

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO**

**ISABELA NEVES FERRAZ**

**GOVERNANÇA DE COMUNIDADES VIRTUAIS: ENTENDENDO A  
TRANSFORMAÇÃO DA ESTRUTURA E DOS PROCESSOS DE CONTROLE  
EM PROJETOS DE *SOFTWARE* LIVRE**

**BRASÍLIA 6 DF**  
**2019**

**ISABELA NEVES FERRAZ**

**GOVERNANÇA DE COMUNIDADES VIRTUAIS: ENTENDENDO A  
TRANSFORMAÇÃO DA ESTRUTURA E DOS PROCESSOS DE CONTROLE EM  
PROJETOS DE *SOFTWARE* LIVRE**

Tese apresentada ao Programa de Pós-graduação em Administração da Universidade de Brasília, como requisito à obtenção do grau de Doutora em Administração.

**Orientador:** Prof. Carlos Denner dos Santos  
Júnior, Dr.

**BRASÍLIA 6 DF**

**2019**

**ISABELA NEVES FERRAZ**

**GOVERNANÇA DE COMUNIDADES VIRTUAIS: ENTENDENDO A  
TRANSFORMAÇÃO DA ESTRUTURA E DOS PROCESSOS DE CONTROLE EM  
PROJETOS DE *SOFTWARE* LIVRE**

Tese apresentada ao Programa de Pós-Graduação em  
Administração da Universidade de Brasília, como  
requisito à obtenção do grau de Doutora em  
Administração.

Data de aprovação: \_\_\_\_\_ de \_\_\_\_\_ de 2019.

Comissão examinadora:

---

**Doutor, Carlos Denner dos Santos Júnior** ó Professor Orientador  
Universidade de Brasília

---

**Doutor, Alex Sandro Gomes** ó Membro Externo  
Universidade Federal de Pernambuco

---

**Doutor, Igor Fabio Steinmacher** ó Membro Externo  
Northern Arizona University

---

**Doutor, Paulo Henrique de Souza Bermejo** ó Membro Interno  
Universidade de Brasília

---

**Doutora, Gisela Demo** ó Suplente  
Universidade de Brasília

## AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que ajudaram a tornar este trabalho de tese uma realidade.

Ao professor Dr. Carlos Denner, meu orientador, sou profundamente grata pela paciência, palavras de incentivo e ensinamentos valiosos que possibilitaram o meu desenvolvimento como pesquisadora e me conduziram ao alcance da meta que sempre almejei.

À banca, formada pelos professores Dr. Alex Sandro Gomes, Dr. Igor Steinmacher e Dr. Paulo Bermejo, pela atenção nos momentos em que precisei e importantes contribuições para a elaboração desta tese.

Aos profissionais que tive a honra de entrevistar na pesquisa de campo, especialmente no estudo dos casos CoGrOO, Noosfero e Kytos, a quem atribuo a essência dos resultados deste trabalho, agradeço pela disponibilidade de tempo e acolhimento durante todo o processo de coleta dos dados.

Aos docentes do PPGA, pela oportunidade de crescimento acadêmico, e aos demais servidores do Programa, em particular Edvânia e Selma, pelo apoio institucional concedido ao longo desses quatro anos.

Aos membros do grupo de pesquisa Socie-dados, pelas sugestões que permitiram o aprimoramento do projeto de tese.

Aos colegas da DPR Ana Caroline, Jossane, Luciana Couto, Luciana Xavier, Thaís e Newton, pela compreensão e apoio nesse período que precisei conciliar o trabalho e o doutorado.

À Ana Carolina Rezende, Cecília Cesar, Cristiana Amaral, Everton Verga, Junia Falqueto, Luiz Fernando, Nathália Melo, Silvia Onoyama e Sueli Menelau, pessoas especiais que tive a chance de conhecer melhor no PPGA, pelas constantes demonstrações de encorajamento e solidariedade no compartilhamento de conhecimentos.

Aos amigos da vida, com ênfase aos que compõem os grupos "Grupo Temporário", "Só os três..." e "Amigas Atual" (eles entenderão), por serem os meus melhores incentivadores, mesmo nos momentos mais difíceis.

Aos professores Dr. Jairo Dornelas e Dr. Jorge Correia, pessoas que me acompanharam desde a graduação na UFPE e que se tornaram minhas referências acadêmicas, por terem me encorajado e compartilhado comigo as minhas preocupações.

A toda a minha família, sobretudo, aos meus pais, Maria de Lourdes e Roberto, aos meus irmãos, Maíra e Thiago, ao meu marido, Henrique, aos meus sogros, Henrique e Márcia, e à Mércia, além de vários outros familiares, pelo amor irrestrito e torcida que nunca faltaram.

A Henrique e Roberto, novamente, deixo o meu agradecimento, não só pela inteligência, mas também pela espontânea dedicação a mim, saibam, ambos, que vocês foram a base a qual tanto precisei para trilhar esta jornada.

*Minha filosofia, na sua essência, é o conceito do Homem como um ser heróico, tendo a felicidade como o propósito moral da sua vida, a conquista produtiva como sua mais nobre atividade, e a razão como seu único referencial.*

Ayn Rand

## RESUMO

A finalidade da presente pesquisa é entender a governança e as suas transformações, a partir da consideração da estrutura e dos processos de controle identificados em projetos de *software* livre comunitários. A tese central é que a governança se torna mais formal em função da trajetória de desenvolvimento vivenciada por esses projetos. A investigação apresentou dois momentos bem definidos: o primeiro, dedicado ao ensaio teórico, teve o propósito de abordar a governança dos projetos colaborativos comunitários, com foco na literatura de *software* livre, comparando esses projetos com as organizações tradicionais; o segundo, voltado à realização da pesquisa empírica, quando três casos nacionais de projetos de *software* livre, organizados no formato de comunidades virtuais, foram selecionados, a partir de critérios, para o estudo da governança e das suas transformações no decorrer das trajetórias de desenvolvimento desses projetos. Vale ressaltar que a pesquisa empírica fez uso da abordagem qualitativa, com emprego da estratégia de estudo de casos múltiplos, cuja coleta de dados se baseou em entrevistas e exame documental, e a fase analítica na técnica de análise de conteúdo. Os resultados obtidos por meio do ensaio teórico e da pesquisa empírica permitiram o levantamento de algumas evidências e explicações sobre o tema. Cada caso seguiu por caminhos distintos, não se verificando, nesse percurso, o desenvolvimento dos projetos em fases evolutivas, mas momentos de maior e de menor atividade aos quais a governança demonstrou se adaptar. No decorrer de cada trajetória, procedimentos de gestão formais foram implementados e conviveram com os aspectos informais inerentes ao formato comunitário. Tal formalização, no entanto, aconteceu principalmente em virtude das relações que os projetos estabeleceram com atores organizacionais diversos, na busca dos recursos necessários à sua sobrevivência. O ensaio teórico e a pesquisa empírica contribuíram no sentido de revelar que diferentemente dos estudos que buscam associar os projetos de *software* livre comunitários a agrupamentos não hierárquicos e informais, esses coletivos podem formalizar as suas práticas, ainda que traços comunitários informais sejam mantidos. Numa perspectiva gerencial, os resultados identificaram que os responsáveis pelos projetos precisaram recorrer ao ambiente para conseguir junto a outras organizações (patrocinadores, fundações, parceiros, órgãos de fomento etc) os meios necessários para se manterem funcionando no longo prazo. Para o alcance desses resultados foi fundamental que o estudo da governança dos projetos de *software* livre comunitários, apresentado nesta tese, considerasse não só as comunidades, mas toda a história e os contextos nos quais esses projetos estiveram envolvidos.

**Palavras-chave:** Governança. Projetos de *software* livre. Comunidades virtuais. Transformações. Desenvolvimento.

## ABSTRACT

The purpose of this research is to understand governance and its transformations from the structure and control processes identified in community free software projects. The central thesis is that governance becomes more formal due to the development trajectory experienced by these projects. The research presented two well-defined stages: the first one, dedicated to the theoretical essay, aimed at addressing the governance of community collaborative projects by focusing on free software literature and comparing these projects with traditional organizations; the second stage focused on empirical research, when three national cases of free software projects organized in virtual communities were selected based on criteria for the study of governance and its transformations in the course of developmental trajectories of these projects. It is worth mentioning that the empirical research was a qualitative approach, by using the multiple case study strategy, whose data collection was based on interviews and documentary examination and the analytical phase being based on the technique of content analysis. The results through the theoretical essay and the empirical research allowed the collection of some evidence and explanations on the subject. Each case followed different patterns, thus not being verified the development of projects in evolutionary phases but moments of greater and lesser activity to which governance has shown to adapt. During each trajectory, formal management procedures were implemented and coexisted with the informal aspects inherent to the community format. However, such formalization happened mainly because of the relationships projects have established with diverse organizational actors in the search of the necessary resources for their survival. The theoretical essay and the empirical research have contributed to revealing that unlike the studies seeking to associate the projects of community free software to non-hierarchical and informal groupings, these collectives can formalize their practices, even if informal community traits are maintained. From a managerial perspective, the results identified that those responsible for the projects needed to use the environment in order to get the necessary resources from other organizations (sponsors, foundations, partners, funding agencies, etc.) to keep them working in the long term. In order to reach these results, it was fundamental that the study on the governance of community free software projects addressed in this thesis have considered not only the communities but all the history and contexts in which these projects were involved.

**Keywords:** Governance. Free software projects. Virtual communities. Transformations. Development.



## LISTA DE FIGURAS

<b>Figura 1</b> ó Diagrama geral de elaboração do trabalho de tese.....	13
<b>Figura 2</b> ó Transformações da governança no desenvolvimento dos projetos colaborativos comunitários .....	29
<b>Figura 3</b> ó Diagrama conceitual da tese .....	66
<b>Figura 4</b> ó Desenho metodológico da pesquisa .....	83

## LISTA DE QUADROS

<b>Quadro 1</b> ó Comparação entre organizações tradicionais e projetos colaborativos comunitários .....	27
<b>Quadro 2</b> ó Estruturas tradicionais <i>versus</i> estruturas inovadoras.....	45
<b>Quadro 3</b> ó Critérios de seleção dos casos para estudo .....	71
<b>Quadro 4</b> ó Características gerais das entrevistas .....	76
<b>Quadro 5</b> ó Categorias pré-definidas para a pesquisa a partir da literatura .....	80
<b>Quadro 6</b> ó Identificação e perfil dos entrevistados do projeto CoGrOO / OpenNLP .....	85
<b>Quadro 7</b> ó Papéis identificados nos projetos de software livre comunitários da Apache .....	96
<b>Quadro 8</b> ó Identificação e perfil dos entrevistados do projeto Noosfero .....	108
<b>Quadro 9</b> ó Identificação e perfil dos entrevistados do projeto Kytos .....	126
<b>Quadro 10</b> ó Aspectos formais e informais da estrutura organizacional e decisória.....	159
<b>Quadro 11</b> ó Processos de controle formais e informais .....	162
<b>Quadro 12</b> ó Principais transformações da governança.....	165
<b>Quadro 13</b> ó Aproximação dos casos ao aporte conceitual e teórico da tese.....	166

## LISTA DE SIGLAS

<b>ASL</b>	Associação de Software Livre Brasil
<b>CCSL</b>	Centro de Competência em Software Livre
<b>CDT</b>	Centro de Informática
<b>Epusp</b>	Escola Politécnica da Universidade de São Paulo
<b>Febes</b>	Fórum Brasileiro de Economia Solidária
<b>Finep</b>	Financiadora de Estudos e Projetos
<b>IME</b>	Instituto de Matemática e Estatística
<b>Lappis</b>	Laboratório Avançado de Produção, Pesquisa e Inovação em Software Livre
<b>MCTI</b>	Ministério da Ciência, Tecnologia e Inovação
<b>MIT</b>	Massachusetts Institute of Technology
<b>MP</b>	Ministério do Planejamento, Desenvolvimento e Gestão
<b>NCC</b>	Núcleo de Computação Científica
<b>OT</b>	Organização Temporária
<b>P&amp;D</b>	Pesquisa e Desenvolvimento
<b>QualiPSO</b>	Quality Platform for Open Source Software
<b>RM</b>	Release Manager
<b>Senac</b>	Serviço Nacional de Aprendizagem Comercial
<b>SLTI</b>	Secretaria de Logística e Tecnologia da Informação
<b>SPRACE</b>	São Paulo Research and Analysis Center
<b>TIC</b>	Tecnologia da Informação e Comunicação
<b>UnB</b>	Universidade de Brasília
<b>Unesp</b>	Universidade Estadual Paulista
<b>USP</b>	Universidade de São Paulo

## SUMÁRIO

<b>1 CONSIDERAÇÕES INICIAIS .....</b>	<b>10</b>
1.1 Objetivo geral da tese.....	12
1.2 Objetivos específicos da tese.....	12
<b>2 ENSAIO TEÓRICO.....</b>	<b>14</b>
<b>Resumo .....</b>	<b>14</b>
2.1 Introdução.....	14
2.2 Projetos colaborativos organizados em comunidades.....	17
2.3 Governança em projetos colaborativos organizados em comunidades.....	19
2.3.1 Mecanismos estruturais .....	20
2.3.2 Mecanismos processuais.....	23
2.4 Principais reflexões sobre os projetos colaborativos organizados em comunidades.....	27
2.5 Conclusões.....	30
<b>3 PESQUISA EMPÍRICA .....</b>	<b>33</b>
<b>Resumo .....</b>	<b>33</b>
3.1 Introdução.....	34
3.1.1 Contextualização.....	34
3.1.2 Objetivos.....	37
3.1.2.1 Objetivo geral.....	37
3.1.2.2 Objetivos específicos.....	37
3.2 Referencial teórico .....	38
3.2.1 Comunidades virtuais .....	38
3.2.1.1 Projetos de <i>software</i> livre comunitários .....	40
3.2.2 Governança em organizações tradicionais .....	42
3.2.2.1 Visão clássica da estrutura.....	43
3.2.2.2 Visão clássica do controle.....	45
3.2.3 Governança em projetos de <i>software</i> livre comunitários .....	48
3.2.3.1 Mecanismos estruturais em projetos de <i>software</i> livre comunitários .....	49
3.2.3.2 Mecanismos processuais de controle em projetos de <i>software</i> livre comunitários.....	53
3.2.3.3 Desenvolvimento das comunidades de <i>software</i> livre.....	56
3.2.4 Aporte teórico da tese.....	59
3.2.4.1 Abordagem dos <i>stakeholders</i> .....	60
3.2.4.2 Teoria da dependência de recursos.....	62

3.2.4.3 Organizações temporárias.....	64
3.2.5 Desenvolvimento do argumento central da tese .....	65
3.3 Método.....	67
3.3.1 Delineamento da pesquisa .....	68
3.3.2 Seleção dos casos .....	69
3.3.3 Coleta de dados .....	71
3.3.3.1 Entrevistas.....	72
3.3.3.1.1 Seleção dos participantes da pesquisa .....	74
3.3.3.2 Pesquisa documental .....	77
3.3.4 Análise de dados .....	78
3.3.5 Cuidados metodológicos.....	81
3.3.6 Desenho da pesquisa .....	82
3.4 Análise dos resultados.....	84
3.4.1 Projeto CoGrOO .....	84
3.4.1.1 Trajetória de desenvolvimento do projeto CoGrOO.....	86
3.4.1.2 Estrutura do projeto CoGrOO.....	92
3.4.1.2.1 Estrutura do projeto CoGrOO: interface com o projeto OpenNLP.....	94
3.4.1.3 Processos de controle do projeto CoGrOO.....	99
3.4.1.3.1 Processos de controle do projeto CoGrOO: interface com o projeto OpenNLP .....	102
3.4.2 Projeto Noosfero .....	107
3.4.2.1 Trajetória de desenvolvimento projeto Noosfero.....	108
3.4.2.2 Estrutura do projeto Noosfero.....	115
3.4.2.3 Processos de controle do projeto Noosfero.....	119
3.4.3 Projeto Kytos .....	125
3.4.3.1 Trajetória de desenvolvimento do projeto Kytos.....	126
3.4.3.2 Estrutura do projeto Kytos.....	131
3.4.3.3 Processos de controle do projeto Kytos.....	135
3.4.4 Análise conjunta dos casos .....	141
3.4.4.1 Estrutura.....	142
3.4.4.2 Processos de controle .....	147
3.4.4.3 Transformações decorrentes do desenvolvimento dos projetos.....	152
3.4.4.3.1 Associação dos casos ao aporte teórico e conceitual da tese .....	155
3.5 Conclusões.....	158
3.5.1 Síntese dos resultados.....	158

3.5.1.1 Estrutura organizacional e decisória.....	159
3.5.1.2 Processos de controle .....	161
3.5.1.3 Transformações e associação dos casos ao aporte teórico e conceitual da tese.....	164
3.5.2 Confronto com os objetivos e com a tese proposta.....	167
<b>4 CONSIDERAÇÕES FINAIS.....</b>	<b>170</b>
4.1 Limitações e direcionamentos futuros do estudo.....	172
<b>REFERÊNCIAS.....</b>	<b>174</b>
<b>APÊNDICE A ó CARACTERÍSTICAS GERAIS DO PROJETO E ROTEIRO DE ENTREVISTA .....</b>	<b>184</b>
<b>APÊNDICE B ó E-MAIL SOLICITANDO INDICAÇÃO DE PROJETO PARA ESTUDO DE CASO.....</b>	<b>186</b>
<b>APÊNDICE C ó E-MAIL PARA AGENDAMENTO DAS ENTREVISTAS .....</b>	<b>187</b>
<b>APÊNDICE D ó FORMULÁRIO ENCAMINHADO PARA INTEGRANTES DO PROJETO APACHE OPENNLP.....</b>	<b>188</b>

## 1 CONSIDERAÇÕES INICIAIS

A presente seção tem o propósito registrar considerações fundamentais para melhor entendimento da realização deste trabalho de tese, cujos passos envolveram dois momentos bem definidos: o primeiro, quando as atenções da pesquisadora se detiveram na elaboração do ensaio teórico sobre a temática da governança em projetos colaborativos comunitários; o segundo, quando da concentração de esforços no preparo do projeto de pesquisa e na operacionalização da sua parte empírica. Ambos os momentos, convém destacar, intrinsecamente relacionados ao atendimento dos objetivos da tese.

A primeira fase, em que se contém o ensaio teórico, teve como cerne das preocupações esclarecer o que seriam projetos colaborativos em formato comunitário voltados para a elaboração de determinado artefato, bem como explicitar as diferenças de governança entre esses projetos e o modelo tradicional de organização. A literatura utilizada foi basicamente a de projetos comunitários da área de *software* livre, uma vez que a intenção da tese, em sua parte empírica, seria a de aprofundar a compreensão do funcionamento desses coletivos. De modo geral e didático, sem a pretensão de afirmar que os dois modos de produção (o comunitário e o tradicional) constituem extremos polarizados, verificou-se, a partir dos estudos analisados para o artigo, que existem distinções entre as formas de funcionamento das iniciativas comunitárias em relação aos empreendimentos tradicionais. No entanto, tais distinções, em virtude da variedade de comunidades existentes, precisam ser contextualizadas em função das características de desenvolvimento de cada projeto colaborativo comunitário. As evidências apontaram que alguns projetos colaborativos comunitários, por apresentarem certo amadurecimento, já podem ter formalizado elementos estruturais e de controle em sua governança que levam a um funcionamento que os aproximam das organizações tradicionais, ainda que atributos comunitários sejam mantidos.

Os resultados iniciais alcançados pelo ensaio teórico foram apresentados em congressos relevantes da área de administração e de sistemas de informação, representados pelo Encontro da Associação Nacional dos Programas de Pós-Graduação em Administração (EnANPAD 2017) e no *Americas Conference on Information Systems* (AMCIS 2018). Atualmente, o artigo está sendo formatado para a publicação em revista científica.

A segunda fase, em direção à pesquisa empírica, efetivou-se com a ida da pesquisadora a campo para verificação da tese central proposta. Assim, a partir da decisão de que os projetos colaborativos seriam representados por agrupamentos da área de *software* livre em formato de comunidades virtuais, buscou-se, nesse ambiente, obter o entendimento

da governança pelos elementos estruturais e processos de controle identificados nos coletivos. No entanto, além de identificar os procedimentos de governança adotados, a investigação dos projetos de *software* livre selecionados para a pesquisa teve o propósito de compreender o perfil de formalização dessa governança, ao considerar a história e as características de desenvolvimento desses projetos. Mesmo o estudo apresentando um recorte transversal, tornou-se possível uma averiguação mais dinâmica da governança, por meio de uma perspectiva temporal que identificasse as transformações havidas nos projetos de *software* livre, e que são relevantes para a explicação da configuração atual do exercício da gestão nesses coletivos. Tal dimensão temporal foi conseguida, principalmente, no conteúdo das narrativas dos participantes que conheciam os projetos de *software* livre analisados, bem como pela pesquisa documental empreendida sobre esses mesmos projetos. Pretende-se que os resultados alcançados nesta fase empírica sejam convertidos em artigo para a publicação em congressos acadêmicos da área de administração e sistemas de informação, e, posteriormente, para submissão à revista científica.

Como será possível constatar adiante, as duas etapas deste estudo, apresentadas nas próximas seções, foram determinantes no sentido de possibilitar o oferecimento das contribuições acadêmicas e gerenciais almejadas pela tese. Para alguns pesquisadores, possuir o formato comunitário pode significar a existência de um arranjo organizacional anárquico, ou sem nenhum meio de gestão formal estabelecido para o alcance dos resultados pretendidos. Nesta pesquisa, no entanto, partiu-se do pressuposto que alguma governança, mesmo que de forma embrionária, já existe em iniciativas desenvolvedoras de artefatos competitivos e de qualidade, como é o caso de alguns coletivos voltados para a produção aberta de *software* livre. Além disso, a diversidade de projetos comunitários de *software* livre existentes torna imperativo que tais projetos sejam compreendidos a partir de suas histórias e das transformações decorrentes das respectivas trajetórias de desenvolvimento vivenciadas. Ter essa preocupação no estudo da governança, evita que as mesmas características sejam atribuídas de forma indistinta para qualquer iniciativa que eventualmente esteja ancorada no modelo virtual de desenvolvimento comunitário. Todos esses tópicos, portanto, foram examinados neste trabalho de tese, na passagem do ensaio teórico e por ocasião da pesquisa empírica, o cerne da segunda fase da investigação.

Diante do exposto, a questão central desta tese, em torno da qual se pautaram todos os esforços direcionados aos estudos teórico e empírico realizados, consiste na seguinte: *como as transformações que ocorrem no decorrer da trajetória de desenvolvimento dos projetos de software livre comunitários podem levar à formalização da governança?*



## 1.1 Objetivo geral da tese

Com o propósito de responder a questão de pesquisa, o objetivo principal da presente tese é o de entender como as transformações que acontecem na trajetória de desenvolvimento dos projetos de *software* livre comunitários levam a eventuais formalizações da governança desses projetos.

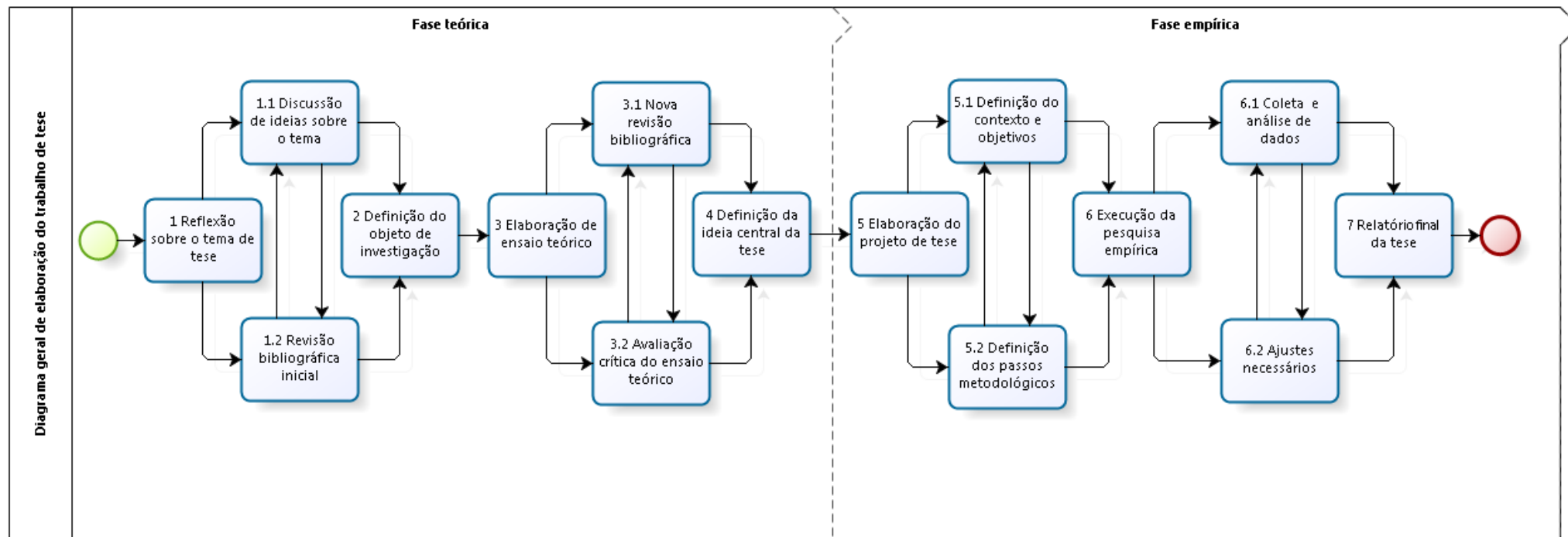
## 1.2 Objetivos específicos da tese

Os objetivos específicos delineados, que levaram a realização das fases teórica e empírica trazidas nesta tese, foram:

- a) identificar, a partir da revisão da literatura de *software* livre, as características gerais e de governança que distinguem os projetos colaborativos comunitários;
- b) verificar, empiricamente, como a governança, na perspectiva da estrutura e dos processos de controle, ocorre e se transforma na trajetória de desenvolvimento dos projetos de *software* livre comunitários;
- c) entender, a partir das transformações verificadas, como a formalização da governança acontece na trajetória de desenvolvimento dos projetos de *software* livre comunitários.

De forma alinhada aos objetivos, o diagrama a seguir propõe-se a ilustrar todas as etapas realizadas no percurso desta investigação. Esse diagrama evidencia que a transição do ensaio teórico para a pesquisa empírica se deu dentro da lógica de um raciocínio que teve o propósito de unir as diversas partes do estudo. Há, como se tenta demonstrar, um ponto de partida e um ponto de chegada, neste último formando-se, naturalmente, um conjunto de dados, análises e procedimentos que vieram de ensejar o sumário dos capítulos, subcapítulos e demais seções desta tese.

**Figura 1** Diagrama geral de elaboração do trabalho de tese



Fonte: Elaborada pela autora (2019)

## 2 ENSAIO TEÓRICO

Governança de projetos colaborativos organizados como comunidades: reflexões teóricas sobre as iniciativas de *software* livre

### Resumo

Os projetos colaborativos organizados em comunidades, diferentemente das organizações tradicionais, normalmente hierárquicas e formalizadas, têm a significativa participação de voluntários e são formados por indivíduos distribuídos geograficamente, engajados e interconectados por meio de uma plataforma tecnológica em um trabalho de interesse mútuo, criando e compartilhando livremente produtos baseados no conhecimento. Apesar de existirem várias pesquisas sobre o tema, quando comparados às organizações burocráticas, os agrupamentos comunitários ainda carecem de um melhor entendimento sobre como governam suas ações para alcance dos resultados coletivos. Assim, com o propósito de contribuir com a teoria organizacional, o presente ensaio buscou aprofundar a compreensão dos projetos colaborativos organizados em comunidades e dos seus aspectos de governança, comparando-os com as organizações tradicionais. Para atender a este objetivo de pesquisa foi analisada a literatura sobre teorias organizacionais e *software* livre. Como reflexões do ensaio, considera-se que, apesar de ser possível distinguir os arranjos comunitários de outros tradicionais, sobretudo em função dos seus aspectos estruturais e dos processos de controle empregados, um fator crucial que nem sempre é considerado nessa diferenciação refere-se às transformações decorrentes do desenvolvimento dos projetos. Nesse sentido, acredita-se que, conforme os projetos comunitários cresçam ou se tornem bem-sucedidos, as transformações decorrentes do desenvolvimento aproxima esses projetos ao funcionamento das organizações tradicionais, ainda que características do modelo comunitário possam ser mantidas.

**Palavras-chave:** Projetos colaborativos. Comunidades. Tecnologia. Governança.

### 2.1 Introdução

O cenário atual em que as organizações estão inseridas tem possibilitado o surgimento de negócios modernos e flexíveis, em resposta à complexidade e às mudanças constantes do ambiente (Adler, Heckscher, & Prusak, 2011; Fjeldstad, Snow, Miles, & Lettl, 2012). Nesse contexto, arranjos organizacionais caracterizados por terem maior fluidez e

adaptabilidade (Faraj, Jarvenpaa, & Majchrzak, 2011), e também por fazerem uso intensivo das tecnologias da informação e comunicação (TICs) na interação dos seus integrantes, têm surgido na sociedade, como é o caso dos projetos colaborativos organizados em comunidades (O'Mahony & Lakhani, 2011; Seidel & Stewart, 2011).

Esses projetos incorporam fortemente o trabalho voluntário (Shah, 2006; Seidel & Stewart, 2011) e são constituídos por integrantes que embora não estejam no mesmo espaço físico, possuem um propósito comum e atuam em ações colaborativas que acontecem em espaços virtuais, disponibilizados pela Internet, para criar e compartilhar recursos e saberes (Lee & Cole, 2003; Crowston, Wei, Eseryel, & Howison, 2007; Fjeldstad et al., 2012). Como resultado dessas ações encontram-se o surgimento de inovações, a solução de problemas, dentre outras produções baseadas no conhecimento e disponibilizadas publicamente (Chen & O'Mahony, 2009; O'Mahony & Lakhani, 2011). Um exemplo bastante disseminado de projetos colaborativos comunitários são as iniciativas de *software* livre, nas quais o código-fonte de um programa é disponibilizado abertamente ao público para receber contribuições, de modo que o relacionamento entre os indivíduos para o aprimoramento do *software* acontece principalmente em comunidades existentes na Internet (Shah, 2006; Crowston et al., 2007; Santos Júnior, Kuk, Kon, & Pearson, 2013).

Os projetos colaborativos estruturados em comunidades, portanto, têm uma forma de funcionamento que difere da perspectiva tradicional de organização (Panchal, 2010; Lindberg, Berente, Gaskin, & Lyytinen, 2016). Ainda usando como referência o setor de *software*, enquanto nas organizações que operam nos moldes tradicionais, tal como a Microsoft na produção do Windows, o desenvolvimento do programa acontece em uma estrutura de trabalho mais burocrática e formal, com atividades confidenciais para fins comerciais, nos milhares de projetos comunitários iniciados na Internet esse desenvolvimento surge em um espaço virtual onde pontificam as relações informais de trabalho e a livre criação e compartilhamento de conhecimentos (Lee & Cole, 2003; Xu, Xu, & Lin, 2005; Seidel & Stewart, 2011). Assim, compreender como as comunidades organizam as suas atividades para alcançar os resultados esperados constitui um importante foco de investigação para posicionar tais arranjos na literatura das teorias organizacionais. Corroborando essa perspectiva, O'Mahony & Ferraro (2007) destacam que, apesar de um problema de pesquisa relevante ser como as organizações governam suas ações para atingir objetivos, a maior parte das investigações aborda as formas mais burocráticas de organização, enquanto menor atenção tem sido dada ao formato em comunidade.

Como consequência disso, as comunidades permanecem tendo um papel de menor destaque nos estudos organizacionais, o que leva à necessidade de aperfeiçoamento desse campo de pesquisa (O'Mahony & Ferraro, 2007; O'Mahony & Lakhani, 2011), para que possa refletir os empreendimentos modernos que surgiram com os avanços das TICs (Zheng, Zhao, & Stylianou, 2013). Indo além, faz-se necessário que os estudos expliquem melhor como produtos inovadores e complexos são desenvolvidos nos modelos comunitários (Lindberg et al., 2016), e como tais modelos se organizam na falta de controles tradicionais e de meios de coordenação mais formalizados (Panchal, 2010; Arazy, Daxenberger, & Lifshitz-Assaf, 2016). No alcance dessas questões o conceito de governança é fundamental, pois envolve os meios que direcionam e controlam os indivíduos em iniciativas total ou parcialmente autônomas, como são os projetos colaborativos comunitários (Markus, 2007; Li-Ying & Salomo, 2013).

Diante do exposto, o objetivo do presente ensaio teórico é definir e descrever o que são projetos colaborativos em formato de comunidade e quais os aspectos de governança que sobressaem nesses projetos, diferenciando-os das organizações tradicionais. Pretende-se responder às seguintes perguntas de pesquisa: 1) O que são projetos colaborativos em formato de comunidades? 2) Quais os principais mecanismos de governança encontrados nesses projetos? Na busca por respostas às questões de pesquisa propostas, também foram apresentadas reflexões críticas sobre o desenvolvimento dos projetos colaborativos organizados em comunidades e as transformações da governança decorrentes desse desenvolvimento. No atendimento desse objetivo foram utilizadas as literaturas de *software* livre e teorias organizacionais.

É importante frisar que os projetos colaborativos organizados em comunidades, nas pesquisas do tema, apresentam níveis de maturidade distintos (West & O'Mahony, 2005), abordam diversas visões da governança (Markus, 2007; Li-Ying & Salomo, 2013) e são retratados em diferentes contextos (O'Mahony & Lakhani, 2011). No que se refere ao contexto, tais projetos podem ser representados por arranjos intra e interorganizacionais, ou por coletivos livres, que funcionam além dos limites físicos de determinada organização (O'Mahony & Lakhani, 2011). Considerando isso, a ideia deste ensaio teórico é a de trazer à tona, em comparação com os empreendimentos tradicionais, os atributos gerais e de governança dos projetos comunitários configurados como coletivos livres. No curso dessa ideia, no entanto, reconhece-se que esses projetos têm a possibilidade de mudar a forma de operar em função do desenvolvimento alcançado. Assim, as suas características não se

apresentam estáticas ou polarizadas em relação aos arranjos tradicionais, já que transformações podem ocorrer ao longo do tempo.

Além da presente introdução, este ensaio conta com uma seção que explica o que são os projetos colaborativos comunitários. Posteriormente, são apresentados os aspectos de governança que se destacam nas comunidades, bem como as principais percepções sobre as características dessas comunidades, seguindo-se das conclusões e das referências usadas na pesquisa.

## **2.2 Projetos colaborativos organizados em comunidades**

Antes de abordar especificamente os projetos colaborativos comunitários, faz-se necessário entender o que é uma comunidade. A definição de comunidade tem lugar central nas ciências sociais. Apesar das discussões sobre o tema terem se iniciado desde o século XIX, com destaque para estudos clássicos da sociologia como os de Tönnies, Weber e Durkheim, o interesse pelo conceito permanece até os dias atuais (Cohen, 2013). Gläser (2001) compreende a comunidade como um conjunto de indivíduos que percebem ter algo em comum uns com os outros, sendo as suas ações e interações influenciadas por essa percepção. Lave e Wenger (1991) afirmam que a comunidade não necessariamente implica em co-presença, grupo com identificação bem definida ou fronteiras visíveis, mas refere-se a indivíduos que participam de um sistema de atividades no qual compartilham entendimentos sobre o que fazem, o que isso significa em suas vidas e para as suas respectivas comunidades.

Nas pesquisas organizacionais, apesar do interesse pela temática da comunidade acontecer há pelo menos trinta anos, as suas definições ainda são dispersas e os níveis de análise não estão claros (O'Mahony & Lakhany, 2011). As comunidades investigadas podem se encontrar em diferentes estágios de maturidade (West & O'Mahony, 2005) e ser representadas tanto por empreendimentos em fase inicial de funcionamento, quanto por iniciativas que já estão em fase avançada de operação, constituindo projetos bem-sucedidos (West & O'Mahony, 2005; Crowston et al., 2007; Crowston, Jullien, & Ortega, 2013). No que diz respeito ao âmbito de atuação, as pesquisas abordam desde grupos intraorganizacionais voltados para partilha de conhecimentos (Wenger, 1998; 2000), passando pelos projetos interorganizacionais, com o objetivo de realizar cooperação técnica (Hargrave & Van de Ven, 2006; Aldrich & Ruef, 2006), até chegar aos arranjos livres, compostos por indivíduos que transcendem as fronteiras organizacionais e que atuam em produções baseadas no conhecimento (Lee & Cole, 2003; O'Mahony & Ferraro, 2007; Panchal, 2010). O presente

ensaio teórico, como já relatado, concentra-se em abordar as características gerais e de governança que sobressaem nos agrupamentos que se encontram neste último caso (o dos arranjos livres), sem ter a pretensão de oferecer uma visão única da arquitetura comunitária em termos de definição ou de unidade de análise.

Os projetos colaborativos organizados em comunidades são formados por indivíduos que têm um interesse mútuo em determinada área do conhecimento, formando um empreendimento conjunto, com a intenção de compartilhar e desenvolver conhecimento relevante a esse interesse (Wenger, 1998; 2000). Esses projetos incorporam de forma expressiva o trabalho voluntário distribuído (Crowston et al., 2007; Lindberg et al., 2016) e são representados por grupos cujos integrantes atuam conjuntamente em torno de um objeto comum de trabalho (O'Mahony & Lakhany, 2011), o que pode resultar na resolução de um problema, na produção de determinada tecnologia ou de outros novos conhecimentos (Panchal, 2010; Faraj et al., 2011). Nos projetos comunitários, as trocas e as produções de novos saberes pelos seus membros, ocorridas no interior das redes de práticas compartilhadas desenvolvidas (Wenger, 2000), são realizadas de forma aberta e têm como resultado uma ampla variedade de produtos baseados no conhecimento (Dahlander & O'Mahony, 2011; Seidel & Stewart, 2011). A coprodução de conhecimentos nas comunidades baseia-se no princípio da auto-organização (Fjeldstad et al., 2012; Arazy et al., 2016). Essa é uma característica notada principalmente nos estágios iniciais dos coletivos, uma vez que são os próprios atores integrantes dos projetos os responsáveis pelo gerenciamento de práticas colaborativas internas, na ausência ou presença limitada da hierarquia e de procedimentos de trabalho mais formalizados e estruturados (Crowston et al., 2007; Panchal, 2010). Assim, por ter essa configuração, a natureza permeável das fronteiras dos projetos comunitários e a auto-organização percebidas no agrupamento tornam esses arranjos poderosos *locus* de criatividade e inovação (Lee & Cole, 2003; Dahlander, Frederiksen, & Rullani, 2008).

Como os integrantes dos projetos colaborativos dispostos em comunidades não necessariamente estão em um mesmo espaço físico, o uso das tecnologias da informação e comunicação (TICs) torna-se crucial para o surgimento e operação dessa arquitetura (Li et al., 2008; Seidel & Stewart, 2011). São os ambientes virtuais propiciados pela difusão das TICs que tornam possível que os integrantes de uma comunidade, mesmo dispersos no tempo e espaço, tenham um *lugar* comum para colaborar, de modo que possam compartilhar ideias e criar em conjunto novos artefatos (Lee & Cole, 2003; Zhan, Bai, & Liu, 2007).

Os projetos colaborativos de *software* livre constituem um dos domínios mais disseminados no estudo da arquitetura em comunidade, tanto na literatura de sistemas de

informação, quanto na literatura organizacional (Seidel & Stewart, 2011). O sistema operacional Linux é um dos casos mais conhecidos e de sucesso (Panchal, 2010; Grabher & Ibert, 2014). Esses projetos funcionam em comunidades *online*, nas quais indivíduos ou organizações abrem o seu *software* com o propósito de receber contribuições de voluntários e desenvolvedores heterogêneos que se encontram dispersos além das fronteiras formais de uma organização, seja para resolver problemas, seja para desenvolver novos aplicativos (Lee & Cole, 2003; Crowston et al., 2007; Li, Yoo, & Zang, 2016). Apesar das pesquisas em *software* livre serem frequentemente citadas como modelo de organização baseada no trabalho comunitário *online*, esse formato organizacional se estende para domínios distintos, a exemplo da produção coletiva de artigos no Wikipedia.org, bem como da produção de artefatos em diversas áreas do conhecimento, como biotecnologia, nanotecnologia, serviços financeiros, cuidados da saúde, esporte etc (Seidel & Stewart, 2011; Kolbjørnsrud, 2016).

### **2.3 Governança em projetos colaborativos organizados em comunidades**

Embora existam várias pesquisas que investigam a arquitetura comunitária, especialmente no contexto de *software* livre (Shaikh & Vaast, 2016), a teoria da governança ainda se encontra mais concentrada no modelo convencional de organização, com estrutura hierárquica e processos marcados pela formalização (Child & Rodrigues, 2003). Diante dessa realidade, é necessário que a disciplina da governança seja aprofundada nos estudos sobre as comunidades com atuação virtual (O'Mahony & Ferraro, 2007; Li-Ying & Salomo, 2013), evidenciadas por uma organização do trabalho que, mesmo se diferenciando dos arranjos tradicionais, tem operado na produção de bens e serviços inovadores e de alta qualidade (Lee & Cole, 2003; Panchal, 2010).

A abordagem da governança nas organizações, e em especial na esfera dos arranjos comunitários, é algo complexo e multidimensional, pois envolve diversos elementos tais como a estrutura organizacional e decisória, os processos técnicos e de gestão e os direitos de propriedade da produção realizada (Markus, 2007). Li-Ying e Salomo (2013) destacam que a dificuldade em compreender a governança nos modelos comunitários com atuação *online* advém da ausência de clareza acerca do seu significado nesses modelos e das divergências sobre quais meios de direção são apropriados aos diferentes tipos de comunidades existentes. Além disso, a própria governança é um fenômeno intrincado nos projetos comunitários (Markus, 2007), já que esses coletivos possuem uma natureza colaborativa aberta e distribuída, na qual indivíduos heterogêneos decidem onde, com quem e no que trabalhar, o



que dificulta o gerenciamento das atividades produtivas (Dahlander & Wallin, 2006; Dahlander et al., 2008). Em função dessas questões, as pesquisas encontram-se dispersas em diversos aspectos da governança, o que torna a projeção de um desenho da governança nos projetos comunitários virtuais algo particularmente difícil de ser obtido (Li-Ying & Salomo, 2013; Kolbjørnsrud, 2016).

Apesar do desafio em se alcançar um entendimento prático e teórico comum sobre o tema da governança nas comunidades (Markus, 2007; Li-Ying & Salomo, 2013), algumas definições do termo são encontradas na literatura do tema. Demil e Lecocq (2006) compreendem a governança como um *framework* institucional que regula as transações entre os atores da comunidade. Para Markus (2007), a governança é representada pelos meios usados para alcançar a direção e o controle dos indivíduos em um projeto comunitário. Com base nas definições anteriores, Li-Ying e Salomo (2013) concebem a governança como uma estrutura dinâmica de mecanismos formais e informais, que regulam as práticas conjuntas desenvolvidas pelos membros das comunidades, ensejando o controle, com o propósito de ordenar, reduzir conflitos e trazer ganhos mútuos para os seus membros.

A partir dessas definições chega-se à compreensão de que a principal finalidade dos mecanismos de governança é assegurar o direcionamento e o controle dos indivíduos que atuam nos ambientes comunitários (Demil & Lecocq, 2006; Dahlander et al., 2008). Outra questão referente à governança, notada nas definições, diz respeito à percepção de que esse conceito é algo dinâmico (Shah, 2006; O'Mahony & Ferraro, 2007; Li-Ying & Salomo, 2013), o que torna possível o seu entendimento não só em termos da estrutura que define a configuração do arranjo organizacional, mas também em função dos processos que gerenciam as ações dos indivíduos para a consecução de resultados coletivos (Markus, 2007; Li-Ying & Salomo, 2013). Partindo dessa constatação, este ensaio teórico, ao abordar a temática da governança nas comunidades, busca entendê-la por meio dessas duas vertentes principais: a estrutural, que envolve o desenho organizacional, de papéis e de decisões do projeto (Demil & Lecocq, 2006; Markus, 2007); e a processual, que envolve os esquemas de controle empregados na dinâmica do trabalho colaborativo para assegurar o alcance dos objetivos estabelecidos pela comunidade (Child & Rodrigues, 2003; Lattemann & Steiglitz, 2005; Xu et al., 2005).

### 2.3.1 Mecanismos estruturais

Em sua essência, a estrutura constitui o alicerce sobre o qual se apoiam todos os componentes que fazem uma organização ter vida (Simeray, 1978). A base fornecida pela estrutura permite que pessoas, recursos e tarefas sejam coordenados para o alcance dos objetivos da organização (Mintzberg, 2011; Aldrich, 2007). Em algumas pesquisas, a terminologia *arquitetura* tem sido usada como sinônimo de estrutura, referindo-se ao desenho universal das características das organizações (Carroll & Hannan, 2000; Maier, Emery, & Hilliard, 2001). Nesse sentido, ambas as definições são usadas para descrever como determinado sistema encontra-se arranjado (Eppinger & Browning, 2012; Fjeldstad et al., 2012). Dois tipos básicos de estrutura são encontrados nas organizações: a formal, baseada nas relações oficiais de trabalho; a informal, surgida espontaneamente a partir das interações sociais (Vasconcellos & Hemsley, 2002; Mintzberg, 2011). Os elementos que compõem a estrutura, a seu turno, são representados pela especialização, departamentalização, centralização, formalização, hierarquia e amplitude de controle (Aldrich, 2007; Robbins, 2010).

É a partir da combinação entre os tipos de estrutura e da variação existente entre os elementos anteriormente citados, que diferentes formatos organizacionais são estabelecidos (Mintzberg, 2011), tais quais os explicitados a seguir com base em Vasconcellos e Hemsley (2002) e Jones (2013). No formato funcional, a divisão do trabalho acontece pelo agrupamento de pessoas que ocupam posições e tarefas semelhantes, com habilidades similares ou que fazem uso dos mesmos recursos. Na configuração divisional, os grupos são estabelecidos com base em demandas específicas de produtos, mercados ou clientes. Na composição baseada em projetos, os indivíduos são reunidos em equipes interfuncionais, formadas de acordo com a natureza do empreendimento. No desenho matricial são utilizados dois ou mais tipos de departamentalização sobre o mesmo grupo de pessoas, mas geralmente envolve a combinação da organização funcional e por projetos. Na configuração em redes, os grupos de indivíduos são coordenados por contratos e acordos, sem existir uma autoridade hierárquica.

Enquanto os modelos tradicionais se caracterizam por exibir maior nível de formalização, unidade de comando, especialização elevada e comunicação vertical, as disposições mais modernas, tais como a matricial e por projetos, apresentam um menor grau de formalização, baixa especialização e comunicação que não depende de níveis hierárquicos (Vasconcellos & Hemsley, 2002). Os projetos colaborativos comunitários, foco do presente ensaio, estão inseridos nesse cenário das tipologias modernas (Li-Ying & Salomo, 2013), apresentando arquitetura flexível e adaptável ao ambiente em que se inserem (Galbraith,

2010), de forma que possuem várias particularidades em relação às organizações tradicionais, no que diz respeito ao seu modo de estruturação (Demil & Lecocq, 2006; Fjeldstad et al., 2012).

A definição de autoridade nos arranjos comunitários não se configura pela existência de uma hierarquia na qual os níveis inferiores das organizações possuem uma relação de subordinação com os níveis superiores (Aldrich, 2007; Robbins, 2010). Nesses arranjos, o trabalho apresenta uma perspectiva horizontalizada, que é definida pelo estabelecimento de relações laterais de autoridade entre os membros do agrupamento (Dahlander & O'Mahony, 2011; Seidel & Stewart, 2011). Nas relações laterais, a autoridade dos indivíduos é função das suas habilidades e é notada pelos direitos de decisão ou responsabilidades que assumem, tornando-se maior quanto mais próximos esses indivíduos se encontram do centro da comunidade (Dahlander & O'Mahony, 2011). Em decorrência da sua organização horizontalizada, as comunidades apresentam uma cadeia de comando pouco definida, o que faz com que predominem no seu âmbito as interações decorrentes da estrutura informal e a existência de relações de trabalho menos formalizadas (Seidel & Stewart, 2011; Arazy et al., 2016).

O ordenamento do trabalho nas comunidades possui algumas especificidades, uma vez que a adesão dos membros aos projetos depende fortemente do trabalho voluntário e, nesses casos, não é estabelecido um vínculo empregatício por contrato formal de trabalho e nem os membros recebem remuneração financeira (Lattemann & Stieglitz, 2005; Shah, 2006; Seidel & Stewart, 2011). Ademais, os projetos comunitários estão fundamentados no compartilhamento aberto de conhecimentos, um indicativo de que qualquer indivíduo pode participar e ter acesso ao que é produzido, bem como deixar de participar da comunidade (Lee & Cole, 2003; Seidel & Stewart, 2011). Por ter esse perfil, as fronteiras que definem os contornos de um arranjo comunitário são fluidas e pouco definidas (Faraj et al., 2011), de modo que as contribuições dos integrantes tendem a ser mais fracas nas imediações periféricas do agrupamento e se tornam mais fortes conforme se aproximam do núcleo central, onde estão os indivíduos mais ativos do projeto e que têm maior autoridade (Seidel & Stewart, 2011).

No que tange ao sistema de comunicação dos projetos comunitários, como os seus integrantes estão distribuídos geograficamente, os recursos de TIC são essenciais para desenvolver as atividades colaborativas do grupo (Zheng et al., 2013; Hamersly, 2015). Dentre os recursos que compõem a infraestrutura tecnológica das comunidades, destacam-se o uso listas de *e-mail*, *chat*, videoconferências, grupos virtuais, fóruns de discussão, dentre

outros (Zhan et al., 2007; Seidel & Stewart, 2011). O uso desses recursos diminui a assimetria de informações nas comunidades e dá suporte à transparência das ações desempenhadas (Lattemann & Stieglitz, 2005). Apesar dos projetos comunitários serem dependentes das TICs e o seu uso ser uma das características distintivas deles (Zhan et al., 2007), existem comunidades que também têm interações face a face e assumem um formato híbrido, presencial e virtual a um só tempo (O'Mahony & Ferraro, 2007; Grabher & Ibert, 2014).

Em relação à definição de responsabilidades, ainda que alguns papéis sejam identificados nos projetos comunitários, o fato da cadeia de comando ter um caráter mais informal e pouco definido, principalmente quando os integrantes se afastam do núcleo da comunidade (Seidel & Stewart, 2011), faz com que os indivíduos realizem diversas funções e tarefas que se sobrepõem (Faraj et al., 2011; Arazy et al., 2016). Nesse contexto, em vez de papéis formais de trabalho, são comumente encontrados nas comunidades papéis emergentes, que estão no centro dos processos de produção do conhecimento e que surgem à medida que as atividades de trabalho são estabelecidas (Zammuto, Griffith, & Majchrzak, 2007; Arazy et al., 2016). São papéis, portanto, que se desenvolvem na própria execução das tarefas, em que não há um planejamento ou alguém que é designado previamente para aquela função específica (Lee & Cole, 2003), o que demonstra, em geral, uma baixa especialização desses papéis nas comunidades. Contudo, ainda que exista essa turbulência e mobilidade dos papéis no nível individual, a estabilidade do trabalho na arquitetura comunitária é conseguida pelo fato da atividade colaborativa ser centrada em torno do artefato produzido, o que dá um direcionamento aos integrantes dos projetos (Arazy et al., 2016).

Um dos papéis de destaque e que normalmente é bem definido no ambiente comunitário é o da liderança (Xu et al., 2005), pois a sua atuação tem um relevante impacto no sucesso dos projetos, no sentido de possibilitar a construção dos relacionamentos, fornecer informações sobre o trabalho, persuadir as pessoas a oferecerem o seu apoio e de observar e avaliar os membros da equipe (Druskat & Wheeler, 2003). É a liderança que desempenha o papel mais importante no controle dos projetos comunitários virtuais, implementando estratégias e práticas de gestão que têm impacto no desempenho e na viabilidade do projeto (Xu et al., 2005; Hamersly, 2015).

### 2.3.2 Mecanismos processuais

Os mecanismos processuais de governança identificados nas iniciativas comunitárias estão fundamentados nos processos de controle (Lattemann & Stieglitz, 2005; Xu et al.,

2005; Markus, 2007), executados com o propósito de assegurar que indivíduos heterogêneos, e cujos objetivos divergem parcialmente, se movam em uma ação cooperativa e direcionem os seus comportamentos para a consecução dos resultados organizacionais (Ouchi, 1979; Das & Teng, 1998). Várias tipologias de controle são encontradas na literatura do tema. A classificação trazida por Ouchi (1979) divide o controle em três mecanismos: 1) os de mercado, os quais mensuram e remuneram as contribuições individuais com base nos resultados alcançados; 2) os burocráticos, que estão centrados na hierarquia e na legitimação da autoridade para controle dos comportamentos; e 3) os de clã, que têm como cerne a socialização dos indivíduos para convergência de objetivos. Vilariño e Schoenherr (1987) dividiram o controle também em três categorias: 1) o direto, praticado por meio da imposição da ordem e da vigilância expressa; 2) o estrutural, produzido nos níveis mais formalizados da organização, envolvendo aspectos técnicos e burocráticos; e 3) o difuso, que tem como base os pressupostos culturais. Scott (1995), por sua vez, trouxe os seguintes enfoques de controle: 1) o regulativo, que usa um nível superficial e direto, ao abranger leis, sanções, normas e vigilância; 2) o normativo, que está respaldado na certificação, por meio do reconhecimento e titulação; e 3) o cognitivo, que se encontra alicerçado em premissas culturais.

A partir das tipologias exemplificadas, é possível entrever que o controle varia ao longo de duas vertentes fundamentais, a dos mecanismos formais e a dos informais de controle (Ouchi, 1979; Vilariño & Schoenherr, 1987; Scott, 1995; Das & Teng, 1998; Xu et al., 2005). Seguindo essa orientação, dentre as espécies formais de controles, localiza-se o controle de comportamento, que recompensa os indivíduos com base no grau em que seguem os procedimentos, assim como o controle de resultados, em que os atores são gratificados em função dos objetivos cumpridos. Já como modalidade de controle informal, que enfatiza os relacionamentos entre os membros de um agrupamento, encontra-se o controle social, que se apoia em ideologias e em princípios culturais decorrentes desses relacionamentos (Ouchi, 1979; Das & Teng, 1998; Xu et al., 2005).

No contexto dos projetos comunitários virtuais, até mesmo por conta de esses arranjos diferirem em seus aspectos estruturais das organizações com funcionamento tradicional, como já sublinhado anteriormente, são utilizados meios de governança que visam a aumentar o compromisso dos seus participantes, agindo não só sobre a sua motivação em contribuir (Shah, 2006), mas também no sentido de adequar os seus comportamentos às finalidades dos projetos (Lattermann & Stieglitz, 2005; Xu et al., 2005). No curso dessas questões, o modo de assegurar o controle nas comunidades é viabilizado especialmente por mecanismos processuais com um perfil mais democrático (O'Mahony & Ferraro, 2007) e

voltados à socialização dos indivíduos (Seidel & Stewart, 2011), em contraponto ao uso de meios mais formais e centrados no comportamento das pessoas e nos resultados alcançados, notadamente encontrados nas organizações tradicionais (Lattermann & Stieglitz, 2005; Xu et al., 2005). Contudo, é relevante frisar que, mesmo diante da fluidez dos arranjos comunitários (Li et al., 2016), procedimentos de controle burocráticos, tais como o estabelecimento de regras, normas e regulamentos, também podem ser empregados e combinados nesses arranjos com as formas de controle social existentes (Markus, 2007; O'Mahony & Ferraro, 2007; Chen & O'Mahony, 2009). Xu et al. (2011) também perceberam a presença de controle de resultados no estudo de comunidades de *software* livre, porém, de forma menos formalizada do que nos times tradicionais.

Os processos de controle democráticos atuam por meio de uma maior oportunidade de participação dos integrantes na produção e na gestão da comunidade (German, 2003; O'Mahony & Ferraro, 2007), assim como pela transparência das ações de governança, fornecendo uma maior visibilidade em termos do conteúdo produzido, dos papéis desempenhados, do processo de tomada de decisão e das relações de trocas e construção de conhecimentos (Vincent & Camp, 2004; Markus, 2007). As ações democráticas nas comunidades dependem da existência de um contexto de meritocracia nos projetos, de forma a incentivar contribuições de qualidade dos membros (O'Mahony & Ferraro, 2007), por meio do reconhecimento de mérito técnico e profissional (Dahlander & O'Mahony, 2011), status, assunção de responsabilidades e oportunidades para melhorar o seu próprio desenvolvimento (Von Krogh, Spaeth, & Lakhani, 2003; Stewart, 2005). Markus (2007) afirma que essa abertura democrática das comunidades também tem um caráter motivacional, o que torna os projetos que ostentam esse formato participativo e transparente mais bem-sucedidos na missão de atrair participantes dispostos a colaborar. Além disso, o uso de elementos democráticos nas comunidades não apenas limita a base de autoridade, mas também se reveste de função adaptativa, pois permite que o sistema de governança da comunidade se ajuste à medida que os integrantes aprendem a interpretar a liderança e a autoridade em um ambiente comunitário (O'Mahony & Ferraro, 2007).

No que se refere aos mecanismos processuais sociais, nota-se nos projetos comunitários o emprego de formas de controle que promovem a conscientização dos indivíduos no sentido de alinhar os seus objetivos com os propósitos do coletivo (Lattermann & Stieglitz, 2005; Xu et al., 2005). Com base nas tipologias de controle exemplificadas anteriormente, esses mecanismos integram a abordagem de clã trazida por Ouchi (1979), o controle difuso de Vilariño e Schoenherr (1987) e o enfoque cognitivo de Scott (1995), que

tem como pressuposto o processo de socialização intensiva dos membros de uma organização por meio do estabelecimento de uma cultura compartilhada (Xu et al., 2005; Seidel & Stewart, 2011). Envolve, ademais, a promulgação de valores, hábitos, crenças, regras e normas que devem ser internalizados pelos atores (Ouchi, 1979), e que funcionam como um contrato coletivo ao ajudar os participantes das comunidades a entenderem os comportamentos que são aceitos, valorizados e aqueles que não são permitidos (Xu et al., 2005; Li-Ying & Salomo, 2013).

Seidel e Stewart (2011) afirmam que esse controle cultural, reforçado pela interação entre os fundadores e os contribuintes dos projetos, é algo crítico para a manutenção e o crescimento de um arranjo comunitário. Os mecanismos processuais usados no controle social auxiliam no fortalecimento da confiança e da cooperação entre os atores de um grupo, por conseguirem integrar pessoas heterogêneas, que estão distantes entre si e que se comunicam principalmente pelo suporte da TIC (Lattermann & Stieglitz, 2005; Zhan et al., 2007), no sentido de fazer com que as suas ações, na ausência de uma proteção contratual explícita, aconteçam em torno de um propósito comum (Dahlander & Wallin, 2006; O'Mahony & Ferraro, 2007).

A cultura estabelecida nas comunidades se consolida pelas práticas compartilhadas construídas pelos indivíduos ao longo da realização do trabalho colaborativo (Wenger, 1998). Essas práticas, a seu turno, manifestam-se por meio de rotinas, símbolos, histórias, metáforas, protótipos, ritos, linguagem etc, que integram o repertório de representações da comunidade (Wenger, 1998; Seidel & O'Mahony, 2014). É a existência desse repertório compartilhado de representações que produz entre os membros das comunidades uma referência sobre como interagir, alinhar o trabalho e ter um entendimento comum das ações que executam (Okhuysen & Bechky, 2009; Seidel & O'Mahony, 2014). Ademais, o repertório compartilhado de representações fortalece o senso de identificação e de pertencimento ao coletivo (Seidel & O'Mahony, 2014), principalmente quando os membros estão mais próximos do centro do agrupamento, o que permite a definição dos limites do grupo e a consolidação da sua identidade (Wenger, 1998; 2000; Seidel & Stewart, 2011).

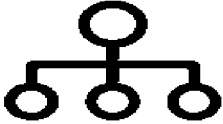
Em consonância com o direcionamento dado pelos mecanismos sociais, outras formas de controle verificadas nos projetos comunitários são representadas pelo controle entre os pares e pelo autocontrole (Child & Rodrigues, 2003; Arazy et al., 2016; Kolbjørnsrud, 2016). O controle entre os pares tem como pressuposto o ajustamento mútuo dos atores por meio da interação durante a realização de determinada tarefa (Sutanto, Kankanhalli, & Tan, 2011), sendo verificado principalmente em ambientes caracterizados pela incerteza sobre os

fins e os meios, como é o caso dos projetos que funcionam em comunidades virtuais (Fjeldstad et al., 2012). Já o autocontrole diz respeito àquela modalidade em que o indivíduo regula a si próprio em relação às ações que executa (Lattermann & Stieglitz, 2005; Kolbjørnsrud, 2016), tendo como antecedentes a complexidade da tarefa, a avaliação ambígua do desempenho e a falta de regras e de procedimentos para completar uma tarefa (Xu et al., 2005). Na classificação de Scott (1995), apresentada anteriormente, ambas as formas de controle integram o enfoque cognitivo, que se caracteriza por ser discreto e difuso, bem como por alicerçar-se em premissas culturais que conduzem os indivíduos a exercerem um alto grau de autocontrole e de controle entre os pares nas organizações em que atuam.

## 2.4 Principais reflexões sobre os projetos colaborativos organizados em comunidades

O Quadro 1, elaborado com base na literatura organizacional e de *software* livre, apresenta um resumo das características gerais e de governança que têm maior predominância nos projetos colaborativos comunitários, em comparação com as organizações tradicionais.

**Quadro 1** Comparação entre organizações tradicionais e projetos colaborativos comunitários

<b>Organizações tradicionais</b>	<b>Organizações virtuais comunitárias</b>
	
Menor flexibilidade adaptativa (Adler et al., 2011; Fjeldstad et al., 2012)	Maior flexibilidade adaptativa (Adler et al., 2011; Faraj et al., 2011; Fjeldstad et al., 2012)
Confidencialidade e finalidade comercial (Lee & Cole, 2003; Xu et al., 2005; Chen & O'Mahony, 2009; Seidel & Stewart, 2011)	Livre criação e compartilhamento de conhecimentos (Lee & Cole, 2003; Demil & Lecocq, 2006; Seidel & Stewart, 2011)
Membros dependentes em suas escolhas para contribuir (Seidel & Stewart, 2011; Lindberg et al., 2016)	Membros independentes em suas escolhas para contribuir, mas com produção interdependente (Seidel & Stewart, 2011; Lindberg et al., 2016; Kolbjørnsrud, 2016)
Hierárquica e com autoridade formal (Lee & Cole, 2003; Demil & Lecocq, 2006; Seidel & Stewart, 2011; Arazy et al., 2016; Lindberg et al., 2016)	Horizontalizada e com autoridade lateral (Panchal, 2010; Dahlander & O'Mahony, 2011; O'Mahony & Lakhany, 2011; Seidel & Stewart, 2011)
Estrutura organizacional e decisória bem definida (Vasconcellos & Hemsley, 2002; Mintzberg, 2011; Demil & Lecocq, 2006; Lindberg et al., 2016)	Estrutura organizacional e decisória pouco definida, com forte atuação da liderança (Xu et al., 2005; Seidel & Stewart, 2011; Arazy et al., 2016; Lindberg et al., 2016)
Limites bem estabelecidos (Child & Rodrigues, 2003; Lee & Cole, 2003;	Limites fluidos, com fronteiras permeáveis de adesão



<b>Organizações tradicionais</b>	<b>Organizações virtuais comunitárias</b>
Chen & O'Mahony, 2009)	(Faraj et al., 2011; Seidel & Stewart, 2011; Li et al., 2016)
Trabalho formalizado por contrato e remuneração (Lattermann & Stieglitz, 2005; Dahlander & Wallin, 2006; O'Mahony & Ferraro, 2007)	Significante incorporação do trabalho voluntário (Shah, 2006; Seidel & Stewart, 2011; Lindberg et al., 2016)
Infraestrutura física e tecnológica (Lee & Cole, 2003; O'Mahony & Lakhany, 2011; Seidel & Stewart, 2011)	Infraestrutura essencialmente tecnológica (Lee & Cole, 2003; Seidel & Stewart, 2011; Zheng et al., 2013)
Controle com perfil mais formal e burocrático (O'Mahony & Ferraro, 2007; O'Mahony & Lakhany, 2011; Seidel & Stewart, 2011; Arazy et al., 2016)	Controle com perfil mais informal, democrático e com forte presença da meritocracia (O'Mahony & Ferraro, 2007; Panchal, 2010; Arazy et al., 2016)
Controle de comportamentos, controle de resultados (Lattermann & Steiglitz, 2005; Xu et al., 2005; Markus, 2007)	Controle social, controle entre pares e autocontrole (Lattermann & Steiglitz, 2005; Xu et al., 2005; Kolbjørnsrud, 2016)

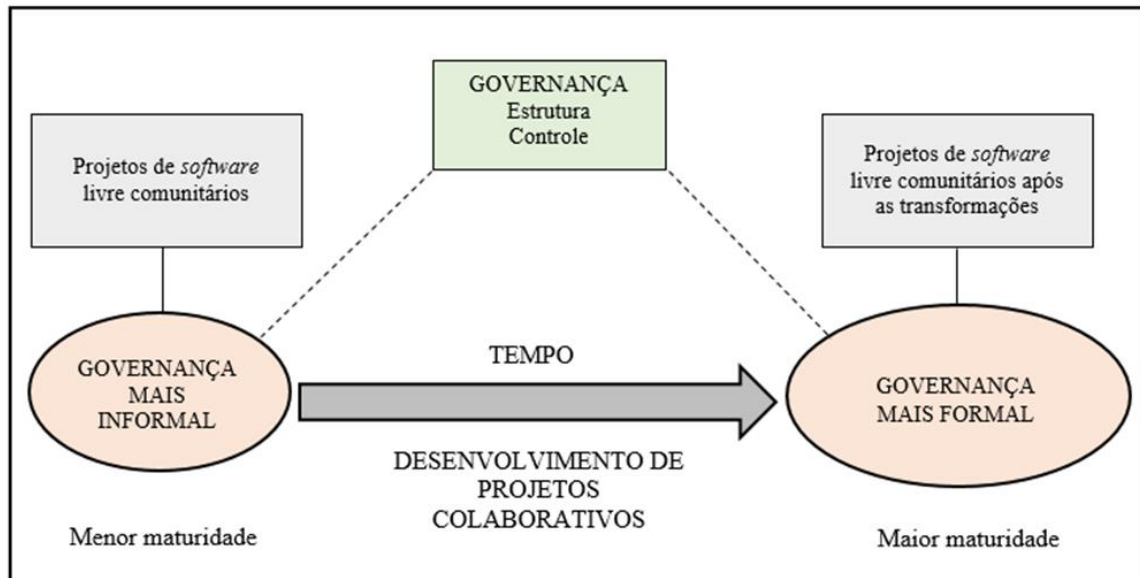
**Fonte:** Elaborado pela autora, com base na revisão de literatura realizada (2019)

Expostas as diferenças entre os empreendimentos tradicionais e em comunidades, uma questão fundamental a considerar no estudo dos projetos colaborativos comunitários refere-se ao desenvolvimento dos agrupamentos (Crowston et al. 2007; Crowston et al., 2013). Como as características dos projetos comunitários e os mecanismos de governança são dinâmicos, podendo ser estabelecidos, ajustados ou abandonados, de acordo com a necessidade, existem variações desses projetos em função da complexidade e do crescimento alcançado (Xu et al., 2005; Forte, Larco, & Bruckman, 2009; Li-Ying & Salomo, 2013). Assim, presume-se, que na medida em que os projetos começam a se expandir e a amadurecer, a tendência é que os mecanismos de gestão usados se modifiquem para acompanhar essas transformações (Lattermann & Stieglitz, 2005; Xu et al., 2005). Em outras palavras, espera-se que o coletivo deixe de apresentar alguns atributos de um arranjo do tipo *obazarö*, com troca aberta de conhecimentos, por meio da aceitação de contribuições indiscriminadas e configuração mais informalizada, tal como preconizado por Raymond (1999) ao se referir às comunidades de *software* livre, para então se aproximar de uma arquitetura condizente com a visão catedral ou tradicional de organização, com estrutura e processos de coordenação e controle mais formalizados (Lattermann & Stieglitz, 2005; Xu et al., 2005; Panchal, 2010).

A Figura 2 expressa essa ideia central, demonstrando que no decorrer do desenvolvimento dos projetos colaborativos comunitários, como os de *software* livre, a governança passa por transformações que podem resultar numa maior formalização desses projetos. Quando isso acontece, tais projetos têm a possibilidade de aproximar o seu modo de

funcionamento ao das organizações tradicionais, ainda que alguns traços comunitários sejam mantidos.

**Figura 2** ó Transformações da governança no desenvolvimento dos projetos colaborativos comunitários



**Fonte:** Elaborada pela autora com base na revisão de literatura realizada (2019)

A evolução dos projetos, inclusive, pode fazer com que as comunidades desenvolvam relacionamentos com outros projetos, ou passem a ser geridas, apoiadas e patrocinadas por organizações (West & O'Mahony, 2005; Lattermann & Stieglitz, 2005; Li-Ying & Salomo, 2013). Na Internet, diga-se a propósito, são encontrados projetos dos mais diversos tipos, representados tanto por comunidades que não têm muitos contribuintes além dos seus fundadores iniciais, ou que perderam sua popularidade ao longo do tempo, até comunidades em estágio avançado de desenvolvimento (Chengalur-Smith & Sidorova, 2003; Stewart & Gosain, 2006; Seidel & Stewart, 2011). De acordo com Stewart e Gosain (2006), são esses projetos maduros que têm recebido a maior parte da atenção das pesquisas sobre a arquitetura em comunidades, em contraposição a projetos menores e menos maduros.

Na elaboração deste ensaio, ao se tentar esclarecer os atributos gerais e de governança dos empreendimentos comunitários, diferenciando-os dos tradicionais, com foco especial na literatura de *software* livre, foi percebido que projetos que tiveram êxito e estão em estágios avançados de maturidade, como é o caso do Linux, Apache, Gnome, dentre outros, são bastante citados como exemplos de iniciativas alicerçadas no formato de comunidades (Lee & Cole, 2003, Shah, 2006; Seidel & Stewart, 2011; Lindberg et al., 2016). Outras pesquisas, no entanto, ao conjecturar sobre os arranjos organizados em comunidades,

tiveram o cuidado de reconhecer que o desenvolvimento dos projetos torna a estrutura e os processos de controle mais elaborados no âmbito desses coletivos (Lattermann & Stieglitz, 2005, Xu et al., 2005, Panchal, 2010). Em função disso, é de fundamental importância ressaltar que as características comunitárias apresentadas aqui tão só refletem a essência da arquitetura comunitária, o que possibilita cogitar que embora projetos desenvolvidos sejam citados como ilustração dessa arquitetura, na realidade, ao serem investigados em detalhes, podem até se afastar dessa essência em alguns desses aspectos e aproximar-se de um perfil mais tradicional de funcionamento.

## 2.5 Conclusões

Os projetos colaborativos organizados em comunidades, dos quais são exemplos notáveis as iniciativas de *software livre*, são empreendimentos que ainda precisam ser mais estudados na literatura organizacional. Constituem arranjos portadores de atributos bastante singulares que os diferenciam das organizações ditas tradicionais (Lee & Cole, 2003; Oø Mahony & Lakhany, 2011; Seidel & Stewart, 2011) e que requerem formas de governança adaptadas a essa realidade, diferindo das noções clássicas de gerenciamento (Demil & Lecocq, 2006; Markus, 2007; Oø Mahony & Ferraro, 2007; Li-Ying & Salomo, 2013). No presente ensaio, além do delineamento das características gerais dos projetos comunitários, tentou-se também examinar e apreender o conceito de governança, a partir da vertente estrutural e dos processos de controle empregados nesses projetos.

Do ângulo da estrutura, a governança de projetos colaborativos diferencia-se do padrão convencional pela observância de relações de autoridade não amparadas na subordinação hierárquica. Essas relações ocorrem, sobretudo, numa perspectiva horizontalizada, pelo reconhecimento dos saberes e das habilidades dos indivíduos. A via escolhida para a organização do trabalho exerce influência crucial no estabelecimento dessa linha de exercício da autoridade no âmbito do agrupamento, pois está baseada na livre adesão ao projeto e no compartilhamento aberto de informações e conhecimentos. Essa via, no plano da definição de responsabilidades, faz com que o papel da liderança tenha função-chave no êxito do empreendimento, ainda que os demais papéis estabelecidos sejam pouco estruturados. Por sua vez, quando o foco é a comunicação, destaca-se a utilização das TIC pelos indivíduos da comunidade como solução para o alcance dos propósitos estabelecidos, notoriamente em situações em que a dispersão física dos membros do coletivo é uma realidade.

Na perspectiva processual, os mecanismos de governança dos projetos comunitários são representados por ações de controle que têm forte alinhamento com os elementos estruturais. É função dos mecanismos processuais garantir o trabalho colaborativo no âmbito de um projeto de contornos fluídos, com forma de operação essencialmente virtual, marcada por fatores motivacionais dos integrantes distanciados dos paradigmas tradicionais, fundada na ideia central do compartilhamento de conhecimentos e da informalidade contratual, entre outras características. Para guiar e evitar discrepâncias entre os comportamentos dos indivíduos com vistas ao alcance dos resultados do coletivo, os mecanismos processuais usados baseiam-se essencialmente em meios democráticos que incentivam a participação, bem como em processos socializantes que estão ancorados em práticas de trabalho compartilhadas.

Apesar de este ensaio alcançar a sua proposta de apresentar alguns dos atributos gerais e de governança que distinguem os projetos comunitários, pelo que é possível notar na própria literatura sobre o tema, a governança nessas comunidades não se afigura como algo rígido e definitivo. Ao contrário, constitui um arcabouço de contornos flexíveis e dotado de forte adaptabilidade ao ambiente e à evolução do projeto, transformando-se à medida que esse projeto se torna mais complexo e bem-sucedido, com a finalidade de garantir a sobrevivência e superar desafios (Lattermann & Stieglitz, 2005; Xu et al., 2005; Crowston et al., 2007; Panchal, 2010; Li-Ying & Salomo, 2013).

Nesse último ponto, o que se percebe é que, apesar do aspecto do desenvolvimento ser algo crucial para o estudo da governança, nem sempre ele é discutido nas pesquisas, o que faz com que projetos de sucesso sejam vistos como arranjos que seguem a lógica comunitária, sem haver, em parte significativa dos casos, a preocupação em debater a circunstância temporal e de amadurecimento em que eles se encontram. Isso ocorre principalmente na área de *software* livre, na qual existe uma maior concentração das publicações sobre governança em comunidades *online*, onde projetos como Linux, Apache, Gnome, algumas vezes são citados como exemplos de empreendimentos comunitários, sem existir uma contextualização acerca das transformações pelas quais passaram essas estruturas. Tais projetos, inclusive, contam, atualmente, com o apoio de outras organizações no seu gerenciamento, o que permite pensar, a despeito de terem surgido baseados no contexto da colaboração em comunidades, que podem estar operando, em função do crescimento alcançado, de modo mais próximo ao das organizações tradicionais, ainda que alguns traços comunitários sejam mantidos.

Realizadas essas considerações, espera-se que estudos futuros tragam novos aportes de conhecimentos que complementem as perspectivas da literatura aqui discutidas, como

forma de caracterizar a essência das iniciativas comunitárias, diferenciando essas iniciativas do formato tradicional de organização. Indo um passo além, espera-se que novas pesquisas se detenham com maior atenção aos detalhes atrelados ao desenvolvimento dos projetos colaborativos. A abordagem dessas questões, no campo teórico e empírico, concorreria para preencher importantes lacunas quanto à explicação de mudanças nos atributos e meios de governança eventualmente ocorridas desde o surgimento do coletivo até o seu estágio atual ou mais avançado de maturidade. Tais ideias, uma vez levadas à prática, não só na área de *software* livre, mas também em outros tipos de comunidades, permitiriam esclarecer se os projetos comunitários, ao evoluírem ou obterem êxito, se aproximam das constituições típicas das organizações tradicionais, ou se de fato o fazem preservando, em diferentes graus, os seus valores de origem.

### 3 PESQUISA EMPÍRICA

As transformações da governança na trajetória de desenvolvimento dos projetos de *software* livre comunitários: evidências empíricas a partir da análise da estrutura e dos processos de controle.

#### **Resumo**

Vários estudos nas áreas de administração e de sistemas de informação têm abordado os arranjos organizacionais em formato de comunidades, tais como os representados pelos projetos colaborativos de *software* livre. Com o propósito de contribuir para o avanço das pesquisas no tema, o presente trabalho assumiu o objetivo de analisar a governança, na perspectiva da estrutura e dos processos de controle, assim como as suas transformações, em projetos de *software* livre comunitários, a partir da apreciação da trajetória de desenvolvimento desses projetos. A pesquisa fez uso da abordagem qualitativa por meio da estratégia de estudo de casos múltiplos, os quais foram escolhidos a partir dos objetivos delimitados e de critérios estabelecidos pela literatura, já que a intenção era selecionar projetos que apresentassem diferentes características de desenvolvimento. A coleta de dados realizou-se mediante entrevistas com participantes que tinham diferentes perfis e que integravam o grupo central dos coletivos, bem como por meio de pesquisa documental nos ambientes virtuais utilizados e de notícias divulgadas acerca dos projetos investigados. O exame do material coletado apoiou-se no emprego da técnica de análise de conteúdo. Os resultados obtidos possibilitaram não só por identificar os principais mecanismos de governança utilizados pelos projetos de *software* livre comunitários, mas também explicitar as transformações que ocorreram nesses projetos na trajetória de desenvolvimento vivenciada. Nessas transformações, ao considerar os empreendimentos numa perspectiva mais ampla, indo além da comunidade, foi possível verificar que os coletivos apresentaram momentos de maior e de menor atividade, de modo que o desenvolvimento não demonstrou acontecer em um ciclo sequencial evolutivo. Além disso, constatou-se que os relacionamentos estabelecidos com atores organizacionais externos, especialmente para a obtenção de recursos, influenciaram na formalização da estrutura e dos processos de controle, sendo fundamentais para a sobrevivência dos agrupamentos.

**Palavras-chave:** Projetos de *software* livre comunitários. Governança. Estrutura. Controle. Desenvolvimento.

### 3.1 Introdução

Embasado nos resultados alcançados no ensaio teórico, este capítulo traz o delineamento do contexto e da problemática de estudo, assim como dos objetivos gerais e específicos que se pretende alcançar na pesquisa proposta.

#### 3.1.1 Contextualização

Nos últimos anos, os avanços e a difusão das tecnologias da informação e comunicação (TICs), bem como as rápidas transformações e o aumento da complexidade do ambiente organizacional (Schreyögg & Sydow, 2010; Seidel & Stewart, 2011), têm possibilitado o surgimento de negócios modernos, flexíveis e dirigidos para a livre troca e geração de conhecimentos, a exemplo dos empreendimentos em formato de comunidades virtuais (Faraj, Jarvenpaa, & Majchrzak, 2011; Arazy, Daxenberger, & Lifshitz-Assaf, 2016).

Essas comunidades constituem iniciativas caracterizadas pela dispersão geográfica dos seus integrantes e pelo uso intensivo de recursos tecnológicos nas atividades executadas (Li-Ying & Salomo, 2013), sendo formadas por conjuntos de indivíduos que colaboram voluntariamente por meio de plataformas *online*, em áreas de trabalho de interesses comuns (O'Mahony & Lakhany, 2011). Tais iniciativas podem resultar na resolução de problemas, em inovações ou na geração de diversos tipos de novos produtos e/ou conhecimentos (O'Mahony & Lakhany, 2011; Snow, Fjeldstad, & Lettl, 2011; Lakhani, Lifshitz-Assaf, & Tushman, 2013; Kolbjørnsrud, 2016). Assim, cada vez mais as comunidades virtuais têm sido *locus* de criatividade, inovação e coprodução de artefatos baseados nos esforços, experiências e saberes dos seus membros (Lee & Cole, 2003; Lindberg, Berente, Gaskin, & Lyytinen, 2016).

Na literatura sobre o tema das comunidades virtuais são apontadas diversas origens, naturezas e locais em que esses coletivos se fazem presentes (O'Mahony & Ferraro, 2007; Li-Ying & Salomo, 2013). Nesse sentido, as pesquisas apontam situações que vão desde comunidades dos tipos intra e interorganizacionais, até os agrupamentos independentes, sendo esses últimos representados por projetos colaborativos que funcionam além das fronteiras das organizações tradicionais (Lee & Cole, 2003; O'Mahony & Ferraro, 2007).

Em relação aos projetos colaborativos independentes, um fenômeno comumente disseminado é o das comunidades de *software* livre (Li-Ying & Salomo, 2013). Essas comunidades são estruturas criadas na Internet com a intenção de abrirem o processo produtivo de *software* para populações mais amplas, mediante a participação significativa de

voluntários em suas atividades colaborativas (Shah, 2006; Seidel & Stewart, 2011). Tratam-se, portanto, de agrupamentos formados por desenvolvedores e usuários que, apesar da distância física, interagem em ambiente virtual, onde o código-fonte do programa é disponibilizado publicamente com vistas ao recebimento de contribuições externas que visem à resolução de problemas e à incorporação de melhorias desse programa (Von Hippel & Von Krogh, 2003; Santos, Kuk, Kon, & Pearson, 2013).

Assim como acontece em outros empreendimentos comunitários voltados para o desenvolvimento de artefatos, a lógica de produção dos projetos de *software* livre organizados em comunidades difere da observada nas organizações tradicionais, uma vez que, nessas últimas, o *software* é criado de forma restrita, em um ambiente hierárquico, com estrutura e processos de controle formalizados (Lee & Cole, 2003; Seidel & Stewart, 2011; Arazy et al., 2016). No entanto, ainda que as comunidades de *software* livre tenham particularidades no que diz respeito aos seus formatos e processos de gestão (Xu, Xu, & Lin, 2005; Kolbjørnsrud, 2016), esses empreendimentos têm se destacado pela atuação no desenvolvimento de aplicações inovadoras e de alta qualidade (Lee & Cole, 2003; Forte & Lampe, 2013; Arazy et al., 2016; Bauer, Franke, & Tuertscher, 2016). Dessa forma, é fundamental que os estudos que abordem o tema das comunidades *online*, como as de *software* livre, se aprofundem nas especificidades dos seus modos de operação (Bauer et al., 2016), no sentido de compreender como essas comunidades, na ausência ou menor presença de meios formais, conduzem suas ações para o alcance de objetivos ou metas previamente estabelecidos (Lattemann & Stieglitz, 2005; Panchal, 2010; Arazy et al., 2016).

Um conceito-chave para a verificação de como o trabalho é realizado nas organizações em geral é o da governança. Esse fenômeno multifacetado envolve os meios formais e informais orientadores dos esforços dos integrantes de um coletivo para o alcance dos resultados pretendidos (Sitkin, Cardinal, & Bijlsma-Fralkema, 2010; Foss & Klein, 2013; Li-Ying & Salomo, 2013). Nas comunidades de *software* livre a governança é abordada a partir de várias vertentes, envolvendo aspectos como estrutura, processos de controle e de comunicação, além dos termos da licença de *software* e procedimentos técnicos de negócio (Markus, 2007). No entanto, em virtude da flexibilidade típica desses arranjos comunitários, o estudo da governança nos projetos colaborativos de *software* livre requer uma abordagem dinâmica que considere as mudanças decorrentes do desenvolvimento de cada comunidade (Lattemann & Stieglitz, 2005; Xu et al., 2005; Panchal, 2010).

Diante do panorama apresentado, o que se nota na literatura de *software* livre é que, apesar das características do modelo comunitário serem ressaltadas no sentido de diferenciar



essas comunidades dos formatos organizacionais tradicionais (Lee & Cole, 2003; Seidel & Stewart, 2011; Arazy et al., 2016), nem sempre as transformações ocorridas e o desenvolvimento alcançado são considerados como aspectos fundamentais dessa diferenciação. Em decorrência disso, projetos colaborativos de *software* livre mais maduros em termos de funcionamento são vistos como negócios que funcionam próximos à lógica informal de gestão, em oposição ao modelo hierárquico de organização (Raymond, 1999; Demil & Lecocq, 2006; Markus, 2007), sem que essa visão, no entanto, encontre apoio numa contextualização suficiente no que concerne à trajetória de desenvolvimento desses empreendimentos (Ferraz & Santos Júnior, 2018).

Essa lacuna, a da ausência de explicação sobre o crescimento de cada projeto, faz com que uma diversidade de iniciativas virtuais seja vista como exemplo de arranjos organizacionais comunitários, sem o aprofundamento das suas características de funcionamento e do modo como se desenvolveram. Tal fato pode ser notado em trabalhos como os de Lee e Cole (2003), Shah (2006), Seidel e Stewart (2011) e Lindberg et al. (2016), que citam projetos de *software* livre de sucesso, a exemplo do Linux, sem, contudo, realizar a análise das modificações ocorridas nessas comunidades no decorrer do tempo.

A seu turno, Lattemann e Stieglitz (2005), Xu et al. (2005) e Panchal, (2010), apesar de reconhecerem a existência de diferentes estágios de desenvolvimento que os negócios de *software* livre podem apresentar, não se basearam no exame empírico e em conclusões práticas de suas abordagens dentro dos limites do problema em tela. Assim, comunidades que surgiram de forma independente, no âmbito organizacional, ou até aquelas que se desenvolveram ao ponto de atrair outras organizações para o seu campo de funcionamento, não deveriam ser vistas, indistintamente, como representantes do modelo comunitário, sem a apropriada consideração das variações estruturais e dos processos de controle decorrentes dos modos de operação e do nível de crescimento alcançado.

Outra questão observada em relação às pesquisas sobre o formato comunitário, tal como os projetos colaborativos de *software* livre, é que em comparação com as organizações tradicionais formais, esses projetos ainda precisam ser mais investigados em seus aspectos internos de organização, no que se refere às ações pelas quais gerenciam suas atividades de produção coletiva de artefatos (O'Mahony & Ferraro, 2007; O'Mahony & Lakhany, 2011; Forte & Lampe, 2013; Germonprez, Kendall, Kendall, & Mathiassen, 2016). O mesmo acontece em relação ao conceito de governança, uma vez que a maioria dos estudos sobre esse tema ainda se encontra concentrada nos modelos hierárquicos de organização do trabalho (Child & Rodrigues, 2003; Kolbjørnsrud, 2016). Em relação ao conceito de governança,

convém ressaltar, ainda que os ambientes comunitários como os de *software* livre sejam percebidos como resistentes à formalização que se vê nas organizações tradicionais, um mínimo de gestão se faz necessário para que os resultados dos projetos sejam alcançados. Ou seja, mesmo que de forma incipiente, acredita-se que a governança seja um elemento presente e que se transforma dentro de qualquer estrutura comunitária que tenha atividade produtiva voltada para inovação e/ou criação de artefatos e conhecimentos (Crowston & Howison, 2006; Forte & Lampe, 2013).

Realizadas estas considerações, é em meio a esta problemática que se coloca a presente tese, cuja intenção é proporcionar respostas à seguinte questão de pesquisa: *como a governança, representada pelos elementos estruturais e processos de controle, aparece e se transforma no desenvolvimento dos projetos de software livre comunitários?* O propósito da pesquisa é o de contribuir com os estudos organizacionais não só por buscar entender a governança em espaços comunitários que operam fundamentalmente ancorados nas TICs, mas também por considerar o desenvolvimento dos projetos de *software* livre como requisito crucial para a compreensão dos aspectos da estrutura e dos processos de controle observados nesses coletivos. Com esse foco, acredita-se ser possível verificar que tipo de empreendimento os projetos de *software* livre comunitários se tornam em função da sua história e do amadurecimento alcançado, e o quanto o nível de desenvolvimento deles aproximam-nos ou os distanciam, em termos de governança, do modelo comunitário.

### **3.1.2 Objetivos**

#### **3.1.2.1 Objetivo geral**

Analisar a governança e as transformações eventualmente ocorridas no desenvolvimento dos projetos de *software* livre comunitários, a partir da análise dos elementos estruturais e dos processos de controle observados nesses projetos.

#### **3.1.2.2 Objetivos específicos**

Como desdobramentos do objetivo geral, encontram-se os seguintes objetivos específicos:

- a) delinear as principais características observadas nas trajetórias de desenvolvimento dos projetos de *software* livre comunitários;

- b) identificar os procedimentos de governança representados pelos elementos estruturais e processos de controle nos projetos de *software* livre comunitários;
- c) analisar a transformação da governança no desenvolvimento dos projetos de *software* livre comunitários.

## 3.2 Referencial teórico

### 3.2.1 Comunidades virtuais

As comunidades virtuais contam com significativa participação de voluntários e são representadas por grupos de indivíduos que possuem interesses comuns (O'Mahony & Lakhany, 2011), reunidos por intermédio de uma plataforma tecnológica, com o propósito de compartilhar e coproduzir conhecimentos dentro dessa área de interesse (Faraj et al., 2011; Li-Ying & Salomo, 2013). Embora encontros presenciais sejam possíveis, a comunicação mediada pelas tecnologias disponibilizadas pela Internet é a principal forma de interação dos membros dessas comunidades (Zhan, Bai, & Liu, 2007; Grabher & Ibert, 2014).

Como a maior parte dos participantes de uma comunidade virtual estão dispersos geograficamente, o uso das tecnologias da informação e comunicação (TICs) torna-se fundamental para interconectar e permitir o engajamento dos atores nas suas atividades colaborativas, transcendendo eventuais barreiras de tempo e espaço (Lee & Cole, 2003; Faraj et al., 2011; Forte & Lampe, 2013; Bauer et al., 2016). De acordo com Seidel e Stewart (2011), o advento da Internet e a difusão dos recursos tecnológicos tornaram possível a comunicação entre as pessoas que estão distantes umas das outras de forma instantânea e pouco onerosa, oportunizando aos integrantes das comunidades a interação em um espaço virtual que constitui o lugar comum para a realização do trabalho colaborativo. Deve-se às facilidades de comunicação possibilitadas pela Internet e às avançadas aplicações da *web 2.0* a crescente proeminência do engajamento das pessoas nestes tipos de comunidades (Zheng, Zhao, & Stylianou, 2013). Diversas são as tecnologias que podem ser empregadas e combinadas na infraestrutura das comunidades virtuais, de que são exemplos as *wikis*, os *blogs*, fóruns de discussão, *home page*, *e-mail* e listas de *e-mail*, *chats*, webconferências, repositórios de documentos e arquivos multimídia, redes sociais etc (Jensen & Scacchi, 2010; Paily, 2013; Lindberg et al., 2016).

Nas comunidades virtuais, as soluções necessárias para a produção de novos saberes não se concentram em uma única pessoa, mas estão dispersas entre os vários atores que as

integram, o que faz com que as trocas de experiências, informações e recursos complementares nos espaços *online* constituam uma vantagem, ao possibilitar a formação de uma inteligência coletiva no interior desses arranjos (Bonabeau 2009; Bauer et al., 2016). Em decorrência das intensas trocas virtuais de saberes, os produtos gerados por meio das atividades colaborativas estão baseados nos conhecimentos dos membros dos grupos (Seidel & Stewart, 2011) e podem resultar na criação de inovações, na resolução de problemas diversos ou na geração de outros novos conhecimentos (Forte & Lampe, 2013; Kolbjørnsrud, 2016; Bauer et al., 2016).

Nos coletivos virtuais que seguem a lógica comunitária, tais como os iniciados de forma independente (Grabher & Ibert, 2014), a ideia é que sejam auto-organizados, o que significa que são os próprios integrantes que realizam a gestão das atividades nessas comunidades (O'Mahony & Lakhani, 2011; Arazy et al., 2016; Bauer et al., 2016). Além disso, espera-se que estes modelos apresentem uma estruturação e processos de controle menos formalizados do que os das organizações tradicionais, em virtude, principalmente, da abertura do processo produtivo e do emprego da expressiva mão de obra voluntária (Shah, 2006; Arazy et al., 2016; Lindberg et al., 2016).

Seidel e Stewart (2011) destacam que a arquitetura comunitária se caracteriza basicamente por apresentar quatro características principais: fronteiras periféricas fluidas e informais de adesão; elevada incorporação do trabalho voluntário; produção baseada em informações e conhecimentos; compartilhamento de conhecimentos significativamente aberto. Por todos esses atributos, os arranjos comunitários virtuais tendem a um funcionamento cuja lógica diverge daquela que prevalece na organização tradicional, caracterizada, sobretudo, por um perfil comercial e voltado ao lucro, assim como pela estruturação hierárquica, com colaboradores remunerados financeiramente e processos formais de trabalho (Seidel & Stewart, 2011; Arazy et al., 2016).

A literatura sobre os arranjos constituídos no formato de comunidades, em especial os virtuais, encontra-se bastante dispersa em diferentes áreas do conhecimento, o que faz com que o tema receba abordagens distintas em função do tipo empreendimento comunitário a que se refere (Li-Ying & Salomo, 2013), a exemplo das comunidades de prática (Fang & Chiu, 2010; Wenger, 2011), comunidades de usuários (Zheng et al., 2013), comunidades de *software* livre (Bauer et al., 2016), times virtuais (Sutanto, Kankanhalli, & Tan, 2011; Hamersly, 2015), redes de inovação organizacional (Von Hippel, 2007), dentre outras. Além dessas diferentes abordagens, as comunidades virtuais também são investigadas em diversos contextos, que envolvem tanto as iniciativas constituídas no âmbito dos espaços

organizacionais, a exemplo das comunidades intra e interorganizacionais, quanto àquelas que surgem de modo independente e funcionam fora dos limites da organização (O'Mahony & Lakhani, 2011; Li-Ying & Salomo, 2013). Apesar da variedade de domínios e contextos em que os empreendimentos comunitários virtuais são verificados, uma das iniciativas que se destaca é a das comunidades de *software* livre (Shah, 2006; Seidel & Stewart, 2011; Li-Ying & Salomo, 2013), vista em detalhes a seguir, por constituir o foco de estudo desta pesquisa.

### 3.2.1.1 Projetos de *software* livre comunitários

Diante da diversidade de empreendimentos que seguem o paradigma comunitário (Shah, 2006; Li-Ying & Salomo, 2013), os projetos colaborativos de *software* livre, até os dias atuais, continuam sendo uma das iniciativas mais disseminadas de empreendimentos baseados no formato de comunidades virtuais (Lee & Cole 2003; Bauer et al., 2016). Conforme já destacado anteriormente, as comunidades de *software* livre mantêm as principais características dos arranjos comunitários virtuais, nos quais o trabalho é realizado de forma descentralizada, por membros dispersos geograficamente, representados em sua maioria por voluntários não remunerados financeiramente, que colaboram em espaços virtuais mediados pelas TICs (Lee & Cole, 2003; Shah, 2006). Além disso, assim como os outros arranjos comunitários virtuais, as comunidades de *software* livre têm uma forma fluida e flexível de se organizar, pois os seus membros, interações e resultados mudam constantemente ao longo do tempo (Faraj et al., 2011; Li, Yoo, & Zhang, 2016). Tais características influenciam diretamente no modo de gerenciamento das atividades produtivas executadas nessas comunidades (Lattemann & Stieglitz, 2005; Xu et al., 2005).

No estudo de *software* livre é comum ser encontrado o termo *Free/Libre and Open Source Software* (FLOSS), como uma forma mais ampla e neutra de se referir ao *software* livre, desenvolvido e lançado sob uma licença de código aberto (Crowston et al. 2007). Para evitar entrar nas discussões existentes em torno dessa denominação, nesta pesquisa foi utilizado o termo *software* livre. A condução de um projeto de *software* livre é uma atividade que requer intensa aplicação de conhecimentos (Von Krogh, Spaeth, & Lakhany, 2003), pois está centrada numa rede de desenvolvedores e usuários diversos, os quais colaboram e trocam experiências e saberes para desenvolver produtos inovadores e complexos (Lee & Cole, 2003; Shah, 2006; Jensen & Scacchi, 2010). Os indivíduos que atuam nesses projetos o fazem por diversas motivações (West & O'Mahony, 2005), abarcando desde causas intrínsecas, tais como prazer, objetivos de aprendizagem, questões ideológicas, necessidades de cooperação

etc, até razões extrínsecas, a exemplo de perspectivas de emprego e recebimento de benefícios materiais ou simbólicos como status, reputação e prestígio (Xu et al., 2005; Lattemann & Stieglitz, 2005).

As comunidades de *software* livre diferem da produção comercial de *software* (Seidel & Stewart, 2011) porque são estruturas que promovem a aberta utilização do programa e a disponibilização do seu código-fonte, de forma gratuita, pela Internet, para que qualquer pessoa possa acessá-lo. Esses critérios são garantidos, na maior parte das vezes, por licenças como a *Berkeley Software Distribution* (BSD) e a *General Public License* (GPL) (West & O'Mahony, 2005; Lattemann & Stieglitz, 2005; Xu et al., 2005; Kolbjørnsrud, 2016). A ideia central das iniciativas de *software* livre é que todos que delas tomam parte possam realizar contribuições e tenham o direito de usar, modificar, corrigir erros e distribuir o programa, o qual está protegido contra a propriedade intelectual privada (Lattemann & Stieglitz, 2005; Xu et al., 2005). Em contrapartida, nas organizações tradicionais de *software*, a produção ocorre em um contexto de trabalho formal e mantido por relação contratual, no qual o *software* é licenciado e vendido de forma que os usuários não têm acesso ao seu código-fonte, o que faz com que a correção de *bugs* e o suporte seja dependente da empresa que comercializa o programa (Xu et al., 2005). Contudo, mesmo com características particulares de funcionamento, se comparadas com as organizações tradicionais de *software*, as comunidades de *software* livre têm atuado na fabricação de produtos competitivos e de qualidade, o que pode ser notado a partir de projetos colaborativos de sucesso, como o Linux, Apache, Gnome, Debian, dentre outros (Lattemann & Stieglitz, 2005; Shah, 2006; Bauer et al., 2016).

Apesar de o *software* livre ser um produto comum, produzido no âmbito das comunidades virtuais, o seu desenvolvimento pode apresentar uma natureza modular (Von Krogh et al., 2003; Lindberg et al., 2016). Tal característica possibilita que a elaboração da maioria desses sistemas aconteça por meio de contribuições sequenciais, em que o trabalho é visto como complementar e a programação do *software* é dividida em subprojetos paralelos, desenvolvidos de forma independente, sem seguir uma ordem cronológica, sendo integrados posteriormente em um *software* ativo (Lattemann & Stieglitz, 2005; Kolbjørnsrud, 2016). Essa possibilidade de projetar e desenvolver o *software* com um alto grau de modularidade e de forma aberta e livre, inclusive, é uma característica a mais que diferencia as comunidades de *software* livre de outras comunidades virtuais colaborativas (Lindberg et al., 2016; Kolbjørnsrud, 2016). Contudo, é importante frisar que mesmo que a arquitetura das comunidades de *software* livre permita essa atuação mais autônoma e independente de seus

membros, as dependências e interfaces existentes entre os módulos criados, e necessárias ao desenvolvimento de projetos, fazem com que os resultados produzidos sejam interdependentes (Jensen & Scacchi, 2010; Lindberg et al., 2016).

Vistas as características fundamentais das comunidades de *software* livre, nota-se que, ainda que esses projetos compartilhem os principais atributos dos arranjos comunitários, também possuem outros traços que são inerentes ao tipo de comunidades que representam.

### 3.2.2 Governança em organizações tradicionais

Os mecanismos de governança, em termos gerais, referem-se aos meios formais e informais que as organizações empregam para influenciar os seus membros e até outras partes interessadas a contribuírem para a consecução de determinados objetivos (Foss & Klein, 2013; Kolbjørnsrud, 2016). Assim, esses mecanismos podem ser observados tanto no contexto interno, quanto nos relacionamentos interorganizacionais, tendo como principal motivação assegurar a compatibilidade e a consistência de comportamentos operacionais em direção a situação ou estados pretendidos (Albers, 2005; Theurl, 2005; Foss & Klein, 2013). O conceito de governança organizacional é extremamente abrangente e varia em função do contexto em que é empregado, uma vez que não pertence a um único campo de estudos e possui relações e sobreposições a diversas disciplinas, como a sociologia, a economia, a psicologia, a ciência política, dentre outras (Mello & Slomski, 2010; Foss & Klein, 2013).

Um estudo clássico sobre a governança nas organizações, e que tem as suas raízes na economia, a partir das contribuições de Ronald Coase em 1937, é o da Teoria dos Custos de Transação (Tadelis, 2010). Nesse estudo, Williamson (1985) revela que deve existir um alinhamento entre a estrutura de governança das organizações às características das transações que elas realizam, como forma de mitigar os custos das transações que têm o contrato como unidade básica (Williamson, 1985; Zylbersztajn, 2005). Ao longo do tempo, vários outros estudos sobre a governança foram e continuam sendo elaborados, com aplicação em diferentes áreas de trabalho, mas, em sua maioria, ainda baseados em esquemas hierárquicos de organização (Child & Rodrigues, 2003; Kolbjørnsrud, 2016). Como forma de tentar superar esse viés, faz-se necessário que as pesquisas sobre governança lancem também o seu foco aos novos formatos organizacionais existentes e sensivelmente marcados por estruturas de funcionamento mais flexíveis e descentralizadas (Child & Rodrigues, 2003), e que contam com o suporte das TICs, dentre as quais se destacam as iniciativas comunitárias virtuais (O'Mahony & Ferraro, 2007; Kolbjørnsrud, 2016).

Na abordagem dos mecanismos de governança trazida pela literatura organizacional, apesar das múltiplas dimensões pelas quais a governança pode ser retratada, um dado concreto é que frequentemente o conceito aparece associado a componentes que dizem respeito à estrutura e ao controle, enquanto meios voltados ao monitoramento e coordenação das atividades dos arranjos organizacionais (Sitkin et al., 2010; Foss & Klein, 2013). Esta pesquisa segue essa mesma orientação, ao acolher a ideia de governança a partir da literatura tradicional de estrutura e de controle nas organizações, concentrando-se, porém, na abordagem das comunidades de *software* livre. Na seção seguinte são apresentadas as definições preliminares de estrutura e de controle, as quais antecedem o aprofundamento desses componentes da governança no campo das comunidades de *software* livre.

### 3.2.2.1 Visão clássica da estrutura

A estrutura é um elemento imprescindível em qualquer formato organizacional. Ela constitui a base para a atuação do elemento humano e de todas as outras forças que fazem uma organização ter vida (Simeray, 1978). Vasconcellos e Hemsley (1989) entendem a estrutura como o resultado de estudos e decisões à luz dos quais a autoridade é distribuída, as atividades são especificadas e um sistema de comunicação é delineado, de modo a permitir que os indivíduos realizem tarefas e exerçam suas competências para atingir os objetivos da organização. As variáveis estruturais, reforçadas pelos mecanismos de controle, devem encorajar a atividade colaborativa dos indivíduos, bem como dar ensejo a que estes empreguem seus esforços no sentido do êxito da organização (Lorsch, Lawrence, & Dalton, 1970).

Dois tipos de estrutura que se apresentam de forma intrincada e de difícil distinção são percebidos nas organizações. O primeiro tipo refere-se à estrutura formal, aquela deliberadamente planejada e que retrata as relações oficiais de trabalho que balizam os níveis de autoridade e responsabilidades das unidades que compõem as organizações (Vasconcellos & Hemsley, 1989; Mintzberg, 2011). O organograma é a representação gráfica da estrutura formal da organização, sendo um reflexo da sua própria imagem (Simeray, 1978; Vasconcellos & Hemsley, 1989). O segundo tipo refere-se à estrutura informal, que surge de forma espontânea nos grupos de trabalho por meio das interações pessoais, nas quais os indivíduos trocam informações que não passam pelos canais formais de comunicação (Vasconcellos & Hemsley, 1989; Mintzberg, 2011). Esses dois tipos de estrutura podem



ocorrer de forma simultânea e possuem variações na forma como se manifestam em cada organização (Vasconcellos e Hemsley, 1989).

No delineamento da estrutura seis elementos-chave normalmente são considerados e aplicados em qualquer configuração organizacional. Eles variam de forma independente uns dos outros e em diferentes níveis ou graus no modo como acontecem nas organizações, tal como prescrevem as análises de Robbins (2010), Mintzberg (2011) e Aldrich (2007). A especialização, segundo esses autores, refere-se ao grau em que as tarefas na organização são subdivididas em funções separadas. A departamentalização envolve os modos e critérios pelos quais indivíduos, atividades e recursos organizacionais são agrupados. A cadeia de comando ou hierarquia retrata a linha única de autoridade que vai do topo da organização até o escalão mais baixo, e determina quem se reporta a quem na organização. A amplitude de controle compreende a quantidade de subordinados que determinada chefia ou liderança possui. A centralização ou descentralização abrange o grau em que a tomada de decisão e a autoridade se apresenta concentrada em um único ponto da organização (centralização) ou é compartilhada e delegada aos diferentes níveis organizacionais (descentralização). A formalização diz respeito ao grau de padronização, regulamentos e procedimentos usados na condução das tarefas, de modo que quanto maior é a padronização das atividades, menor é a autonomia para se decidir sobre elas.

Em função da departamentalização, diferentes formatos organizacionais podem ser identificados (Robbins, 2010; Mintzberg, 2011; Jones, 2013). No formato funcional, a departamentalização é baseada na especialização (conhecimentos, habilidades, processos ou funções de trabalho) e deriva em uma hierarquia de autoridade (Mintzberg, 2011; Robbins, 2010). O agrupamento funcional faz com que as pessoas sejam reunidas em diferentes fluxos de trabalho, o que pode resultar no surgimento de carreiras especializadas de acordo com a área de *expertise* dos indivíduos (Mintzberg, 2011). No formato divisional, a departamentalização é definida pelos requisitos de localidade, produtos, processos e clientes que a organização busca atender (Jones, 2014). No formato por projetos, a departamentalização é praticada pela formação temporária de equipes de trabalho multidisciplinares para atuação em uma missão específica. Nesse formato, os indivíduos trabalham continuamente em planos que se renovam ou novos desafios que surgem ao término de cada ciclo (Robbins & Coulter, 2007). No formato em redes, a departamentalização dos atores organizacionais acontece em um desenho não hierárquico, devido às necessidades de cooperação, mas com relacionamentos regulamentados por acordos e contratos (Snow, Miles, & Coleman, 1992; Jones, 2013). No formato matricial, finalmente,

a departamentalização ocorre pela combinação de dois ou mais formatos organizacionais, geralmente os formatos funcional e por projetos (Galbraith, 2014; Jones, 2013).

Em sentido amplo, todos os aspectos envolvidos na estrutura devem buscar o alcance de uma congruência com a situação em que a organização se encontra, especialmente em termos de tempo de existência, ambiente em que opera e seu sistema técnico (Mintzberg, 2011). Segundo Vasconcelos e Hemsley (1989), a estrutura deve funcionar em profunda sintonia com a natureza das atividades que a organização realiza e com o espaço no qual atua, exibindo desse modo, e por efeito disso, diversas combinações e relacionamentos entre os tipos e elementos estruturais (Vasconcellos e Hemsley, 1989; Mintzberg, 2011; Jones, 2013). Resumidamente, é possível afirmar que as organizações variam entre dois polos principais, conforme apresenta o Quadro 2.

**Quadro 2** ó Estruturas tradicionais *versus* estruturas inovadoras

<b>Estruturas tradicionais</b>	<b>Estruturas inovadoras</b>
Maior formalização	Menor formalização
Maior especialização	Menor especialização
Unidade de comando	Multiplicidade de comando
Comunicação vertical	Comunicação horizontal
Formas clássicas de departamentalização	Formas avançadas de departamentalização

**Fonte:** Elaborado pela autora, com base em Vasconcellos e Hemsley (1989)

Como demonstrado pelo Quadro 2, as organizações podem variar ao longo de dois extremos, ou seja, desde estruturas mais tradicionais, até estruturas mais inovadoras, existindo múltiplas possibilidades entre esses dois pontos. Nas organizações mais inovadoras se encontram as estruturas matriciais, por projetos e em rede, assim como outros desenhos mais modernos e flexíveis, que fazem uso intenso dos recursos de TIC, tal qual as iniciativas comunitárias (Seidel & Stewart, 2011).

### 3.2.2.2 Visão clássica do controle

Desde os estudos clássicos procedentes da teoria weberiana, até os dias atuais, o conceito de controle é tido como um aspecto fundamental em todas as organizações, mas que ainda precisa ser mais investigado (Vilariño & Schoenherr, 1987; Sitkin et al., 2010). É por meio do controle que se promove o alinhamento entre as capacidades, atividades e desempenho dos indivíduos aos propósitos organizacionais (Sitkin et al., 2010). Na visão de Ouchi (1979), o controle tem como foco fazer com que um conjunto de indivíduos ou unidades, que compartilham parcialmente objetivos congruentes, se movam em ações

cooperativas que levem ao atingimento dos resultados esperados para as organizações. São várias as classificações do controle trazidas pela literatura que aborda o tema, dentre as quais, apenas para citar alguns exemplos, se destacam as definidas por Ouchi (1979), Vilariño e Schoenherr (1987) e Scott (1995).

Para Ouchi (1979), são três os principais dispositivos de controle, a saber: os de mercado, que agem sobre resultados e lidam com a mensuração e a remuneração das contribuições individuais; os burocráticos, que levam em consideração a avaliação do desempenho e a legitimação da autoridade em hierarquias, de sorte que permitem o controle de comportamentos nas organizações; os de clã, que alcançam a cooperação por meio da socialização intensiva dos indivíduos, suportada por elementos culturais, de modo que os objetivos desses indivíduos coincidam ou se sobreponham aos da organização. Ainda na visão desse autor, o desenho de uma organização deve buscar um equilíbrio entre a mensuração dos comportamentos, dos resultados e a socialização dos indivíduos, com vistas a obter a cooperação entre pessoas, a partir da convergência dos seus objetivos e das respectivas organizações em que atuam.

Vilariño e Schoenherr (1987) se reportam aos tipos de controle direto, estrutural e difuso: o direto, como aquele exercido mediante a imposição de ordem, vigilância expressa e regulação; o estrutural, visto como discreto, aquele que corresponde aos níveis formalizados da organização, viabilizado pelas normas, regulamentos, hierarquia e critérios de seleção; o difuso, por fim, aquele que não opera diretamente sobre o processo de tomada de decisão, mas sim por intermédio de pressupostos culturais que levam ao compromisso e à identidade dos indivíduos com a organização, o que o torna um tipo difícil de ser alcançado. A classificação de Scott (1995), por sua vez, traz como enfoques de controle os denominados de regulativo, normativo e cognitivo. Para Scott (1995) o enfoque regulativo é superficial e está ancorado na vigilância, normas técnicas, aspectos legais e sanções que regulam as atividades; o enfoque normativo é exercido pela regulação e sujeição às premissas de decisão, baseando-se na certificação, no reconhecimento e na titulação; o enfoque cognitivo tem como fundamento a adoção de um quadro comum de referência ou de uma situação, tendo o seu aporte em premissas culturais. Essas duas tipologias, a de Vilariño e Schoenherr (1987) e Scott (1995), possuem correspondências entre si, e variam entre práticas de controle que vão das mais visíveis até aquelas que são menos perceptíveis nas organizações (Carvalho, 1999).

Diante das classificações exemplificadas, percebe-se que as formas de controle organizacional variam ao longo de duas vertentes principais, isto é, a formal e a informal (Ouchi, 1979; Das & Teng, 1998). Nesse sentido, enquanto os controles alicerçados na cultura

e na socialização intensiva, tais como o de clã, o difuso e o cognitivo, integram a classe dos meios informais, os demais tipos, voltados essencialmente para os resultados e os comportamentos, distinguem-se como meios mais formalizados (Ouchi, 1979; Das & Teng, 1998). O controle formal lida com desempenho e limites claros, ao empregar regras codificadas, objetivos, procedimentos e regulações que denotam padrões desejáveis de resultados ou comportamentos, ao passo que o controle informal (ou social) lida com a interação social das pessoas, e utiliza valores, normas e a cultura para encorajar os comportamentos desejáveis (Das & Teng, 1998). No caso específico do comportamento, a ideia da formalização visa a reduzir a sua variabilidade e a mantê-lo dentro de parâmetros previamente estabelecidos, com o propósito de aumentar a capacidade de predição do desempenho do capital humano nas organizações (Mintzberg, 2011).

O direcionamento dado pelo controle social abrange duas outras formas de controle, representadas pelo controle entre pares e pelo autocontrole, ambas exigindo fortes laços de compromisso e identidade das pessoas com a organização (Scott, 1995; Carvalho, 1999). O controle entre pares ocorre quando indivíduos que se encontram em um mesmo nível organizacional, ou em um mesmo campo de trabalho, exercem o controle lateral e não hierárquico sobre os comportamentos dos seus pares (Loughry, 2009). Desta maneira, o controle entre pares permite o ajustamento mútuo de atitudes, algo útil à coordenação do trabalho pelo simples processo de comunicação informal (Mintzberg, 2011). O autocontrole, a seu turno, é praticamente automático, e por definição, exercido pelos membros da organização sobre as atividades que eles mesmos executam (Das & Teng, 1998; Van Maanen, 2010). Nesse sentido, são os próprios indivíduos que determinam e direcionam espontaneamente os seus comportamentos aos objetivos organizacionais (Das & Teng, 1998; Van Maanen, 2010).

Tendo em vista as diversas classificações de controle existentes, tais como as que foram aqui apresentadas, é relevante destacar que os mecanismos de controle organizacionais não se pautam por um único tipo de controle e podem ser combinados de diversos modos e graus (Ouchi, 1979). Uma questão que cada vez mais vem sendo considerada nos estudos sobre o controle refere-se ao papel determinante das tecnologias como ferramentas de mediação e de execução dos processos de controle nas organizações (Orlikowski & Baroudi, 1991; Sitkin et al., 2010). As modernas TICs têm mudado o papel dos gestores nas organizações, não apenas porque permitem o monitoramento e armazenamento de informações dos eventos sob observação, com rapidez e eficácia, mas também por facilitar a implementação de práticas de controle que abarcam uma ampla gama de interesses e relacionamentos interpessoais internos (Orlikowski, 1991; Dunbar & Statler, 2010). As

comunidades de *software* livre constituem referência de um formato de empreendimento virtual cujos processos de controle são fundamentalmente amparados pelas TICs (Xu et al., 2005; Seidel & Stewart, 2011).

### 3.2.3 Governança em projetos de *software* livre comunitários

Apesar de existirem várias pesquisas na área de sistemas de informação e de administração que tomam como foco as comunidades de *software* livre (Seidel & Stewart, 2011), os aspectos gerenciais verificados nesses coletivos, voltados essencialmente para a produção de artefatos tecnológicos, ainda necessitam de melhor compreensão (O'Mahony & Ferraro, 2007; Germonprez et al., 2016). As comunidades de *software* livre, assim como as demais iniciativas organizadas no formato comunitário virtual, dadas às características específicas de suas operações, requerem dispositivos de governança adaptados ao seu contexto, já que os procedimentos tradicionais de gestão podem não ser adequados ao ambiente dessas comunidades (Lattemann & Stieglitz, 2005; Xu et al., 2005; Kolbjørnsrud, 2016).

Markus (2007), ao abordar a governança no contexto das comunidades de *software* livre, afirma que o conceito é representado pelos meios que viabilizam o direcionamento e o controle de indivíduos, total ou parcialmente autônomos, no sentido de contribuir conjuntamente em um projeto para desenvolvimento de programas computacionais. Essas práticas de governança acontecem nos espaços virtuais suportados pelos recursos de TIC (Lindberg et al., 2016). Em qualquer projeto comunitário de *software* livre a governança é um aspecto que tenta não só garantir que as contribuições individuais sejam dirigidas à elaboração de produtos inovadores e de alta qualidade (Lee & Cole, 2003; Markus, 2007), mas também possibilitar a promoção de um ambiente favorável à produtividade e atrativo ao recebimento de contribuições relevantes por parte dos participantes do projeto (Lattemann & Stieglitz, 2005; Xu et al., 2005).

Contudo, apesar da importância da governança em empreendimentos comunitários como os de *software* livre, o seu estudo ainda é algo bastante controverso na literatura do tema (Li-Ying & Salomo, 2013), pelo fato dessas comunidades serem vistas como arranjos abertos e diferenciados da lógica hierárquica formal e lucrativa das organizações tradicionais (Lee & Cole, 2003; Faraj et al., 2011; Seidel & Stewart, 2011). Parte significativa das pesquisas que abordam os projetos de *software* livre indica que esses projetos são pouco estruturados, não hierárquicos e que fazem uso de mecanismos de gestão que têm menor

formalização (Lindberg et al. 2016; Arazy et al., 2016; Kolbjørnsrud, 2016). Outros estudos, no entanto, além de reconhecerem essas características, adicionam como conclusões que os mecanismos de governança por eles utilizados são bastante flexíveis e que se modificam ao tempo em que as comunidades amadurecem e seus elementos e conexões se tornam mais intrincados (Lattemann & Stieglitz, 2005; Xu et al., 2005; Jensen & Scacchi, 2010; Panchal, 2010). Faraj et al. (2011) afirmam que mesmo constituindo sistemas adaptativos caracterizados pela fluidez, cujas ações e interesses se transformam no decorrer do tempo, tais iniciativas, ainda assim, são claramente distinguíveis das organizações tradicionais nas suas atividades colaborativas de troca e de criação de conhecimentos.

Ao reconhecer essas diferentes abordagens trazidas pela literatura, o essencial em relação ao conceito de governança é que ele esteja voltado para as peculiaridades do projeto, com vistas a garantir a eficiência e o crescimento do coletivo (Xu et al., 2005; Lattemann & Stieglitz, 2005). Além disso, a governança precisa ser considerada a partir de uma perspectiva dinâmica, capaz de captar as modificações e necessidades dos negócios de *software* livre (Xu et al., 2005; Lattemann & Stieglitz, 2005; Li-Ying & Salomo, 2013). Dentre os múltiplos prismas por meio dos quais a governança nas comunidades de *software* livre pode ser abordada (Markus, 2007; Li-Ying & Salomo, 2013), dois deles se destacam e foram os escolhidos para serem objeto de análise neste trabalho: a estrutura (O'Mahony & Ferraro, 2007; Seidel & Stewart, 2011; Lindberg et al. 2016) e os processos de controle (Lattemann & Stieglitz, 2005; Xu et al., 2005).

Nas próximas seções apresenta-se uma visão geral dos mecanismos estruturais e processuais de controle, tendo como base as principais percepções registradas na literatura sobre as características da governança nos modelos comunitários de *software* livre. Na última seção, finalmente, as considerações versam acerca das adaptações que podem acontecer na governança, de acordo com a trajetória de desenvolvimento dessas comunidades.

### 3.2.3.1 Mecanismos estruturais em projetos de *software* livre comunitários

De acordo com Lee e Cole (2003), as comunidades de *software* livre possuem uma estrutura formada por duas camadas: o núcleo, que é composto pela liderança e pelos desenvolvedores e outros mantenedores do projeto; uma região periférica, na qual se encontram os demais desenvolvedores e usuários do *software*. Enquanto as fronteiras periféricas são abertas e possibilitam a adesão livre e informal de qualquer indivíduo, os níveis internos são mais fechados e menos permeáveis, conforme o indivíduo move-se em

direção ao centro do projeto (Seidel & Stewart, 2011). Tal característica reside no fato de ser no núcleo do projeto onde reside uma maior autoridade de decisão, o que impõe aos membros periféricos a exigência de ganhar mais experiência para ter acesso a esse espaço (Von Krogh et al., 2003; Fang & Neufeld, 2009). Seidel e Stewart (2011), ao comparar a estrutura formatada em comunidade com a de outros desenhos organizacionais, identificam que embora haja a possibilidade de algumas características semelhantes entre ambos os casos, os negócios comunitários como os de *software* livre têm particularidades que as tornam um fenômeno único. Em relação às organizações funcionais, as comunidades possuem uma estrutura de trabalho horizontalizada e sem uma clara divisão de funções. Em comparação com as organizações matriciais, suas cadeias de comando são mais informais. Em contraponto com as organizações em rede, os relacionamentos existentes não estão baseados em contratos.

Como já se deu a entender, uma das características essenciais das comunidades de *software* livre refere-se ao fato desses agrupamentos se diferenciarem das organizações tradicionais pela ausência de autoridade central e de estruturação hierárquica (Lee & Cole, 2003; Panchal, 2010). Assim, ao invés de estarem ancoradas em relações verticais de subordinação, nas quais as instâncias inferiores da organização subordinam-se às superiores (Demil & Lecocq, 2006), adotam uma perspectiva horizontalizada, formada por redes de práticas compartilhadas (Brown & Duguid, 1991; Wenger, 2000), onde a autoridade dos seus integrantes acontece de forma lateral (Dahlander & O'Mahony, 2011). O exercício lateral de autoridade faz com que a forma de progressão no coletivo seja diferente da noção tradicional, uma vez que os indivíduos não se deslocam no sentido vertical, mas se movimentam da periferia em direção ao centro do projeto (Dahlander & O'Mahony, 2011).

Na progressão do tipo lateral, os indivíduos da comunidade de *software* livre ganham maior responsabilidade e direitos de decisão em mais áreas do projeto e em relação ao trabalho coletivo desenvolvido, sem necessariamente ter que gerenciar o desempenho e a carreira de outros integrantes (Lingo & O'Mahony, 2010; Dahlander & O'Mahony, 2011). São os membros mais ativos na oferta de contribuições técnicas ao código do projeto e na resolução de problemas complexos os candidatos mais prováveis para assumir maiores responsabilidades nas comunidades de *software* livre. Esses membros são vistos pelos outros participantes como sendo conhecedores do projeto e tomadores de decisão mais confiáveis (Von Hippel & Von Krogh, 2003; Dahlander & O'Mahony, 2011). Contudo, segundo Dahlander e O'Mahony (2011), apesar da progressão lateral acontecer em decorrência do conhecimento técnico, na proporção em que os participantes ganham autoridade no trabalho coletivo, passam também a se envolver com outras funções relacionadas ao controle gerencial

do projeto (Fleming & Waguespack, 2007). Isso ocorre principalmente quando os projetos de *software* livre se tornam mais complexos, já que os módulos ou subprojetos existentes precisam ser gerenciados e integrados (Panchal, 2010; Dahlander & O'Mahony, 2011; Lindberg et al., 2016). Nos sistemas horizontais, como é o caso das comunidades de *software* livre, o termo colaboração em vez da denominação comando é a chave para a realização do trabalho coletivo (Zabusky, 1997).

No que concerne à especialização de papéis, quatro grupos gerais de responsabilidades podem ser identificados nas comunidades de *software* livre: a liderança do projeto, representada geralmente por seus fundadores; os desenvolvedores do núcleo, que atuam no desenvolvimento e manutenção do *software*, ao adicionar funcionalidades e corrigir erros; os desenvolvedores periféricos, que contribuem com o código, reportam erros e sugerem funcionalidades; os usuários, que são aqueles que apenas utilizam o *software* (Ye & Kishida, 2003; Xu et al., 2005). No entanto, a despeito da distribuição mais ampla de responsabilidades, os autores que versam sobre esse assunto destacam que embora os projetos de *software* livre tenham uma estrutura de papéis que é pouco definida (Lee & Cole, 2003; Arazy et al., 2016), contam com a significativa atuação da liderança nas principais ações do coletivo (Seidel & Stewart, 2011).

Na prática, a execução de papéis pelos indivíduos das comunidades de *software* livre tem um perfil emergente e pouco especializado, uma vez que acontece no próprio processo de realização do trabalho colaborativo, não existindo uma designação ou planejamento previamente estabelecidos (Lee & Cole, 2003; Zammuto, Griffith, Majchrzak, & Dougherty, 2007; Arazy et al., 2016). Desse modo, é comum que ocorram sobreposições no desempenho de tarefas, a exemplo do time de desenvolvedores que pode desempenhar diversas funções tanto no núcleo, quanto na periferia da comunidade (Lee & Cole, 2003; Seidel & Stewart, 2011). É também em decorrência dessa sobreposição de papéis, fato que explica a participação dos contribuintes em múltiplas funções do projeto, que se encontra a dificuldade para obtenção de uma visão formalizada e rígida da cadeia de comando nas estruturas comunitárias de *software* livre (Seidel & Stewart, 2011). Contudo, mesmo diante da pouca formalização dos papéis executados, a estabilidade desses papéis emergentes é alcançada, em virtude do processo produtivo do *software* livre ser centrado na coprodução do artefato e pelos ajustes mútuos que acontecem de acordo com as necessidades desse artefato (Arazy et al., 2016).

Mesmo que os projetos de *software* livre não tenham funções deliberadamente especificadas, o exercício da liderança comumente se apresenta de modo bem definido (Lee &



Cole, 2003), atuando como condicionante fundamental para o sucesso desses projetos (Xu et al., 2005; Forte & Lampe, 2013). É a liderança que assume a responsabilidade pelos vários subsistemas que o projeto abrange e que toma as principais decisões referentes ao código-fonte e às escolhas de qual código deve ser incluído no código-fonte para o lançamento oficial do *software* (Lee & Cole, 2003). A atividade da liderança, portanto, é decisiva na governança, no sentido de promover a viabilidade e o crescimento dos coletivos de *software* livre (Xu et al., 2005), de forma que os seus representantes constituem figura protagonista na estrutura das comunidades (Seidel & Stewart, 2011).

Em relação ao processo decisório, a literatura sobre os arranjos comunitários, em especial os de *software* livre, traz diferentes enfoques, o que demonstra a ausência de um pensamento convergente neste assunto. Paradoxalmente, ao mesmo tempo em que se reconhece que as decisões relevantes e de maior impacto estão concentradas na autoridade do coletivo (Lee & Cole, 2003; West & O'Mahony, 2005; Seidel & Stewart, 2011), sugere-se que os integrantes dessas comunidades dispõem de maior independência na realização de suas atividades (Seidel & Stewart, 2011). Isso acontece não só pela distância física existente entre participantes, mas também pelo fato de as comunidades de *software* livre constituírem agrupamentos não hierárquicos e massivamente formados por voluntários (Seidel & Stewart, 2011), os quais podem tomar decisões sobre as atividades que participam em resposta aos seus interesses e objetivos (Panchal, 2010). A própria possibilidade de as atividades produtivas dos projetos de *software* livre acontecerem em subprojetos ou módulos (West & O'Mahony, 2005; Lindberg et al., 2016) também permite uma maior liberdade na atuação do desenvolvimento do *software*, o que faz com que o trabalho seja desempenhado em uma escala mais descentralizada do que acontece nas organizações formais (O'Mahony & Ferraro, 2007; Chen & O'Mahony, 2009; Jensen & Scacchi, 2010; Panchal, 2010).

Em síntese, embora as comunidades de *software* livre possam centralizar as decisões mais importantes, referentes ao desenho, implementação e priorização do *software* (Lee & Cole, 2003), a ideia principal trazida pela literatura é que esses projetos tenham o perfil de coletivos auto-organizados, nos quais são os próprios integrantes que realizam o gerenciamento do trabalho que executam (Panchal, 2010; Jensen, 2010; Fjeldstad, Snow, Miles, & Lettl, 2012).

### 3.2.3.2 Mecanismos processuais de controle em projetos de *software* livre comunitários

Ainda que elementos do controle formal possam ser notados no âmbito das comunidades de *software* livre (O'Mahony & Ferraro, 2007), o entendimento mais comum entre estudiosos da matéria é que elas se destacam, em sua essência, por apresentar maior informalidade ou pouca estruturação também neste aspecto (Lattemann & Stieglitz, 2005; Xu et al., 2005; Jensen & Scacchi, 2010; Lindberg et al., 2016; Kolbjørnsrud, 2016). Acredita-se que a eficácia de mecanismos formais nos sistemas organizados em formato comunitário e baseados principalmente no trabalho voluntário seja menor, uma vez que não é possível a penalização ou recompensa financeira e não existe uma proteção contratual, dependência formal ou existencial ao projeto (Lattemann & Stieglitz, 2005).

O uso de processos informais de controle nas comunidades de *software* livre deve-se não apenas à distância geográfica dos seus integrantes, em grande maioria, voluntários que se comunicam com o suporte das TICs, mas também às diferentes motivações que levam os indivíduos a ingressarem nesses coletivos (Lattemann & Stieglitz, 2005). Por constituírem empreendimentos caracterizados pela heterogeneidade da composição dos seus integrantes, nele subsiste a necessidade de alinhamento dos comportamentos individuais com os objetivos do projeto, para que seja possível atingir os resultados pretendidos (Lattemann & Stieglitz, 2005). Nesse sentido, sobressaem nos projetos de *software* livre o emprego de procedimentos informais de gestão representados, principalmente, pelo controle social, controle entre pares e autocontrole (Lee & Cole, 2003; Xu et al., 2005; Kolbjørnsrud, 2016).

O controle social nos projetos de *software* livre busca fortalecer o sentimento de comunidade com base na socialização intensiva e em pressupostos culturais (Xu et al., 2005; Lattemann & Stieglitz, 2005). Esse controle envolve o compartilhamento de objetivos e valores com a intenção de alcançar uma orientação ética e comportamental dos indivíduos, bem como o desenvolvimento de um senso comum de pertencimento e de identificação com a comunidade (Xu Xu et al., 2005; Lattemann & Stieglitz, 2005). Na implementação do controle social, diferentes instrumentos de governança são introduzidos para ativação de culturas comuns, criando conceitos morais semelhantes, mediante rituais ou cerimônias, tais como encontros regulares entre os membros da comunidade, diretrizes voltadas aos comportamentos operacionais, intenso emprego de ferramentas de TI, etc (Lattemann & Stieglitz, 2005). A reputação também é um aspecto relevante do controle social, já que permite o entendimento de quais comportamentos são aceitáveis ou valorizados no projeto (Xu et al., 2005). Os líderes ou gerentes geralmente são portadores de boa reputação e vistos

como pessoas que trazem contribuições relevantes para o projeto; enquanto os participantes contraproducentes estão expostos a sofrer sanções como o julgamento por parte dos seus pares, a perda de reputação ou até a exclusão do coletivo (Lattemann & Stieglitz, 2005).

O controle social nos projetos de *software* livre está ancorado em práticas coletivas desenvolvidas por meio das redes de relacionamentos mantidas por seus integrantes (Wenger, 1998). Essas práticas conectam os integrantes das comunidades e são resultado do engajamento dos indivíduos em ações cujos significados eles negociam uns com os outros (Wenger, 1998). O conjunto de práticas possibilita a formação de um repertório compartilhado de atitudes e procedimentos, o qual inclui elementos como rotinas, ferramentas, modos de fazer as coisas, histórias, gestos, símbolos, ações, ou conceitos, que as comunidades produzem ou adotam no curso de sua existência (Wenger, 1998; Seidel & O'Mahony, 2014). A criação do repertório compartilhado de práticas leva à necessidade de seu gerenciamento, uma vez que ele deve destinar-se a criar um entendimento comum das atividades desenvolvidas e a orientar os comportamentos individuais rumo aos resultados pretendidos (Seidel & O'Mahony, 2014). Além de servir como um mecanismo orientador de ações nos grupos de trabalho, esse repertório define os limites de pertencimento aos arranjos comunitários e influi na criação da identidade do coletivo (Wenger, 1998). Tendo como alicerce o controle social, outras duas formas de controle verificadas nos projetos de *software* livre são o autocontrole e o controle entre pares.

Em relação ao autocontrole, as próprias plataformas tecnológicas que sustentam as práticas das comunidades, ao permitirem a modularização do código de *software* e contribuições individuais com altos níveis de colaboração, enfatizam a instituição da autonomia e da independência na realização do trabalho e reduzem os esforços de coordenação (Crowston, Wei, Howison, & Wiggins, 2012; Howison & Crowston 2014; Lindberg et al., 2016). A visão de que o trabalho nos projetos de *software* livre acontece de forma auto-organizada, também ressalta a ocorrência do autocontrole nos coletivos, significando a certificação de que são os próprios membros dos projetos que selecionam como, quando e em que gostariam de trabalhar (Dahlander, Frederiksen, & Rullani, 2008; Arazy et al., 2016). Uma vez que o processo de desenvolvimento de *software* livre não é algo rotineiro, o autocontrole é algo fundamental e a liderança do projeto tem um papel essencial para estabelecê-lo, inclusive no momento da seleção de membros e do trabalho a ser executado por eles (Xu et al., 2005). Os antecedentes organizacionais do autocontrole envolvem a complexidade da tarefa, a avaliação de desempenho ambígua e a falta de regras e procedimentos para realização das atividades (Xu et al., 2005).

O controle entre pares é uma alternativa de se garantir a qualidade das aplicações produzidas e os seus mecanismos são definidores da governança colaborativa nas comunidades de *software* livre (Lattemann & Stieglitz, 2005; Kolbjørnsrud, 2016). É o papel central conferido ao artefato, no caso o *software*, que permite a existência de coordenação tácita entre os pares nas atividades produtivas das comunidades (Howison & Crowston 2014; Arazy et al., 2016). O principal objetivo do controle entre pares é possibilitar que outras pessoas revisem o trabalho realizado por determinado indivíduo, tendo como pressuposto a crítica do trabalho alheio para corrigir erros, resolver problemas, bem como promover melhorias e trocas de informações e conhecimentos em relação a esse trabalho (Lee & Cole, 2003). Esta modalidade de monitoramento mútuo reduz a assimetria de informações (Child & Rodrigues, 2003; Kolbjørnsrud, 2016), possibilita o reconhecimento do trabalho desenvolvido entre os pares, e se propõe a atuar como fonte de motivação, recompensa e de promoção do status social nos projetos (Lakhani & Von Hippel, 2003; Stewart, 2005; Kolbjørnsrud, 2016).

Apesar dos processos informais de controle assumirem um maior destaque nas comunidades de *software* livre, isso não anula a possibilidade da aplicação de procedimentos formais no gerenciamento dos coletivos, nem sequer a coexistência das duas perspectivas de controle (formal e informal) (O'Mahony & Ferraro, 2007; Panchal, 2010). No entanto, acredita-se que a formalização verificada nos arranjos comunitários seja diferente dos controles hierárquicos observados em organizações tradicionais (Panchal, 2010).

Espera-se que as comunidades de *software* livre sustentem as suas atividades em processos de controle que tenham um perfil mais democrático, com o propósito de aumentar as oportunidades de participação, a transparência das ações empreendidas e a responsabilidade dos seus tomadores de decisão (Lattemann & Stieglitz, 2005; O'Mahony & Ferraro, 2007; Shaikh & Vaast, 2016). São esses os atributos que tornam comum o estabelecimento de decisões pautadas no consenso e na participação efetiva dos membros do coletivo (Sharma et al., 2002; Panchal, 2010). A governança democrática, por sinal, pode ser mais eficaz para motivar contribuições e atrair novos participantes para as comunidades, do que as recompensas formais e os benefícios privados (Shaikh & Vaast, 2016). Além dessa abertura democrática, a meritocracia também é um atributo comumente observado nas comunidades de *software* livre e que influencia bastante nos mecanismos de controle, já que o seu emprego tem o potencial de reduzir conflitos e políticas desnecessárias dentro desses agrupamentos (Shaikh & Vaast, 2016). Valendo-se da meritocracia, normalmente os líderes e gestores dos negócios de *software* livre tornam-se integrantes que adquirem uma reputação

reconhecida, principalmente em razão dos conhecimentos e das contribuições que trazem para o projeto (Lattemann & Stieglitz, 2005; O'Mahony & Ferraro, 2007).

### 3.2.3.3 Desenvolvimento das comunidades de *software* livre

As características gerais e de governança das comunidades, em especial as das comunidades de *software* livre, necessitam serem contextualizadas, uma vez que, tal como reconhecem Faraj et al. (2011), os arranjos comunitários são fluidos e estão constantemente em modificação. Apesar de ser possível descrever os atributos dos empreendimentos comunitários em sua forma orgânica, baseada em um funcionamento que transcende determinada organização (Wynn, 2003; West & O'Mahony, 2005), a simples admissão de que as iniciativas de *software* livre passam por sucessivas transformações conforme aumentam de complexidade e se desenvolvem, justifica a atitude de contextualizá-las na sua forma de funcionamento e em razão dos progressos alcançados (Lattemann & Stieglitz, 2005).

As mudanças decorrentes do crescimento dos projetos de *software* livre atingem diretamente vários dos seus aspectos de governança, a exemplo da estrutura e dos processos de controle adotados (Lattemann & Stieglitz, 2005; Xu et al., 2005). Diferentes níveis de desenvolvimento fazem com que convivam no espaço virtual diversos tipos de comunidades de *software* livre, abarcando desde arranjos menores e com poucos integrantes, além dos seus fundadores iniciais, até os empreendimentos mais maduros, que podem contar, inclusive, com a participação de outras organizações em seu funcionamento (Stewart & Gosain, 2006; Seidel & Stewart, 2011). Nesse último caso, incluem-se projetos de grande sucesso como o Linux, Debian, Apache, Gnome, dentre outros, que são amplamente citados e estudados nas pesquisas sobre as comunidades de *software* livre (West & O'Mahony, 2005; David & Shapiro, 2008; Chen & O'Mahony, 2009).

Algumas abordagens do ciclo de vida de projetos voltadas, especificamente, para as comunidades de *software* livre são encontradas na literatura do tema e explicam as transformações pelas quais as características organizacionais dessas comunidades passam no decorrer do tempo e em função do seu amadurecimento (Wynn, 2003; Lattemann & Stieglitz, 2005; Xu et al., 2005). Nessas pesquisas, as seguintes fases de desenvolvimento são identificadas:

- **Introdução** ó Um desenvolvedor ou um grupo pequeno de desenvolvedores identificam determinada necessidade de *software* e trabalham para produzir uma versão inicial do programa. Esses fundadores garantem a infraestrutura tecnológica

básica para que outros usuários tenham acesso à aplicação, de forma a atrair novos integrantes. A estrutura e o controle da comunidade se apresentam informais e a liderança do projeto atua basicamente no direcionamento dos esforços da equipe e comunicando os objetivos do projeto.

- **Crescimento** ó O projeto cresce em número de participantes, os quais podem atuar tanto em áreas periféricas, quanto em áreas internas, como desenvolvedores, e assim contribuindo mais efetivamente com o código. O projeto apresenta alguma formalização de sua estrutura e dos processos de controle e conta com uma liderança e equipe de gestão que avalia a inclusão das contribuições dos integrantes no lançamento do *software*. Tais representantes atuam como aceleradores do projeto e empregam as tecnologias voltadas ao gerenciamento da comunidade.
- **Maturidade** ó O número de integrantes do projeto cresce para um tamanho máximo. A liderança e a equipe de gestão atuam mais na avaliação do código ou no gerenciamento do projeto do que no desenvolvimento do *software* em si. O produto do projeto apresenta múltiplas versões e lançamentos. A liderança e a equipe de gestão têm como foco a sustentação do projeto, o que faz com que a estrutura e os processos de controle da comunidade passem a ser mais formalizados e rígidos. No entanto, apesar de existir maior formalização, essas comunidades permanecem com configurações flexíveis e buscam evitar o controle burocrático.
- **Declínio (ou reavivamento)** ó O projeto entra em um estágio final marcado pela diminuição do quantitativo de integrantes e menor número de *downloads* do *software*. O foco da liderança e da equipe de gestão é no suporte e na manutenção da funcionalidade do *software* existente. No entanto, em alguns casos, essa comunidade pode ter um reavivamento em resposta a uma nova versão do *software*, mudanças no mercado ou novas descobertas dos desenvolvedores, de sorte que o projeto entra em um novo estágio de crescimento ou maturidade.

Diante do exposto, apesar de autores como Raymond (1999) afirmarem que os projetos de *software* livre funcionam numa lógica próxima a de um *õbazarö*, na qual os participantes são voluntários e se ordenam de forma não hierárquica e informal, em oposição à visão catedral de organização, que é hierárquica e formal, é possível notar que tais empreendimentos são dinâmicos e que podem sofrer diversas alterações ou combinações dessas fases ao longo do tempo. Ainda que a literatura possa trazer uma visão que contrapõe os projetos de *software* livre comunitários a outros formatos organizacionais mais tradicionais (Seidel & Stewart, 2011), acredita-se, na medida em que esses projetos se desenvolvem, que a

tendência é a de suas estruturas e processos de controle tornarem-se mais formalizados (Lattemann & Stieglitz, 2005; Xu et al., 2005; Panchal, 2010; Forte & Lampe, 2013). Em outras palavras, as comunidades saem de uma conjuntura que é informal e emergente (Arazy et al., 2016) e passam a incorporar alguns elementos formais na sua configuração (O'Mahony & Ferraro, 2007).

No entanto, ainda que a maturidade possibilite a maior formalização das relações de autoridade, dos papéis desempenhados, da estrutura decisória e dos processos de controle verificados nas comunidades de *software* livre (Wynn, 2003; Lattemann & Stieglitz, 2005; Panchal, 2010), segundo os autores estudiosos da matéria, espera-se que essas comunidades permaneçam com traços únicos em que sobressaem o formato flexível e a informalidade na gestão das relações de trabalho (Xu et al., 2005; Seidel & Stewart, 2011; Faraj et al., 2011). Tal fato decorre, principalmente, pelo fato dos projetos de *software* livre serem adeptos da livre troca de conhecimentos e abrirem espaço para um grande quantitativo de voluntários, geograficamente dispersos, o que permite maior liberdade de atuação e amplificação de ações colaborativas mediadas pelas TICs (Demil & Lecocq, 2006; Seidel & Stewart, 2011).

Em relação à centralização das decisões, a literatura sobre os projetos de *software* livre traz um entendimento ainda não muito consolidado, pois ao mesmo tempo em que se espera que esses projetos fiquem mais formalizados com o seu crescimento, se preconiza que eles passem a apresentar, no seu amadurecimento, uma maior delegação das responsabilidades e descentralização das atividades dos seus integrantes (Wynn, 2003; Lattemann & Stieglitz, 2005; Panchal, 2010). Essa visão de que são os próprios membros dos coletivos que fazem a gestão de suas respectivas tarefas, atuando concomitantemente de forma autônoma e interdependente, perfaz uma característica do modelo comunitário que merece ser mais bem compreendida no contexto geral do processo de formalização desses tipos de projetos (Seidel & Stewart, 2011; Snow, Fjeldstad, & Lettl, 2011).

Outro aspecto relacionado ao nível de maturidade dos negócios de *software* livre e que tem impacto significativo na formalização da sua estrutura e dos processos de controle empregados, refere-se à presença de outras organizações nas atividades das comunidades (West & O'Mahony, 2005; Li-Ying & Solomo, 2013). É comum a constatação de que alguns projetos que surgiram de forma autônoma transformem-se e passem a ser geridos ou apoiados por empresas ou por organizações sem fins lucrativos (Li-Ying & Solomo, 2013), como é o caso da entrada de patrocinadores (West & O'Mahony, 2005) ou fundações (Kelty, 2008; Germonprez et al., 2016). Nessas situações, podem ocorrer mudanças no curso de funcionamento e a geração de certa tensão no gerenciamento dos esforços coletivos para o

desenvolvimento do *software* (West & O'Mahony, 2005). Tal fato acontece porque o projeto passa a ter uma organização formal em seu contexto, com funcionários que são remunerados por essa organização, ao mesmo tempo em que tenta permanecer com uma filosofia comunitária de origem, na qual conta com o engajamento de voluntários sem contrato e nem empregador comum (West & O'Mahony, 2005; Shaikh & Vaast, 2016).

O funcionamento mais formal das iniciativas comunitárias de *software* livre é aspecto que também suscita melhor entendimento e que tem levado os pesquisadores a questionarem algumas das ideias aceitas sobre como essas comunidades funcionam e organizam as suas atividades (Shaikh & Vaast, 2016). Pois, ao mesmo tempo em que o formato comunitário ressalta as características informais no gerenciamento dos coletivos, com estrutura e processos de controle pouco desenvolvidos, contrastando com as organizações hierárquicas e tradicionais (Arazy et al., 2016; Bauer et al., 2016), existem correntes de pensamento na literatura que abordam a questão da formalização das iniciativas comunitárias (Wynn, 2003; Lattemann & Stieglitz, 2005; Xu et al., 2005; Shaikh & Vaast, 2016). Panchal (2010) destaca que, em alguns cenários, é possível a utilização de uma abordagem híbrida nos projetos de *software* livre, combinando aspectos informais da auto-organização com processos estruturados de gestão. Assim, podem existir os mais diversos tipos de variações entre as comunidades de *software* livre, no sentido de aceitar mais ou menos novos membros, serem altamente estruturadas e controladas ou anarquistas, bem como terem diversas regras de engajamento (Germonprez et al., 2016).

#### 3.2.4 Aporte teórico da tese

O aporte da teoria organizacional escolhido para esta tese foi o a abordagem dos *stakeholders*, a teoria da dependência de recursos e as organizações temporárias. As teorias, de forma geral, buscam trazer respostas às necessidades dos indivíduos de dar sentido ao mundo em que se encontram e de obter um conjunto de conhecimentos, com vistas a compreender, explicar e prever determinado contexto (Gregor, 2002). No entendimento de Popper (2005), as teorias constituem afirmações universais que têm como propósito racionalizar, explicar e dominar o mundo para capturar o seu significado. Após a apresentação do aporte teórico, considerando também o conjunto de conceitos trazidos anteriormente, apresenta-se a seção que desenvolve o argumento central desta pesquisa.



### 3.2.4.1 Abordagem dos *stakeholders*

De acordo com Freeman (1984), denominam-se *stakeholders* os indivíduos ou grupos de indivíduos que têm a capacidade de influenciar o desempenho de certas organizações, inclusive de serem também afetados pelos procedimentos de gestão e objetivos destas entidades. O conceito de *stakeholders*, desde a sua primeira apresentação no livro *Strategic Management: A Stakeholder Approach*, do autor supracitado, vem recebendo amplo uso em estudos organizacionais, aparecendo em tais pesquisas por meio de diferentes definições e formas de mensuração avaliativa, como as que são vistas a seguir.

Donaldson e Preston (1995) apontam que a abordagem dos *stakeholders* divide-se em três dimensões analíticas principais: a descritiva, a instrumental e a normativa. A perspectiva descritiva tem como proposta relatar e, por vezes, explicar os comportamentos e as características específicas da organização frente aos *stakeholders*, entendendo-a como uma constelação de interesses cooperativos e competitivos. A visão instrumental enfatiza as conexões do gerenciamento da organização, tanto na busca dos seus objetivos, quanto em relação aos *stakeholders*. Esses estudos instrumentais, a partir do uso de metodologias variadas, baseiam-se no pressuposto de que a aderência das estratégias e políticas implementadas às expectativas dos *stakeholders* permite um melhor desempenho da organização comparativamente aos seus concorrentes. A dimensão normativa, por fim, busca trazer inferências sobre a natureza das relações entre os *stakeholders* e a organização, incluindo parâmetros morais e filosóficos. Vale mencionar que essas três dimensões nem sempre aparecem nas pesquisas de forma isolada, mas, em vez disso, frequentemente combinadas.

Um dos modelos de análise de *stakeholders*, formulado por Mitchell, Agle e Wood (1997), aborda a questão da mensuração do conceito por meio de uma classificação a partir de três variáveis: *poder*, *legitimidade* e *urgência*. Quanto mais dessas variáveis o *stakeholder* possuir, maior relevância ele terá para a organização. Assim, para separar estes influenciadores, utiliza-se uma divisão baseada nos conceitos de *definitivos* (apresentam as três características), *especiais* (que apresentam duas características) e *latentes* (apresenta apenas uma característica). Já Hill e Jones (1998) separam os *stakeholders* entre *internos* e *externos*, ambos detendo forte influência nas decisões estratégicas da organização. Os *stakeholders* internos são os acionistas e empregados, incluindo os executivos, gerentes e membros do conselho da empresa. Os *stakeholders* externos são todos os outros indivíduos ou grupos que têm algum tipo de interesse na empresa. Tipicamente, este último grupo

compreende consumidores, fornecedores, governos, sindicatos, comunidades locais e o público em geral (Hill & Jones, 1998; Pinto & Oliveira, 2004).

Rowley (1997), por sua vez, aponta, ao contrário do que se possa pensar, que as firmas não respondem a cada ator individualmente, mas sim por meio de uma interação múltipla com vários *stakeholders*. Essa interação conta com a participação de atores ou grupos que, além de possuírem ligação com a organização, possuem também relações entre si. Utiliza-se então uma perspectiva de rede social no estudo de *stakeholders*, em que são levados em conta os relacionamentos entre os indivíduos nos moldes de estrutura. O autor aponta, ainda, que a própria organização focal não é simplesmente o ponto central de seus próprios *stakeholders*; na verdade, ela é também um *stakeholder* diferenciado de muito outros pontos focais dentro do seu sistema social.

Apesar da abordagem dos *stakeholders* ter sido amplamente empregada nas pesquisas envolvendo corporações empresariais privadas, cada vez mais ela vem sendo utilizada também em análises de organizações públicas e sem fins lucrativos (Bryson, 2018). Isso decorre, em especial, pela preocupação dessas organizações em alocar recursos e atender com maior eficiência os indivíduos ou grupos com os quais mantêm contatos, em certos casos sob a égide das leis. Além disso, é comum que a abordagem dos *stakeholders* seja usada em conjunto com outras teorias organizacionais, como a teoria institucional ou a teoria da dependência de recursos. O principal ponto em que essas duas teorias pontificam como recursos auxiliares, cada uma dentro do seu campo de especificidades, reside no fato de ambas preconizarem que as organizações necessitam estar atentas às demandas e expectativas externas, caso queiram sobreviver no ambiente em que se encontram (Oliver, 1991; Frooman, 1999; Rossetto & Rossetto, 2005). Enquanto a perspectiva institucional apresenta uma visão mais conformista e passiva da organização com relação ao ambiente em que se encontra, a visão da teoria da dependência de recursos defende um pensamento mais voluntarista e proativo (Rossetto & Rossetto, 2005).

Independente do contexto ou da teoria que seja alocada à abordagem dos *stakeholders*, o essencial é que a organização seja compreendida como um agrupamento de atores voltados a problemas comuns, de modo que os seus propósitos devem ater-se a gerenciar interesses, necessidades e pontos de vista nem sempre convergentes (Friedman & Miles, 2006). Tudo isso deve ser percebido como uma via de mão dupla, pois os *stakeholders* igualmente podem influenciar nas estratégias da organização, o que faz com que essa abordagem retrate também a questão da gestão de contradições e conflitos, potenciais ou

efetivos, entre indivíduos e/ou grupos, buscando solucioná-los (Frooman, 1999; Friedman & Miles, 2006).

#### 3.2.4.2 Teoria da dependência de recursos

A teoria da dependência de recursos (TDR) tem influência significativa que se estende especialmente sobre as disciplinas da sociologia, psicologia e gestão, abrangendo diferentes tipos e domínios organizacionais (Davis & Cobb, 2009). A obra seminal e histórica de Pfeffer e Salancik, de 1978, *The External Control of Organizational: a resource dependence*, consolidou as principais ideias expressas nessa teoria, que se tornou uma das mais influentes nos estudos organizacionais e na gestão estratégica (Hillman, Withers, & Collins, 2009; Dress & Heugens, 2013).

Segundo Pfeffer e Salancik (2003), para que uma organização possa ser compreendida, o contexto no qual ela funciona deve ser analisado. Partindo desse pressuposto, a TDR considera que os fatores externos exercem influência sobre o comportamento das organizações, que se caracterizam por ser um sistema aberto (Pfeffer & Salancik, 2003). Contudo, ainda que reconheça a inter-relação entre a organização e o contexto, a TDR sobressai por trazer uma concepção mais pró-ativa do empreendimento, já que os gestores, por meio de diversas táticas, também podem atuar no sentido de reduzir as incertezas, os riscos e as dependências ambientais (Hillman et al., 2009).

Visto que existe essa interdependência das organizações e dos ambientes em que funcionam, para conseguir atender as demandas existentes elas podem realizar reconfigurações da sua estrutura interna, assim como lançar mão de diversas estratégias, de forma a garantir os recursos que necessitam para sobreviver e serem competitivas (Pahlevan Sharif & Kyid Yeoh, 2014). Esse papel de prover e garantir os recursos relevantes normalmente recai sobre a figura do responsável pela administração das organizações (Pahlevan Sharif & Kyid Yeoh, 2014). Os gestores são os responsáveis pelas denominadas escolhas estratégicas, que envolvem não apenas o estabelecimento das formas estruturais, mas também a manipulação das características ambientais e a escolha de padrões de desempenho (Child, 1972).

Dentre as diversas formas pelas quais as organizações podem responder às dependências de recursos, encontram-se os processos de integração vertical ou horizontal, que envolvem arranjos tais como interconexões, aquisições, fusões, alianças ou acordos organizacionais, etc (Pfeffer & Salancik, 2003; Davis & Cobb, 2009; Drees & Heugens,

2013). A ampla gama de ações formais ou informais desenvolvidas pelos gestores têm o propósito principal de garantir os recursos essenciais, aumentar o poder nas relações de troca e diminuir as subordinações e as incertezas que os empreendimentos enfrentam em relação aos demais (Pfeffer & Salancik, 2003). A ênfase no poder e a articulação de um repertório de práticas e estratégias são características que distinguem a TDR de outras teorias e abordagens (Davis & Cobb, 2009). Davis e Cobb (2009), ao exemplificar como o poder e as dependências se manifestam na TDR, fazem uso de raciocínio estruturado à partir de proposições, a saber: B depende de A na medida em que A tem poder sobre B; além disso, a soma do poder pode não ser zero, já que quando A e B têm poder um sobre o outro, eles se tornam interdependentes. É comum que as organizações tentem reduzir o poder dos outros sobre ela, significando buscar aumentar o seu próprio poder sobre os outros (Hillman et al., 2009).

Pfeffer (1987) esclarece que um dos argumentos básicos da TDR é que as organizações não são autônomas, ou seja, elas são restringidas por uma série de interdependências com outras organizações, o que faz com que ações com resultados incertos e voltadas para a sobrevivência precisem ser gerenciadas. Em suma, três ideias são centrais à TDR: o contexto social é relevante; as organizações precisam ter estratégias para buscar maior autonomia e atender os seus interesses; o poder é algo importante para entender as ações internas e externas das organizações (Davis & Cobb, 2009). Apesar da TDR existir há mais de 30 anos, evidências demonstram que o interesse por essa teoria permanece até os dias atuais (Davis & Cobb, 2009). Mesmo que estudos como os de Casciaro e Piskorski (2005) considerem a TDR mais uma metáfora atraente do que uma teoria que permita a realização de investigação empírica testável, na presente tese, acredita-se que ela é uma perspectiva viável para entender os empreendimentos colocados em pauta e suas relações ambientais (Hillman et al., 2009; Dress & Heugens, 2013).

Apresentadas as percepções centrais da abordagem dos *stakeholders* e da TDR, nota-se que ambas as visões convergem ao reconhecerem as influências mútuas existentes entre as organizações e os atores internos e externos que existem e gravitam em torno delas, dentro do ambiente em que se encontram. Desse modo, a partir da combinação dessas duas visões como lente de análise, a presente tese propõe-se ao objetivo de estudar a governança de projetos de *software* livre que apresentam formato comunitário, considerando as transformações que eles sofrem no decorrer do tempo.

### 3.2.4.3 Organizações temporárias

Os projetos de *software* livre assumidos por este trabalho de tese foram aqueles que formaram uma comunidade virtual em torno da produção do *software* livre. No entanto, apesar dessa escolha, a investigação não se deteve apenas no funcionamento da comunidade em si, uma vez que todo o contexto e a história de formação do coletivo foram considerados. Os empreendimentos formados pelos projetos foram examinados como arranjos voltados para o alcance de resultados previamente definidos, o que permite uma associação desses arranjos ao conceito de organizações temporárias, apresentado a seguir.

As organizações temporárias (OTs) são representadas por grupos de atores interdependentes, temporalmente limitados, formados para a realização de uma tarefa complexa (Burke & Morley, 2016). Geralmente, as definições sobre o conceito convergem ao afirmar que as OTs possuem um ponto de terminação determinado anteriormente, fixado por uma data específica ou pela obtenção de um estado ou condição previamente definidos (Bakker et al., 2009; Bakker, 2011). Esse período temporal de atividades das OTs pode alternar-se entre o curto e o longo prazo (Bakker, 2011).

À parte a questão da temporalidade, as OTs são estruturas que têm forte orientação para tarefas (Bakker, 2011), em especial para aquelas que se caracterizam pela complexidade decorrente das interdependências, pela ambiguidade de interesses e pela incerteza dos resultados (Bakker, 2011; Burke & Morley, 2016). Devido a esses traços, as OTs carecem da coordenação dos trabalhos desenvolvidos em equipe, sendo o papel do líder visto como crucial (Bakker, 2011). O exercício da liderança nos esforços de coordenação tende a possuir um perfil participativo, com uma atuação que se volta especialmente para o gerenciamento da geração de ideias e de tomadas de decisões, que são atividades que se revezam continuamente no âmbito das OTs (Burke & Morley, 2016).

As OTs estão intrinsecamente relacionadas ao ambiente externo, representado pelo contexto social e organizacional em que atuam, uma vez que é nesse ambiente onde se localizam os principais recursos necessários à sobrevivência delas (Grabher, 2004). Esse entrelaçamento ao contexto pode se tornar um dilema para as OTs, pois, ao mesmo tempo em que essas estruturas buscam autonomia, elas também precisam adquirir os recursos necessários ao seu funcionamento e competitividade (Burke & Morley, 2016). O contexto externo, convém destacar, envolve não só o sistema organizacional no qual as OTs estão incorporadas, mas também o meio social mais amplo, que abrange relações com outras organizações inseridas no ambiente (Grabher, 2004). Essa interação com o ambiente, em

conjunto com a questão da temporalidade, são alguns dos atributos mais importantes das OTs (Sydow et al., 2004).

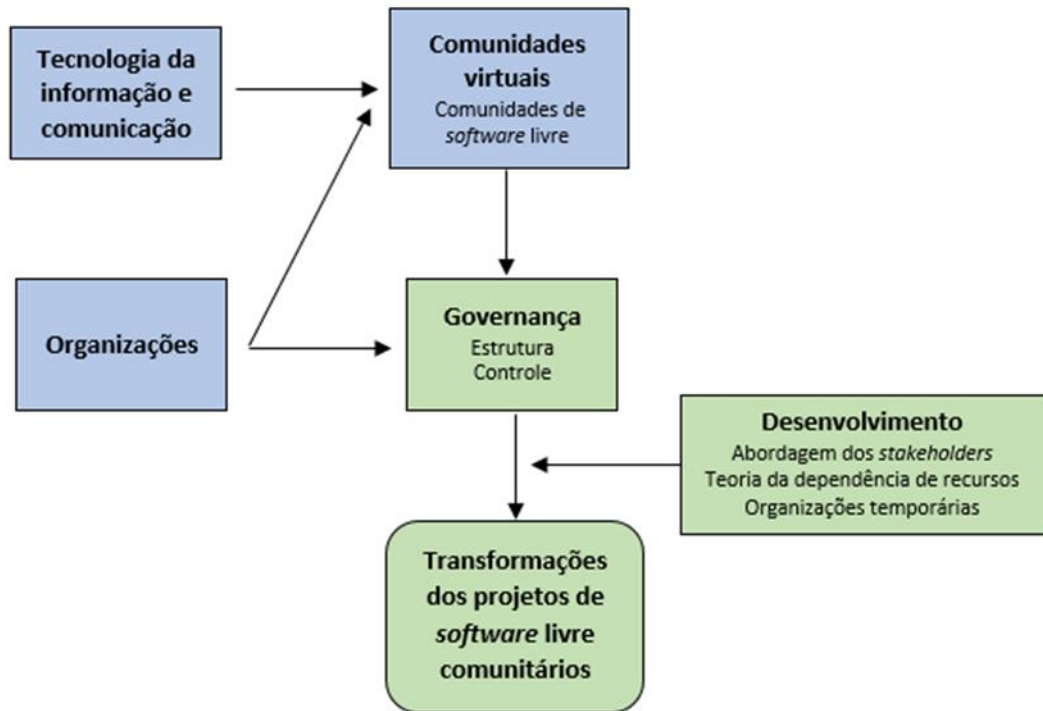
Tradicionalmente, as OTs eram compreendidas como arranjos complementares às estruturas organizacionais permanentes (Shenhar, 2001). Em quadro posterior, no entanto, configurações alternativas, como os projetos interorganizacionais e organizações baseadas em projetos, passaram também a ser considerados como representantes das OTs (Bakker, 2011; Burke & Morley, 2016). Sendo assim, essa modalidade de organização pode assumir as mais diversas formas, indo além dos agrupamentos intra e interorganizacionais, envolvendo também coletivos formados por diferentes indivíduos que cooperam e integram recursos e *expertises* para o alcance de resultados (Sydow et al., 2004).

Em virtude da amplitude do conceito e da multiplicidade de formatos que as OTs podem assumir, a literatura sobre essas organizações encontra-se bastante fragmentada (Lundin & Söderholm, 1995; Bakker, 2011; Burke & Morley, 2016). Diversos são os assuntos que frequentemente aparecem associados às OTs, tais como temporalidade, tarefas, comunicação, normas, definição de papéis, liderança, tomada de decisão, estrutura, técnicas de coordenação, etc (Lundin & Söderholm, 1995; Bakker, 2011). Outro aspecto abarcado pelo vasto e fragmentado número de publicações existentes sobre as OTs e que tem apoio substancial das pesquisas, refere-se ao fato de que as equipes que são mais novas, ou que estão em estágios iniciais de desenvolvimento são fundamentalmente diferentes das equipes tidas como *maduras* (Kozsłowski et al., 2009; Burke & Morley, 2016).

### 3.2.5 Desenvolvimento do argumento central da tese

A partir da revisão de literatura, bem como do arcabouço teórico apresentado, é possível identificar os principais conceitos trabalhados nesta tese. A figura seguinte ilustra esses conceitos e as suas relações, de acordo com a lógica delineada para a pesquisa.

**Figura 3** ó Diagrama conceitual da tese



**Fonte:** Elaborada pela autora (2019)

Como é possível perceber, em conjunto com a revisão de literatura sobre projetos de *software* livre comunitários e governança, na perspectiva da estrutura e do controle, o aporte teórico-conceitual envolvido pela abordagem dos *stakeholders*, teoria da dependência de recursos e organizações temporárias, também foi utilizado nas análises desta investigação. Este aporte de conhecimentos, normalmente retratado no âmbito das organizações privadas, assim como o próprio estudo da governança, constituíram um desafio, dado que são temáticas que não possuem consenso e trazem muitas opiniões divergentes no que se refere à sua aplicação no campo dos arranjos comunitários. A ideia, portanto, foi buscar entender o funcionamento dos projetos de *software* livre de forma a captar todo o contexto e história desses empreendimentos, não se limitando somente ao âmbito da comunidade em si.

Ao trazer essa visão mais ampla, a intenção é conseguir alcançar a compreensão das possíveis transformações da governança vivenciadas no decorrer da trajetória de desenvolvimento dos coletivos. Assim como ocorre em qualquer outra organização, presume-se que os projetos de *software* livre comunitários também estabeleçam relações institucionais com atores externos, uma vez que precisam buscar no ambiente em que se inserem os recursos necessários a sua sobrevivência no decorrer do tempo. No curso dessas vivências, ainda que traços informais possam se sobressair como atributos inerentes às iniciativas

organizadas no formato de comunidades, acredita-se que formalizações da governança se façam necessárias para acompanhar as etapas evolutivas desses projetos.

Isto posto, em consonância com os objetivos delimitados, é exatamente na problemática da transformação decorrente do crescimento das iniciativas de *software* livre, que se insere este estudo. A tese central é de que a governança, representada pela estrutura e processos de controle, transforma-se e torna-se mais formal em função da trajetória de desenvolvimento que os projetos de *software* livre comunitários apresentam. Como proposições decorrentes desta tese central, espera-se que, ao se desenvolverem, os projetos de *software* livre comunitários:

- Em relação à estrutura, apresentem maior hierarquia, especialização de papéis e descentralização decisória, ainda que algumas deliberações relevantes do projeto possam acontecer de forma centralizada;
- Em relação ao controle, utilizem-se de processos cada vez mais formais (controle direto, de comportamento e de resultados), mesmo que possam sobressair nesses coletivos o emprego dos procedimentos informais (autocontrole, controle entre pares, controle social).

A tese central e as proposições apresentadas buscam exprimir que a principal finalidade deste esforço investigativo se destina a obter uma melhor compreensão acerca da dinâmica do desenvolvimento dos projetos de *software* livre ancorados em um funcionamento comunitário. Será possível chegar a definições mais precisas quanto ao estado dos mecanismos de governança, dos tipos de arranjos e dos marcos que definem as trajetórias de amadurecimento desses coletivos, neste último aspecto, detendo-se, o presente estudo, na apreciação da aproximação ou distanciamento dos negócios de *software* livre às características inerentes aos modelos comunitários.

### 3.3 Método

As seções anteriores trouxeram a contextualização do tema, os objetivos, assim como os fundamentos conceituais e teóricos necessários ao entendimento das ideias centrais desta tese. Este capítulo, a seu turno, apresenta os passos metodológicos executados no campo com o propósito de realizar a verificação empírica dessas ideias.

A descrição metodológica é momento fundamental no sentido de garantir que a investigação tenha aceitação e validade acadêmica e científica. O método, conforme



apresentado nas seções seguintes, envolve todos os procedimentos sistematizados que viabilizam a coleta e a análise dos resultados da pesquisa.

### 3.3.1 Delineamento da pesquisa

A escolha dos procedimentos metodológicos deriva, primordialmente, de perspectivas tomadas anteriormente nos planos ontológicos e epistemológicos. São pressupostos adotados nesses níveis que determinam a visão de mundo do pesquisador (Guba & Lincoln, 1994), bem como a sua noção de sujeito e das relações desse sujeito com o objeto investigado (Morgan & Smircich, 1980).

De acordo com a visão de Guba e Lincoln (1994), qualquer posição paradigmática adotada pelo pesquisador pode ser combinada com as diferentes abordagens de pesquisa existentes, sejam elas quantitativas ou qualitativas. O presente trabalho de tese pautou-se pela realização de uma pesquisa de caráter qualitativo, que tenta se posicionar em meio termo diante dos paradigmas pós-positivista e interpretativista, visto que a maior preocupação durante toda coleta e análise dos dados residiu em trazer um entendimento do contexto sob exame. Apesar dessa tentativa de neutralidade, parte-se do pressuposto de que a realidade se configura como uma construção social, pois apresenta no seu cerne as perspectivas de mundo dos atores que participam dela, não existindo, portanto, neste aspecto, uma visão pré-concebida da situação por parte da pesquisadora (Orlikowski & Baroudi, 1991; Klein & Myers, 1999).

Apesar de pesquisa qualitativa exibir uma trajetória crescente e de sucesso, trata-se de uma abordagem que não tem definição totalmente consolidada, uma vez que compreende diferentes visões de mundo, o que a afasta de um único núcleo paradigmático (Morgan & Smircich, 1980; Flick, 2009). Em princípio, a abordagem qualitativa procura obter informações detalhadas sobre o fenômeno estudado e parte da necessidade de alcançar um *insight* profundo sobre a dinâmica social da organização ou do contexto em análise (Alavi, 2014). Por outro lado, a pesquisa qualitativa tem uma natureza essencialmente descritiva, de modo que a palavra tem lugar de destaque na sua concepção, o que faz com que o ambiente e as pessoas nele inseridas devam ser apreendidas holisticamente e não reduzidas a meras variáveis (Godoy, 2012). Por fim, acredita-se que o bom estudo qualitativo não se limita a concluir e a confirmar o que se espera que seja o resultado da pesquisa, mas, muito pelo contrário, deverá produzir novas ideias e formas de ver as coisas e as pessoas que estão sendo examinadas (Flick, 2009).

Dentro da abordagem qualitativa, este trabalho optou pela execução de um estudo de casos múltiplos, que tem se mostrado bastante proeminente nas investigações organizacionais (Eisenhardt & Graebner, 2007). O estudo de caso, em sua essência, tem como proposta o entendimento da dinâmica presente em contextos especiais (Eisenhardt, 1989), com foco na descrição detalhada e rica do ambiente no qual o objeto de estudo se encontra (Eisenhardt & Graebner, 2007). É a partir dessa descrição que cada caso é percebido como algo familiar, o que acelera o próprio processo posterior de análise dos dados (Eisenhardt, 1989). Outra característica do estudo de caso, é que, comumente, enquanto estratégia, ele tende a mostrar resultados muitas vezes opostos às perspectivas iniciais do pesquisador, ou seja, contém um maior viés para a contradição das noções pré-concebidas do que para a verificação dessas mesmas noções (Flyvbjerg, 2007). A escolha neste trabalho pela variante do estudo de caso, envolvendo a análise de casos múltiplos, decorreu da necessidade da pesquisadora em ter mais de um contexto de investigação, na tentativa de alcançar análises mais consistentes e significativas sobre a governança e o desenvolvimento dos projetos de *software* livre comunitários.

### 3.3.2 Seleção dos casos

Na realização de pesquisas, cada caso escolhido deve servir a um propósito específico dentro do escopo global da investigação (Yin, 2010). Desse modo, ainda que o estudo de casos múltiplos possa exigir maior tempo e recursos da parte do pesquisador (Yin, 2010), os resultados obtidos mediante a apreciação de mais de uma unidade de análise, tal como aconteceu nesta pesquisa, possibilita a realização de comparações mais sólidas (Eisenhardt, 1989), envolvendo semelhanças, diferenças ou particularidades entre situações, o que contribui favoravelmente aos interesses da investigação. A seleção de casos de forma aleatória não é algo preferível, pois os contextos devem ser escolhidos e examinados para preencher lacunas teóricas ou até mesmo oferecer exemplos extremos (Eisenhardt, 1989).

Como esta pesquisa tentou compreender a governança de projetos de *software* livre formatados em comunidades, por meio dos elementos estruturais e dos processos de controle, considerando o desenvolvimento e as transformações ocorridas nesses projetos, a escolha de casos que tivessem características distintas de funcionamento foi algo relevante para atender ao propósito central trazido pela tese. Nessa escolha, utilizou-se uma perspectiva importante do estudo de casos múltiplos, a de tipos polares ou extremos, cuja finalidade reside em fomentar contraste entre os cenários analisados, de modo a averiguar a repetição de padrões,

ou a existência de singularidades, que definam cada um desses cenários (Eisenhardt & Graebner, 2007).

Com base nessas considerações, para o estabelecimento de triangulação e também como forma de aumentar a confiabilidade da investigação (Gaskell, 2002), o quantitativo de unidades de análise determinado para a pesquisa foi de três casos. Intencionalmente selecionados, e em congruência com a necessidade de comprovação e potencial consecução dos objetivos do estudo, procurou-se assegurar que os três casos tivessem forte correspondência com o aparato conceitual levantado na literatura sobre a governança de comunidades de *software* livre. Assim, uma vez que a proposta inicial era obter casos representados por iniciativas organizadas em comunidades virtuais e portadoras de diferentes atributos de desenvolvimento, com base nos estudos de Whyn (2004), Xu et al. (2005), Latterman e Stieglitz (2005) e Panchal (2010), os projetos eleitos caracterizaram-se, enfim, por diferirem, entre si, em relação aos seguintes critérios: ano de surgimento; número de *releases* estáveis (versões do *software*); número atual de *committers* (desenvolvedores com acesso de gravação no repositório principal do código-fonte); licença usada e presença de fundações ou patrocinadores. Tais critérios corroboram com a ideia de que existem variações na governança de projetos de *software* livre em função do perfil de desenvolvimento. O outro critério considerado na escolha dos casos foi que deveriam tratar-se de projetos comunitários surgidos no Brasil e que se tornaram soluções competitivas de *software* livre.

Para chegar aos possíveis casos de estudo, a pesquisadora entrou em contato, por *e-mail*, com pesquisadores e desenvolvedores da área de *software* livre, solicitando-lhes a indicação de projetos comunitários que fossem competitivos no cenário nacional. Complementarmente, a pesquisadora mapeou alguns projetos nacionais a partir das programações disponibilizadas na Internet de eventos realizados no Brasil, no ano de 2017, tais como o Fórum de Tecnologia em *Software* Livre, Fórum Internacional de *Software* Livre e o Fórum Goiano de *Software* Livre. Para cada um dos projetos identificados, a pesquisadora levantou informações gerais e os contatos dos responsáveis, por meio dos *sites* existentes e/ou dos conteúdos disponibilizados na plataforma de hospedagem de código-fonte denominada *GitHub*.

De posse desses dados, registrados em planilha Excel, e sem perder de vista os critérios definidos quanto à representatividade do caso e à disponibilidade de informantes-chave em cada um dos cenários, isto é, indivíduos dispostos a colaborar com o estudo, foram escolhidos os três casos principais deste trabalho de tese. O quadro seguinte detalha cada um desses casos em função da aplicação dos critérios definidos para este estudo.

**Quadro 3** ó Critérios de seleção dos casos para estudo

<b>Características de desenvolvimento</b>	<b>Projeto Kytos</b>	<b>Projeto CoGrOO / Apache OpenNLP</b>	<b>Projeto Noosfero</b>
<b>Ano de surgimento</b>	2014	2003 / 2000	2007
<b>Número de <i>releases</i> estáveis</b>	3 (2018.2)	17 (4.3.1) / 34 (1.9.0)	191 (2.8.0)
<b>Número atual de <i>committers</i></b>	6	3 / 23 (considerando que parte do código do CoGrOO está na comunidade Apache OpenNLP)	12
<b>Licença utilizada</b>	MIT	Apache License 2.0	GNU AGPLv3
<b>Presença de Fundação</b>	Não	Sim (considerando que parte do código do CoGrOO está na comunidade Apache OpenNLP)	Não
<b>Presença de Patrocinador</b>	Sim	Não	Sim

**Fonte:** Elaborado pela autora, com base em dados disponibilizados na Internet (2019)

### 3.3.3 Coleta de dados

O processo de investigação qualitativa é altamente iterativo, pois envolve constantes idas e vindas às etapas em realização. É importante reconhecer que nem os construtos buscados em campo, nem a pergunta e os objetivos da pesquisa, são garantias para um determinado desenho de estudo, pois esses elementos representam tentativas que ao longo do processo investigativo podem ser modificados e readequados (Eisenhardt, 1989). Dessa forma, é legítimo alterar protocolos de coleta e adicionar fontes de dados, ou mesmo casos, no decorrer do estudo, inclusive, com sobreposição entre as fases de coleta e de análise dos dados (Eisenhardt, 1989). Tais considerações, é importante reconhecer, ganharam contornos reais nas tarefas práticas deste trabalho de tese.

A intuição tem papel mais importante na pesquisa qualitativa do que na pesquisa quantitativa, uma vez que essa última tem rotinas mais objetivas e estruturadas (Flick, 2009; Hair et al., 2015). Em virtude dessa característica da pouca linearidade que se observa na investigação qualitativa, em que se fazem presentes técnicas menos rígidas e que sugerem possibilidades de interpretação mais subjetivas (Hair et al., 2015), neste trabalho de tese procurou-se a descrição transparente do processo de investigação, tomando-a como algo fundamental (Eisenhardt & Graebner, 2007) e aspecto influente sobre a qualidade e rigor científico da abordagem empreendida (Flick, 2009).

É comum que o estudo de caso combine a utilização de múltiplas fontes de coleta, já que um dos meios para se alcançar a boa comparação entre unidades de análise reside no

exame de dados obtidos de diferentes maneiras (Eisenhardt, 1989; Eisenhardt & Graebner, 2007). À parte dos pressupostos filosóficos ou epistemológicos adotados, as investigações devem empregar fontes de dados que sejam coletados de formas distintas (Tran, 2014), com a intenção de chegar a demonstrações mais compreensivas e substanciais, e de maneira que a fraqueza de determinada fonte possa ser compensada por achados advindos de outra origem (Takhar-Lail & Chitakunye, 2015). Tal procedimento resulta em conclusões mais robustas e possibilita a minimização dos riscos de se incorrer em falsas interpretações (Eisenhardt, 1989; Dyer & Wilkins, 1991; Tran, 2014; Alavi, 2014). Além disso, como as pesquisas com mais de uma unidade de análise geralmente se defrontam com a ameaça de desenvolver descrições muito extensas e às vezes também sem profundidade (Alavi, 2014), o uso de diferentes fontes de dados pode permitir o aprofundamento e entendimento mais completo do fenômeno averiguado (Tran, 2014).

Por todas as alegações anteriormente elencadas, no interesse deste trabalho de tese optou-se pelo uso de duas fontes de coleta de dados: a entrevista e a pesquisa documental. A inclinação por ambas as formas de coleta pautou-se na ideia de trazer as percepções sobre governança de projetos de *software* livre comunitários, e acerca das transformações que vivenciaram em função de suas trajetórias de desenvolvimento, não apenas na perspectiva das experiências e opiniões dos atores envolvidos nesses coletivos, mas também considerando o ângulo das evidências documentais como complemento às narrativas então compiladas.

### 3.3.3.1 Entrevistas

As entrevistas tendem a ser a principal fonte de coleta de dados em estudos de caso, pois são bastante eficientes para prover conteúdos ricos e empíricos, uma vez que, presenciais, ou não, oportunizam um campo mais extenso para acolher questões humanas e a captura de variada gama de interpretações para uma determinada situação (Eisenhardt & Graebner, 2007; Yin, 2010). A utilização de entrevistas nas pesquisas pode assumir um perfil não estruturado, estruturado, ou semi-estruturado, de acordo com as necessidades do pesquisador em ter ou não um roteiro de perguntas previamente estabelecido (Richardson, 1999). Nas investigações qualitativas, a maioria dos pesquisadores têm preferência pela realização de entrevistas semi-estruturadas (Brinkmann, 2014). Essa modalidade permite que o pesquisador ao mesmo tempo em que use como guia uma lista de perguntas baseada nos seus interesses, tenha, também, a flexibilidade de explorar dos seus entrevistados as questões que considere mais relevantes (Longhurst, 2003). Nesse caso, as perguntas são

suficientemente abertas ao ponto de permitir que novos questionamentos surjam a partir das respostas oferecidas pelos entrevistados, de forma a sondar informações ou esclarecimentos adicionais aos relatos fornecidos (Wengraf, 2001; Longhurst, 2003).

Em estudos de casos múltiplos, para fins de comparação, é interessante que as entrevistas mantenham um mínimo de padronização. Desse modo, ao realizá-las, os pesquisadores devem tentar fazer as mesmas perguntas a todos os participantes, ou fazê-las de forma semelhante, de forma a não deixar de fora um tópico ou pergunta relevante para algum dos casos (Flick, 2009). Em relação à realização de entrevistas semi-estruturadas, especificamente, a ideia é que seja desenvolvido um roteiro de perguntas norteador para ser aplicado de forma mais ou menos constante em cada uma das entrevistas (Flick, 2009).

Pela flexibilidade permitida e pelo seu potencial, este trabalho de tese fez uso de um roteiro de entrevista semi-estruturado. Com base em categorias norteadoras advindas da literatura sobre o tema dos projetos de *software* livre formatados em comunidades, a pesquisadora formulou as questões a serem aplicadas junto aos participantes do estudo. No entanto, ainda que tivesse um roteiro semi-estruturado, o que garantiu certo padrão aos questionamentos feitos aos atores objeto da sondagem, a pesquisadora teve também a liberdade de aprofundar em certas particularidades de cada caso, a partir das respostas dos entrevistados, quer por meio de novas perguntas, quer mediante a solicitação de esclarecimentos adicionais.

O roteiro utilizado nesta pesquisa foi submetido à pré-teste prévio, com integrantes de diferentes projetos de *software* livre organizados de forma comunitária. Como a ideia era ter perguntas claras que se adequassem às comunidades com diferentes perfis, bem como estimular os entrevistados a contarem sobre eventos que aconteceram ao longo da história dos seus respectivos projetos, fazer essa testagem possibilitou ajustes para melhorias das questões propostas. Todas as entrevistas do pré-teste foram agendadas por *e-mail* e aconteceram presencialmente, mediante *Skype*, ou *Google Hangouts*. Ao todo, cinco entrevistas de testagem foram realizadas, com aproximadamente uma hora de duração cada uma delas. A principal restrição na escolha dos entrevistados para o pré-teste é que eles tivessem uma noção geral da história e da gestão atual dos coletivos em que atuavam. Os cinco entrevistados pertenciam aos seguintes projetos: LibreOffice Brasil, CoGrOO, Noosfero, Rocket.chat e Wordpress Brasil. O CoGrOO e o Noosfero, como será visto adiante, foram dois dentre os três casos que foram selecionados para esta investigação.

As entrevistas do pré-teste também funcionaram como teste-piloto para a pesquisadora, no sentido de perceber que, para os propósitos da tese, ter acesso aos

representantes principais dos projetos, com atuação desde o seu surgimento e/ou de diferentes fases da história dos coletivos, seria algo fundamental para entender as transformações pelas quais essas iniciativas passaram ao longo do tempo. Tal percepção, inclusive, corroborou com o critério de escolha de projetos nacionais para este estudo, sendo desconsiderados, portanto, projetos grandiosos e mundialmente reconhecidos, em especial pela dificuldade de acesso aos desenvolvedores que atuam neles desde o início. O porte desses casos famosos, também dificultaria o acesso amplo às informações necessárias e com a consistência pretendida para o alcance dos objetivos desta tese, durante o tempo de realização da pesquisa.

#### 3.3.3.1.1 Seleção dos participantes da pesquisa

Na pesquisa qualitativa a amostragem pode seguir critérios previamente definidos, de acordo com as necessidades da investigação, sendo o mais importante a sua representatividade, não no sentido estatístico do termo, mas sim na visão de que as informações coletadas sejam capazes de representar a relevância do fenômeno (Flick, 2009). A nomeação dos participantes de uma pesquisa por simples conveniência só deve acontecer quando nenhum outro critério técnico puder ser aplicado, sendo esse o último meio a recorrer para realizar a tarefa (Flick, 2009). Para autores como Gaskell (2002) e Oliveira (2012), como a ideia de amostra baseia-se em critérios estatísticos, essa denominação não deveria ser aplicada ao universo da pesquisa qualitativa, o que torna mais apropriado o uso do termo seleção, em vez da nomenclatura amostra, para se referir a escolha dos sujeitos participantes do estudo. Neste trabalho de tese foi considerada a percepção trazida por Gaskell (2002) e Oliveira (2012).

A seleção dos participantes de uma pesquisa pode acontecer por acaso ou de modo intencional (Gaskell, 2002). Na pesquisa qualitativa, a seleção intencional, devidamente embasada, é a forma mais apropriada a esforços desta espécie. Assim, os pesquisadores não correm o risco de deixar de fora indivíduos que tenham relevância para os interesses da investigação (Rodrigues, 2005), uma vez que os requisitos para a escolha dos participantes são determinados previamente no planejamento da pesquisa (Oliveira, 2012). Uma das técnicas mais conhecidas para a realização da seleção intencional é a da *ôbola de neve* (*snowball*), que envolve a localização dos participantes de interesse e a solicitação de que eles indiquem outros sujeitos que também possam participar da pesquisa (Merriam, 1998). Segundo Morrow (2005), essa forma de definir os participantes é legítima cientificamente, e

permite a identificação de indivíduos na condição de bons exemplares para a coleta de informações sobre o fenômeno que se averigua.

Na presente investigação, seguindo-se o protocolo que definiu os casos para estudo (ver seção 3.3.2), partiu-se para a escolha dos sujeitos informantes das entrevistas semi-estruturadas. A seleção aconteceu de modo intencional, sendo os entrevistados representados por integrantes da liderança, desenvolvedores e partes interessadas que mantiveram participação ativa no grupo central dos coletivos examinados. A atuação no grupo central foi uma forma de garantir que os participantes das entrevistas fossem informantes que tivessem noção da história e das ações de governança voltadas ao alcance de resultados do projeto.

No entanto, como nem todos os indivíduos atuaram no projeto desde o surgimento, os primeiros entrevistados de cada caso foram criteriosamente definidos em função de serem pessoas que perpassaram por todo o desenvolvimento desse projeto e que estão ativos até os dias atuais. Ter o acesso a esses primeiros entrevistados, em decorrência do conhecimento que detinham sobre o projeto, foi algo essencial para esta pesquisa. Assim, durante a análise das entrevistas e dos documentos, sempre que necessário, a pesquisadora teve a abertura de recorrer a eles, para esclarecer dúvidas ou solicitar alguma explicação sobre determinada questão. Para a seleção dos demais entrevistados, a técnica aplicada na escolha intencional dos participantes foi a da *õbola de neve*, que, como já explicado, baseia-se na indicação de outras pessoas tidas como aptas a auxiliar no entendimento das questões centrais levantadas pela tese. Assim como aconteceu no pré-teste, essas entrevistas foram agendadas pela pesquisadora via *e-mail* e aconteceram presencialmente ou por intermédio do *Google Hangouts* e *Skype*, em dias e horários que foram previamente definidos.

Em um dos casos, especificamente o do CoGrOO, como foi identificado que a história do projeto em dado momento apresentou interface do código-fonte com outro projeto global, denominado *Apache OpenNLP*, um procedimento complementar de coleta de dados foi aplicado, além das entrevistas semi-estruturadas. Desse modo, as informações que não conseguiram ser alcançadas pela entrevista com o informante-chave desse caso, ou pelos documentos consultados, foram complementadas por meio de questionamentos enviados via *e-mail* aos participantes do projeto *Apache OpenNLP*. Assim, conforme será visto na seção de análise dos dados, mesmo que todo o caso seja descrito na perspectiva de desenvolvimento do CoGrOO, como as transformações da governança ocorridas envolveram o projeto *Apache OpenNLP*, entender o funcionamento desse último projeto também foi necessário para alcançar os objetivos estabelecidos por esta pesquisa. As perguntas formuladas para obtenção desses esclarecimentos adicionais encontram-se no Apêndice, ao final da tese.



O quadro seguinte apresenta o perfil geral de cada caso um dos casos no que se refere às informações coletadas mediante as entrevistas semi-estruturadas e os questionamentos por *e-mail* (no caso da interface CoGrOO / Apache OpenNLP).

**Quadro 4** ó Características gerais das entrevistas

<b>Projeto</b>	<b>Quantidade de entrevistas</b>	<b>Número de páginas transcritas</b>
<b>Kytos</b>	5	66
<b>CoGrOO</b>	3 + 2 formulários via <i>e-mail</i>	47
<b>Noosfero</b>	6	95

**Fonte:** Elaborado pela autora (2019)

Uma característica particular deste trabalho, no que diz respeito a escolha dos entrevistados, é que o uso da saturação teórica não foi algo buscado como parâmetro para estabelecer o quantitativo de entrevistas. No protocolo seguido, mais importante do que alcançar a saturação teórica, era conseguir a disponibilidade de pessoas conhecedoras da realidade investigada e ter percepções distintas do contexto sob exame, para que assim fosse possível obter uma narrativa consistente com os propósitos da tese. Nesse sentido, o trabalho teve uma aproximação maior com os pressupostos trazidos pela amostragem estrutural, baseada na realização de entrevistas com atores ocupantes de papéis distintos que participam de um sistema social ou que interagem com ele (Shah & Gorbatai, 2014). Acredita-se que a amostragem estrutural tenha utilidade quando empregada para compreender os relacionamentos dos indivíduos e o sistema social em que atuam, assim como para a apreensão de como os aspectos da estrutura de governança desse sistema social moldam os comportamentos e relações individuais (Shah & Gorbatai, 2014).

Dito isto, como a amostragem estrutural demonstra ser procedimento apropriado para examinar a estruturação e o funcionamento dos sistemas sociais, a partir da ótica de atores ocupantes de papéis heterogêneos, esta tese converge com essa técnica. Assim, foram realizadas entrevistas com atores representados pela liderança, desenvolvedores e partes interessadas envolvidas no grupo central dos projetos de *software* livre comunitários investigados. Não houve assim, a restrição de um único perfil de entrevistados, por considerar ser relevante na compreensão mais abrangente do caso, ter diferentes percepções sobre a mesma realidade. Ainda que os demais entrevistados não tivessem conhecimento tão profundo do caso, como os informantes-chave tinham, cada um deles contribuiu com visões complementares, que permitiram trazer um retrato mais consistente do contexto social estudado.

### 3.3.3.2 Pesquisa documental

A pesquisa documental é um processo sistemático de avaliar e rever dados secundários representados por correspondências, relatórios, documentos administrativos, pareceres formais, artigos publicados na mídia, dentre outros materiais pertencentes ao contexto de estudo no qual o pesquisador está atuando (Yin, 2010; Alavi, 2014). Assim como outras técnicas de obtenção de dados usadas na pesquisa qualitativa, o levantamento documental requer que os dados sejam atentamente examinados e interpretados com a intenção de desenvolver um significado (Bowen, 2009). O acesso ao acervo documental produzido pelo caso analisado é bastante utilizado de forma suplementar ou em combinação com outros métodos qualitativos, como forma de validar os resultados, reduzir vieses (Yin, 2010), ou até de sugerir novas questões e situações a serem esclarecidas (Bowen, 2009).

Os documentos têm a vantagem de constituir fonte de dados que pode ser revisada múltiplas vezes e que não foi criada deliberadamente com o propósito de atender aos objetivos de determinada pesquisa (Yin, 2010). Em contrapartida, ao analisar os documentos produzidos em uma organização, o pesquisador precisa ser criterioso e não deve enxergá-los como uma base informacional precisa e completa, sendo necessário considerar os contextos em que foram produzidos e a determinação da autenticidade, credibilidade, acurácia e representatividade dos mesmos (Bowen, 2009).

Nesta investigação a pesquisa documental foi realizada em dois momentos. O primeiro deles consistindo do levantamento das informações sobre os projetos de *software* livre mapeados como possibilidades para o estudo de casos múltiplos, conforme descrito na seção 3.3.2. Nesse contato inicial com o campo de estudos, a pesquisadora empreendeu uma pesquisa documental nos *sites* e na plataforma *GitHub* com a finalidade de obter informações, então registradas em planilha de Excel, que pudessem ser utilizadas na aplicação dos critérios estabelecidos para a seleção dos casos.

O segundo momento ocorreu quando foi efetivada a pesquisa documental dos casos, compreendida pela etapa da coleta de dados. Nessa etapa, o acesso aos documentos funcionou como um meio de complementar as informações obtidas mediante as entrevistas com os integrantes do grupo central dos projetos de *software* livre comunitários investigados. Toda a atenção voltou-se no sentido de apurar registros de *e-mails*, conteúdos disponibilizados em *sites*, documentos diversos (*slides*, relatórios, manuais, boletins etc), ou até notícias

divulgadas, que pudessem ser úteis no registro da história e das transformações que ocorreram em cada caso analisado, em especial, na compreensão dos aspectos de governança.

No que se refere aos *e-mails*, em particular, para ter acesso aos registros a pesquisadora se cadastrou nas listas abertas das comunidades referentes a cada caso. Assim, entre o período de agosto até outubro de 2018, foi realizada uma análise das comunicações ocorridas. A intenção dessa análise possibilitou não só uma maior aproximação da pesquisadora com o contexto investigado, mas também proporcionou que cada um dos casos pudessem ser melhor compreendido em termos da sua rotina de funcionamento e dos conteúdos que costumam ser disseminados entre os participantes. Excepcionalmente no caso CoGrOO, devido a sua interação com o projeto OpenNLP, a pesquisadora acompanhou de forma suplementar as listas de *e-mail* desse outro projeto e da incubadora da fundação Apache. Vale salientar, que os *e-mails* da incubadora serviram basicamente como informações adicionais para auxiliar a pesquisadora na compreensão da dinâmica entre comunidade e fundação, já que o projeto OpenNLP passou por uma fase de incubação antes de entrar na Apache.

Seja no levantamento das informações sobre os potenciais casos, seja na coleta de dados, com os casos já definidos, a pesquisa documental foi um momento basilar para melhor conhecer o campo de estudos de modo geral e as atividades desenvolvidas nos projetos de *software* livre examinados. A clareza e a fidedignidade trazidas pelos documentos de como esses projetos com atuação comunitária se estruturam para organizar as contribuições em torno do código-fonte do *software* foi um passo fundamental, além das entrevistas, na explicação do fenômeno investigado.

### 3.3.4 Análise de dados

A fase de análise pressupõe que algum tipo de transformação aconteça em relação aos dados coletados (Gibbs, 2008). O objetivo maior da análise dos dados é conseguir ir além deles, de modo a encontrar sentido e compreensão ao que eles revelam (Gaskell, 2002). No caso das pesquisas qualitativas, a ideia é que os volumosos dados textuais coletados de múltiplas fontes sejam processados mediante o emprego de procedimentos analíticos que possibilitem a sua transformação em uma análise coerente e que traga evidências de sustentação para a pesquisa (Gibbs, 2008). Essa análise e interpretação dos dados implicam em tempo e esforço do pesquisador, assim como a sua imersão no corpo do texto (Gaskell, 2002).

O alcance de uma análise clara, criteriosa e original, no entanto, requer uma abordagem estruturada dos dados coletados, o que pode ser conseguido por meio do emprego de *software* de análise de dados qualitativos (Gibbs, 2008). Nesta pesquisa, especificamente, como foi obtido um grande volume de dados, em virtude das entrevistas e da pesquisa documental que foram realizadas junto aos casos representados pelos projetos de *software* livre comunitários, o uso de ferramenta tecnológica possibilitou maior organização e robustez às análises. O *software* utilizado no suporte à análise qualitativa dos dados coletados neste trabalho foi o ATLAS.ti.

A técnica de análise de dados escolhida para ser aplicada nesta tese foi a da análise de conteúdo, tal como preconizada por Bardin (2010), na qual são utilizados procedimentos sistemáticos a partir dos conteúdos textuais coletados. Essa técnica é bastante disseminada por pesquisadores em estudos qualitativos, sendo amplamente reconhecida na área de estudos organizacionais (Mozzato & Grzybovski, 2011; Alavi, 2014). Ainda que algumas quantificações possam ser realizadas com a intenção de oferecer suporte analítico, a essência deste trabalho esteve embasada nos procedimentos da corrente qualitativa da análise de conteúdo (Krippendorff, 2013; Mayring, 2014).

A análise de conteúdo consiste essencialmente na realização de quatro etapas principais (Bardin, 2010): 1) pré-análise ó envolve a sistematização e organização do material textual coletado, com o objetivo de torná-lo operacional, no sentido de conduzir um esquema preciso de desenvolvimento das operações sucessivas em um plano de análise; 2) codificação ó corresponde a um processo de marcação das unidades de análise, em que os dados brutos coletados passam por uma transformação (recorte, agregação e enumeração) que permite o alcance de uma representação exata de seu conteúdo; 3) categorização ó consiste em uma operação de criação de classes, que têm o objetivo de reunir um grupo de elementos em razão das suas características comuns baseadas em critérios semântico, sintático, léxico ou expressivo; 4) inferência ó baseia-se em uma atividade de indução com o propósito de trazer as percepções e a interpretação do pesquisador aos fatos constatados no conteúdo sob sua análise. Vale ressaltar que, tal como aconteceu nesta pesquisa, essas etapas não necessariamente acontecem de maneira linear e sequencial (Bardin, 2010; Krippendorff, 2013).

Para o que é de interesse deste estudo e como mecanismo norteador para a elaboração do roteiro de entrevista, algumas categorias ou dimensões decorrentes da literatura foram definidas previamente, referentes aos conceitos de estrutura e controle organizacional. O quadro, a seguir, descreve cada uma delas em função dos indicadores que possibilitam a sua

operacionalização no campo. Contudo, vale ressaltar que, o estabelecimento prévio das categorias, não significa dizer que todas elas foram necessariamente observadas na pesquisa empírica e nem que outras novas categorias não pudessem ser descobertas durante o próprio procedimento analítico dos dados coletados. Até mesmo pela flexibilidade da pesquisa qualitativa, e pelo caráter iterativo que apresenta, a pesquisadora manteve-se aberta para que os dados analisados pudessem trazer novas descobertas em relação aos conceitos trabalhados nesta investigação.

**Quadro 5** ó Categorias pré-definidas para a pesquisa a partir da literatura

<b>Conceito</b>	<b>Dimensão</b>	<b>Indicadores</b>
<b>Estrutura</b>	Hierarquia	Linha de autoridade, quem se reporta a quem na organização, relações de subordinação.
	Especialização de papéis	Diferentes papéis distribuídos entre os indivíduos, divisão do trabalho, especificação de responsabilidades e funções.
	Centralização	Tomada de decisão e autoridade concentrada em um determinado indivíduo ou nível organizacional.
<b>Controle</b>	Direto	Supervisão direta, vigilância expressa, imposição de ordem, normas técnicas e sanções.
	Controle de comportamento (burocrático)	Padronização, regras, regulamentos das atividades, legitimação da autoridade em hierarquias, formalização, especialização, centralização.
	Controle de resultados	Mensuração, recompensas com base nos resultados individuais alcançados, competição, trocas monetárias e econômicas.
	Social/cultural	Crenças, valores, culturas compartilhadas, produção conjunta de repertório de práticas que define a identidade do coletivo, socialização dos indivíduos.
	Controle entre pares	Ajustamento mútuo entre os indivíduos, crítica ao trabalho dos pares.
	Autocontrole	Autonomia, independência na realização do trabalho.

**Fonte:** Elaborado pela autora, com base em Vasconcellos e Hemsley (1989), Ouchi (1979), Loughry (2009) e Van Maanen (2010)

Todos os dados coletados para esta tese, por meio das entrevistas e pesquisa documental, foram transformados em material textual que possibilitou a aplicação da técnica de análise de conteúdo como procedimento analítico. Desse modo, seguindo os passos da técnica de análise de conteúdo, tal como preconizado por Bardin (2010), foi possível obter interpretações do material coletado que permitiram compreender a governança dos projetos de *software* livre comunitários investigados, em função dos seus aspectos estruturais e de controle, bem como considerando as transformações decorrentes da trajetória de desenvolvimento desses projetos.

Sendo assim, primeiramente, cada um dos casos estudados foram aprofundados em descrição individual, e, posteriormente, realizadas as análises fazendo as comparações entre

os casos, com foco na identificação de padrões, semelhanças ou diferenças desses projetos de acordo com as suas características de desenvolvimento. O resultado desse procedimento analítico intra e interprojetos, apresentado nas próximas seções, permitiram não só que a governança e as transformações decorrentes do desenvolvimento de cada caso pudessem ser retratadas, mas também a verificação da tese central desta pesquisa. Nesse aspecto foi possível verificar, em decorrência do crescimento alcançado pelos projetos, a formalização existente, assim como a aproximação ou o distanciamento deles do modelo comunitário. Além disso, também foi possível trazer algumas considerações sobre qual tipo de empreendimento as comunidades de *software* livre se tornam em função do desenvolvimento vivenciado.

### 3.3.5 Cuidados metodológicos

No decorrer desta investigação, algumas precauções foram tomadas para garantir dados mais fidedignos e a transparência do processo de pesquisa. Assim, desde o momento da entrada da pesquisadora em campo existiu a preocupação de que medidas fossem postas em prática para evitar e minimizar os erros ou problemas decorrentes da operacionalização das ações necessárias à realização da pesquisa.

O primeiro cuidado que a pesquisadora teve foi o de ter contato prévio com o campo de estudo, antes de iniciar a fase empírica da tese. O estabelecimento desse primeiro contato, viabilizado pela participação em eventos de *software* livre, foi fundamental para estabelecer uma aproximação inicial e para compreender de maneira geral como os indivíduos interagem nos encontros voltados para a troca de experiências sobre os projetos dessa área. Ademais, permitiu que a pesquisadora fizesse contatos que poderiam ser úteis nas fases posteriores da investigação.

Outro cuidado presente durante a investigação residiu no fato da pesquisa ser desenvolvida por etapas, ainda que essas etapas, pela própria natureza e flexibilidade da pesquisa qualitativa, não tenham acontecido de forma totalmente linear e sequencial, sendo marcadas por apresentar um caráter iterativo. Nesse sentido, com base na revisão de literatura empreendida, cada caso para estudo foi selecionado com base em critérios, considerando-se também a representatividade e a disponibilidade de acesso aos informantes e documentos, para, após esse momento, ser iniciada as fases de coleta e análise dos dados.

No que diz respeito às entrevistas, as informações levantadas junto aos sujeitos escolhidos para participar da pesquisa, tiveram a sua análise realizada logo em momento

posterior. Ter essa dinâmica nas análises permitiu que, ao ser constatada a necessidade de complementos ou aprofundamentos das narrativas obtidas, a pesquisadora tivesse a liberdade de retornar ao campo para buscar junto aos informantes-chave de cada caso os esclarecimentos adicionais que fossem relevantes aos propósitos da investigação. O apoio fornecido pelos informantes-chave foi algo crucial ao longo de toda coleta e análise dos dados, pois, como já relatado, esses indivíduos mostraram-se acessíveis e eram conhecedores profundos da história de cada projeto escolhido para o estudo.

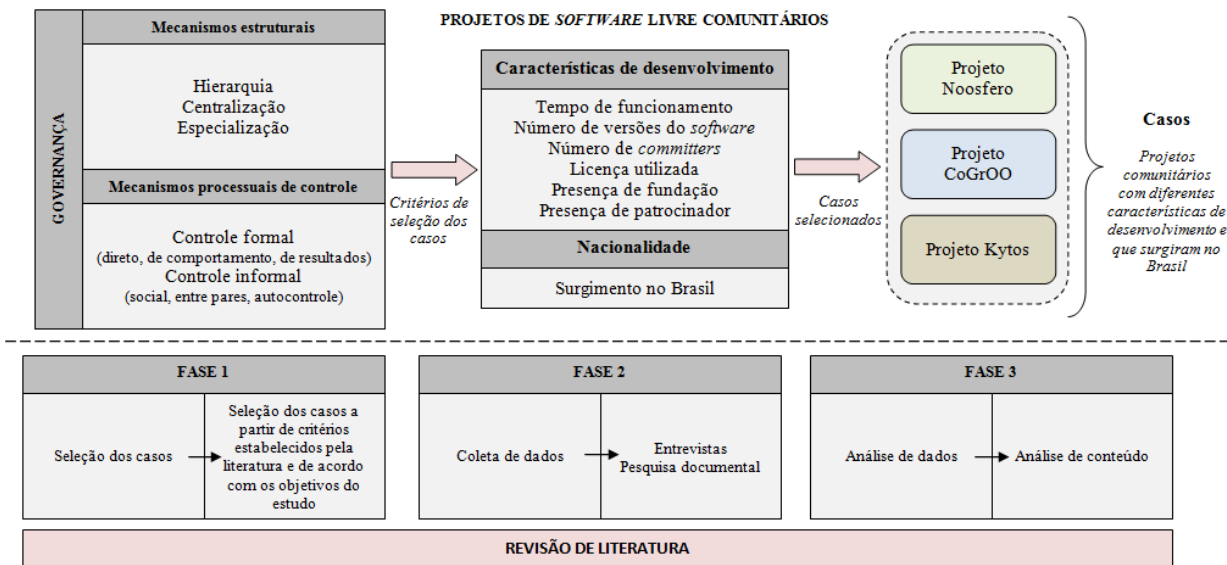
No que diz respeito aos questionamentos trazidos para as entrevistas, a pesquisadora fez a testagem prévia do roteiro semi-estruturado, tal como descrito na seção 3.3.3.1.1. A realização desse pré-teste permitiu não só a adaptação de algumas perguntas presentes no roteiro para o aprimoramento, mas também que a pesquisadora tivesse contato precedente com o campo estudado, antes das entrevistas oficiais dos casos. Em conjunto com a participação em eventos de *software* livre, as entrevistas de testagem funcionaram como um exercício prático para iniciação no campo de pesquisa, o que possibilitou uma maior familiarização com o tema e um conhecimento mais preciso do funcionamento dos projetos de *software* livre comunitários. Para todas as entrevistas desta tese, a pesquisadora tomou a precaução de apresentar a ideia geral do estudo e solicitou a autorização para fazer a gravação das conversas. No final desse processo, as entrevistas individuais gravadas foram transcritas, sendo acordado com os participantes que a qualquer tempo eles poderiam solicitar e ter acesso ao conteúdo produzido sobre o seu projeto de atuação.

Toda a pesquisa documental também esteve baseada em protocolos, com registros feitos em planilhas e arquivos digitais, no sentido de organizar as ações executadas no decorrer da coleta e análise dos dados. Finalmente, os dados textuais, decorrentes das entrevistas transcritas em editor de texto eletrônico e dos documentos acessados, consistiram em materiais manuseados com o suporte do *software* Atlas.ti. O uso desse *software* foi de suma importância para evitar a dispersão dos dados e as dificuldades que porventura acontecessem nas análises devido ao grande volume de informações armazenadas. Ter a estruturação dos dados ancorada em ferramenta analítica tecnológica constituiu uma cautela metodológica a mais, que em conjunto com as demais precauções tomadas e aqui apresentadas, buscou garantir maior qualidade para a pesquisa e a diminuição de subjetividades e vieses decorrentes das atividades executadas pela pesquisadora.

### 3.3.6 Desenho da pesquisa

O desenho de uma pesquisa é determinado principalmente pelos objetivos do estudo, a natureza do fenômeno, o contexto e os meios disponíveis (Tran, 2014). Nesse sentido, o pesquisador deve escolher um plano que ofereça informações relevantes sobre as questões de pesquisa e que permita o desempenho da sua função de modo mais eficiente (Hair et al., 2015). Para o que é de interesse desta tese, a figura seguinte traz em detalhes os passos teóricos e metodológicos desenvolvidos na execução da pesquisa.

**Figura 4** o Desenho metodológico da pesquisa



**Fonte:** Elaborada pela autora (2019)

Como é possível observar, as categorias de governança analisadas nesta investigação, envolveram os aspectos estruturais e processuais de controle, no âmbito do cenário delimitado, configurado pelos projetos de *software* livre com atuação comunitária virtual. Assim, tendo como base as percepções trazidas pela literatura dessa área de estudos, assim como os objetivos definidos para a pesquisa, foram estabelecidos os critérios para a escolha dos casos examinados. Em decorrência da aplicação desses critérios, os múltiplos casos componentes da pesquisa foram selecionados para análise, sendo representados por três projetos de *software* livre comunitários que surgiram no Brasil e apresentavam diferentes trajetórias de desenvolvimento, denominados Noosfero, CoGrOO e Kytos.

Na prática, toda a realização da pesquisa empírica passou por três fases distintas: a primeira, envolvendo o estabelecimento de contatos com representantes da área de *software* livre e o levantamento de informações sobre os possíveis casos para estudo, para que, a partir da aplicação dos critérios estabelecidos fossem selecionados os casos participantes; a



segunda, abrangendo a coleta de dados, que envolveu a realização de entrevistas e a pesquisa documental referente aos projetos escolhidos para o estudo; a última fase, finalmente, sendo representada pela análise dos dados textuais coletados, que aconteceu por meio da técnica de análise de conteúdo. Todas essas fases foram necessárias no sentido de verificar no plano empírico a tese central e as proposições trazidas por esta tese (seção 3.2.5).

### 3.4 Análise dos resultados

Em sintonia com os aspectos metodológicos apresentados, este capítulo se detém na análise dos dados coletados na pesquisa de campo. Feita uma prévia caracterização dos casos e dos entrevistados, os resultados da pesquisa foram discutidos em detalhe. Primeiramente, cada caso selecionado foi descrito de forma individual, para, em momento posterior, ser apreciado de maneira conjunta com os demais. Todas as análises abordaram o assunto central da tese, isto é, a temática da governança, na perspectiva da estrutura e do controle, considerando as transformações decorrentes do desenvolvimento dos projetos de *software* livre comunitários.

#### 3.4.1 Projeto CoGrOO

O projeto CoGrOO é um corretor gramatical acoplável ao OpenOffice.org e que auxilia na escrita de textos, verificando se a gramática empregada se encontra de acordo com a norma culta. O *software* identifica uma série de erros, tais como os referentes à colocação pronominal, concordância nominal e verbal, uso de crase, regência nominal e verbal, entre outros erros comuns da língua portuguesa. Para cumprir com esse propósito, o CoGrOO emprega técnicas de processamento de linguagem natural (PNL), ou em inglês, *natural language processing* (NLP). Essa subárea da inteligência artificial realiza a interação entre máquina e linguagens humanas naturais, de modo a possibilitar que os computadores tenham a capacidade de compreender e compor textos. No decorrer da trajetória de desenvolvimento do CoGrOO, descrita em detalhes adiante, esse projeto passou a ter uma interseção de parte significativa do seu código-fonte com o projeto de porte global Apache OpenNLP.

O projeto Apache OpenNLP, a seu turno, se trata de uma biblioteca que suporta um conjunto de ferramentas voltadas ao aprendizado de máquina Java, direcionadas para o processamento de texto em linguagem natural, ponto no qual converge com o *software* CoGrOO. O projeto teve início no ano 2000, por iniciativa de dois alunos da área de

informática da Universidade de Edimburgo. Ao longo dos anos, a biblioteca apresentou um crescimento considerável pela sua aplicação na academia e na indústria, bem como pela realização de publicações e pesquisas sobre o *software*. Uma importante mudança ocorrida no projeto OpenNLP aconteceu quando ele foi submetido ao processo de incubação promovido pela *Apache Software Foundation*, uma fundação sem fins lucrativos, financiada por doações individuais e por patrocinadores corporativos, a qual fornece serviços e suporte para o funcionamento de projetos de *software* livre comunitários em todo o mundo. Após esse processo de incubação, que é direcionado para o amadurecimento do projeto, a biblioteca OpenNLP passou a integrar a fundação Apache, lugar onde ele se encontra operando atualmente.

Exposta a visão geral dos dois projetos, considera-se relevante explicar que, apesar de a presente pesquisa se concentrar em contar a trajetória de desenvolvimento do projeto CoGrOO, a opção por considerar adicionalmente os aspectos de governança do projeto Apache OpenNLP se mostrou oportuna e foi adotada no curso de realização das entrevistas. Em conversa com o principal desenvolvedor do projeto CoGrOO (identificado como E1), que desde o ano de 2010 participa do projeto OpenNLP, a pesquisadora constatou que ambos os projetos, em determinado momento, direcionaram os seus fluxos em um ponto comum. Essa convergência aconteceu quando o desenvolvedor E1, por meio de iniciativa individual, decidiu incorporar grande parte do código do CoGrOO ao projeto Apache OpenNLP. Como consequência dessa ação, o gerenciamento recente de parcela do código do CoGrOO passou a acontecer também pela via da comunidade Apache OpenNLP, ainda que os dois projetos permaneçam existindo de forma independente. É por esse motivo que o presente caso, ainda que explicado na perspectiva do projeto CoGrOO, incorpora na sua narrativa, de forma complementar, as práticas identificadas no projeto Apache OpenNLP no que concerne aos aspectos da governança (estrutura e controle). Os entrevistados de cada um desses projetos se encontram discriminados no Quadro 6 a seguir.

**Quadro 6** Identificação e perfil dos entrevistados do projeto CoGrOO / OpenNLP

<b>Código</b>	<b>Função no projeto</b>	<b>Observação</b>
<b>E1</b>	Desenvolvedor	Principal desenvolvedor do CoGrOO, que também participa ativamente do projeto Apache OpenNLP
<b>E2</b>	Desenvolvedor	Desenvolvedor que participou de determinada fase do projeto CoGrOO
<b>E3</b>	Analista técnico	Docente responsável por conceber a ideia inicial do CoGrOO e estabelecer vínculos com outros docentes e pesquisadores
<b>F1</b>	Desenvolvedor	Desenvolvedor pertencente ao projeto Apache OpenNLP
<b>F2</b>	Desenvolvedor	Desenvolvedor pertencente ao projeto Apache OpenNLP

**Fonte:** Elaborado pela autora (2019)

Para o alcance dos objetivos delimitados, foram realizadas entrevistas com 3 participantes do projeto CoGrOO, um deles, aliás, também participante do projeto Apache OpenNLP (E1). Além dessas entrevistas, formulários por *e-mail* com perguntas abertas foram enviados para os cinco principais desenvolvedores da comunidade Apache OpenNLP, obtendo-se resposta de dois deles (F1 e F2). Principalmente em relação ao Apache OpenNLP, as evidências documentais disponibilizadas no *site* desta fundação tiveram um papel essencial para complementar o levantamento de informações obtido pelos formulários e entrevista com o informante E1. O caso CoGrOO, portanto, se diferencia dos demais focalizados nesta pesquisa, em virtude do aspecto interseccional com o OpenNLP.

#### 3.4.1.1 Trajetória de desenvolvimento do projeto CoGrOO

O surgimento do projeto CoGrOO aconteceu em agosto do ano de 2003, quando um professor universitário que, à época, trabalhava no Serviço Nacional de Aprendizagem Comercial (Senac-SP), teve a oportunidade de participar do evento Semana de *Software Livre* no Legislativo. Nesse evento, o docente ouviu de um dos palestrantes do Tribunal de Contas do Distrito Federal que existia a necessidade de ser criado um corretor gramatical para o OpenOffice.org. De posse dessa informação, o referido professor, algum tempo depois, em conjunto com outros colegas docentes e pesquisadores, amadureceu a ideia e decidiu elaborar um projeto voltado para o desenvolvimento de um mecanismo de correção ortográfica que pudesse ser acoplado ao OpenOffice.org. A forma como se deu esse início pode ser notada na própria fala do docente, na entrevista identificado como indivíduo E3:

Na época, eu trabalhava no Senac aqui de São Paulo, e tinha um diretor nosso que falava bastante de *software* livre e houve um evento em Brasília, um congresso, era um evento focado em *software* livre, patrocinado pelo Congresso Nacional. E aí, eu assisti esse evento, e um dos gestores do Tribunal de Contas falou que era uma necessidade bastante forte deles de ter uma ferramenta como esta, um corretor gramatical associado ao *OpenOffice*. E eu peguei o grupo de professores conhecidos... e aí nós três começamos esse projeto (E3).

O projeto elaborado recebeu o nome de CoGrOO, denominação que traz as letras iniciais das palavras *o*corretor gramatical para OpenOffice.org. Uma vez finalizado, surgiu a oportunidade para submissão desse projeto ao edital de *Software Livre* da Financiadora de Estudos e Projetos (Finep), empresa pública vinculada ao Ministério da Ciência, Tecnologia e Inovação (MCTI), com atuação proeminente no financiamento de projetos com foco no

progresso da ciência, tecnologia e inovação. Apesar de os trabalhos do CoGrOO terem começado, efetivamente, por volta de 2004, antes mesmo da aprovação do edital da Finep, foi com o resultado positivo desse edital que os primeiros grandes passos para a elaboração do corretor gramatical se concretizaram. O aporte financeiro concedido pela Finep viabilizou contratações de bolsistas de estágio e de iniciação científica para o desenvolvimento do *software*, o que aconteceu de forma bastante ativa nos anos de 2004 e 2005. Nesse período, o projeto também contou com o suporte de infraestrutura do laboratório de pesquisa da Escola Politécnica da Universidade de São Paulo (Epusp). Essa dependência inicial do projeto ao aporte financeiro da Finep pode ser notada nos trechos seguintes das entrevistas:

Em 2004, ele teve um financiamento do governo, através do Finep, que foi quando teve o início desse desenvolvimento (E1).

Logo apareceu um edital da Finep, que trouxe recursos para vários projetos de *software* livre no Brasil. Escrevemos o nosso projeto, tivemos a verba aprovada, montamos um laboratório (E3).

No início do ano de 2006, um pouco antes do cronograma estipulado pelo edital da Finep, houve o lançamento da primeira versão do corretor gramatical CoGrOO. Esse lançamento, ainda uma versão do *software* carente de melhorias em sua parte técnica, tal como relatado pelos entrevistados E1 e E3, teve uma recepção significativa por parte dos usuários do *OpenOffice.org*, o que trouxe visibilidade e reconhecimento para toda a equipe envolvida no projeto. No entanto, nesse mesmo ano de 2006, com o término do apoio financeiro da Finep, alguns integrantes que participavam do CoGrOO de forma remunerada começaram a sair do projeto. Sobre essa questão, vale esclarecer que, embora o CoGrOO se caracterizasse como um *software* livre aberto à comunidade, com o código disponível na plataforma *sourceforge*, os indivíduos que atuavam no desenvolvimento dele, nas palavras de E1 ã...eram muito ligados ao projeto, não pelo *software* livre em si, mas, mais pela bolsa... o que atraía as pessoas era a bolsa e não o *software* livre...ö.

Com o término da bolsa concedida pela verba da Finep, seguiu-se um intervalo de tempo em que o CoGrOO se manteve funcionando pelo trabalho de voluntários, o caso de E1, anteriormente bolsista, como também devido ao apoio institucional da Universidade de São Paulo (USP), mediante interfaces com trabalhos e disciplinas acadêmicas. Alguns fatos ocorridos nesse período merecem destaque e são relatados nos parágrafos adiante.

No ano de 2006, por exemplo, dois membros do CoGrOO conseguiram obter bolsas do programa Google *Summer of Code*. A ideia desse programa consistia em fazer com que estudantes e organizações atuantes na área de *software* livre fossem remunerados para escrever código durante o período das férias de verão. O resultado disso é que os participantes

do CoGrOO e o próprio *software* terminaram ganhando atenção em escala global, até mesmo no sentido de ter acesso a novas oportunidades profissionais decorrentes desse prestígio. A importância do programa patrocinado pelo Google foi enfatizada por E3:

Uma coisa legal também que a gente teve, eu não sei se alguém lembrou, é que a gente acabou ganhando dois patrocínios do Google *Summer of Code* [...], que é um cursinho de dois meses, que eles dão uma bolsa, uma bolsa bem interessante de estudos para desenvolvedores de *software* livre. É uma espécie de um patrocínio, para eles poderem trabalhar em prol do projeto CoGrOO (E3).

Além dessa participação de estudantes no Google *Summer of Code*, os integrantes do CoGrOO solicitaram e conseguiram outro apoio do Google, em que recebeu, como incentivo, o benefício da disponibilização de um código linguístico. Todo esse processo, desde o momento da solicitação, se encontra no relato do entrevistado E3:

[...] vamos pedir para o Google, que é uma empresa que incentiva *software* livre. Aí, listamos um monte de pesquisas que a gente podia pedir para o Google e mandamos para o responsável por essas coisas de convênios, de incentivos. Aí eles falaram assim [bom, esse monte de coisa aqui não dá para gente ajudar, mas isso aqui], que era uma licença, um código linguístico [...] A mágica da coisa é que vêm as anotações feitas por linguistas e com essas anotações a gente podia usar os modelos estatísticos treinados e podíamos começar a desenvolver todas as ferramentas em cima disso (E3).

No que concerne especificamente ao apoio acadêmico recebido da USP, destaca-se, além das pesquisas desenvolvidas, o fato de que o *software* também foi utilizado em disciplinas do Programa de Pós-graduação em Ciência da Computação dessa universidade. De acordo com os entrevistados, alguns docentes daquela instituição não só orientaram trabalhos de conclusão de curso e dissertações sobre o *software*, como fizeram uso do CoGrOO em disciplinas ministradas.

Mesmo diante de recursos como os que foram garantidos por meio do Google *Summer of Code* e do apoio institucional concedido pela USP, a comunidade de desenvolvimento do CoGrOO, no decorrer dos anos, sempre passou por momentos de dificuldades. Muitas vezes o corretor ficou sob a responsabilidade de um único desenvolvedor, representado por E1. Trechos das entrevistas com os desenvolvedores E1 e E2 e com o docente E3 retratam essa situação:

Eu comecei no projeto como um aluno de iniciação científica. Nessa fase desenvolvemos a versão 1.0. Mais tarde, desenvolvemos a versão 2.0, como projeto de conclusão de curso. Aí sim, ele realmente funcionou bem, a gente conseguiu fazer uma versão bem mais funcional [...] Nesse tempo, entre 2007 e 2008, eu consegui criar a versão 3, sozinho, não tinha nenhum outro programador (E1).

O CoGrOO já [...] assim, teve fases do projeto de ter um desenvolvimento mais ativo, depois teve fase que ele ficou mais, digamos assim, morno (E2).

Eu acho que a época que ele começou a ser, realmente, usada a ferramenta, foi um ano depois do TCC, do trabalho de conclusão de curso do [nome do entrevistado E1] e do grupo dele, essa versão dois, né? A partir daí, [nome do entrevistado E1] passou a ser o principal mantenedor da ferramenta e nesse meio tempo, durante uns três anos, mais ou menos, quatro anos, o CoGrOO começou a ser razoavelmente bem usado nas empresas, por vários brasileiros, inclusive no Metrô de São Paulo. (E3).

Por ser recorrente a dificuldade de conseguir desenvolvedores dispostos a colaborar voluntariamente, um importante passo dado para que o *software* continuasse tendo a manutenção necessária ocorreu, no ano de 2010, com a incorporação do CoGrOO ao Centro de Competência em *Software* Livre (CCSL), do Instituto de Matemática e Estatística (IME) da USP. A integração ao CCSL trouxe não só o apoio de outros docentes, mas também proporcionou a divulgação do projeto e o acesso à infraestrutura fornecida pelo centro. Esse momento é comentado pelos entrevistados E1 e E3:

Em 2010, assim como aconteceu em 2007, esse projeto foi entrando em decadência de novo, por falta de desenvolvedores, aí nós conseguimos colocar ele no CCSL. Nós tentamos de novo criar uma comunidade de desenvolvedores no torno dele, mas, de novo, a mesma dificuldade. Assim, a gente conseguiu criar uma versão nova, a versão 4.0.

E depois, teve uma época também, que a gente vendeu a ideia lá para o pessoal do IME. O IME é o Instituto de Matemática e Estatística, que fica do lado da Poli... E eles compraram também a ideia e hospedaram lá no CCSL, Centro de Competência de *Software* Livre (E3).

Outro acontecimento significativo dentro do período em que o projeto esteve no CCSL se deu quando numa disciplina sobre *software* livre, cujos esforços foram direcionados para a elaboração de um portal, como forma de consolidar a comunidade de desenvolvedores e de usuários do CoGrOO. No entanto, ainda que a iniciativa do portal tenha sido positiva, em termos de tornar mais estruturada a identidade do CoGrOO, permaneceu o problema de se manter desenvolvedores ativos trabalhando no projeto. Tanto E1 quanto E2, por terem sido alunos daquela disciplina, relatam, com conhecimento de causa, o que houve de fato.

Com a parceria do CCSL, nós tentamos uma outra abordagem, que foi o seguinte, nós criamos o portal chamado CoGrOO comunidade [...] era uma iniciativa de tentar trazer pessoas, os usuários do projeto, para participar. Nós tivemos até uma certa aderência. A ideia era que as pessoas poderiam reportar erros no corretor gramatical. Ela foi bem útil [...] Eu usei bastante isso durante um período que eu estava na USP e usei um subproduto disso na minha dissertação de mestrado, mas eu não consegui reverter isso, de transformar o *software* numa forma mais fluida, como era nosso objetivo inicial (E1).

A disciplina era de *software* livre, que era um projeto de *software* livre, aí, por conveniência e interesse, eu quis me aproximar um pouco mais

concretamente do CoGrOO. Nessa época, eles estavam trabalhando com o desenvolvimento do CoGrOO comunidade, que é o acesso *Web* para comunidade [...] e aí eu trabalhei mais diretamente também para essa disciplina, trabalhando com muitos aspectos sobre o *site*, uma página de estatística, trabalhando um pouco para coletar informações sobre o *site*, sobre se o *site* era utilizado e um pouco também na área de usabilidade (E2).

Fruto de esforço em comunidade, é inegável que mesmo se tratando de um *software* livre e havendo conseguido alcançar vasto número de usuários interessados no seu desenvolvimento, com o término da remuneração dos bolsistas, concedida pela Finep, o CoGrOO se deparou com períodos de dificuldades. Apesar das inúmeras tentativas acadêmicas que aconteceram na USP, o declínio do número de desenvolvedores foi, por várias vezes, uma realidade adversa ao projeto. A respeito destas fases difíceis, E1 relata:

O *open source*, na minha visão, ele funciona melhor quando os usuários são também os desenvolvedores. No nosso caso, ele era um corretor gramatical e as pessoas que usavam ele, na maioria das vezes, não eram programadoras e não poderiam fazer parte, diretamente da comunidade de desenvolvimento. Então não tinha uma troca, a gente desenvolvia um *software* para o usuário final e eles não contribuíam de volta com código (E1).

Após o período em que o CoGrOO esteve no CCSL/IME/USP, houve um afastamento natural dos demais desenvolvedores e docentes que vinham atuando no projeto. É partir desse ponto que E1 assume um papel central para que o CoGrOO continue se mantendo ativo e recebendo a manutenção necessária. O nível de conhecimento dele sobre o projeto e a sua continuidade, como será possível notar nas seções sobre os aspectos de governança, tornou o seu relato central para o entendimento do caso CoGrOO. Na fala do entrevistado E3 fica clara a importância que E1 teve e permanece tendo para o projeto:

[Nome do entrevistado E1] foi o mais destacado desenvolvedor. Apesar de não ter sido o primeiro foi o que teve mais tempo de desenvolvimento, né? Depois ele fez o trabalho de conclusão de curso, e aí depois ele continuou, ele fez o mestrado em cima disso, mesmo depois do mestrado, ele continuou ainda mantendo forte [...] Ele foi o desenvolvedor mais destacado [...] acabou assumindo mesmo o papel de gestão que era meu, eu acabei naturalmente transferindo [...] não formalmente, mas naturalmente transferindo, porque o projeto ficou numa fase que virou quase que um órfão (E3).

Por volta de 2010, com a responsabilidade de continuar mantendo o *software* funcionando, E1 observou que outro projeto do qual participava, denominado OpenNLP, apresentava algumas importantes interfaces com o projeto CoGrOO, ainda que não fosse direcionado para o usuário final.

Desde 2010 eu fui percebendo essa questão, de que o projeto não iria vingar como um *software* livre. Aí eu entrei nesse outro projeto, que é o Apache

OpenNLP. Desde 2010, começa então a ter esses dois projetos [...] Ele não é tão simples de utilizar. Diferente do CoGrOO que era voltado para usuários finais, o OpenNLP é uma biblioteca [...] para usar ele tem que ser programador (E1).

Vale ressaltar que E1 teve a oportunidade de conhecer profundamente o OpenNLP, presenciando, inclusive, algumas transformações que ocorreram no projeto, a exemplo de quando esteve um período incubado na Apache e, posteriormente, se tornou um projeto oficial dessa fundação. Para se tornar um projeto Apache, vale destacar, existe toda uma formalização que envolve submissão de proposta a edital específico, bem como, no caso de aprovação do projeto, a exigência de um período de incubação para amadurecimento, até que seja realmente considerado um projeto comunitário pertencente à fundação.

Naquela época o OpenNLP ainda fazia parte da *sourceforge* e era independente [...] Eu comecei a mandar os *patches* [...]. Então a gente submeteu o projeto para incubação na Apache. A gente foi incubado [...] Essa história da Apache começou mais ou menos em 2011. Ele foi depois promovido a *top level*, como eles chamam, quer dizer, saiu da incubação e foi para a parte *top level* (E1).

Por ter tido a chance de conhecer de forma aprofundada o OpenNLP, assim como de perceber alguns pontos de convergência entre ele e o CoGrOO, E1 vislumbrou uma oportunidade para garantir que o CoGrOO continuasse operando. Por iniciativa individual desse desenvolvedor, cerca de 90% do código do CoGrOO foi incorporado ao OpenNLP, por meio de um processo de correções conhecido como *patch*. Com a transferência do código e devido ao fato de o OpenNLP, desde 2011, ter se tornado um projeto Apache, a gestão desse código passou, então, a contar também com a infraestrutura da referida fundação, em vez de permanecer sob a responsabilidade de um único desenvolvedor, como vinha acontecendo antes. Esse processo é explicado por E1:

Um dos planos que eu tive foi o seguinte [...] o CoGrOO era um projeto gigante, tinha muita coisa e era muito difícil de manter. Aí, eu conheci esse projeto o OpenNLP, que era um projeto que tinha uma parte de interseção. O que o CoGrOO faz a mais é a parte base da análise gramatical. Então, o que eu fui fazendo desde 2010? Eu fui pegando tudo que o CoGrOO tinha e que o OpenNLP não tinha e fui passando para lá, através de um processo que se chama *patch* [...] até que a comunidade do OpenNLP me chamou para fazer parte dela. Então eu me tornei *committer* desde 2010 (E1).

O CoGrOO e o Apache OpenNLP, no entanto, como já ficou dito, permanecem existindo como iniciativas independentes, ainda que a gestão do código do primeiro aconteça de forma dependente do segundo projeto. O CoGrOO que antes era acessado principalmente pelo *sourceforge* e fazia uso da licença *Lesser General Public License* versão 2.0 (LGPLv2),



desde sua última versão, até os dias de hoje (versão 4.3.1), se encontra disponível na plataforma *GitHub* e teve a sua licença modificada para a Apache 2.0.

Só para você entender como o CoGrOO ficou no final das contas, exatamente, depois que eu coloquei boa parte do código, tipo 90% do código para o OpenNLP [...] o CoGrOO ficou pequenininho e hoje ele é mais simples, na verdade, é bem mais simples e mais enxuto, pois só faz a parte de correção gramatical. E o OpenNLP, todo mundo que tem me perguntado o que que eu preciso para fazer processamento de linguagem em português, hoje eu indico o OpenNLP e não mais o CoGrOO (E1).

É por conta do encontro dos dois projetos, tal como explicado em detalhes anteriormente, que de modo diferenciado dos demais casos focalizados por esta tese, a governança, na perspectiva da estrutura e dos processos de controle, será analisada considerando o projeto CoGrOO individualmente, e o projeto OpenNLP, também em separado. Não se deve deixar de considerar que parte do código do CoGrOO passou a ser incorporado ao segundo projeto, fato este, pela sua relevância e desdobramentos, que veio a justificar a extensão das análises dedicadas a ambos, superando os espaços concedidos aos demais casos.

#### 3.4.1.2 Estrutura do projeto CoGrOO

No período de funcionamento individual do CoGrOO, a estruturação para a produção do corretor gramatical aconteceu bastante atrelada, pelo menos de início, ao edital da Finep e, em momento posterior, ao movimento acadêmico de disciplinas e publicações que aconteceram sobre o projeto.

Na fase de imersão acadêmica, a hierarquia do projeto se adequou ao papel formal existente e que regula a relação entre docentes e os estudantes vinculados a eles. Sendo assim, enquanto os docentes atuavam mais na gestão do projeto em si, os discentes tiveram participação maior no desenvolvimento e nas atividades operacionais voltadas para manter o projeto funcionando. Em relação à hierarquia, o docente E3 observou:

Existia uma hierarquia natural. Eu fui o idealizador aqui do projeto. Mas, no meio acadêmico, você tem que respeitar algumas hierarquias naturais. Eu nunca fui professor na Poli. O professor [nome do professor] era um professor doutor lá na Poli, então a gente tinha que usar essa hierarquia. A gente montou um laboratório, o professor [nome do professor] seria o coordenador natural do projeto, principalmente perante os órgãos de fomento, que eles valorizam muito isso, claro. Então precisou ter essa hierarquia. E eu, da minha parte, eu tinha uma tarefa, que era orientar os alunos, mas não era aquilo distante. Era uma coisa muito próxima. Eu

sempre tive uma comunicação muito fácil com os alunos, principalmente por ser professor [...] não professor na Poli (E3).

Na opinião de E2, que atuou como um dos desenvolvedores do projeto, existia certa hierarquia também em função dos conhecimentos de cada indivíduo, que nem sempre era percebido de modo claro: ãEu acho que não explícita, né? Existia todo o tipo de experiência, então era natural que eu buscasse sempre a opinião das pessoas que tinham um contato maiorö.

A prevalência de uma estrutura hierárquica formal na relação docente/aluno ficou explícita quando o edital da Finep estava em vigência, já que os estudantes que trabalhavam como desenvolvedores do projeto assumiram o papel formal de bolsistas. Os docentes atuavam como gestores do projeto e os alunos exerciam a função de desenvolvedores, seja como bolsistas, quando recebiam remuneração viabilizada pela Finep, seja como voluntários, quando o CoGrOO, após o término das concessões das bolsas, passou a contar inerentemente com o apoio acadêmico para se manter ativo.

Apesar de nessa visão macro os papéis de gestão e dos desenvolvedores acontecerem de forma bem definida, na prática, principalmente na equipe de desenvolvimento, os integrantes poderiam realizar múltiplas funções, de acordo com as necessidades do projeto. Assim, ainda que cada módulo a ser desenvolvido fosse dividido em áreas com atribuições de tarefas aos membros, conforme a necessidade, esses membros poderiam ajudar uns aos outros nas atividades que deveriam ser finalizadas. No decorrer do tempo, várias frentes envolvendo diversas funções foram estabelecidas, a exemplo de atribuições para tirar dúvidas sobre o programa, para a divulgação do *software*, dentre outras, que precisavam de pessoas colaborando de forma não restrita a um único papel. O entrevistado E3 tangenciou na sua resposta essa questão dos papéis existentes no CoGrOO:

A gente tinha algumas frentes de trabalho naquela época e um dos grandes desafios que a gente tinha era fazer integração com o *Office*. Então, a gente tinha um esforço grande de como integrar a ferramenta com o *OpenOffice*. Então, tinha tarefas bem definidas, quem iria cuidar do motor de conversão, quem iria cuidar das regras, tinha papéis bem definidos (E3).

Um papel bem delimitado e que sempre esteve presente no CoGrOO foi o da liderança, não só a representada pelos docentes que atuaram na gestão do projeto, em especial, quando ele manteve vinculação com a Finep, como também aquela que emergiu após o término da execução do edital, e que se concentrou em torno da figura de E1. Em toda a trajetória do CoGrOO, portanto, alguma liderança se fez presente no sentido de guiar os

esforços para o alcance dos resultados almejados. Os fragmentos das entrevistas apresentam as percepções sobre a liderança existente no CoGrOO:

O professor [nome do professor] era atuante mesmo, tanto que a primeira versão do CoGrOO foi ele que escreveu. Então ele tinha sim um papel de coordenação, e eu tinha um papel mais próximo dos estagiários. Eu sentava com eles, via o que eles estavam fazendo, via os resultados, analisava os resultados (E3).

E1 sempre foi o líder, a meu ver. Eu sei que antes de eu ter contato, teve também o professor [nome do entrevistado E3], mas eu não tive um contato muito direto, porque ele se afastou do projeto. Mas ele que iniciou mesmo a iniciativa desse trabalho, imagino que ele tenha tido um papel de liderança um pouco mais ativo, acabou se distanciando e [nome do entrevistado E1], como o principal programador, era o líder prático (E2).

No que tange à estrutura decisória, a equipe do CoGrOO sempre buscou conversar e ouvir diferentes opiniões, no sentido de tentar o consenso antes de ser tomada alguma medida. No entanto, ainda que houvesse essa abertura com os participantes usufruindo de liberdade para externar suas impressões, as decisões que não encontravam consenso, ou que eram mais estratégicas, terminavam se concentrando naqueles indivíduos com maior experiência, representados pelos docentes ou desenvolvedores que estavam no CoGrOO desde o início. Quando o projeto, ao longo dos anos, sofreu o revés do afastamento dos docentes envolvidos na gestão, várias das questões sobre o futuro do CoGrOO ficaram concentradas na figura do desenvolvedor E1. Os trechos das entrevistas que corroboram este fato são apresentados a seguir:

A gente procurava ouvir a opinião de todo mundo, inclusive dos alunos, porque, como eles estavam com a mão na massa, às vezes eles tinham uns *insights* ali que eram muito bons. Então a gente se reunia, principalmente o pessoal dessa parte técnica (E3).

Acho que cada um opinava e se não houvesse um consenso, teria um pouco dessa hierarquia, e a gente menos experiente, respeitaria ou acataria a decisão de outras pessoas. Mas dificilmente eu diria que houve um conflito, em que realmente eu não concordei. Mas, de forma geral, a gente tinha uma coisa mais [...] tipo um consenso (E2).

#### 3.4.1.2.1 Estrutura do projeto CoGrOO: interface com o projeto OpenNLP

Com a incorporação de parcela significativa do código do CoGrOO ao projeto OpenNLP, realizada pelo desenvolvedor E1, a gestão desse código passou também a contar com o suporte da fundação Apache.

Em termos de infraestrutura, a Apache forneceu ao OpenNLP, assim como fornece aos demais projetos que dela fazem parte, todo um aparato técnico e profissional, que envolve

desde ferramentas necessárias ao desenvolvimento do *software*, até a assistência de uma equipe multidisciplinar de profissionais. Essa é a forma que a Apache tem de garantir que os projetos de *software* livre permaneçam existindo, independente da participação dos voluntários individuais. Este foi o entendimento passado por E1.

A Apache fornece umas ferramentas profissionais de desenvolvimento, que são muito caras. A gente, por exemplo, usa para programar o *IntelliJ*. O *IntelliJ* é uma ferramenta paga, bem cara, que é fornecida pela Apache [...] Nós temos suporte de infraestrutura, suporte jurídico, suporte de *marketing*, desse tipo de coisa. Nossa marca é registrada, a Apache registrou [...] Então, a Apache tem uma equipe de TI, que ajuda a gente a manter o *site* no ar, todo o sistema funcionando. O dinheiro que a Apache recebe, acaba indo muito para isso [...] A Apache tem que pagar advogado [...] Porque essas pessoas não são voluntárias. Um advogado não vai ser voluntário. Tem que pagar uma secretária, tem que pagar pessoas do suporte. Então a Apache é como uma empresa mesmo [...] E termina dando suporte para as comunidades (E1).

Dentre as TICs usadas pelo OpenNLP encontram-se a *home page* do projeto, as listas de *e-mail*, a plataforma de hospedagem do código-fonte e o *chat*. O *chat*, convém destacar, é usado de forma acessória e em situações específicas, já que a Apache solicita que toda a comunicação referente ao projeto aconteça preferencialmente pelas listas abertas de *e-mail*, conforme ficou destacado por E1:

Na Apache, a gente tem que se comunicar por *e-mail*, tem que ser tudo aberto. *E-mail* público é o método principal de comunicação. Nós quebramos um pouco essa regra e usamos também o *Slack*. O *Slack* é um sistema de *chat*. Então, toda comunicação pública tem que ser via *e-mail* (E1).

O OpenNLP apresenta uma hierarquia formal que é resultante dos papéis estabelecidos previamente entre os seus integrantes. Além dessa hierarquia formal, existem também relações de autoridade que não se amparam na subordinação hierárquica, acontecendo em função dos conhecimentos adquiridos pelos membros, principalmente por aqueles que são mais antigos e detêm maior conhecimento sobre o projeto. Essa perspectiva lateral de autoridade se fundamenta na própria filosofia de meritocracia que a Apache busca propagar. Nela, os membros que se destacam pela realização de contribuições importantes e pelo compromisso nos projetos em que atuam, seriam, naturalmente, aqueles escolhidos para assumir responsabilidades maiores nos agrupamentos. No projeto OpenNLP é possível notar a coexistência dessas duas formas de hierarquia, na palavra do entrevistado E1:

Hoje são 23 *committers*, nem todos ativos [...] Além dessa hierarquia formal, existe também no projeto uma hierarquia que é função do conhecimento adquirido no decorrer da existência do projeto [...] A gente tem os desenvolvedores mais antigos [...] Em geral a gente acaba tendo uma

autoridade [...] A gente é responsável pela visão mais crítica das coisas. Então se uma pessoa vem com uma ideia muito mirabolante, a gente acaba tendo que entender e conversar, para ver se a gente segue essa linha ou não. Em geral, o [nome da pessoa] faz isso muito bem, eu ajudo ele, e cada vez mais outras pessoas estão pegando [...] Eu e ele somos mais antigos, mas tem algumas pessoas que já estão pegando também isso (E1).

De modo aderente à hierarquia existente no OpenNLP, alguns papéis se encontram delimitados no projeto. Diga-se, a propósito, que além dos papéis comumente encontrados nas comunidades de *software* livre, tais como os de usuários ou de desenvolvedores, outras posições funcionais são estabelecidas pela fundação Apache, como demonstra o Quadro 7 a seguir.

**Quadro 7** ó Papéis identificados nos projetos de software livre comunitários da Apache

<b>Papéis</b>	<b>Perfil</b>	<b>Responsabilidades</b>
Membros da ASF (Apache <i>Software Foundation</i> )	Indivíduos indicados pelo Conselho atual da ASF.	Possuem o direito legal de eleger o Conselho, se candidatar às eleições do Conselho e propor comitê de membros ou projetos para a incubação.
<i>Chair</i> do PMC	Indivíduo nomeado pelo Conselho da ASF para presidir os membros do PMC.	Atua como os demais membros do PMC, no entanto, com a incumbência adicional de entregar relatórios trimestrais sobre o status do projeto ao Conselho da ASF.
Membros do PMC ( <i>Project Management Committee</i> )	Indivíduos representados por <i>committers</i> , indicados pelos outros membros do PMC.	Realizam a governança da comunidade, fazendo a supervisão e o monitoramento legal do projeto junto à ASF.
<i>Release manager</i>	Indivíduo escolhido dentre os <i>committers</i> , podendo existir rotatividade desse papel no projeto.	Assume a condução do processo de liberação de uma nova versão ou pacote do <i>software</i> . A responsabilidade final da liberação, no entanto, é do PMC e do <i>chair</i> .

**Fonte:** Elaborado pela autora, com base nas informações disponíveis no *site* da Apache (2019)

Os papéis regulamentados pela Apache ressoam no OpenNLP. Nesse projeto, no entanto, é possível notar que apesar do *chair* formalmente se posicionar como o líder do projeto, a rigor não necessariamente exerce esta liderança. O *chair* do OpenNLP, também conhecido como vice-presidente (VP), termina assumindo papel mais burocrático, uma vez que dentre as suas competências se encontram: o envio dos relatórios trimestrais sobre o status do projeto, exigidos pelo Conselho da ASF; o gerenciamento das reuniões e listas de *e-mail* da comunidade; o estabelecimento de regras e procedimentos das atividades pelas quais os demais membros do PMC são responsáveis. Em outras palavras, o *chair* tem uma atuação mais expressiva no sentido de estabelecer a interface entre o OpenNLP e o Conselho da ASF, bem como com vistas a garantir o cumprimento das questões legais definidas pela fundação.

Em trechos da entrevista com E1, apontam-se compromissos assumidos pelo *chair* do OpenNLP.

Em 2011, 2012, quando a gente entrou na Apache, a gente teve que eleger uma pessoa que a gente chama de *chair*, a gente chama também de VP, que é vice-presidente do projeto. Esse papel dele é muito mais burocrático do que de gestão, por exemplo. Porque é responsabilidade dele gerar relatórios, às vezes iniciar uma votação, é mais burocrático do que propriamente gerencial [...] Ele é responsável por enviar o relatório. Então a cada 3 meses, ele tem que enviar um relatório do projeto. O que é a gente faz, a gente cria o relatório com ele, ele abre no Google *Drive* lá um documento, monta ou escreve o relatório, ou alguém monta para ele, a gente vai colocando as coisas e ele fica para enviar (E1).

O *chair* assume a função de um líder mais burocrático e a liderança no projeto OpenNLP acaba acontecendo de forma mais compartilhada, assumindo tal incumbência os integrantes que demonstram ter maior habilidade no sentido influenciar o comportamento dos participantes do agrupamento. No período em que a entrevista com E1 foi realizada, um dos integrantes do OpenNLP vinha exercendo esse papel de liderança, agindo de forma a motivar os membros do coletivo e a direcionar os rumos do projeto para a consecução dos resultados esperados. E1 aborda um pouco este aspecto da liderança compartilhada na entrevista:

A liderança do projeto ela é muito mais compartilhada. Tem um cara, um dos últimos *committers*, ele tem um papel de líder até maior. Só que é uma coisa bem autônoma, ele não tem nenhum cargo, nenhum título no projeto para ser o líder. Ele, por exemplo, incentiva novas *releases*, incentiva força-tarefa para revisar *bugs*, mas isso é uma tarefa muito compartilhada, ora um, ora outro faz isso, não temos uma figura que faz isso somente (E1).

O papel de membro do PMC, a seu turno, se converte numa formalização a mais da função do *committer*, de modo que além de ter acesso ao repositório de gravação do código, o integrante também ganha o direito de votar nas decisões referentes à comunidade e de propor que outros *committers* se tornem integrantes do próprio PMC. No OpenNLP, quando o desenvolvedor que já é *committer* e continua se mantendo ativo no projeto, de modo a fornecer contribuições relevantes, ele pode ser convidado para se tornar integrante do PMC, conforme relata E1:

Depois de se tornar *committer*, a gente vai ficar de olho nessa pessoa por mais um tempo, então, se essa pessoa estiver com contribuições frequentes, depois de 1 ano, a gente convida ela para fazer parte do comitê de gestão, ela se torna PMC do projeto (E1).

O desempenho do papel do *release manager*, por sua vez, costuma ser alternado entre os *committers* do OpenNLP, o que possibilita que vários integrantes tenham a oportunidade de alcançar a visibilidade proporcionada por essa posição junto à Apache. Existe

a preocupação que todos os *committers* em determinado momento assumam essa responsabilidade de fazer o lançamento das versões, tal como afirma E1:

A gente tem um papel que é o de *release manager*, que é o cara que vai ser responsável pela *release*. Esse papel a gente rotaciona. Então toda vez que vai ter uma *release*, a gente pergunta quem quer fazer, é rotacional. Eu já fiz algumas vezes [...] só que depois de ter feito várias vezes eu prefiro não fazer e então dar oportunidade para outras pessoas. Então, geralmente, a gente puxa pessoas novas para fazer isso (E1).

Apesar de ser algo comum nos projetos de *software* livre, o papel de *committer*, no OpenNLP, é a base para que o integrante possa assumir outras funções definidas pela Apache no projeto. Entretanto, há um detalhe: formalmente estabelecido e com responsabilidades demarcadas pela Apache, que vão além do acesso à gravação no repositório do código, os critérios para que determinado desenvolvedor se torne *committer* são fluidos e estão mais relacionados ao mérito do indivíduo do que a qualquer outra coisa. Essa questão é focalizada por E1:

O papel de *committer* por sua vez, surge por meio do reconhecimento do trabalho do integrante do projeto. Ao mandar contribuições relevantes para o código, esse indivíduo pode ser convidado por qualquer outro *committer* do grupo, para se tornar também *committer* [...] O papel da pessoa contribuir com o projeto enviando um *patch* [...] um *patch* é uma correção de *software*, colocando uma nova funcionalidade, alguma coisa assim. Se o *patch* for bom a gente fica de olho na pessoa [...] ela manda outro *patch*, a gente continua de olho. No segundo ou terceiro a gente já pode convidar a pessoa, ou até no primeiro, dependendo [...] se a contribuição for grande e consistente para entender que aquela pessoa é boa, o processo envolve uma votação entre os PMCs e posteriormente um convite (E1).

Além desses papéis apresentados, outras funções podem ser assumidas por determinado membro do coletivo, concomitantemente as que foram formalmente atribuídas a ele. Sendo assim, é fato que os integrantes do projeto não se limitam a um único papel no âmbito do empreendimento. Por exemplo, quando um indivíduo desenvolve determinado módulo, é comum que se torne uma espécie de gestor daquele módulo, por ser a pessoa que detém o maior conhecimento sobre ele. Outros membros, de acordo com os seus interesses particulares, podem livremente optar por atuar na elaboração de textos técnicos e artigos acadêmicos que ajudem na divulgação do projeto, e assim por diante. Ou seja, além dos papéis prévios definidos pela Apache, cada integrante sempre tem a seu dispor a oportunidade de assumir outros encargos informais, mas igualmente importantes para o funcionamento do projeto. Esse aspecto foi abordado por E1:

A comunidade acaba tendo pessoas que se especializam em alguma coisa, por exemplo, tem um *committer* que é um ótimo programador, então ele acaba tendo um papel, às vezes, de otimização de código, mas não é nada

formalizado, ele contribui com o projeto dessa forma. No projeto existem professores universitários que acabam contribuindo muito com coisas mais inovadoras, quando a gente vai apresentar um *paper*, eles são melhores de escrever o *paper*. A gente já teve um cara que trabalhava na NASA que contribuiu e tudo mais, tem várias empresas assim do porte do Google (E1).

Os projetos comunitários da Apache são iniciativas movidas principalmente pelas tomadas de decisões dos seus integrantes, porque há autonomia no desenvolvimento do *software* e pelo fato de o coletivo prover da liberdade de projetar as suas próprias regras de governança, desde que estejam de acordo com a política da fundação. As tomadas de decisão dos projetos Apache se inclinam, preferencialmente, pelo uso de processo conhecido como reuniões de consenso.

Na tentativa de alcance do consenso, as questões que exigem o estabelecimento de votação são operacionalizadas por essa regra, pelas listas de *e-mail*, de modo que números são utilizados na quantificação dos votos da seguinte forma: +1, quando o voto é positivo; 0, quando o indivíduo quer se abster do voto ou não tem opinião; -1, quando o voto é negativo. É a partir da contagem resultante da pontuação dos votos que geralmente se chega ao consenso. Na ocorrência de pontuação negativa, a Apache solicita que seja descrito no *e-mail* uma proposta alternativa, ou uma explicação detalhada das razões que fundamentaram o voto. Em alguns casos específicos, a Apache tem regras de votação mais detalhadas, tal como acontece no lançamento de novas versões do *software*, quando os membros do PMC possuem votos vinculantes, ou seja, votos que precisam acontecer para que a liberação da nova versão seja efetivada. Assim, apesar da busca do consenso, é possível notar que nas decisões mais relevantes, os integrantes que atuam na gestão do projeto é que possuem maior poder opinativo.

#### 3.4.1.3 Processos de controle do projeto CoGrOO

De origem concebida nos moldes de um projeto de *software* livre com abertura comunitária, o CoGrOO, em vários momentos da sua trajetória operativa esteve vinculado a uma instituição acadêmica, no caso, a USP. Assim, nesse período, recebeu e internalizou a influência de processos formais de controle, em particular do controle praticado pelos docentes sobre os discentes atuantes no desenvolvimento do corretor gramatical.

A partir do instante em que o CoGrOO passou a se beneficiar do financiamento previsto no edital da Finep, formalizações foram necessárias, tal qual a contratação de estudantes bolsistas remunerados. Na corrente destes contratos, como é normal ocorrer, uma



série de especificações haveriam de ser cumpridas, tanto pelos docentes, no sentido de prestar contas por meio dos relatórios exigidos pelo órgão de fomento, quanto pelos bolsistas, uma vez que estes também precisavam cumprir objetivos estabelecidos, dentro do cronograma estipulado, informando o andamento de atividades nas reuniões de acompanhamento. E3 refere-se a esses aspectos na entrevista.

A gente tinha, na verdade algumas coisas bem claras que eram [...] estava lá no escopo do projeto Finep. Isso era bem claro para gente, bem formal, bem claro, mas tudo o que veio depois foi acontecendo informalmente. Inclusive essa iniciativa do Google, tudo foi uma coisa bem informal e foi se fazendo... Nós fizemos mais um desenvolvimento tipo catedral do que bazar [...] Até por conta de Finep no meio e tudo, a gente não podia simplesmente fazer um negócio muito [...] O desenvolvimento do tipo bazar funciona, mas não quando você tem patrocinadores ali atrás, que você tem que prestar contas para os financiadores (E3).

Encerrado o apoio da Finep, embora apenas uma parte dos estudantes tenha permanecido no projeto como voluntários, o interessante a frisar é que a relação formal passou a ser estabelecida em função das disciplinas e das orientações de pesquisas acadêmicas que envolveram o CoGrOO. Nessas disciplinas e pesquisas, entregas específicas (relatórios, apresentações etc), reuniões de acompanhamento e datas eram solicitadas pelos docentes para que os alunos operacionalizem suas tarefas cumprindo prazos definidos.

No transcurso da pesquisa de campo, especialmente das entrevistas realizadas, foi perceptível que tanto por ocasião do apoio da Finep, quanto no período das pesquisas e disciplinas acadêmicas envolvendo o CoGrOO, o controle formal preponderou e esteve mais orientado para os resultados da equipe desenvolvedora do projeto e menos propenso a instituir regras comportamentais para os membros do coletivo.

Nada obstante, deve-se reconhecer que existiram várias situações de controle fora desse contexto, em que a gestão do CoGrOO se pronunciou de modo informal. Esta informalidade ganhou contornos muito nítidos nos momentos de interação que aconteciam entre os membros do projeto, principalmente entre indivíduos da equipe de desenvolvimento. Tal como relatado por E2, tais circunstâncias se associavam à necessidade de discussão dos rumos do empreendimento ou então surgiram para resolver algum problema relevante, na maior parte das vezes, acontecendo de forma virtual:

A gente sempre conversava com alguma frequência, a gente discutia sobre os próximos rumos, discutia o que a gente deveria trabalhar, então, às vezes, vamos dizer assim, tinham grandes discussões mais filosóficas, para onde o projeto quer ir, onde a gente quer seguir, a gente conversava [...] A gente não tinha uma periodicidade. Então a gente quando tinha algum problema, alguma questão, a gente conversava por *chat* e discutia o problema [...] Às

vezes discutia por *e-mail* também, quando as pessoas estavam ocupadas ou tinha que envolver outras pessoas (E2).

A interação com o público externo e com a ampla comunidade de usuários do corretor, traço, então, reconhecido por todos os entrevistados na pesquisa, com frequência acontecia mediante eventos, publicações e entrevistas em que os membros do CoGrOO tiveram a oportunidade de participar durante a fase de maior sucesso do projeto. O próprio portal do CoGrOO foi uma criação para alavancar a interação com a comunidade externa e atrair novos usuários e desenvolvedores. Nesse portal, fez-se a divulgação da identidade visual do corretor gramatical por meio da sua logomarca. Os integrantes do projeto, representados pelo docente E3 e desenvolvedor E2, afirmaram que mesmo após terem se afastado do coletivo, permaneceram tomados por um senso de pertencimento ao CoGrOO. No caso de E3, esse vínculo foi e é ainda mais forte, em virtude da ideia inicial do corretor ter começado por sua iniciativa. E2, por sua vez, destaca, além da importância que o CoGrOO teve na sua vida profissional, que até hoje ainda costuma ajudar usuários e a responder dúvidas sobre o projeto. Essas declarações são confirmadas nos trechos a seguir:

Na verdade, eu tenho uma certa paternidade no projeto. O empreendedorismo inicial foi da minha parte [...] Eu que trouxe a ideia e juntei as pessoas. Acho que esse foi o meu principal mérito, que foi juntar as pessoas certas, as pessoas que tinham vontade de fazer. Eu acho que também eu vendi a ideia [...] Consegui colocar as pessoas corretas, passei o que eu sabia também para os alunos e aproveitei o talento deles também, porque eles eram alunos muito talentosos. [Nome do entrevistado E1], especialmente, era muito talentoso. Apesar de eu não escrever uma linha de código, eu estava lá orientando eles, eu estava ajudando, às vezes corrigindo algumas coisas. Então eu me considero o pai da criança também (E3).

Eu não considero que eu tive, assim, uma contribuição muito significativa. Mas eu sempre procurei ajudar, então, além de programação propriamente dita, a gente conversava, dava ideia. Também tinha outra atividade que eu sempre procurei fazer e ainda faço até hoje, que é ajudar as pessoas que estejam querendo utilizar [...] Teve um momento que a gente saiu na revista Broffice, aliás, eu ajudei um pouco na redação, revisão lá do artigo. E foi uma coisa que ficou marcada em muitos momentos e é um projeto muito grande, pelo menos em número de usuários, uma coisa sempre interessante na minha carreira profissional [...] Também teve o desenvolvimento com o *software* livre [...] Então isso é uma coisa que só agrega na minha carreira (E2).

Na fala de E1, principal desenvolvedor, fica evidente que o CoGrOO trouxe não só uma visibilidade para os integrantes que trabalharam no projeto, mas também foi um sucesso em termos da comunidade de usuários que conseguiu atrair e reunir.

O CoGrOO acabou sendo muito usado no Brasil pelos usuários do *OpenOffice*. Ele teve períodos com 2000 *downloads* por semana. Então a comunidade de usuários dele foi muito grande, eu fui convidado para dar

palestras [...] Além da comunidade de usuários do *OpenOffice*, teve muitos usuários da academia, como alunos da PUC do Sul, Paraná [...] Ele é usado por alunos da USP (E1).

O controle entre pares, embora posto em prática de uma forma pouco definida, era algo habitual nas atividades de desenvolvimento do *software*. Como forma de garantir a qualidade do corretor gramatical, os membros do grupo se ajudavam entre si e realizavam testes do código, como meio de revisar o trabalho dos demais membros da equipe. E3 mencionou como essa revisão e testes aconteciam:

Tinha alguns momentos que a gente olhava os resultados, né? A gente fazia testes e avaliava o desempenho dos algoritmos de acordo com as bases que a gente tinha estabelecido. Por exemplo, a gente montou uma base de erros para facilitar o teste automatizado. Aí você fazia umas modificações e testava naquela base e via o quanto que ele acertou na correção, ou o quanto que ele errou, e com base nisso a gente avaliava (E3).

Outra forma de controle informal identificada no CoGrOO, bastante influenciada pelo reconhecimento e identificação dos membros com o projeto, é o autocontrole. Essa forma de proceder devido às várias oscilações do quantitativo de desenvolvedores à disposição do projeto, após o término do financiamento obtido da Finep, se mostrou essencial para que o CoGrOO persistisse funcionando no decorrer do tempo. Sendo assim, os desenvolvedores que permaneceram atuando como voluntários nas diferentes fases do projeto, ainda que tivessem autonomia para a realização das atividades, seja por interesses pessoais, seja profissionais, por meio do autocontrole na realização das contribuições ao CoGrOO, mantiveram o *software* ativo. Um deles, inclusive, como já mencionado, o integrante E1, mesmo quando houve o distanciamento dos demais membros, até hoje, de forma voluntária, mantém a preocupação de ver o *software* operando. Esse exemplo de autocontrole constou na fala do entrevistado E2, ao resgatar que o trabalho no CoGrOO era bem autônomo e com caráter voluntário. Era praticamente proatividade da minha parte...ö.

#### 3.4.1.3.1 Processos de controle do projeto CoGrOO: interface com o projeto OpenNLP

Com a incorporação do código do CoGrOO ao OpenNLP, e, na sequência, a integração deste último projeto à fundação Apache, o exercício do controle passa a assumir conotações diferentes. Até mesmo pelo porte da Apache, que comporta mais de 350 iniciativas de *software* livre no seu portfólio de projetos, o controle formal permeia praticamente todos os processos principais necessários para manter certo padrão de

funcionamento, de forma a garantir que a operação dessas iniciativas aconteça dentro dos preceitos legais da fundação.

Na própria definição da estrutura (apresentada na seção 3.4.1.2.1), é possível notar que a Apache delimita processos de controle formais sobre como deve acontecer a ocupação dos papéis definidos pela fundação, bem como sobre as responsabilidades que se encontram atreladas a cada um desses papéis. Para a função do *chair*, por exemplo, figura que tem a incumbência de entregar ao Conselho da ASF os relatórios trimestrais sobre o status do projeto, o *site* da Apache disponibiliza um *template* com todas as orientações e informações que devem estar presentes no conteúdo do relatório. Em outro processo, vinculado às atribuições do *release manager*, a Apache disponibiliza em sua página, para o *committer* que assumiu essa responsabilidade, todo o detalhamento das ações que devem ser executadas para o lançamento de novas versões do *software*. Os comportamentos que refletem a estrutura do processo decisório, também são estabelecidos no *site* da Apache ([www.apache.org](http://www.apache.org)). A busca do consenso apoia-se em todo um procedimento traçado sobre a forma de como alcançá-lo, em particular nas reuniões de votação promovidas.

Junto aos esquemas de controle dos papéis e do processo decisório dos projetos da Apache, a página da fundação disponibiliza outros conteúdos que focalizam, diretamente, o comportamento que se espera dos integrantes dos projetos. Trata-se do código de conduta, cuja ideia central é evitar discrepâncias na forma de agir dos participantes dos projetos, que podem ocorrer até mesmo em virtude das diferentes nacionalidades e culturas dos participantes do empreendimento. As regras são válidas não só para todos os canais de comunicação virtuais utilizados pelos projetos, mas também para os encontros presenciais. Determinados procedimentos de controle deliberados pela Apache são bem visíveis no OpenNLP, conforme relata E1:

Quando a gente entra na Apache, temos que seguir as regras que estão todas explícitas lá no *site* da Apache [...] É enorme, uma coisa gigante. Mas são regras do tipo, qual é o critério de você se tornar um PMC? Qual é o critério de você virar um *committer*? A gente nunca precisou fazer isso, mas qual é a regra de tirar? Como é que você aprova uma *release*? (E1).

O entrevistado E1 destacou, além desses procedimentos estabelecidos pela fundação, que o OpenNLP também define, previamente, os seus procedimentos internos de controle, tal como acontece no lançamento de novas versões:

Por exemplo, uma *release*, tem que ficar aberta para votação durante 72 horas. Tem que dar a chance de todo mundo testar. Testando, tem que ter pelo menos 3 PMCs, que são esses membros do comitê gestor do projeto, que aprovam. E internamente a gente faz as regras do tipo, para aprovar tem

que passar em todos os testes, tem alguns clientes-chave que a gente tem, por exemplo, para a gente pedir para testar. Eu mesmo testo os meus códigos da empresa, para ver se funcionam tudo, esse tipo de coisa (E1).

Outro conteúdo disponibilizado na página da Apache, voltado para direcionar o comportamento dos integrantes dos projetos, se refere ao *Contributor License Agreement* (CLA). Todos os desenvolvedores que se tornam *committers* precisam imprimir, assinar e digitalizar esse contrato, fazendo o envio dele em formato de PDF para a secretaria da Apache. Essa é a forma que os *committers* oficializam a sua concordância com os termos de licença da fundação, o que envolve a questão da propriedade intelectual das contribuições dadas ao *software* do projeto. Quando o *committer* é um funcionário de determinada empresa que foi designado para trabalhar em algum dos projeto da Apache, existe um documento específico que deve ser assinado adicionalmente (CLA Corporativo). Vale salientar que a licença que o projeto OpenNLP e o CoGrOO utilizam atualmente é a mesma: a Apache 2.0, caracterizada pela possibilidade de ser usada em qualquer projeto, mesmo que não seja pertencente à fundação. Essa licença está ancorada nos pressupostos comunitários seguidos pela Apache em relação ao desenvolvimento colaborativo e à distribuição aberta do *software* para propósitos diversos.

Em relação aos resultados que precisam ser alcançados em cada projeto, a Apache permite certa liberdade para que cada empreendimento especifique suas metas. Nesse caso, no OpenNLP, nota-se alguma flexibilidade, ainda que o projeto apresente alguns direcionamentos traçados na sua página *wiki*, que são, na realidade, mais planos para aperfeiçoamento do que objetivos e metas propriamente ditos, conforme o relato de E1.

O projeto, em particular, é bem simples, porque é uma biblioteca. O que nós temos é nossas metas para próximas *releases*, que seria um *roadmap*, né? A gente tem uns planos de aperfeiçoamento do projeto. A nossa *wiki* tem uns *deadlines*. Tem *proposals*, *feature proposals* e *improvement proposals*. A gente tem que manter melhor isso aqui, mas a ideia é essa [...] Isso também é discutido na *mailling list* ou no *chat* (E1).

Embora o OpenNLP, assim como os demais projetos da Apache, conte com uma série de procedimentos formalizados, os membros atuantes permanecem sendo voluntários e possuem autonomia no desempenho das atividades que realizam. Mas, para que as pessoas direcionem as ações para o desenvolvimento do OpenNLP, é preciso a existência de um forte controle social.

Em determinadas situações, o condicionante para que os indivíduos assumam responsabilidades no OpenNLP aparece em razão de alguma necessidade individual dos participantes ou do próprio projeto. Como o OpenNLP é integrado por indivíduos, em sua

maioria, de diferentes empresas do mundo, algumas necessidades profissionais da realidade desses indivíduos terminam por impulsionar o desenvolvimento do projeto. Não é raro que no âmbito de suas respectivas organizações esses indivíduos operem no desenvolvimento do *software*. Esta nuance confirma, portanto, uma interface entre o ambiente informal do projeto e o contexto formal no qual os participantes atuam, e o fazem com o uso de recursos fornecidos pela empresa em que trabalham. Esse paralelo entre o informal e o formal ficou explícito na entrevista de E1:

Nós não temos patrocínio, então a gente acaba realmente fazendo no tempo que a gente tem. Se a gente for impor isso a gente perde as pessoas. Tem que ser bem leve e bem autônomo para conseguir fazer. Então o que é que acontece? Por exemplo, tem um *committer*, que trabalha em uma grande multinacional. Esta empresa autorizou ele a fazer um módulo novo, que ele vai usar na empresa dele [...] Então ele vai fazer no tempo da [nome da empresa], a [nome da empresa] vai doar o código para o projeto. Éu quando fiz aqui um projeto que era um detector de língua, eu fiz para a empresa em que eu trabalho. Então eu peguei a autorização com meus colegas da empresa [eu vou fazer isso aqui, mas eu queria fazer *open source*, porque depois a gente tem que manter esse código, esse código aqui que é para todo mundo usar. Beleza? Beleza]. Fui lá e fiz. Então a gente acaba usando, desenvolvendo muito ali dentro coisas que nós precisamos no nosso dia a dia em nossos empregos também (E1).

A filosofia de meritocracia que a Apache busca propagar é uma das formas de ensinar o controle social e também de propiciar os meios para que os membros se mantenham ativos e demonstrem compromisso com o projeto. Com efeito, a assunção de papéis e responsabilidades, no âmbito do OpenNLP, é algo que impulsiona as ações desses membros. Ao trazerem contribuições relevantes para a comunidade, os indivíduos podem alcançar não só o reconhecimento entre os pares do projeto, mas também da própria Apache e da comunidade externa, uma vez que a fundação tem credibilidade e visibilidade mundial. Trechos da entrevista com E1 atestam essas afirmações.

Eu faço mentoria de novos contribuidores, por exemplo. No ano passado, dado todo esse envolvimento que eu tive, eu fui convidado a me tornar membro da Apache. Lá é por meritocracia, então são bem poucos membros no grupo. Em nível de milhares de contribuidores que a Apache tem, são centenas de membros, o que são poucos. E hoje eu faço o papel de planejamento anual da própria Apache, contribuo com os votos de novos membros, de projetos que entram na Apache e que saem, então eu acabo tendo uma voz mais nível organizacional dentro da própria Apache [...] Quais são os benefícios em ser Apache? Os benefícios são essa confiança. Quando um projeto é Apache, você sabe que o projeto segue uma série de restrições, de condições, então você pode confiar mais. As empresas confiam mais nesse tipo de projeto. É melhor do que eu ir no *GitHub* e pegar algum projeto. Para um projeto ser Apache, você sabe que ele segue uma série de condições. Então é mais fácil de conseguir pessoas (E1).

As trocas de experiências entre os integrantes também são mecanismos que fortalecem o senso de identificação com o projeto. No OpenNLP, a maioria dos encontros entre os participantes acontece de forma virtual. No entanto, ocorrem encontros presenciais, inclusive promovidos pela Apache, que são abertos para todos os projetos da fundação. O ApacheCon é uma conferência patrocinada por empresas, que acontece desde 1998, na qual os projetos da fundação têm a oportunidade de estreitar laços e trocar conhecimentos. Além desses encontros presenciais ou virtuais frequentados pelos membros do OpenNLP, há também os eventos técnicos e acadêmicos, que oferecem não só visibilidade para o projeto, mas também o estabelecimento de contatos que ultrapassam as fronteiras do projeto. E1 discorre sobre esses encontros e interações entre a comunidade interna e externa ao projeto OpenNLP:

A Apache em si, tem um evento anual que acontece na América do Norte, esse ano de 2018 vai ser em Vancouver. A própria Apache patrocina a ida de algumas pessoas quando estas comprovam que não tem condições de ir com recursos próprios [...] No OpenNLP em si, a gente faz o seguinte: a gente manda muito trabalho ou *paper* para algumas conferências, para os eventos de *software* em si, ou evento técnico. Aí quando aprova, o pessoal se encontra. Mas, em geral, é o pessoal que tá na Europa que vai. Marcamos encontros virtuais [...] por exemplo, quando a gente vai fazer um artigo ou fazer a submissão de uma palestra e tem que fazer uma coisa rápida, a gente faz um *hangout*. É bem complexo porque você imagina pessoas do mundo inteiro, cada um com um sotaque falando. Eu particularmente prefiro escrever, porque leio e é mais fácil de entender (E1).

Feitas estas considerações, percebe-se que todo o contexto no qual os integrantes do OpenNLP se encontram inseridos influencia para a disseminação do autocontrole, ao considerar que os membros do coletivo priorizem as suas ações para o desenvolvimento do projeto. Dada esta autonomia, membros como E1 dedicam, espontaneamente, parte do seu tempo e esforços para trabalhar em alguma atividade do projeto.

Eu uso o meu tempo livre. Ultimamente, assim, eu acabo ficando bem ocupado com o trabalho no dia a dia, e eu tô contribuindo menos do que eu gostaria. Mas no ano passado, por exemplo, eu contribuí bastante. Então acaba sendo muito de épocas. Tem um plano de desenvolvimento de um projeto novo para empresa, que vai me deixar trabalhar de novo no OpenNLP. Esse novo projeto que eu vou trabalhar vai me permitir que eu trabalhe um pouco mais na comunidade (E1).

O controle entre pares, por sua vez, por ser crucial para garantir a qualidade do código das comunidades de *software* livre, demonstrou ser algo comum dentro do projeto OpenNLP. A própria forma como a comunidade se encontra organizada, seja pela plataforma *GitHub*, seja Apache *JIRA* (rastreador de problemas do projeto), faz com que de maneira informal as contribuições dadas pelos desenvolvedores tenham de passar pela revisão de

algum *committer* para serem aceitas. Ou seja, nota-se que não existe uma formalidade previamente fixada para ditar como esta revisão entre pares deve acontecer. A Apache fornece liberdade aos projetos que dela fazem parte, caso haja interesse, para que possam fazer uso da política R-T-C (*review-then-commit*), na qual todas as alterações realizadas em determinado código se sujeitem à aprovação por consenso para serem confirmadas.

Retomando o caso específico do encontro entre o CoGrOO e o OpenNLP, o entrevistado E1, em sua percepção, viu nessa interseção uma possibilidade de que mais pessoas pudessem fazer a supervisão do código, de forma que haveria uma ampliação em número de colaboradores e aportes de conhecimento.

É um projeto que acaba sendo uma coisa bem técnica, bem específica, não são muitas pessoas no mundo que trabalham com isso, então é legal ter contato com essas pessoas. E, outra coisa, ele que me ajudou nessa questão de tornar uma parte do CoGrOO que eu tinha, mais bruta, então quando eu movi para o OpenNLP agora tem 23 pessoas praticamente cuidando daquele código, que antes era só eu (E1).

Os contatos profissionais que o OpenNLP propiciou para E1, haja vista que a iniciativa converge com a área de atuação profissional dele, se converteram em fator de impulso à movimentação do código do CoGrOO, constituindo, sem dúvidas, uma oportunidade para a continuidade do corretor gramatical. Esta circunstância afastou entraves e deu margem para que o código do CoGrOO continuasse obtendo manutenção necessária, não só por contar com a supervisão de membros do OpenNLP, mas também com toda a infraestrutura e visibilidade que os projetos incorporados pela Apache adquirem.

### 3.4.2 Projeto Noosfero

O projeto Noosfero é uma plataforma *web* para redes sociais e de economia solidária. Além das funcionalidades comuns de redes sociais, tais como grupos de amigos e comunidades, o projeto oferece um sistema de gestão de conteúdos (*content management system* ó CMS) que torna o perfil de qualquer usuário individual ou comunidade um *site* completo, o que inclui: diferentes conteúdos ó texto, galeria de imagens, *blogs*, *uploads* de arquivos etc; *feed* de notícias (em inglês, *really simple syndication* ó RSS); discussões temáticas; agenda de eventos e outras funcionalidades. Vale ainda destacar que o Noosfero é um *software* livre expansível (*plugins*) e customizável, que teve origem em 2007, na Cooperativa de Tecnologias Livres (Colivre).



O perfil dos representantes do Noosfero entrevistados se encontra discriminado no quadro a seguir.

**Quadro 8** ó Identificação e perfil dos entrevistados do projeto Noosfero

<b>Código</b>	<b>Função</b>	<b>Observação</b>
<b>E4</b>	Desenvolvedor	Fundador da Colivre, com atuação na criação e desenvolvimento do Noosfero
<b>E5</b>	Desenvolvedor	Fundador da Colivre, com atuação na criação e desenvolvimento do Noosfero
<b>E6</b>	Desenvolvedor e <i>release manager</i>	Membro da Colivre e desenvolvedor do Noosfero, no qual exerce atualmente o papel de <i>release manager</i>
<b>E7</b>	Desenvolvedor	Desenvolvedor do Noosfero e um dos viabilizadores da parceria entre o projeto e o Serpro
<b>E8</b>	Desenvolvedor	Fundador da Colivre, com atuação na criação e desenvolvimento do Noosfero
<b>E9</b>	Docente e colaborador	Atuou em diversas frentes voltadas para a colaboração com o projeto Noosfero, principalmente na articulação entre a academia e o projeto

**Fonte:** Elaborado pela autora (2019)

Da parte da pesquisadora houve uma preocupação de oferecer diferentes visões sobre o Noosfero, por meio de entrevistas, tanto com pessoas que estão no empreendimento desde o início quanto com colaboradores que operaram no sentido de construir interfaces do projeto com outras instituições. Como o Noosfero é um *software* livre, que teve origem e até hoje tem a sua manutenção e patrocínio associados à Colivre, a maioria dos entrevistados foi composta por indivíduos que tiveram, ou que ainda têm contato próximo com essa cooperativa. No que diz respeito ao contato do Noosfero com outras instituições, dois dos entrevistados, E7 e E9, são pessoas que atuaram efetivamente no sentido de promover o projeto em instituições públicas. No caso específico de E9, esse indivíduo foi representante da inserção da iniciativa no ambiente acadêmico. Essas relações do Noosfero com a Colivre e com organizações públicas foram cruciais para o funcionamento do projeto no decorrer dos anos.

#### 3.4.2.1 Trajetória de desenvolvimento projeto Noosfero

A Colivre é uma empresa de serviços de tecnologia da informação, criada em 2006, que se especializou no desenvolvimento de soluções livres para a *web*. O grupo de fundadores da Colivre era formado por militantes do movimento de *software* livre, que se reuniram com a

proposta de elaborar e difundir essa ideia por meio de soluções oferecidas para o setor privado e público, tal como declarou E4:

Como a base da cooperativa da Colivre é esse viés de *software* livre, que tá muito dentro da empresa, então a militância das pessoas que criaram a cooperativa já era em cima de *software* livre. A cooperativa foi fundada com pessoas também que vinham do movimento da economia solidária e cooperativismo e do *software* livre. Então o *software* livre era algo que estava em nosso radar. A maioria das pessoas que estava na cooperativa, a energia de criar a empresa foi também de criar um espaço para viver com *software* livre, porque é algo que todo mundo já fazia de forma voluntária. A gente já trabalhava com *software* livre de maneira voluntária e a gente queria começar a viver de forma econômica e financeira disso também (E4).

A partir da celebração de dois contratos comerciais, um com o Fórum Brasileiro de Economia Solidária (Febes), direcionado para elaboração de canal de informações, e outro com uma organização Suíça, para criação de sistema de conteúdo digital para projetos de inclusão na África, foi que em 2007 surgiu uma oportunidade certa para o lançamento do Noosfero. E4 e E5, em entrevista, discorreram sobre esse acontecimento:

Foi assim que o Noosfero surgiu, através de um contrato comercial para resolver um problema específico de um cliente, que era o Febes, Fórum Brasileiro de Economia Solidária, e a internet.org, uma organização da Suíça, que queria criar uma rede social. Então a gente juntou as demandas que eram bem parecidas, eles queriam algo parecido, a gente resolveu juntar os dois clientes numa solução só. E aí foi assim que começou o desenvolvimento do Noosfero (E4).

O Noosfero foi uma cooperativa [...] 16 pessoas prestando um serviço para 2 clientes, que cresceu e ainda presta serviços a partir de organismos internacionais, e nacionais, órgãos públicos e entidades. O Noosfero, o modelo de negócio dele, não tá pré-estabelecido por rumo. Ele atende pessoas e forma comunidade (E5).

Apesar de ter surgido no escopo de uma demanda comercial, por se tratar de uma iniciativa de *software* livre, o Noosfero chamou a atenção de outras pessoas da comunidade externa, as quais se identificaram com o projeto e quiseram contribuir. E4 relatou como se deu a ampliação de grupos interessados no projeto.

O cliente chegou e a gente resolveu o problema dele. Depois chegou outro cliente, depois chegou outro cliente [...] nesse caminho, atores externos tomaram conhecimento do projeto, simpatizaram com a comunidade e começaram a entrar na comunidade. Aí a comunidade começou a surgir. Então, não tem um modelo de negócio. Não foi uma coisa pensada, planejada [...] (E4).

Como é possível perceber, as raízes do Noosfero estiveram bastante concentradas na Colivre; algo, todavia, que foi mudando, conforme o projeto ganhava notoriedade. De acordo com E6, *release manager* do projeto:

A comunidade no Noosfero era basicamente a Colivre que desenvolvia e o desenvolvimento do Noosfero foi todo guiado com base nesses serviços [...] A Colivre conseguiu prospectar ou vender. Então aí, ao longo desse período onde não existia a comunidade Noosfero, a Colivre era a única mantenedora do projeto (E6).

As circunstâncias começaram a mudar quando outros indivíduos e grupos passaram a contribuir com o Noosfero de forma paralela à comunidade, ainda que cada qual no âmbito das organizações às quais pertenciam, como destacaram E5 e E6:

São pessoas às vezes remuneradas também, mas remunerada por outra entidade, empresa que organiza, outro grupo, mas que investiu em uma mesma solução que nasceu dentro de uma cooperativa (E5).

Ao longo do tempo, começaram a surgir outros agentes na comunidade que vão pensar também em desenvolver o Noosfero como uma plataforma que precisava atender esses propósitos individuais (E6).

Há, então, um espaço temporal, em que o Noosfero se destaca, na medida em que empresas que tinham serviços atrelados ao projeto começaram a ver no *software*, como afirmou E6, ãuma solução interessante para alcançar os seus objetivos... usando o Noosfero para suas linhas individuaisö.

Nos parágrafos seguintes se faz a retrospectiva de algumas iniciativas que influenciaram positivamente no avanço experimentado pelo Noosfero e para as quais dois personagens desempenharam ações de suma importância, os entrevistados E9 e E7. O primeiro profissional, atualmente professor universitário, com passagens de trabalho na Associação de *Software* Livre Brasil (ASL), na qual administrou o seu portal, pelo Centro de Competência em *Software* (CCSL/USP), durante a fase do seu doutoramento, e pela Universidade de Brasília, na função de docente. O segundo profissional, egresso da Colivre, é atualmente funcionário do Serviço Federal de Processamento de Dados (Serpro), empresa pública nacional com atuação na área de tecnologia da informação. Assim, na linha do tempo do Noosfero, sob o protagonismo dos entrevistados em tela, se encontram os eventos notáveis do projeto, apresentados a seguir.

Em 2008, quando E9 administrava o portal da Associação de *Software* Livre Brasil (ASL), junto com outros representantes, teve a iniciativa de fazer com que esse portal (*software* livre.org) se tornasse a rede social das comunidades de *software* livre no Brasil. Como o portal utilizado na época estava passando dificuldades técnicas, os integrantes da ASL, dentre os quais estava E9, aproveitaram os recursos de um contrato de manutenção que existia com a Colivre e decidiram que o portal *software*livre.org passaria a adotar o Noosfero.

Por volta de 2009, E9, na época doutorando na USP, vinculado ao Centro de Competência em *Software* Livre (CCSL), promoveu uma articulação entre o Noosfero e um

projeto da Comissão Europeia, denominado *Quality Platform for Open Source Software* (QualiPSo). Como o QualiPSo se relacionava com a pesquisa de doutorado de E9, ao saber que eles estavam querendo desenvolver a rede social desse projeto, o referido entrevistado apresentou o Noosfero como solução viável. O resultado dessa intermediação fez com que a Colivre fosse contratada e recebesse recurso financeiro para desenvolver a rede social QualiPSo, usando o Noosfero. Nesse período, o Noosfero também passou a ser utilizado pelos pesquisadores do CCSL, quando E9 utilizou o *software* como base para o desenvolvimento de uma ferramenta de análise de código-fonte, chamada Mezero, que atualmente não está mais em funcionamento. O desenvolvimento dessa ferramenta, com ênfase nos esforços de E9, no entanto, foi significativamente importante no sentido de possibilitar que o Noosfero fosse adaptado para ter uma infraestrutura de *plugins* em funcionamento.

Em 2011, quando E9 retornou do doutorado sanduíche, o CCSL foi procurado por um docente para fazer um projeto de rede social da USP, sendo que a universidade já tinha um projeto de rede social conhecido como Stoa, que se encontrava estagnado. Diante desse cenário, esse docente, em conjunto com outros pesquisadores do CCSL, decidiu elaborar um projeto para reformular o Stoa, sendo a Colivre a empresa contratada, em virtude da sua competência técnica na área para atuar nessa reformulação. O recurso financeiro advindo da USP, então, foi investido na Colivre para a execução do projeto no qual o Noosfero foi utilizado, sendo E9 o coordenador desse projeto. Na entrevista com E9, o docente falou da sua experiência com esse projeto do Stoa:

Nesse projeto a gente atacou os principais problemas técnicos do Noosfero. Eventualmente eu fazia o pareamento com os desenvolvedores da Colivre, em coisas mais simples, mas aí deu um salto de qualidade, porque aí, a rede da USP nasce com 60 mil usuários. O Stoa hoje é relativamente usado, está meio abandonado, mas deu esse *boom* (E9).

A parceria do Noosfero com o Serpro foi viabilizada especialmente pelo integrante E7, um ex-funcionário da Colivre, que se tornou servidor público dessa instituição. Assim, mesmo após perder o vínculo formal com a Colivre, E7 continuou atuando como mantenedor voluntário do projeto. Já no Serpro, E7 teve a oportunidade de se envolver na demanda para construção de uma rede social corporativa interna, direcionada para melhorar a comunicação dos funcionários. Nessa demanda, E7 percebeu a possibilidade de integrar o Noosfero, algo que foi aprovado pela instância superior do órgão, sendo dado o início ao projeto, tal como relatado pelo entrevistado:

Tinham soluções que precisavam para resolver esse problema e nós conseguimos botar o Noosfero para resolver. Aqui no Serpro, por exemplo, tinha uma demanda de melhorar a fluidez da comunicação dos funcionários e

criar uma rede social corporativa interna. O pessoal comprou a ideia e hoje a gente usa rede social corporativa interna e vende para clientes do Serpro também (E7).

Esse projeto do Serpro rendeu frutos importantes, pois foram cerca de dois anos trabalhando no desenvolvimento em que o Noosfero era o código principal. Além disso, quando o Serpro conseguia vender essa solução para determinado cliente, o Noosfero era a base tecnológica utilizada, o que trouxe uma grande visibilidade para esse *software* livre, principalmente no âmbito do setor público.

Um projeto de destaque desenvolvido pelo Serpro para a Presidência da República, que fez uso do *software* livre Noosfero, foi o Participa.br. Esse projeto de grande porte, que aconteceu em torno de 2013 e 2016, tinha como proposta criar ferramentas de comunicação para que os diversos atores da sociedade pudessem ter um canal de diálogo com o Governo Federal na construção de políticas públicas. Por meio de articulação entre a presidência do Serpro e um funcionário da Secretaria-Geral da Presidência da República, houve o aporte de recursos nesse projeto, sendo um dos principais *cases* de sucesso do uso do Noosfero. Vale salientar que no desenvolvimento desse projeto, além do Serpro, houve a colaboração de diversos consultores e agentes externos, como é o caso de integrantes do Laboratório Avançado de Produção, Pesquisa e Inovação em *Software* Livre (Lappis) da Universidade de Brasília, época em que E9 era docente desta instituição. A grandiosidade do projeto e o destaque que ele teve na mídia na época trouxeram resultados positivos para a visibilidade da comunidade do Noosfero. Os entrevistados falaram um pouco sobre esse momento de grande relevância:

De repente surgiu um outro ator contribuindo também [...] Aí em 2014, por exemplo, foi articulado o Participa.br, que também moveu toda uma equipe, eu já não tava na Colivre, aí eu fui consultor também desse projeto, e várias outras pessoas também foram consultoras, que também contribuíram com contribuições externas [...] O [nome do entrevistado E9] encabeçou inúmeros projetos onde ele conseguiu subir instâncias de Noosfero, colocou pessoas para ajudar [...] Cada iniciativa dessa fazia o Noosfero, e estas coisas que eles desenvolviam novas, eram trazidas de volta para toda a comunidade (E4).

Naquele período foi de 2014, 2015 e 2016 a gente tava com um projeto, o Participa.br, que hospedaram aqui no Serpro. A Presidência da República pagava para o Serpro para manter. A gente tinha um time aqui com 8 pessoas que tocavam o projeto. Aí nesse período a gente fez bastante contribuição lá no Noosfero (E7).

Em paralelo a esse contexto no qual existia toda essa movimentação em volta do Participa.br, por volta de 2013, um integrante da Secretaria de Logística e Tecnologia da Informação (SLTI), vinculada ao Ministério do Planejamento, Desenvolvimento e Gestão

(MP), entrou em contato com E9, então docente da Universidade de Brasília (UnB), para um novo projeto. Esse projeto tinha como escopo a reformulação do portal do *software* público brasileiro. Assim, por meio de uma descentralização de recursos do MP para o Centro de Informática (CDT) da UnB, foi realizada uma cooperação com esse propósito. E9 atuou como coordenador do projeto, que teve início em 2014, gerenciando uma equipe formada por alunos da universidade contratados como bolsistas, integrantes da Colivre, mas que atuaram como *experts* na área e não representando a cooperativa, dentre outros profissionais contratados para atender os objetivos formalmente delimitados nesse projeto. A duração aproximada do projeto foi de 30 meses, período em que vários problemas de atualização e de segurança do Noosfero foram resolvidos. Outra iniciativa liderada por E9, também por volta de 2013, ocorreu com a criação do *site* da UnB do campus do Gama. E9 relatou um pouco dessa experiência:

Aconteceu uma coisa em paralelo ainda em 2013. Eu cheguei na UnB do Gama e não tinha *site*. Aí eu consegui 4 bolsas de estágio. O que a gente fez? Noosfero. Aí, se você entrar em fga.unb.br é Noosfero. E qual é o lance? Cada usuário, cada conta de usuário, já tem o perfil, já tem uma página, então, assim, para cada conta do professor, automaticamente, já grava a sua página. A gente botou uma cara bonitinha, criava uma conta para o professor, com foto, ela já abria a sua página. Em um mês, todos os professores da FGA tinham a sua conta (E9).

A partir do exposto, percebe-se que essas inúmeras iniciativas em outras organizações que atuaram no desenvolvimento do Noosfero, representadas especialmente pelo setor público (Serpro, universidades e Governo Federal), trouxeram uma dinâmica diferenciada para o aperfeiçoamento do *software* livre. Nesse sentido, ao mesmo tempo em que essas organizações paralelas desenvolviam o Noosfero para ser usado de acordo com os seus interesses, havia um retorno desse desenvolvimento para a comunidade do *software*. Esse retorno para a comunidade aconteceu, sobretudo, pela atuação de voluntários que ao utilizarem o Noosfero em projetos específicos da sua realidade profissional, tiveram a preocupação de levar esse desenvolvimento de volta para o projeto comunitário. Segundo E9, a ideia central propagada é que ao invés de o Noosfero ser segmentado em projetos diferentes, numa ação conhecida como *fork*, que exista na comunidade uma integração das otimizações realizadas no *software*. Conseguir ter essa dinâmica de forma estabelecida entre os múltiplos atores desenvolvedores do Noosfero, no entanto, nem sempre aconteceu de forma tranquila. Como será visto em detalhes nas seções posteriores, houve vários debates entre os integrantes do projeto, para que a abertura da comunidade conseguisse manter a qualidade do código-fonte do *software*.

Ainda que as iniciativas paralelas tenham possibilitado o desenvolvimento do Noosfero, o *software*, desde o início, sempre esteve ligado ao patrocínio da Colivre, em termos da sustentação do projeto com recursos próprios, algo que acontece até os dias atuais, conforme destacado pelos entrevistados:

Eu também vejo que a comunidade Noosfero precisa em algum momento ter essa entidade formal, para que ela consiga ter sustentabilidade independente da Colivre. É um ônus para Colivre hoje, manter uma série de coisas que são da comunidade, pagar servidor, manter isso, fazer comunicação [...] Esse ônus todo é da Colivre. Na comunidade Noosfero, antes a Colivre era o *core* sozinha, mas hoje a Colivre é um ator da comunidade Noosfero que ainda é central. Porque como ela ganha dinheiro com o Noosfero, dos contratos que ela que tem, parte dessa grana financia essa infraestrutura que a Colivre acaba bancando. Como a Colivre depende do Noosfero para sobreviver, ela tem um interesse maior nisso (E4).

Hoje o Noosfero continua sendo sustentado majoritariamente pela Colivre, como sempre foi ao longo da história. Porque a gente nunca conseguiu desenvolver [...] tanto a Colivre quanto os demais agentes da comunidade, nunca conseguiu desenvolver algum modelo de negócios que a gente pudesse realmente ter alguma fonte fixa de recursos para desenvolver o Noosfero como plataforma mesmo. O domínio do Noosfero quem mantém é a Colivre, o servidor no qual tá a intranet é a Colivre que mantém, que paga o servidor, equipe e gestão do servidor [...] A Colivre doou para comunidade, um servidor ao qual todos os desenvolvedores têm acesso. Apesar de depender do financeiro da Colivre, mas do ponto de vista de quem tem acesso e de quem pode mexer lá, os desenvolvedores da comunidade têm (E6).

Atualmente, após mais de 10 anos de funcionamento, o que se nota é que o Noosfero é um *software* livre nacional que tem conseguido se manter operando ao longo do tempo, conforme revelou E4:

A gente deu certo em termos de sustentabilidade, por exemplo, é um projeto que já tem 10 anos, praticamente, e que tá ativo. Tem uma comunidade ativa, então, dá para dizer que deu certo. Parte do dar certo, que eu considero, é a rede social de pessoas militantes e de pessoas investindo, e conseguindo recursos, conseguindo contratos, conseguindo articular iniciativas, de manter o desenvolvimento ativo. Porque a parte econômica é o que bota a lenha na fogueira, vamos dizer assim (E4).

Em termos de conseguir um modelo de governança que permita ao Noosfero ser um empreendimento comunitário independente e autossustentável, de forma autônoma em relação à Colivre, isso é algo que sempre vem sendo discutido pelos membros da cooperativa, como relatado por E5:

A cooperativa é uma transformação de um modelo de negócio, porque o modelo de negócio é uma coisa muito inovadora com *software* livre, principalmente no Brasil. Então a gente não achou a fórmula [...] a gente testou vários modelos e até hoje testa (E5).

Mesmo enfrentando períodos de maior atividade e outros mais parados em termos de desenvolvimento, o Noosfero continua sendo um caso de projeto de *software* livre que vem conseguindo se manter ativo ao longo do tempo. Atualmente o *software* está na sua versão 2.8.0 e faz uso da licença GNU General Public License, versão 3.

#### 3.4.2.2 Estrutura do projeto Noosfero

Parte significativa da infraestrutura do Noosfero permanece associada à Colivre e é beneficiada pelos contratos comerciais celebrados pela cooperativa. Além de ter esse patrocínio da Colivre para se manter funcionando, na interação dos integrantes da comunidade o Noosfero faz uso de diversas TICs voltadas para a comunicação dos indivíduos. Dentre as tecnologias citadas pelos entrevistados, destaca-se o uso de plataforma de hospedagem de código-fonte, como o *GitHub* e o *GitLab*, chamadas de voz, principalmente por meio do Google *Hangouts*, *chat* e listas de *e-mail*.

A hierarquia formal do projeto acontece por meio da separação de papéis, representados pelos *committers* e pelo *release manager*. Esses papéis passaram a ser estabelecidos simultaneamente ao crescimento do número de integrantes da comunidade. Além desses papéis formais, é perceptível no Noosfero uma hierarquia mais informal e que acontece fundamentalmente em função da experiência e das contribuições dos integrantes, principalmente da parte daqueles que estão no projeto desde o seu início. Sobre essas questões, destaca-se um trecho da entrevista de E4:

Tem uma hierarquia mesmo, oficial, tem um papel que é chamado *release manager*. E assim, num caso de [...] vamos dizer, de empate na comunidade, se a decisão vai ser x ou y, ele tem a palavra final. Além disso, existe uma certa hierarquia também mais distribuída, que tem aquelas pessoas que são desenvolvedores *core* do Noosfero, que são os *committers*, a galera do início, não é essa galera que tá entrando (E4).

A hierarquia existente entre o *core* time de *committers*, especialmente a percebida em torno do *release manager*, era mais rígida e com o tempo foi se tornando algo mais flexível, com a entrada de novos integrantes na comunidade do projeto Noosfero.

Existe uma hierarquia porque os membros do *core team*, de certa forma, têm um poder maior de incorporar código que outras pessoas que não são parte do *core* não tem. Mas hoje eu noto que essa hierarquia é muito mais fluida. No passado, ela era mais rígida. O *release manager*, de certa forma, era mais rígido. Hoje é uma coisa mais fluida com a entrada de novos membros para o time. A pessoa querendo contribuir, tendo contribuições aceitas, ela é aceita de forma mais natural do que antigamente. Na minha visão, esse processo está mais simples de ser feito (E7).



O entrevistado E6, por sua vez, notou a existência de hierarquia não no sentido de indivíduo, mas numa perspectiva macro, que envolve as organizações que trabalharam com o Noosfero. Nesse sentido, a Colivre, por ser a empresa que criou o Noosfero e que atua até hoje como principal mantenedora, termina tendo maior autoridade para trilhar os rumos do projeto:

Talvez exista certa hierarquia no sentido dessa questão do quão próximo está do projeto, então, acho que, querendo ou não, por a Colivre ser tão implicada com esse negócio de empresa do Noosfero, é complicado para outros agentes que usam o Noosfero de forma pontual rebater com a Colivre numa coisa. Pelo nível de desenvolvimento que a Colivre tem com o projeto e investimento também, eu acho que acaba meio que levando todo mundo para o caminho que ela segue (E6).

Em relação aos papéis associados a essa hierarquia, existem alguns critérios previamente estabelecidos pela comunidade para a escolha dos integrantes que irão exercê-los. No caso dos *committers*, a percepção de que o indivíduo merece assumir o papel é algo considerado com atenção. Os entrevistados relataram como os critérios para se tornar *committer* do Noosfero têm acontecido no projeto.

Na verdade, a pessoa solicita, e se mais 2 ou 3 pessoas que já são *core committers* aprovam, ele pode entrar. Isso significa que ele tem permissão de escrita no repositório diretamente. A solicitação acontece por meio do *merge request*, que é um código que tá sendo proposto de ser incluído no repositório, que outras pessoas precisam aprovar. Se 2 outros, que já são *committers* diz [eu aprovo que essa pessoa pode virar *committer*], a gente aceita aquele *merge request* e alguém vai lá e bota ele com permissão de escrita no repositório. Esse protocolo já tá documentado. Então, o protocolo para que uma pessoa entre é que ele crie uma *merge request* no arquivo com a lista de quem é *committer*, com o pedido incluído (E4).

A regra básica é aquela que eu falei, a pessoa só precisa ter alguma experiência no Noosfero. Inclusive, no pedido da pessoa, a gente determina que ela tem que listar quais as contribuições que ela já tem no Noosfero, seja na forma de discussão, seja no desenvolvimento de código, seja na forma de complementação ou resolução de um problema. Uma vez que ela consiga mostrar que ela tem alguma contribuição, basta que dois desenvolvedores da comunidade sejam a favor para ela se tornar desenvolvedor também (E6).

A pessoa precisa fazer um pedido para ser *committer* e tem que ser aprovado por dois ou mais dos *committers* atuais. Para você querer alguém fazendo parte desse grupo, você tem que ter certa segurança de que a pessoa já tem um histórico de contribuições. E a gente instituiu um processo de aceitar novas pessoas e de lá para cá tem várias outras pessoas que são *committer* do projeto que não são da Colivre (E8).

Quanto ao papel de *release manager* (RM), ainda que para tanto o indivíduo não precise necessariamente ser um integrante da Colivre, na prática, até hoje, todos os que executaram esse papel eram funcionários da cooperativa. Isso se explica, principalmente, pelo suporte financeiro dado pela Colivre desde o lançamento do Noosfero, o que inclui também o

papel do RM, já que o indivíduo precisa dedicar tempo e esforço do seu trabalho no desempenho desta função. Embora esteja documentado que o *release manager* seja o responsável por indicar o próximo, a rotatividade desse encargo no projeto não demonstrou acontecer de forma estruturada. Assim, ainda que o mérito seja considerado como um critério de escolha, as pessoas da Colivre que assumiram o papel de RM sempre foram aquelas que demonstraram ter interesse e disponibilidade para operar na função, por tempo indefinido. Os entrevistados abordaram as suas visões sobre o papel de RM:

O RM é interno da Colivre. É uma parte que, vamos dizer assim, é um pouco não bem estruturado. Ele passa o bastão para você e diz [...] [você vai ser *release manager* agora] (E4).

Hoje eu sou RM do Noosfero, porque a Colivre me banca para ser RM. Se a Colivre não me bancasse, eu teria que fazer isso voluntariamente, ou então abriria o posto para quem quisesse. O Noosfero, por exemplo, nunca teve um *release manager* fora da Colivre, não por não deixar ter, mas porque nunca ninguém quis (E6).

As entrevistas alcançaram também aspectos sobre as funções definidas para o papel de RM, que basicamente é a pessoa responsável por fazer o gerenciamento do projeto e o processo de lançamento das novas versões do Noosfero.

Quais são as funções do *release manager*? Fechar o produto para lançamento. E aí, para fechar o produto para lançamento, tem que atender os interesses todos, de quem tá contribuindo. Não é mais só a Colivre que pauta os rumos do projeto [...] tem uma comunidade ali. Então quando a funcionalidade tá em conflito, o papel do *release manager* é garantir o atendimento desses interesses (E5).

Tudo que a comunidade precisa que seja feito, ele tem que fazer, que é exatamente isso, quando faz o lançamento de uma versão nova, isso é divulgado no portal do projeto, é divulgado no repositório, ele escreve um documento com as mudanças, com as soluções que foram feitas, tudo isso é papel do *release manager* (E4).

Em relação à liderança, o próprio *release manager*, por ser o gestor do projeto, também deteve a incumbência formal de atuar como líder, no sentido de estabelecer práticas e direcionamentos que garantissem a qualidade do *software*. No entanto, apesar dessa responsabilidade formal, outras pessoas, principalmente as mais antigas e detentoras de conhecimentos avançados sobre o Noosfero, também assumiram a tarefa de definir os rumos do projeto. Isso demonstra que a liderança não fica limitada apenas ao exercício do papel de RM, sendo uma responsabilidade compartilhada em função da experiência e do mérito dos integrantes. Alguns trechos das entrevistas exemplificam como a liderança no Noosfero tem acontecido no projeto.

Sobre essa questão de ser líder, existe o papel chamado *release manager* [...] Por exemplo, eu já fui *release manager* [...] outras pessoas já foram. O *release manager*, no final das contas, é a pessoa mais preocupada com a qualidade do produto. Ele é um papel bem central, tanto na Colivre, quanto na comunidade Noosfero. É o gestor do projeto Noosfero (E4).

Eu acredito que hoje o RM tem esse papel de liderança. Ele frequentemente, tem puxado uma reunião onde todos os interessados no projeto compartilham os seus contextos. De certa forma, isso vai criando uma ideia de produto e também acaba havendo ponto de sinergia. Hoje o pessoal da Colivre que são os membros mais ativos. Então é uma liderança meritocrática, não é uma coisa que é imposta, porque eles são quem mais estão desenvolvendo o projeto hoje (E7).

No Noosfero, na prática, você precisa ter alguém para liderar, por exemplo, o processo de lançamento, definir um calendário e tomar decisões quando não houver consenso. Então, isso é o que a gente chama de *release manager* ou gestor de lançamento [...] (E8).

Em paralelo aos papéis mais atrelados à comunidade, os atores formais que fazem interface com o Noosfero, demonstram ter toda uma organização interna, em termos de infraestrutura e do estabelecimento de funções e reponsabilidades, para que o desenvolvimento do *software* aconteça em suas instâncias. No caso específico da Colivre, fica claro que nos contratos comerciais referentes ao Noosfero, além das responsabilidades existentes em torno do desenvolvimento do *software*, existem os papéis administrativos:

Então assim, para sustentar esse desenvolvimento do *core* da aplicação tem a área meio. Então, tem ali o papel de negociação [...] das pessoas que não são de código, mas que articularam o pagamento da remuneração, do *core* ali para sair essas coisas (E5).

Ainda que responsabilidades sejam atribuídas no contexto comunitário do Noosfero, na prática, os membros não necessariamente se limitam ao exercício de uma única função:

As coisas não eram muito segmentadas. A gente participava de todo o processo de desenvolvimento, infraestrutura, trabalho de documentação, tradução, enfim, a gente fazia tudo ponta a ponta. Não é como projetos que são muito grandes que você tem um time que é só tradução, um time que é só comunicação [...] Lá a gente acaba meio que permeando todas essas áreas. Todo mundo atua em todos os papéis, só que em alguns momentos algumas pessoas são focadas mais em alguns pontos e outros noutros [...] (E7).

Nos processos de tomadas de decisões no contexto da comunidade, especialmente aqueles referentes a escolhas estratégicas e direcionamentos sobre o futuro do Noosfero, ainda que o projeto tenha a preocupação de ouvir as diversas opiniões existentes, quando o consenso não é alcançado, cabe ao RM o poder da decisão final. No entanto, embora essa regra de decisão centralizada no RM de fato exista, o que é mesmo habitual são as decisões tomadas de forma conjunta, sobretudo considerando os pareceres dos membros mais experientes. O

*release manager* atual do Noosfero e o entrevistado E8, abordaram este assunto da centralização decisória.

E essa pessoa tem o diferencial de numa situação de se os desenvolvedores não chegaram a um consenso em relação a alguma coisa que precisa ser decidida no projeto, o *release manager* tem o voto de minerva (E6).

O *release manager* coordena o processo e é a pessoa que a gente confia para tomar as decisões que precisam ser tomadas quando não tem consenso (E8).

Uma das decisões mais complexas ao longo da história do Noosfero ocorreu com o crescimento da comunidade, quando entidades externas à Colivre passaram a contribuir com o código do projeto. Esta expansão tornou o código do Noosfero extremamente intrincado, fazendo com que a comunidade recorresse a novas regras para aceitação das contribuições externas. De um lado, os atores que não eram da Colivre reivindicavam a necessidade de uma abertura comunitária para a recepção de suas contribuições. De outro, a Colivre, que até então era a principal e única mantenedora do projeto, tinha o desafio de pensar uma forma para que essa abertura acontecesse, no entanto, sem prejudicar a qualidade do código. Após intensos debates, a decisão final tomada levou à regulamentação dos papéis e responsabilidades existentes até hoje no Noosfero, bem como ao estabelecimento de critérios precisos e claros para que determinado código fosse integrado ao código-fonte principal do projeto.

Isto posto, mesmo com essa abertura da comunidade do Noosfero, até mesmo pelo fato de a sustentação do *software* ainda estar muito integrada com a Colivre, é inerente que nas questões centrais do projeto a cooperativa, na figura do RM, tenha uma influência maior nas decisões. Mas, com exceção desses casos específicos, em especial nos projetos paralelos formados em torno da comunidade, as decisões relativas ao desenvolvimento costumam ser tomadas pelos próprios interessados, no seu dia a dia, com maior autonomia.

#### 3.4.2.3 Processos de controle do projeto Noosfero

Apresentada a estrutura organizacional e decisória do Noosfero, destaca-se o fato de o Noosfero se tornar um projeto de grande porte, com múltiplos agentes envolvidos e que, por isso, precisou ter alguns dos seus processos de controle bem definidos.

Esse aspecto é sobremaneira notado nos critérios para um indivíduo se tornar *committer*, caso em que o pretendente precisa da aprovação de 2 ou mais integrantes do coletivo, após a formalização do pedido em algum dos canais de comunicação do projeto. É necessário, inclusive, ao fazer a solicitação, que a pessoa comprove o mérito para se tornar *committer*, assinalando suas contribuições em favor do Noosfero. Em relação ao *release*

*manager*, embora a escolha para esse posto geralmente aconteça de forma rotativa entre os membros da Colivre, assim como a função de *committer*, existem atribuições bem delimitadas em torno das responsabilidades atinentes a esta função, como já visto na seção anterior.

Uma das obrigações do RM se refere ao lançamento de versões, ainda que na prática a periodicidade não aconteça em função de um calendário rígido. As opiniões adiante falam sobre o lançamento de versões do Noosfero.

O Noosfero tinha adotado uma estratégia que é de janelas de tempo para lançamentos. Por exemplo, a cada três meses lança uma nova versão. O que tiver pronto, naquela janela de tempo, é lançado. É o *release manager* que faz o lançamento, é ele que pega o material, junta tudo que tá pronto até aquele momento. O que não tá desenvolvido, ele já agenda para o próximo *release*, tira dali, enfim, notifica as pessoas que tem que notificar sobre o lançamento novo, manda mensagem para lista. É o *release manager* que faz tudo isso (E4).

Só lanço versões quando eu sinto que já tá precisando lançar, ou então quando tá uma versão que já teve muitas correções. Assim, existe sempre um indicativo de lançar uma versão de manutenção a cada mês e versão de novas funcionalidades a cada dois meses, mas isso é meio que só uma referência (E6).

Normalmente, a gente definia um calendário para lançar o máximo possível de versões, com a maior frequência possível. O RM é a pessoa que é a dona desse processo [...] [então, a gente vai definir essas datas, vai anunciar quando começa, a partir de agora ninguém mais adiciona novas funcionalidades, vamos completar as traduções e testar e vai garantir que tudo o que tiver que ser corrigido antes do lançamento vai ser corrigido] (E8).

Independentemente dos trabalhos comerciais desempenhados pela Colivre, da ação de atores internos ou externos, da imprescindível necessidade de atualizações de novas versões, ou de quaisquer circunstâncias afins, algo concreto nos processos de controle exercidos pelos integrantes da comunidade é que nada que prejudique a fluidez do código será acrescentado ao repositório do *software*. Outra opção dada, é que aquilo que foi desenvolvido venha a ser colocado na forma de *plugin*. São pontos de vista sobre este assunto:

O que é que acontece, quando começa a ter cruzamento com outros atores, com outros interesses, no curso do Noosfero, aí se dá os canais de administração e gestão e sempre pautando principalmente a qualidade. E aí a gente não pode colocar nada de código aqui que prejudique essa infraestrutura [...] porque a gente tem cliente. Eu falo a gente, porque eu estava lá, a gente não pode aceitar qualquer coisa aqui que possa prejudicar os nossos clientes. Então a gente tem uma gestão mesmo de qualidade, que hoje já se torna mais flexível, porque esse ambiente tá distribuído (E5).

Isso foi uma das razões pela qual hoje o Noosfero tem a questão de plugins muito grande. Porque a gente chegou num ponto que o Noosfero já tinha [...] muitas pessoas com interesses diferentes e as organizações de empresas que estavam atuando com o Noosfero queriam objetivos diferentes, mas aquilo

tinha que funcionar junto de alguma maneira. Quando a gente precisa resolver alguma demanda de cliente, a gente vai ver o que é que esse cliente quer fazer e o que a gente pode incorporar no código do Noosfero, que vai ser útil para todo mundo (E6).

Desde o início a gente tentou quase sempre não ter no código do Noosfero nada do que fosse específico de um cliente [...] A gente desenvolveu uma infraestrutura de plugins, então se precisava de alguma coisa que é código, que não tenha nada de configuração e que precisa ser desenvolvido, mas que não faz sentido no contexto geral, a gente desenvolvia isso (E8).

O Noosfero tem procedimentos descritos referentes ao desenvolvimento do código. Nesse sentido, um membro que passa a utilizar e contribuir com o *software* livre Noosfero, precisa estar ciente de que as suas ações devem seguir os parâmetros estabelecidos pela licença, que atualmente é a GNU GPL, versão 3. A declaração de E4 aborda a questão da concordância com os termos de licença.

Na parte técnica, tem um documento no próprio código fonte, que tem coisas de guias de estilo, como você deve escrever o código [...] O Noosfero usa a licença GPL. Aí, por exemplo, ninguém pode incluir um código que não seja nessa licença, por exemplo. Todo mundo que contribuir com o Noosfero, já de antemão, concordou que o código que o que ele tá fazendo vai tá sob aquela mesma licença. A licença também meio que já dá um norte em relação a isso [...] (E4).

O controle entre pares é uma prática comumente usada pelos desenvolvedores do Noosfero. Na documentação de desenvolvimento do projeto essa forma de controle se encontra formalizada, sendo estabelecido que pelo menos um *committer* deve fazer a revisão do código antes de ele ser inserido no repositório. Além disso, nessa documentação, fica explícito que, caso o *committer* não tenha o seu código revisado em uma semana, ele mesmo poderá fazer a integração desse código ao repositório. Os integrantes do Noosfero que não são *committers*, embora não tenham a palavra final sobre o que é autorizado a entrar no repositório, têm a possibilidade de contribuir com a identificação de *bugs*, revisão de códigos, testes, por meio dos *commits* enviados. Essas interações de revisão referentes ao código, por sua vez, ocorrem na própria plataforma de hospedagem do código-fonte utilizada pelo projeto. Além dessas revisões, os membros do grupo costumam realizar testes automatizados no Noosfero, que auxiliam na identificação de problemas no código. Mesmo com essa normatização sobre o desenvolvimento, na prática, os entrevistados destacaram que a revisão do código não acontece de forma burocrática.

Quando algum usuário descobre um *bug*, se for usuário que tem conhecimento técnico, ele mesmo vai lá no *GitLab* e cria uma *issue*. Inclusive, esse é um estímulo que as pessoas façam isso, que elas relatem esses erros. Os testes automatizados são uma forma de identificar erro, mas

dentro de uma linha evolutiva. Não identifica erro numa situação estática, se não usar, não vai acontecer erro, o código tá parado ali. Os testes automatizados não vão identificar nenhum erro novo. Mas se o código do Noosfero evoluiu, aí o teste automatizado pode identificar um erro naquela evolução [...] O processo de entrada do código sempre passa pela revisão, que é uma prática que é adotada pela comunidade Noosfero. Mesmo as pessoas que tem a permissão de escrever diretamente no repositório, enviam uma solicitação para que os pares possam revisar, porque um segundo olhar identifica erros que aquela pessoa não viu. Então é uma prática saudável (E4).

Mas é uma revisão do ponto de vista [...] não é burocrático o processo, não. Tem um processo que manda o *merge request*, e ele é revisado. Se passar dias e não for revisado, o desenvolvedor que desenvolveu ele tem o direito de aprovar. Porque antigamente tinha um problema muito grande, tinha muito código para ser revisado. Ficava lá, ninguém revisava e nunca entrava, então, você desestimula as pessoas a participarem. Nesse último ano para cá, isso deu uma guinada, é tanto que você não vê quase nada para revisar, porque teve essa preocupação maior da comunidade (E7).

A gente tem os *committers* [...] E aí qualquer contribuição de alguém de fora precisa ser revisada e aprovada por uma ou mais dessas pessoas. Assim, mesmo entre a gente, a gente sempre usou esse processo de revisão de código. Já faz um certo tempo que a gente definiu um processo de aceitar novas pessoas (E8).

As políticas formuladas e executadas tendo por objeto a comunidade Noosfero aconteceram principalmente em função da descentralização que houve no projeto. O envolvimento de diversos atores externos no Noosfero foi alvo de um intenso debate no âmbito da comunidade, no sentido de definir os rumos da governança em torno do desenvolvimento do projeto. Assim, a Colivre, que até então terminava centralizando de certo modo a questão do desenvolvimento do *software*, teve a necessidade de em conjunto com os demais participantes do projeto estabelecer regras, vigentes até os dias atuais, Essas regras possibilitaram que a comunidade ao mesmo tempo que dispusesse de abertura no recebimento das contribuições externas, conseguisse manter a qualidade do código. Os papéis assumidos também passaram a ter responsabilidades atreladas a eles e deveriam, principalmente no caso dos *committers*, obedecer ao critério da meritocracia. A partir das respostas dadas pelos entrevistados:

Então, com o surgimento desses conflitos, a gente desenvolveu umas políticas de abertura do projeto para outras pessoas se tornarem desenvolvedores com permissão. A gente criou regras, em comum acordo com a comunidade e abriu o Noosfero na época para os desenvolvedores que já tinham experiência de outras organizações se tornarem *committers* também (E6).

3 anos atrás, na verdade, janeiro de 2015 [...] os únicos *committers* eram da Colivre, e a gente precisava de uma solução para iniciar um projeto de aceitar novas pessoas. Então, simplesmente, eu propus isso aí, com base na minha experiência de *software*. Porque a gente faz *software* livre, sempre foi

aberto, mas na prática, até essa época, ele foi meio que controlado de perto pela Colivre. Existia certa pressão da comunidade por isso. A gente nunca quis ser o dono eterno do Noosfero, era só uma questão prática justificada pelo nosso volume de trabalho (E8).

Além disso, existem também os procedimentos formalmente executados nos projetos paralelos que caminharam paralelos ao Noosfero. Apesar de esses outros procedimentos não ocorrerem dentro da comunidade em si, os resultados tiveram repercussão no *software* livre Noosfero. Esse controle formal verificado além da comunidade acontece, por exemplo, nos projetos executados pela Colivre para atendimento de algum cliente específico. Na verdade, existe todo um protocolo que é pré-definido pela cooperativa e que deve ser seguido, de forma que os objetivos prévios estabelecidos no escopo do empreendimento, com prazo estabelecido, precisam ser alcançados pelos funcionários da Colivre. Nesse sentido, encontra-se a declaração do entrevistado que trabalha na Colivre:

Na Colivre, a gente tem o gestor, a gente tem reunião, tem metas de planejamento, todo um processo organizacional na execução do projeto, mas que não necessariamente está relacionado com o Noosfero (E6).

Vale ressaltar que essa é uma situação bastante específica, que assim como os projetos do Serpro, da UnB, dentre outros, há toda uma rotina de comportamentos e resultados estabelecidos de modo formal pela organização ou pelos representantes que estão envolvidos diretamente em tais iniciativas. Ou seja, cada entidade tem a liberdade de ter a sua própria dinâmica formal de desenvolvimento. Essa questão é abordada por dois dos entrevistados:

Então, tudo que é feito no Noosfero, geralmente, todos os trabalhos que eu fiz em relação ao Noosfero de desenvolvimento mesmo, de entregar alguma coisa, nunca foi para o projeto de *software* livre Noosfero, sempre foi para o cliente x, y, z, que tá pagando, para aquela coisa, então, tinha sempre um prazo. Mas as pessoas que contribuem voluntariamente, quando o produto tá pronto ela manda. E aí, cada ator que está usando o Noosfero em seu contexto, vão ter prazos próprios (E4).

Tem cronograma [...] gestor de projeto, tem a parte burocrática para se adequar a algumas exigências que os clientes da gente pedem e tem os desenvolvedores. Dentro do time de desenvolvimento, a dinâmica que a gente estabeleceu na nossa equipe é a mesma que era lá fora no Noosfero. Então todo mundo atua em todas as pontas do desenvolvimento. Necessitando, atua em todas (E7).

De modo integrado à formalização, a incidência de procedimentos informais de controle foi notada no Noosfero, especialmente, pela via da socialização e interações existentes entre os membros do projeto.

No que diz respeito à interação interna, apesar de a comunidade do Noosfero ter apresentado períodos de maior atividade e outros menos ativos, sempre que possível os



integrantes mostraram cultivar o costume de fazer encontros dos seus integrantes. Nesse aspecto, em virtude até da descentralização dos membros, encontros sempre foram promovidos, majoritariamente, por meios virtuais, no sentido de verificar as atividades que estão sendo realizadas no projeto e de pensar coisas que poderiam ser executadas no futuro. Essa dinâmica tem sido mantida pelo *release manager* atual nos espaços oficiais da comunidade, mas sem uma formalização no sentido da periodicidade, ou dos assuntos discutidos. Sobre essa questão, os entrevistados manifestaram as suas opiniões:

A comunidade Noosfero, mais através do [nome do entrevistado E6], como *release manager*, tem puxado isso. Ele tem feito encontros virtuais da comunidade Noosfero por *voice*, com *hangout* ou alguma solução de voz. Porque tá ficando descentralizado, tá começando a ter necessidade disso. No passado, era muito no *core* da Colivre, já tava todo mundo lá e não era necessário. Mas agora a comunidade tá começando a levantar requisitos que são da comunidade. Aquilo que eu te falei, a comunidade não demanda nada, sempre é algum cliente que pede. Mas, nesses espaços, tá começando a ter esse exercício de coisas que são de necessidade do projeto, que não tem nenhum cliente pagando, e que precisa ser feito. Cada pessoa que tá lá fala, eu tô trabalhando nisso, eu tô fazendo isso [...] até tal dia eu vou entregar isso, semana passada eu tentei fazer tal coisa e deu erro, eu tô precisando de ajuda aqui, será que alguém consegue me ajudar? (E4).

A gente tem 3 espaços oficiais de discussão. Tem a lista de discussão que é o primeiro de todos. A gente tem um *chat* que a gente usa para coisas do dia a dia e é própria do Noosfero. A gente tem as reuniões da comunidade que acontecem mensalmente, quando existe comunidade, nesse cenário que eu falei, que em horas que aparece um monte de gente envolvida e aí a gente precisa organizar esse monte de gente (E6).

Outra forma de interação entre membros e que proporciona visibilidade para a comunidade externa ao projeto são as participações nos eventos da área de *software* livre. Nesse aspecto, os membros do Noosfero tiveram a oportunidade de ministrar palestras em todo o País com o foco em divulgar e trazer novos membros interessados em participar do projeto, como, por exemplo, no Fórum Brasileiro de *Software* Livre (FISL). A importância da participação nesses eventos é inegável.

É o que motiva. Por exemplo, quando eu tava ativo na comunidade, às vezes eu recebia uma remuneração para fazer algo relacionado à interface, mas me motivava querer tá no FISL [...] Cansei de submeter em nome da comunidade Noosfero propostas de palestra (E5).

O próprio reconhecimento que o Noosfero obteve por meio de projetos externos relevantes, premiações, reportagens, também são mecanismos que lhe garantem um reconhecimento e que funcionam como uma motivação para os integrantes voluntários participantes da comunidade.

Eu acho que existe um reconhecimento, assim, cultural. Todos os *commits* que a pessoa faz ficam registrados e isso tá disponível em todas essas ferramentas, dicionário de repositório. Eles mostram quem são os contribuidores (E6).

Os membros que contribuem atualmente com o Noosfero, e até os que estão afastados, todos demonstraram ter consciência do impacto que o projeto proporcionou no exercício das suas respectivas vidas profissionais. A própria percepção de existência da meritocracia no âmbito da comunidade também se destacou como um fator motivador e que teve importância no sentido de trazer um reconhecimento do trabalho realizado. Na declaração do entrevistado E8 este traço é exemplificado: ão desenvolvedor que hoje é o líder do projeto, ele entrou como estagiário da Colivre, virou sócio e hoje ele é o *release manager* também.

A socialização que aconteceu e continua acontecendo entre os integrantes do Noosfero internamente, bem como em face da comunidade externa, encorajam nos integrantes o autocontrole e o desejo de contribuir de alguma forma com o projeto. Mesmo entre aqueles indivíduos que fizeram parte da Colivre apenas durante algum período, se torna visível a existência de um senso de compromisso e de responsabilidade com o funcionamento do Noosfero, ainda que atualmente na situação de voluntários. Seja por interesses profissionais, seja pela busca de reconhecimento, de alguma forma esses integrantes permanecem auxiliando de forma autônoma, mas alinhada com os propósitos do Noosfero, como se vê na resposta de um dos entrevistados:

Hoje, a gente aprovou uma demanda aqui no Serpro para poder desenvolver o Noosfero, e eu estou há duas semanas desenvolvendo. Nessas duas semanas, eu faço a demanda que o cliente solicitou para o Serpro. Além disso faço contribuições para comunidade também, entendeu? Algumas coisas dessa demanda eu aproveito e vou fazendo essas contribuições. Então, normalmente, eu aproveito esses momentos para poder contribuir mais também com a comunidade. Se não der, quando eu estou em casa mesmo, eu faço contribuições mais pontuais (E7).

### 3.4.3 Projeto Kytos

O Kytos é uma plataforma de código aberto, que tem o propósito de facilitar o desenvolvimento e a implantação de redes definidas por *software* (*software defined networks* ó SDN). O projeto surgiu no ano de 2014, mas está em desenvolvimento ativo desde 2016, no âmbito do Núcleo de Computação Científica (NCC) da Unesp. A solução SDN é uma arquitetura que torna as redes ágeis e flexíveis, ao possibilitar que o administrador faça o

gerenciamento dessas redes, sem a necessidade de interligá-las e organizá-las fisicamente. Por ser uma plataforma SDN leve e de desenho modular, a ideia é que o Kytos seja facilmente utilizado para instalar, usar, desenvolver e compartilhar aplicativos de rede. A identificação dos indivíduos entrevistados no Kytos é apresentada no Quadro 9 a seguir.

**Quadro 9** Identificação e perfil dos entrevistados do projeto Kytos

<b>Código</b>	<b>Função</b>	<b>Observação</b>
<b>E10</b>	Coordenador/líder e desenvolvedor	Idealizador e líder do Kytos, e também funcionário do NCC da Unesp
<b>E11</b>	Diretor Científico do Núcleo de Computação Científica (NCC) da Unesp e <i>Project Integrator</i> (PI) do Kytos e de outros projetos do NCC	Docente da Unesp e diretor do NCC, que atua como PI, fazendo a articulação do Kytos e de outros projetos do NCC com o setor privado
<b>E12</b>	Desenvolvedor	Desenvolvedor que atuou como bolsista do projeto
<b>E13</b>	Desenvolvedor	Desenvolvedor que atuou como bolsista do projeto
<b>E14</b>	Desenvolvedor	Desenvolvedor que iniciou como voluntário e atualmente é bolsista do projeto

**Fonte:** Elaborado pela autora (2019)

As entrevistas foram realizadas com diferentes pessoas e ocupantes de diferentes funções no projeto: o diretor do NCC, o coordenador do Kytos e mais três desenvolvedores contratados como bolsistas do projeto. Dois desses desenvolvedores (E12 e E13), apesar de atualmente estarem afastados do projeto, tiveram importância fundamental para o empreendimento, em razão da quantidade de contribuições dadas ao código-fonte. O entrevistado E10, pela sua posição de liderança e o status de idealizador do projeto, atuou como informante-chave do Kytos, e foi ele a quem a pesquisadora recorreu nos momentos em que precisou esclarecer as dúvidas que surgiram durante o processo de análise das entrevistas e dos documentos examinados. O entrevistado E11 é docente da Universidade Estadual Paulista (Unesp) e diretor científico do NCC, atuando também como *Project Integrator* (PI), uma vez que, além de fazer a gestão administrativa, promove a articulação dos projetos do NCC (caso do Kytos) com o setor privado.

#### 3.4.3.1 Trajetória de desenvolvimento do projeto Kytos

O Kytos foi desenvolvido pela equipe do Núcleo de Computação Científica (NCC) do Instituto de Física Teórica, pertencente à Universidade Estadual Paulista (Unesp), em

parceria com o *São Paulo Research and Analysis Center* (SPRACE), centro que atualmente também integra esse núcleo. Para compreender as origens do Kytos, faz-se necessária uma explanação geral sobre como o NCC funciona e estabelece relacionamentos com atores organizacionais externos.

O NCC tem atuação expressiva em projetos de instrumentação, *hardware* e redes de computadores, de modo que compõe comunidades e possui parcerias nacionais e internacionais nessa área, destacando-se os contatos com os seguintes atores: Rede Nacional de Ensino e Pesquisa (RNP), *Academic Network at São Paulo* (ANSP), *California Institute of Technology* (Caltech), *Conseil Européen pour la Recherche Nucléaire* (Cern), *Americas Lightpaths* (AmLight), apenas para citar alguns exemplos. Os relacionamentos com essas entidades oportunizam aos integrantes do NCC a realização de projetos conjuntos, bem como a participação em diversos eventos no Brasil e no exterior.

Em um desses eventos, mais especificamente, no *SuperComputing*, uma das conferências mais relevantes do mundo no segmento da computação de alto desempenho, redes e armazenamento, realizado anualmente nos Estados Unidos, um dos integrantes do NCC teve a chance de participar da demonstração e transferência de dados utilizando os controladores SDN *open source* existentes. A consequência da participação nessa conferência possibilitou que o referido integrante, nesta pesquisa representado pelo entrevistado E10, identificasse uma série de limitações desses controladores, tanto no funcionamento, quanto no escopo de atuação, tal como descrito nas palavras dele:

Há alguns anos eu tenho ido aos EUA, para esse evento, para participar dessas demonstrações, e testamos alguns controladores SDN que existiam na época, todos *open source*, mas eles tinham algumas limitações ou eles eram feitos para um outro propósito, que não era o propósito do mundo acadêmico, das redes de pesquisa (E10).

No ano de 2014, em trabalho conjunto com membros do Caltech, a partir da identificação das limitações existentes nos controladores SDN, este membro do NCC iniciou o desenvolvimento de uma biblioteca *open source* denominada *python-openflow* (biblioteca que realiza a comunicação com o protocolo *openflow*). A criação dessa biblioteca foi o primeiro passo para o início do projeto Kytos. Em princípio, a ideia não era ter o desenvolvimento de um controlador SDN completo, mas, sim, obter uma biblioteca que possibilitasse a realização de testes e trouxesse maior entendimento da própria tecnologia SDN aos integrantes do NCC, já que na época se tratava de uma tecnologia recente. O entrevistado E10 abordou essa questão.

Eu sou contratado para resolver os problemas do núcleo de computação científica e consequentemente alguns problemas que a gente tava tendo era com relação a essas demonstrações que aconteciam nos EUA. Como essa tecnologia de SDN é uma tecnologia nova, o protocolo *openflow* era novo, tinha uma série de limitações, e a gente resolveu desenvolver a nossa própria biblioteca para poder facilitar os nossos testes e facilitar também o nosso entendimento da tecnologia, por assim dizer (E10).

Em virtude da identificação e alinhamento deste entrevistado com a filosofia de *software* livre, no decorrer da sua trajetória profissional e dos contatos estabelecidos com outras comunidades, desde os primeiros instantes, houve uma preocupação de que o desenvolvimento da biblioteca acontecesse de forma aberta, em formato comunitário virtual, por meio da plataforma *GitHub*.

Eu sempre tive um *background* muito forte de *software* livre. Então, desde o primeiro *commit* o projeto já estava no *GitHub*, desde as primeiras semanas, mesmo quando eu tava trabalhando sozinho nesse projeto (E10).

Sendo assim, em função dessa abertura e disponibilização da biblioteca ao público externo, pessoas interessadas passaram a utilizá-la, de forma que o projeto apresentou certo crescimento e visivelmente se tornou mais efetivo em 2016, conforme os relatos a seguir.

Inicialmente, não era a ideia fazer um controlador SDN completo. A ideia [...] a gente começou fazendo na biblioteca, em *python*, que hoje é chamada de *python openflow*, e essa biblioteca é que faz a comunicação com o protocolo *openflow*. E isso foi amadurecendo, foi evoluindo, algumas pessoas foram usando essa biblioteca e a gente foi crescendo, ganhando corpo, ganhando forma e o projeto acabou se tornando uma plataforma (E10).

O projeto teve início em 2014, que era praticamente o auge de tecnologia de rede, *openflow*. Foi ali que o projeto ficou muito popular e teve vários outros projetos análogos ao Kytos, e daí então teve esse pico inicial e o projeto veio caminhando e buscando voluntários. Depois, em 2016 praticamente o projeto tomou corpo devido a vários outros interessados, parcerias com outras universidades, pesquisadores, que também tinham interesse em escrever teses de mestrados, explorar novos protocolos em cima do Kytos (E14).

Um dos acontecimentos fundamentais que impulsionou o desenvolvimento do Kytos, ocorreu no final de 2015, quando houve a entrada do patrocínio da empresa Huawei. Essa organização chinesa da área de telecomunicações investiu recursos financeiros e materiais no Kytos por intermédio da Lei da Informática. Essa lei, em linhas gerais, estimula que empresas desenvolvedoras de bens ou serviços de informática invistam em projetos e atividades de pesquisa e desenvolvimento (P&D) para pleitear, junto ao Ministério da Ciência, Tecnologia e Inovação (MCTI), a obtenção de benefícios fiscais (Lei nº 8248/91).

A parceria com a empresa Huawei aconteceu principalmente em decorrência dos contatos estabelecidos pela gestão do NCC com entidades externas. Nesse aspecto, destaca-se

o protagonismo do docente e diretor do NCC (o entrevistado E11), principal articulador desses contatos e promotor de parcerias externas. Em uma das oportunidades, este diretor, em conjunto com outros parceiros acadêmicos, viajou a convite da Huawei para conhecer a empresa na China. Desse relacionamento, resultou o interesse da Huawei em investir no projeto Kytos, como declarou o próprio docente:

Veja, a gente tem uma boa reputação no mercado em termos de eficiência no processamento de alto desempenho, nós fomos procurados por eles e eles convidaram para uma visita à China, para conhecer um pouco a empresa. Nós fomos para a China e na China nós desenvolvemos esse tipo de relacionamento (E11).

Na relação entre a Huawei e o projeto Kytos, o MCTI desempenhou, e desempenha, um papel fundamental, já que é um dos principais responsáveis pelo acompanhamento necessário ao cumprimento da supracitada lei. Para conseguir os recursos do patrocínio, os integrantes do Kytos precisam, constantemente, informar em relatórios repassados para a Huawei os principais andamentos do projeto. A Huawei, por sua vez, a partir dos relatórios recebidos realiza a prestação de contas junto ao ministério, para que assim seja viabilizada a isenção fiscal prevista em lei. Nas palavras do entrevistado E10, ãa função do MCTI é fiscalizar a empresa e não o projeto... que é um projeto sério e que os relatórios estão sendo cumpridos.

Com o patrocínio da Huawei tornou-se possível a contratação de bolsistas para auxiliar no desenvolvimento do projeto. Nas palavras de dois entrevistados (E10 e E11), a obtenção do patrocínio veio em momento oportuno e crucial para o crescimento do projeto.

Então nós chegamos numa hora boa, fizemos uma proposta e começamos com uma sobra de caixa que eles tinham no final do ano e aí a gente começou a desenvolver e o projeto progrediu nos últimos 3 anos (E11).

Esse aporte da Huawei chegou no momento que a gente estava precisando e foi quando, de fato, deslançou o projeto para a gente com as primeiras contratações e tudo mais (E10).

As transformações que permitiram esse crescimento do projeto, portanto, foram viabilizadas essencialmente pela entrada de recursos do patrocínio. A biblioteca criada nos estágios iniciais tornou-se um repositório à parte, constituindo, atualmente, um subprojeto que é abarcado pelo projeto macro da plataforma SDN Kytos, conforme explicou E10:

O projeto acabou se tornando uma plataforma por si só [...] A biblioteca é um repositório à parte. Ela é um projeto pequeno hoje, do projeto como um todo, que é o Kytos, o que a gente conhece hoje como Kytos, que é a plataforma Kytos (E10).

Apesar de o projeto ter o apoio da Huawei, os entrevistados foram enfáticos ao garantir que esse patrocinador exerce uma influência mínima em relação ao desenvolvimento do *software*:

A gente está fazendo pesquisa e desenvolvimento para coisas que nós consideramos relevantes. Porque se não tivesse, eu não teria me envolvido nele, porque a gente não é prestador de serviço, nós somos uma universidade e como universidade, uma das obrigações-mestra nossa é gerar conhecimento, gerar produto relevante (E11).

No período que eu estava lá, que eu vivenciei o projeto, eu acho que a interação foi muito pequena. Eu, pessoalmente, durante o desenvolvimento do projeto, nas discussões, no que vamos fazer, no que não vamos eu não vi interação, tirando os relatórios que a gente tinha que entregar (E13).

Até os dias atuais, o Kytos permanece adotando a lógica de funcionamento aberta, na qual qualquer pessoa pode ter acesso ao seu código-fonte e realizar contribuições para o projeto, um detalhe frisado por E10 ao afirmar que ãos primeiros bolsistas já entraram sabendo da licença, já sabendo de como iriam trabalhar, que teria que ser tudo público, tudo aberto.õ Essa participação comunitária ajudou principalmente na identificação de *bugs* que permitiram o aperfeiçoamento do projeto e tornaram essa abertura do Kytos algo positivo, conforme o relato de E13:

Tem todo um processo padrão de *software* livre, de você olhar lá os termos, se alinhar, assinar, todo o código que escreveu, e tal [...] Essas coisas ajudaram a achar alguns erros que a gente não teria achado sozinho. Então, para o cenário do projeto como *software* livre, acho que tiveram alguns passos que foram dados por pessoas que, mesmo que elas não tenham participado ativamente, foi benéfico para o projeto no final. Nesse aspecto foi bom, porque a gente acabou tendo uma visão diferente (E13).

No entanto, como se deduz, apesar de ser um *software* livre e disponibilizado no formato comunitário, a contratação de bolsistas para o desempenho das atividades de desenvolvimento do projeto se fez necessária. Isso pode ser explicado pela especificidade do *software* e do seu direcionamento para aplicação em ambientes acadêmicos, o que faz com que a comunidade do Kytos tenha atratividade para um público de desenvolvedores específico e restrinja a participação só voluntária. Essa questão foi abordada pelo entrevistado E10:

O projeto ele não é um projeto tão grande assim. Então, a gente está ainda tentando criar essa comunidade. Não existe essa comunidade tão forte. A gente tem um canal no *Slack*, tem algumas pessoas que ficam *online*, mas a gente não é projeto de *software* livre ainda maduro o suficiente, apesar da gente estar aí desde 2015, mais ou menos, trabalhando com isso, um nicho muito específico de provedores de redes acadêmicas (E10).

O entrevistado E13, desenvolvedor e ex-bolsista, declarou que a visibilidade adquirida pelo Kytos, no sentido de atrair colaboradores voluntários, ainda é algo recente e

que tem acontecido especialmente em função das pesquisas que possibilitaram maior divulgação do projeto:

Outras pessoas de fora podem participar, mas isso não era comum, porque a gente começou a ficar sendo conhecido a partir do segundo semestre de 2016. Agora que a gente já tem redes estáveis, que a gente começou a apresentar o controlador num monte de lugar, que tem um pessoal que tá desenvolvendo algumas pesquisas específicas que botou nos *papers*, começou a dar visibilidade para o Kytos (E13).

Apesar de contar de modo proeminente com o suporte de bolsistas, alguns voluntários têm participado do Kytos, representados, principalmente, por indivíduos integrantes das parcerias nacionais e internacionais estabelecidas pelo NCC e listadas anteriormente. Esses voluntários, em conjunto com os bolsistas, têm atuado em diversas frentes necessárias ao desenvolvimento do projeto. O *software* faz uso da licença MIT, que é uma licença de *software* livre permissiva criada pelo *Massachusetts Institute of Technology* (MIT), encontrando-se atualmente disponível na versão 2018.2.

#### 3.4.3.2 Estrutura do projeto Kytos

A partir da narrativa apresentada da história do Kytos, nota-se que apesar de ter surgido de uma iniciativa individual, o projeto desde o início teve o suporte de infraestrutura física e tecnológica da Unesp, e, posteriormente, contou com os recursos do patrocínio da empresa Huawei. Além de contar com esse apoio, o Kytos faz uso de recursos de TIC que possibilitam tanto a comunicação interna dos seus integrantes, quanto a interação com a comunidade externa. Dentre esses recursos, destacam-se a utilização de *home page*, visando a divulgação de informações sobre o projeto, as listas de *e-mail*, a plataforma de hospedagem do código-fonte, o *chat* e a videoconferência. Conforme dito nas entrevistas, em termos de infraestrutura e de gestão necessários ao funcionamento do Kytos o projeto é viabilizado por três meios principais:

Do ponto de vista financeiro, ainda é o recurso da Huawei, que entra para conseguir pessoal dedicado para trabalhar no projeto. Do ponto de vista estrutural, ainda é a Unesp, o Sprace, que o mantém lá dentro. E do ponto de vista ideológico, tem [nome do entrevistado E10], que empurra, impulsiona o projeto. Se não tivesse tendo os avanços que eles estão conseguindo, talvez o projeto estaria menos maduro, com menos funcionalidade, ou com menos visibilidade (E13).

Na condução do Kytos é percebida a existência de uma hierarquia formal, cuja autoridade maior é o diretor científico do NCC, que assume a responsabilidade administrativa dos projetos do núcleo e faz o gerenciamento dos contatos externos desses projetos com



instituições públicas e privadas. Subordinado a esse diretor encontra-se o líder ou coordenador do projeto, que é o indivíduo a quem se atribui a ideia de criação do Kytos, sendo também o responsável pela gestão direta da equipe de desenvolvimento do *software*. No grupo de desenvolvedores, finalmente, localizam-se os bolsistas contratados para o projeto e os voluntários que atuam informalmente. Nas palavras dos entrevistados:

Tem hierarquia, obviamente. Eu não tenho nem conhecimento nem vontade de colocar a mão em código, mas um funcionário direto meu liderou essa iniciativa. Então as pessoas contratadas para o projeto ficaram sob a supervisão direta dele (E11).

O diretor administra o projeto de forma mais administrativa, digamos assim. Ele não está muito envolvido com o desenvolvimento do projeto em si, mas apenas com questões de contratação, gerência dos recursos dos relatórios da Huawei, do Ministério de Ciência e Tecnologia. Eu estou mais com o papel de coordenador [...] porque tenho um pouco a visão do projeto como um todo e abaixo de mim existem apenas os bolsistas que estão desenvolvendo. A gente tem uma hierarquia *flat*, a gente não tenta manter uma hierarquia para baixo, até porque cada um toma conta de uma área do projeto ou coisa desse tipo (E10).

Verifica-se, assim, que o diretor realiza o trabalho por meio de uma perspectiva mais macro da gestão do projeto, responsabilizando-se por questões administrativas e pelas relações institucionais do Kytos com atores organizacionais externos. Já o coordenador, por sua vez, é figura presente não só no desenvolvimento da tecnologia, como em toda a coordenação dos bolsistas e voluntários, uma atuação necessária para que esse desenvolvimento do *software* aconteça. Após o coordenador, no entanto, ou seja, entre os desenvolvedores, o projeto não apresenta relações de subordinação, de modo que todos evidenciaram ter uma atuação igualitária nas suas contribuições em favor do empreendimento. Alguns dos desenvolvedores entrevistados falaram sobre a visão que possuem da hierarquia:

Eu acho que logo no início do projeto, quando eu entrei, pelo menos, essa hierarquia era um pouco menos visível. Eu acho que a coisa estava com um pouco mais [...] uma gestão mais horizontal das coisas. Mas depois, com o tempo, a hierarquia ficou bem clara (E12).

[Nome do entrevistado E10] dedica tempo para isso. Ele dedica o tempo dele, do trabalho dele na Unesp para isso, então o trabalho acaba indo adiante por causa do tempo que ele dedica, mas não existe hierarquia além dele. Todos éramos iguais (E13).

Bom, hierarquia existe. O projeto como um todo faz parte da Unesp. Então, apesar de ser *open source*, tem hierarquias de desenvolvimento e de gestão como um todo. Falando na parte de desenvolvimento, a hierarquia era praticamente mínima. [Nome do entrevistado E10] acaba sendo o líder do projeto. Já que ele foi o fundador, ele conhece os valores [...] a visão inicial do projeto, e para onde ele acha que o projeto deveria estar indo. O time de desenvolvimento é composto por várias pessoas. Em média, normalmente, são 5 desenvolvedores, mas não existia nenhuma hierarquia (E14).

Em relação aos papéis formais que aparecem associados a essa hierarquia, tal como já relatado anteriormente, destaca-se a existência do diretor, do coordenador e dos bolsistas. O coordenador do projeto mostrou ser figura fundamental ao longo de toda a história do projeto, assumindo a liderança que direciona as principais ações e os rumos do desenvolvimento do Kytos. Esses papéis que assumem contornos formais convivem com os dos integrantes voluntários, que operam especialmente nas atividades de desenvolvimento ou colaborando com as documentações do projeto. Esses integrantes voluntários, vale ressaltar, pertencem principalmente às instituições da área de redes acadêmicas com as quais o NCC mantém parcerias, tais como a AmLight, ANSP e RNP.

Como o projeto é relevante para as atividades deles, eles gostam de discutir as *features* que vão entrar no projeto, porque, querendo ou não, eles vão usar. Eles participam das reuniões, colaboram com código, com documentação com a escrita de requisitos e no projeto como um todo eles são bem ativos (E10).

Apesar de existir a divisão de papéis formais em termos de gestão, no desenvolvimento do *software*, em si, ainda que responsabilidades sejam atribuídas aos integrantes, estes não se limitam apenas a executar as funções de modo especializado. Sendo assim, conforme a necessidade do projeto, os integrantes podem trabalhar em diferentes frentes, além dos papéis que lhes forem formalmente atribuídos. Os entrevistados comentaram essa possibilidade de atuação em múltiplas funções do projeto:

Eu acho que as pessoas vão sendo demandadas pelo que tem no momento, independente da *expertise* de cada um ou do que cada um quer fazer, atuar, então era muito sob demanda nesse sentido. De acordo com o que a hierarquia decide (E12).

Existiam algumas responsabilidades maiores, que eram de um e de outro. Por exemplo, um dos bolsistas que trabalhou comigo ficou responsável por montagem da esteira de integração contínua, mas não significa que só ele trabalhasse nisso, a gente ajudava. Eu fiquei responsável pelo desenvolvimento da biblioteca, que dá suporte para plataforma, mas também não era só eu que mexia nela (E13).

De forma alinhada com a gestão formal do Kytos, nota-se, também, a existência de outra gestão com caráter informal, associada aos conhecimentos e experiências adquiridos pelos integrantes em relação ao projeto. Nessa hierarquia informalizada, é visível que um contexto de meritocracia tenta ser estabelecido, de acordo com as percepções do coordenador sobre as habilidades de determinado integrante, em ganhar mais direitos sobre o projeto. Na figura central do coordenador, e por iniciativa dele, os desenvolvedores mais ativos do Kytos podem também ser convidados a se tornar *committer* do projeto. O próprio coordenador relatou como essa autoridade lateral se manifesta no contexto do Kytos:

A gente tem os desenvolvedores mais antigos, que conhecem mais do código. Eles entendem um pouco mais do código e têm um pouco mais de liberdade no quesito de mexer nos repositórios, na autenticação, autorização, e existem os colaboradores novos, que são colaboradores que estão entrando, que geralmente são os colaboradores externos. Eles ganham esses direitos [...] São conquistados ao longo do tempo à medida que ele vai provando que ele está engajado no projeto ou coisa do tipo, mas a gente não tenta criar uma estrutura definida no núcleo. Basicamente, o que a gente faz é por tempo de contribuição e pelo empenho dessa pessoa no projeto (E10).

O desenvolvedor E14 corrobora com essa visão de que existe um contexto de meritocracia ao citar a sua própria experiência no Kytos, uma vez que iniciou como voluntário e depois começou a trabalhar no projeto como bolsista remunerado.

O meu contato com o Kytos aconteceu há praticamente um ano, quando eu estava trabalhando na RNP. A gente tem uma rede que precisava de um controlador SDN e parte do projeto fazia com que a gente ficasse se atualizando com o que havia disponível no mercado e o Kytos era um desses projetos, era uma alternativa *open source*, aberta, que também tinha uma parceria muito forte com a universidade. Foi a partir de então que eu acabei acompanhando o projeto. Nesses últimos seis meses eu estou como bolsista. No início eu entrei como voluntário (E14).

Ao mesmo tempo em que o Kytos tenta conciliar hierarquia e papéis que transitam entre o formal e o informal, existe internamente uma forte preocupação do coordenador com a manutenção da qualidade do código, o que faz com que as decisões consideradas por ele como estratégicas e relevantes para o futuro do projeto aconteçam de modo centralizado. Até mesmo em decorrência de E10 ter sido o criador do projeto e exercer a autoridade formal que lhe é delegada pelo diretor do NCC, para gerenciar a equipe de desenvolvimento, existe uma responsabilidade na gestão dele que tem repercussão direta na questão da centralização de decisões, então percebida pelos desenvolvedores. No entanto, embora a centralização em torno das questões estratégicas seja uma realidade, o coordenador afirmou que tem o cuidado de assegurar que todos os envolvidos tenham espaço para manifestarem as suas opiniões nas tomadas decisões, o que significa, na prática, buscar o consenso sempre que possível:

O que acontece é que eu exponho o meu desejo, deixo bem à vontade o que a gente gostaria de implementar, as pessoas criticam, discordam, concordam. Como é uma equipe pequena [...] a gente tenta o consenso (E10).

No decorrer do tempo, até mesmo por conta de o líder estar envolvido em outras demandas do NCC que extrapolam o Kytos, na percepção dos entrevistados, passou a existir a necessidade de o coordenador delegar mais as decisões, tornando, assim, o ambiente mais participativo. Essa maior descentralização que tem sido vivenciada no decorrer do crescimento natural do Kytos tornou a centralização de determinados assuntos inviável para o fluxo de funcionamento do projeto, ainda que algumas deliberações permaneçam

concentradas no topo da hierarquia. O próprio coordenador do empreendimento reconheceu que só mais recentemente tem se preocupado em incentivar maior participação dos interessados no processo decisório.

No começo do projeto as decisões eram feitas internamente, por nós que estávamos tentando montar o projeto, não tinha muita gente. Mas agora a gente faz uma reunião aberta a cada seis meses, que é o *Kytos Summit*. As primeiras foram feitas presencialmente. Mas, atualmente, por falta de dinheiro, a gente tem feito *online* de forma remota no *Hangouts*. Mas a gente sempre inclui a todos, é bem aberta essa reunião, todo mundo pode participar, discutir e tomar decisões juntos (E10).

### 3.4.3.3 Processos de controle do projeto Kytos

Como o Kytos é uma iniciativa que, além do suporte fornecido pela Unesp, recebe patrocínio da empresa Huawei, alguns processos de controle precisaram ser internamente regulamentados para garantir o funcionamento do projeto em sintonia com esses atores organizacionais. Na página do projeto, por exemplo, encontram-se disponíveis diversas informações sobre de que maneira as pessoas podem colaborar com o Kytos, como contribuinte individual, *committer*, afiliado (associação de entidades beneficentes ou sem fins lucrativos que queiram usar ou apoiar o projeto) ou até mesmo na condição de patrocinador. Para se tornarem colaboradores, os indivíduos ou entidades interessadas precisam explicitar que estão cientes e de acordo com os termos de licença do projeto.

Tem um guia do contribuidor, que é um manual que tem orientações de como eles devem contribuir para o projeto, ou seja, como eles devem submeter o *pull-request*, como que eles devem abrir uma *issue*, como eles devem documentar as coisas. É um guia simples, pequeno, sem muitos detalhes, apenas para gente ter um controle da qualidade do código. Tem apenas esse manual que a gente deve seguir. Primeiro, a pessoa tem que assinar o nosso documento de contribuidor. Dentro do *site* kytos.io existe um CLA, que é o *Kytos Contributions Licence Agreements*, que todo o projeto de *software livre* tem esse documento para se no futuro a gente resolver trocar de MIT para Apache, não fechar o código, porque não é nossa intenção jamais fechar o código, mas se tiver alguma divergência, alguma coisa nesse sentido, a gente não precise colher todas as assinaturas dos desenvolvedores do passado (E10).

Tinha a necessidade de assinar um termo de cessão de propriedade intelectual e um detalhe técnico de que todos os *committers* tinham que assinar com uma chave de criptografia. Fora isso, assim, não via nenhum outro tipo de norma ou regulamento (E12).

No nosso caso ali do Kytos, a gente tem todas essas diretivas de contribuição de como os colaboradores podem estar instalando os projetos, contribuindo. E tem, literalmente, um passo a passo explicando como funciona, quais são as expectativas também. Então, na página de documentação, já tem algumas

seções específicas de como colaborar com o projeto e daí já existe uma divisão, porque a gente identificou, ao longo do tempo, que tem dois usuários específicos. Tem alguns que só querem usar o projeto como usuário final e outros como desenvolvedor (E14).

O entrevistado E12 destacou que grande parte dos procedimentos formais foi descrita no *site* do Kytos por volta do final de 2017, o que pode ser explicado até mesmo pelo crescimento efetivo que o projeto vinha exibindo desde 2016. Dentre as formalizações definidas no projeto, ainda que os passos para se tornar *committer* estejam descritos no *site*, na prática, a ocorrência desse papel está atrelada à percepção subjetiva do coordenador do projeto, no sentido de considerar, ou não, a pessoa como merecedora de assumir essa responsabilidade. Percebe-se, então, que existe uma relação do papel formal de *committer* com a hierarquia informal e o modelo de meritocracia que se tenta implementar no projeto.

Em relação aos resultados que precisam ser alcançados, como o Kytos recebe recursos de patrocínio, o projeto precisa constantemente informar à Huawei sobre o status das atividades desenvolvidas. Essa prestação de contas acontece por meio de relatórios para que a empresa, por sua vez, forneça as informações necessárias ao cumprimento das prerrogativas estabelecidas pela Lei da Informática junto ao MCTI. Apesar de o Kytos ter autonomia para o desenvolvimento do *software*, algumas entregas específicas para a Huawei precisaram ser cumpridas pelos integrantes do projeto. Os relatos dos entrevistados trazem o entendimento sobre o controle formal existente em decorrência do patrocínio.

Nas etapas iniciais, a gente prometeu algumas atividades, algumas tarefas [tal módulo precisa estar pronto, tal funcionalidade precisa estar pronta]. Então, nesses relatórios a gente manda os links dessas coisas que a gente prometeu e que a gente concluiu. No relatório a gente fala o que entrou em cada *release*: [tivemos uma *release* no mês tal, nessa *release* entrou tais *features*, nós fizemos isso em conformidade com aquilo que a gente tinha prometido]. Eu acredito que, inclusive, algumas pessoas da Huawei estejam inscritas na nossa lista de discussão, mas a gente não manda nenhum *e-mail* especial para os patrocinadores, a gente é bem independente. O relatório é a parte burocrática e as *releases* representam o nosso desenvolvimento do dia a dia (E10).

A gente, em geral, apresenta o projeto em *milestones* que estão sendo cumpridas. A gente tem que mostrar que essas *milestones* foram efetivamente desenvolvidas e cumpridas a contento. Então, basicamente é isso. O Ministério tem uma formalização, um roteiro para esse tipo de relatório e a gente procura seguir aquilo (E11).

Na realidade, quem tem que prestar conta é a empresa e a gente tem que, indiretamente, prestar conta para a empresa, que deve prestar conta ao Ministério, porque é uma legislação fiscal (E12).

Os relatórios repassados para a Huawei possuem periodicidade anual e o patrocínio dessa empresa tem sido renovado nos últimos anos de funcionamento do Kytos. A construção

do relatório se caracteriza por ocorrer de maneira colaborativa entre os integrantes do projeto, com a supervisão do coordenador, documento em que são relatadas especialmente as atividades desenvolvidas pelos bolsistas contratados com a verba do patrocínio. Apesar de o trabalho ser colaborativo, a responsabilidade final de revisão e envio do relatório, de maneira articulada com o patrocinador, é função executada pelo diretor científico do NCC.

Quem faz os relatórios sou eu com a ajuda da equipe. Eles me passam o relatório da parte deles, do que eles fizeram. O diretor só revisa, ele coordena para ter certeza que aquilo vai ser entregue para pessoa correta da Huawei, que aquilo está em conformidade com aquilo que a gente prometeu (E10).

A publicação de novas versões constituiu um processo bem definido no projeto, existindo uma periodicidade para os lançamentos das versões não estáveis e das estáveis. Nas reuniões que o projeto costuma realizar a cada seis meses, conhecidas como o *Kytos Summit*, costuma-se, inclusive, debater direcionamentos para o lançamento da próxima versão do *software*. Quando uma nova versão é publicada, existe sempre a divulgação dela pelos meios virtuais usados pelo projeto, em especial no *Slack* (ferramenta interna para colaboração de equipes), nas listas de *e-mail* e no *GitHub*. O líder, por ser a pessoa que coordena todo o processo de lançamento de versões, relatou como ocorre o cumprimento desse processo.

Então nós temos a cada seis meses um *release* estável. Mas entre essas *releases* estáveis, nós temos três versões beta. Então, por exemplo, a gente tem em 2018 o Beta 1, 2 e 3. Depois a gente lança a *release* 2018.1, que é a versão estável do 2018.1. Quando a gente vai para o 2018.2, a gente tem Beta 1, Beta 2 e Beta 3, até lançar a 2018. 2, a final. Então, quando a gente faz essas reuniões a cada 6 meses, a gente define quais são as *issues*, quais são os problemas que vão entrar na B1, quais são os problemas que vão entrar na B2. Então, por uma questão de organização, de prioridade, a gente dá sempre prioridade pela ordem em que elas foram elencadas nos *milestones* (E10).

Os desenvolvedores envolvidos no projeto destacaram que apesar da autoridade exercida pelo coordenador no acompanhamento do fluxo de lançamento de versões, ultimamente notaram que esse processo vem ocorrendo de forma mais descentralizada. Além disso, várias etapas da publicação da licença têm acontecido de modo automatizado. Essas constatações são percebidas nas falas de dois desenvolvedores, um já afastado do projeto, e outro, atualmente, bolsista do Kytos:

As versões têm uma periodicidade específica, isso está no *site*, quais são as datas, como são calculadas as datas para os *releases* [...] dois por ano. Na minha época, acontecia assim: [nome do entrevistado E10] olhava todos os repositórios do projeto, junto com todo mundo. Ele era o principal responsável por fazer as compilações e de criar as *tags* no *GitHub*. Quando

eu saí, isso estava mudando, porque ele tem outras obrigações lá na Unesp (E13).

Basicamente tem as versões maiores, que ocorrem a cada 6 meses. Só que no decorrer disso também tem algumas versões menores que vão saindo ao longo do tempo. E a maneira que vai saindo isso é da seguinte forma: uma vez que as funcionalidades vão sendo entregues, alguns probleminhas vão sendo corrigidos. A gente tem todo o mecanismo interno de fazer a geração automática de versão e disponibilizar automaticamente para os usuários uma nova versão do Kytos. Então, é tudo automatizado. Tem o processo escrito, mas é algo que o projeto está bem preparado para iniciar uma nova versão o quanto antes e divulgar externamente. O responsável pelo processo hoje está sendo [nome do entrevistado E10]. A gente divulga nas listas de *e-mail*, *GitHub* e também em outros meios alternativos (E14).

Em paralelo a esses processos formalizados de controle, existem práticas que ocorrem no projeto por meio de um perfil informal, com o foco em promover interações entre os participantes do projeto. Nesse aspecto, ainda que reuniões de trabalho aconteçam regularmente para acompanhamento do trabalho formal desempenhado pelos bolsistas, outros encontros com caráter mais informal para debater questões específicas do trabalho também acontecem no Kytos, principalmente pelos meios virtuais. Os entrevistados relataram suas visões sobre os encontros de trabalho do projeto:

A gente faz reunião via *hangout*, duas vezes por semana, mas com as pessoas mais próximas. Porque a gente tem que cumprir essas metas que a gente prometeu no relatório. Na verdade, essas reuniões semanais é mais no *Scrum*. É parte da metodologia ágil você ter reuniões diárias, digamos assim, só que ao invés da gente fazer reuniões diárias, a gente faz a cada dois dias. Não é bem uma reunião com pauta, não é nada disso, é mais uma reunião para saber o que a pessoa fez. É mais uma reunião para destravar do que para travar, não duram mais de meia hora. Na verdade, é uma reunião só para poder a turma toda saber o que eles estão fazendo [Está com alguma dúvida? Vou por aqui ou vou por ali?] A ideia é que seja uma reunião bem rápida mesmo (E10).

O time de desenvolvimento se encontra duas vezes por semana. Varia o dia, mas ultimamente está sendo terça e quinta. Basicamente, cada membro explica o status do que está desenvolvendo, se precisa de ajuda, revisões e tudo o mais (E14).

Além das interações direcionadas à resolução de questões referentes ao projeto, o Kytos, nos últimos anos, tem organizado um evento voltado não só para a integração dos seus membros, mas também com vistas a estabelecer contatos com o público externo, que é o chamado *Kytos Summit*. Esse evento, que acontece a cada seis meses, teve as suas primeiras edições em moldes presenciais, mas, pela própria dificuldade de operacionalização, as últimas edições passaram a acontecer nos espaços virtuais do coletivo, com destaque para o uso da plataforma do Google *Hangouts*. Esse evento, além de promover a interação entre os

indivíduos, é um momento de articulação para o planejamento de ações futuras a serem desempenhadas no projeto.

Como esse projeto é um projeto que quem usa são provedoras de redes acadêmicas, a gente tenta fazer um evento a cada 6 meses, em que a gente discute o futuro do projeto, as *features* que vão entrar no projeto [...] A gente fez um evento aqui no ano passado, que tem até algumas fotos no *site*, que deu aproximadamente 50, 60 pessoas. Mas eram todos parceiros. Nem todos ali estavam, de fato, efetivamente contribuindo para o projeto diretamente. Antes esses encontros eram presenciais e agora não são mais (E10).

Como o Kytos tem vários usuários e parceiros, o Kytos tem uma espécie de conferência, com o nome de Kytos *Summit* e que acontece a cada 6 meses. Normalmente, acontece na Unesp [...] e daí os participantes vêm, trazem sugestões e todo mundo colabora ali com as ideias novas e o que eles gostariam de estar vendo, e eventualmente a gente tenta encontrar quais são os interesses mais em comum. Depois, ao longo desses 6 meses a gente vai dividindo esses interesses em *milestones* que seriam entregáveis. É mais ou menos assim que fica o nosso *roadmap* (E14).

O reconhecimento dos integrantes do projeto foi outro meio de promoção do controle social. Esse reconhecimento apareceu de modo atrelado à visão da meritocracia existente no Kytos, sendo uma das formas encontradas pela gestão para trazer maior motivação aos membros ativos do projeto, sejam bolsistas ou voluntários. Na percepção da gestão, os desenvolvedores que se destacarem pelas contribuições dadas ao *software* possuem a chance de participar de eventos ou de ministrar cursos no Brasil e no exterior, na área de atuação do projeto.

O reconhecimento se dá mais no sentido de mandar as pessoas para fazer as apresentações. As pessoas que são mais destacadas no projeto a gente envia para dar alguns cursos, tanto no Brasil, como para fora. Tanto nos EUA, na Bulgária, tem os eventos que o próprio Cerne organiza [...] Então, aqueles que estão mais ativos no projeto, acabam viajando para dar esses treinamentos. Como o projeto está ganhando visibilidade agora, eu acredito que isso vai ficar um pouco mais visível nos próximos meses (E10).

Quando a gente está nas conferências, as pessoas vêm os desenvolvedores ali, chegam, agradecem e pedem novas coisas. Então, acho que o reconhecimento acaba sendo bem legal (E14).

Essa participação em eventos, assim como as produções técnicas e científicas sobre o Kytos apresentadas em congressos ou conferências da área, também influencia no sentido de aumentar a visibilidade não só do projeto e dos seus integrantes, mas também o estabelecimento das interações com atores externos.

De vez em quando a gente tem alguns eventos internacionais que a gente vai apresentar. Sempre estão indo uma ou duas ou três pessoas, e a gente acaba se encontrando também, batendo um papo e adiantando algumas coisas. Todo mundo vê um evento interessante, um evento importante, a gente vai e



acaba escrevendo um *paper* [...] Ou então, as vezes a gente é até convidado para ir, e a gente acaba indo (E10).

O desenvolvedor E12 destacou que os próprios recursos de TIC utilizados pelo Kytos, por permitirem que alguns registros das contribuições dadas pelas pessoas ao projeto fiquem registradas virtualmente, se caracterizam por ser meios que promovem o reconhecimento do trabalho realizado.

As pessoas sabiam quem contribuía mais ou não. Pelo lado individual, é fácil saber quem está produzindo mais, contribuindo mais. E se você acompanhar os grupos de discussão, seja por *e-mail* ou o que for, você consegue ver as pessoas que são mais dedicadas (E12).

Outra medida que também permite ao Kytos interação e visibilidade perante a comunidade externa acontece mediante as notícias divulgadas, algo especialmente notado quando o projeto recebeu o patrocínio da Huawei. A liderança do empreendimento, inclusive, destacou que a Unesp tem jornalista científico voltado para fazer as divulgações para a imprensa dos principais acontecimentos do projeto como, por exemplo, quando os integrantes do Kytos participam das demonstrações do *software* na conferência SuperComputing, que acontece nos Estados Unidos e que é um evento anual de referência no mundo na área de computação de alto desempenho.

O controle social propiciado pelas interações dos integrantes do Kytos, entre os membros internos, e entre estes com o público externo, ou por meio das parcerias, também se alinham com a modalidade de direcionamento dos comportamentos representada pelo autocontrole. Nesse sentido, enquanto os integrantes que são bolsistas exercem trabalho formal no projeto, os indivíduos que são voluntários e possuem maior autonomia, contribuem com o Kytos principalmente pela afinidade com o empreendimento e também pela relevância do *software* no campo profissional em que atuam. A existência dessa identificação demonstrou ser um fator que faz com que os esforços dos voluntários sejam guiados em conformidade com os objetivos do projeto.

Esses colaboradores externos contribuem da forma como eles acham que devem contribuir, porque afinal de contas a gente não está pagando bolsa para eles. Um exemplo é o pessoal da AmLight. Eles estão contribuindo muito mais com especificação, com documento. A gente não cobra. Eles mandam para gente e a hora que chega, a gente [tem documento novo para ler, tem especificação nova para ler [...] vamos contribuir, vamos revisar] (E10).

Mesmo entre os desenvolvedores bolsistas nota-se também que existe certa autonomia para que eles possam fazer contribuições extras ao projeto, além das obrigações rotineiras que possuem por serem formalmente contratados.

Eu tinha autonomia, não existia uma regra que me impedisse de criar e desenvolver coisas para dentro do projeto. Cada um pode fazer o que quer, tanto dentro quanto fora, o pessoal tem liberdade e autonomia para participar, para mandar código (E12).

Finalmente, outra forma de controle que se revelou importante para a garantia da qualidade do projeto Kytos se refere ao controle entre pares. Embora exista um monitoramento, feito pelo coordenador, de tudo o que entra no repositório do projeto, sempre houve um forte incentivo para que os pares revisem o código entre si, mesmo que eles sejam *committers* e tenham poder de escrita no repositório do *software*. O entrevistado E10 relatou como ocorre esse incentivo para a revisão das contribuições externas e também entre pares:

Todos os desenvolvedores, mesmo os antigos, mesmo os que têm acesso ao repositório, eles fazem *pull requests*, que são os pedidos de alterações na nossa árvore principal. Isso passa não só por mim, mas por outras pessoas também. Eles mesmos revisam os *pull requests* dos outros desenvolvedores. Não é porque eu estou como coordenador, que apenas eu tenho que revisar. Eu reviso, mas eles também revisam entre si, apenas para ter mais olhos para o código e coisa desse tipo (E10).

Contudo, nota-se que E10, por ser o líder/coordenador responsável formal pelo desenvolvimento, bem como o indivíduo que teve a ideia inicial do Kytos e das suas principais diretrizes, costuma ele próprio revisar especialmente as contribuições que envolvam o *core* do projeto, como forma de garantir o alinhamento delas aos objetivos traçados. Esse acompanhamento converge com a questão da centralização das decisões mais estratégicas do projeto, sendo essa a forma encontrada para garantir a qualidade do código e o direcionamento dos esforços aos rumos estabelecidos pela gestão do Kytos.

Se é uma contribuição de documentação, por exemplo, qualquer pessoa pode revisar e aceitar. Se é uma contribuição de *layout*, de *design*, da parte gráfica, qualquer um pode aceitar. Se é uma contribuição de um componente de uma aplicação nossa, geralmente é aquela pessoa que está mais envolvida naquela aplicação que revisa. E se é uma coisa muito *core*, muito do núcleo, que envolve muitas peças, aí eu reviso. A não ser que eu esteja muito confortável que aquele código não vai quebrar nada, eu acabo aceitando diretamente, mas geralmente eu peço para algum outro membro que trabalhou naquele componente revisar também junto comigo (E10).

#### 3.4.4 Análise conjunta dos casos

Descritos os casos separadamente, nesta seção a intenção é fazer uma análise conjunta da governança, considerando a estrutura, os processos de controle e as transformações identificadas no desenvolvimento dos projetos. A ideia deste esforço é efetuar

um contraponto entre conceitos, abordagens e teorias trazidas pela tese, com a finalidade de identificar semelhanças ou particularidades entre as unidades de análise estudadas.

#### 3.4.4.1 Estrutura

No que diz respeito à infraestrutura física e tecnológica, percebe-se que, independente de o projeto ter surgido por meio de iniciativa individual (caso do Kytos), ou organizacional (CoGrOO e Noosfero), em algum momento todos esses projetos passaram a funcionar vinculados a um contexto institucional. Nesse aspecto, a realidade se mostrou convergente com o que a literatura relata sobre os projetos comunitários, que podem variar em função das suas origens, naturezas e locais de funcionamento (O'Mahony & Ferraro, 2007; O'Mahony & Lakhani, 2011; Li-Ying & Salomo, 2013). O CoGrOO resultou de um projeto acadêmico e durante vários anos esteve funcionando no contexto da USP. Posteriormente, teve continuidade na comunidade global OpenNLP, atualmente integrante da fundação Apache. O Noosfero, desde o início, se caracterizou como empreendimento mantido pelos contratos de negócio da cooperativa Colivre. O Kytos, a seu turno, apesar de ter sido uma iniciativa individual, logo passou a ser desenvolvido no âmbito da Unesp. A existência dessas organizações em torno dos projetos permite notar que o apoio institucional é fator fundamental para a disponibilidade de infraestrutura física e tecnológica básica visando ao desenvolvimento do *software*.

Além de receberem apoio institucional, os projetos, segundo as entrevistas, evidenciaram fazer uso intenso dos recursos de TIC nas atividades produtivas. Por serem projetos de *software* livre comunitários, os empreendimentos disponibilizaram o código-fonte do *software* na Internet com a intenção de formar uma comunidade de usuários e desenvolvedores dispostos a realizar contribuições para aperfeiçoamento desse *software* (Shah, 2006; Seidel & Stewart, 2011). Essa estrutura fluida e aberta para a adesão de colaboradores se caracteriza pela virtualidade das operações e pela dispersão geográfica dos indivíduos do coletivo, o que torna o uso dos recursos de TIC essencial para interconectar e realizar os trabalhos colaborativos que transcendem o tempo e o espaço (Lee & Cole, 2003; Faraj et al., 2011; Seidel & Stewart, 2011; Bauer et al., 2016).

Entre as TICs mais citadas e comuns nos casos analisados, foram citadas: plataforma de hospedagem do código-fonte, usada também para o controle de versões do *software*, como *GitHub*, *GitLab* e *Sourceforge*; plataforma de *chat* e vídeo, em especial, o uso do *Google Hangouts*; ferramentas de comunicação e colaboração em equipe, a exemplo do *Slack*; listas

de *e-mail*; e *home page*, usadas para consolidar e divulgar as informações dos projetos. Tais tecnologias coincidem com o conjunto de ferramentas mencionado na literatura de comunidades virtuais (Jensen & Scacchi, 2010; Paily, 2013; Lindberg et al., 2016). Entretanto, é importante sublinhar que as listas de *e-mail* demonstraram ser o meio de comunicação mais relevante em todos os projetos, não só por viabilizar uma comunicação não assíncrona, mas também por permitir o registro das interações entre os membros dos grupos.

Em relação à hierarquia e aos papéis dos integrantes dos projetos, no geral as informações apuradas indicaram que ao longo de suas trajetórias de desenvolvimento os coletivos tiveram que lidar tanto com aspectos formais, quanto informais.

No que tange à hierarquia, os projetos apresentaram a formalização de indivíduos designados para exercer a gestão do coletivo, adotando em paralelo a figura de uma autoridade que transcende o esquema formal estabelecido. Nesse aspecto, é possível notar a coincidência com o que autores como Dahlander e O'Mahony (2011) relataram sobre os projetos de *software* livre, apontando-os como agrupamentos que admitem uma autoridade lateral, alicerçada na experiência e em função das contribuições técnicas dadas pelo indivíduo ao projeto. Os projetos, então, se caracterizaram por exibir duas realidades, isto é, a existência de uma gestão hierárquica formalizada em convivência com relações horizontais de autoridade internas.

Alinhados com esse modelo de hierarquia formal e informal, encontram-se os papéis desempenhados nos trabalhos colaborativos dos projetos. No rol dos papéis formais associados à gestão dos coletivos, merecem destaque: no CoGrOO, durante vários momentos, a responsabilidade formal de gerenciar o desenvolvimento do *software* esteve concentrada nos docentes; no Kytos, o acompanhamento pelo docente diretor do NCC, assim como pelo coordenador ou líder do projeto, funcionário da Unesp; no Noosfero e no OpenNLP, um papel comum a esses dois projetos foi o do *release manager* (RM). No entanto, convém referir, enquanto no Noosfero o RM se caracterizou por ser um papel instituído não só para fazer o lançamento de versões, mas também para realizar a gestão do projeto, no OpenNLP, tal função se distinguiu por focalizar especificamente o processo de lançamento de versões. Os papéis formais de gestão próprios do OpenNLP foram os de *chair* ou vice-presidente, bem como os dos membros do comitê gestor do projeto (PMC). Principalmente no Noosfero e no OpenNLP, as responsabilidades atreladas a cada um dos papéis formalmente estabelecidos, foram explicitamente especificadas em documentos próprios, denotando a preocupação em se ter normas escritas.

Na área de desenvolvimento do *software*, foi notada a presença de equipes destacadas por exercer funções formais ou informais. Dentre os encarregados da execução de papéis formais estiveram os profissionais e bolsistas contratados pelos projetos, notadamente no CoGrOO e no Kytos. No primeiro, a contratação aconteceu desde quando o projeto recebeu recursos da Finep. No Kytos, a contratação é uma realidade, desde que o projeto recebeu o patrocínio da Huawei, prática que permanece até os dias atuais. Os papéis informais observados estiveram concentrados nos desenvolvedores voluntários, engajados nas atividades colaborativas dos projetos. O caso do Noosfero, no entanto, merece uma explicação à parte, pois as equipes de desenvolvedores ao mesmo tempo em que eram formadas por voluntários que contribuía com a comunidade do projeto, também eram constituídas por funcionários formais que utilizavam o *software* nos projetos executados no contexto da organização em que estavam envolvidos. Por iniciativa destes funcionários, os códigos desenvolvidos nas suas esferas profissionais e respectivas organizações retornaram à comunidade do projeto por meio de contribuições voluntárias, ensejando um ganho para o coletivo, do ponto de vista técnico.

Ainda em relação à equipe de desenvolvimento, um papel comum a todos os casos, é o de *committer*, desenvolvedor, voluntário ou não, que tem poder de escrita de código no repositório principal do projeto. Em alguns dos casos, os *committers* apareceram como grupos formalmente constituídos, como foi possível notar no Noosfero e no OpenNLP; em outros casos, o papel demonstrou acontecer informalmente, como no CoGrOO e no Kytos. No CoGrOO, o compartilhamento de senhas para escrita no repositório era algo que naturalmente acontecia conforme os membros vinculados à USP começavam a trabalhar efetivamente no desenvolvimento. No Kytos, por sua vez, apesar de existirem definições prévias sobre o exercício dos papéis, o ato de se tornar *committer*, especificamente, apareceu atrelado à percepção do coordenador do projeto, que só após considerar que determinado indivíduo é merecedor, por ter trazido contribuições relevantes, realiza o convite para a execução desta função.

No geral, é procedente afirmar que em todos os projetos analisados, os papéis formais e informais convivem durante todo o desenvolvimento do *software*. Embora as pesquisas sobre comunidades de *software* livre destaquem que esses projetos contam com a significativa colaboração de voluntários (Shah, 2006; Seidel & Stewart, 2011), os casos averiguados comprovaram que nem sempre é isso o que acontece. Principalmente no âmbito de projetos muito específicos, como no caso do Kytos, atuante no desenvolvimento de redes acadêmicas com a tecnologia SDN, ou no caso do CoGrOO, que teve uma ampla comunidade

de usuários, mas uma visível restrição de desenvolvedores, a contratação de profissionais foi a solução encontrada para manter o projeto operando em determinados momentos. Essas contratações em conjunto com as demais formalizações de alguns papéis existentes nos casos, demonstrou ser algo que contradiz as percepções normalmente destacadas na literatura, de que os projetos de *software* livre comunitários se apresentam com uma estrutura de papéis pouco definida (Lee & Cole, 2003; Arazy et al., 2016). Nesse aspecto, em diferentes momentos da pesquisa de campo, verificaram-se nos projetos investigados papéis formalizados para garantir o desempenho de responsabilidades vistas como essenciais para a produção do *software*.

Não obstante algumas divergências em relação à literatura sobre *software* livre tenham sido verificadas, certas aproximações também foram percebidas. Por exemplo, independente dos papéis serem formais ou não, em todos os projetos os entrevistados relataram existir flexibilidade no exercício dos papéis dentro dos coletivos. Em função das necessidades dos projetos, forças-tarefas chegaram a ser compostas, de forma a permitir que os indivíduos pudessem exercer outras funções além das que lhes foram formalmente atribuídas. Isso comprovou que os indivíduos não ficaram limitados a uma única função durante toda a sua atuação no projeto. Essa constatação coincide com o que preconiza Lee e Cole (2003), Zammuto et al. (2007) e Arazy et al. (2016), ao relatarem que nos espaços comunitários os indivíduos podem executar múltiplos papéis, havendo sobreposição de funções, de acordo com as necessidades do momento.

Outra questão comum em todos os projetos se relacionou à função da liderança. O papel de líder se mostrou preponderante na definição dos rumos a serem seguidos pelos participantes dos projetos. No Kytos, a liderança se concentrou, fundamentalmente, no coordenador, que foi também o criador do projeto. Nos demais casos, diferentemente, a liderança surgiu de forma alternada, de acordo com o contexto vivenciado pelo projeto. No CoGrOO, a liderança esteve a cargo dos docentes e do entrevistado E1, que se tornou o principal desenvolvedor do *software*, após o afastamento dos demais integrantes. No Noosfero, como já destacado, o RM foi designado como o integrante que seria gestor do projeto, tornando-se o principal líder do coletivo. No OpenNLP, apesar de o *chair* ser uma espécie de líder do projeto, na verdade se destacou para uma atuação maior em atividades burocráticas, o que fez com que outros membros do grupo com o perfil de liderança terminassem assumindo essa função.

Com ou sem a liderança exercida por um membro formalmente designado, ou distribuída informalmente entre outros integrantes do projeto, essa função em todos os casos se mostrou indispensável para direcionar as atividades em torno do *software* na busca de

resultados. Tal constatação converge com a visão de que a liderança é decisiva na governança, ao cumprir a missão de promover a unidade e o alinhamento de comportamentos e objetivos no âmbito dos coletivos de *software* livre (Xu et al., 2005; Seidel & Stewart, 2011).

Na esfera da tomada de decisões, ainda que todos os casos tenham evidenciado abertura para que os participantes pudessem manifestar suas opiniões, na inexistência do consenso, ou quando as decisões tomadas envolveram questões estratégicas sobre os rumos do projeto, a centralização das decisões foi visível. No CoGrOO, essa centralização aconteceu principalmente em função dos docentes que estiveram à frente do projeto em momentos específicos, bem como sob a responsabilidade do desenvolvedor E1, que em várias fases do projeto, como já dito, se viu como o responsável por assumir este encargo. No OpenNLP, foi notória a adoção de um processo de consenso formalizado, conhecido como reunião de consenso. Nele, por meio das listas de *e-mail*, os membros do PMC votam sobre determinado assunto, prevalecendo a solução dada pela contagem final dos votos. Apesar dessa busca do consenso, os representantes com poder de voto nas questões relevantes do projeto eram de fato aqueles pertencentes ao comitê de gestão do projeto (membros do PMC). No caso Noosfero, na ocorrência de algum impasse na tomada de decisão, o indivíduo formalmente incumbido para realização da escolha final é o RM. O Kytos também tem na sua estruturação decisória, as escolhas mais estratégicas realizadas pelo diretor do NCC e coordenador do projeto, ainda que exista a liberdade dos integrantes manifestarem suas opiniões sempre que julgarem necessário.

O Noosfero e o OpenNLP, neste cenário, mostraram ser projetos que possuem regulamentações formais sobre a forma de como as decisões devem ser tomadas. Nos demais projetos, as decisões ocorreram de acordo com o momento vivenciado pelo agrupamento, sem um processo previamente definido. Majoritariamente, todavia, a centralização das decisões de maior impacto sobre o projeto foi algo confiado a um ou mais indivíduos do coletivo, seja em razão de um papel formalmente delineado para tanto, seja em decorrência da influência de colaboradores mais experientes dentro do projeto. Essa percepção da pesquisa empírica converge com o paradoxo existente nos estudos sobre iniciativas de *software* livre, pois, ao mesmo tempo em que os empreendimentos vão se tornando descentralizados, no que se refere às decisões de maior impacto nos rumos do projeto, a centralização é inevitável, recaindo sobre os membros formalmente designados ou mais experientes (Lee & Cole, 2003; West & O'Mahony, 2005; Seidel & Stewart, 2011).

#### 3.4.4.2 Processos de controle

Os processos de controle formal, direcionados para os comportamentos dos integrantes e resultados dos coletivos, foram observados em alguns fluxos de atividades. Nesse aspecto, destacaram-se: a escolha de representantes e definições de responsabilidades para exercício de determinado papel; as tomadas de decisão, em especial aquelas estratégicas referentes às escolhas estratégicas e de definição dos rumos do projeto; as regulamentações das contribuições recebidas no projeto, principalmente as referentes a mudanças no código-fonte do *software*; as rotinas necessárias para o lançamento de versões; a elaboração de relatórios para prestação de contas ou acompanhamento dos projetos. Os casos analisados que se destacaram por apresentar processos bem definidos e delimitados formalmente, foram o OpenNLP, o Kytos e o Noosfero. Alguns exemplos captados nas entrevistas e pesquisa documental são descritos a seguir.

O OpenNLP, assim como os demais projetos da Apache, se caracterizou por seguir uma série de procedimentos discriminados pela fundação em seu *site*, envolvendo não só a estruturação e responsabilidades dos papéis, mas também o estabelecimento de alguns processos. Ademais, a Apache tem um código de conduta elaborado para direcionar o comportamento dos participantes, bem como exige que os indivíduos que se tornem *committers* façam a assinatura de um documento confirmando que estão de acordo com os termos de licença do projeto. Nesse último ponto, o Kytos, projeto que recebe patrocínio da Huawei, com funcionamento no âmbito da Unesp, se aproxima do projeto OpenNLP. Os indivíduos dispostos a colaborar com o Kytos precisam estar cientes de que passarão a ser regidos pelos termos de licença do *software* expostos na página do projeto. O Kytos também evidenciou a formalização de certos papéis, além do diretor do NCC e do coordenador, a exemplo da contratação de profissionais com os recursos do patrocínio da Huawei. Tanto o Kytos, como o OpenNLP obedecem à regra de entregar relatórios de acompanhamento dos seus projetos para a empresa patrocinadora e para a fundação Apache, respectivamente.

O Noosfero, a seu turno, patrocinado e mantido pela Colivre, em virtude dos debates que ocorreram entre os participantes para que o projeto se tornasse mais aberto ao recebimento das contribuições externas, levou a cooperativa a elaborar um documento com uma série de formalizações. Este documento, disponibilizado no meio virtual utilizado pelo projeto, expõe as orientações não só sobre o exercício dos papéis, mas também a propósito das formas pelas quais as contribuições em volta do *software* devem ocorrer.



O CoGrOO, diferente dos demais projetos, embora não tenha apresentado processos bem definidos durante na sua trajetória, algo que só ocorreu quando houve a interface com o OpenNLP, teve momentos em que ações formalizadas se tornaram necessárias. Esses períodos ocorreram, especialmente, quando o desenvolvimento do corretor recebeu o financiamento da Finep, bem como quando o *software* foi utilizado em disciplinas ministradas e pesquisas realizadas no âmbito da USP. Em tais fases, as diversas forças-tarefas existentes para elaboração e aperfeiçoamento do projeto tinham compromissos e cronogramas de entregas delimitados, basicamente executados pelos bolsistas ou estudantes envolvidos, sob a supervisão dos docentes da universidade.

A partir das situações retratadas, o que se nota quando os projetos são considerados ao longo das suas respectivas trajetórias de desenvolvimento, é que formalizações em torno de comportamentos e resultados ficam evidenciados em diversas situações. Assim, apesar de as pesquisas sobre *software* livre frisarem que essas iniciativas se sobressaem pela informalidade e pouca estruturação dos processos (Lattemann & Stieglitz, 2005; Xu et al., 2005; Lindberg et al., 2016; Kolbjørnsrud, 2016), os casos averiguados em campo não se mostraram em sintonia com essas afirmações, uma vez que, dependendo do momento vivenciado pelos projetos, processos formais passaram a ter lugar nas atividades dos coletivos.

As formalizações em torno dos processos de comportamentos e resultados aconteceram principalmente nas ocasiões em que os projetos estabeleceram relações com outras organizações formais. Sendo assim, os casos como o CoGrOO na interface com OpenNLP, o Noosfero e o Kytos foram os mais impactados na definição de processos formalizados, por serem projetos que integram fundação de *software* livre, ou que são patrocinados, o que termina exigindo uma definição mais concreta das condutas e entregas esperadas dos seus colaboradores, ainda que eles sejam voluntários. Mesmo no caso do CoGrOO, em que os relacionamentos com atores externos aconteceram em períodos específicos, a formalização se fez presente.

Essa percepção de que as comunidades de *software* livre podem se organizar de modo híbrido, com processos formais e informais, foram consideradas nas pesquisas de O'Mahony e Ferraro (2007) e Panchal (2010). No entanto, é comum que as pesquisas sobre o tema ainda costumem associar as iniciativas comunitárias de *software* livre a contextos marcados pela informalidade da estrutura e do controle. A diferença substantiva identificada nesta tese, no entanto, é que não foi só o crescimento dos projetos que levaram à necessidade de definição dos processos, mas, sim, os relacionamentos com outras organizações formais.

O controle informal ou social, baseado na socialização e interação intensiva entre integrantes de um grupo, foi a outra modalidade de controle identificada nos projetos. A socialização intensiva, vale ressaltar, tem repercussão direta nos outros modos de controle, representados pelo autocontrole e pelo controle entre pares.

Na promoção da socialização dos integrantes dos projetos merecem destaque os recursos de TIC, sobretudo por conta da dispersão geográfica dos participantes. O uso desses recursos tecnológicos possibilitou a criação de um espaço virtual comum, permitindo as trocas de informações e conhecimentos entre os atores com vistas a realizar os trabalhos colaborativos comunitários de desenvolvimento do *software* (Seidel & Stewart, 2011; Zheng et al., 2013). Essa comunicação permite que membros heterogêneos, e com diversas motivações, direcionem os seus esforços para a concretização dos resultados almejados nos projetos. Nesse sentido, a tecnologia demonstrou ser, em todos os casos analisados, uma ferramenta que também auxilia na consolidação das práticas de controle informal observadas nos projetos, em especial entre os integrantes voluntários.

Apesar de a maior parte das comunicações acontecer por meios virtuais, encontros presenciais para debater questões sobre o projeto também foram relatados na pesquisa de campo. O Noosfero promoveu encontros presenciais, especialmente no começo da execução do projeto, quando ele estava centralizado na Colivre e com a maioria dos integrantes localizada em Salvador. No CoGrOO, os desenvolvedores que estudavam na USP destacaram que tiveram diversos momentos de interação presencial durante o período em que a referência no desenvolvimento do *software* esteve na universidade. O Kytos, uma vez que faz uso da infraestrutura da Unesp, desde sempre oportuniza que alguns dos seus integrantes consigam se encontrar presencialmente, com mais frequência alcançando aqueles que são contratados formalmente para o projeto. No OpenNLP, cuja interface com o CoGrOO já se deu a conhecer, as oportunidades de encontros presenciais são raras e acontecem, sobretudo, no evento ApacheCon, que é uma conferência anual promovida pela fundação.

Além dos encontros, presenciais ou virtuais, direcionados para trocas de experiências sobre o projeto em si, os participantes também destacaram eventos nos quais tiveram a oportunidade de interagir com atores externos aos projetos. Nestes processos de socialização mais amplos, destacam-se as participações dos integrantes em eventos técnicos ou acadêmicos da área de *software* livre ou da área de conhecimento a que o projeto pertence. No Nosfero foi mencionada a oportunidade de os integrantes ministrarem palestras em encontros sobre *software* livre. No Kytos, CoGrOO e OpenNLP, por sua vez, foram enfatizados eventos nos quais trabalhos técnicos ou acadêmicos sobre os projetos foram apresentados. O Kytos, em

particular, chegou a promover algumas edições do evento chamado *Kyotos Summit*, nos quais palestras técnicas e sociais foram realizadas com a intenção de promover o encontro presencial dos integrantes do coletivo ou de interessados em conhecer o trabalho do projeto.

Adicionalmente às atividades direcionadas para a interação dos indivíduos, outra forma de controle social nitidamente percebida esteve associada ao reconhecimento do trabalho executado nos projetos, prática associada à meritocracia. Nesse aspecto, os membros experientes e que trouxeram contribuições relevantes, além de serem reconhecidos entre os seus pares, se destacaram por exercer influência significativa nas decisões mais importantes do projeto. Conforme preconizam Lattemann e Stieglitz, (2005) e O'Mahony e Ferraro (2007), espaços em que a meritocracia é percebida aumentam as oportunidades de participação e permitem a transparência das ações, o que reduz a necessidade de serem estabelecidas políticas desnecessárias nos agrupamentos comunitários.

Além deste tipo de visibilidade interna, os entrevistados relataram, também, as oportunidades externas de reconhecimento proporcionadas pelos seus respectivos projetos. Situações como reportagens, entrevistas, premiações e até momentos em que recursos relevantes conseguiram ser captados, foram citados como momentos relevantes perante o público externo, no reconhecimento do projeto e dos seus principais representantes. Essa visão do reconhecimento como uma forma de controle social está alinhada com o que se espera de ambientes comunitários de *software* livre, já que é uma forma de motivação que se diferencia por não fazer uso de recompensas formais e de benefícios privados (Shaikh & Vaast, 2016).

Todas essas vivências, por meio das interações internas ou externas que ocorreram nos casos investigados foram essenciais para que os indivíduos criassem um senso de identificação e de pertencimento com os projetos (Xu Xu et al., 2005; Lattemann & Stieglitz, 2005), visto até mesmo entre aqueles que atualmente estão afastados ou menos ativos. Ter essa referência e identidade em relação aos projetos confirmou ser algo que influencia no autocontrole manifestado pelos indivíduos, principalmente entre aqueles que realizaram ou que ainda realizam contribuições como voluntários nos empreendimentos. Os próprios recursos tecnológicos utilizados, por permitirem autonomia e independência na execução do trabalho, também fornecem suporte para o autocontrole, reduzindo os esforços de coordenação no coletivo (Crowston et al., 2012; Howison & Crowston 2014; Lindberg et al., 2016). Nessa perspectiva, apesar de prevalecer nos casos os trabalhos realizados de modo descentralizados, o autocontrole figurou como procedimento fundamental não só para guiar

esforços, como também para orientá-los em direção aos resultados comuns esperados no desenvolvimento do *software*.

No tocante à realização de revisão das contribuições recebidas em torno do *software*, tal como já referido anteriormente, os projetos manifestaram possuir regras formalizadas, sobretudo, para que as colaborações externas submetidas por meios virtuais passem, previamente, pela análise de algum *committer* do projeto. No entanto, quando as contribuições, em especial aquelas que incidem no código, são realizadas pelos próprios *committers*, a revisão da contribuição termina acontecendo de modo informal, pelo processo conhecido como controle entre pares. Nesses casos, ainda que os *committers* tenham poder de escrita no repositório, uma prática que revelou acontecer com frequência foi a da revisão do código por algum outro *committer* do coletivo, principalmente pelos desenvolvedores mais experientes do projeto. O Noosfero foi o único projeto que tentou instituir regra explícita nesse sentido, mesmo assim com certa flexibilidade. Nela, se após uma semana o código não for revisado por nenhum par do coletivo, o próprio *committer* que propôs o código tem autonomia para inseri-lo no repositório.

Independentemente de acontecer de modo formalizado, ou não, a revisão das contribuições dadas em relação ao código, em todos os casos, se revelou uma prática extremamente incentivada entre os integrantes do projeto em qualquer período vivenciado pelo coletivo. Neste quesito, o do controle entre pares, os projetos investigados convergiram com as pesquisas de *software* livre, no sentido de reconhecer que o criticismo ao trabalho alheio e as trocas de experiências possibilitadas pela revisão auxiliam para assegurar a qualidade do *software* produzido (Lee & Cole, 2003; Kolbjørnsrud, 2016).

Em termos de controle informal, portanto, percebe-se que há um alinhamento entre o que foi verificado nos casos examinados empiricamente e os estudos sobre os projetos de *software* livre. Os processos de controle informais, em conjunto com o intenso uso dos recursos de TIC, foram essenciais para permitir a identificação dos indivíduos com os projetos em que atuam, de modo a direcionar suas ações para o atingimento dos objetivos do coletivo (Xu Xu et al., 2005; Lattemann & Stieglitz, 2005). Todos os meios usados na socialização intensiva dos membros se solidificaram em um conjunto de práticas compartilhadas, que é único em todos os agrupamentos (Wenger, 1998; Seidel & O'Mahony, 2014). Essas práticas constituem reflexos das histórias particulares vivenciadas, o que é perceptível em todos os casos investigados, já que cada projeto encontrou um caminho específico que viabilizasse o desenvolvimento do *software*.

### 3.4.4.3 Transformações decorrentes do desenvolvimento dos projetos

A análise da história e da governança dos casos investigados, assim como das transformações pelas quais os projetos de *software* livre comunitários passaram, permite que algumas questões centrais sejam pontuadas. Apesar de os casos estudados nesta tese terem se pautado pela escolha de empreendimentos nacionais e com diferentes características de desenvolvimento, todos os projetos de *software* livre deveriam ter em comum o fato de serem iniciativas organizadas no formato de comunidades. Assim, seguindo o que a literatura preconiza como arranjos comunitários, os projetos colaborativos estudados foram os que transcenderam as fronteiras de determinada organização, por meio da disponibilização do código-fonte na Internet, com abertura suficiente para que desenvolvedores ou usuários pudessem acessar ou contribuir com esses empreendimentos (Lee & Cole, 2003; Shah, 2006; Jansen, 2010).

Em relação à governança, especificamente, compreendida nesta tese a partir dos conceitos de estrutura e controle, diferenças em relação ao que os estudos da temática de *software* livre costumam relatar foram constatadas na pesquisa empírica. Nesses estudos, comumente, enfatiza-se que a produção comunitária de *software* livre se diferencia visivelmente das organizações tradicionais de *software* (Seidel & Stewart, 2011). Em outras palavras, ressalta-se que enquanto as organizações tradicionais se amparam em uma estrutura hierárquica e em procedimentos formais, as comunidades de *software* livre operam dentro de uma lógica notadamente informal em termos de estruturação e dos processos de controle empregados (Lattemann & Stieglitz, 2005; Xu et al., 2005; Seidel & Stewart, 2011). Autores como Raymond (1999), inclusive, chegaram a afirmar que enquanto as organizações tradicionais se assemelham a uma catedral, com hierarquia e processos formais, as iniciativas de *software* livre funcionam numa lógica próxima a de um *öbazarö*, isto é, operam de modo horizontal e informal.

A análise dos dados desta tese, obtidos mediante entrevistas e documentos, revelou ser imprescindível no estudo dos projetos de *software* livre comunitários considerar toda a história e o contexto no qual os projetos estiveram inseridos, antes de serem realizadas afirmações gerais, como as trazidas anteriormente. Na prática, o campo dos projetos de *software* livre comunitários, quando visto de forma mais ampla, mostrou-se muito mais denso do que o normalmente propagado nos estudos sobre o tema. Em cada um dos casos analisados, verificou-se que os projetos passaram por uma série de modificações no decorrer

da sua trajetória de desenvolvimento, mudanças que também afetaram a governança implementada.

O CoGrOO se destacou por ser um projeto de *software* livre que durante vários anos teve o seu funcionamento vinculado à infraestrutura da USP. Durante a duração deste vínculo, alguns fatos foram cruciais para o desenvolvimento do *software* se manter ativo, tais como: o recebimento de recursos financeiros da Finep; o uso do *software* em disciplinas e pesquisas acadêmicas; o recebimento de recursos financeiros do programa Google *Summer of Code*; a integração do projeto ao CCSL. Nas fases vivenciadas pelo CoGrOO, o entrevistado E1, contratado inicialmente como bolsista, e depois atuando como voluntário, se destacou por ser o principal desenvolvedor do corretor gramatical. Após a fase positiva da USP, ao perceber que a comunidade do *software* estava estagnada, E1 vislumbrou em outro projeto do qual participava a possibilidade de o código do CoGrOO continuar recebendo a devida manutenção. Sendo assim, outro momento de grande relevância ocorreu quando esse desenvolvedor realizou a transferência de grande parte do código do CoGrOO ao projeto global OpenNLP, atualmente incorporado à Fundação Apache.

O Noosfero se caracterizou por ser um projeto originário de contratos comerciais desenvolvidos pela cooperativa Colivre. Em vista disso, no decorrer dos anos, a cooperativa atuou como a principal patrocinadora do projeto, o que acontece até os dias de hoje. No entanto, ao longo dos anos, outros atores externos alinhados com essa perspectiva de *software* livre também começaram a utilizar o Noosfero para criar as suas redes sociais. Sobressaem entre esses atores, organizações como a USP, o Serpro, o Governo Federal e a UnB do Gama. Os projetos paralelos executados no âmbito e com os recursos dessas organizações formais, criaram em torno do Noosfero uma nova dinâmica de desenvolvimento do *software*. Sob influência dessa dinâmica, os desenvolvedores envolvidos nesses projetos paralelos demonstraram ter a preocupação de retornar o aperfeiçoamento do código à comunidade central do Noosfero, por meio de contribuições voluntárias. Apesar de essa lógica de trabalho ter sido crucial para o desenvolvimento do *software*, e continuar a ocorrer até hoje, o Noosfero ainda é um projeto bastante dependente da infraestrutura fornecida pelo patrocínio da Colivre.

O Kytos é o projeto de *software* livre mais recente em relação aos demais. Apesar de ter sido originado de uma iniciativa individual, logo no início, o projeto passou a contar com a infraestrutura da Unesp. Um momento basilar para que o desenvolvimento do Kytos passasse a ser realizado de forma efetiva ocorreu quando o projeto recebeu o patrocínio da empresa chinesa Huawei. Esse patrocínio, vale salientar, proporcionou que recursos financeiros fossem

angariados ao projeto por intermédio da Lei da Informática, de responsabilidade do MCTI. Além de se manter até os dias atuais com os recursos advindos desse patrocínio e o suporte da Unesp, o Kytos também conta com a participação de voluntários oriundos especialmente dos contatos e parcerias que o NCC mantém com outras instituições.

Passadas em revista as principais transformações que ocorreram em cada um dos casos, é percebido que mesmo os projetos tendo seguido por diferentes caminhos, todos eles tiveram em comum as principais fases de desenvolvimento do *software* impulsionadas pelos relacionamentos estabelecidos com outras instituições. Nesse percurso, é inerente reconhecer que a governança se mostrou um conceito fluido, modificando-se em função de momentos de transformações vivenciados pelos projetos. Essa visão adaptativa da governança é algo, inclusive, que converge com os estudos da área de *software* livre, os quais destacam que o conceito se encontra sempre em constante ajustamento às transições sucedidas no ambiente em que os projetos estão inseridos (Lattermann & Stieglitz, 2005; Xu et al., 2005; Jensen & Scacchi, 2010; Panchal, 2010; Faraj et al., 2011).

Por ter esse caráter dinâmico, os projetos de *software* livre comunitários devem ser estudados de modo contextualizado, em vez de ser fornecido um retrato estático da comunidade existente. Com isso, é possível inferir que o cenário desses projetos é muito mais complexo do que aquele que normalmente a literatura tem conseguido delimitar e explicar. Em especial nas questões referentes às diferenças de governança entre projetos de *software* livre e organizações tradicionais, o emprego dos procedimentos formais e informais identificados na presente investigação, se configurou de modo muito mais intrincado do que a dicotomia da *catedral versus* *obazarö* que as pesquisas geralmente apontam.

Nos estudos que reconheceram a possibilidade de os projetos de *software* livre formar arranjos híbridos, envolvendo meios formais e informais de governança (Lattermann & Stieglitz, 2005; Xu et al., 2005; Panchal, 2010), as atenções estiveram direcionadas para aspectos teóricos e para as mudanças ocorridas ao longo dos estágios de desenvolvimento desses empreendimentos. Cada um desses estágios, vale sublinhar, materializando-se em função do crescimento alcançado pelo projeto. Na prática, no entanto, os casos averiguados não demonstraram ter etapas bem definidas de desenvolvimento. Os achados alcançados permitiram verificar que momentos de maior ou menor atividade surgiram de maneira intercalada em cada projeto, sendo determinados principalmente pela acessibilidade a recursos (materiais, financeiros e humanos) que garantissem as operações do projeto. No curso dessas observações, como já relatado, os procedimentos formais aplicados nos projetos, apareceram de modo bastante associado aos relacionamentos estabelecidos com organizações formais, por

meio de parcerias, patrocínio, órgãos de fomento, ou da incorporação à fundação de *software* livre.

#### 3.4.4.3.1 Associação dos casos ao aporte teórico e conceitual da tese

Os projetos de *software* livre, objetos deste estudo, além de terem em comum o fato de funcionarem no âmbito de uma instituição fornecedora de suporte de infraestrutura física e tecnológica, também apresentaram, no decorrer da sua história, relacionamentos com diferentes organizações. Entendendo que os *stakeholders* são grupos ou pessoas que afetam ou podem ser afetados por determinada organização (Freeman, 1984), pode-se considerar que os atores externos com os quais os projetos mantiveram contato constituem os seus correspondentes *stakeholders*. Seguindo essa lógica, como principais *stakeholders* do CoGrOO citam-se a USP, a Finep, o Google e o projeto OpenNLP da fundação Apache. O Noosfero, além da Colivre, também teve no seu entorno vários *stakeholders*, tais como a USP, o Serpro, a UnB, o Governo Federal, dentre outros. Em volta do Kytos, como *stakeholders*, despontaram a Unesp, a Huawei, além de instituições e comunidades parceiras, operando em conjunto.

Ter essa compreensão de que os agentes com os quais os projetos de *software* livre interagem constituem *stakeholders* permite observar que existe uma influência mútua entre os projetos estudados e os seus grupos de interesse (Freeman, 1984; Hill & Jones, 1998; Frooman, 1999). A lógica de influência recíproca entre projeto e *stakeholders* poder ser verificada no Noosfero, quando começaram a surgir organizações paralelas que fizeram uso de procedimentos formais e de recursos para desenvolverem suas respectivas redes sociais usando o *software* deste empreendimento. Como contrapartida, alguns representantes simpatizantes da causa de *software* livre que participaram desse processo de desenvolvimento nas referidas organizações, demonstraram, igualmente, a preocupação de contribuir de forma voluntária com o Noosfero, ao retornar o código desenvolvido para a comunidade de origem do *software*.

Ainda nesse assunto, outro exemplo é a própria relação da Colivre com o Noosfero, pois, ao mesmo tempo em que a cooperativa patrocinava e dava suporte financeiro ao projeto, também sobrevivia dos contratos comerciais que envolviam o *software*, algo que acontece até hoje. A própria existência de conflitos, quando os *stakeholders* solicitaram maior independência do projeto em relação à Colivre, para que pudessem contribuir de forma aberta com o Noosfero, é um exemplo dessa influência mútua entre ambos (Frooman, 1999;



Friedman & Miles, 2006). Como consequência disso, a Colivre precisou definir processos formais para o funcionamento da comunidade, de forma a garantir que essa abertura assegurasse a qualidade do *software*.

Visto que a interação entre os empreendimentos e os *stakeholders* foi uma realidade enfrentada por todos os casos analisados, nota-se que o contexto assume um papel crucial no sentido de potencialmente alterar os comportamentos dos projetos (Pfeffer & Salancik, 2003). É justamente nesse aspecto que a teoria da dependência de recursos (TDR) pode ser aplicada aos projetos de *software* livre comunitários. As dinâmicas percebidas ao longo da trajetória de desenvolvimento dos projetos investigados permitem notar que o principal motor que condiciona os relacionamentos estabelecidos com os *stakeholder* é a necessidade de obter recursos para a sobrevivência do negócio. Assim, em todos os casos, recursos tiveram de ser buscados no ambiente para garantir que os projetos continuassem funcionando no passar dos anos, à base de infraestrutura fornecida por alguma instituição, ou pelo apoio de fundação, patrocinador e órgãos de fomento. O Kytos, apenas para exemplificar, além do suporte de infraestrutura física e tecnológica ofertada pelo NCC da Unesp, por meio de contatos realizados pelo gestor do projeto, conseguiu o patrocínio da empresa Huawei, que viabilizou também a obtenção de recursos financeiros e tecnológicos necessários ao desenvolvimento do *software*.

Dentro da concepção trazida pela TDR (Hillman et al., 2009), percebe-se que os projetos não são apenas influenciados pelo ambiente, mas também possuem uma postura proativa diante dele. Sendo assim, os responsáveis pelos projetos possuem autonomia para fazer uso de diversas táticas gerenciais com a intenção de garantir no ambiente os recursos de que precisam para sobreviver (Pahlevan Sharif & Kyid Yeoh, 2014). Essa busca constante por recursos aconteceu em diversos momentos do CoGrOO, que, além do apoio da USP e do financiamento fornecido pela Finep, teve que buscar formas de garantir mais recursos durante a sua trajetória, o que envolveu, inclusive, o recebimento de verba do programa Google *Summer of Code*. Posteriormente, quando um de seus desenvolvedores se tornou, em dado momento, o único responsável pelas operações do projeto, encontrou no OpenNLP uma forma de o código do CoGrOO continuar recebendo a manutenção necessária. Assim, uma vez que o OpenNLP passou a integrar a Apache, indiretamente o projeto passou também a contar com o apoio dessa fundação.

Outra associação admissível sobre as iniciativas de *software* livre comunitárias refere-se à aproximação delas ao conceito de organizações temporárias (OTs). Nesse aspecto, considerando o funcionamento dos projetos comunitários em seu contexto e a partir das

transformações que ocorreram ao longo do tempo, ainda que traços de comunidade inerentemente permaneçam existindo, outras características sobressaem no desenvolvimento desses projetos. Tal como expressa a própria definição das OTs (Burke & Morley, 2016), os projetos estudados se caracterizaram por ser formados por agrupamentos de atores interdependentes que direcionam as suas ações para a realização de tarefas complexas. Percebe-se, então, que o tema das OTs se apresenta amplo o suficiente para abranger desde agrupamentos organizacionais, até outros diversos tipos de coletivos, nos quais os indivíduos cooperam e integram recursos e *expertises* para o alcance de resultados (Sydow et al., 2004).

Dentre diversos tipos de grupos que são considerados como OTs se encontram os arranjos organizacionais baseados em projetos, tal como definido por Bakker (2011) e Burke e Morley (2016). Os casos analisados, quando vistos numa visão mais ampla, também são empreendimentos que possuem semelhanças com as organizações baseadas em projetos. Seguindo essa lógica, cada versão do *software* lançada marca o término de um projeto, dando início a um novo ciclo no qual novos aperfeiçoamentos passarão a acontecer no *software* até à sua consolidação em uma próxima versão. Observa-se, portanto, porque as iniciativas de *software* livre investigadas, de modo convergente com as OTs, têm um forte direcionamento para tarefas (Bakker, 2011), que acontecem fundamentalmente em torno do lançamento das versões do *software*. Além disso, o lançamento de versões também pode ser compreendido como um ponto de terminação, algo percebido nas OTs, já que ocorre em função da fixação de uma data específica ou pela obtenção de um estado ou condição do empreendimento previamente definidos (Bakker et al., 2009; Bakker, 2011). Outra forma de perceber as iniciativas de *software* livre como OTs, ocorre por meio da própria associação dos projetos com diferentes *stakeholders*, representados especialmente pelos atores organizacionais externos. Nessas associações, ajustamentos estruturais e dos processos de controle demonstraram acontecer no contexto do projeto, demarcando os contornos de um arranjo organizacional temporário, que se modifica conforme novos relacionamentos com *stakeholders* são estabelecidos.

Outras características que aproximam os projetos investigados das OTs residem no fato de que assim como ocorre nessas organizações, a liderança demonstrou ser uma função de fundamental relevância (Bakker, 2011), no sentido de coordenar e direcionar o trabalho colaborativo. O exercício dessa liderança, inclusive, acontece de forma participativa (Burke e Morley, 2016), o que foi percebido nos casos, pois, mesmo que algumas decisões sejam centralizadas, normalmente, os integrantes do projeto tiveram a liberdade de manifestar suas opiniões durante o processo decisório.

As temáticas dos *stakeholders*, da TDR, e das OTs estão intrinsicamente relacionadas ao ambiente externo. É nesse contexto que envolve não só o sistema organizacional em que as OTs estão incorporadas, mas também o meio social no qual ocorrem os relacionamentos com os *stakeholders*, que se localizam os principais recursos necessários à sobrevivência das organizações (Grabher, 2004). Aproximando-se das OTs, os casos investigados demonstraram que os projetos de *software* livre, embora insistam em dispor de condições para um funcionamento autônomo (Burke & Morley, 2016), na prática (nos casos examinados) precisaram recorrer ao contexto externo e às relações com os *stakeholders* para conseguir os meios dos quais necessitavam para se manterem ativos. Isso foi algo observado em cada um dos projetos analisados nesta pesquisa, os quais ao longo das suas trajetórias de desenvolvimento não ficaram dependentes apenas da atuação de voluntários, mas tiveram de recorrer ao ambiente, nele buscando obter os recursos financeiros, humanos e materiais para desenvolver o *software*.

### 3.5 Conclusões

Esta seção reúne as conclusões mais relevantes procedentes das análises dos dados coletados por meio das entrevistas e da pesquisa documental efetuada. A compilação dos resultados é apresentada, e, posteriormente, são feitas as confrontações desses resultados com os objetivos da pesquisa e a tese proposta.

#### 3.5.1 Síntese dos resultados

A partir do levantamento realizado mediante pesquisa documental e junto a profissionais da área de *software* livre, assim como da aplicação de critérios decorrentes da literatura e dos objetivos propostos na pesquisa, foi possível chegar aos três casos selecionados para o estudo de casos múltiplos. Os critérios definidos, vale salientar, consideraram que além dos casos terem surgido no Brasil, deveriam apresentar diferentes características de desenvolvimento (Whyn, 2004; Xu et al., 2005; Latterman & Stieglitz, 2005; Panchal, 2010). Além disso, a própria conveniência e disponibilidade dos entrevistados em cooperar com a pesquisa precisaram ser consideradas na escolha dos casos, para o alcance da consistência dos resultados trazidos nesta pesquisa.

Com as três unidades de análise do estudo de casos múltiplos definidas, iniciou-se o momento em que foi seguido o protocolo delineado na seção metodológica. Assim, conforme

o estabelecido, a coleta de dados se concentrou na realização de entrevistas e da pesquisa documental, como definida por Flick (2009) e Bowen (2009). A análise de dados, a seu turno, com o suporte do *software* Atlas.ti, fez uso da técnica de análise de conteúdo, nos moldes trazidos por Bardin (2010). Da operacionalização desses procedimentos de coleta e análise dos dados, foram realizadas as verificações referentes à governança, na perspectiva da estrutura e dos processos de controle, assim como às principais transformações ocorridas no desenvolvimento dos projetos de *software* livre comunitários.

### 3.5.1.1 Estrutura organizacional e decisória

No que diz respeito à estrutura organizacional e decisória dos casos analisados, notou-se que ao longo da sua trajetória todos eles contaram com a infraestrutura física e tecnológica fornecida por alguma instituição, destacando-se a existência de órgãos de fomento, patrocinadores ou de fundação de *software* livre. As tecnologias da informação e comunicação também apareceram como recursos cruciais nas atividades colaborativas virtuais executadas pelos projetos de *software* livre, principalmente em virtude da descentralização geográfica dos integrantes. Dentre os principais recursos de TIC utilizados, sobressairam-se: as listas de *e-mail*; uso de plataforma de hospedagem de código-fonte e para controle de versões, tais como *GitHub*, *GitLab*, *Sourceforge*; *home page*, para divulgação de informações sobre o projeto; ferramentas direcionadas para a comunicação por chat e/ou vídeo, com ênfase no uso da multiplataforma do Google *Hangouts*.

O Quadro 10 oferece uma visão geral das principais percepções sobre a estrutura organizacional e decisória dos projetos de *software* livre comunitários. Essas percepções foram apresentadas em função do modo formal ou informal como ocorreram nos agrupamentos examinados.

**Quadro 10** ó Aspectos formais e informais da estrutura organizacional e decisória

<b>Estrutura</b>	<b>Formal</b>	<b>Informal</b>
<b>Hierarquia</b>	A existência de uma hierarquia formal foi notada principalmente por meio da separação entre o grupo que realiza a gestão do projeto e a equipe que atua no desenvolvimento do <i>software</i> .	A hierarquia informal foi identificada nas relações laterais e horizontais de autoridade, nas quais sobressaem os integrantes do grupo central do coletivo em função do nível de conhecimento e das contribuições trazidas ao projeto.

<b>Estrutura</b>	<b>Formal</b>	<b>Informal</b>
<b>Papéis</b>	Os papéis de gestão formais apareceram associados às figuras dos docentes, diretor e coordenador do projeto, <i>release manager</i> , <i>chair</i> , e membros do comitê de gestão do projeto. No desenvolvimento, os papéis formais identificados foram os de <i>committers</i> , bolsistas e estudantes.	Todos os projetos apresentaram voluntários colaborando em torno do <i>software</i> . Destacaram-se entre os voluntários, integrantes de organizações externas com as quais os projetos mantiveram contatos. Mesmo com a definição formal de papéis, conforme a necessidade, ficou constatado que os integrantes podem exercer múltiplas funções no projeto.
<b>Centralização das decisões</b>	Mesmo com a descentralização dos participantes, os casos caracterizaram-se por apresentar decisões mais estratégicas e definidoras dos rumos de desenvolvimento do <i>software</i> , centralizadas na equipe de gestão do projeto.	Tal como aconteceu na hierarquia, percebe-se que os integrantes do projeto com maior experiência e que atuam no grupo central são os que costumam ter as suas opiniões consideradas quando uma decisão relevante precisa ser tomada.

**Fonte:** Elaborado pela autora, com base na análise dos dados da pesquisa (2019)

Em relação à hierarquia, atrelada à estruturação dos papéis, similaridades foram verificadas nos casos, visto que todos eles se distinguiram por conviver com aspectos formais e informais no decorrer do seu desenvolvimento. A hierarquia informal foi perceptível, em especial dentre os membros mais antigos e detentores de maior conhecimento sobre o projeto, o que aproxima os casos do que a literatura de *software* livre afirma, sobre a existência de uma autoridade lateral e horizontalizada (Dahlander & O'Mahony, 2011). Já na hierarquia formal, que apareceu associada a papéis definidos, percebeu-se nos agrupamentos a existência de uma divisão clara e de relação de subordinação entre os indivíduos responsáveis pela gestão do projeto e aqueles atuantes na equipe de desenvolvimento do *software*.

Alguns papéis formalmente identificados foram os de *release manager* (RM) e *committer*, visto tanto no Noosfero, quanto no OpenNLP. No entanto, no Noosfero o RM tem a incumbência adicional de, além de ser o responsável pelo lançamento de versões, fazer a gestão do projeto. O OpenNLP conta com outros papéis mais específicos para a gestão do projeto, tais como o *chair* e os membros do PMC, sendo o primeiro o que concentra os trabalhos burocráticos do coletivo. No Kytos e no CoGrOO, que em dado momento converge com o OpenNLP, a divisão de papéis de modo formal foi claramente visível entre os responsáveis pela gestão, e a equipe de desenvolvimento do projeto. Ambos os projetos também tiveram em comum o fato de terem na equipe de desenvolvimento um misto entre desenvolvedores contratados formalmente e voluntários, o que se distancia da visão de que as

iniciativas de *software* livre seriam significativamente formadas por voluntários (Shah, 2006; Seidel & Stewart, 2011).

Independentemente de ser exercida de modo formal, ou não, uma função presente em todos os casos foi a da liderança. A liderança não necessariamente apareceu concentrada em um único indivíduo do projeto, mas distribuída principalmente entre os indivíduos com maior perfil para assumir esse papel. Essa percepção de que a existência de liderança é crucial para que os projetos de *software* livre alcancem os seus resultados, foi abordada por autores como Lee e Cole (2003) e Xu et al. (2005). Outro ponto de convergência com a literatura esteve relacionado aos papéis desempenhados nos coletivos. Apesar de alguns papéis serem formalmente designados, na prática, todos os projetos apresentaram flexibilidade no exercício deles, de modo que os indivíduos podem exercer mais de uma função no agrupamento. Essa sobreposição de papéis nos projetos de *software* livre havia sido abordada nas pesquisas de Zammuto et al. (2007) e Arazy et al. (2016).

Finalmente, no que diz respeito às decisões, o mesmo paradoxo presente na literatura dos projetos de *software* livre (Lee & Cole, 2003; West & O'Mahony, 2005; Seidel & Stewart, 2011) também foi verificado nos casos analisados. Nesse sentido, ao mesmo tempo em que os projetos desta investigação se caracterizaram por terem passado por um processo de descentralização da comunidade de desenvolvedores e de usuários, as decisões estratégicas referentes ao projeto estiveram centralizadas em representantes específicos desses projetos. Essa centralização demonstrou acontecer não só em torno dos gestores formais, mas também dos membros com maior experiência atuantes nos projetos, por serem os indivíduos considerados capacitados para tomar tais decisões. Assim, ainda que os casos tenham abertura para ouvir as partes interessadas, a decisão demonstrou ficar a cargo de um pequeno grupo de pessoas.

### 3.5.1.2 Processos de controle

Os processos de controle identificados nos casos da pesquisa mostraram estar intrinsecamente relacionados com os aspectos da estrutura, expostos anteriormente. Desse modo, assim como visto na questão da estrutura, o emprego de processos de controle formais em conjunto com os informais também se caracterizou por ser algo visível nos projetos. O Quadro 11, a seguir, faz um resumo dos processos de controle formais e informais identificados nos casos desta investigação.

**Quadro 11** ó Processos de controle formais e informais

<b>Controle</b>	<b>Processos</b>
<b>Formal</b>	Contribuições de atores externos; critérios de escolha e definição das responsabilidades atreladas aos papéis; lançamento de versões do <i>software</i> ; processo de tomada de decisão; regras de conduta; declaração de concordância com os termos de licença do projeto; entregas de relatórios de prestação de contas dos projetos; elaboração de pesquisas e trabalhos acadêmicos de disciplinas; reuniões de acompanhamento junto aos bolsistas e estudantes colaboradores do projeto.
<b>Informal</b>	Encontros presenciais ou virtuais entre os integrantes para acompanhamento dos trabalhos desenvolvidos; organização de eventos presenciais ou virtuais abertos ao público; reconhecimento dos integrantes que se destacaram pelas contribuições realizadas ao projeto; participação em eventos; publicação de artigos em eventos ou revistas científicas; revisão do código do <i>software</i> entre os <i>committers</i> ; alinhamento dos trabalhos independentes dos voluntários à produção interdependente do <i>software</i> .

**Fonte:** Elaborado pela autora, com base na análise dos dados da pesquisa (2019)

Como processos de controle formal destacaram-se os fluxos preestabelecidos referentes: às rotinas para contribuições dos atores externos ao projeto; aos critérios de escolha para os integrantes exercerem papéis e definição das responsabilidades atreladas a cada um desses papéis; às etapas necessárias para lançamento de versões do projeto; ao processo de tomada de decisão, em especial sobre as decisões que envolveram escolhas estratégicas; à definição de regras de conduta para os participantes do projeto; à declaração de concordância explícita com o conteúdo exposto nos termos de licença dos projetos; ao acompanhamento do projeto, principalmente para a elaboração de relatórios de prestação de contas.

Em cada um dos casos averiguados, o controle formal apareceu atrelado às fases vivenciadas pelos projetos. No CoGrOO, o controle formal foi evidente quando o projeto recebeu financiamento da Finep, esteve vinculado às disciplinas e pesquisas desenvolvidas na USP, até o momento em que passou a fazer interface com o projeto OpenNLP, atualmente gerenciado pela fundação Apache. No Noosfero, a interdependência com a Colivre sempre aconteceu, mesmo quando o projeto passou a ter contatos com atores de organizações externas, que foi, inclusive, o motivo que levou à necessidade de a cooperativa de promover a formalização de alguns processos. No Kytos, além de existir uma relação de dependência desde o início com a Unesp, o projeto conseguiu estabelecer o relacionamento com a empresa patrocinadora Huawei.

Embora algumas pesquisas tenham chegado a abordar a existência de processos formais de controle no âmbito dos arranjos comunitários, essa ocorrência geralmente esteve associada à expansão de projetos globais de *software* livre, tal como o Linux e o Debian, por

exemplo (O'Mahony & Ferraro, 2007; Panchal, 2010). Na análise dos projetos desta pesquisa, no entanto, os procedimentos formais e informais de controle pareceram acompanhar os momentos de maior e de menor atividade verificados em torno do desenvolvimento do *software*, principalmente nos relacionamentos estabelecidos pelos projetos com outras organizações formais.

Os processos de controle informais, a seu turno, que fazem uso de mecanismos de socialização e interação intensiva para direcionar os esforços dos indivíduos ao atingimento dos resultados dos coletivos, também estiveram presentes nos casos desta pesquisa. Nesse sentido, além do controle social, as outras duas modalidades de controle que apareceram nos projetos investigados de forma entrelaçada com essa questão da socialização foram o autocontrole e o controle entre pares.

O controle social predominou nos projetos pelas interações virtuais ou presenciais constantes entre os participantes. Os encontros virtuais, que evidenciaram serem os mais comuns, demonstraram ter como principal propósito o debate de diversas questões sobre o projeto, ocorrendo conforme a necessidade. Apesar de não ter uma periodicidade rígida, todos os casos apontaram fazer regularmente esses encontros, seja para acompanhar o que os integrantes têm feito, seja para pensar questões futuras sobre o projeto. Vale salientar, que nos projetos que tiveram contratação de funcionários para atuar no desenvolvimento, com ênfase no CoGrOO e o Kytos, essa rotina de reuniões demonstrou acontecer de maneira mais formalizada. Os encontros presenciais, por sua vez, embora não tão frequentes quanto os virtuais, também foram percebidos nos agrupamentos desta pesquisa, principalmente nas oportunidades em que os integrantes conseguiram estar próximos fisicamente.

Além dos encontros internos entre os integrantes do coletivo, os casos relataram também a ocorrência de interações com atores externos ao projeto. Nesse aspecto, destacaram-se os contatos estabelecidos com o público por meio de palestras ministradas e de artigos apresentados em eventos técnicos ou científicos. O reconhecimento dos membros, tanto internamente pela questão da meritocracia existente, quanto pelo recebimento de prêmios, entrevistas e publicações que ocorreram sobre os projetos também demonstrou ser uma forma de controle social.

Todos esses aspectos de controle social tiveram influência perceptível no autocontrole dos integrantes, principalmente entre os que exerceram atividades nos agrupamentos de forma voluntária. Assim, ainda que os membros tenham demonstrado nas entrevistas ter certa autonomia nos trabalhos que executaram, por conta desse envolvimento



com o projeto, as suas ações individuais terminaram mostrando-se articuladas com a produção final interdependente do *software*.

Finalmente, outro achado da pesquisa, também convergente com a teoria, esteve na questão da revisão das contribuições propostas (Lee & Cole, 2003; Lattemann & Stieglitz, 2005; Howison & Crowston 2014; Kolbjørnsrud, 2016). Apesar de as colaborações externas acontecerem de modo mais formal, sendo necessariamente revisadas após submetidas, quando essas contribuições revelaram acontecer por meio dos *committers* do projeto, o processo informal identificado foi o de controle entre pares. Essa prática, em que os *committers* revisam os trabalhos uns dos outros, tornou-se claramente identificada ao longo do desenvolvimento do *software* em todos os casos, constituindo uma garantia adicional para manter a qualidade do código-fonte dos projetos.

Nota-se, assim, que o controle social, o autocontrole e o controle entre pares se caracterizaram pelo senso de identificação e de pertencimento ao projeto, o que permitiu a criação de uma identidade do coletivo. Desse modo, as práticas passaram a ser compartilhadas entre os membros e trouxeram um entendimento comum das atividades desenvolvidas, orientando os comportamentos individuais aos resultados pretendidos (Wenger, 1998; Seidel & O'Mahony, 2014).

### 3.5.1.3 Transformações e associação dos casos ao aporte teórico e conceitual da tese

Em relação às características dos casos que demonstraram aproximá-los dos projetos de *software* livre comunitários, destaca-se que todos eles, desde o início até o momento atual, tiveram o seu código-fonte disponível na Internet para o recebimento de contribuições. Essa abertura comunitária, no entanto, não significou que o projeto apresentasse uma linha evolutiva ascendente e com uma significativa comunidade de voluntários sempre ativa, como era esperado (Shah, 2006; Seidel & Stewart, 2011; O'Mahony & Lakhany, 2011). Pelo contrário, todos os casos tiveram períodos de maior e de menor produtividade, dependendo da fase vivenciada. Além disso, apresentaram características diferentes em relação à atratividade do projeto, de modo que alguns deles, inclusive, para se manter funcionando recorreram à contratação formal de profissionais para atuar na produção do *software*.

Nesse sentido, o CoGrOO que era um *software* voltado ao usuário final, apresentou uma comunidade restrita de desenvolvedores, de forma que precisou recorrer à contratação de profissionais ao receber o financiamento da Finep. Já o OpenNLP, apesar de ter incorporado parte do código do CoGrOO, o projeto é uma biblioteca que funciona no âmbito de uma

fundação, o que faz com que desenvolvedores de todo o mundo contribuam com ele. O Noosfero tem como voluntários desenvolvedores que usaram o *software* nas organizações em que atuam, retornando o desenvolvimento com contribuições à comunidade do projeto. O Kytos, além de ser voltado para desenvolvedores, tem foco específico em redes de universidade, de modo que precisou recorrer à contratação de desenvolvedores com o patrocínio recebido da Huawei.

Em relação às transformações verificadas nos casos, algo que se destacou de modo visível refere-se ao fato de que os momentos mais ativos dos projetos estiveram associados aos relacionamentos estabelecidos com instituições formais. O CoGrOO esteve associado à USP, inclusive ao CCSL desta universidade, recebeu recursos da Finep e do Google, além de ter parte do seu código atualmente integrado ao projeto OpenNLP, que funciona integrado à fundação Apache. O Noosfero, até os dias atuais, tem como principal patrocinador a Colivre, além de ter mantido parcerias de desenvolvimento com vários outros atores organizacionais externos. O Kytos, finalmente, além de contar com o suporte do NCC da Unesp, também tem parcerias com redes e comunidades nacionais e internacionais da sua área e o patrocínio da empresa Huawei.

Diante das transformações vivenciadas pelos projetos, em que contatos com outras organizações foram estabelecidos, percebe-se que, tal como exposto pela literatura, a governança é um conceito fluido e em constante adaptação (Lattemann & Stieglitz, 2005; Xu et al., 2005; Jensen & Scacchi, 2010; Panchal, 2010; Faraj et al., 2011). A estrutura e o controle dos casos investigados caracterizaram-se por serem muito mais complexos do que a visão que aproxima os projetos de *software* livre comunitários a um empreendimento fortemente informalizado e não hierárquico (Raymond, 1999; Seidel & Stewart, 2011; Arazy et al., 2016). Na realidade, o formal e o informal demonstraram conviver em sintonia com os acontecimentos dos projetos, apresentando traços mais formalizados, sobretudo quando nos contatos mantidos com outras organizações, voltados, especialmente, para obtenção de recursos necessários ao funcionamento dos coletivos. Outra constatação é que o desenvolvimento dos projetos não seguiram estritamente a trajetória evolutiva de crescimento apontada nas pesquisas de Lattemann e Stieglitz (2005) e Xu et al. (2005). Durante o percurso seguido por cada empreendimento, momentos de maior e de menor atividade foram notados principalmente nos projetos mais antigos. O Quadro 12 a seguir resume as principais transformações na estrutura e nos processos de controle verificados nos casos examinados.

**Quadro 12** ó Principais transformações da governança

<b>Dimensões</b>	<b>Características das transformações</b>
<b>Estrutura</b>	Os contatos estabelecidos com atores organizacionais externos levaram à formalização da hierarquia e de papéis. Contudo, essa formalização não impediu a existência de uma autoridade lateral em função das contribuições dadas ao projeto, assim como que os integrantes exercessem múltiplos papéis no coletivo. As decisões estratégicas demonstraram acontecer de modo centralizado, independente do momento vivenciado pelo projeto.
<b>Controle</b>	Os contatos estabelecidos com atores organizacionais externos fizeram com que processos de controle fossem estabelecidos no desenvolvimento do <i>software</i> . Mas, independentemente do momento vivenciado, principalmente em relação aos indivíduos voluntários, notou-se que as práticas informais de controle promoveram o alinhamento dos comportamentos e um senso de pertencimentos ao projeto, no sentido de garantir a qualidade do <i>software</i> .

**Fonte:** Elaborado pela autora, com base na análise dos dados da pesquisa (2019)

Com base nesse painel de transformações, algumas aproximações também puderam ser estabelecidas entre os casos desta pesquisa e o aporte conceitual e teórico utilizado, envolvendo: a abordagem dos *stakeholders*; a teoria da dependência de recursos; as organizações temporárias. Essas associações são resumidas no Quadro 13 a seguir.

**Quadro 13** ó Aproximação dos casos ao aporte conceitual e teórico da tese

<b>Aporte conceitual e teórico</b>	<b>Aproximação identificada nos casos</b>
<b>Abordagem dos <i>stakeholders</i></b>	Desenvolvimento de relações com atores interessados ( <i>stakeholders</i> ); influência mútua entre os <i>stakeholders</i> e os projetos; necessidade dos projetos de gerenciar as expectativas dos <i>stakeholders</i> .
<b>Teoria da dependência de recursos</b>	Influência externa do ambiente; os recursos externos são cruciais para sobrevivência dos projetos; influência mútua entre o ambiente e o projeto; postura proativa dos gestores para buscar os recursos no ambiente por meio de estratégias.
<b>Organizações temporárias</b>	Interdependência dos atores para realização de tarefas complexas; integração de resultados e <i>expertises</i> para alcance de resultados comuns; orientação para projetos (lançamento de versões); ponto de terminação estabelecido; a liderança mostrou-se um fator crucial para direcionar o grupo; relacionamentos de interesse essencial com o ambiente

**Fonte:** Elaborado pela autora, com base na revisão de literatura realizada (2019)

As informações trazidas nesta seção corroboram que o entendimento amplo dos arranjos comunitários, como o que foi realizado nesta tese constituiu o caminho escolhido para o conhecimento da natureza operativa dos projetos de *software* livre comunitários. Uma vez que as comunidades dos projetos estão inseridas em todo um ambiente de transformações e relacionamentos com outros atores externos, o estabelecimento da contextualização se mostra fundamental nos estudos que buscam entender a governança desses coletivos.

### 3.5.2 Confronto com os objetivos e com a tese proposta

À vista dos resultados das análises, se tornou imperativo confrontá-los com os objetivos delimitados no início desta tese. No que diz respeito ao primeiro objetivo, os casos da pesquisa foram descritos individualmente em função das principais características observadas nas suas trajetórias de desenvolvimento, o que envolveu a história, a estrutura organizacional e decisória e os processos de controle observados nos projetos. Essa descrição geral das unidades de análise se pautou pelos levantamentos das entrevistas e da pesquisa documental realizada, o que foi fundamental para o entendimento aprofundado de cada projeto de *software* livre, objeto da investigação.

Vencido o primeiro objetivo, as outras duas metas preconizadas para a pesquisa envolveram a análise conjunta dos casos, criando margem para uma comparação dos projetos entre si, bem como para associações aos conceitos e teorias trazidos pela tese. O segundo objetivo se concentrou na análise dos casos em função dos procedimentos de governança identificados, representados pelos elementos estruturais e processos de controle. Já a resposta ao terceiro objetivo, a seu turno, trouxe um mapeamento das transformações que aconteceram ao longo do desenvolvimento dos projetos e que determinaram mudanças na sua governança.

Todas essas fases executadas na pesquisa, tomadas em conjunto, permitem reconhecer que os objetivos um a um foram considerados de forma a atender ao objetivo geral e a pergunta central de pesquisa da tese. A consequência desses passos foi tornar possível uma compreensão da governança e das suas transformações, a partir da análise dos elementos estruturais e dos processos de controle dos projetos de *software* livre comunitários investigados. De modo alinhado aos objetivos propostos, a tese central da pesquisa encontrou sua identidade na ideia de que a governança se transforma e se torna mais formal em função da trajetória de desenvolvimento que os projetos de *software* livre comunitários apresentam, todavia nos moldes de avanços e recuos, como reflexos dos momentos vivenciados pelos projetos. Em relação a essa tese central, passando em revista todos os resultados encontrados, algumas impressões sobre tais projetos merecem destaque.

- Em todos os casos, ficou visível a dificuldade de se estabelecer a separação entre a comunidade e as iniciativas formais que são desenvolvidas por essas estruturas no âmbito dos projetos. Ao mesmo tempo em que os empreendimentos foram criados dentro da filosofia comunitária, em que o código tornou-se disponível para que outras pessoas pudessem utilizá-lo e/ou contribuir, para aperfeiçoamento dele na prática, as atividades de desenvolvimento distinguiram-se por extrapolar a

comunidade em si. Daí, portanto, a importância da governança ser estudada considerando não só a comunidade dos projetos, mas todo o contexto no qual ela se insere e os relacionamentos que estabelecem ao longo do tempo no interior e no entorno da mesma.

- A governança, verificada nesta pesquisa pela estrutura e processos de controle, demonstrou ser um conceito em constante transformação, adaptando-se em função da trajetória de desenvolvimento vivenciada pelos projetos. No decorrer dessas mudanças os procedimentos formais e informais de governança empregados nos projetos demonstraram acontecer de modo muito mais intrincado do que o esperado. Assim, ainda que nas trajetórias de desenvolvimento dos projetos traços comunitários tenham permanecido, certas e até expressivas formalizações fizeram-se necessárias na gestão dos coletivos, principalmente em virtude dos relacionamentos estabelecidos com outros atores organizacionais externos.
- A evidência notada em todos os casos, é que os contatos realizados com outras organizações aconteceram essencialmente para a obtenção de recursos (financeiros, materiais ou humanos) necessários à sobrevivência do projeto. Notou-se que a liderança, seja exercida formalmente, seja praticada de modo informal, demonstrou ser crucial na busca desses recursos no ambiente em que os projetos se encontram inseridos e nas definições dos rumos tomados pelos projetos. Mesmo que voluntários tenham atuado nos projetos investigados, o fato é que não se detectou uma constância na participação deles, reforçando a hipótese de que os recursos desempenham papel fundamental no sentido de permitir o funcionamento em longo prazo dos empreendimentos comunitários, contando com um quadro mínimo de colaboradores.
- Ainda que a literatura do tema especifique fases de desenvolvimento pelas quais os projetos de *software* livre comunitários passam, ou até características válidas para diferenciá-los dos arranjos organizacionais tradicionais, nos casos averiguados essa lógica de funcionamento não se mostrou bem definida. Embora características informais comunitárias tenham sido mantidas, as transformações que levaram ao desenvolvimento dos projetos estiveram fundamentalmente associadas às relações desenvolvidas com organizações externas para obtenção de recursos. Nessas relações, formalizações foram necessárias na governança e demonstraram a capacidade de conviver com a informalidade dos espaços comunitários. Sendo assim, ao invés de fases de desenvolvimento acontecerem

numa perspectiva evolutiva e de crescimento dos projetos, na realidade, esses projetos apresentaram momentos de maior e de menor atividade fundamentalmente guiados pela existência de recursos obtidos junto a organizações formais.

Diante do exposto, o que se percebe, nos casos examinados, é que a imagem idealizada do *obazarö* e da informalidade para definir os arranjos comunitários não foi sustentada pelas evidências colhidas em campo. A própria ideia de que os projetos de *software* livre seriam mantidos de forma significativa pela presença de voluntários não aconteceu de modo autônomo, isto é, sem a existência de uma organização formal no direcionamento desses voluntários. Durante toda a trajetória de desenvolvimento dos projetos, em especial nos mais antigos, não houve um crescimento em etapas sequenciais, mas, sim, momentos mais ativos e outros de inflexão, que, por sua vez, levaram esses projetos a buscar recursos em seus ambientes para sobreviverem a médio e longo prazos. Nessa busca por recursos, formalizações internas vieram em resposta aos relacionamentos mantidos pelos projetos com atores organizacionais externos, representados por parceiros, órgãos de fomento, patrocinadores ou fundações. Consequentemente, a governança se mostrou um conceito flexível e adaptável às vivências dos projetos, de modo que os aspectos formais de estrutura e de controle implementados demonstraram conviver de forma entrelaçada com as práticas informais inerentes ao funcionamento das iniciativas colaborativas comunitárias.

#### 4 CONSIDERAÇÕES FINAIS

Conforme já exposto, este trabalho de tese teve duas etapas estabelecidas: a do ensaio teórico e a da pesquisa empírica. O ensaio teórico, sob o amparo da revisão de literatura na área de *software* livre, buscou distinguir as características gerais e de governança dos projetos colaborativos comunitários comparando-os às organizações tradicionais. Ademais, trouxe reflexões críticas sobre a importância dos estudos do tema, ao considerar as modificações ocorridas no desenvolvimento dos projetos, em virtude do caráter dinâmico da governança nestes empreendimentos. A pesquisa empírica, por sua vez, envolveu a ida da pesquisadora ao campo, visando a realização de um estudo de casos múltiplos de projetos de *software* livre comunitários. Na fase empírica, foram consideradas as reflexões levantadas no ensaio teórico, de forma que a intenção foi a de compreender a governança e as suas transformações no decorrer da trajetória de desenvolvimento dos projetos selecionados. Os três casos nacionais escolhidos atenderam a critérios estabelecidos pela literatura e com base nos objetivos propostos para a tese. Em ambas as pesquisas, na teórica e na empírica, a governança foi retratada por meio da perspectiva da estrutura organizacional e decisória e dos processos de controle.

Revisados os direcionamentos trazidos pelos estudos teórico e empírico, vale salientar que todos os resultados alcançados estiveram pautados pela verificação da tese proposta, fundamentada na ideia de que os projetos colaborativos de *software* livre, organizados no formato comunitário virtual, se tornariam mais formais em função da trajetória de desenvolvimento vivenciada. Esta tese, em parte, teve a sua confirmação, mas também identificou particularidades adicionais sobre as transformações dos projetos. No conjunto, as pesquisas desenvolvidas neste trabalho conseguiram trazer contribuições teóricas e gerenciais, como as apresentadas nos parágrafos seguintes.

Na perspectiva teórica, as evidências alcançadas permitiram observar que, apesar de existirem concepções na literatura dos projetos de *software* livre comunitários que traçam características gerais às quais associam esses projetos a um funcionamento não hierárquico e informal, na realidade os fatos podem não acontecer dessa forma. Em todos os casos averiguados a governança demonstrou ser um conceito flexível, que se acomoda a mudanças havidas no domínio dos projetos e no contexto no qual eles estão inseridos, o que faz com que cada coletivo apresente especificidades atreladas às trajetórias de desenvolvimento vivenciadas.

À parte dos caminhos distintos, semelhanças ou convergências também foram identificadas nos casos examinados. Em todos os projetos, procedimentos de governança

formais foram utilizados em algum momento e demonstraram conviver de forma entrelaçada com os atributos informais inerentes aos projetos colaborativos comunitários. A formalização, no entanto, não se caracterizou por acontecer em fases evolutivas decorrentes do crescimento dos projetos, tal como preconizado em algumas pesquisas de *software* livre. Pelo contrário, nos casos submetidos à investigação, períodos de maior e de menor produtividade foram observados, de modo que os aspectos formais da governança apareceram associados majoritariamente aos relacionamentos estabelecidos pelos projetos com outras organizações formais, tais como patrocinadores, fundações, órgãos de fomento, parceiros etc. Tais relacionamentos aconteceram, sobretudo, pela necessidade intrínseca dos projetos de conseguirem no ambiente os recursos necessários às suas operações em médio ou longo prazo. Essas constatações permitiram a associação dos resultados com o aporte conceitual e teórico usado neste trabalho de tese, no qual os projetos de *software* livre comunitários se aproximam do perfil de funcionamento das organizações temporárias, sendo este o motivo pelo qual precisam interagir e gerenciar interdependências com *stakeholders*, para obter no ambiente os meios necessários para permanecerem ativos.

Do ponto de vista gerencial, os resultados permitiram confirmar a importância que os gestores ou líderes dos projetos de *software* livre comunitários possuem e exercem no direcionamento dos agrupamentos que integram. Independentemente de a governança acontecer do modo formal ou informal, os responsáveis pela gestão precisam ser proativos e lançar mão de táticas para acessar os apoios e recursos que demandam os seus respectivos projetos. Mesmo que o formato comunitário permita o engajamento de voluntários na produção do *software*, o recebimento de contribuições abertas não demonstrou ser suficiente para que os projetos de *software* se mantenham ativos no decorrer dos anos. A inconstância do trabalho voluntário e a permanente necessidade de obter novos recursos tornam os relacionamentos com outros atores organizacionais basilares para a manutenção dos projetos. Dessa forma, ficou patente que a governança das iniciativas comunitárias precisa ser pensada pelos gestores para além da comunidade, abarcando o projeto como um todo e o contexto no qual estas iniciativas estão inseridas. Nas inter-relações dos projetos de *software* livre comunitários e outras organizações com a finalidade da aquisição de recursos, no entanto, formalizações da estrutura e do controle podem ocorrer, ainda que atributos comunitários sejam mantidos.



#### 4.1 Limitações e direcionamentos futuros do estudo

Ainda que precauções tenham sido tomadas no curso de toda a investigação, sobretudo no procedimento metodológico, a existência de limitações é algo incontestável e que se faz presente em qualquer pesquisa.

A primeira limitação se refere à subjetividade nos acontecimentos retratados, tanto por parte da pesquisadora, quanto dos respondentes das entrevistas. Ainda que se tenha tentado ter uma visão imparcial dos fatos, a subjetividade provou ser fenômeno inerente às investigações qualitativas, principalmente por envolver interpretações pessoais sobre os fatos apurados. A própria realização da pesquisa qualitativa, estruturada na estratégia de estudos de casos múltiplos, faz com que os resultados alcançados sejam um reflexo da realidade sob investigação, o que restringe generalizações para além dos contextos estudados.

A escolha dos casos, embora pautada em critérios objetivos, por meio da indicação de profissionais com conhecimento no tema de *software* livre e do levantamento documental em eventos nacionais da área, também demonstrou ser outro fator limitante. Por mais que a pesquisadora tenha buscado a neutralidade, a acessibilidade aos dados e a representatividade dos casos precisaram ser consideradas na escolha das possíveis unidades de análise para estudo. Sendo assim, os casos também foram selecionados pela abertura propiciada à pesquisadora, já que no entendimento da trajetória de desenvolvimento deles seria fundamental ter pessoas que estivessem dispostas a relatar em detalhes as suas vivências no contexto investigado.

Outra limitação se refere à restrição de tempo para atender aos prazos definidos na tese. A existência de um cronograma que precisava ser cumprido para a coleta e a análise de dados, nem sempre uma variável sob total controle do pesquisador, revelou ser um aspecto limitante no sentido de se conseguir alcançar uma visão longitudinal dos casos.

Em um dos casos, especificamente, houve uma limitação na coleta de dados, uma vez que ele não era o principal alvo do levantamento de campo. Essa situação específica envolveu o CoGrOO, visto que o OpenNLP não era uma unidade objeto da investigação, mas que passou a ser considerada como tal pela interface que tem com o código do corretor ortográfico. Como forma de compensar a ausência de entrevistas com outros integrantes do OpenNLP, além do entrevistado E1, a pesquisadora realizou o envio de formulários e fez uma pesquisa documental mais detalhada na página virtual da Apache, com o foco em conseguir compreender a governança desse projeto indiretamente incidente nas operações do projeto previamente selecionado para estudo (o CoGrOO).

De modo associado com as limitações identificadas, novos estudos e reflexões podem ser trazidos com a finalidade de ampliar os conhecimentos acumulados nesta pesquisa sobre a governança de projetos de *software* livre comunitários. Como proposta de agenda para pesquisas adicionais, apontam-se as seguintes ideias:

- Realização de investigações em moldes semelhantes ao desta tese, no sentido de verificar se as tendências aqui encontradas sobre a governança se repetem, considerando um número mais amplo de projetos de *software* livre comunitários, não só nacionais, mas também internacionais, ou até mesmo outros tipos de empreendimentos baseados na lógica comunitária;
- Elaboração de pesquisas que considerem a governança a partir de outras variáveis além da estrutura organizacional e decisória e dos processos de controle, uma vez que o conceito é vasto o suficiente para acomodar diferentes meios formais ou informais utilizados, com vistas ao direcionamento das ações para o alcance dos resultados organizacionais esperados;
- Desenvolvimento de estudos comparativos sobre governança que considerem as trajetórias de desenvolvimento de projetos com diferentes perfis de maturidade, a exemplo de projetos com patrocinadores ou integrados a fundações de *software* livre *versus* projetos que não contem com este tipo de suporte organizacional;
- Execução de pesquisas qualitativas que utilizem procedimentos metodológicos diferentes dos que foram empregados nesta tese, a exemplo da netnografia nos ambientes virtuais utilizados pelos projetos de *software* livre comunitários;
- Produção de estudos mistos, ou que utilizem a abordagem quantitativa e longitudinal, com foco na verificação de associações entre aspectos formais e informais da governança nos projetos de *software* livre comunitários, inclusive em combinação com outras variáveis, de forma que seja possível obter relações e generalizações com maior grau de confiabilidade.

## REFERÊNCIAS

- Adler, P., Heckscher, C., & Prusak, L. (2011). Cómo construir una empresa colaborativa: cuatro claves para crear una cultura de confianza y trabajo en equipo. *Harvard Business Review*, 89(6), 44-52.
- Alavi, S. (2014). Netnography: an internet-optimized ethnographic research technique. In A. Takhar-Lail, & A. Ghorbani (Eds.). *Market Research Methodologies: Multi-Method and Qualitative Approaches: Multi-Method and Qualitative Approaches*, 71-77.
- Albers, S. (2005). *The design of alliance governance systems*. Köln (DE): Kölner Wissenschaftsverlag.
- Aldrich H. E., & Ruef, M. (2006). Organizations evolving. *TJ International Ltd. óPadstow*.
- Aldrich, H. E. (2007). *Organizations and environments*. Stanford: Stanford University Press.
- Arazy, O., Daxenberger, J., Lifshitz-Assaf, H., Nov, O., and Gurevych, I. (2016). Turbulent stability of emergent roles: The dualistic nature of self-organizing knowledge coproduction. *Information Systems Research*, 27(4), 792-812.
- Bakker, R. M. (2011). It's only temporary: Time and learning in inter-organizational projects.
- Bakker, R. M., Cambré, B., & Provan, K. G. (2009). The resource dilemma of temporary organizations: A dynamic perspective on temporal embeddedness and resource discretion. *Temporary organizations: Prevalence, logic and effectiveness*, 201-219.
- Bardin, L. (2010). *Análise de Conteúdo*. Lisboa: Edições 70.
- Bauer, J., Franke, N., & Tuertscher, P. (2016). Intellectual property norms in online communities: How user-organized intellectual property regulation supports innovation. *Information Systems Research*, 27(4), 724-750.
- Bonabeau, E. (2009). Decisions 2.0: The power of collective intelligence. *MIT Sloan management review*, 50(2), 45.
- Bowen, G. A. (2009). Document Analysis as a Qualitative Research Method. *Qualitative Research Journal*, 9(2), 27-40.
- Brinkmann, S. (2014). Interview. In *Encyclopedia of critical psychology*. Springer New York, 1008-1010.
- Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation. *Organization science*, 2(1), 40-57.
- Burke & Morley, 2016 Burke, C. M., & Morley, M. J. (2016). On temporary organizations: A review, synthesis and research agenda. *Human relations*, 69(6), 1235-1258.
- Carroll, G. R., & Hannan, M. T. (2000). Why corporate demography matters: Policy implications of organizational diversity. *California Management Review*, 42(3), 148-163.
- Carvalho, C. A. P. D. (1999). El control organizativo en las organizaciones no gubernamentales: un estudio de la realidad española. *Revista de Administração Contemporânea*, 3(3), 37-50.
- Casciaro, T., & Piskorski, M. J. (2005). Power imbalance, mutual dependence, and constraint absorption: A closer look at resource dependence theory. *Administrative science quarterly*, 50(2), 167-199.

- Chen, K. K., & O'Mahony, S. (2009). Differentiating organizational boundaries. In B. G. King, T. Felin, & D. A. Whetten (Eds.). *Studying Differences Between Organizations: Comparative Approaches to Organizational Research, Research in the Sociology of Organizations*. Bingley: Emerald Group Publishing Limited, 183-220.
- Chengalur-Smith, S., & Sidorova, A. (2003). Survival of open-source projects: A population ecology perspective. *In Proceedings of the International Conference on Information Systems (ICIS)*.
- Child, J. (1972). Organizational structure, environment and performance: The role of strategic choice. *Sociology*, 6(1), 1-22.
- Child, J., & Rodrigues, S. B. (2003). Corporate governance and new organizational forms: Issues of double and multiple agency. *Journal of Management and Governance*, 7(4), 337-360.
- Cohen, A. P. (2013). *Symbolic construction of community*. Routledge.
- Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, 18(4), 65-85.
- Crowston, K., Jullien, N., & Ortega, F. (2013). Sustainability of Open Collaborative Communities: Analyzing Recruitment Efficiency. *Technology Innovation Management Review*, 20-26.
- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44(2).
- Crowston, K., Wei, K., Li, Q., Eseryel, Y., & Howison, J. (2007). Self-organization of teams in free/libre open source software development. *Information and Software Technology Journal*, 49(6), 564-575.
- Dahlander, L., & O'Mahony, S. (2011). Progressing to the center: Coordinating project work. *Organization science*, 22(4), 961-979.
- Dahlander, L., & Wallin, M. W. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35(8), 1243-1259.
- Dahlander, L., Frederiksen, L., & Rullani, F. (2008). Online communities and open innovation. *Industry and innovation*, 15(2), 115-123.
- Das, T. K., & Teng, B. S. (1998). Between trust and control: Developing confidence in partner cooperation in alliances. *Academy of management review*, 23(3), 491-512.
- David, P. A., & Shapiro, J. S. (2008). Community-based production of open-source software: What do we know about the developers who participate? *Information Economics and Policy*, 20(4), 364-398.
- Davis, G. F., & Cobb, J. A. (2009). Resource dependence theory: Past and future. In S. B. Bacharach (Ed.). *Research in the sociology of organizations*. London: Elsevier.
- Demil, B., & Lecocq, X. (2006). Neither market nor hierarchy nor network: The emergence of bazaar governance. *Organization studies*, 27(10), 1447-1466.
- Donaldson, T., & Preston, L. E. (1995). The stakeholder theory of the corporation: Concepts, evidence, and implications. *Academy of management Review*, 20(1), 65-91.
- Drees, J. M., & Heugens, P. P. (2013). Synthesizing and extending resource dependence theory: A meta-analysis. *Journal of Management*, 39(6), 1666-1698.

- Druskat, V. U., & Wheeler, J. V. (2003). Managing from the boundary: The effective leadership of self-managing work teams. *Academy of Management Journal*, 46(4), 435-457.
- Dunbar, R. L., & Statler, M. (2010). A historical perspective on organizational control. *Organizational control*, 16-47.
- Dyer, W. G., & Wilkins, A. L. (1991). Better stories, not better constructs, to generate better theory: A rejoinder to Eisenhardt. *Academy of Management Review*, 16, 613-619.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, 14(4), 532-6550.
- Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50(1), 25-632.
- Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications*. MIT press.
- Fang, Y. H., & Chiu, C. M. (2010). In justice we trust: Exploring knowledge-sharing continuance intentions in virtual communities of practice. *Computers in Human Behavior*, 26(2), 235-246.
- Fang, Y., & Neufeld, D. (2009). Understanding sustained participation in open source software projects. *Journal of Management Information Systems*, 25(4), 9-50.
- Faraj, S., Jarvenpaa, S. L., & Majchrzak, A. (2011). Knowledge collaboration in online communities. *Organization science*, 22(5), 1224-1239.
- Fjeldstad, Ø. D., Snow, C. C., Miles, R. E., & Lettl, C. (2012). The architecture of collaboration. *Strategic Management Journal*, 33(6), 734-750.
- Fleming, L., & Waguespack, D. M. (2007). Brokerage, boundary spanning, and leadership in open innovation communities. *Organization science*, 18(2), 165-180.
- Flick, U. (2009). *Qualidade na pesquisa qualitativa: coleção pesquisa qualitativa*. Bookman Editora.
- Flyvbjerg, B. (2007) Five misunderstandings about case-study research. *Qualitative research practice*, 390-404.
- Forte, A., & Lampe, C. (2013). Defining, understanding, and supporting open collaboration: Lessons from the literature. *American Behavioral Scientist*, 57(5), 535-547.
- Forte, A., Larco, V., & Bruckman, A. (2009). Decentralization in Wikipedia governance. *Journal of Management Information Systems*, 26(1), 49-72.
- Foss, N. J., & Klein, P. G. (2013). Organizational governance. In Wittek R., Snijders, T. A. B., & Nee, V. (Eds.). *The Handbook of Rational Choice Social Research*. Stanford, CA: Stanford University Press, 513-6555.
- Freeman, R. E. (2010). *Strategic management: A stakeholder approach*. Cambridge university press.
- Friedman, A. L., & Miles, S. (2006). *Stakeholders: Theory and practice*. Oxford University Press on Demand.
- Frooman, J. (1999). Stakeholder influence strategies. *Academy of management review*, 24(2), 191-205.
- Galbraith, J. R. (2010). The multidimensional and reconfigurable organization. *Organizational Dynamics*, 39(2), 115-125.

- Galbraith, J. R. (2014). *Designing organizations: Strategy, structure, and process at the business unit and enterprise levels*. John Wiley & Sons.
- Gaskell, G. (2002). Entrevistas individuais e grupais. In: Bauer, M. W.; Gaskell, G (Orgs.). *Pesquisa Qualitativa com Texto, Imagem e Som*. Petrópolis: Vozes, 64-89.
- German, D. M. (2003). The GNOME project: A case study of open source, global software development, *Software Process Improvement and Practice*, 8(4), 2016-215.
- Germonprez, M., Kendall, J. E., Kendall, K. E., Mathiassen, L., Young, B., & Warner, B. (2016). A theory of responsive design: A field study of corporate engagement with open source communities. *Information Systems Research*, 28(1), 64-83.
- Gibbs, G. R. (2008). *Analysing qualitative data*. Sage.
- Glaser, B. G. (2001). *The grounded theory perspective: Conceptualization contrasted with description*. Sociology Press.
- Godoy, A. S. (2012). Estudo de caso qualitativo. In Godoi, C. K., Bandeira-de-Mello, R., & Silva, A. B. (Orgs.). *Pesquisa qualitativa em estudos organizacionais: paradigmas, estratégias e métodos*. 2. ed. São Paulo: Saraiva, 115-146.
- Grabher, G. (2004). Temporary architectures of learning: Knowledge governance in project ecologies. *Organization studies*, 25(9), 1491-1514.
- Grabher, G., & Ibert, O. (2014). Distance as asset? Knowledge collaboration in hybrid virtual communities. *Journal of Economic Geography*, 14(1), 97-123.
- Gregor, S. (2002). A theory of theories in information systems. In S. Gregor and D. Hart (Eds.), *Information Systems Foundations: Building the Theoretical Base*. Australian National University, Canberra, 1-20.
- Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research*, 2, 163-194.
- Hair, J., Babin, B., Money, A., & Samouel, P. (2015). *Fundamentos de métodos de pesquisa em administração*. Bookman Companhia.
- Hamersly, W. J. (2015). *Business governance best practices of virtual project teams*.
- Hargrave, T., & Van de Ven, A. (2006). A Collective Action Model of Institutional Innovation. *Academy of Management Review*, 31(4), 864-888.
- Hill, C. W. L.; Jones, G. R. (1998). *Strategic Management Theory: An Integrated Approach*. Boston: Houghton Mifflin.
- Hillman, A. J., Withers, M. C., & Collins, B. J. (2009). Resource dependence theory: A review. *Journal of management*, 35(6), 1404-1427.
- Howison, J., & Crowston, K. (2014). Collaboration through superposition: How the IT artifact as an object of collaboration affords technical interdependence without organizational interdependence. *MIS Quarterly*, 38(1), 29-50.
- Jensen, C., & Scacchi, W. (2010). Governance in open source software development projects: A comparative multi-level analysis. In *IFIP International Conference on Open Source Systems*. Springer, Berlin, Heidelberg, 130-142.
- Jones, G. R. (2013). *Organizational theory, design, and change*. Upper Saddle River, NJ: Prentice Hall.

- Kelty, C. (2008). *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 67-93.
- Kolbjørnsrud, V. (2016). Agency problems and governance mechanisms in collaborative communities. *Strategic Organization, Special Issue Article*, 1-36.
- Kozlowski, S. W., Watola, D. J., Jensen, J. M., Kim, B. H., & Botero, I. C. (2009). Developing adaptive teams: a theory of dynamic team leadership. Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches, 113-155.
- Krippendorff, K. (2013). *Content Analysis. An Introduction to Its Methodology*. Sage, London.
- Lakhani, K. R., & Von Hippel, E. (2003). How open source *software* works: öfreeö user-to-user assistance. *Research policy*, 32(6), 923-943.
- Lakhani, K. R., Lifshitz-Assaf, H., & Tushman, M. (2013). Open innovation and organizational boundaries: task decomposition, knowledge distribution and the locus of innovation. *Handbook of economic organization: Integrating economic and organizational theory*, 355-382.
- Lattemann, C., & Stieglitz, S. (2005). Framework for governance in open source communities. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, Hawaii, 192-201.
- Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York: Cambridge university press.
- Lee, G. K., & Cole, R. E. (2003). From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization science*, 14(6), 633-649.
- Li, X., Yoo, Y., & Zhang, Z. (2016). Searching for öStabilityö in Fluidity: A Routine-based View of Open Source *Software* Development Process. *Proceedings of the International Conference on Information Systems (ICIS)*.
- Lindberg, A., Berente, N., Gaskin, J., & Lyytinen, K. (2016). Coordinating interdependencies in online communities: A study of an open source *software* project. *Information Systems Research*, 27(4), 751-772.
- Lingo, E. L., & O'Mahony, S. (2010). Nexus work: Brokerage on creative projects. *Administrative Science Quarterly*, 55(1), 47-81.
- LióYing, J., & Salomo, S. R. (2013). Design of governance in virtual communities: definition, mechanisms, and variation patterns. *International Journal of Collaborative Enterprise*, 3(4), 225-251.
- Longhurst, R. (2003). Semi-structured interviews and focus groups. *Key methods in geography*, 117-132.
- Lorsch, J. W., Lawrence, P. R., & Dalton, G. W. (Eds.). (1970). *Organizational Structure and Design*. Irwin-Dorsey.
- Loughry, M. (2009). Peer control in organizations. In S. B. Sitkin, L. B. Cardinal, & K. Bijlsma-Frankema (Eds.). *Control in organizations: New directions in theory and research*. Cambridge, UK: Cambridge University Press.

- Lundin, R. A., & Söderholm, A. (1995). A theory of the temporary organization. *Scandinavian Journal of management*, 11(4), 437-455.
- Maier, M. W., Emery, D., & Hilliard, R. (2001). Software architecture: Introducing IEEE standard 1471. *Computer*, (4), 107-109.
- Markus, M. L. (2007). The governance of free/open source *software* projects: monolithic, multidimensional, or configurational? *Journal of Management & Governance*, 11(2), 151-163.
- Mayring, P. (2014). *Qualitative content analysis: theoretical foundation, basic procedures and software solution*.
- Mello, G. R. D., & Slomski, V. (2010). Electronic governance index of Brazilian states (2009): within the executive branch. *JISTEM ó Journal of Information Systems and Technology Management*, 7(2), 375-408.
- Merriam, S. (1998). *Qualitative research and case study applications in education*. San Francisco: Jossey-Bass.
- Mintzberg, H. (2011). Criando organizações eficazes: estrutura em cinco configurações. São Paulo: Atlas.
- Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). Toward a theory of stakeholder identification and salience: Defining the principle of who and what really counts. *Academy of management review*, 22(4), 853-886.
- Morgan, G., & Smircich, L. (1980). The case for qualitative research. *Academy of Management Review*, 5(4), 491-500.
- Morrow, S. (2005). Quality and trustworthiness in qualitative research in counseling psychology. *Journal of Counseling Psychology*, 52(2), 250-260.
- Mozzato, A. R., & Grzybovski, D. (2011). Análise de conteúdo como técnica de análise de dados qualitativos no campo da administração: potencial e desafios. *Revista de Administração Contemporânea*, 15(4), 731-747.
- O'Mahony, S., & Ferraro, F. (2007). The Emergence of Governance in an Open Source Community. *Academy of Management Journal*, 50(5), 1079-1106.
- Okhuysen, G. A., & Bechky, B. A. (2009). 10 coordination in organizations: An integrative perspective. *The Academy of Management Annals*, 3(1), p. 463-502.
- Oliveira, D. M. T. (2012). Amostra qualitativa e recrutamento. (2012). In Perdigão, D. M., Herlinger, M., & White, O. M. (Eds.). *Teoria e prática da pesquisa aplicada*. Elsevier. Brasil.
- Oliver, C. (1991). Strategic responses to institutional processes. *Academy of management review*, 16(1), 145-179.
- O'Mahony, S., & Lakhani, K. R. (2011). Organizations in the shadow of communities. In C. Marquis, M. Lounsbury, R. Greenwood (Eds.). *Communities and Organizations, Research in the Sociology of Organizations*. Bingley: Emerald Group Publishing Limited, 3-36.
- Orlikowski, W. J. (1991). Integrated information environment or matrix of control?: The contradictory implications of information technology.
- Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information systems research*, 2(1), 1-28.



- Ouchi, W. G. (1979). A Conceptual Framework for the Design of Organizational Control Mechanisms. *Management Science*, 25(9), 833-848.
- Pahlevan Sharif, S., & Kyid Yeoh, K. (2014). Independent directors' resource provision capability in publicly-listed companies in Malaysia.
- Paily, M. U. (2013). Creating constructivist learning environment: Role of " web 2.0" technology. In *International Forum of Teaching and Studies*. American Scholars Press, 9(1), 39.
- Panchal, J. H. (2010). Coordination in collective product innovation. *Proceedings of the International Mechanical Engineering Congress and Exposition (ASME)*, Vancouver, 333-346.
- Pfeffer, J. (1987). A resource dependence perspective on intercorporate relations. *Intercorporate relations: The structural analysis of business*, 25-55.
- Pfeffer, J., & Salancik, G. R. (2003). *The external control of organizations: A resource dependence perspective*. Stanford University Press.
- Pinto, M. S. L., & Oliveira, R. R. (2004). Estratégias competitivas no setor elétrico brasileiro: uma análise dos interesses e expectativas dos atores da Chesf. *Revista de Administração Contemporânea*, 8, 131 - 55.
- Popper, K. (2005). *The Logic of Scientific Discovery*. Routledge.
- Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3), 23-49.
- Richardson, R. J. (1999) Pesquisa Social: métodos e técnicas. São Paulo: Atlas.
- Robbins, S. P. (2010). *Organizational Behavior*. Pearson Education.
- Robbins, S. P., & Coulter, M. (2007). Managing change and innovation, *Management*, 384-415.
- Rodrigues, M. C. P. (2005). *Ação social das empresas privadas: como avaliar resultados? a metodologia EP2ASE*. FGV Editora.
- Rossetto & Rossetto, 2005 ROSSETTO, C. R.; ROSSETTO, A. M. Teoria institucional e dependência de recursos na adaptação organizacional: uma visão complementar. *RAE-eletrônica*, 4(1), 1-22.
- Rowley, T. J. (1997). Moving beyond dyadic ties: A network theory of stakeholder influences. *Academy of management Review*, 22(4), 887-910.
- Santos, C., Kuk, G., Kon, F., & Pearson, J (2013). The attraction of contributors in free and open source *software* projects. *Journal of Strategic Information Systems*, 22, 26-45.
- Schreyögg, G., & Sydow, J. (2010). Crossroads: organizing for fluidity? Dilemmas of new organizational forms. *Organization science*, 21(6), 1251-1262.
- Scott, W. R. (1995). *Institutions and organization*. Thousand Oaks: Sage Publications.
- Seidel, M. D. L., & Stewart, K. J. (2011). An initial description of the C-form. In C. Marquis, M. Lounsbury, & R. Greenwood (Eds.). *Communities and Organizations, Research in the Sociology of Organizations*. Bingley: Emerald Group Publishing Limited, 37-72.
- Seidel, V. P., & O'Mahony, S. (2014). Managing the Repertoire: Stories, Metaphors, Prototypes, and Concept Coherence in Product Innovation. *Organization Science*, 25(3), 691-712.

- Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source *software* development. *Management Science*, 52(7), 1000-1014.
- Shah, S. K., & Gorbatai, A. D. (2015). Structural sampling: A technique for illuminating social systems. *Handbook of qualitative organizational research*, p. 251-261.
- Shaikh, M., & Vaast, E. (2016). Folding and unfolding: balancing openness and transparency in open source communities. *Information Systems Research*, 27(4), 813-833.
- Sharma, S., Sugumaran, V., & Rajagopalan, B. (2002). A framework for creating hybrid open source software communities. *Information Systems Journal*, 12(1), p. 7-25.
- Shenhar, A. J. (2001) Contingent management in temporary, dynamic organizations: The comparative analysis of projects. *The Journal of High Technology Management Research*, 12(2): 239-271.
- Simeray, J. P. (1978). *A estrutura da empresa*. Rio de Janeiro: LTC.
- Sitkin, S. B., Cardinal, L. B., & Bijlsma-Frankema, K. (2010). *Organizational Control*. Cambridge: Cambridge University Press.
- Snow, C. C., Fjeldstad, Ø. D., & Lettl, C. (2011). Organizing continuous product development and commercialization: The collaborative community of firms model. *Journal of Product Innovation Management*, 28(1): 3-16.
- Snow, C. C., Miles, R. E., & Coleman Jr, H. J. (1992). Managing 21st century network organizations. *Organizational dynamics*, 20(3), 5-20.
- Stewart, D. (2005). Social status in an open-source community. *American Sociological Review*, 70(5), 823-842.
- Stewart, K. J., & Gosain, S. (2006). The moderating role of development stage in free/open source *software* project performance. *Software Process: Improvement and Practice*, 11(2), 177-191.
- Sutanto, J., Kankanhalli, A., & Tan, B. C. (2011). Deriving IT-mediated task coordination portfolios for global virtual teams. *IEEE Transactions on Professional Communication*, 54(2), 133-151.
- Sydow, J., Lindkvist, L., & DeFillippi, R. (2004). Project-based organizations, embeddedness and repositories of knowledge: Editorial. *Organization Studies*, 25(9), 1475-1489.
- Tadelis, S. (2010). A Tribute to Oliver Williamson: Williamson's Contribution and its Relevance to 21st Century Capitalism. *California Management Review*, 52(2), 159-166.
- Takhar-Lail, A., & Chitakunye, P. (2015). Reflexive introspection: Methodological insights from four ethnographic studies. *Journal of Business Research*, 68(11), 2383-2394.
- Theurl, T. (2005). From corporate to cooperative governance. *Economics of Interfirm Networks, Tübingen*, 149-192.
- Tran, B. (2014). Triangulation in Organizational Research: Validating Knowledge. In Takhar-Lail, A., & Ghorbani, A. *Market Research Methodologies: Multi-Method and Qualitative Approaches: Multi-Method and Qualitative Approaches*, 93.
- Van Maanen, J. (2010). *Identity work and control in occupational communities*. Cambridge University Press.
- Vasconcellos, E., & Hemsley, J. R. (2002). *Estrutura das organizações*. São Paulo: Thomson Pioneira Learning.

- Vilariño, J. P., & Schoenherr, R. A. (1987). Racionalidad y control en las organizaciones complejas. *Reis*, 39, 119-139.
- Vincent, C., and Camp, J. (2004). Looking to the Internet for models of governance, *Ethics and Information Technology*, 6(3), p. 161-173.
- Von Hippel, E. (2007). The sources of innovation. In *Das Summa Summarum des Management*. Gabler, 111-120.
- Von Hippel, E., & Von Krogh, G. (2003). Open source *software* and the 'private-collective' innovation model: Issues for organization science. *Organization Science*, 14(2), 209-223.
- Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source *software* innovation: a case study. *Research Policy*, 32(7), 1217-1241.
- Wenger, E. (1998). *Communities of practice. Learning, meaning and identity*. Cambridge: Cambridge University Press.
- Wenger, E. (2000). Communities of practice and social learning systems. *Organization*, 7(2), 225-246.
- Wenger, E. (2011). *Communities of practice: A brief introduction*.
- Wengraf, T. (2001). *Qualitative research interviewing: Biographic narrative and semi-structured methods*. Sage.
- West, J., & O'Mahony, S. (2005). Contrasting community building in sponsored and community founded open source projects. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, Hawaii, 196c.
- Williamson, O. E. (1985). *The economic institutions of capitalism: firms, markets, relational contracting*. New York: Free Press.
- Wynn, D. E. (2003). Organizational Structure of Open Source Projects: A Life Cycle Approach. *Abstract for 7<sup>th</sup> Annual Conference of the Southern Association for Information Systems*, Georgia.
- Xu, B., Xu, Y., & Lin, Z. (2005). Control in open source *software* development. *Proceedings of the Americas Conference on Information Systems (AMCIS)*, Omaha, 433.
- Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation Open Source *Software* developers. In *Proceedings of the 25th international conference on software engineering* (pp. 419-429). IEEE Computer Society.
- Yin, R. K. (2010) *Estudo de caso: planejamento e métodos*. Porto Alegre: Bookman.
- Zabusky, S. E. (1997). Computers, clients, and expertise: Negotiating identities in a nontechnical world. S. R. Barley, J. E. Orr, (Eds.). *Between Craft and Science: Technical Work in U.S. Settings*. ILR Press, Ithaca, NY, 129-153.
- Zammuto, R. F., T. L, Griffith, A., Majchrzak, D. J., & Dougherty, S. F. (2007). Information Technology and the Changing Fabric of Organization. *Organization Science*, 18(5), 749-762.
- Zhan, Y., Bai, Y., & Liu, Z. (2007). Virtual Team Governance: Addressing the Governance Mechanisms and Virtual Team Performance. In W. Wang, Y. Li, Z. Duan, L. Yan, H. Li, and X. Yang. (Ed). *Integration and Innovation Orient to E-Society*. Boston: Springer, 282-288.
- Zheng, Y., Zhao, K., & Stylianou, A. (2013). The impacts of information quality and system quality on users' continuance intention in information-exchange virtual communities: An empirical investigation. *Decision Support Systems*, 56, 513-524.

Zylbersztajn, D. (2005). Papel dos contratos na coordenação agro-industrial: um olhar além dos mercados. *Revista de Economia e Sociologia Rural*, 43(3), 385-420.

## APÊNDICE A 6 CARACTERÍSTICAS GERAIS DO PROJETO E ROTEIRO DE ENTREVISTA

### 1 IDENTIFICAÇÃO DO PROJETO

1.1 Nome do projeto: \_\_\_\_\_

1.2 Ano que surgiu o projeto: \_\_\_\_\_

1.3 Número de *committers*: \_\_\_\_\_

1.4 O projeto possui fundação em sua gestão?

( ) Sim. Especificar:

\_\_\_\_\_

( ) Não

1.5 O projeto possui patrocinador(es) ou alguma outra organização envolvida em sua gestão?

( ) Sim. Especificar:

\_\_\_\_\_

( ) Não

### 2 RECURSOS DE TIC USADOS PELO PROJETO

2.1 Quais recursos de tecnologia da informação e comunicação (TIC) o projeto utiliza em seu funcionamento?

( ) Listas de *e-mail*

( ) Fórum de discussão eletrônico

( ) *Wikis*

( ) *Blogs*

( ) *Chat* / Mensagens instantâneas

( ) Audioconferência

( ) Videoconferência

( ) *Home page*

( ) Plataforma para desenvolvimento. Qual(is)? ( ) *GitHub*

( ) *GitLab*

( ) Outra(s). Especificar:

\_\_\_\_\_

( ) Redes Sociais. Qual(is)? ( ) *Facebook*

( ) *Twitter*

( ) Outra(s). Especificar:

\_\_\_\_\_

( ) Outra(s) tecnologia(s).

Especificar: \_\_\_\_\_

### 3 IDENTIFICAÇÃO DO ENTREVISTADO

3.1 Data: \_\_/\_\_/\_\_\_\_

3.2 Entrevista N°: \_\_\_\_\_

3.3 Gênero: ( ) Feminino ( ) Masculino

3.4 Tempo em que atua na comunidade: \_\_\_\_\_ ano(s) \_\_\_\_\_ meses

3.5 Função(ões) ocupada(s) na  
comunidade: \_\_\_\_\_

<b>Questões para entrevista com gestores ou desenvolvedores do grupo central do projeto</b>
1- Como o projeto surgiu? Fale sobre a história e as principais transformações pelas quais o projeto passou desde que surgiu até o momento atual.
2- No caso de existência de fundações, patrocinadores, ou outras organizações externas, como essas organizações interagem com o projeto e influenciam no seu funcionamento?
3- Como acontece a sustentação do projeto em relação aos recursos e à infraestrutura necessária ao seu funcionamento?
4- Existe alguma hierarquia entre os membros que atuam no projeto?
5- Como acontece a divisão de papéis entre os membros do projeto?
6- Como são tomadas as decisões no projeto?
7- O projeto possui regras, normas ou outros procedimentos que determinem a forma como os integrantes devem realizar as suas atividades? Especifique esses procedimentos.
8- De que forma ocorre a definição de objetivos e metas do projeto?
9- Como acontece a entrada e a saída de um membro do grupo central do projeto?
10- Os integrantes do projeto costumam realizar reuniões de trabalho e eventos (presenciais ou virtuais)? Como esses encontros acontecem?
11- De que maneira o projeto promove o reconhecimento do trabalho realizado pelos integrantes?
12- Como é realizado o controle da qualidade do código produzido no projeto (identificação de erros ou necessidades de melhorias do <i>software</i> )?
13- De que forma é percebida a autonomia na realização do trabalho no projeto?

**APÊNDICE B ó E-MAIL SOLICITANDO INDICAÇÃO DE PROJETO PARA  
ESTUDO DE CASO**

Prezado (a) Professor (a),

Sou Isabela Neves Ferraz, aluna de doutorado do Programa de Pós-Graduação em Administração da Universidade de Brasília (PPGA/UnB). A minha tese tem a orientação do Professor Dr. Carlos Denner.

O tema do trabalho que estou desenvolvendo refere-se ao uso de mecanismos de governança em projetos de *software* livre comunitários. Encontro-me, atualmente, em fase de pesquisa de campo, necessitando de informações que me permitam compreender como ocorre a governança desses projetos.

Diante do exposto, gostaria de saber se você poderia me indicar casos nacionais de sucesso e competitivos de projetos de *software* livre comunitários, bem como os respectivos contatos de *e-mail* dos responsáveis pelo projeto indicado?

Ficarei extremamente grata se receber o seu valioso apoio. Agradecendo antecipadamente a sua atenção, fico no aguardo de alguma resposta.

Cordialmente,

Isabela Neves Ferraz

**APÊNDICE C ó E-MAIL PARA AGENDAMENTO DAS ENTREVISTAS**

Prezado (a),

Sou Isabela Neves Ferraz, aluna de doutorado do Programa de Pós-Graduação em Administração da Universidade de Brasília (PPGA/UnB).

A minha tese tem como temática a governança de projetos de *software* livre comunitários. Esse trabalho tem a orientação do Prof. Dr. Carlos Denner.

Atualmente, estou em fase de coleta de dados, por meio da aplicação do meu roteiro de entrevista. Para isso, preciso conversar com responsáveis/participantes de comunidades de *software* livre que sejam competitivas no cenário nacional, como é o caso do projeto [nome do projeto]. Sendo assim, gostaria de saber se você teria disponibilidade para agendarmos uma conversa para falar sobre a história e a governança desse projeto?

Podemos ter essa conversa por Google *Hangouts* ou por *Skype*, na data e horário de acordo com a sua conveniência. A conversa teria aproximadamente 40 minutos. Fico no aguardo do seu retorno, e, desde já, agradeço bastante a sua atenção!

Cordialmente,

Isabela Neves Ferraz



## APÊNDICE D ó FORMULÁRIO ENCAMINHADO PARA INTEGRANTES DO PROJETO APACHE OPENNLP

### Governance in free software projects

The form has thirteen open questions. I emphasize that the information provided will be used only for academic purposes and the respondent will not be identified in the survey. You can put links that lead to examples of your answer. Feel free to write everything you think.

Thank you so much for your help.

Isabela Ferraz

OBS: If you prefer, you can answer the questions in your native language.

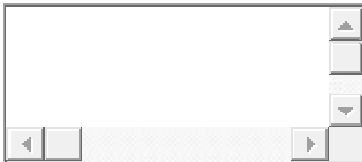
\*Obrigatório

E-mail address\*

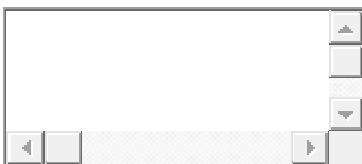
1) How did your entry in the project happen?

An empty text input field with a light gray background and a thin border. It has a vertical scrollbar on the right side and a horizontal scrollbar at the bottom.

2) What are the advantages and disadvantages of the project for being part of the Apache Foundation?

An empty text input field with a light gray background and a thin border. It has a vertical scrollbar on the right side and a horizontal scrollbar at the bottom.

3) What were the main changes that happened to the project after it entered the Apache Foundation?

An empty text input field with a light gray background and a thin border. It has a vertical scrollbar on the right side and a horizontal scrollbar at the bottom.

4) How do you perceive the existence of hierarchy in the project (i.e.: who reports to whom, hierarchy based on experience etc.)?

5) What are the main roles identified in the project and how are people chosen to play these roles?

6) Do you perceive that there is leadership in the project? Explain how this happens in the project.

7) How are decisions made in the project? How does Apache Foundation influence these decisions?

8) Does the project have rules, standards or procedures that regulate its activities? Describe the main procedures.

9) How does the project define its goals and objectives?

10) How do project work meetings happen and what subjects are discussed at these meetings?

11) How is the quality control of the code that enters the repository accomplished? Explain how this happens in the project.

12) Do you perceive that you have autonomy or independence in carrying out activities on the project? Explain how this happens in the project.

13) What advantages and disadvantages do you realize from being part of the project?