar, os projetos representam um repositório base e todos os repositórios criados utilizando este repositório base como ponto de partida. Neste estudo, os *commits* foram considerados como medida de contribuição ao ecossistema.

Durante a seleção de medidas para análise de colaboração na dimensão social do ecossistema, além dos dados quantitativos de contribuição nos projetos, seja ela textual ou código, fez-se necessário desenvolver medidas que englobassem os aspectos temporais. Outra necessidade identificada durante o estudo preliminar foi definir uma forma de relacionar os usuários, visto que a partir dos dados coletados não existiam ligações explícitas entre dois pares de usuários. Desta maneira, foram propostas e desenvolvidas duas medidas para relacionar explicitamente os usuários da dimensão social do SECO:

- **Nível de Contribuição**: medida que visa identificar o nível de contribuição CL_{mp} de um determinado indivíduo m em um projeto p. Esta medida considera o tempo em seu cálculo, visto que o colaborador modifica sua forma de colaboração com o passar do tempo. Para isso, é considerada a diferença temporal entre a data do seu último *commit* neste repositório com o período de observação dos dados.

  Quanto maior o nível de contribuição, maior será a chance de um colaborador efetuar uma nova contribuição neste projeto. Desta forma, para cada semana w que um colaborador m realizou um *commit* c em um projeto p, é calculada a diferença entre o tempo de observação t_0 e a data t_c, referente a outro *commit* no mesmo repositório p, de acordo com Equação A.1.

$$CL_{mp} = \sum_{h=1}^{w} \frac{1}{(t_0 - t_c)} c \tag{A.1}$$

  Essa equação tem como objetivo penalizar as contribuições mais antigas, dando maior importância para as contribuições recentes, mas sem deixar de considerar todo o histórico de um colaborador do projeto. Os valores foram normalizados entre 0 e 1,

onde os valores mais próximos a 0 indicam um menor nível de colaboração recente, dando indícios de que o usuário está inativo. Valores próximos a 1 indicam que o usuário está colaborando recentemente no projeto, que é um indicativo de que o usuário tem maior probabilidade de estar ativo no momento. Esta atividade recente indica um recurso disponível que pode ser utilizado com o objetivo de continuar com a evolução dos artefatos e estreitando a comunidade dos seus objetivos.

- **Relacionamento por Contribuição**: A partir dos níveis de contribuição dos usuários em projetos, foi criado um relacionamento entre pares de desenvolvedores a partir das contribuições em código fonte realizadas pelos colaboradores.

  Esse *relacionamento por contribuição*, R, entre dois colaboradores i e j foi definido a partir dos projetos, que possuem contribuições dos dois colaboradores.

$$n_{ij} = P(i) \bigcap P(j) \tag{A.2}$$

  Esse relacionamento soma todos os CL entre pares de cientistas para cada projeto que eles tenham em comum (Equação A.2) e dividido pela soma do total de projetos P para cada colaborador. Todos os valores foram normalizados entre 0 e 1, onde valores mais próximos de 1 indicam um relacionamento mais forte entre os pares de usuários.

$$R_{ij} = \frac{\sum_{k \epsilon n_{ij}}^{n_{ij}} (CL_{ik} + CL_{jk})}{\|P\,(i)\| + \|P\,(j)\|} \tag{A.3}$$

  A Equação A.3 resume o cálculo de R, que tem como objetivo capturar o quanto um par de usuários está alinhado através da sua colaboração recente, onde i e j representam dois colaboradores, P representa o conjunto de projetos que cada colaborador participou, CL representa o nível de contribuição de um colaborador em um projeto k, comum ao par de colaboradores.

A etapa de modelagem das redes complexas, deu origem a três redes diferentes. A primeira rede modelada foi a rede de contribuições. Esta rede possui dois tipos de nós, que são usados para relacionar projetos e participantes. O relacionamento entre os nós é a existência de alguma participação, através de contribuição no código, do participante

no projeto.

A segunda rede modelada foi a rede de comentários, que também relaciona usuários e projetos, mas os relacionamentos foram criados a partir da existência de comentários. A última das redes modeladas relaciona apenas os usuários através do relacionamento definido na Equação A.3.

Para a análise das redes no estudo preliminar foi considerada a centralidade de *betweenness*, que representa a quantidade de caminhos mínimos que passam por um determinado nó. A partir desta centralidade, é possível identificar nós críticos, que representam possíveis pontes. A saída deste nó pode impactar a rede como um todo, visto que as comunicações tendem a passar mais vezes pelo nó, e com sua saída a informação pode deixar de chegar a certos pontos da rede.

Finalizando as atividades de preparação, foram produzidas visualizações a partir dos dados coletados e das redes estabelecidas. As visualizações produzidas neste trabalho foram criadas a partir do software Gephi, apresentado por BASTIAN *et al.* (2009), um software *open source* para análise de grafos e redes complexas.

Análise

Após a coleta dos dados, foi possível avaliar, quantitativamente, as contribuições no GitHub, assim como a frequência de atividades. A avaliação foi realizada considerando 499 projetos, e cada um com os 100 colaboradores mais ativos, ordenados pela quantidade de *commits* realizados.

Do total de repositórios avaliados, pouco mais de um quinto possuíam comentários, totalizando 104 repositórios distintos. O conjunto de comentários foi gerado por aproximadamente 31% dos colaboradores no ecosistema, evidenciando que a maioria dos usuários não participa na troca de mensagens.

Outra avaliação realizada foi com relação à quantidade de projetos distintos que um determinado colaborador participou. Esta avaliação foi realizada com o intuito de identificar se os participantes fazem parte de múltiplos projetos de desenvolvimento ou se o comportamento normal é participar em apenas um projeto.

Com o objetivo de distinguir os colaboradores de acordo com a quantidade de

projetos em que eles contribuíram foram criados quatro grupos, para sintetizar as análises. O primeiro grupo compreende participantes que contribuíram em apenas um projeto do ecossistema, e esse grupo representa mais de 65% dos colaboradores avaliados, totalizando 1445 usuários. O segundo grupo compreende os usuários que colaboraram em mais de um projeto até um total de cinco. A média de projetos por colaborador é de 2,2 e o desvio padrão é de 3,6. Desta forma, o segundo grupo visa capturar os usuários que estão até um desvio padrão do conjunto. Este grupo possui 601 membros, que correspondem a pouco mais de 27%. O terceiro grupo é formado por usuários que colaboraram de 6-10 projetos distintos, resultando em 89 usuários. Por fim, o último grupo representa os usuários que colaboraram em mais de 10 projetos, que compreende 65 usuários. A Figura A.8 sintetiza essa informação.

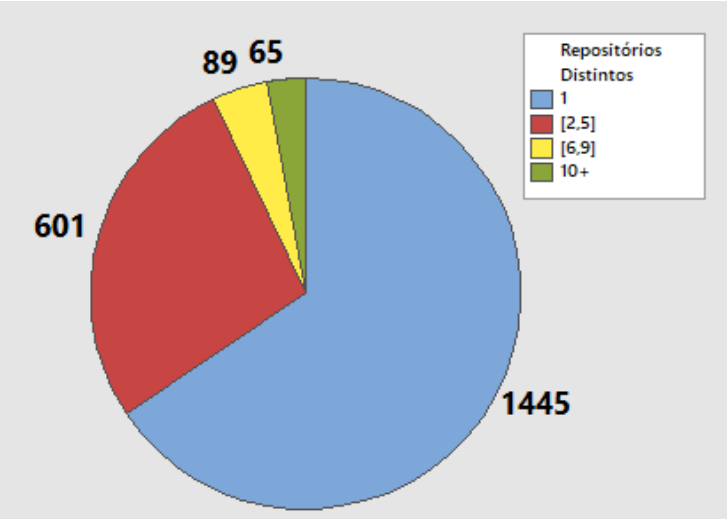


Figura A.8: Distribuição dos usuários com relação à quantidade de repositórios.

Após a separação dos usuários por grupos, é possível notar que a maioria dos usuários colabora em apenas um projeto, mas que existe uma quantidade considerável de participantes que estão presentes em múltiplos projetos. Desta forma, ao aumentar a quantidade de colaboradores que participam em mais de um projeto espera-se que o ecossistema estreite os laços entre as comunidades, e que isso possa auxiliar em momentos de instabilidade, dando mais robustez ao SECO.

A rede de contribuições é apresentada na Figura A.9, que mostra como os usuários estão distribuídos sob a perspectiva dos grupos descritos anteriormente. Projetos são representados por nós pretos e os grupos 1,2,3 e 4 são apresentados, respectivamente, em

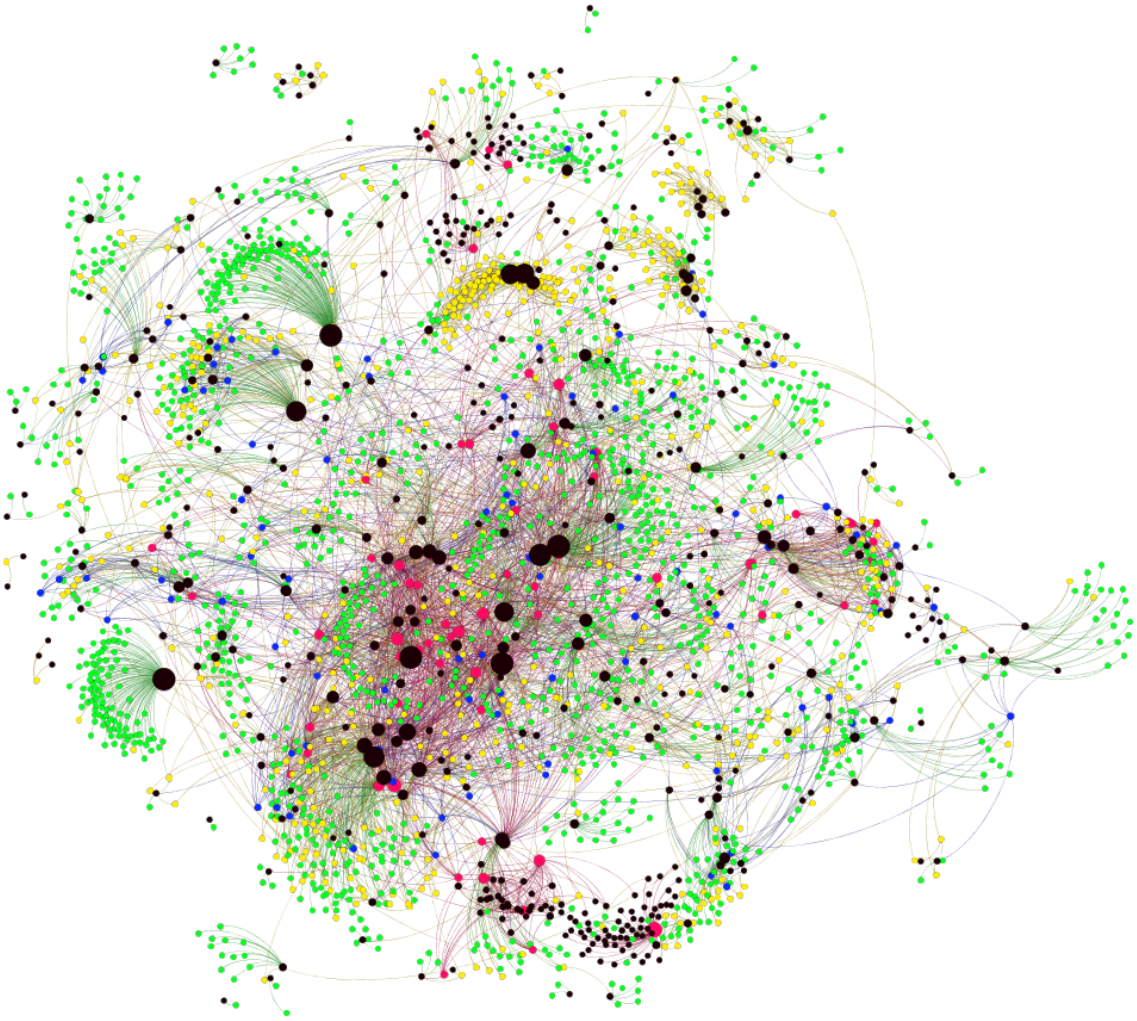


Figura A.9: Rede de contribuições do Eclipse. Nós pretos representam repositórios e os participantes pertencentes aos grupos 1, 2, 3 e 4 são representados, respectivamente, em verde, amarelo, azul e vermelho.

verde, amarelo, azul e vermelho.

É possível notar que os colaboradores dos grupos 3 e 4 estão localizados na parte central da rede. Isso é um reflexo do modelo de distribuição utilizado para representar visualmente a rede modelada. Inicialmente, a técnica de distribuição OpenOrd (MARTIN *et al.*, 2011) foi utilizada visto que ela tem como objetivo aumentar a distinção entre *clusters*. Após aplicar o algoritmo OpenOrd, foi realizada uma contração para reduzir o espaço de apresentação do grafo, e por fim, o algoritmo de Fruchterman-Reingold (FRUCHTERMAN & REINGOLD, 1991) foi aplicado aumentando a uniformidade referente à distância entre nós. Além disso, cada nó teve seu tamanho ajustado de acordo com seu grau, para destacar usuários e repositórios. O tamanho teve um limite superior, prevenindo que alguns nós pudessem sobrepor partes da rede complexa apresentada.

A rede complexa apresentada possui 2666 nós, com 2199 colaboradores e 467 projetos. O usuário de login *eclipsewebmaster*, que representa o time Eclipse, foi removido da representação visual visto que ele possui contribuições pontuais na maioria dos projetos, mas não representa um participante. Além disso, 32 projetos foram removidos, pois 14 não possuíam participantes e 18 possuíam apenas *commits* do usuário *eclipsewebmaster*. Após as remoções, foram identificadas 21 componentes conexas, onde a componente principal possui por volta de 2400 nós e as outras componentes possuem menos de 100 nós.

Com o objetivo verificar a correlação entre a quantidade de participantes e de contribuições foi construído um gráfico de dispersão das duas variáveis. Além disso, também foi realizado o cálculo das correlações de Pearson e Spearman, verificando se elas eram linearmente proporcionais ou se existia uma relação monotônica. O valor da correlação de Pearson foi de 0,888 e da correlação de Spearman foi de 0,964. O intervalo das variações é de -1 até 1, onde os valores mais próximos de 1 indicam forte correlação positiva e valores próximos a -1 indicam alta correlação negativa. A partir deste cálculo foi identificada uma forte relação monotônica, pois o valor da correlação de Spearman é muito próximo a 1.

Os dados coletados foram então fatiados temporalmente, capturando aspectos da evolução do ecossistema. Os dados avaliados correspondem a atividade registrada entre os anos de 2006 e 2016. Neste período, o ecossitema teve duas etapas distintas. Até o ano de 2011 a quantidade de *commits* e usuários manteve um crescente, tanto dos usuários quanto da quantidade dos *commits*. Em 2010 o ecossistema teve um salto na quantidade de usuários e contribuições mas nos anos subsequentes o crescimento da quantidade de contribuições se manteve estável.

A Figura A.10 apresenta a quantidade de participantes e contribuições entre os anos de 2006 e 2016. O gráfico é apresentado em escala logarítmica visto que desta forma é possível combinar diferentes intervalos, considerando que os participantes estão na casa de centenas e as contribuições na casa dos milhares. Além disso, esta escala auxilia na avaliação temporal, pois este tipo de escala representa melhor taxas de crescimento, evitando que os dados sejam representados sempre no formato de cauda pesada, não deixando claro a taxa de crescimento com relação ao ponto anterior.

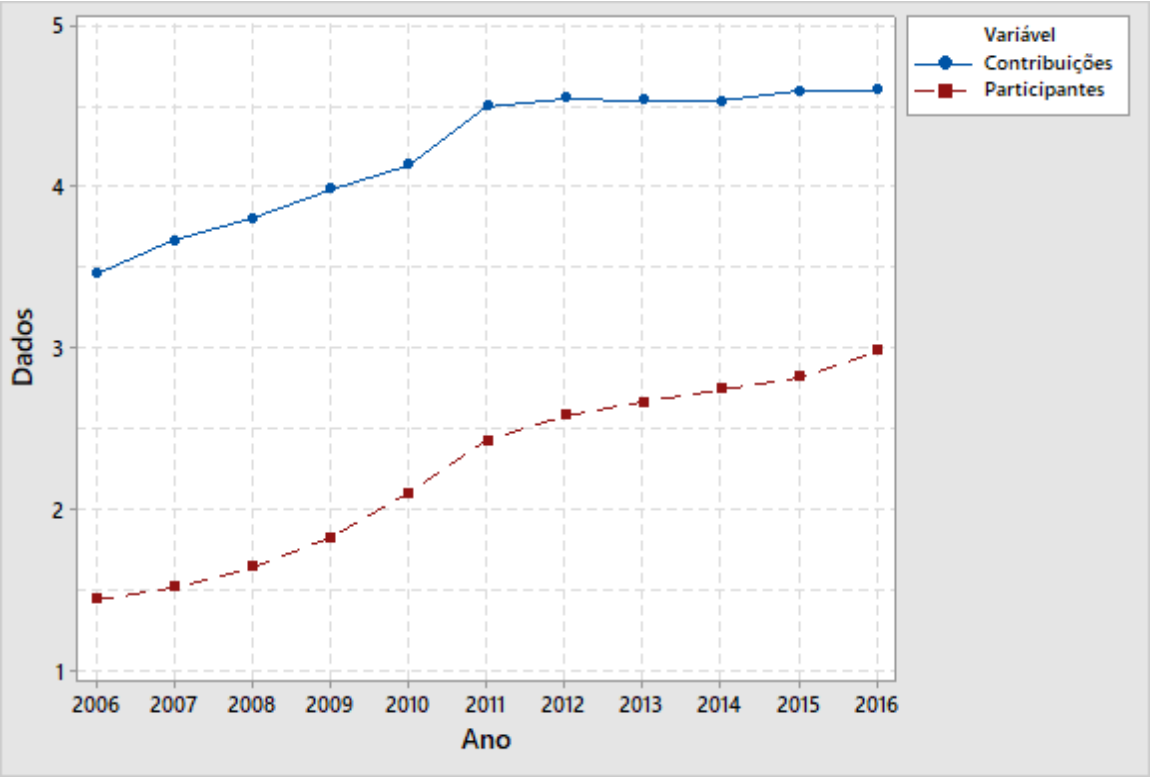


Figura A.10: Evolução da quantidade de participantes e contribuições relizadas durante os anos de 2006 e 2016.

Observando a Figura A.10, é possível notar que após 2011, a taxa de crescimento dos colaboradores foi sempre positiva mas que a quantidade dos *commits* se manteve estável. Isso acontece pois uma grande quantidade de colaboradores se envolvem no ecossistema para resolver pequenos problemas.

Mesmo que os novos colaboradores não contribuam com a mesma frequência que colaboradores já presentes no ecossistema, essa taxa de crescimento na quantidade de participantes deve ser vista de maneira positiva, visto que isto mantêm o ecossistema saudável e possibilita que os envolvidos utilizem estratégias para reter uma parte dos novos colaboradores.

Existe a possibilidade de refinar as avaliações, como por exemplo aumentando a granularidade com relação ao tempo avaliado (considerando o mesmo período entre os anos de 2006 e 2016). Desta forma, os dados foram fatiados em trimestres e semanas. A divisão por trimestres faz com que o período possa ser avaliado em 44 fatias de tempo diferentes, apresentadas na Figura A.11, onde é possível notar de maneira mais clara a redução da quantidade de contribuições que ocorreram nos anos de 2012 e 2013.

Aumentando a granularidade, foi gerado o gráfico de evolução realizando ava-

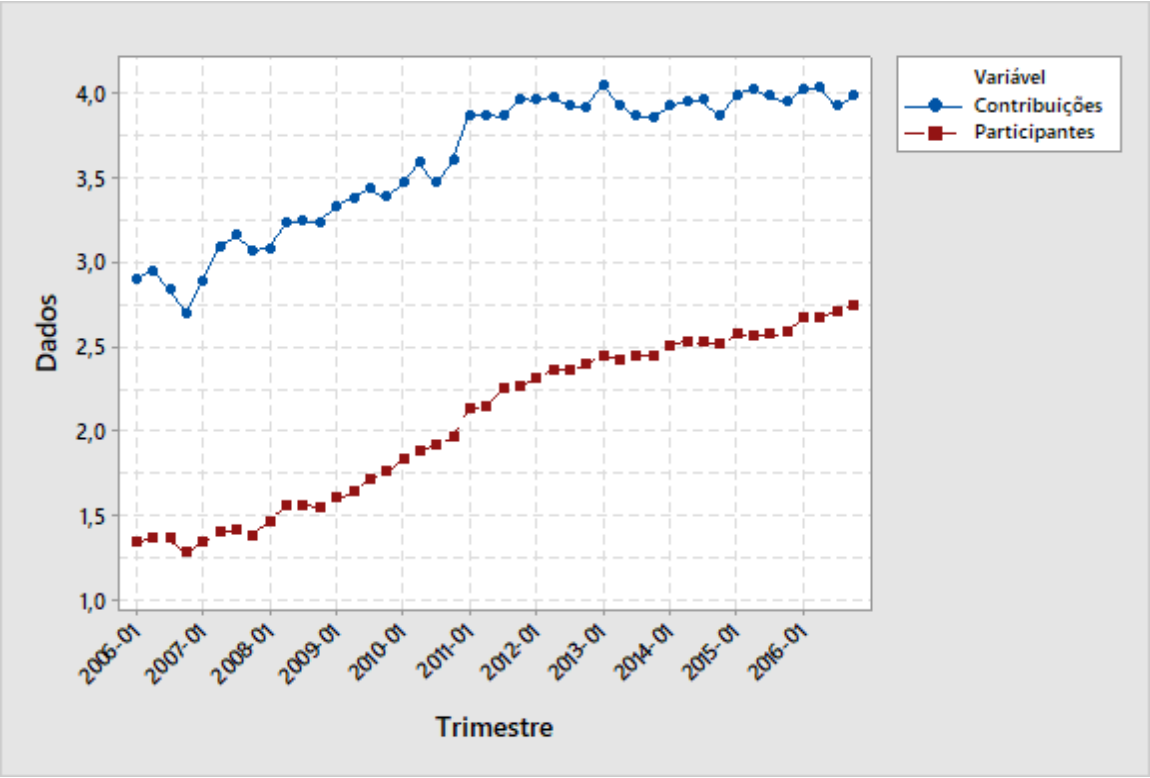


Figura A.11: Evolução da quantidade de participantes e contribuições relizadas durante os anos de 2006 e 2016, fatiados por trimestre.

liações semanais, apresentados na Figura A.12. Ao aumentar o nível de detalhe, algumas análises podem ser prejudicadas, como por exemplo sobre a quantidade de contribuições que diminuiu entre 2012 e 2013, entretanto diferentes análises se tornam possíveis. Observando o gráfico, é possível notar que próximo ao início de cada ano existe uma grande queda na quantidade de participantes e contribuições. Esta diminuição drástica representa o período da primeira e última semana do ano, dando indícios que grande parte dos participantes deixa de se dedicar às atividades, a fim de se envolver em atividades sociais referentes ao natal e comemorações de início do ano.

A rede está em constante evolução visto que alguns repositórios se tornam estáveis e necessitam de menos atenção quando comparados a novos projetos que surgem da necessidade da indústria e dos usuários. Figura A.13 fornece uma representação visual da rede de contribuição nos últimos 3 anos. Os repositórios são representados em vermelho e os colaboradores em verde. Como pode ser visto, os repositórios principais do ecossistema constantemente recebem contribuição e permanecem muito ativos durante todo o perído. Alguns projetos possuem múltiplos repositórios, por exemplo o Eclipse Titan[28], circulado

[28]https://projects.eclipse.org/projects/tools.titan

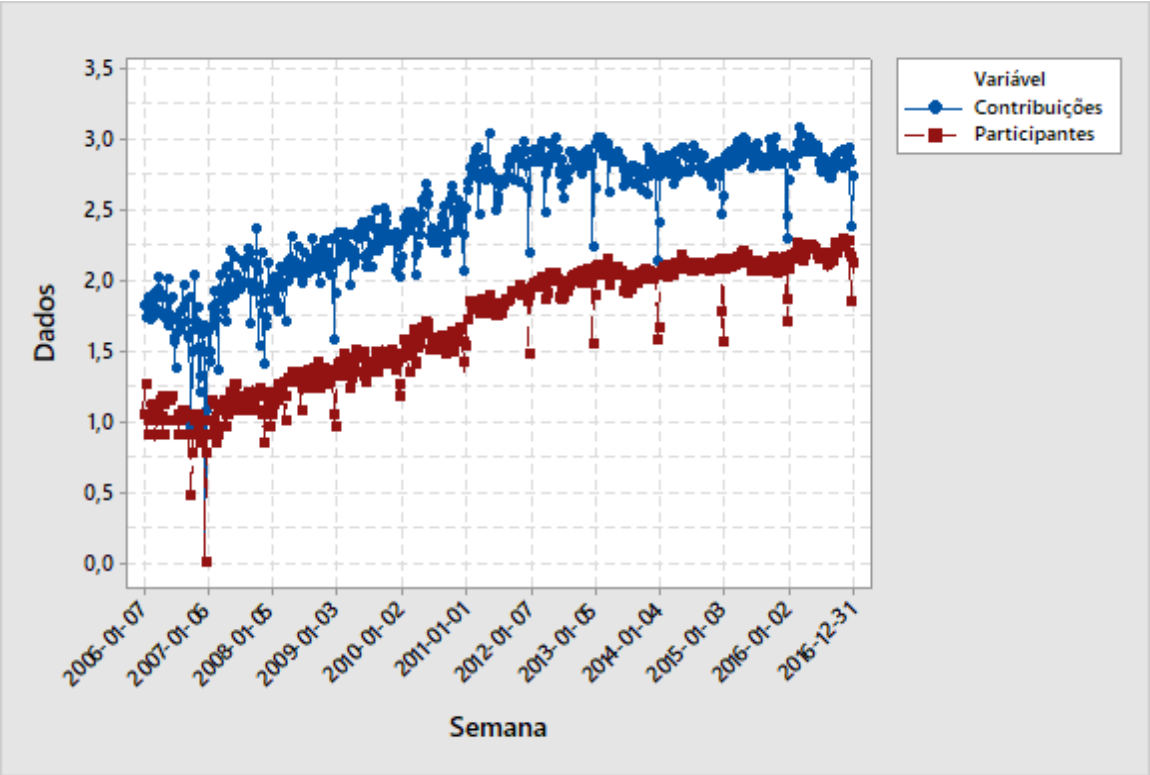


Figura A.12: Evolução da quantidade de participantes e contribuições relizadas durante os anos de 2006 e 2016, fatiados semanalmente.

na Figura A.13, uma ferramenta para auxílio em testes capaz de auxiliar no desenvolvimento de casos de teste, lógica de execução além de proporcionar um conjunto de testes executáveis para diferentes plataformas. O Titan consiste de um *core* baseado em um ambiente Unix/Linux e em plugins do Eclipse. O projeto iniciou em 2014 e teve seu primeiro release em 2015, contando com colaborações de indivíduos e por organizações, no caso do Titan a companhia Ericsson AB. Durante a coleta, o Titan possuía 61 repositórios e é apresentado na Figura A.13, localizando-se na parte esquerda da visualização da rede complexa, circulado em azul. Na seção seguinte da Figura A.13, referente ao ano de 2016, temos o projeto representado no mesmo local mas a quantidade de repositórios mostra-se menos representativa, visto que apenas 25 dos 61 repositórios tiveram contribuições neste ano. Isto pode ser explicado pelo fato de que repositórios periféricos se tornaram estáveis e só necessitarão de novas alterações caso o núcleo do projeto necessite. O ano de 2017 segue o mesmo comportamento de redução da quantidade de repositórios com atividade, reduzindo ainda mais sua representatividade na figura apresentada.

A outra rede complexa modelada durante a etapa de preparação foi a rede de comentários, que relaciona repositórios. A partir dos dados coletados e da estratégia de

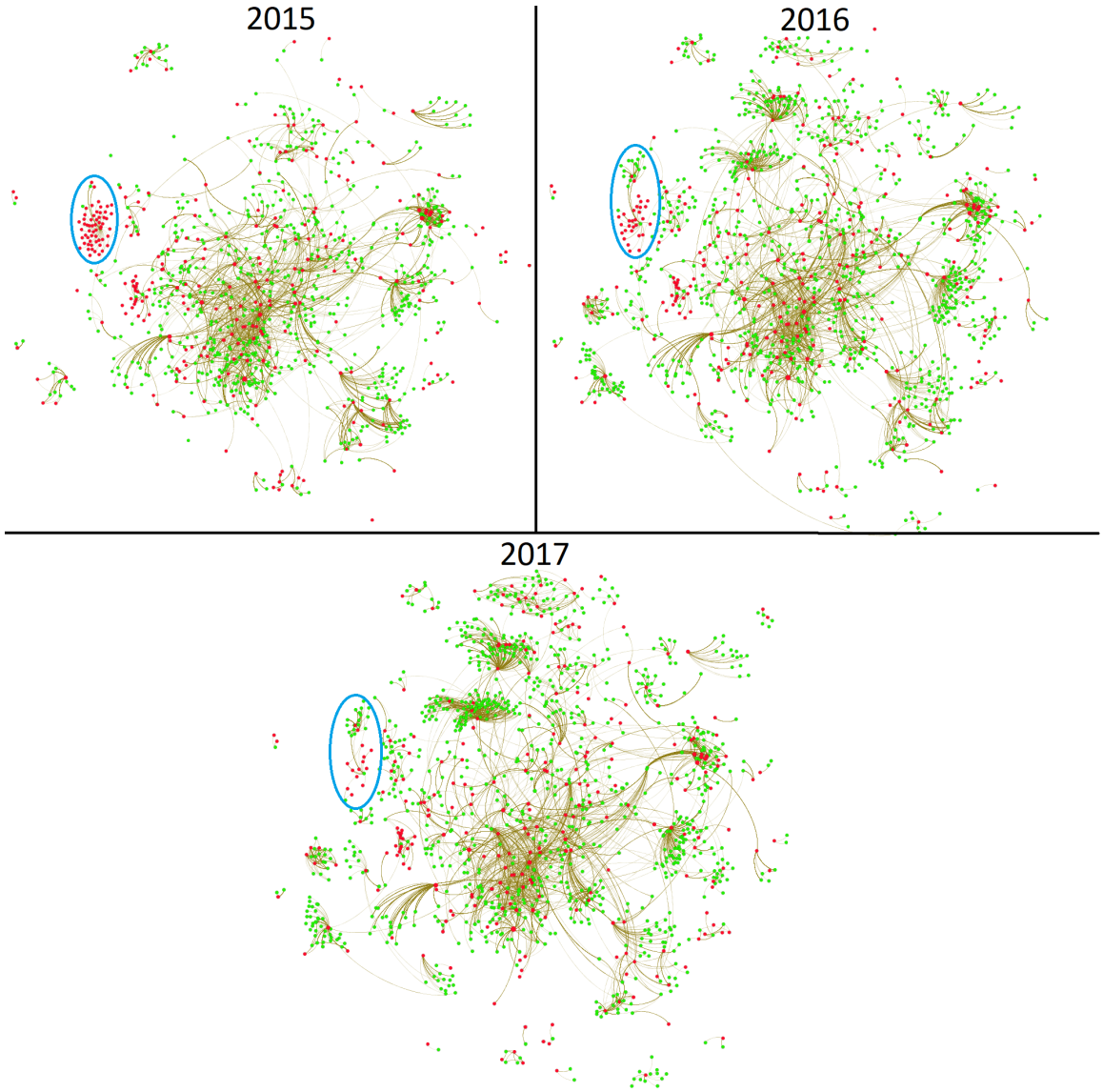


Figura A.13: Evolução da rede de contribuição entre os anos de 2015 e 2017.

visualização a rede foi instanciada e pode ser visualizada na Figura A.14. O peso dos relacionamentos entre usuários e repositórios é a quantidade de vezes em que o usuário realizou uma contribuição textual, representados graficamente pela espessura da aresta. Os nós também tiveram seu tamanho ajustado de acordo com seu grau e seguiram o mesmo padrão de cores definido na Figura A.9. Os nós e arestas tiveram um limite superior na visualização para impedir que outras partes da rede fossem ocultadas.

Na parte inferior da Figura A.14, destacada com uma seta "A", é possível notar um denso grupo de colaboradores que pertencem a um componente conexo da rede. Este grupo representa componentes do Eclipse OMR[29], uma plataforma para auxílio na

[29]https://www.eclipse.org/omr

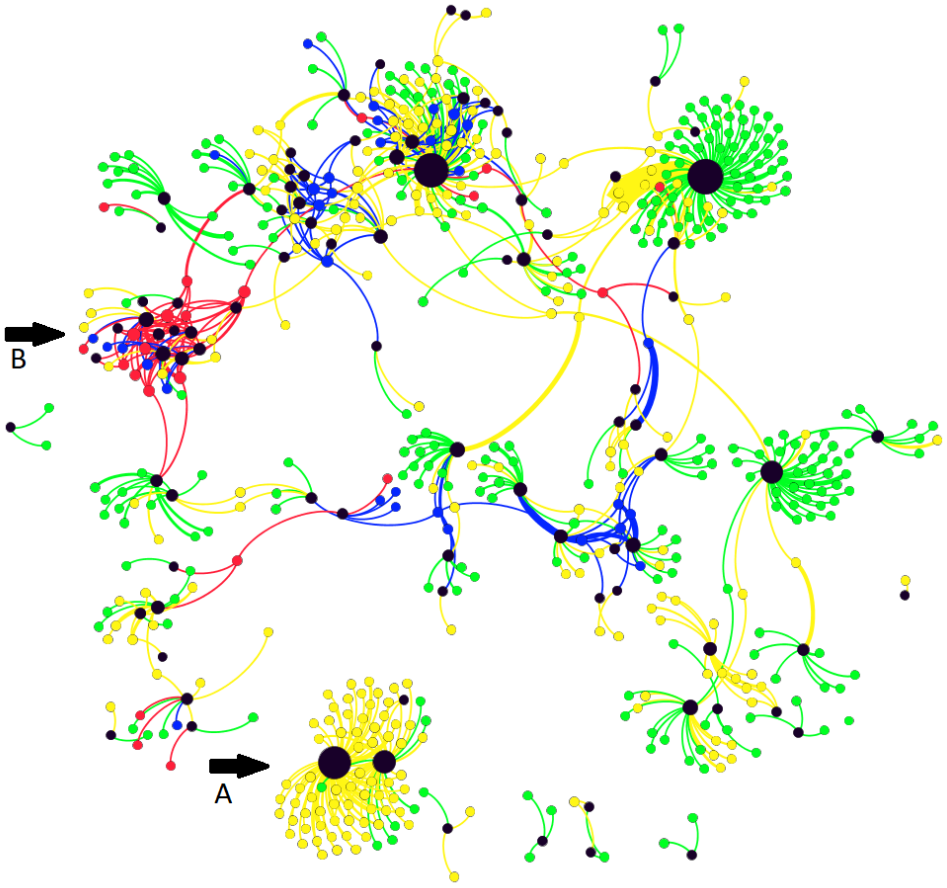


Figura A.14: Rede de comentários, representando os usuários classificados por grupos.

construção de linguagens de execução robustas que podem ser executadas em diversas plataformas de hardware e sistemas operacionais. No canto esquerdo da Figura A.14, destacado com uma seta "B", existem alguns repositórios pertencentes ao projeto Xtext[30], um *framework* para desenvolvimento de linguagens de programação e linguagens de domínio específico. Na mesma região, também são apresentados os usuários do projeto elk[31], responsáveis por auxiliar na criação de diagramas. O relacionamento entre os participantes destes dois projetos pode ser explicado devido ao alinhamento de objetivos entre os dois projetos, visto que o projeto elk visa auxiliar na criação de novas linguagens promovidas pelo Xtext. Isso representa um indício que os usuários tendem a participar em projetos que são relacionados entre si, e que isto pode estar relacionado aos motivos de engajamento dos usuários nos projetos.

A última das redes avaliadas foi a rede social construída a partir dos dados coletados e do relacionamento criado a partir da Equação A.3, que relaciona os participantes a

[30]https://www.eclipse.org/Xtext
[31]https://www.eclipse.org/elk/

partir das suas contribuições nos projetos, considerando aspectos temporais. A rede teve estratégia de apresentação similar às apresentadas anteriormente. A rede social extraída possui 14 componentes conexas, onde a maior componente possui a maior parte dos participantes (95,39%), e todos os outros componentes possuem menos de 30 participantes. Os participantes que não se relacionaram a nenhum outro participante foram removidos durante a construção da rede, e, por fim, a rede possui 2196 nós e 90741 relacionamentos entre eles.

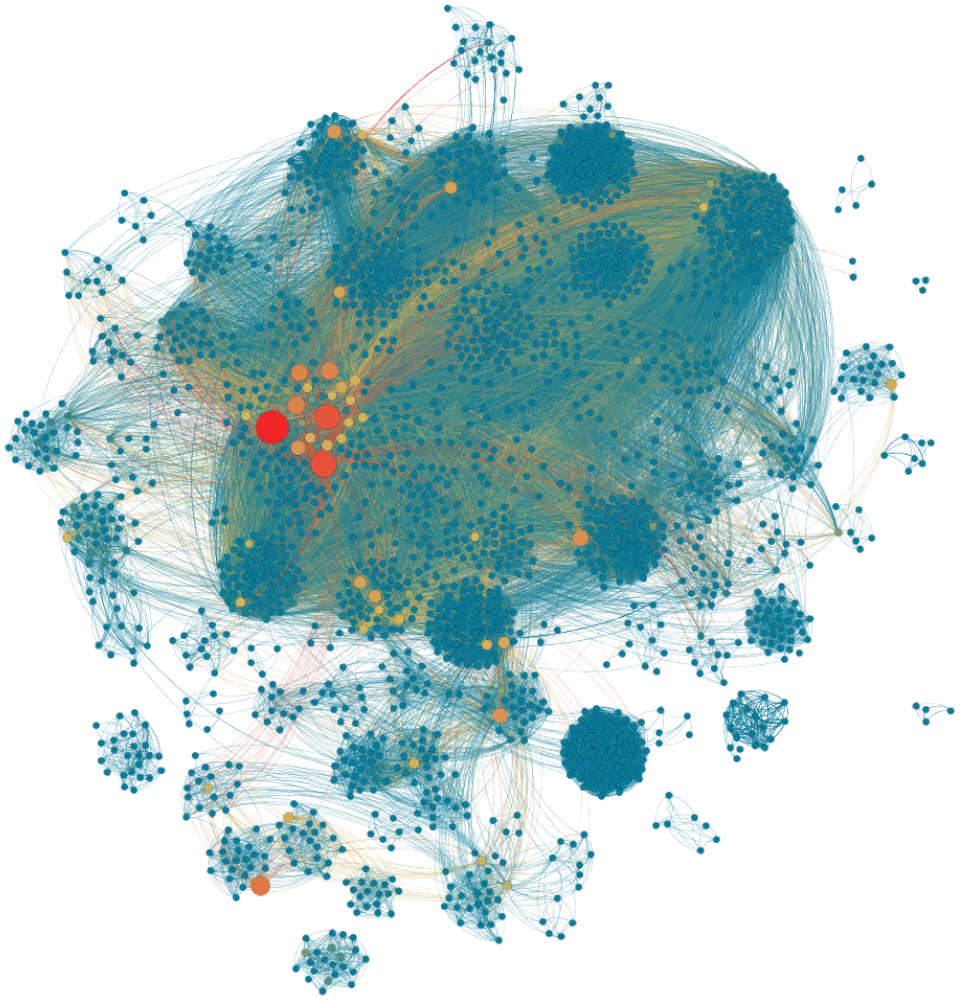


Figura A.15: Rede social construída a partir do relacionamento calculado a partir das contribuições de código dos colaboradores do SECO.

Assim como nas outras visualizações, os nós tiveram seu tamanho ajustado, mas nesta rede o critério foi a centralidade de *betweenness*. Este também foi o critério para ajuste das cores dos nós, onde cores mais quentes indicam valores maiores e as cores mais frias indicam nós com baixa centralidade de *betweenness*. Como pode ser visto na Figura B.2, os nós com grande centralidade estão conectados a um grande conjunto da

rede. Estes participantes são capazes de estabelecer relacionamentos entre comunidades densamente conectadas, e prover uma opção para que exista troca de informação entre as comunidades.

Também é possível notar participantes que podem ser tomados como referência em um determinado grupo, visto que é possível destacar estes participantes tendo seu tamanho maior e cor quente. Estes usuários normalmente estão relacionados a grande parte dos usuários na comunidade e se conectam em outras comunidades.

A visualização da Figura B.2 também mostra que existem seções de nós altamente conectados entre si, formando grupos de colaboração que possuem poucos relacionamentos com participantes de outros grupos. Neste caso, quando um usuário necessita de ajuda em algum item, o auxílio provavelmente será fornecido por um participante da mesma comunidade.

Por outro lado, alguns colaboradores estão conectados a uma grande quantidade de usuários e em diferentes *clusters*. Este tipo de usuário pode ser visto de forma global na rede a partir da sua cor. Quando o ecossistema necessitar iniciar um novo projeto ou aproximar diferentes comunidades, estes serão os usuários mais indicados para auxiliar pela sua experiência e maneira de colaborar em diferentes projetos do ecossistema.

Coleta das métricas do GQM

O próximo passo deste estudo consiste nos cálculos das métricas previstas para as questões de pesquisa definidas no planejamento. Através dos cálculos é possível embasar as respostas das respectivas questões de pesquisa utilizando critérios matemáticos.

A primeira métrica é referente à quantidade de atividades desenvolvidas durante a execução do processo. O processo apresentado possui 12 atividades a serem realizadas, entretanto, durante o estudo preliminar, algumas atividades foram inseridas para refinar e fazer com que o processo pudesse ser reproduzido. Desta forma, além das 12 atividades, foram realizadas outras 5 atividades que não foram contempladas na concepção inicial do processo. Estas atividades seguem listadas abaixo, assim como os motivos que justificam a adição de tais atividades no processo.

- ***Seleção das fontes de dados:*** durante a primeira atividade do estudo, que com-

preende a identificação das possíveis fontes de dados, diferentes fontes são encontradas, cada uma com suas características, sejam elas de disponibilidade de dados, facilidade no acesso, alinhamento com os objetivos da pesquisa, dentre outros. Diante disso, foi necessário criar esta nova atividade para selecionar quais as fontes de dados seriam as mais adequadas.

- ***Integração de diferentes fontes de dados:*** as etapas de identificação e seleção de dados podem retornar múltiplas fontes, cada uma com suas peculiaridades. Levando isso em conta, uma atividade que visa integrar as diferentes fontes de dados com o objetivo de fornecer um repositório central para análise fez-se necessário, para diminuir os esforços nas atividades de construção do ambiente e de análise dos dados.

- ***Identificação dos elementos de análise:*** durante a avaliação das fontes selecionadas, existe a possibilidade de existirem elementos que não são relevantes ao escopo da pesquisa. Uma fonte de dados pode armazenar dados referentes às pessoas envolvidas, projetos, artefatos e outros, mas dependendo da análise a ser realizada alguns destes elementos podem não ser necessários. Desta forma, uma atividade para identificar os elementos que irão compor a análise se fez necessária para reduzir os esforços de coleta e análise.

- ***Aplicação de filtros:*** a partir dos dados coletados, alguns deles podem fugir do escopo da análise, contendo elementos que não serão utilizados posteriormente. Desta forma, alguns filtros podem ser úteis, como, por exemplo, a delimitação de um período de observação ou de entidades a serem ignoradas em uma análise específica.

- ***Definição de características temporais:*** durante a execução do estudo, notou-se que as questões temporais não estavam descritas no processo, desta forma foram adicionadas atividades deixando explícita a atividade de análise temporal, delimitando tanto o período como a forma de divisão do tempo.

Diante disso, o resultado da **M1** foi de 5 atividades desenvolvidas durante a execução do processo que não estavam explícitas na proposta original.

Com o auxílio dos especialistas durante e após a execução do processo, também foram identificados pontos para melhorias. Ficou claro que o processo podia atender às necessidades dos envolvidos, mas muitas atividades não foram detalhadas o suficiente.

Desta forma, a atividade de extração de dados e análise foram transformadas em subprocessos. Além disso, a análise das redes complexas também foi transformada em um subprocesso. Desta forma, ao final da revisão do processo pelos especialistas e da execução do processo, temos 39 atividades, assim como algumas decisões que visam ajustar a forma de ataque às questões planejadas pelos especialistas, mais que triplicando as 12 atividades inciais. Vale destacar que grande parte dessas atividades visa deixar o processo mais claro, como, por exemplo, a atividade de análise de redes complexas que se transformou em 7 atividades. Diante disso, o resultado da **M2**, que soma a quantidade de atividades identificadas na revisão do processo por especialistas, foi de 27, que representa a diferença entre as atividades do processo inicial para o processo melhorado.

Após a avaliação das métricas referentes à **Q1**, que visa verificar se o processo possui todas as informações capazes de gerar conhecimento explícito, foi verificado que o processo auxiliou na obtenção de conhecimento, mas alguns detalhamentos são necessários.

Para responder a **Q2**, que tem como objetivo verificar se o uso de redes complexas pode auxiliar os *stakeholders* na identificação dos desenvolvedores que mais contribuíram sob diferentes aspectos no desenvolvimento de software, foram analisadas as respectivas métricas. Como resultado da **M3**, tivemos três diferentes redes modeladas, sendo elas uma rede de contribuições dos participantes em repositórios; uma rede relacionando usuários e repositórios através de comentários e uma rede social, construída a partir dos dados das contribuições dos participantes.

A **M4**, que representa a quantidade de desenvolvedores identificados que mais contribuem, também foi calculada considerando a centralidade de *betweenness* na rede social gerada. A centralidade de *betweenness* representa a quantidade de caminhos mínimos que passam pelo nó, quanto maior a centralidade, maior a quantidade dos caminhos mínimos que passam pelo nó. É interessante observar os nós que possuem esta centralidade alta, pois eles possuem destaque na rede, visto que a informação tende a passar mais vezes pelo nó. Isso explicita uma dependência da comunidade nesses usuários para

que elas continuem trocando informações.

Dos 2192 colaboradores, 1558 possuíam o valor de centralidade igual a zero, indicando que eles não fazem parte de nenhum caminho mínimo entre dois nós na rede gerada. Desta forma, foram avaliados os 634 participantes que fazem parte de algum caminho mínimo. Os valores de centralidade foram normalizados e apresentados na Figura A.16, onde é possível ver que existe uma grande cauda de participantes com níveis de centralidade baixa.

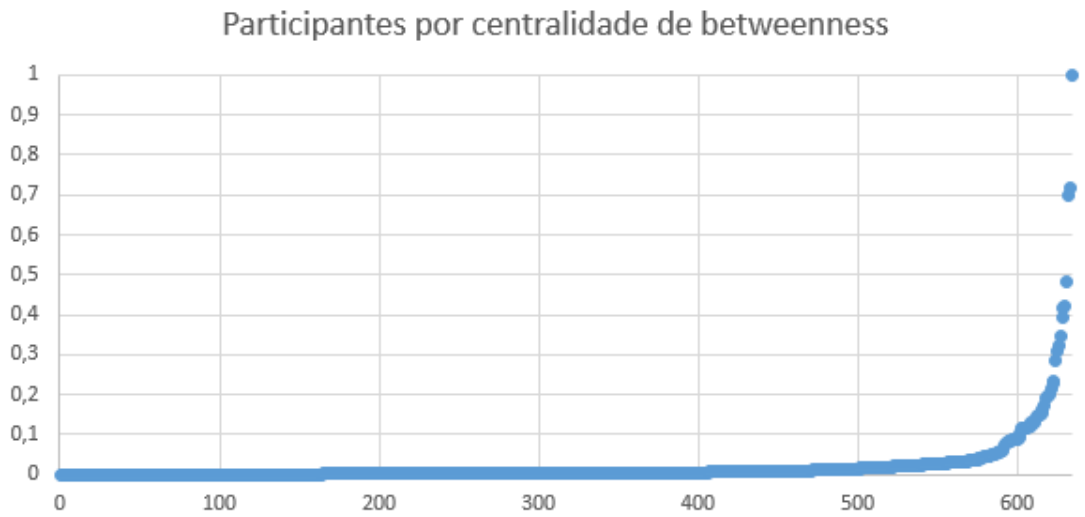


Figura A.16: Participantes de acordo com a centralidade de *betweenness* normalizada.

Para identificar quais são os desenvolvedores que mais contribuíram foi utilizada a centralidade de *betweenness*. Os desenvolvedores que estão um desvio padrão acima da média foram considerados como os mais importantes. De acordo com os dados coletados, temos que a média é de 2,657 e o desvio padrão é de 1,0971, ambos os valores estão em escala logarítmica de base 10. No total, 42 desenvolvedores estão pelo menos 1 desvio padrão acima da média, e 20 desenvolvedores estão 2 desvios padrões acima da média.

Desta forma, como resultado da **M4** temos 42 desenvolvedores que se destacaram com relação aos 2192 presentes na rede social gerada a partir dos dados de contribuições.

As métricas **M5**, **M6** e **M7** foram definidas para auxiliar na captura da evolução do ecossistema. As medições foram realizadas considerando os anos e avaliadas desta forma e por isso o valor total das métricas é dado pela soma das quantidades separadas por ano.

Assim sendo, a métrica **M5** foi calculada considerando as contribuições dos par-

Tabela A.1: Total de contribuições realizadas por ano.

Ano	**Contribuições**
2006	2877
2007	4625
2008	6370
2009	9709
2010	13726
2011	31483
2012	35306
2013	34256
2014	33875
2015	38849
2016	39335
Total	**250411**

ticipantes do SECO entre os anos de 2006 e 2016. Os valores são apresentados na Tabela A.1, referente ao gráfico da Figura A.10 onde temos como resultado o total de 250.411. As métricas **M6** e **M7** não foram calculadas visto que, durante a análise, tais dados não foram divididos temporalmente.

A partir da execução do GQM, temos que as questões **Q1** e **Q2** foram respondidas com resultados positivos, e a **Q3** foi parcialmente respondida visto que duas das três métricas estabelecidas não conseguiram ser medidas. Entretanto, entendeu-se que o objetivo foi atendido visto que a resposta da **Q3** não anula os resultados obtidos nas duas outras questões. Assim, temos indícios que o processo atende ao objetivo de avaliar o ecossistema a partir da sua dimensão social através da modelagem e análise de redes complexas para identificação dos colaboradores de maior importância.

A impossibilidade de responder a **Q3** e as observações dos especialistas durante a execução do estudo preliminar apontam para a necessidade da melhoria do processo, que será apresentada no próximo capítulo. Além da avaliação positiva os especialistas indicaram pontos para estender e melhorar as análises, que também foram considerados durante a atividade de melhoria do processo.

B Apêndice

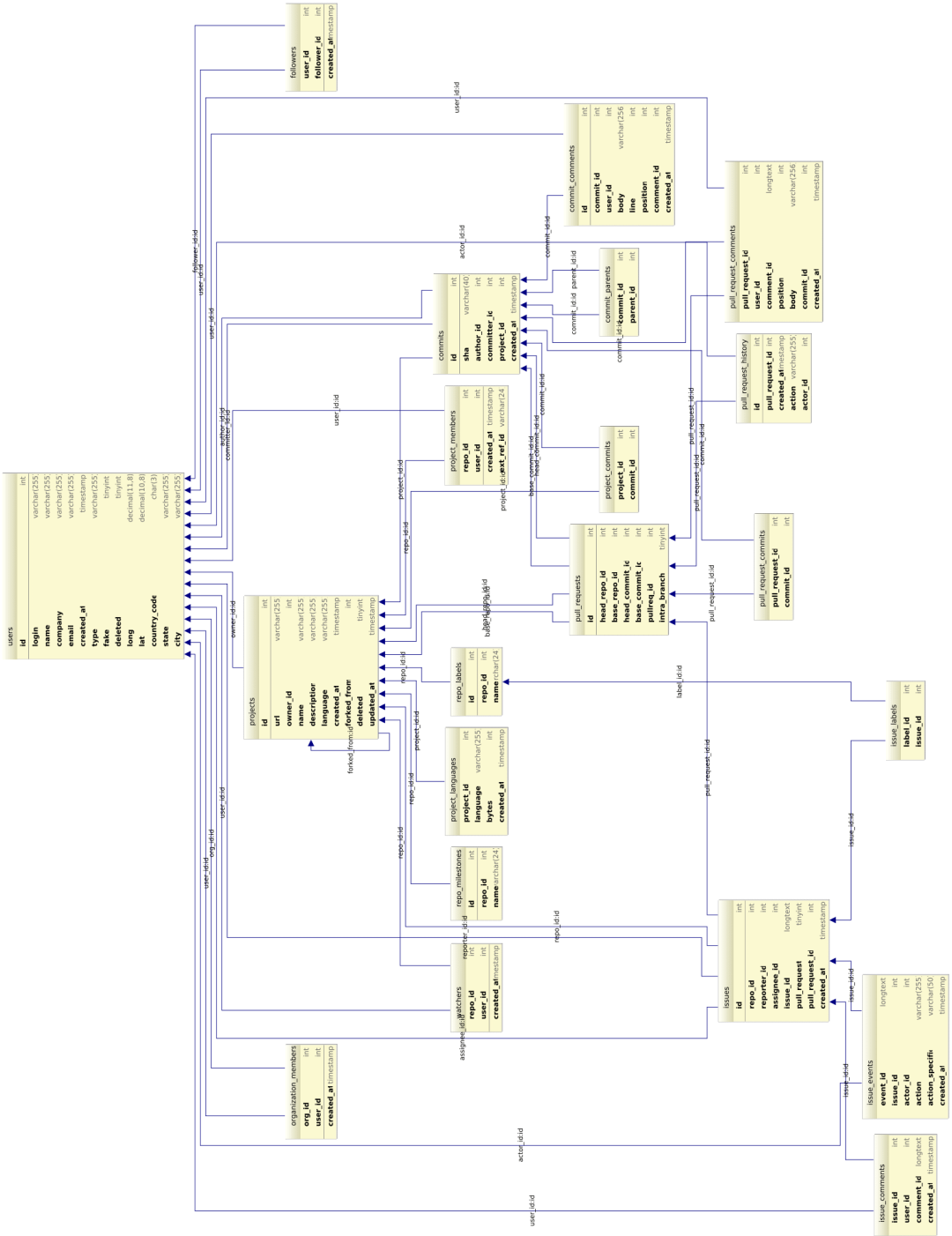


Figura B.1: Esquema relacional da base GHTorrent.

Tabela B.1: Quantidade de *commits* e de repositórios distintos, presentes no *dataset* de avaliação do sSECO-Process estendido, agrupados de acordo com o ano.

Ano	Total de Commits	Repositórios com atividade
2004	20	1
2005	169	6
2006	84	2
2007	221	6
2008	361	13
2009	3135	111
2010	15918	476
2011	33503	1411
2012	46254	3229
2013	75183	5254
2014	80073	5188
2015	40934	3578
2016	33766	2801
2017	20228	1711

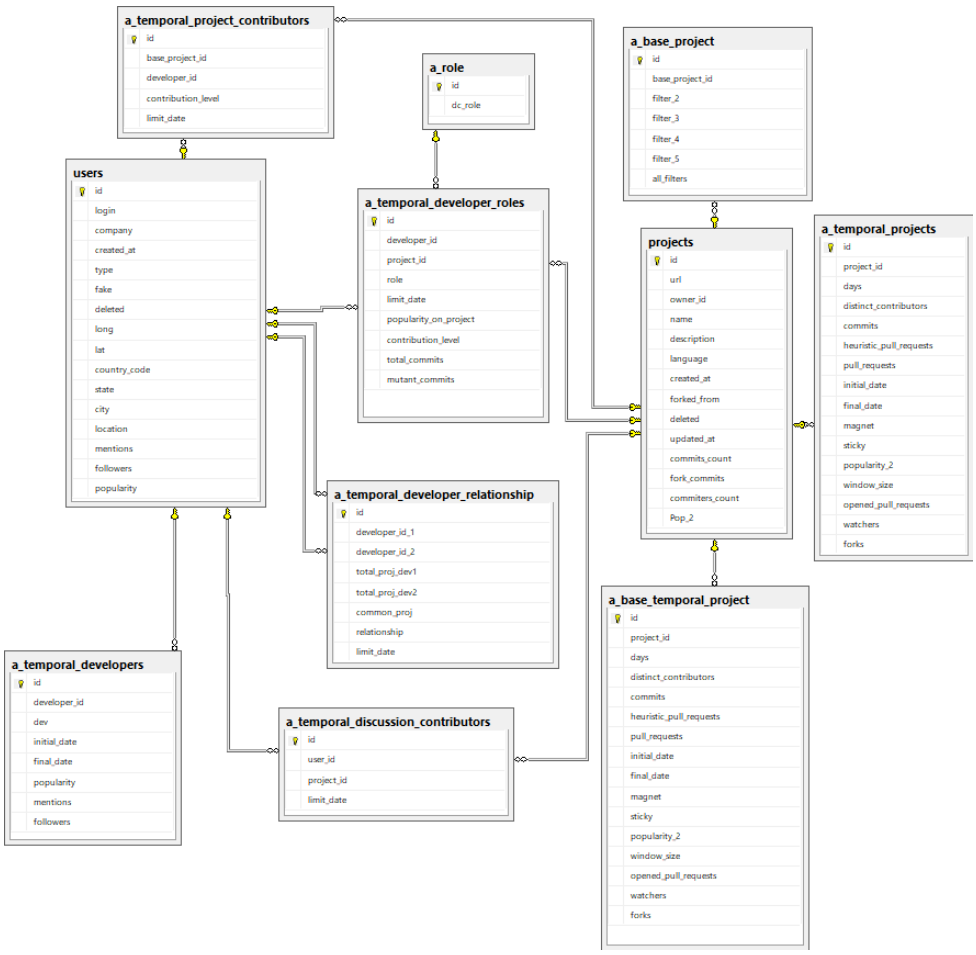


Figura B.2: Esquema relacional das principais estruturas do sSECO para apoio da análise da dimensão social em SECOs através do tempo.