

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UM MODELO PARA APOIAR A ESTRUTURAÇÃO DO
CONHECIMENTO NAS INTERAÇÕES EM COMUNIDADES
VIRTUAIS DE *SOFTWARE* LIVRE

Licia de Cassia Nascimento Pinto

Orientadora
Flávia Maria Santoro

RIO DE JANEIRO, RJ – BRASIL
Setembro de 2009

UM MODELO PARA APOIAR A ESTRUTURAÇÃO DO CONHECIMENTO NAS
INTERAÇÕES EM COMUNIDADES VIRTUAIS DE *SOFTWARE* LIVRE

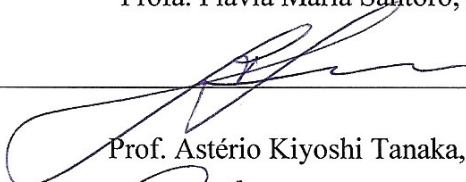
Licia de Cassia Nascimento Pinto

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA
OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-
GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO
DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO
EXAMINADORA ABAIXO ASSINADA.

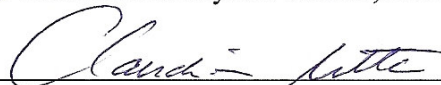
Aprovada por:



Profa. Flávia Maria Santoro, D.Sc. – UNIRIO



Prof. Astério Kiyoshi Tanaka, Ph.D. – UNIRIO



Profa. Claudia Lage Rebello da Motta, D.Sc. – PPGI e NCE/UFRJ

RIO DE JANEIRO, RJ – BRASIL

Setembro de 2009

P659 Pinto, Licia de Cassia Nascimento.
Um modelo para apoiar a estruturação do conhecimento nas interações em comunidades virtuais de software livre / Licia de Cassia Nascimento Pinto, 2009.
164f.

Orientador: Flávia Maria Santoro.
Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

1. Software livre. 2. Comunidades virtuais de software livre. 3. Gestão do conhecimento. 4. Fórum (Debates). Modelo de argumentação (Informática). 6. Interação (Informática). I. Santoro, Flávia Maria. II. Universidade Federal do Estado do Rio de Janeiro (2003-). Centro de Ciências Exatas e Tecnologia. Curso de Mestrado em Informática. III. Título.

CDD 005.3

*Ao meu marido Alexandre e ao meu filho Bruno,
que sempre estiveram comigo
para a realização deste sonho.*

Agradecimentos

Agradeço a Deus pela força e coragem para enfrentar este grande desafio que é fazer uma dissertação e, ainda, ser mãe e esposa.

Escrever esta dissertação me proporcionou realizar mais um sonho, obter um aprendizado de grande valor, gerar um conhecimento enorme e entender que na vida tudo é possível desde que se tenha dedicação, disciplina e perseverança. Desenvolver uma dissertação é um trabalho muito árduo, mas não impossível. Vale a pena o esforço!

Agradeço a minha orientadora, Flávia Maria Santoro que me orientou e me apoiou em todos os momentos, mostrando-me os caminhos válidos e como superar as dificuldades existentes, sendo amiga e motivadora. Ela me deu a oportunidade de integrar o grupo de pesquisa NP2TEC, de trabalhar num projeto de pesquisa sobre SIMULAÇÃO DE PROCESSOS na Petrobras, juntamente com um grupo competente de professores e profissionais do CEFET, no projeto de pesquisa COLMÉIA com a COPPE/UFRJ, e ainda me orientou na bolsa REUNI. Quantas oportunidades maravilhosas e projetos acadêmicos de pesquisa foram realizados. Obrigada por tudo!

Também agradeço à participação do Prof. Astério Kiyoshi Tanaka nos seminários, contribuindo com sugestões e ponderações importantes e participando da banca.

Agradeço a Profa. Claudia Lage Rebello da Motta por participar da banca e contribuir com suas sugestões.

Aos professores do PPGI pela intensa dedicação à pesquisa e pelas direções de grande valia dadas durante os seminários.

Aos amigos do mestrado pelas conversas e comentários enriquecedores. Obrigada pela força e companheirismo: Rosária Viana, Betânia Oliveira, Leonardo Gatti, Hesley Py, Marcelo Monsores, João Gonçalves e todos os outros que convivi.

À amiga Taísa Alves por me escutar, dividir angústias e trocar “figurinhas”.

Uma lembrança à amiga Thalita Moraes que não está mais entre nós, mas foi uma companheira e amiga durante minha trajetória no mestrado.

Ao novo companheiro de projeto, Tiago Nicola Veloso, por me apoiar e ter compromisso para realização desse trabalho.

Aos técnicos administrativos da secretaria, Alessandra Nascimento e Douglas Brito, pela agilidade na resolução dos pedidos e na obtenção de documentos. Além das conversas com a Alessandra, as quais me fizeram ganhar uma amiga.

Ao grupo de pesquisa NP2TEC, Profa. Renata Araújo, Profa Fernanda Baião, Claudia Cappelli, Andrea Magalhães, Vanessa Nunes, Ercília Moreira, Hadeliane Iendrike e os demais participantes.

À Comunidade VivaOLinux pelo apoio e participação no trabalho. Confirmando que a colaboração é tudo.

À amiga Patrícia Alves pelas revisões e presteza em me ajudar.

A todos os meus amigos que me apoiaram e se preocuparam comigo.

À Profa. Marinilza Bruno de Carvalho pelo incentivo.

À minha grande amiga, companheira e comadre Maria Teresa Gouvêa, que me incentivou a fazer o mestrado e sempre esteve preocupada em me ajudar. Obrigada por tudo!

À minha mãe, Leila Regina, por ajudar a cuidar do meu filho Bruno, nessa fase tão atarefada da minha vida.

Ao meu enteado, Lucas de Almeida, pela ajuda nas tarefas em casa.

Ao meu pequeno filho, Bruno Pinto de Almeida, por entender que a mamãe tinha de estudar.

Ao meu marido, Alexandre de Almeida, que me apoiou, incentivou-me, patrocinou esse sonho e ouviu minhas preocupações. Muito obrigada pela força e companheirismo.

NASCIMENTO, Licia de Cassia. **Um modelo para apoiar a estruturação do conhecimento nas interações em Comunidades Virtuais de *Software* Livre.** UNIRIO, 2009. 164 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

O conhecimento é atualmente o maior bem compartilhado entre integrantes de Comunidades Virtuais de *Software* Livre, onde para resolver problemas individuais de desenvolvimento de *software*, os participantes criam soluções, geram novos conhecimentos e permitem seu reuso. As Comunidades Virtuais de *Software* Livre são comunidades que particularmente atuam de forma colaborativa, diferenciando-se das Comunidades de Prática comuns, porque além do interesse pelo mesmo tema, constroem colaborativamente um produto ou serviço de *software*, utilizando-se das interações realizadas para se comunicarem e ultrapassarem as distâncias geográficas e depois divulgando para toda a sociedade usufruir, zelando assim, pela liberdade do *software*. Para que esse conhecimento não se perca com o tempo, é proposto um modelo de argumentação que pretende viabilizar a estruturação do conhecimento ressaltando as atividades do ciclo da Gestão do Conhecimento: captura, representação, armazenamento, recuperação e reutilização do conhecimento nas interações de forma organizada. Para a avaliação deste modelo foi desenvolvido um fórum estruturado, organizando os artefatos disponibilizados e apoiando os participantes na busca de conhecimento. Concluiu-se com a realização de Estudos de Caso que a interação estruturada apoia a discussão e facilita o acesso ao conhecimento gerado.

Palavras-chave: Conhecimento, Gestão do Conhecimento, Comunidades Virtuais de *Software* Livre, *Software* Livre, Interação, Modelo de Argumentação, Fórum.

ABSTRACT

Knowledge is now one of the largest values for the members of the Free Software Virtual Communities, which to solve individual problems of software development by creating solutions and generating new knowledge while allowing its reuse. The Free Software Virtual Communities are particularly collaborative, differing from the ordinary Communities of Practice, since beside their interest in the same issues, their members actually build software products or services using the interactions in order to communicate and overcome geographical distances, and then spreading them to the whole society benefit, ensuring the freedom of software. For this knowledge not to become lost over time, we propose an argumentation model in order to structure knowledge, highlighting the knowledge management activities cycle: capture, representation, storage, retrieval and reuse knowledge within the interactions. To evaluate this model a structured discussion forum was developed, organizing the artifacts available and supporting the participants later in the knowledge search. Case studies performed concluded that structured interaction supports the discussion and facilitates access to knowledge generated.

Keywords: Knowledge, Knowledge Management, Free Software Virtual Communities, Free Software, Interaction, Argumentation Model, Forum.

Sumário

Capítulo 1 – Introdução.....	14
1.1. Motivação e Justificativas.....	15
1.2. Caracterização do Problema	17
1.3. Hipótese da pesquisa	18
1.4. Objetivos da dissertação	18
1.5. Metodologia de pesquisa	19
1.6. Estruturação da dissertação.....	20
Capítulo 2 - Comunidades Virtuais de Software Livre.....	22
2.1. Conceito de <i>Software</i> Livre	23
2.2. O <i>Software</i> Livre no Brasil.....	26
2.3. O processo de desenvolvimento de <i>Software</i> Livre	28
2.4. Comunidades Virtuais de Software Livre	37
Capítulo 3 - Gestão do Conhecimento em Comunidades Virtuais de Software Livre ...	42
3.1. Gestão do Conhecimento.....	43
3.2. Gestão do Conhecimento na Engenharia de <i>Software</i>	46
3.3. Gestão do Conhecimento nas Comunidades de Prática	49
3.4. Gestão do Conhecimento nas Comunidades de <i>Software</i> Livre	51
Capitulo 4 - Estudo Exploratório: avaliação das comunidades	55
4.1. Metodologia da Pesquisa	55
4.2. Análise das Comunidades Virtuais de Software Livre.....	56
4.2.1 Identificação de Comunidades	56

4.2.2	Acesso às ferramentas usadas pelas comunidades	57
4.2.3	Inserções nas comunidades	58
4.2.4	Definição de categorias de análise.....	59
4.2.5	Avaliação das comunidades: Resultados obtidos.....	62
4.3.	<i>Survey</i> : Levantamento por meio de Questionário	66
4.3.1	Interpretação dos resultados	79
Capítulo 5 – Modelos de argumentação.....		83
5.1.	Conceitos básicos	84
5.2.	Modelo IBIS	84
5.3.	Modelo GD-IBIS.....	86
5.4.	Modelo de Esquemas de Toulmin.....	87
5.5.	Modelo ColaOnto.....	88
5.6.	Quadro Comparativo dos Modelos de Argumentação	90
Capítulo 6 – Um modelo de estruturação de conhecimento em interações nas Comunidades Virtuais de <i>Software</i> Livre		92
6.1.	Modelo Proposto	93
6.2.	Diagrama de classes	93
6.3.	ArFoLi (Argumentação para Fórum de <i>Software</i> Livre): Modelo de Argumentação.....	97
6.4.	Quadro comparativo dos Modelos de Argumentação e o Modelo ArFoLi	102
6.5.	Análise de dois Cenários de Aplicação do Modelo	103
Capítulo 7 – Fórum Estruturado para Comunidades Virtuais de <i>Software</i> Livre - Fórum Livre		113
7.1.	Fórum estruturado	114
7.2.	Características técnicas do fórum estruturado	115
7.3.	Funcionalidades do fórum estruturado	116
Capítulo 8 – Estudos de Caso.....		123
8.1.	Estudo de caso.....	124

8.2.	Descrição dos Estudos de Caso.....	125
8.2.1.	Caso 1 – Grupo de alunos da UNIRIO envolvidos com <i>Software Livre</i>	125
8.2.2.	Caso 2 – Comunidade VivaOLinux	130
8.3.	Conclusão geral dos Estudos de Caso	133
8.4.	Limitações dos Estudos de Caso	133
	Capítulo 9 – Conclusões e Trabalhos Futuros	135
9.1.	Pesquisa realizada	136
9.2.	Considerações Finais.....	137
9.3.	Contribuições	138
9.4.	Trabalhos Futuros.....	138
	Referências Bibliográficas	140
	Apêndice A – Questionário 1	149
	Apêndice B – Requisitos Funcionais do Fórum Livre.....	154
	Apêndice C – Casos de Uso	156
	Apêndice D – Questionário 2	160
	Apêndice E – Telas do Fórum Livre.....	162

Lista de Figuras

Figura 1 - Apresentação da dissertação em capítulos	21
Figura 2 - Espiral do conhecimento (NONAKA e TAKEUCHI, 1997; NONAKA, TOYAMA e HIRATA, 2008)	44
Figura 3 - Modelo “rosca” de Gestão do Conhecimento (WENGER, 2004).....	50
Figura 4 – Gráfico Regiões do Brasil	67
Figura 5 – Gráfico Formação dos Participantes	68
Figura 6 – Gráfico Papéis na comunidade	69
Figura 7 – Gráfico Formas de interação.....	70
Figura 8 – Gráfico Tipo de material compartilhado nas comunidades.....	71
Figura 9 – Gráfico Quantidade de material compartilhado.....	72
Figura 10 – Gráfico Repositório dos materiais das comunidades	73
Figura 11 – Gráfico Busca de informação	74
Figura 12 – Gráfico Tipos de mensagem durante as interações.....	75
Figura 13 – Gráfico Respostas das questões debatidas.....	76
Figura 14 – Gráfico Busca por assunto.....	77
Figura 15 – Gráfico Indicação de material na comunidade	78
Figura 16 – Gráfico Confiança nas informações.....	79
Figura 17 – Modelo IBIS (RITTEL e Kunz, 1970).....	85
Figura 18 – Modelo GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002)	86
Figura 19 - Modelo Toulmin (TOULMIN, 2009)	88
Figura 20 – Modelo ColaOnto (MALEEWONG <i>et al.</i> , 2008).....	89
Figura 21 - Modelo Conceitual.....	94
Figura 22 - Modelo ArFoLi.....	100
Figura 23 - Comunidade Lussumo	115
Figura 24 - Estruturação do Fórum Livre	118
Figura 25 - Tela de discussões.....	118
Figura 26 - Tela das categorias.....	119
Figura 27 - Tela de uma discussão	120
Figura 28 – Tela com resposta, nova questão e argumento	120
Figura 29 - Tela de encerramento da discussão.....	121
Figura 30 – Tela de discussão (resposta, nova questão e argumento)	127
Figura 31 – Estruturação da questão sobre navegadores para desenvolvimento web. .	131
Figura 32 – Tela do Fórum Livre de uma questão.....	162
Figura 33 – Tela do Fórum Livre com resposta, nova questão e pontuação.....	163
Figura 34 – Tela do Fórum Livre com argumento, resposta e pontuação	163
Figura 35- Tela do Fórum Livre com resposta, pontuação e encerramento.....	164

Lista de Tabelas

Tabela 1 - Comunidades Virtuais de Software Livre	40
Tabela 2 - Análise das Comunidades Virtuais de <i>Software</i> Livre	65
Tabela 3 – Comparação dos Modelos de Argumentação.....	90
Tabela 4 – Comparação entre os Modelos de Argumentação e o Modelo ArFoLi.....	102

Capítulo 1 – Introdução

Neste capítulo é apresentada a pesquisa realizada nesta dissertação, discutindo a motivação e as justificativas, o problema, a hipótese de pesquisa, os objetivos da dissertação, o método adotado e a estruturação da dissertação.

1.1. Motivação e Justificativas

Atualmente, o *Software Livre* não tem mais o estereótipo de anarquia ou de trabalho realizado por *hackers*, mas sim de projetos de sucesso, apoiados por comunidades virtuais que os desenvolvem. O *Software Livre* se baseia nos princípios de liberdade para o usuário: usar, copiar, estudar, modificar e redistribuir código fonte, (FSF, 2008). A liberdade é o foco principal para que as comunidades existam e tenham como objetivo promover uma tecnologia livre específica.

As Comunidades Virtuais de *Software Livre* podem ser caracterizadas como Comunidades de Prática (WENGER, 2006) que realizam a divulgação e o uso de ferramentas de *Software Livre*, disseminam o conhecimento do *Software Livre* e apoiam o desenvolvimento de projetos de *Software Livre*. Ao contrário das Comunidades de Prática convencionais que focam em um assunto específico, as Comunidades Virtuais de *Software Livre* trabalham colaborativamente na geração de produtos de *software*, e agregam valor ao conhecimento gerado e disseminado por seus participantes.

O *Software Livre* propicia uma forma de trabalhar diferente da tradicional, em que existe transposição de barreiras geográficas, flexibilização de horários, criação de novos modelos de processos de trabalho e atendimento a diversos ramos de negócios. Os *softwares* proprietários e o movimento de *Software Livre* diferem quanto ao foco, de acordo com TAURION (2004), o *software* proprietário visa a atender a demanda dos clientes e também a um negócio específico, enquanto o movimento de *Software Livre* visa a atender a sociedade, ou seja, qualquer pessoa ou empresa que tenha acesso ao código publicado.

O processo de desenvolvimento de *Software Livre* envolve o voluntariado e a criação de comunidades, tendo como premissas a colaboração, a contínua evolução do código fonte e a melhoria do *software*. Segundo REIS (2003), “É curioso que possa ter sucesso mundial um modelo de desenvolvimento aparentemente fundamentado no trabalho de amadores e voluntários, coordenados de maneira pouco formal e com a utilização de ferramentas extremamente simples. Mais surpreendente é a percepção informal entre a comunidade e os meios de comunicação de que o *software* produzido por estes projetos tem qualidade”.

A compreensão do fenômeno de *Software Livre* vai além das discussões sobre disponibilidade de código fonte ou formas de financiamento. O papel da comunidade de

colaboradores, as formas de colaboração e a recompensa pessoal criam um contexto complexo. Um dos manifestos de maior impacto relacionados à mudança de paradigma de desenvolvimento em comunidades de *Software Livre* trata de sua caracterização como um processo horizontal, coletivo, colaborativo e em rede, tendo a Internet como ponto chave de interação.

Este modelo de trabalho, conhecido como o “modelo bazar” (RAYMOND, 2001), proclama a descentralização do planejamento e a execução do desenvolvimento por comunidades heterogêneas, como vínculos fracos entre seus membros e dotadas de informalidade. “Para construir uma comunidade de desenvolvimento, você precisa atrair pessoas, fazer com que se interessem pelo que você está fazendo, e mantê-las alegres sobre a quantidade de trabalho que estão fazendo. O entusiasmo técnico constitui uma boa parte para atingir isto, mas está longe de ser toda história. A personalidade que você projeta também importa.” (RAYMOND, 1998).

Segundo REIS (2003), “Projetos de *Software Livre* são organizações virtuais formadas por indivíduos que trabalham juntos no desenvolvimento de um *Software Livre* específico. Estes indivíduos trabalham geograficamente dispersos, utilizando ferramentas simples para coordenar e comunicar seu trabalho através da Internet.” As Comunidades Virtuais de *Software Livre* se disseminam e trazem consigo a força da descentralização, apesar da distância geográfica, das diferentes agendas dos voluntariados, da colaboração, do uso de repositórios de desenvolvimento de projetos, da disseminação do conhecimento e do desenvolvimento tecnológico.

Desta forma, o reuso do conhecimento é fundamental para as Comunidades Virtuais de *Software Livre*. Segundo VON KROGH, SPAETH e HAEFLIGER (2005), o reuso acontece em projetos de desenvolvimento de *Software Livre* por meio de compartilhamento de linhas de código fonte, componentes e métodos. Os autores identificaram esta questão através de entrevistas realizadas com desenvolvedores de 15 projetos com diferentes objetivos.

RUDZI e JONSON (2003) analisaram em projetos bem sucedidos desenvolvidos por Comunidades Virtuais de *Software Livre* (Apache, GCC, Gnome, KDE, *kernel* do Linux e Mozilla) o uso de práticas (tais como, FAQs, Web Sites, *Howtos*, IRC, Sistemas de Monitoramento de Problemas, *Mailing Lists*, Weblogs e *Wikis*), de acordo com os aspectos (acurácia e valor, dinâmica, validade, concorrência, confidencialidade, acessibilidade e papéis, categorização e mapeamento, busca e localização da *expertise*),

com a visão da comunidade (as quatro formas de criação de conhecimento: socialização, externalização, internalização e combinação) e com as necessidades suportadas por práticas pela Engenharia de *Software*. Esses autores concluíram que as práticas utilizadas pelas comunidades criam um ambiente de desenvolvimento para os projetos, onde podem ser observados modos de conversão do conhecimento e que, de maneira geral, suportam a criação e o compartilhamento do conhecimento.

Desta forma, a aplicação dos princípios da Gestão do Conhecimento se torna bastante relevante no contexto das Comunidades Virtuais de *Software* Livre. Uma das questões importantes é a forma como o conhecimento é compartilhado por meio das interações. Observa-se que não existe uma preocupação com a organização desse tipo de conhecimento de forma a facilitar o seu uso por parte dos membros das comunidades.

Essa dissertação foi motivada pela participação no projeto COLMÉIA, uma parceria entre o PESC-COPPE/UFRJ e o PPGI-CCET/UNIRIO (www.uniriotec.br/padct). O foco deste projeto foi a caracterização do processo de desenvolvimento de *software* nas Comunidades Virtuais de *Software* Livre. O desafio foi buscar abordagens para ampliar as vantagens e a cultura de trabalho preconizado pelos processos de desenvolvimento de *Software* Livre, por meio da compreensão de sua colaboração intrínseca, da proposta de ferramental de apoio à sua realização eficaz. As áreas enfatizadas foram: colaboração, reutilização, gerência de configuração de *software* e experimentação e a evolução de projetos. Dentre os objetivos específicos do projeto, encontrava-se a compreensão dos aspectos de colaboração e Gestão do Conhecimento existentes no processo de desenvolvimento de *Software* Livre, bem como a especificação e o desenvolvimento de protótipos de ferramentas de apoio, em que esta dissertação está inserida.

1.2. Caracterização do Problema

Pesquisas apontadas anteriormente mostram que o conhecimento é o ponto central das comunidades de *Software* Livre e os seus membros participam ativamente nesse processo de troca de conhecimento. Desta forma, mecanismos de suporte à Gestão do Conhecimento se tornam fundamentais para apoiar os participantes de comunidades a acessarem o conhecimento.

Neste contexto, surgem as seguintes questões: o conhecimento compartilhado pelos membros das Comunidades Virtuais de *Software* Livre é acessado com facilidade? A identificação de informações úteis que apoiem os desenvolvedores fica explícita? Os participantes conseguem sanar suas dúvidas com facilidade ao interagirem no desenvolvimento de projetos? As ferramentas de interação facilitam o acesso ao conhecimento desejado?

Para detalhar as questões relacionadas ao problema foi realizado um *Survey* nas Comunidades Virtuais de *Software* Livre, cujos resultados estão descritos no Capítulo 4. Então, o problema a ser tratado nesta dissertação é:

Como facilitar o acesso ao conhecimento gerado nas interações entre os participantes de Comunidades Virtuais de *Software* Livre?

1.3. Hipótese da pesquisa

Baseada em resultados de pesquisas realizadas nesta dissertação, por meio de Estudo Exploratório e *Survey*, a hipótese enunciada foi: “Se o conteúdo das interações entre os participantes for estruturado segundo um modelo, então é possível facilitar o acesso ao conhecimento compartilhado nas interações entre os participantes no contexto das comunidades”.

Com isso, focando no problema apresentado, é proposto um modelo de estruturação no qual o conhecimento esteja acessível aos participantes, ou seja, as informações das interações ficam estruturadas de forma que o conhecimento gerado possa ser acessado com facilidade.

1.4. Objetivos da dissertação

O objetivo da pesquisa foi analisar como as Comunidades Virtuais de *Software* Livre geram conhecimento, e desenvolver um modelo de estruturação do conhecimento gerado por meio de interações, em que esse conhecimento esteja acessível aos participantes da comunidade. Para isso os seguintes objetivos específicos foram estipulados:

- Realização de pesquisa exploratória para descobrir as questões que necessitam ser tratadas;
- Especificação de modelo de argumentação para organizar as interações;

- Implementação do modelo;
- Avaliação do modelo.

1.5. Metodologia de pesquisa

De acordo com RUDIO (2002), o método é “o caminho a ser percorrido, demarcado, do começo ao fim, por fases ou etapas”. A pesquisa realizada nesta dissertação foi iniciada com uma busca bibliográfica para entender os conceitos do *Software Livre* e como a Gestão do Conhecimento poderia apoiar as necessidades das Comunidades Virtuais de *Software Livre*. Depois, foi necessário entender como as comunidades funcionavam por meio de um Estudo Exploratório e ainda obter a visão dos participantes dessas comunidades, suas necessidades.

Com base no Estudo Exploratório, foram definidos os requisitos para um ambiente de suporte à estruturação do conhecimento e desenvolvido um modelo conceitual e um modelo de argumentação. Uma aplicação de fórum foi implementada para viabilizar a avaliação desta proposta. A avaliação da solução foi realizada por meio de Estudos de Caso: um com alunos de graduação da UNIRIO, engajados em *Software Livre*, e outro com participantes da comunidade VivaOLinux. A etapa final foi a análise das interações realizadas no fórum estruturado.

Assim, a metodologia adotada foi composta pelas seguintes etapas:

1ª. Etapa:

- Pesquisa Bibliográfica;
- Estudo Exploratório nas Comunidades Virtuais de *Software Livre*:
 - Análise das Comunidades Virtuais de *Software Livre*;
 - *Survey*.

2ª. Etapa: Desenvolvimento do Modelo.

3ª. Etapa: Especificação / Implementação do Ambiente.

4ª. Etapa: Avaliação do Modelo por meio de Estudo de Caso.

5ª. Etapa: Análise do Resultado.

O método de Estudo de Caso foi utilizado para avaliar as interações nas Comunidades Virtuais de *Software Livre* com a introdução de um modelo de argumentação para organizar e estruturar o conhecimento, ou seja, os participantes

interagindo em um problema. De acordo com YIN (2005), a definição de Estudo de Caso é “uma investigação empírica que investiga um fenômeno contemporâneo dentro de seu contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos”.

O Estudo de Caso foi aplicado com o uso da ferramenta desenvolvida, o Fórum Livre, porque dessa forma a análise pode ocorrer em ambiente real, sem interferências externas, isto é, com um fluxo de discussão sendo realizado por participantes de comunidades, nas quais eles inseriram seu conhecimento sobre o assunto, deram sua opinião e geraram conhecimento.

1.6. Estruturação da dissertação

A dissertação foi estruturada em nove capítulos. A introdução oferece uma visão geral da pesquisa realizada; depois são apresentados: os conceitos e características das Comunidades Virtuais de *Software* Livre, os conceitos da Gestão do Conhecimento, o Estudo Exploratório destas comunidades, os conceitos sobre Modelos de Argumentação, o desenvolvimento do Modelo Proposto (ArFoLi), a implementação do Fórum Estruturado para Comunidades Virtuais de *Software* Livre (Fórum Livre), a descrição dos Estudos de Caso com as avaliações, as Conclusões e os Trabalhos Futuros. Finalizando esta dissertação, são apresentadas as Referências Bibliográficas utilizadas e os Apêndices.

A Figura 1 apresenta graficamente a estruturação dessa dissertação em capítulos.

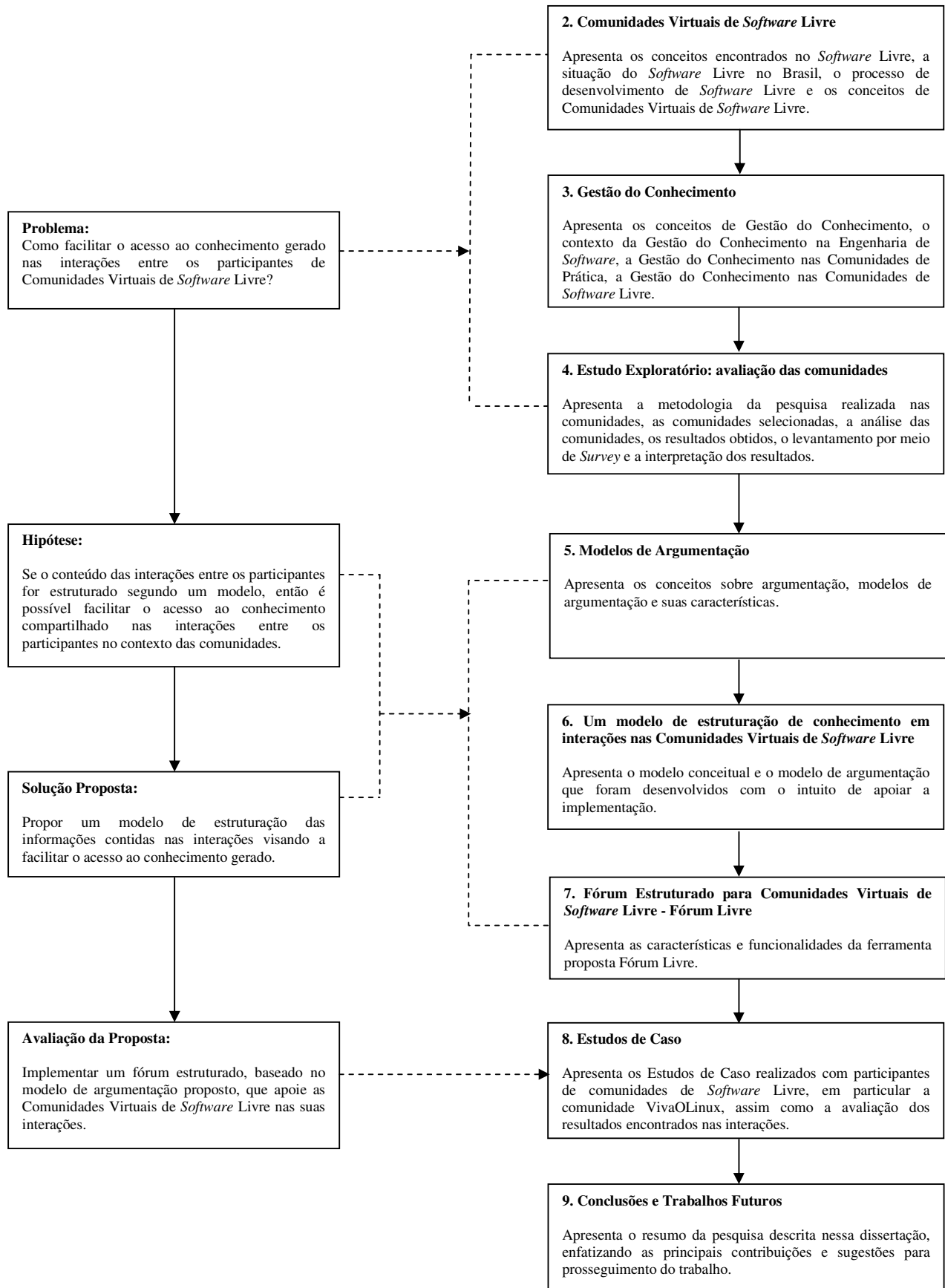


Figura 1 - Apresentação da dissertação em capítulos

Capítulo 2 - Comunidades Virtuais de Software Livre

Este capítulo apresenta os conceitos de *Software Livre*, um breve histórico para introduzir a sua filosofia, a forma como os participantes trabalham e as características das Comunidades Virtuais de *Software Livre*. Também é mostrado por meio de uma pesquisa realizada pela SOFTEX (2005), o cenário do *Software Livre* no Brasil. O *Software Livre* pode ser encarado como uma mudança de paradigma apoiado pela revolução ocorrida com o advento da Internet e a diminuição das barreiras impostas pelo *software* proprietário, já que se antes centralizar significava poder, hoje a descentralização de pessoas e o conhecimento significam avanço tecnológico.

2.1. Conceito de *Software* Livre

O *Software* Livre, conforme a definição da FSF (2008), é o *software* que dá ao usuário a liberdade para estudar, copiar, modificar e compartilhar. É por esse motivo que o *software* é livre, devido ao usuário se sentir livre. O *Software* Livre tem como alicerce o ideal de liberdade, seja pela redistribuição, seja pela modificação ou acesso. É ser livre para não pedir ou pagar pela permissão. *Software* Livre é uma questão de liberdade e não de preço. Para entender o conceito, deve-se pensar em "liberdade de expressão", não em "cerveja grátis" (GNU, 2008).

O *Software* Livre é um movimento social, em que as pessoas querem participar, contribuir e se unir em projetos, construindo uma confiança e uma reputação (SCCACHI, 2008). Os participantes também desafiam a tecnologia e o tempo (SCCACHI, 2008).

O *Software* Livre surgiu em 1983 com Richard Stallman que criou o Projeto GNU e, posteriormente, a *Free Software Foundation* (CAMPOS, 2006). O Projeto GNU tem como objetivo criar um sistema operacional totalmente livre, no qual o usuário tenha o direito de usar, modificar e redistribuir o programa (código fonte), desde que todos tenham os mesmos direitos. GNU é um sistema operacional com os princípios do UNIX, e seu significado é GNU Não é UNIX (GNU is Not UNIX), sendo que GNU é um mamífero e o logotipo do projeto (FSF, 2008).

Em 1991, o sistema operacional GNU estava quase todo pronto, faltando apenas o *kernel*. O *kernel*, então, foi desenvolvido por Linus Torvalds, usando todo o código do sistema operacional GNU. Esse *kernel* ficou conhecido como Linux, contração de Linus e Unix (CAMPOS, 2006).

Alguns exemplos de *Software* Livre, de acordo com TAURION (2004), são: o sistema operacional Linux (www.ubuntu-br.org), o ambiente gráfico KDE (www.kde.org), o compilador GCC (<http://gcc.gnu.org/index.html>), o servidor web Apache (www.apache.org), o conjunto de ferramentas OpenOffice.org (www.openoffice.org), o navegador web Firefox (Mozilla) (www.mozilla.com/firefox), o banco de dados PostgreSQL (www.postgresql.org), o banco de dados MySQL (www.mysql.com), o servidor de aplicação Jboss (www.jboss.org), a linguagem de programação Perl (www.perl.org), a linguagem de programação Python

(www.python.com), o servidor de arquivo e impressão Samba (www.samba.org), o ERP Compiere (www.compiere.com).

O *Software* Livre se refere à existência simultânea de quatro tipos de liberdade para os seus usuários, definidas pela FSF (2008). As quatro liberdades básicas associadas ao *Software* Livre são:

- A liberdade de executar o programa, para qualquer propósito (liberdade nº 0);
- A liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades (liberdade nº 1). Acesso ao código fonte é um pré-requisito para esta liberdade;
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2);
- A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade nº 3). Acesso ao código fonte é um pré-requisito para esta liberdade.

O *software* proprietário visa a atender a demanda dos clientes e a um negócio específico enquanto o movimento de *Software* Livre visa a atender a sociedade, ou seja, qualquer um, pessoa física ou jurídica, que tenha acesso ao código publicado. *Software* Livre é baseado em *copyleft*, ou seja, uma licença que é uma extensão das quatro liberdades básicas implementada na forma de uma obrigação. “O *copyleft* diz que qualquer um que distribui o *software*, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa. O *copyleft* garante que todos os usuários tenham liberdade.” De acordo com a FSF (2008), caso se receba um *software* com uma licença livre que inclua cláusulas de *copyleft*, e se optar por redistribuí-lo (alterando ou não), é necessário que o *software* seja mantido com o mesmo tipo de licença recebida.

A característica de *copyleft* não é encontrada em todos os *Softwares* Livres. A licença GNU-GPL (*General Public License*), adotada pelo *kernel* Linux, é um exemplo de uma licença *copyleft*. Outras licenças livres, como a licença BSD (*Berkeley Software Distribution*) ou a licença ASL (*Apache Software License*), a licença MPL (*Mozilla Public License*) não contêm a característica de *copyleft*, segundo CAMPOS (2006).

Em 1998, segundo CAMPOS (2006), foi criada a *Open Source Initiative* (OSI), formada pela comunidade e por profissionais do mercado interessados em *Software Livre*, porém insatisfeitos com a postura filosófica do movimento de *Software Livre* criado pela FSF (2008), ou seja, a ideia de liberdade sem restrições, de divulgar livremente o *software*. Eles acreditavam que esse movimento atrasava a adoção do *Software Livre* no ambiente corporativo, devido ao tipo de licença adotada (GPL) para o produto gerado. Com isso, adotou-se o termo *Open Source* (Código Aberto) para fazer uma ligação com os *Softwares Livres* e focar na utilização do *software* de código aberto como uma solução adequada e sem filosofia como a *Free Software Foundation* pregava. Pode-se então distinguir a *Free Software Foundation* da *Open Source Initiative* da seguinte forma: a primeira está voltada para o *software* baseado em questões éticas, direitos e liberdade, e a segunda baseada numa questão técnica e sem conflitos com questões éticas, de forma a aproximar o *Software Livre* das empresas.

A diferença entre o movimento da *Free Software Foundation* e o movimento da *Open Source Initiative* está na filosofia e nos objetivos de cada entidade, e se um *software* é livre, ele também é *open source* e vice-versa, segundo CAMPOS (2006), ou seja, a primeira é muito ideológica e a segunda tem com foco na evolução, no mercado de desenvolvimento. A *Free Software Foundation* está baseada na filosofia da liberdade para o *software* e a *Open Source Initiative* tem o objetivo de desenvolver em grupo e não deixar que se fique preso a um fornecedor de *software* com suas imposições de licença e uso.

Do ponto de vista da FSF (2008) “a relação entre o movimento *Free Software* e o movimento *Open Source* é justamente como o oposto de uma figura em relação à *Free Software Foundation*. Nós discordamos nos princípios básicos, mas concordamos (mais ou menos) com as recomendações práticas. Assim, nós podemos fazer e trabalhar juntos em vários projetos específicos. Nós não vemos o movimento *Open Source* como um inimigo. O inimigo é o *software* proprietário.” Isso significa que os dois movimentos possuem posições diferentes um com foco na sociedade e um com foco nas empresas.

A *OPEN SOURCE INITIATIVE* (2008) tem como objetivo o processo de desenvolvimento de *Open Source Software*, que funciona com base na alteração do código fonte pelos pares distribuídos e na transparência do processo, com qualidade, confiabilidade, baixo custo e inexistência de um fornecedor que tire proveito somente para si da produção do *software*. Segundo a *OPEN SOURCE INITIATIVE* (2008), *open*

source não significa somente acesso ao código fonte e sim atender aos dez critérios da Definição de Código Aberto, que são: (i) livre redistribuição de *software* sem taxas de vendas, (ii) acesso ao código fonte para facilitar as mudanças, (iii) a licença deve permitir modificações e trabalhos derivados a partir de um código fonte, (iv) integridade do autor do código fonte, para que a modificação do mesmo seja identificada com seu autor, novo nome e novo número de versão derivado do original, (v) não discriminar pessoas ou grupos que contribuam com o processo de desenvolvimento alterando o código fonte, (vi) não discriminação contra áreas de atuação, (vii) distribuição de licença onde os direitos associados ao programa devem aplicar-se a todos a quem o programa é redistribuído sem a necessidade de execução de uma licença adicional por essas partes, (viii) a licença não deve ser específica para um produto, (ix) a licença não deve estar restrita ao outro *software* desenvolvido, ou seja, não deve colocar restrições sobre um outro *software* que é distribuído junto com o *software* licenciado e (x) licença neutra em relação à tecnologia, não estando subordinada a qualquer pessoa ou estilo tecnologia ou interface. Como exemplos de projetos nesse estilo de desenvolvimento têm-se: Mozilla Firefox, Apache HTTP *Server*, Jaca SDK, Linux, MySQL.

Em diversas pesquisas sobre *Software* Livre (FSF , 2008; *Open Source Initiative*, 2008; SOWE *et al.*, 2007; KOCH, 2004; ELLIOTT e SCACCHI, 2003; SCACCHI, 2007), os dois termos são utilizados de forma a caracterizar uma ideologia ou uma terminologia, nesta dissertação será utilizado o termo *Software* Livre como definido por KOCH (2004).

2.2. O *Software* Livre no Brasil

O Brasil, segundo TAURION (2004), possui 50% do mercado de *software* da América Latina, seus desenvolvedores possuem nichos especializados para explorar, como os *softwares* de gestão empresarial para pequenas e médias empresas, *software* embarcado, *software* de automação comercial, *software* para o segmento financeiro, tornando-se competitivo mundialmente nesses nichos. O Brasil é pioneiro em sistemas como a entrega de impostos de renda pela Internet e a votação eletrônica. A indústria brasileira, segundo TAURION (2004), pode ser considerada uma das principais forças entre os países em desenvolvimento e o Brasil se destaca como um dos países que mais crescem no setor de TI (Tecnologia da Informação).

O modelo de *Software* Livre/Código Aberto tem despertado o interesse e suscitado reflexões nos mais diversos âmbitos (governo, academia, empresas etc.), no Brasil e no exterior. O surgimento de uma rede virtual de desenvolvedores e usuários, complexa, auto-organizada, com motivações diversas e a existência de novas formas de licenciamento de *software* sinalizam a introdução de variáveis no setor de *software*. O *Software* Livre desponta como opção estratégica para o desenvolvimento tecnológico com vista à inclusão social, a partir de experiências bem sucedidas em diversas localidades do Brasil, de acordo com a SOFTEX (2005).

A pesquisa realizada pelo Observatório Econômico da Softex e o Departamento de Política Científica e Tecnológica da UNICAMP, com o apoio do Ministério de Ciência e Tecnologia (SOFTEX, 2005), indica que, apesar de não se tratar de uma ruptura tecnológica, o modelo *Software* Livre/Código Aberto traz uma nova forma de desenvolver e licenciar *software* que está quebrando modelos tradicionais de apropriabilidade e de desenvolvimento tecnológico.

O fenômeno de construção, interação e geração de resultados pelas comunidades é algo sem precedentes na história do setor de *software*. Em boa parte dessas comunidades inexistem laços formais para participação e parece haver um crescente fluxo de geração de novas comunidades e do processo de aprendizagem coletiva (SOFTEX, 2005).

Os resultados da pesquisa da SOFTEX (2005) contradizem alguns mitos em relação ao modelo *Software* Livre/Código Aberto no Brasil. O perfil dos desenvolvedores brasileiros é semelhante ao perfil europeu, que é bastante profissionalizado, com a predominância de profissionais qualificados: administradores de sistemas, técnicos de redes, empresários, pesquisadores e estudantes com nível superior. Dentre as empresas desenvolvedoras, segundo a SOFTEX (2005) há o predomínio de pequenas empresas, mas grandes empresas também já adotam este modelo para realização de negócios.

Quanto aos usuários, o perfil se inverte. Há predomínio de grandes organizações, com destaque para os setores de tecnologias da informação e comunicação, governo, comércio e educação. Suas principais motivações são econômicas (diminuição de custos) e técnicas (desenvolvimento de novas habilidades), de acordo com a SOFTEX (2005).

A pesquisa de REIS (2003) identificou algumas características referentes às comunidades brasileiras de desenvolvimento de *Software Livre*. O perfil das comunidades brasileiras avaliadas é o de pequenos grupos, com cinco indivíduos em média, em que tanto são desenvolvedores quanto usuários dos *softwares* em torno do qual se organizam. É muito frequente que existam participantes com mais de cinco anos de experiência nestas comunidades, o que representa uma base sólida para que um projeto avance. Os exemplos dessas comunidades são: OpenOffice (<http://openoffice.org.br>) e a Mozilla (<http://mozilla.org.br>).

O Brasil pode tirar proveito das Comunidades Virtuais de *Software Livre*, ao visar no desenvolvimento e amadurecimento tecnológico, integrando desenvolvedores dispersos no país e promovendo a melhoria contínua da tecnologia.

Com a mudança de paradigma no contexto do desenvolvimento, advinda do movimento de *Software Livre*, não só os desenvolvedores, mas também as empresas começam a prestar atenção no que está acontecendo tecnologicamente. Com isso, diversas empresas questionam se vale a pena trocar sua plataforma fechada por uma plataforma sem restrições e aberta, visto fatores como custo, vontade e habilidade de lidar com esse tipo de plataforma (DEDRICK e WEST, 2004). Essa nova plataforma aberta requer mudanças sociais e políticas para se adaptar aos novos padrões, para isso, é necessário entender os benefícios da mudança para a empresa. Isso demonstra que as empresas estudam a possibilidade de migrar para o *Open Source* desde que os benefícios e seus riscos fiquem evidentes.

O ambiente *Open Source* é um ambiente de baixo custo baseado em um forte modelo de negócio e apoiado por comunidades especializadas em determinados assuntos tais como: linguagens alternativas, bibliotecas digitais, iniciativas de publicação, grupos de interesse acadêmico especial e blogs. Desta forma, está apto a oferecer diversos serviços para seus usuários (BUNTINE *et al.*, 2005).

Nesse ambiente, segundo SCCACHI (2008) e O'REILLY (1999), novos tipos e práticas, processos e formas organizacionais para descobrir, observar, analisar, modelar e simular tecnologia são apresentados. Assim, é importante entender como funciona o processo de desenvolvimento no *Software Livre*.

2.3. O processo de desenvolvimento de *Software Livre*

SOMMERVILLE (1995) define processo de desenvolvimento de *software* como “um conjunto de atividades e resultados associados que produzem um produto de *software*.” O estilo tradicional de desenvolvimento é composto basicamente de três fases: Especificação, Desenvolvimento e Manutenção. O processo da Engenharia de *Software* e o processo de Engenharia de *Software* Livre (*Open Source*) foram pesquisados por VIXIE (1999) que aponta a diferença entre o rigor da Engenharia de *Software* tradicional e o desenvolvimento do *Software* Livre em mentalidade e forma de trabalho.

Para VIXIE (1999), o processo proposto na Engenharia de *Software* tradicional é composto por sete etapas: Levantamento de Requisitos de Mercado, Projeto em Nível de Sistema, Projeto Detalhado, Implementação, Integração, Teste de Campo do Sistema e Suporte. Os Requisitos de Mercado estão ligados à criação de um documento com as necessidades do cliente, que se deseja alcançar com o produto, e uma lista de características do produto. O Projeto em Nível de Sistema é a descrição do produto com seus módulos e as interações entre eles. O Projeto Detalhado é o documento que descreve a chamada dos módulos no sistema em detalhes, a interface e as dependências entre os módulos. A Implementação é a codificação ou a programação do sistema, ou seja, um módulo pode ser considerado implementado quando foi criado, testado e usado corretamente por outro módulo. A Integração acontece depois que todos os módulos estão desenvolvidos, mas pode ser feita de maneira incremental em paralelo à Implementação. O Teste de Campo acontece dentro da organização para verificar o uso do sistema e corrigi-lo. O Suporte é o apoio realizado para correção do sistema durante os Testes de Campo, ou depois do *software* ter sido distribuído, podendo implicar em mudanças no sistema desenvolvido.

Em relação ao processo de Engenharia de *Software* Livre (*Open Source*), VIXIE (1999), aponta que as sete etapas existem, mas de maneira um pouco diferente. No *Software* Livre, os Requisitos de Mercado são voltados para a necessidade ou desejo pessoal do desenvolvedor e o documento gerado é uma lista de *e-mail* (*mailing list*) ou notícias para a comunidade envolvida no projeto do *software*. O Projeto em Nível de Sistema não acontece com o *Software* Livre não financiado, está implícito nos comentários existentes em versões do projeto desenvolvido e não existem as regras formais, mas é levada em conta a reputação dos desenvolvedores, considerando o que ele faz para comunidade e o que os outros participantes sabem sobre ele. O Projeto

Detalhado também não acontece para o *Software* Livre não financiado, o que dificulta a reutilização. A Implementação é o momento que os desenvolvedores experimentam novos estilos de programação, a revisão é informal e os mentores (ou coordenadores do projeto) verificam o código antes do mesmo ser publicado, não possuindo uma única homologação. Um módulo pode ser considerado implementado quando foi criado, testado e utilizado com sucesso. A Integração é realizada informalmente, criando-se mecanismos para que aconteça através da escrita de arquivos *README*, postagem de notas em lista de *e-mails* ou notícias do grupo e com esforços em testes globais. O Teste de Campo acontece não de maneira formal como na Engenharia de *Software* e sim pela revisão em pares dos desenvolvedores envolvidos, sem um plano de testes para o sistema. O Suporte acontece quando são encontradas mensagens de erros (*bugs*) e então, novos pacotes são gerados. Neste momento, os consultores e distribuidores de *software* encontram uma oportunidade financeira com o *Software* Livre.

O processo de manutenção de *software* também é um ponto importante, visto que envolve atualizações e correções. No *Software* Livre, a manutenção é realizada pela comunidade com a liberação de novas versões. KOPONEN e HOTTI (2005) compararam o modelo de manutenção *Open Source* com a ISO/IEC e encontraram similaridades entre os modelos de manutenção. O estudo foi realizado no projeto do servidor Apache HTTP e no navegador Mozilla; os autores observaram que o processo de manutenção no *Open Source* apresenta um ciclo com menos etapas do que na ISO/IEC. Nesses projetos, é feito controle de versões e de *bugs* (MOCKUS *et al.* 2002).

SCACCHI (2002, 2007) também analisou o processo de engenharia de requisitos em projetos *Open Source* em diferentes comunidades e realizou uma comparação em relação ao processo tradicional da Engenharia de *Software*. Ele argumentou que os projetos *Open Source* são desenvolvidos informalmente e ocorre um processo discursivo de construção do *software*, ao invés do processo formal com produção de documentação baseada em padrões. SCACCHI (2002) sugere que por meio da etnografia é possível elicitar, analisar, validar e comunicar o processo de desenvolvimento nos projetos *Open Source*.

NAKAGAWA (2004) concluiu que o processo de desenvolvimento nas comunidades de *Software* Livre tem o objetivo de produzir um *software* com foco na participação de diversos colaboradores. A forma de trabalho se diferencia do tradicional, já que os programadores são colaboradores e voluntários nos projetos onde

atuam. Os desenvolvedores estão dispersos geograficamente e se utilizam das comunidades que participam para trocar conhecimento, divulgar novas soluções de *software* e novas tecnologias. Dessa forma, o processo de desenvolvimento de *Software Livre*, concordando com VIXIE (1999), afirma que tende a ser diferente dos padrões propostos pela Engenharia de *Software*, mas no entanto, podem resultar em produtos com características vantajosas.

De forma contrária, de acordo com a SOFTEX (2005), o desenvolvimento de *software* (proprietário) é realizado por grupos de desenvolvedores dentro de uma empresa ou de empresas contratadas para tal, sob contratos que impedem a divulgação e o uso de informações relacionadas ao produto em desenvolvimento. Esses contratos estão relacionados a questões de sigilo industrial e de propriedade intelectual (direito de autor), e o conhecimento relacionado à produção dos *softwares* é considerado um ativo muito importante da organização proprietária.

O estilo de desenvolvimento de *software*, segundo RAYMOND (1998), pode ser caracterizado como “catedral” ou “bazar”. O estilo Catedral é centralizado, organizado, utilizado por um grupo de desenvolvedores, é o modelo de desenvolvimento típico para *software* proprietário, com a restrição adicional do código fonte não ser normalmente solicitado com as atualizações (por exemplo, o projeto Emacs). O *software* proprietário é a catedral de RAYMOND (1998) devido à sua estruturação e segundo a descrição de TAURION (2004), esse projetos de *software* proprietário adotam padrões de certificações, tais como CMMI (2008). Esses projetos sofrem pressões comerciais, atendem a prazos, cumprem cronogramas, desenvolvem um código fonte fechado, liberam apenas o executável; as suas licenças são copyright, criando uma documentação e uma estrutura de suporte técnico para atender aos clientes.

O estilo Bazar é adotado pelas comunidades de *Software Livre* em que o código é desenvolvido de forma totalmente aberta e pública, utilizando a Internet. RAYMOND (1998) contrasta os dois estilos de desenvolvimento, analisa o impacto do Linux para os “construtores de catedrais” e cita o estilo de desenvolvimento de Linus Torvalds, ou seja, “libere cedo e frequentemente, delegue tudo que você possa, esteja aberto ao ponto da promiscuidade”.

O estilo Bazar é o alicerce das Comunidades Virtuais de *Software Livre* devido à sua característica de desenvolvedores dispersos geograficamente, descentralizado, informal, com pouca coordenação e disposto a atender um projeto de *software* com

qualidade, devido à contribuição de todos os participantes. Esse estilo tem sido aplicado com sucesso no desenvolvimento de *softwares* como Linux e Apache, segundo SILVA e FALBO (2006). Projetos de desenvolvimento pessoais são divulgados a uma comunidade virtual e se tornam projetos coletivos. Apesar disso, também existe o risco de que, se não forem atraentes o suficiente, podem não ir à frente.

TAURION (2004) apresenta os papéis dos participantes deste processo: mantenedores, contribuidores e organizadores. Os mantenedores são os criadores de projetos, os que idealizam um projeto para resolver um determinado problema. Os contribuidores são os que participam voluntariamente das comunidades desenvolvendo o código fonte ou enviando sugestões para os mantenedores. Os organizadores são os que apóiam os mantenedores em projetos, mas não são considerados coordenadores devido à informalidade existente no projeto. Todos esses participantes da comunidade geralmente são desenvolvedores.

O ciclo de vida de um projeto de *Software Livre*, segundo TAURION (2004), é o seguinte:

1. O mantenedor cria a versão inicial do código fonte do *software*, e divulga na Internet, ou seja, no repositório da comunidade virtual;
2. Os contribuidores baixam a versão inicial, testam, avaliam, descobrem erros, corrigem erros, aprimoram o código fonte inicial e enviam para o mantenedor, via *e-mail*;
3. O mantenedor seleciona as melhores contribuições, tentando manter uma uniformidade no código fonte e libera uma nova versão;
4. A nova versão é novamente avaliada pela comunidade.

Esse ciclo é contínuo ou persiste até surgir um projeto mais interessante ou desafiador para os contribuidores. Os projetos têm seu tempo de vida ditado de acordo com o interesse da comunidade (TAURION, 2004), e o que motiva os participantes na comunidade é o reconhecimento pelos pares, o prestígio e a reputação estabelecida, e não o financeiro.

Os desenvolvedores *Open Source* são considerados como cientistas construtores de código fonte que promovem a inovação e a geração de conhecimento, por meio de comunidades pela Internet, para a resolução de seus problemas, e liberdade do

compartilhamento obtendo resultados significativos (DEMPSEY *et al.*, 2002; CAMPBELL-KELLY, 2008).

Nos projetos de *Software Livre* (TAURION, 2004), as decisões são tomadas por uma comunidade e prevalece a meritocracia, ou seja, o sistema em que os indivíduos com mérito se destacam, os que agregam valor a comunidade são reconhecidos. Existem também projetos em que as decisões são tomadas por um grupo específico, e a documentação nesses projetos somente se torna importante quando o *software* atinge determinado grau de disseminação no mercado e são criadas empresas para atuar nas tarefas de distribuição e suporte.

Outra diferença segundo TAURION (2004) é a longevidade. Os *softwares* proprietários são abandonados ou desativados quando se tornam deficitários. Para o movimento de *Software Livre*, a motivação está na comunidade virtual relacionada com a visibilidade e a popularidade do projeto. O lucro do *software* proprietário está no desenvolvimento focado ao negócio, já no *Software Livre* o lucro está nas empresas distribuidoras, no suporte e nos serviços complementares que estes geram ao redor das comunidades virtuais.

Segundo REIS (2003), “Projetos de *Software Livre* são organizações virtuais formadas por indivíduos que trabalham juntos no desenvolvimento de um *Software Livre* específico. Estes indivíduos trabalham geograficamente dispersos, utilizando ferramentas simples para coordenar e comunicar seu trabalho através da Internet.” “Com relação às atividades do processo de *software*, pode-se dizer que a maioria dos projetos tem requisitos fundamentalmente definidos pelos seus autores, e que a base de usuários de grande parte dos *softwares* é composta dos seus desenvolvedores e da comunidade de *Software Livre*. Uma parcela significativa dos projetos baseia-se em outros *softwares* pré-existentes, e em padrões publicados previamente.”

MOCKUS *et al.* (2002) caracterizou os processos de desenvolvimento dos Projetos do Servidor de HTTP Apache e do Navegador Mozilla e, os analisou para responder as seguintes questões: escopo, papéis e responsabilidades, identificação do trabalho a ser feito, atribuição e realização do trabalho de desenvolvimento, pré-lançamento para testes, inspeção e gerenciamento das atualizações (*releases*). A conclusão destes autores foi que elementos dos projetos *open source* e comerciais podem ser combinados. O Mozilla representa um exemplo de migração de um desenvolvimento fechado para o aberto, que obteve melhoria na qualidade com as

contribuições dos desenvolvedores da comunidade, tanto no desenvolvimento quanto nos testes e, um amadurecimento dessa ferramenta.

YE e KISHIDA (2003), bem como SCACCHI *et al.* (2006), SILVA e FALBO (2006) e FELLER e FITZGERALD (2000), concluíram em suas pesquisas que a motivação do desenvolvimento dos projetos *Open Source* está ligada à forma colaborativa como os usuários e desenvolvedores atuam. O sucesso dos projetos está ligado à contribuição, à participação voluntária e ao envolvimento dos membros das comunidades. YE e KISHIDA (2003) argumentam que uma grande força motivacional é o aprendizado gerado e o compartilhamento de conhecimento nas comunidades.

A qualidade também é uma importante questão para o desenvolvimento de *software* e também no âmbito do *Open Source*. Segundo POLANCIC *et al.* (2004), muitos *softwares Open Source* ainda não atingiram maturidade, e por isso é proposto um modelo que avalie o *software* usando métricas de qualidade, usando dados quantitativos. O modelo é composto pelas fases: Identificação de alternativas de *softwares Open Source*, Identificação das métricas, Definição dos pesos das métricas, Aglomeração das métricas e das alternativas, Avaliação das características de qualidade, Definição dos pesos das características de qualidade e Escolha da alternativa de *software Open Source*. O modelo de avaliação garante repetibilidade, reprodutibilidade, imparcialidade e objetividade.

Segundo GARTNER (2007), o sucesso dos projetos de *Software Livre* é caracterizado por uma mistura de intensa participação, engajamento e motivação juntamente com baixos custos de coordenação e então são examinados nove princípios que dão origem a estas propriedades desejáveis, destacando benefícios transferíveis para outros contextos de negócios. Os nove princípios analisados são: padrões, objetivos claros e alinhamento com os requisitos, decomposição do problema, histórico de trabalho em progresso e visibilidade corrente, história de contribuição individual visível, regras claras de envolvimento destinadas à persuasão e o consenso, auto-seleção do participante, melhoria contínua e integração de baixo custo, ausência de barreiras para a participação e equidade e reciprocidade na propriedade e no acesso.

Os padrões, objetivos claros e alinhamento com os requisitos focam em apoiar o esforço do desenvolvimento, diminuindo o debate, a distância entre os usuários e os desenvolvedores e a evolução da solução frente o problema original, facilitando o alinhamento do produto com os requisitos. Os benefícios do *Software Livre* são: menos

espaço para a divergência, esforço colaborativo focado na mesma direção, fácil alinhamento das soluções com os requisitos.

A decomposição do problema está ligada ao “dividir para conquistar, ou seja, criação de módulos, trabalho independente e podendo ser em paralelo. Isso ajuda a acelerar o desenvolvimento, minimizar os custos de integração e melhorar a flexibilidade e a habilidade envolvida. Além do princípio do reuso, observa-se um processo de desenvolvimento mais rápido de módulos independentes, flexibilidade e habilidade de composição.

O histórico de trabalho em progresso e a visibilidade corrente estão ligados ao progresso do desenvolvimento, testes, código e distribuição da aplicação, documentação, monitoramento de problemas e suporte, assim como as discussões e práticas aplicadas, visando às futuras pesquisas e referências. Os benefícios são: redução dos custos de coordenação através de serviços próprios de relatórios de status, rápida inserção dos recém-chegados no desenvolvimento, continuidade quando os participantes deixam a comunidade e um recurso de aprendizagem sobre o produto, incluindo os detalhes que conduzem a decisões de projetos particulares.

A história de contribuição individual visível refere-se às contribuições de todos os tipos enviadas pelos participantes, as quais sustentam a sua reputação, contribuem para regular a confiança e determinar a autoridade do indivíduo na comunidade. Os benefícios são: oportunidade para os indivíduos construírem sua reputação, confiança, status e autoridade reguladas de forma meritocrática e grande entendimento do dinamismo do grupo que apóia na assimilação de novos membros.

As regras claras de envolvimento destinadas à persuasão e o consenso são a forma de controlar e organizar as discussões nas comunidades para resolver conflitos e apoiar a tomada de decisões. Os benefícios são: *stakeholders* em todos os níveis sentem suas opiniões ouvidas e os participantes estão engajados e mais propensos a fazerem contribuições positivas.

A auto-seleção do participante é forma de participação voluntária nas comunidades e que pode estar ligada a oportunidades, ou seja, aprender e desenvolver novas habilidades, melhorar os produtos de outros desenvolvedores, resolver um problema que não podia ser resolvido com *software* proprietário, participar da comunidade e ajudar a tornar uma boa ideia realidade. Como benefício tem-se a motivação garantida e a maximização do engajamento e da criatividade.

A melhoria contínua e integração de baixo custo estão ligadas às atualizações incrementais (*releases*) permitindo testes e detecção de problemas. Os benefícios são: a detecção precoce de falhas de projeto, a detecção realista de problemas em larga escala, o baixo custo de integração e a capacidade de evoluir.

Ausência de barreiras para a participação é a permissão de ingresso nas comunidades sem pré-requisitos ou resistências, e consequente acesso às informações e código. O benefício é o aumento da disponibilidade para participar de comunidades de código fonte aberto.

Equidade e reciprocidade na propriedade e no acesso estão ligadas a licença de uso do *software* produzido, ou seja, à permissão para modificar e redistribuir. Os produtos produzidos por comunidades *open-source* são ligados a licença GPL. Os benefícios são: aumento da disponibilidade para contribuir e a proteção de um valioso conjunto comum contra a apropriação indevida.

Os nove princípios do GARTNER (2007) são relevantes em atividades colaborativas em grande escala, que envolvem participantes em relacionamento formal, e são observadas as evidências do sucesso destes princípios em termos de organizar o trabalho além do desenvolvimento do *software*. A interação e registro do conteúdo mostram-se fundamentais no processo de desenvolvimento do *Software Livre*. Desta forma, prover mecanismos para apoio à disseminação do conhecimento passa a ser neste processo extremamente relevante.

Com tudo isso, pode-se dizer que o *Software Livre* revoluciona o processo de desenvolvimento e estimula o livre compartilhamento, em que a cultura de desenvolvimento proprietário é alterada e uma nova perspectiva se origina, gerando novas oportunidades (YANG e WANG, 2008). O desenvolvimento *Open Source* é uma quebra de paradigma, segundo WU *et al.* (2001), visto que uma comunidade diversificada que contribui continuamente para desenvolvimento e com motivações diferentes para programar atua nesse contexto e os participantes acabam sendo recompensados com a construção do capital humano e o reconhecimento, que gera recompensas financeiras futuras.

O processo de desenvolvimento no *Software Livre* então não envolve uma empresa e sim um grupo de pessoas que colaboram entre si para desenvolver. Esse grupo é chamado de Comunidade Virtual de Software Livre.

2.4. Comunidades Virtuais de Software Livre

De acordo com WENGER *et al.* (2002), as Comunidades de Prática podem ser definidas como grupos de pessoas que compartilham um interesse, um conjunto de problemas ou uma paixão sobre algum assunto, interagindo de maneira contínua para desenvolver conhecimento, de forma a criar uma prática em torno deste assunto. As Comunidades de Prática diferem-se pelas estruturas de grupo de trabalho, seja pelo motivo que une seus membros, seja pela forma de como interagem e pela aplicação que fazem dos conhecimentos compartilhados.

Segundo WENGER (2006), Comunidades de Prática (CoP) são grupos de pessoas que compartilham uma preocupação ou uma paixão por algo que fazem e aprendem como fazê-lo melhor quando eles interagem com regularidade. Existem três características importantes nas Comunidades de Prática: o domínio, a comunidade e a prática. O domínio é definido pelo assunto de interesse dos membros e não simplesmente como um grupo de amigos, é necessário compartilhar competências. A comunidade está ligada aos membros que estão engajados em atividades similares e discussões, em ajuda mútua e no compartilhar as informações, construindo assim relacionamentos que lhes permitem aprender uns com os outros em torno de um domínio. Já a prática, está ligada ao corpo do conhecimento, o “fazer”, a experiência dos participantes da comunidade, que são profissionais e que desenvolvem repositórios para compartilhar recursos em uma prática de compartilhamento.

Segundo WENGER (2006), as Comunidades de Prática são aplicadas a WEB, pois as novas tecnologias como a Internet tem estendido o alcance das interações para além dos limites geográficos das comunidades tradicionais, porém o aumento do fluxo de informação não elimina a necessidade de aprendizado da comunidade. Na verdade, amplia as possibilidades de uma comunidade e apela para novos tipos de comunidades baseadas na partilha de práticas.

As Comunidades de Prática são integradas, de acordo com NEVES (2001) e WENGER (2003) pelos seguintes papéis: coordenador, moderador/mediador, suporte técnico, especialista (*expert*), bibliotecário e facilitador de conhecimento (*knowledge brokers*). GOUVEA (2005) afirma que para um profissional é importante pertencer a uma Comunidade de Prática, porque é muito difícil adquirir conhecimento sozinho, faz-se necessário, portanto, a interação com pessoas para falar dos problemas, pedir ajuda e informação. Um ponto de vista de estímulo a participação em Comunidades de Prática

organizacionais, é estimular a participação na comunidade do empregado de forma a gerar conhecimento para a organização e também recompensá-lo conforme sua atuação (GOUVEA, MOTTA e SANTORO, 2006).

As Comunidades Virtuais de *Software* Livre podem ser consideradas como Comunidades de Prática, com a particularidade de que além de interesse em um tema comum de debate (*Software* Livre), seus membros de fato desenvolvem produtos colaborativamente. Por exemplo, a partir de um problema específico de um dos membros iniciam a implementação conjunta de um *software*. Estas comunidades são formadas por voluntários interessados em participar de projeto de desenvolvimento ou divulgar as ferramentas de *Software* Livre. Existem geralmente os mantenedores que realizam a moderação e a organização da comunidade.

As Comunidades Virtuais de *Software* Livre contam com participantes, processos, tarefas, metas, regras, costumes e operações diferenciadas das comunidades convencionais, visto que seu funcionamento é voltado para ferramentas, programas de *software* (GE *et al.*, 2006). Essas comunidades constroem colaborativamente e compartilham o conhecimento entre seus participantes (GE *et al.*, 2006)

As motivações para participar de um projeto *Open Source* estão relacionadas com necessidades pessoais, altruísmo, reconhecimento externo e retorno no futuro, segundo HARS e OU (2001). Isso faz com que grupos diferentes de pessoas se unam, se esforcem nesses projetos e produzam resultados surpreendentes (HARS e OU, 2001).

As Comunidades Virtuais de *Software* Livre se disseminam e trazem consigo a força da descentralização apesar da distância geográfica, das diferentes agendas dos voluntariados, da colaboração, do uso de repositórios de desenvolvimento de projetos, da disseminação do conhecimento e do desenvolvimento tecnológico. Cada comunidade tem um objetivo, uma motivação e pode estar ligada a um desenvolvimento de um *software*.

Essas comunidades virtuais têm como objetivos:

- a divulgação, apoio e uso de ferramentas de *Software* Livre;
- a disseminação do conhecimento do *Software* Livre;
- o apoio ao desenvolvimento de projetos de *Software* Livre;
- geração de novos *softwares* como produto das comunidades.

As comunidades virtuais estão espalhadas pelo mundo e desenvolvendo diversos projetos de *Software* Livre, seja para disseminação de conhecimento ou para desenvolvimento de ferramentas. A Tabela 1 apresenta exemplos de comunidades.

Tabela 1 - Comunidades Virtuais de Software Livre

COMUNIDADE	SITE
<i>Software Livre</i>	http://softwarelivre.org/
BR-Linux	http://br-linux.org/
VivaoLinux	http://www.vivaolinux.com.br/
LinuxSecurity	http://www.linuxsecurity.com.br/
Dicas-L	http://www.dicas-l.com.br/
NoticiasLinux	http://www.noticiaslinux.com.br/
UnderLinux	http://under-linux.org/
BestLinux	http://www.bestlinux.com.br/
SourceForge.net	http://sourceforge.net/
Freshmeat.net	http://freshmeat.net/
Código Livre	http://www.codigolivre.com.br/
<i>Free Software Foundation</i>	http://www.fsf.org/
<i>Open Source Initiative</i>	http://www.opensource.org/
Centro de Computação da Unicamp	http://www.ccuec.unicamp.br/ccuec
Tigris	http://www.tigris.org/
Google Code	http://code.google.com/intl/pt-BR/
Koders	http://www.koders.com/

As Comunidades Virtuais de *Software Livre* utilizam repositórios virtuais para armazenar e divulgar seus projetos, e comunicação interativa, o que torna a Internet o principal meio para as comunidades virtuais. A comunicação entre os participantes normalmente acontece por meio de *e-mails*, listas de discussão, *chats*, grupos e de fóruns para discussão, ou seja, se utilizando da Internet. Essa comunicação é informal e informativa para a divulgação de conhecimento e apoio aos projetos.

Os ambientes através dos quais as Comunidades Virtuais de *Software Livre* se organizam geralmente são páginas web, onde são divulgadas notícias de interesse da comunidade. Os interessados se inscrevem e participam da comunidade, os projetos e as tecnologias são divulgados e o conhecimento é disseminado e compartilhado.

ELLIOT e SCACCHI (2003) acreditam que as comunidades de *software* livre são comunidades ocupacionais, ou seja, que possuem formas coletivas de crenças, valores e normas, constroem e resolvem seus conflitos apoiadas por meio de ferramentas de mensagens , promovendo a colaboração. A colaboração possui grande relevância nas Comunidades Virtuais de *Software* Livre devido à contribuição que existe entre os pares e a integração para o desenvolvimento de projetos de *Software* Livre.

Segundo a SOFTEX (2005), a colaboração vem se tornando cada vez mais rica, pois ao longo do tempo foram sendo criadas e aperfeiçoadas ferramentas apropriadas para cada atividade. Sistemas distribuídos de controle de versões de *software* e sistemas de rastreamento e controle de *bugs* são dois exemplos de ferramentas para desenvolvedores, enquanto listas de discussão, fóruns e *chats*, *sites* web e outros são utilizados tanto por desenvolvedores quanto por usuários. Muitas vezes, comunidades grandes promovem também encontros presenciais regionais, nacionais ou mesmo internacionais.

O conhecimento nessas comunidades existe em grande quantidade visto que diversos participantes são profissionais e interessados em desenvolvimento de software. O conhecimento é do tipo técnico, de qualquer tecnologia livre (sistema operacional, linguagem de programação ou banco de dados). Devido a esse fato, existe a constante discussão para a obtenção de informações para a resolução de erros ou melhorias. Assim, é propício um estudo para o entendimento da forma como as atividades de organizar, armazenar, compartilhar e reutilizar o conhecimento é feita no âmbito destas comunidades.

Capítulo 3 - Gestão do Conhecimento em Comunidades Virtuais de Software Livre

O conhecimento é atualmente o principal alvo da sociedade e das empresas. Para gerir o conhecimento é necessário entender o cenário em que se está inserido e a necessidade envolvida, não adianta reunir uma coleção de informações ou impor aos envolvidos que participem de qualquer ação se não estiverem claros os objetivos e benefícios. A Gestão do Conhecimento é a forma de organizar o conhecimento existente e compartilhá-lo entre diversos envolvidos, uma comunidade ou uma empresa. O ciclo da Gestão do Conhecimento requer coleta e compartilhamento, ou seja, um trabalho de organização e estruturação colaborativo. As Comunidades Virtuais de *Software Livre* produzem muito conhecimento e têm como característica principal a colaboração. No entanto, para gerir esse conhecimento é necessário entender como é produzido e compartilhado.

3.1. Gestão do Conhecimento

Conhecimento é o “somatório do que se sabe; o conjunto de informações e princípios armazenados pela humanidade”; “fato, estado ou condição de compreender, entendimento” (HOUAISS, 2008). Para DAVENPORT e PRUSAK (1998) conhecimento é o “conjunto de experiências, valores, informações contextuais e insights de especialistas que provêm uma forma para avaliar e incorporar experiências e informação; é originada e se aplica na mente das pessoas; nas organizações, geralmente, encontra-se espalhada não só em documentos ou repositórios, mas também nas rotinas organizacionais, processos, práticas e normas”. O conhecimento é uma forma de desenvolvimento cultural e depende de fatores externos ao indivíduo, podendo ser adquirido e compartilhado de diversas maneiras.

Os pesquisadores (NONAKA e TAKEUCHI, 1997; TERRA, 2000) apontam o conhecimento como um bem, um ativo da organização atual, isto é, algo que poderá gerar um lucro para a própria organização. Isso acontece porque as organizações estão mudando o seu conceito estratégico, voltando-se para o capital intelectual e a inteligência competitiva, o que as leva a garantir vantagens competitivas no mercado.

Ainda para TERRA (2000), o conhecimento passa por transformações que dependem da situação em que o país se encontra (fatores econômicos, fatores culturais), sendo uma eterna reinvenção de conceitos, novas formas de criar produtos e prover serviços. O “aprender ao operar” já não se encontra mais em destaque, ou seja, já não é mais utilizado para aquisição do conhecimento, e sim o “aprender a aprender”. Esse autor faz referência às transformações do conhecimento que ocorrem no Brasil, às mudanças que geram novos conhecimentos nas empresas, na economia do país, na educação.

O conhecimento pode ser classificado como tácito e explícito. O conhecimento tácito é aquele que está na mente das pessoas, que é adquirido pelas experiências pessoais de cada indivíduo. Já o conhecimento explícito é aquele expresso por palavras ou números, representado na forma de documentos, manuais e armazenado em repositório de dados (NONAKA e TAKEUCHI, 1997).

Segundo NONAKA e TAKEUCHI (1997), o conhecimento tácito depende de cada pessoa. Esse tipo de conhecimento é personalizado e de difícil representação e pode ser considerado como o *know-how* da organização. O conhecimento explícito é

expresso com facilidade e pode ser formalizado por meio de documentos, tabelas e gráficos. Esses autores propõem a transformação do conhecimento por meio da Espiral do Conhecimento (representada na Figura 2), que é a interação entre o conhecimento tácito e explícito e apresenta as seguintes formas: externalização, combinação, internalização e socialização.



Figura 2 - Espiral do conhecimento (NONAKA e TAKEUCHI, 1997; NONAKA, TOYAMA e HIRATA, 2008)

A **externalização** é a transformação do conhecimento tácito em explícito. A externalização pode acontecer por meio de conceitos, hipóteses ou modelos. A **combinação** é a transformação de conhecimento explícito em explícito, como os sistemas de informação, que combinam e categorizam conhecimento armazenado em diversos repositórios, redes virtuais, conversas ou reuniões.

Já a **internalização** é a transformação de conhecimento explícito em tácito. Em manuais e os documentos, o conhecimento é explícito. A internalização constitui o processo de aprendizado individual, é o “aprender fazendo”, o *know-how* técnico compartilhado. (NONAKA e TAKEUCHI, 1997)

Por **socialização**, entende-se a transformação do conhecimento tácito em tácito. As observações, as práticas, as imitações e um *brainstorming* em que o conhecimento tácito é compartilhado são exemplos de socialização.

O modelo da Espiral do Conhecimento mostra como os diferentes tipos de conhecimentos são transformados e interagem possibilitando gerar novos

conhecimentos para a organização. O processo composto pelas atividades propostas por NONAKA e TAKEUCHI (1997) e NONAKA, TOYAMA e HIRATA (2008), no qual o conhecimento é obtido, transformado e compartilhado tem sido chamado de Gestão do Conhecimento.

Para TERRA (2005), o conhecimento é o capital da organização e a “Gestão do Conhecimento significa organizar as principais políticas, processos e ferramentas gerenciais e tecnológicos à luz de uma melhor compreensão dos processos de: geração, identificação, validação, disseminação, compartilhamento e uso dos conhecimentos estratégicos para gerar resultados (econômicos) para a empresa e benefícios para os colaboradores internos e externos (*stakeholders*)”. Ele ainda identifica desafios para a Gestão do Conhecimento, os quais são “guiados especialmente por três forças: os enormes volumes de informação que estão sendo criados, armazenados e distribuídos atualmente; a incrível velocidade com que o conteúdo do conhecimento está mudando; e a transformação contínua do local de trabalho. Cada vez mais, para poder sobreviver, as organizações precisam ser mais pró-ativas ao apoiar a criação e reutilização do conhecimento”.

HOU *et al.* (2008) identificam que a Gestão do Conhecimento tornou-se um importante termo na Tecnologia da Informação, pois envolve identificar, reunir, analisar, construir, compartilhar e aplicar conhecimentos e práticas. Para HOU *et al.* (2008) existem 5 fases na Gestão do Conhecimento: definição das informações, aquisição das informações, transformação das informações, construção dos conhecimentos e compartilhamento dos conhecimentos.

Já KING *et al.* (2002) afirmam que a Gestão do Conhecimento é de grande interesse e potencial valor para a Gestão de Negócios, assim como para a função de Sistemas de Informação. Estes identificam dez questões com foco em como realizar o ciclo da Gestão do Conhecimento, ou seja, o uso, obtenção, manutenção, motivação, identificação, avaliação, verificação, projeto, sustento e segurança. As dez questões estudadas são as seguintes: Como usar Gestão do Conhecimento para obter uma vantagem estratégica?; Como conseguir maior apoio à Gestão do Conhecimento?; Como manter um curso para troca do conhecimento organizacional?; Como motivar os indivíduos a contribuir com seus conhecimentos para um sistema de Gestão do Conhecimento?; Como identificar o conhecimento organizacional que devem ser capturados em sistemas de Gestão do Conhecimento?; Como avaliar os custos e

benefícios financeiros da Gestão do Conhecimento?; Como verificar a eficácia, a legitimidade e a importância do conhecimento contribuíam para um sistema de Gestão do Conhecimento?; Qual a melhor forma de conceber e desenvolver um sistema de Gestão do Conhecimento?; Como sustentar o progresso na organização da Gestão do Conhecimento?; Como assegurar segurança ao conhecimento?.

A Tecnologia da Informação então apoia a Gestão do Conhecimento, de acordo com KING *et al.* (2002). As aplicações de Tecnologia da Informação para Gestão do Conhecimento são consideradas os repositórios para o conhecimento, sistemas de boas práticas e lições aprendidas, redes de especialistas e comunidades de prática. A Tecnologia da Informação viabiliza a disseminação do conhecimento, por meio de sistemas de informação que realizam a captura, o armazenamento, a consulta e a recuperação do conhecimento. Além disso, as inovações tecnológicas diminuem as distâncias e as barreiras físicas, que antes eram um problema para as organizações. Hoje, com a intranet, a internet, a videoconferência, *data warehouses*, os sistemas de gerenciamento eletrônico de documentos, o compartilhamento de conhecimento e a produção intelectual de cada indivíduo passaram a ser cada vez mais tangível.

Sendo a Engenharia de Software um domínio de atividades intenso em conhecimento (DE SOUZA, 2003) em que as chaves do sucesso são relacionadas geralmente à experiência e ao conhecimento tácito transmitido durante o desenvolvimento. A Gestão do Conhecimento é altamente relevante (DE SOUZA, 2003), isso vai ao encontro das práticas no âmbito das Comunidades Virtuais de *Software Livre*.

A Gestão do Conhecimento pode ser muito explorada nas Comunidades Virtuais de *Software Livre*, pois elas são uma rica fonte de criação de conhecimento, desenvolvimento de ferramentas, apoio para resolução de problemas, compartilhamento (GE *et al.*, 2006) e também o reuso de conhecimento gerado nas interações.

3.2. Gestão do Conhecimento na Engenharia de *Software*

Na sociedade moderna, apoiada por tecnologias dos mais diversos e diferentes aspectos, o desenvolvimento de *software* é uma atividade de crescente importância e facilitador para as organizações e para a sociedade. O desenvolvimento moderno de *software* é uma atividade de conhecimento intenso e a complexidade do conhecimento

excede a capacidade dos indivíduos de resolver problemas por si só (HENNINGER, 1997).

No contexto da Engenharia de *Software*, segundo KOMI-SIRVIÖ *et al.* (2002), a Gestão do Conhecimento é um conjunto de atividades, técnicas e ferramentas de apoio à melhoria nos processos de desenvolvimento de *software*. KOMI-SIRVIÖ *et al.* (2002) relatam a dificuldade de reuso do conhecimento em diferentes projetos de desenvolvimento de *software* de forma a reduzir os defeitos no *software*, pois o conhecimento não é facilmente recuperável para compartilhamento.

Segundo DESOUZA (2003), a Gestão do Conhecimento serve como apoio aos profissionais em projetos de desenvolvimento por meio da observação de experiências passadas e lições aprendidas. Porém, não é nada trivial capturar este conhecimento tácito, categorizá-lo, disponibilizá-lo em repositórios e disseminá-lo. Segundo DESOUZA (2003), formas alternativas de troca de conhecimento, como por exemplo, a socialização entre os profissionais, deve ser incentivada. Ele identifica que os sistemas de Gestão do Conhecimento devem promover o diálogo e não apenas apontar para repositórios, isto é, estruturar para facilitar o acesso.

As barreiras para que a Gestão do Conhecimento aconteça na Engenharia de *Software* encontram-se nas dificuldades de se criarem práticas para que o conhecimento existente em projetos passados seja disponibilizado da maneira adequada para ser reutilizado em outros projetos e a dificuldade na transformação do conhecimento tácito em explícito nos projetos de desenvolvimento de *software* (DE SOUZA, 2003)

A Gestão do Conhecimento na Engenharia de *Software* não pode estar centrada somente em ferramentas tecnológicas, pois envolve pessoas diferentes e assim, de acordo com KOMI-SIRVIÖ *et al.* (2002) é necessário organizar os processos para a Gestão do Conhecimento para capturar, armazenar e recuperar para reutilizar nos projetos de desenvolvimento de *software*. Essas iniciativas apontam para a necessidade de estruturação do conhecimento em termos de casos e experiências vivenciadas por equipes de desenvolvimento, de forma que possam ser reutilizadas em situações similares (KOMI-SIRVIÖ *et al.*, 2002).

A proposta de HENNINGER (1997) é um método baseado em casos para coletar e gerenciar o conhecimento no desenvolvimento de *software* o contexto organizacional, ou seja, refinar e manter o conhecimento na organização, a fim de não perdê-lo. Para

isso é necessário armazenar os casos com foco na reutilização e resolução de problemas, conhecendo a cultura organizacional e o processo de negócio envolvido.

Por outro lado, FALBO e NATALI (2002) propõem a Gestão do Conhecimento em ambientes de Engenharia de *Software* através de ontologia e agentes de *software*, onde as etapas de captura, pesquisa, recuperação, acesso, disseminação e manutenção estejam organizadas e com qualidade. Para isso, os autores propõem uma ferramenta chamada ControlQ baseada em ODE (Ambientes de Desenvolvimento baseados em Ontologias) para organizar nos ambientes de desenvolvimento à Gestão do Conhecimento por meio de ontologia.

Já ALVES e PESSÔA (2008) analisam a Gestão do Conhecimento (*Knowledge Model - KM*) sob o aspecto da melhoria de processos de desenvolvimento de *software* e encontram na literatura uma classificação quanto aos modelos de maturidade, dos quais se destacam três: *Knowledge Process Quality* (PAUKZEN e PERC, 2002), *KM Capability Assessment Model* (KULKARNI e FREEZE, 2004), e *Knowledge Journey e KM Maturity Model* (KPMG, 1999). As autoras identificaram os modelos que são baseados no CMM e compostos por níveis: *Siemens'KMMM*, *Infosys' KMMM (KM Maturity Model)*, *Paulzen Perc's Process Quality Model (KPQM)* e *Kulkarni e Freeze's Knowledge Management Capability Assessment Model (KMCA)*. A questão da qualidade é importante para a Gestão do Conhecimento em processos de desenvolvimento de *software*, pois um *software* tem que ter qualidade para resolver os problemas adequadamente.

Com o foco de maturidade para Gestão do Conhecimento, FENG (2006) propõe um modelo baseado no CMMI (*Capability Maturity Model Integration*) (CMMI, 2008). Esse modelo analisa o valor da Gestão do Conhecimento para o negócio, qual o investimento e o retorno, quais as áreas que sobressaem na organização, como está a Gestão do Conhecimento na organização em comparação com outras no mercado e quais as iniciativas estão sendo implementadas em organizações similares. A abordagem de FENG (2006) analisa o retorno do investimento, as mudanças, os esforços, os desafios e os benefícios da Gestão do Conhecimento em uma organização de *software*. FENG (2006) analisa que a implementação da Gestão do Conhecimento é um processo contínuo de trabalho diário e gerenciado pelos gestores.

A Gestão do Conhecimento também envolve a colaboração, que pode ser vista como uma aliada à integração da solução na empresa, segundo WANG *et al.*(2002).

WANG *et al.* (2002) desenvolveram uma proposta de modelo de conhecimento distribuído para atender a Gestão do Conhecimento para a engenharia colaborativa, onde o conhecimento é modelado e criando um processo com relações entre os envolvidos para apoiar o desenvolvimento.

A Gestão do Conhecimento então pode servir de apoio à Engenharia de *Software*, sendo esta um intenso contexto de conhecimento, pois durante o desenvolvimento novos conhecimentos são gerados pelo desenvolvedor para a resolução de um determinado problema, onde é necessário organizá-lo, disseminá-lo e reutilizá-lo com qualidade.

3.3. Gestão do Conhecimento nas Comunidades de Prática

O conhecimento humano evolui quando os indivíduos juntamente com seus pares discutem e validam suas informações e para isso se apóiam nas Comunidades de Prática, onde a partir de um tema comum aprimoram seus conceitos, tiram suas dúvidas, resolvem problemas de seu interesse.

Quando os membros dessas comunidades trocam ideias e compartilham conhecimento entre si, eles geram uma compreensão compartilhada a partir de informações aparentemente confusas e conflitantes. Esse processo acaba disseminando conhecimento pela organização, que passa do nível individual para consolidar-se na rede de conhecimentos da organização (NONAKA e TAKEUCHI, 1997).

WENGER *et al.* (2002) propõem sete princípios de gestão para uma efetiva Gestão do Conhecimento em Comunidades de Prática. São eles: projetar com foco na evolução, manter um diálogo entre as perspectivas interna e externa, convidar para diferentes níveis de participação, desenvolvimento tanto de espaços para comunidades públicas como privadas (espaços abertos e fechados), focar no valor, combinar familiaridade e entusiasmo e, criar um ritmo para a comunidade. Esses princípios ajudam a organização nas mudanças trazidas pelas Comunidades de Prática.

Assim, a Gestão do Conhecimento, de acordo com WENGER (2004) exige o contexto organizacional apropriado. É preciso ter processos mapeados (atividades e seus executores) para coordenar a Gestão do Conhecimento e para integrá-la em processos de negócio, tais como a tecnologia para os fluxos de informação, ligações interpessoais, e repositórios de documento, bem como as normas institucionais e culturais devem prestar atenção ao conhecimento. Para esta gestão é proposto um modelo “rosca”,

conforme a Figura 3, para a Gestão do Conhecimento, que envolve estratégia e desempenho, com o objetivo de entender como os três elementos da Comunidade de Prática representam o conhecimento estratégico e para melhorar a desempenho dos membros para aplicar o conhecimento no seu trabalho. A verdadeira Gestão do Conhecimento, segundo WENGER (2004), vista como a gestão de recursos estratégicos do conhecimento, é o processo pelo qual Comunidades de Prática apóiam todo o ciclo do modelo “rosca”, ou seja, Desempenho – domínios, comunidades, práticas – Estratégia – aprendizado, compartilhamento, organização.



Figura 3 - Modelo “rosca” de Gestão do Conhecimento (WENGER, 2004)

As Comunidades de Prática são muitas vezes vistas como uma forma de introduzir a Gestão do Conhecimento nas organizações, minimizando os custos e os riscos e aumentando o valor agregado às pequenas e médias organizações no apoio à disseminação do conhecimento DU PLESSIS (2008). O compartilhamento gerado por essas comunidades dentro de uma organização e a própria atuação delas pelas fronteiras internas existentes, influenciam no processo de utilização do conhecimento dentro das organizações. De acordo com WENGER (2006), o fato dos membros das comunidades serem de diferentes áreas, possuírem conhecimentos específicos diversos e um entendimento compartilhado, os apóia em saber o que deve ser relevante para comunicar e como expor o conhecimento de uma forma utilizável para a organização.

De acordo com KIMIECK (2002), com as Comunidades de Prática, as organizações deixam de ter a concentração de *expertises* sob um controle hierárquico, e passam a criar uma estrutura que considera comunidades informais como agentes ativos nos seus processos internos, e que integra a coordenação das competências chave em um quadro mais amplo da organização.

Os ambientes de Gestão do Conhecimento devem explicitar o conhecimento tácito dos membros das comunidades, fazendo com que eles utilizem o conhecimento armazenado ao mesmo tempo colaborando em uma interação contínua com a organização. Os membros dessas comunidades possuem criatividade e habilidade para inventar maneiras diferentes e fáceis para resolver seus problemas (TERRA, 2001). Essas comunidades então são uma fonte geradora de conhecimento de forma que a Gestão do Conhecimento organize, compartilhe e armazene todo esse conhecimento.

Tais características podem ser atribuídas também as Comunidades Virtuais de *Software Livre* que utilizam a forma de se organizar, evoluem o conceito de aprendizado e saem da organização para a sociedade com suas ferramentas desenvolvidas.

3.4. Gestão do Conhecimento nas Comunidades de *Software Livre*

O *Open Source* é considerado como um instrumento de Gestão do Conhecimento, segundo WIELAND (2004), pois o código fonte de programas é um meio para simbolizar e transmitir conhecimento. As condições do *Open Source* de trabalhar em comunidade, ter repositório central, possuir um meio de comunicação informal como portais e listas de discussão (*e-mail*), controle de acessos e mecanismos de busca flexíveis facilitam a Gestão do Conhecimento, ou seja, o conhecimento gerado é passado para a comunidade (WIELAND, 2004),

O princípio do desenvolvimento *Open Source*, segundo RAYMOND (2004) é aprender que o sigilo é inimigo da qualidade, sendo uma oposição ao tradicional desenvolvimento de *software* e que o trabalho caótico das comunidades em uma base é possível. Com isso é possível a contribuição de diversos participantes, complementando o desenvolvimento com seu conhecimento particular, e compartilhando o conhecimento.

Analisando a Gestão do Conhecimento em projetos *Open Source*, RUDZI e JONSON (2003) levantam os mecanismos presentes no desenvolvimento com foco na aquisição e compartilhamento do conhecimento. Eles partem das seguintes questões:

Como é a Gestão do Conhecimento entendida, e é possível analisar práticas de projeto a partir dessa perspectiva?; Quais são os aspectos da Gestão do Conhecimento do desenvolvimento tradicional do *software* e quais os que são relevantes para o *Open Source*?; Quais são os conhecimentos comuns relacionados com as práticas em projetos *Open Source*?; Quais são os aspectos das práticas da Gestão do Conhecimento em projetos *Open Source*? e, Como diferenciar o desenvolvimento *Open Source* de outros modelos de desenvolvimento, no contexto da Gestão do Conhecimento?.

RUDZI e JONSON (2003) propõem um modelo para analisar a Gestão do Conhecimento nos seguintes projetos de *open source*: Apache, GCC, Gnome, KDE, Desenvolvimento do *kernel* do Linux e Mozilla, observando as práticas: FAQs, Web Sites, *Howtos*, IRC, Sistemas de Monitoramento de Problemas, *Mailing Lists*, Weblogs e *Wikis*. A análise focou na visão de produto (acurácia e valor, dinâmica, validade, concorrência, confidencialidade, acessibilidade e papéis, categorização e mapeamento, busca e localização da *expertise*), a visão da comunidade (as quatro formas de criação de conhecimento, socialização, externalização, internalização e combinação) e as necessidades apoiadas pelas práticas mencionadas com foco na Engenharia de *Software*. As práticas são definidas como o conhecimento e a percepção de um praticante, e não somente as ferramentas usadas para realizar uma tarefa ou as atividades individuais realizadas (RUDZKI e JONSON, 2003).

Os autores RUDZKI e JONSON (2003) observaram cada uma das práticas segundo os critérios e a conversão do conhecimento, mas não existiu um aprofundamento para cada um deles. Por exemplo, uma vez que assuntos diferentes são tratados em canais diferentes de IRC, considerou-se que esta prática resolve o problema de categorização e mapeamento de conhecimento. Eles conseguem identificar pelas práticas que a Gestão do Conhecimento ocorre nos projetos, que o ambiente é propício para o desenvolvimento de *software*, que existe a criação e a transferência de conhecimento focados na conversão desse conhecimento, e como é apoiada a criação e o compartilhamento em projetos *Open Source*.

RUDZKI e JONSON (2003) afirmam que as práticas apoiam o desenvolvimento *Open Source* e que permitiram reconhecer os conhecimentos e capacidades relacionadas com os aspectos da Engenharia de *Software*, ou seja, as práticas são tratadas como a forma de suporte ao desenvolvimento.

Desta forma, apesar dos atores afirmarem que estas práticas analisadas atendem a Gestão do Conhecimento, são focadas apenas na parte da conversão do conhecimento e não em mecanismos para prover o ciclo da Gestão do Conhecimento ou a estruturação do conhecimento.

SOWE *et al.* (2007) pesquisaram sobre a dinâmica do compartilhamento de conhecimento técnico no projeto Debian, realizado por meio das listas de discussão (*e-mail*). O estudo foi feito por meio de métricas e foi discutida a tendência de compartilhamento nas listas, a correlação entre as atividades de postagem e resposta e a natureza da auto-organização das atividades individuais de compartilhamento de conhecimento. Os participantes assumem dois tipos de papéis no modelo proposto, o que busca o conhecimento e o que provê o conhecimento, e são identificados dois momentos do ciclo da Gestão do Conhecimento: onde o conhecimento procurado já está armazenado no repositório e quando o conhecimento não está armazenado. Apesar de ser uma avaliação bastante pontual (um projeto e um tipo de interação), os resultados mostram a forma como o conhecimento de desenvolvimento em projetos e trocado pelas listas de *e-mail*, é criado e compartilhado, isto é, a externalização (a passagem do conhecimento técnico) e a internalização (a aquisição do conhecimento técnico) do conhecimento baseada na troca de conhecimento entre os indivíduos que procuram e os que provêem conhecimento.

A Gestão do Conhecimento nas Comunidades Virtuais de *Software Livre / Open Source* é de grande valia para a reutilização do conhecimento em projetos de desenvolvimento futuros, pois a reutilização tanto de linhas de código, como de componentes, visam agilizar o processo de desenvolvimento e aumentar a qualidade do *software* produzido (VON KROGH *et al.*, 2005). De acordo SOWE *et al.* (2007) e VON KROGH *et al.* (2005), não existe uma estruturação adequada que apóie as Comunidades Virtuais de *Software Livre* a controlar o conhecimento gerado e compartilhado.

Para organizar o conhecimento e torná-lo acessível, a Gestão do Conhecimento tem um papel importante por meio do seu ciclo (KING *et al.*, 2002). Para que identificar as necessidades das Comunidades Virtuais de *Software Livre* neste contexto foi realizado um Estudo Exploratório por meio da análise dos ambientes e um levantamento junto aos participantes, com foco nas interações realizadas para o desenvolvimento de *software* e a resolução de problemas.

A literatura indica que as Comunidades Virtuais de *Software* Livre criam conhecimento quando realizam o desenvolvimento *Open Source*, mas a forma como organizá-lo para que os participantes possam encontrá-lo e reutilizá-lo ainda não foi explorada totalmente (DE SOUZA, 2003; KING *et al.*, 2002; RUDZKI e JONSON, 2003; SOWE *et al.*, 2007;e VON KROGH *et al.*, 2005).

Capítulo 4 - Estudo Exploratório: avaliação das comunidades

Este capítulo apresenta as etapas do Estudo Exploratório realizado nas Comunidades Virtuais de *Software* Livre para entendimento de seu contexto, características e necessidades. A pesquisa foi composta por duas etapas: análise das comunidades e *Survey*. Os resultados do Estudo Exploratório mostram uma visão externa da comunidade, com o intuito de entender e reconhecer o ambiente das comunidades. Já os resultados da aplicação de um questionário aos membros de comunidades mostram uma visão interna através da opinião dos participantes que estão envolvidos diretamente nas interações.

4.1. Metodologia da Pesquisa

Uma pesquisa exploratória estabelece os critérios, métodos e técnicas para a elaboração de uma pesquisa. A “pesquisa é o conjunto de investigações, operações e trabalhos intelectuais ou práticos que tenham como objetivo a descoberta de novos conhecimentos, a invenção de novas técnicas e a exploração ou a criação de novas realidades” (KOURGANOFF, 1990). Para realizar o Estudo Exploratório foram definidos critérios de análise para as comunidades, houve a inserção em cada comunidade assumindo o papel de participante informal e depois foi realizado um *Survey* com os participantes. Por meio desse Estudo Exploratório foi possível perceber como os participantes interagem, como é realizada a divulgação de tecnologia livre, como acontece o desenvolvimento de projetos e como os ambientes das comunidades funcionam do ponto de vista do ciclo do conhecimento.

No caso desta dissertação, era fundamental conhecer as interações nas Comunidades Virtuais de *Software* Livre em duas visões: uma externa (pelo pesquisador) e outra interna (pelos participantes das comunidades), ou seja, um aprofundamento para entender o problema existente nas interações de forma que a

geração do conhecimento seja estruturada, pois não foram encontrados estudos que apresentassem resultados objetivos com este foco.

O objetivo do Estudo Exploratório foi avaliar como as interações ocorrem entre os participantes das Comunidades Virtuais de *Software* Livre. Essa pesquisa foi realizada em duas fases: (i) análise de ambientes de suporte a comunidades de *Software* Livre e (ii) levantamento de impressões de participantes das comunidades através de um questionário.

A primeira fase, análise de ambientes de suporte a comunidades de *Software* Livre, foi composta pelas seguintes etapas: identificação das Comunidades Virtuais de *Software* Livre com participação significativa e projetos de porte na Internet; acesso às ferramentas usadas; pequenas inserções nestas comunidades para observar como funcionam; definição de categorias específicas para análise; avaliação de cada comunidade de acordo com os critérios definidos e organização dos resultados em tabela comparativa.

Na segunda fase, levantamento de impressões de participantes das comunidades através de um questionário visava a busca das opiniões dos participantes dessas comunidades por meio de questões objetivas com foco na Gestão do Conhecimento. Depois foi realizada a elaboração de gráficos e a interpretação dos resultados.

A pesquisa teve foco na identificação dos mecanismos existentes para apoio à Gestão do Conhecimento e pontos de deficiência, de forma a prover insumo para desenvolver uma proposta de um modelo para estruturar o conhecimento existente nas interações se apoiando na Gestão do Conhecimento para as Comunidades Virtuais de *Software* Livre em projetos de desenvolvimento.

4.2. Análise das Comunidades Virtuais de Software Livre

4.2.1 Identificação de Comunidades

As comunidades foram selecionadas com o intuito de abranger os diversos focos existentes, seja de desenvolvimento de projetos, filosofia ou divulgação de tecnologia. Inicialmente, as comunidades escolhidas foram: *Software* Livre, BR-Linux, VivaOLinux, LinuxSecurity, Dicas-L, NoticiasLinux, Under-Linux, Best-Linux, *Source Forge.net*, *Fresh Meat.net*, Código Livre, *Free Software Foundation*, *Open Source Initiative*, Centro de Computação da Unicamp, Tigris, Google Code, Kodres.

As Comunidades Virtuais de *Software* Livre selecionadas são então caracterizadas por serem Comunidades de Prática (CoP), tendo o interesse no assunto de *Software* Livre e trocando experiências, mas além disso por seus participantes trabalharem colaborativamente e voluntariamente na resolução de problemas particulares compartilhados na comunidade (REIS, 2003; e SOWE *et al.*, 2007), os quais acabam por desenvolver ferramentas ou serviços, inovarem tecnologicamente (O'REILLY, 1999) e gerarem um novo conhecimento tecnológico. Essas comunidades geram colaborativamente um produto final, seja uma ferramenta ou uma linha de comando, diferentemente das Comunidades de Prática que possuem somente um tema de discussão em comum.

As comunidades se utilizam de ferramentas como: Linux, ASP, ASP.NET, C/C++, CGI/Perl,ColdFusion, Delphi, JSP, PHP, Python, Visual Basic, XML, Rau-Tu, Nou-Rau, Java, Ajax, Lua, Zope, PostgreSQL, MySQL.

O Linux é atualmente o sistema operacional mais utilizado pelas Comunidades Virtuais de *Software* Livre e é o precursor da ideologia de *Software* Livre, de acordo com as ideias de Stallman, (GNU, 2008).

As ferramentas como: ASP, ASP.NET, C/C++, CGI/Perl,ColdFusion, Delphi, JSP, PHP, Python, Visual Basic, XML, Rau-Tu, Nou-Rau, Java, Ajax, Lua, Zope são utilizadas pelos participantes no desenvolvimento e como tema de discussão.

Os bancos de dados, PostgreSQL e MySQL, são pacotes que podem ser baixados pelas suas páginas e utilizados sem restrições na construção de repositórios.

O foco da análise das comunidades foi levantar os tipos de interações que ocorrem nas comunidades, visando à disseminação do conhecimento. A identificação da comunidade está no seu objetivo, na ferramenta que ela apóia e no endereço da sua página na Internet (*site*). Os membros podem ser profissionais, usuários, estudantes ou professores interessados. A comunidade é geralmente composta por um moderador e diversos participantes, onde todos contribuem de alguma forma para alcançar o objetivo da comunidade.

4.2.2 Acesso às ferramentas usadas pelas comunidades

As ferramentas utilizadas pelas comunidades têm foco na comunicação entre os participantes das comunidades, na divulgação de tecnologia, no desenvolvimento de

software, na discussão de problemas, ou seja, na interação dos participantes para que resolver seus problemas individuais e evoluir a tecnologia livre.

Foi observado que entre as formas utilizadas nas interações, destacaram-se: as notícias, as listas de *e-mails*, os fóruns, os repositórios de *softwares*. Essas formas de interações são assíncronas, ou seja, os participantes divulgam informações, disponibilizam suas questões, contribuem no desenvolvimento de ferramentas a qualquer momento. Dessa forma, eles colaboram e cooperam uns com os outros, sem estarem *on line*, demonstrando que a distância entre eles e seus compromissos pessoais não influem na sua participação nas comunidades.

O acesso às ferramentas acontece de acordo com o interesse do participante e a análise do administrador da comunidade, visto que de cada participante possui um ou mais perfis. De acordo com o seu perfil, o participante acessa uma área e atua na comunidade.

O participante precisa de uma identificação que é a sua conta (*login*) e ao acessar o ambiente da comunidade de forma identificada, ele fica “logado” e acessa as ferramentas disponíveis para atuar na comunidade.

Enfim, existe uma integração entre os interesses dos participantes e os da comunidade. Isso demonstra que existe nas comunidades uma organização informal para elas existirem, mas essa organização acontece informalmente e com respeito ao outro participante, não existindo cobranças ou uma hierarquia rígida.

4.2.3 Inserções nas comunidades

As inserções nas comunidades, nesse Estudo Exploratório, aconteceram com a participação da responsável pela pesquisa em:

- listas de discussão;
- respostas nos fóruns;
- abertura de discussões nos fóruns;
- divulgação de notícias;
- análise do *site* (ambiente ativo na Internet).

As inserções acontecem sempre com o participante “logado” no ambiente, ou seja, identificado e com seu perfil como limitador para sua atuação.

Para participar das listas de discussões (*e-mail*), foi necessário disponibilizar um *e-mail* e entender as regras específicas para envio de mensagens. As respostas nos

fóruns aconteceram a partir da escolha de uma discussão e a inserção da resposta. Já em relação à abertura de discussões nos fóruns, é necessário que o problema a ser discutido seja um tema de interesse da comunidade (um sistema operacional, uma tecnologia, um *software*), pois se uma questão é aberta fora dos interesses ela não é respondida por nenhum participante e o administrador faz o alerta. As notícias são a forma de divulgação de tecnologia livre e passam pela aprovação do administrador da comunidade para serem divulgadas. A análise do ambiente aconteceu acessando todas as funcionalidades, verificando quais eram seus objetivos na comunidade.

Essas inserções serviram para entender como acontece o dia a dia de um participante informal, como as interações aconteciam, como os projetos funcionavam.

Foi identificado que o administrador é quem gerencia toda a comunidade e permite ou não que uma notícia seja divulgada ou um fórum fique ativo. Os fóruns também são controlados por moderadores que são os responsáveis por aquela discussão. O controle existe informalmente nessas comunidades, isto é, a “net etiqueta” é a forma de controlar o comportamento dos participantes nas comunidades.

4.2.4 Definição de categorias de análise

A análise realizada nas comunidades teve foco na identificação de atividades do ciclo da Gestão do Conhecimento: captura, representação, recuperação, armazenamento, compartilhamento e reuso de conhecimento. As seguintes categorias foram definidas para análise das comunidades:

- **Mecanismos de apoio à interação e discussão:** A comunicação é um dos alicerces para que a comunidade promova o conhecimento, já que a troca de informações é intensa e dinâmica e precisa atingir diversos participantes dispersos geograficamente. Os mecanismos de apoio à interação e discussão se referem às ferramentas usadas pelos membros da comunidade para se comunicarem e trocarem informações e questões sobre os projetos que estão desenvolvendo. Exemplos de mecanismos de interação e discussão são mensagens eletrônicas, *chats*, fóruns, etc. Segundo YAMAUCHI *et al.* (2000), em sua pesquisa sobre como a colaboração acontece nos projetos de *Software Livre*, as aplicações utilizadas tais como: CVS, TO-DO Lists e Mailing Lists, apóiam na coordenação do projeto e na comunicação. A pesquisa de SOWE (2007)

no projeto Debian mostra que as listas de discussão permitem o compartilhamento do conhecimento entre os participantes. No ciclo da Gestão do Conhecimento está relacionado a captura, representação, armazenamento e compartilhamento do conhecimento. A captura acontece quando o ambiente guarda uma informação disponibilizada pelo participante, isto é, uma mensagem enviada pela lista. Enquanto a representação são as ideias do participante explicitada na mensagem. O armazenamento acontece na própria mensagem da lista com a resposta de outro participante. E o compartilhamento realiza-se com a abrangência da lista, ou seja, todos os participantes cadastrados recebem as mensagens e o conhecimento embutido.

- **Modelo de estruturação das interações:** O modelo de estruturação das interações permite que a argumentação em uma discussão seja organizada e as mensagens trocadas categorizadas. Modelos de argumentação tais como IBIS (RITTEL e KUNZ, 1970), tratam as mensagens como: Questão, Posição e Argumento, em que a questão está ligada a uma ideia (posição) e a uma justificativa (argumento). Esse tipo de modelo pode ajudar no entendimento da discussão, pois ficam evidenciadas as relações entre as partes do diálogo acrescentadas por cada participante. No ciclo da Gestão do Conhecimento está relacionado à representação do conhecimento.
- **Mecanismos de busca de informação:** A busca por um determinado assunto ou artefato é frequente nas Comunidades de Prática, uma vez que as interações são em grande parte assíncronas. A forma mais simples é a busca por palavras-chave. Outras formas mais sofisticadas requerem taxonomias ou outros modelos de indexação associados aos itens que se deseja encontrar, como o modelo de indexação automática desenvolvido por SALTON (1983). Em relação ao ciclo da Gestão do Conhecimento está relacionado à recuperação do conhecimento.
- **Existência de repositório estruturado:** Um repositório (PFANFESELLER, PFANFESELLER, KROTH, 2001) estruturado é considerado o local (banco de dados) onde são armazenados os artefatos

de interesse da comunidade, de forma categorizada ou organizada para facilitar a busca e a sua recuperação. As comunidades podem ter seus próprios repositórios com artigos, notícias, dicas e o código fonte das ferramentas desenvolvidas. Os repositórios criam uma Memória Organizacional (CONKLIN, 1992) que possibilita o armazenamento do conhecimento para posterior utilização. Repositórios são importantes na Gestão do Conhecimento, pois permitem a captura, armazenamento e recuperação do conhecimento (ABECKER e DECKER, 1999). Em Comunidades Virtuais de *Software* Livre, parte do conteúdo dos repositórios são os produtos desenvolvidos (em forma de código fonte ou outros artefatos de *software*), além dos registros das interações, artigos, notícias, dicas e outros documentos disponibilizados pelos membros. Em relação ao ciclo da Gestão do Conhecimento representa o armazenamento do conhecimento.

- **Mecanismos de recomendação:** Os mecanismos de recomendação funcionam como divulgação de um determinado tema, em que os artefatos compartilhados podem ser apontados para um participante de acordo com seu interesse específico. Recomendações podem ser feitas pelos participantes ou de forma automática pelo sistema. Segundo pesquisas de RESNICK e VARIAN (1997) e MOTTA (1999), os sistemas de recomendação relacionam o artefato a ser recomendado, o mecanismo que realiza a recomendação e a confiabilidade envolvida, como uma forma de relacionamento social entre o que recomenda e quem recomenda. A maioria das comunidades permite a recomendação (não automática) de notícias. Algumas inserem o mecanismo de reputação do participante, o qual é a forma de identificar a contribuição do participante de forma positiva ou negativa na comunidade. Para o ciclo da Gestão do Conhecimento esse critério representa o compartilhamento e reuso do conhecimento.
- **Artefatos de conteúdo:** Os artefatos de conteúdo se referem a todos os itens que contribuem para a geração e disseminação de conhecimento na comunidade, ou seja, wiki, artigos, tutoriais, dicas, configurações, scripts, pergunta sobre tecnologia, documentação de um projeto, os

fóruns dos projetos, blog, as bases de conhecimento com informações sobre projetos e o próprio código fonte. Esta categoria tem o objetivo de identificar que tipos de elementos são compartilhados, a quantidade e a qualidade na organização desses artefatos. A quantidade é analisada para verificar o quanto de material compartilhado existe disponível nas comunidades (pouca, média ou muita). Por exemplo, a comunidade VivaOLinux possui em torno de 3065 artigos sobre Linux, o que pode ser considerado como uma grande quantidade de material somente sobre um determinado assunto. Para o ciclo da Gestão do Conhecimento esse critério é relacionado à representação do conhecimento na forma como é explicitado seu conteúdo.

- **Mecanismos de divulgação de tecnologia livre:** Os mecanismos de divulgação de tecnologia livre são os meios usados pela comunidade para propalar a tecnologia e a própria comunidade. Podem ser: notícias, informações sobre os projetos de desenvolvimento. Esses mecanismos servem como indicadores de como as comunidades difundem o que produzem de conhecimento gerado internamente. Para o ciclo da Gestão do Conhecimento esse critério representa o compartilhamento.

4.2.5 Avaliação das comunidades: Resultados obtidos

A Tabela 1 apresenta os resultados da análise de cada um dos itens descritos para as comunidades listadas. As Comunidades Virtuais de *Software* Livre possuem seus objetivos e um público-alvo bem definidos, o que garante muitas vezes que os seus participantes atuem ativamente. Nas Comunidades Virtuais de *Software* Livre o conhecimento é trocado dinamicamente por meio de mensagens síncronas ou fóruns de discussão, ou seja, quando um participante envia uma mensagem perguntando, por exemplo, como instalar um *software*, e outro responde, descrevendo cada uma das etapas desta instalação. Essa prática pode apoiar o participante no momento de absorver o conhecimento, porém em outros momentos, os outros participantes com dúvidas semelhantes podem ter dificuldade em encontrar o resultado desta interação, criando uma sobrecarga para a comunidade.

O objetivo das comunidades é totalmente voltado para o compartilhamento da tecnologia livre, destacam-se em relação a desenvolvimento de projetos as comunidades

SourceForge.net, FreshMeat.net, Tigris, Google Code e Koders. Essas comunidades trabalham com projetos de desenvolvimento e possuem um repositório do projeto.

A comunicação é um ponto importante e acontece em grande parte por meio das listas de discussão que divulgam os eventos, as notícias e as dicas de tecnologia. Este mecanismo é essencial em todas as comunidades para que os participantes estejam cientes do que acontece no cenário de *Software* Livre e na divulgação do conhecimento. As interações são feitas na maioria das vezes na forma de fóruns e salas de bate-papo. No entanto, observa-se que as mensagens não são estruturadas e existe uma classificação genérica por questão. A falta de estruturação nas mensagens e classificação organizada das informações disponibilizadas muitas vezes dificulta a identificação do conhecimento buscado por um participante. As Comunidades Virtuais de *Software* Livre que se destacam na análise são: VivaOLinux e Under-Linux, pois se organizam de forma a facilitar a interação separando por assunto uma notícia a ser comentada ou uma discussão em fórum.

O modelo de estruturação das interações é composto da questão e da resposta, na sua maioria. As comunidades Under-Linux e a SourceForge.net introduzem o conceito de citações, ou seja, além das questões e respostas é possível inserir citações nas respostas, o que gera uma interação um pouco mais estruturada e organizada, facilitando que outros participantes, ao acessarem posteriormente, tenham um entendimento melhor do que aconteceu durante a interação.

O mecanismo de busca de informação disponível nas comunidades pode não estar adequado às necessidades do participante, pois nem sempre o resultado obtido com a pesquisa através da palavra-chave corresponde à busca ou pode trazer diversas informações não relevantes. Não foram identificados mecanismos de busca mais sofisticados nas comunidades, nem uso de taxonomias ou ontologias para categorizar conhecimento. Todas as comunidades realizam a busca por meio de palavras-chave.

Os repositórios nem sempre estão estruturados de forma clara nas comunidades, como por exemplo, na Under-Linux, cuja organização do repositório acontece por assunto e por mês/ano. A maioria das comunidades analisadas possui repositórios onde são inseridos os produtos dos projetos desenvolvidos e artefatos sobre tecnologia livre, ou seja, quando um projeto é inserido na comunidade, é criada uma pasta virtual na comunidade para que seus artefatos sejam disponibilizados. Nessas comunidades, o conhecimento é armazenado em pastas separadas por assunto e recuperado por meio de

palavras-chaves, o que demonstra que a necessidade da existência de um repositório. As pastas de arquivos são criadas pelos administradores de acordo com o objetivo e a necessidade da comunidade e não podem ter seus nomes (assuntos) alterados pelos participantes.

A recomendação é a forma de indicar a outro participante da comunidade um assunto de interesse e também pode ser utilizada como um multiplicador de conhecimento. Os mecanismos de recomendação encontrados no estudo são apenas utilizados de forma a enviar notícias e comentários para outros participantes. As comunidades Under-Linux, VivaOLinux e FreshMeat.net realizam esse tipo de recomendação. As recomendações são feitas manualmente pelos participantes, e não existem de forma automática.

Os artefatos de conteúdo são utilizados para guardar, de forma eletrônica, o conhecimento e também divulgá-lo entre os demais participantes da comunidade, entre eles, destacam-se os artigos, as dicas, wikis, blog, documentos e o código fonte. As comunidades Google Code e Koders utilizam o conceito de “base de conhecimento” para guardar os artefatos, ou seja, um repositório com os códigos-fonte dos aplicativos.

Os mecanismos de divulgação de tecnologia são a forma de compartilhar a tecnologia entre os membros da comunidade e são destaque as notícias. Em particular a comunidade SourceForge.net possui apenas as informações sobre os projetos que acontecem como forma de divulgação.

Um conceito que ainda é pouco explorado formalmente nas comunidades é o da reputação, ou seja, como o participante é conceituado explicitamente dentro de sua comunidade. Esse conceito é mais utilizado como uma forma de mérito informal entre os participantes e nem sempre fica evidente na comunidade para todos. As comunidades Under-Linux e Source Forge explicitam isso por meio do mecanismo de recomendação, ou seja, quanto mais recomendação o participante realiza mais reputação ele obtém. Já a VivaOLinux usa o conceito de forma a classificar a atuação do participante com as suas respostas, ou seja, uma classificação (*ranking*) é criada onde quem mais atua na comunidade ganha uma posição melhor.

Tabela 2 - Análise das Comunidades Virtuais de Software Livre

ANÁLISE	UNDER-LINUX	VIVAOLINUX	SOURCE FORGE.NET	FRESH MEAT.NET	TIGRIS	GOOGLE CODE	KODERS
Objetivo	Compartilhamento de conhecimento e apoio a projetos de <i>Software Livre</i> .	Desenvolvimento de aplicativos Linux	Desenvolvimento de <i>Software Livre</i>	Desenvolvimento de <i>Software Livre</i>	Desenvolvimento de <i>Software Livre</i>	Disponibilizar softwares desenvolvidos pelo Google de forma livre	Desenvolvimento de <i>Software Livre</i> .
Mecanismo de apoio à interação e discussão	Lista de discussão, Fóruns, Blogs	Lista de discussão, Fóruns, chats, enquetes	Lista de discussão, Fóruns, Blog	Lista de discussão, Chat, Forum	Lista de discussão, Fórum	Lista de discussão, Fórum	Lista de discussão, Fórum, Blog
Modelo de estruturação das interações	Questão, Respostas e Citações referentes a alguma resposta com uma nova resposta.	Questão e Respostas	Questão, Respostas e Citações	Questão e Respostas, porém a visualização depende do seu perfil na comunidade	Questões e Respostas	Questões e Respostas (ao autor e ao participante do fórum)	Questões e Respostas
Mecanismos de busca de informação	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves	Palavras-chaves
Repositório estruturado	Sim, repositório de código fonte e de documentos	Sim, repositório de documentos com categorização	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos	Sim, repositório de código fonte e de documentos
Mecanismos de recomendação	Sim de notícias com comentários e possui o conceito de reputação	Sim de artigos, conceitos (notas) e comentários	Não possui, somente possui o conceito de reputação	Sim de artigos com comentários.	Não	Não	Não mas tem um sistema de <i>feedback</i> sobre o <i>software</i> desenvolvido.
Artefatos de conteúdo	Wiki, Artigos, Tutoriais, Dicas	Artigos, Dicas, Configurações, Scripts, Perguntas com respostas sobre tecnologia	Código fonte, Wiki, Documentação do projeto	Código fonte, estatísticas	Fórum dos projetos, documentos	Blog, Wiki do projeto, vídeos, congressos, Google Code University (base de conhecimento)	Base de conhecimento com informações sobre projetos de desenvolvimento.
Mecanismo de divulgação de tecnologia livre	Notícias	Notícias	Informações sobre os projetos de desenvolvimento	Notícias	Notícias	Notícias	Notícias

O Estudo Exploratório identificou que os ambientes que apóiam as comunidades precisam se estruturar mais de forma a facilitar o armazenamento e a busca do conhecimento. Nas Comunidades Virtuais de *Software* Livre existe conhecimento de forma desestruturada e a colaboração acontece de forma explícita entre os participantes. O ciclo do conhecimento, a transformação de explícito em tácito e vice-versa, fica evidente, porém é necessário um processo de estruturação para que a Gestão do Conhecimento possa ser apoiada adequadamente.

4.3. Survey: Levantamento por meio de Questionário

Um questionário desenvolvido em Plone, que é um Sistema de Gerenciamento de Conteúdo (CMS), (PLONE,2009), foi disponibilizado no *site* do Projeto COLMÉIA (www.uniriotec.br/padct). Cento e setenta e sete (177) pessoas responderam o questionário, porém foram descartadas quarenta e seis (46) respostas devido ao fato de não terem identificado a comunidade de que faziam parte. O link foi enviado para as Comunidades Virtuais de *Software* Livre brasileiras: VivaOLinux, BR-Linux, Dicas-L, BROffice, Ubuntu, Debian, por meio de notícias e lista de *e-mails*, com o objetivo de mensurar o embasar o Estudo Exploratório.

As questões referentes à identificação pessoal do participante eram opcionais (nome, *e-mail*, cidade, formação, comunidade) e as demais questões eram obrigatórias.

As perguntas do questionário estavam relacionadas às seguintes questões:

- região do Brasil em que os participantes se encontravam;
- a formação acadêmica dos participantes;
- comunidades que participaram da pesquisa;
- papel que o participante exerce na comunidade;
- formas de interação entre os participantes;
- tipo de material que costuma-se compartilhar;
- quantidade de material;
- onde é feito o armazenamento dos artefatos;

- grau em que as mensagens trocadas são capazes de responder as questões discutidas;
- grau de facilidade para encontrar itens de conhecimento;
- grau de facilidade para indicar um material;
- grau de confiabilidade nas informações disponibilizadas na comunidade.

A expectativa do questionário era entender e mensurar a visão dos participantes das comunidades, assim como confrontar com a análise dos ambientes realizada. O questionário se encontra no Apêndice A dessa dissertação. Já entre as comunidades que os participantes respondentes fazem parte, destacam-se: VivaOLinux, Ubuntu, Debian, BR-Linux, Dicas-L e BROffice. Ressalta-se que a pesquisa foi divulgada entre comunidades brasileiras. A seguir, segue uma análise de cada uma das questões.

a) Cidade/Estado onde mora.

Os participantes encontram-se principalmente nos estados de: São Paulo, Minas Gerais, Rio de Janeiro e Paraná. Assim por região, os participantes estão mais concentrados nas regiões Sudeste e Sul, conforme a Figura 4.

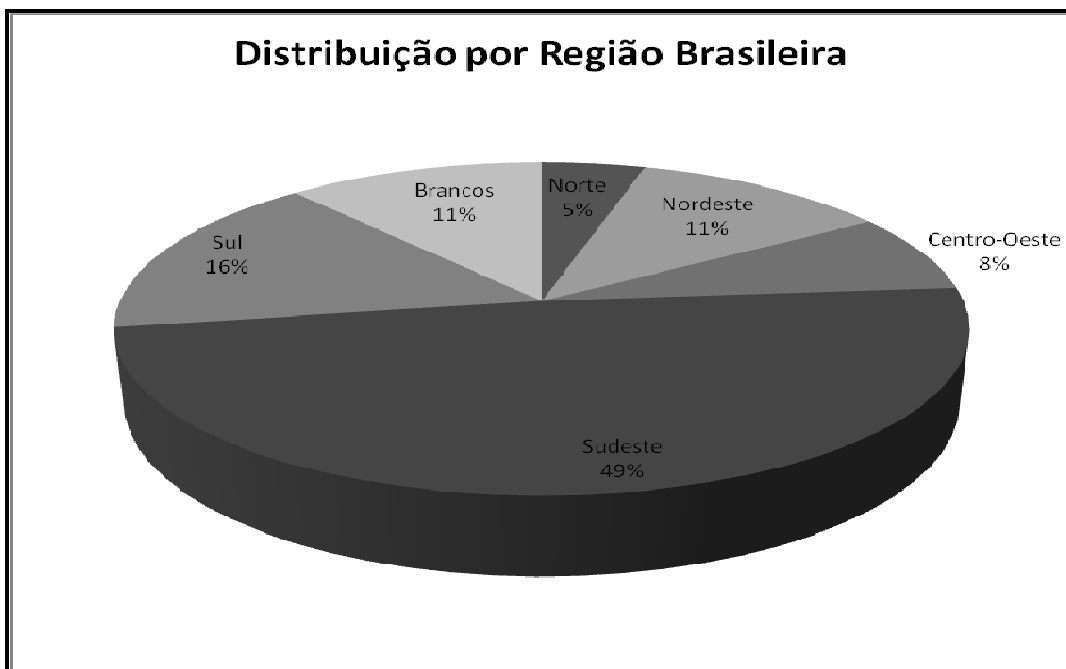


Figura 4 – Gráfico Regiões do Brasil

b) Formação Acadêmica (Curso Técnico, Graduação, Pós-Graduação, Mestrado, Doutorado).

A formação acadêmica dos participantes de Comunidades Virtuais de *Software* Livre que se destaca é a Graduação e seguida dos Cursos Técnicos e Pós-Graduação, conforme a Figura 5. Com isso, verifica-se que os participantes possuem uma especialização definida e grau de escolaridade.

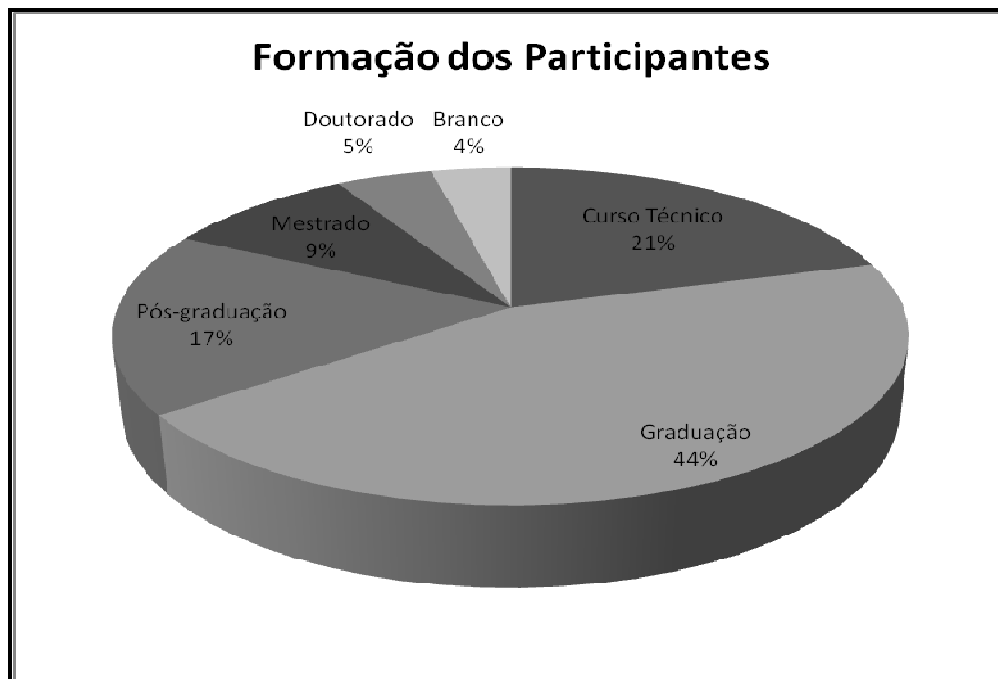


Figura 5 – Gráfico Formação dos Participantes

c) Comunidade Virtual de Software Livre / Open Source que faz parte.

Os participantes respondentes fazem parte das comunidades: VivaOLinux, Ubuntu, Debian, BR-Linux, Dicas-L, BROffice, Fedora, Gnome, LimeSurvey, PSL-Brasil, Openswan-BR, Projeto ACBr, Linux, GUS-BR, ArchLinux, Drupal-br, GNU, Gentoo, ASOLBR, Crux Linux, MandrivaBrasil.org, GeoServer, Inkscape Brasil, Linux Acessível, LDP-BR, GTK, JUG, Phytion, Plone, Zope, Netbeans.org, Mysql, PHP. Com destaque principal para a VivaOLinux, Ubuntu, Debian, Dicas-L, Fedora.

d) Papel na comunidade (Administrador, Moderador, Coordenador, Desenvolvedor, ou Participante Informal).

O administrador é o responsável pela organização da comunidade, por aprovar projetos, aprovar solicitações de usuários e responder a perguntas de usuários. O moderador é o responsável por mediar interações, aprovar artigos, divulgar notícias. O coordenador tem o papel de controlar os projetos em desenvolvimento e é normalmente um participante de confiança do administrador. O desenvolvedor é um participante que contribui em um projeto com o desenvolvimento de um produto para a comunidade, normalmente é um especialista de uma ferramenta. O participante informal é aquele que quer deseja aprender e contribuir com a comunidade, enviando artigos ou notícias sobre tecnologia livre.

Os participantes das comunidades possuem papéis definidos e destaca-se o participante informal, isto é, o participante que atua de forma generalizada onde a comunidade gere conhecimento para ele (TAURION, 2004), de acordo com a Figura 6.

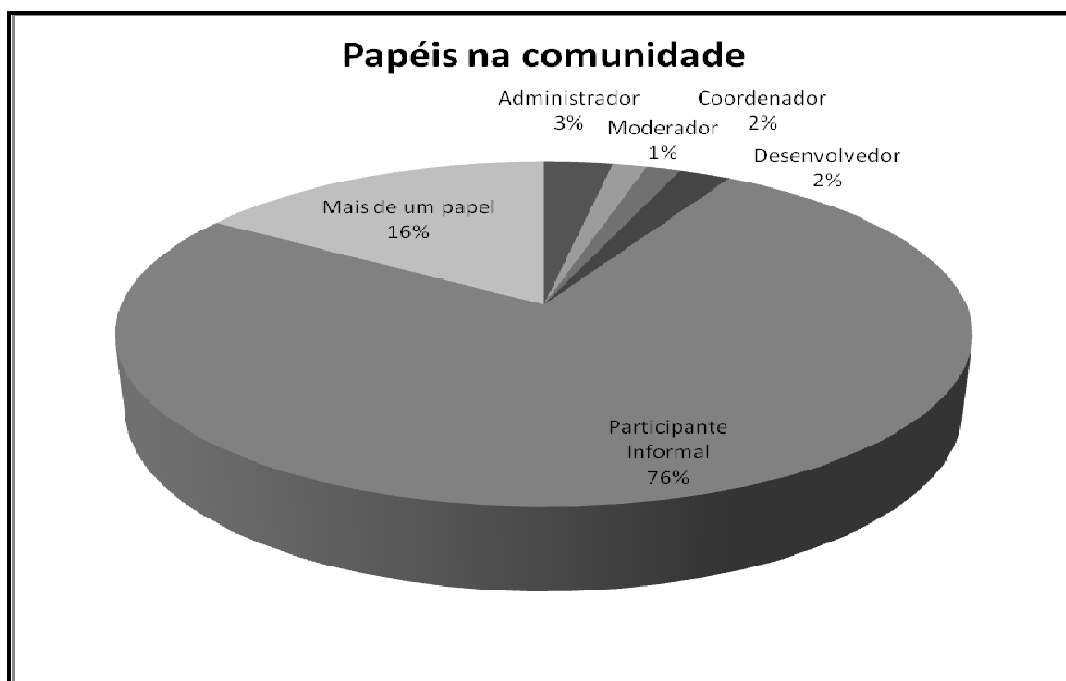


Figura 6 – Gráfico Papéis na comunidade

e) Forma de interação com outros participantes (Fórum, Listas de discussão, Chat, Blog, Outras).

A interação entre os participantes acontece geralmente por meio de mais de uma ferramenta, ou seja, IRC, MSN, ICQ, Wiki, *e-mails* e também fóruns e listas de discussão, como mostra a Figura 7. Já as salas de bate-papo (*chat*) são menos utilizadas

do que os fóruns e as listas de discussão, mostrando que nem sempre as interações acontecem em tempo real.

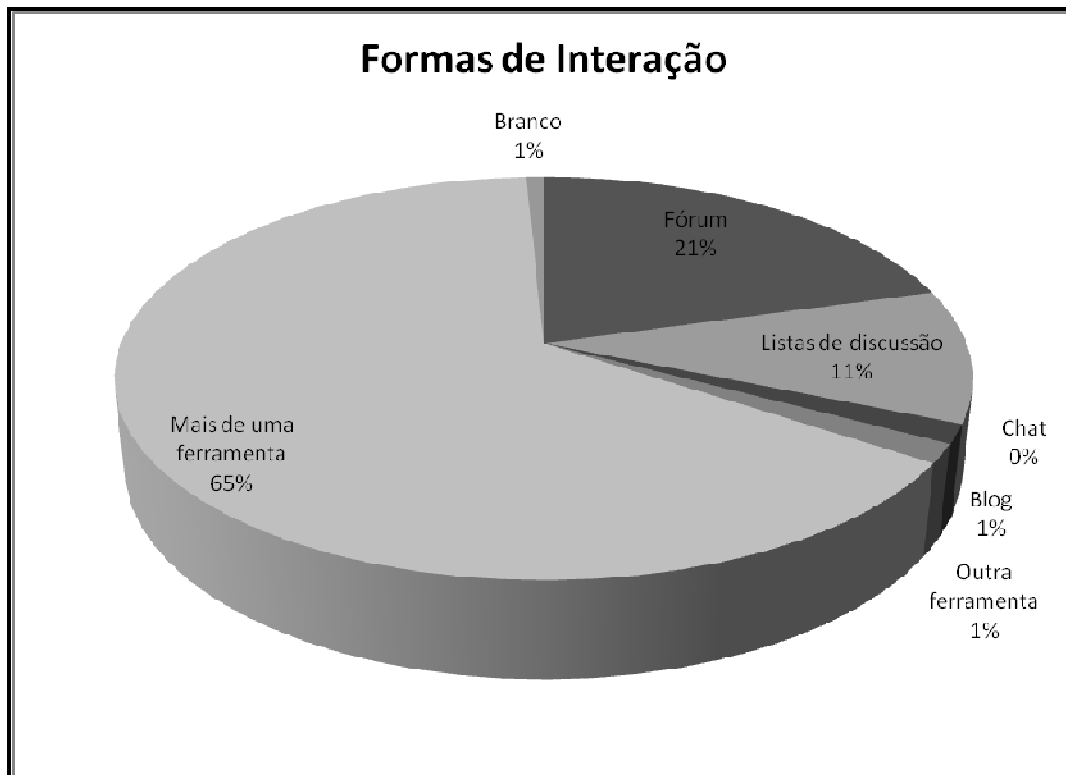


Figura 7 – Gráfico Formas de interação

f) Tipo de material compartilhado entre os participantes (Artigos, Notícias, Manuais, Código fonte, Comentários sobre determinada tecnologia, Outro).

O tipo de material compartilhado nas comunidades tem como destaque os artigos e notícias divulgadas pelos seus participantes e, em seguida, os manuais e o código fonte gerado em projetos, conforme a Figura 8. O código fonte é considerado um produto final dessas comunidades, pois as comunidades são focadas em desenvolvimento, mas em termos de compartilhamento, pela pesquisa realizada, o código fonte não é tão compartilhado quanto os artigos, as notícias e os manuais.



Figura 8 – Gráfico Tipo de material compartilhado nas comunidades

g) Quantidade de material compartilhado na comunidade (Muita quantidade, Média quantidade, Pouca quantidade ou Nenhum material é compartilhado).

Em relação à quantidade de material compartilhado foi identificado que existe nas comunidades uma grande quantidade. Na comunidade VivaOLinux, existem em torno de 3776 artigos, 636 arquivos de configuração, 3295 scripts e 6744 dicas. Assim, somente nessa comunidade existem cerca de 14451 documentos, considerando-se então uma grande quantidade.

Desta forma, há indícios de que existe muito conhecimento sendo compartilhado, como mostra a Figura 9.



Figura 9 – Gráfico Quantidade de material compartilhado

h) Local onde ficam disponíveis os materiais da comunidade (Pastas de arquivos, Mensagens, Fórum de discussões, Links, Blog, Outra).

A disponibilização de material é feita através de fóruns de discussão, mensagens trocadas e links enviados pelos participantes, conforme mostra a Figura 10. Nas interações (através de mensagens trocadas em fórum de discussões) são disponibilizados também materiais, tais como anexos para os participantes, com isso, evidencia-se que as interações compartilham conhecimento entre os participantes.

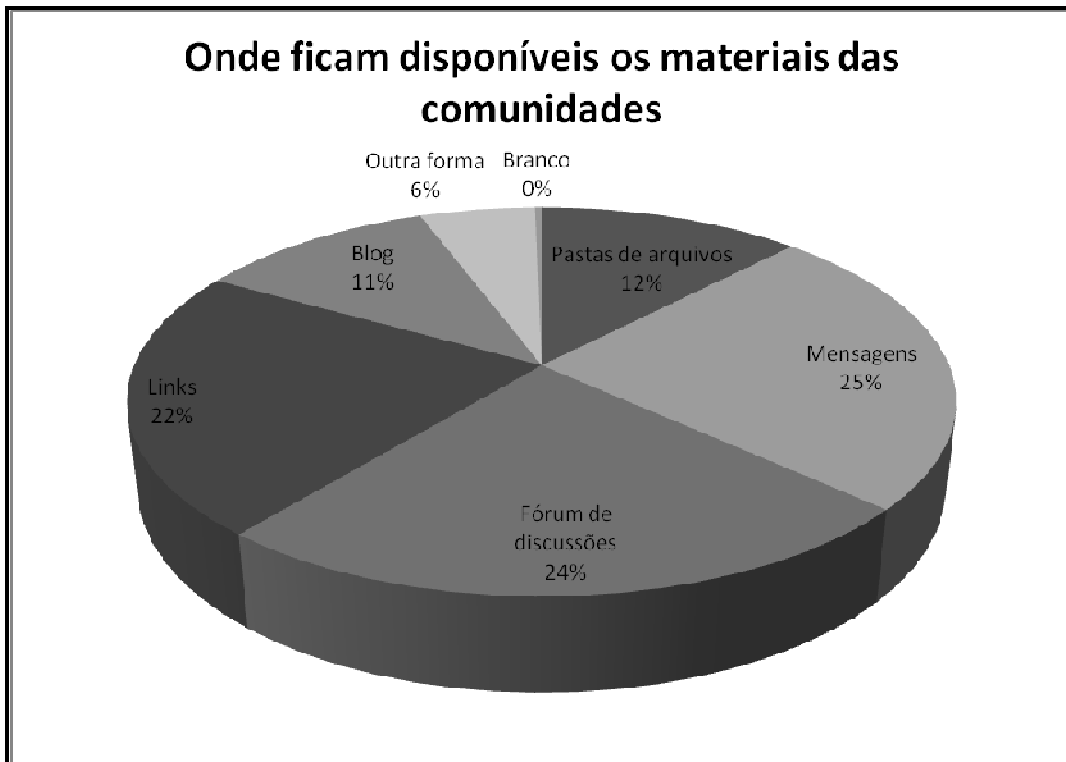


Figura 10 – Gráfico Repositório dos materiais das comunidades

i) Quando tem alguma dúvida ou deseja alguma informação.

A forma mais comum de se tentar sanar uma dúvida é por meio do envio de mensagem para a lista ou fórum, e em seguida através de busca sobre o assunto usando ferramentas disponíveis na comunidade, de acordo com a Figura 11. Essa forma de tirar dúvidas evidencia a colaboração entre os participantes.

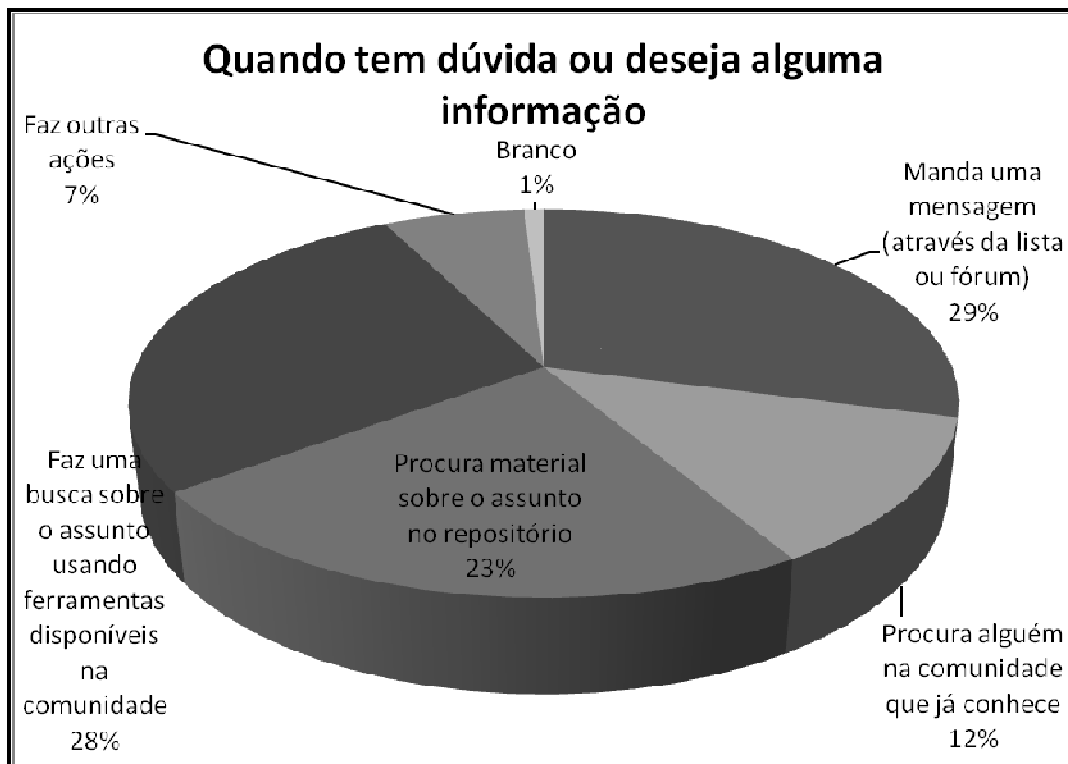


Figura 11 – Gráfico Busca de informação

j) Tipo de mensagem que costuma trocar quando interage com outros participantes da comunidade.

Os tipos de mensagens enviadas durante as interações referem-se à resolução de problemas dos participantes e também ao envio de uma dúvida e suas explicações pelos participantes da comunidade, como mostra a Figura 12. Os tipos de mensagens identificam que os participantes trocam conhecimento durante as interações, ou seja, para a resolução de um problema, os participantes interagem até que o problema seja resolvido, assim como acontece para a resolução de uma dúvida. Isso acontece porque, como a colaboração e a cooperação são características das Comunidades Virtuais de *Software Livre*, os participantes compartilham seu problema pessoal com a comunidade e os demais participantes colaboram com a resolução do problema até que a solução seja encontrada.



Figura 12 – Gráfico Tipos de mensagem durante as interações

1) Mensagens trocadas na sua comunidade respondem as questões debatidas.

Os participantes das comunidades destacam que as suas questões são muitas vezes respondidas por completo (58%), como mostra a Figura 13. Porém, não há uma diferença muito significativa em relação às respostas parciais (40%), o que evidencia que nem sempre os participantes estão satisfeitos com respostas obtidas.

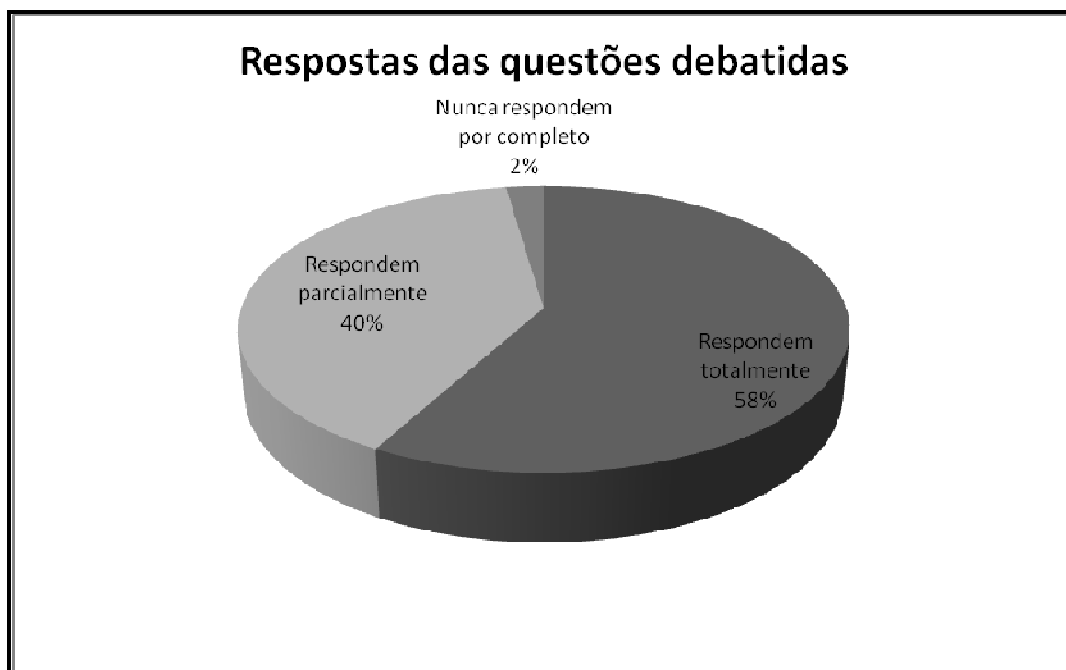


Figura 13 – Gráfico Respostas das questões debatidas

m) Procura por um determinado assunto na comunidade em alguma das modalidades.

A procura por determinado assunto ou para se encontrar o que se deseja na comunidade nem sempre é fácil e rápida como indica a Figura 14. Visto que, conforme a pesquisa, existe muito material compartilhado, seria necessária então uma busca mais eficiente para apoiar os participantes das comunidades a encontrarem o que necessitam.

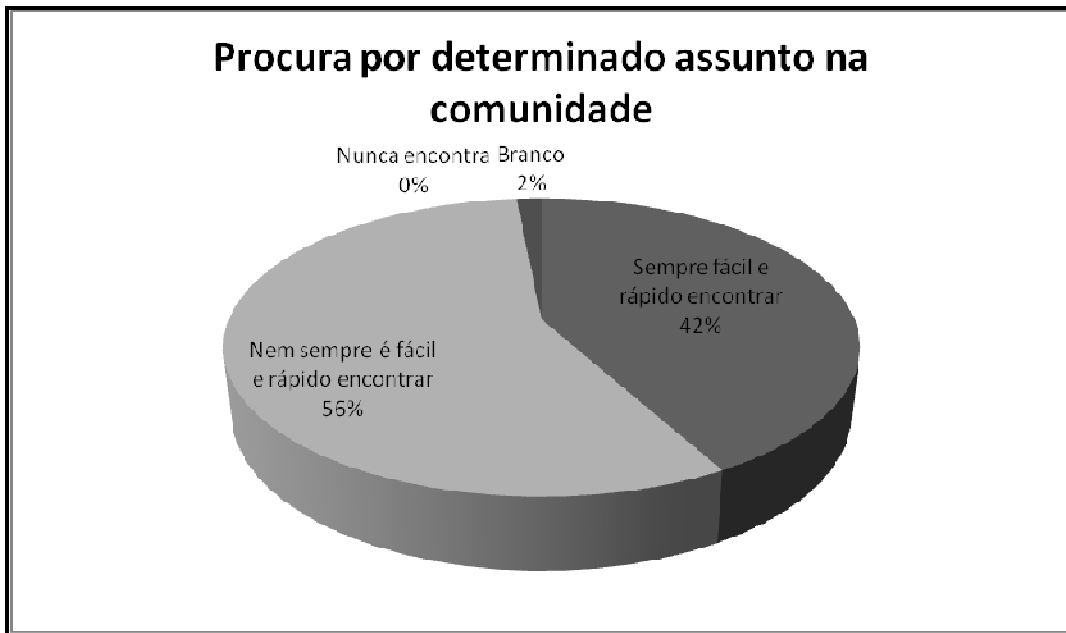


Figura 14 – Gráfico Busca por assunto

n) Possibilidade de indicar um material da comunidade para outro participante acessar.

Em relação à recomendação de material, foi identificado que é possível indicar um material para os outros membros na comunidade, conforme mostra a Figura 15. Isso é um indício de que os participantes são pró-ativos com outros no apoio à resolução de seus problemas e também no compartilhamento do conhecimento gerado e existente na comunidade.



Figura 15 – Gráfico Indicação de material na comunidade

o) Confiança nas informações disponibilizadas por um participante da comunidade.

Sobre a confiança nas informações de um participante, foi identificado que, de maneira geral, todas as informações são úteis para o participante, independente de quem a disponibilizou, ou seja, existe uma relação de confiança em relação à indicação de informações entre os participantes, de acordo com a Figura 16. Os participantes confiam de maneira geral em outros, mesmo não os conhecendo pessoalmente.

Os participantes das Comunidades Virtuais de *Software* Livre colaboram uns com os outros na produção do conhecimento sobre determinada tecnologia, eles são considerados voluntários e estão dispersos geograficamente, com isso a confiabilidade nas informações se torna importante entre ele. Devido a esses fatores, é possível levar em consideração quando um participante recomenda um determinado artigo a outro participante e também como esse participante que indica é visto pela comunidade.

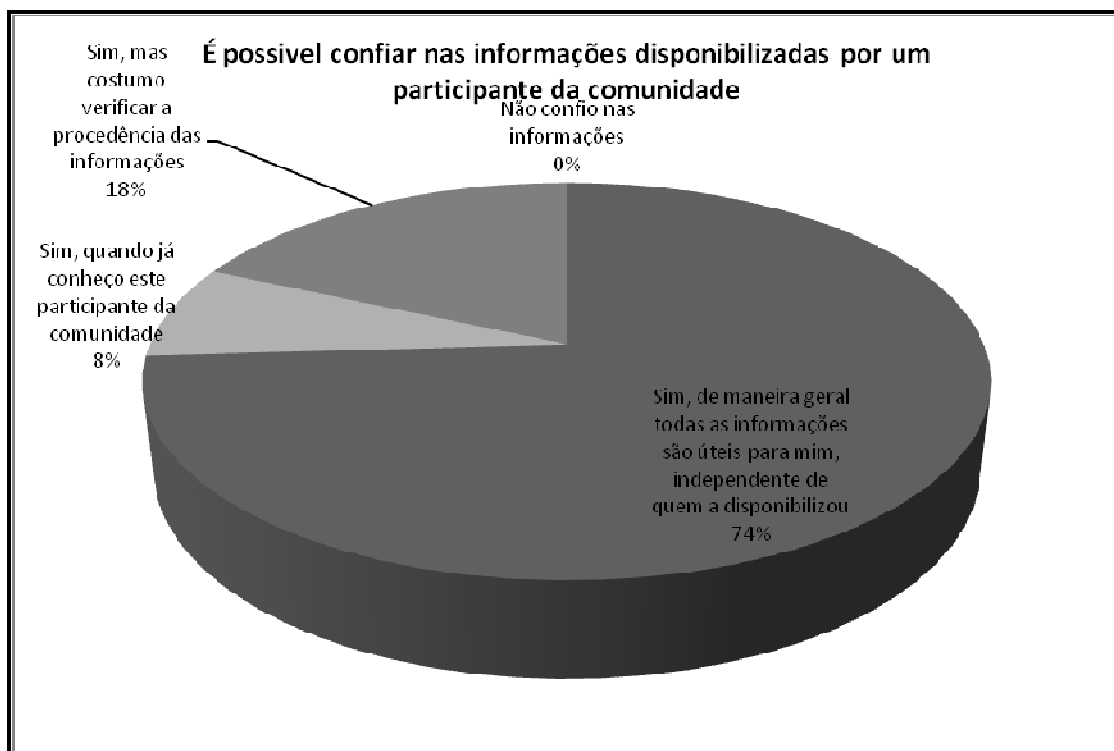


Figura 16 – Gráfico Confiança nas informações

4.3.1 Interpretação dos resultados

Os resultados encontrados evidenciam que os participantes são pró-ativos a colaborar e cooperar uns com os outros, o que demonstra que não existem dificuldades para eles apesar de estarem dispersos geograficamente e serem voluntários. Também é possível confiar nas informações disponibilizadas por um participante da comunidade.

Em relação às interações, estas acontecem através de mais de uma ferramenta e apóiam a criação e compartilhamento do conhecimento tecnológico. As interações possibilitam que os participantes desenvolvam suas ferramentas de forma peculiar e gerem uma inovação tecnológica. O conhecimento é produzido de forma intensa por meio das respostas inseridas em fóruns e em listas de discussão. Isso corrobora com as pesquisas de SOWE *et al.*(2007), VON KROGH, SPAETH e HAEFLIGER (2005).

A busca por determinado assunto é uma questão que precisa de atenção, pois fica evidenciado que nem sempre o assunto que procuram é encontrado facilmente. DE SOUZA (2003) aponta as dificuldades de organizar o conhecimento gerado em projetos de desenvolvimento de *software* em geral.

A recomendação é uma forma de divulgar e compartilhar o conhecimento dentro de fora das comunidades, ou seja, os próprios participantes enviam e recebem material de outros participantes de acordo com seus interesses. Isso fica evidenciado nas respostas dos participantes no questionário.

O Estudo Exploratório validou a análise realizada com as respostas de participantes de Comunidades Virtuais de *Software* Livre e, evidenciou que existe o problema de acesso ao conhecimento pela falta de estruturação nas interações realizadas nessas comunidades.

Assim, os resultados (NASCIMENTO e SANTORO, 2009) mostram que o fórum é o principal mecanismo de interação, mas as interações não são estruturadas. As questões debatidas são respondidas quase na totalidade, e os participantes de maneira geral confiam nas informações que são disponibilizadas. Muito material encontra-se disponível nos repositórios, mas nem sempre é fácil e rápido encontrá-lo.

A estruturação do conhecimento é então uma necessidade para essas comunidades (NASCIMENTO e SANTORO, 2009), o que irá facilitar a Gestão do Conhecimento. Para apoiar tal cenário é necessário um modelo (NASCIMENTO e SANTORO, 2009) que tem como objetivo apresentar características que devem estar presentes em um ambiente de suporte a essas comunidades para auxiliar a estruturação das interações facilitando a busca do conhecimento gerado para melhor compartilhá-lo.

Percebe-se através do resultado deste estudo que as Comunidades Virtuais de *Software* Livre não possuem um modelo geral para organizar as informações que ficam disponíveis nas interações, gerar um encadeamento das opiniões dos participantes, criar um encerramento que leve à conclusão das ideias discutidas, para apoiar à resolução dos problemas dos participantes, o desenvolvimento de ferramentas e facilitar o acesso a estas informações. As discussões em tais comunidades têm características específicas que as distinguem de outras comunidades (NASCIMENTO, SANTORO e VELOSO, 2009) e são resumidas aqui:

- O trabalho colaborativo na resolução de um problema compartilhado por um desenvolvedor.
- As questões são problemas compartilhados por desenvolvedores.
- As mensagens ou contribuições podem se referir a produtos ou versões de produtos em construção pelo grupo que está discutindo (*software*).

- As mensagens reportam a experiência do participante no tema.
- As mensagens são postadas sem formalidade.
- As mensagens apóiam ou contrariam outras mensagens postadas, o que gera uma contribuição de experiências trocadas sobre desenvolvimento.
- A quantidade de mensagens indica o grau de participação na comunidade.
- Os participantes se comunicam ativamente para desenvolver uma solução.
- O resultado da discussão poderá ser de fato implementado por um ou mais participantes e as suas conseqüências serão públicas.
- Não existe na maioria das vezes uma documentação formal sobre o que foi discutido, associada ao produto em questão.
- Existem repositórios para controle de versão dos produtos gerados, mas não das soluções de problemas específicos discutidos nos fóruns.
- A recomendação de artefatos ou mensagens para ajudar os outros em seus problemas ou para indicar uma nova tecnologia.
- A reputação do indivíduo que introduz a contribuição é altamente relevante.
- A reutilização do código fonte desenvolvido é constante gerando evolução tecnológica e inovações.
- Os artefatos gerados são reutilizados de acordo com o problema do participante. Esses artefatos podem ser um documento explicativo sobre uma tecnologia ou o próprio código fonte desenvolvido.
- As interações são a principal fonte geradora de conhecimento nas comunidades.

Com base nas características destacadas foi possível identificar as necessidades das Comunidades Virtuais de *Software* Livre em relação à Gestão do Conhecimento. No próximo capítulo, são discutidos os conceitos relacionados com as formas de estruturação de conhecimento, que levaram à especificação da solução proposta nesta dissertação, ou seja, os modelos de argumentação.

Capítulo 5 – Modelos de argumentação

Este capítulo apresenta uma revisão bibliográfica sobre modelos de argumentação, mostrando as características, os objetivos e a estruturação de cada um. Tais modelos apoiam as interações com seus elementos e estruturaram as ideias inseridas pelos participantes durante as discussões. Os modelos analisados foram: IBIS, GD-IBIS, Esquema de Toulmin e ColaOnto.

5.1. Conceitos básicos

Modelo de argumentação é uma forma de estruturar uma interação embasada no modelo conceitual, ou seja, nos conceitos apresentados para apoiar estruturação do conhecimento, baseada no ciclo da Gestão do Conhecimento, em Comunidades Virtuais de *Software Livre*.

“Argumento é parte do raciocínio que leva a uma conclusão”, MICHAELIS (2009). Para argumentar sobre um determinado assunto ou tema é necessário que se tenha conhecimento sobre o mesmo e justificativas plausíveis para gerar os argumentos, sejam eles contra ou a favor da questão levantada na discussão RELVAS e ANTUNES (2006). Um modelo de argumentação visa estruturar uma discussão de forma a identificar como cada participante está contribuindo, encadeando as informações de forma coerente e facilitando o acesso ao conhecimento gerado, particularmente relacionando-o com os processos de decisão (EDEN, 1989).

Os estudos relacionados as tais modelos surgiram pelas pesquisas de RITTEL e KUNZ (1970), com o propósito de categorizar as mensagens e então foi desenvolvido o modelo IBIS. A base dos modelos de argumentação é o mapeamento de conceitos, que surge com o objetivo de melhorar o desempenho na aprendizagem (NOVAK, 1984). Os modelos de argumentação têm a finalidade de ajudar a construir, através de representações mentais, abstratas e gerais, o conceito de um objeto, tema, problema ou solução. O conceito no caso das Comunidades Virtuais de *Software Livre* pode ser considerado como o assunto discutido na interação. Diversos modelos são propostos na literatura, tais como IBIS (RITTEL e KUNZ, 1970), GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002), Toulmin (TOULMIN, 1958), e ColaOnto (MALEEWONG *et al.*, 2008).

5.2. Modelo IBIS

O modelo IBIS (*Issue Based Information System*) foi desenvolvido para tratar a seguinte situação: diariamente se lida com assuntos ou problemas que não apresentam um acordo entre os intervenientes. Observa-se que são construídos argumentos que apóiam ou contrariam o assunto ou problema (RITTEL e KUNZ, 1970). Os autores RITTEL e KUNZ (1970) verificam que, durante o processo de discussão do problema,

perguntas e respostas são repetidas diversas vezes. Para evitar essa repetição, cada participante constrói ou reforça seus argumentos, que podem ser compartilhados ou confrontados com os dos outros participantes na discussão. Do ponto de vista cognitivo, observa-se um contexto em que, para solucionar um problema, vários participantes possuindo diferentes representações de um mesmo problema, tentam chegar a um acordo (RITTEL e WEBBER, 1973). Os problemas têm as seguintes propriedades, segundo RITTEL e KUNZ (1970) e, TOUCHSTONE (2000):

- Assumem a forma de perguntas.
- A sua origem é fruto de afirmações controversas.
- São particulares a uma situação, sendo que as posições assumidas utilizam informação particular ou situações semelhantes.
- São levados, discutidos, acordados ou substituídos.

O modelo IBIS (RITTEL e KUNZ, 1970), é baseado em três categorias: Questão (*issue*), Posição (*position*) e Argumentação (*argument*). A Questão é a pergunta ou assunto a ser discutido, o problema que se quer resolver. A Posição são as respostas da questão em debate, ou seja, o ponto de vista de cada um. Argumentação é a explicação a uma resposta, ressaltando um posicionamento contra ou a favor do indivíduo. O modelo IBIS gera uma estruturação para as discussões, sendo utilizado como um sistema de apoio à decisão. Esse modelo é o mais utilizado para apoiar os outros modelos. A Figura 17 retrata as relações entre as categorias do modelo IBIS.

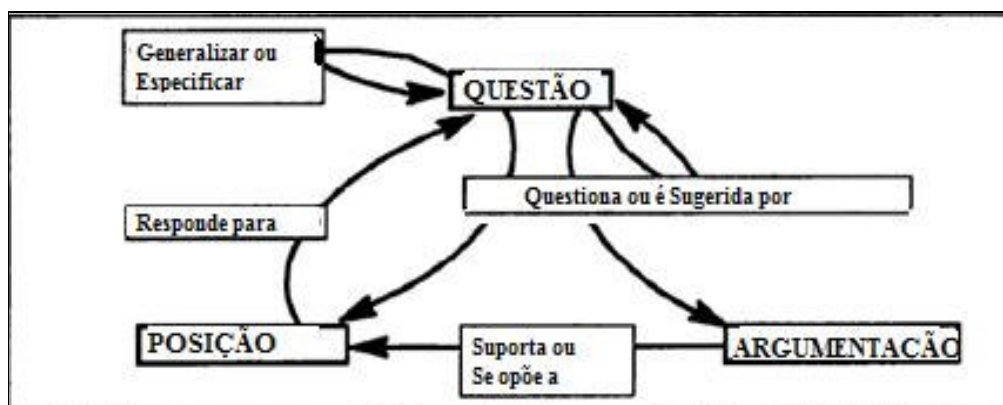


Figura 17 – Modelo IBIS (RITTEL e Kunz, 1970)

O modelo de argumentação IBIS (RITTEL e KUNZ, 1970), tem sido muito utilizado no apoio à estruturação da memória organizacional, com o objetivo de suportar, documentar e coordenar as informações por meio dos elementos de uma discussão, estabelecendo um padrão estruturado e sistemático de comunicação, de acordo com a ferramenta gIBIS, (CONKLIN e BEGEMAN, 1988).

5.3. Modelo GD-IBIS

O modelo de argumentação GD-IBIS (Grupo de Discussão – IBIS), segundo ANDRADE, SAMPAIO e ROCHA (2002), organiza uma discussão tornando-a mais transparente e intuitiva a interação em grupos de discussão. O modelo é composto pelos elementos: Tema, Questão, Posição, Argumento a Favor, Argumento Contra, Anuência e Não Anuência, de acordo com a Figura 18. A partir desse modelo foi gerada uma ferramenta, também chamada GD-IBIS que é um mecanismo de interação assíncrona, de acordo com ANDRADE, SAMPAIO e ROCHA (2002).

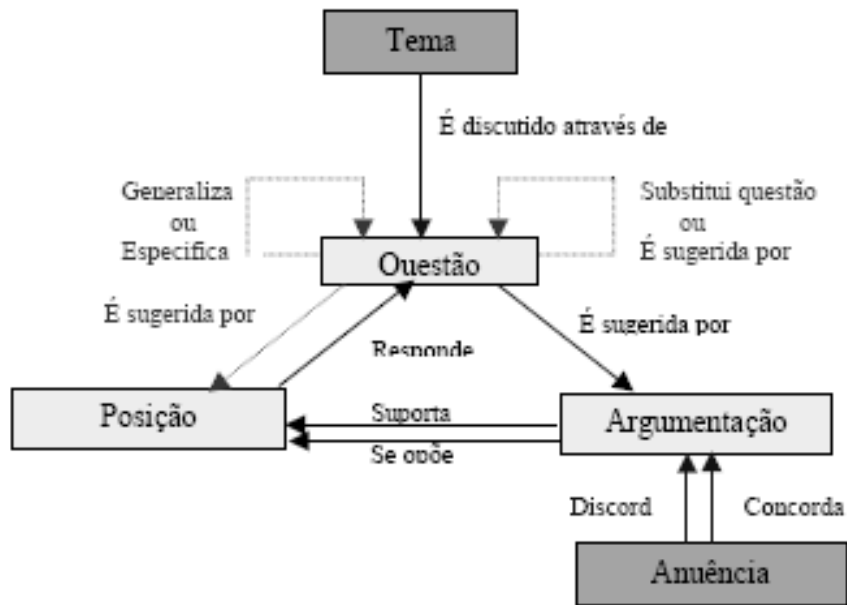


Figura 18 – Modelo GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002)

De acordo com ANDRADE, SAMPAIO e ROCHA (2002), o modelo GD-IBIS é baseado no modelo IBIS (RITTEL e KUNZ, 1970) e as características de seus elementos são:

- Tema representa o assunto a ser discutido;
- Questão é um aspecto a ser discutido e não somente uma pergunta;

- Posição é a opinião sobre uma questão;
- Argumento a favor é o argumento apoiando uma determinada posição;
- Argumento contra é o argumento de oposição a uma determinada posição;
- Anuência é a conformidade com o argumento postado;
- Não anuência é a não conformidade com o argumento postado.

O ambiente GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002) estrutura de forma hierárquica utilizando a categorização das mensagens, permite anexar materiais às mensagens postadas, contém ícones que identificam os elementos da discussão introduz mecanismos para facilitar a estruturação da discussão pelo controle da escolha de categorias, a discussão é representada no formato de árvore e existe a diferenciação entre mensagens lidas e não lidas.

O ambiente GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002) foi experimentado no contexto de EAD (Educação a Distância) num grupo de discentes e foi verificado que este modelo estruturou a discussão, servindo como uma ferramenta de comunicação assíncrona.

5.4. Modelo de Esquemas de Toulmin

O modelo de esquemas de Toulmin (TOULMIN, 1958) possui três categorias fundamentais: Motivo (*ground*), Afirmação (*claim*) e Justificativa (*warrant*), em que os dados sobre determinado assunto, desde que apoiados em uma justificativa existente, levam a uma afirmação, ou seja, um assunto existe pela sua afirmação e justificativa. Para TOULMIN (1958), argumentar é um processo composto por afirmações que levam a conclusões, por meio da identificação dos elementos de um argumento persuasivo, sendo útil para encontrar categorias para as quais o argumento possa ser analisado. Nesse processo, existem formas de oferecer suporte e justificativas às conclusões baseando-se em motivos, dados, fatos e evidências reunidas. O modelo apóia a decisão colaborativa de grupos de discussão.

O esquema de argumentação de Toulmin (TOULMIN, 2009), de acordo com a Figura 19, é composto por: Motivo (*ground or data*) significa a base de persuasão na qual a afirmação é baseada, são os dados ou elementos de prova que apóiam à afirmação, as evidências, as razões. Afirmação (*claim*) é o problema a ser resolvido, ou seja, uma posição a ser adotada sobre um assunto; Justificativa (*warrant*) é princípio de

ligação entre o motivo (*dado*) e a afirmação, mostrando a relevância; Suporte (*backing*) é o apoio à justificativa, as causas que contribuem com justificativa; Qualificador (*qualifier*) é o grau de certeza do dado para justificar e limitar como a afirmação é aplicada, a garantia da afirmação; Contra-prova (*rebuttal*) indica o reforço contra um contra-argumento, isto é, as exceções para a impugnação de contra-exemplos.

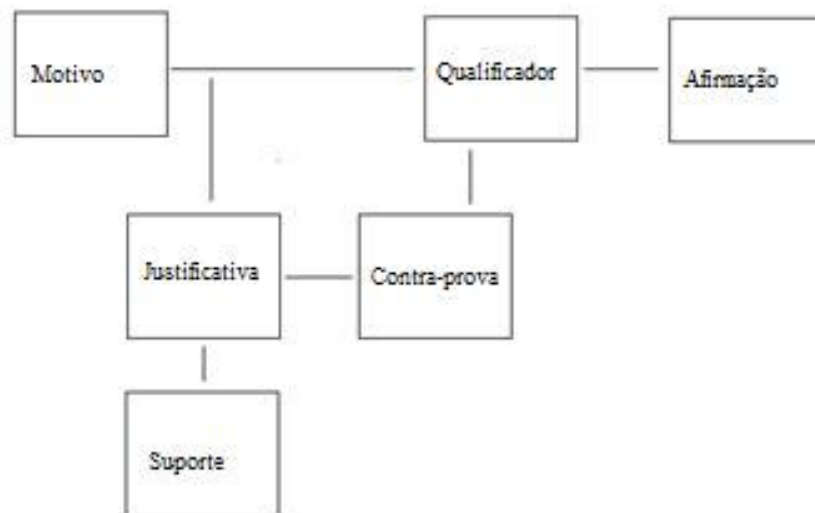


Figura 19 - Modelo Toulmin (TOULMIN, 2009)

5.5. Modelo ColaOnto

O modelo ColaOnto (*Collective Argumentation Ontology*), segundo MALEEWONG *et al.* (2008) é uma ontologia de argumentação coletiva. Foi desenvolvido com base na fusão dos modelos IBIS e Toulmin, com foco na geração de um conhecimento coletivo (inteligência coletiva). Os autores pretenderam estimular a colaboração, a atuação dos participantes em um debate para encontrarem um consenso e chegarem a uma conclusão. O modelo é composto, de acordo com a Figura 20, pelas seguintes categorias: Questão (*issue*), Posição (*position*), Argumentação (*argument*), Suporte (*backing*), Dado (*data*), Contra-prova (*rebuttal*), Indivíduo (*individual*), Qualificador (*qualifier*).

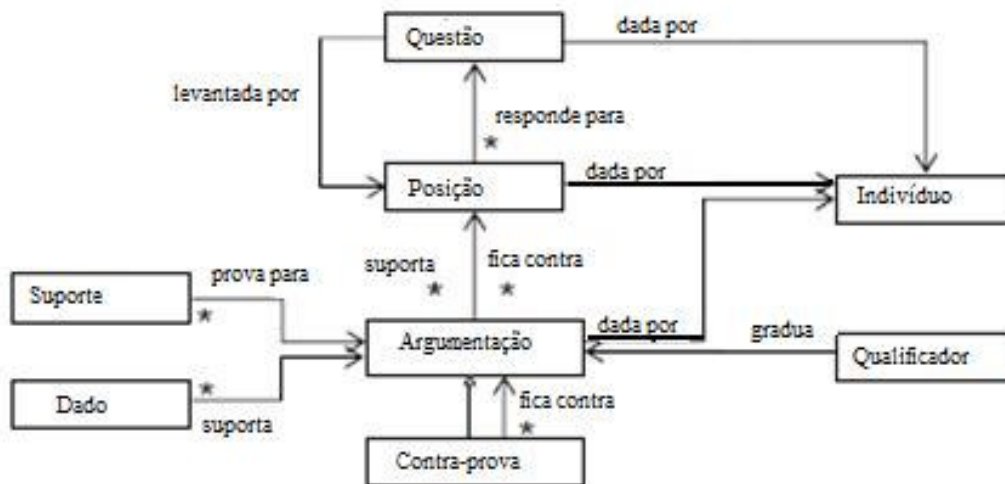


Figura 20 – Modelo ColaOnto (MALEEWONG *et al.*, 2008)

No ColaOnto (MALEEWONG *et al.*, 2008), uma Questão é uma pergunta ou problema levantado por um Indivíduo em relação ao domínio de interesse, de acordo com a Figura 20. Uma Posição é uma solução postada por um indivíduo para a questão e pode ser apoiada ou contrariada pela Argumentação. Para cada argumento, os Dados representam um fato que suporta o argumento, enquanto Suporte é uma prova ou declaração para um argumento. Além disso, um Qualificador expressa o grau individual de certeza concentrado no argumento postado. Um argumento pode ser feito contra por Contra-provas. E uma posição pode gerar uma nova questão, se for necessário que seja resolvida de forma mais detalhada. Neste modelo o grau (*) de importância de cada categoria é calculado de acordo com o método *Delphi* (LINSTONE e TUROFF, 1975). Tal método indica que os indivíduos revejam seus pareceres levando em conta o que ponto de vista os demais participantes até que se chegue a resposta final.

As atividades que compõem a inteligência coletiva, de acordo com MALEEWONG *et al.*, 2008, geradas pelo método são: posicionamento, argumentação, consenso e seleção de ideias. Tais atividades levam à criação de uma questão a ser debatida para a geração de um novo conhecimento pela seleção da melhor solução e assim acontece a construção do conhecimento coletivo.

5.6. Quadro Comparativo dos Modelos de Argumentação

Tabela 3 – Comparação dos Modelos de Argumentação

CATEGORIAS	IBIS	TOULMIN	GD-IBIS	COLAONTO
Questão	X	X (Motivo)	X	X
Posição	X	X (Justificativa)	X	X
Argumentação	X	X (Afirmação)	X	X
Qualificador		X		X
Suporte		X		X
Contra-Prova		X		X
Tema			X	
Anuência			X	
Dado				X
Indivíduo				X

Os quatro modelos têm em comum a necessidade da resolução de um problema de forma estruturada e objetiva, onde a discussão é composta por posições e argumentações individuais que levam à geração de uma conclusão implicitamente, ou seja, a resolução do problema com a participação de diversos pontos de vista e as experiências individuais de cada um. Os modelos possuem uma interseção sobre os conceitos básicos: questão, posição e argumentação. O acréscimo de detalhes nos modelos mais recentes permite uma riqueza maior de informações para entendimento e mesmo embasamento das argumentações. No entanto, deve-se levar em conta que a simplicidade destes modelos deve ser preservada e, portanto muitos detalhes podem acabar por comprometer o seu uso.

Observa-se que nenhum deles incorpora a explicitação do encerramento da discussão ou resultado final, ou mesmo se não se chegou a um resultado final de consenso. A conclusão deve ser apreendida pelo indivíduo. Isso pode levar a distintas interpretações da solução do problema, principalmente quando estes forem muito complexos, ou quando envolverem um grupo grande de pessoas.

O princípio de todos os modelos está na questão a ser discutida e todos partem do modelo IBIS (RITTEL e KUNZ, 1970), sendo esse uma estrutura simples e passível para criação de novos elementos de acordo com o contexto a ser utilizado.

O modelo GD-IBIS (ANDRADE, SAMPAIO e ROCHA, 2002) segue com um propósito educacional para estruturar uma discussão, enquanto o modelo de esquemas de Toulim (TOULIM, 2009) e o modelo ColaOnto (MALEEWONG *et al.*, 2008) se propõem a ser mais abrangentes, Toulim (TOULIM, 2009) tenta criar um esquema de indução sobre o que está sendo discutido enquanto o modelo ColaOnto ao se apropriar de princípios do modelo IBS e do modelo de esquemas de Toulim faz com que a discussão seja conduzida pelos elementos da discussão focando no grau de certeza para um consenso.

Todos esses modelos estão dentro de um contexto de discussão, mas não criam um resultado final de imediato quando a conclusão é finalizada, baseando-se na necessidade de estruturação das interações e de facilitar o acesso a discussão. Desta forma, no capítulo seguinte é apresentado um modelo de argumentação proposto para as Comunidades Virtuais de *Software* Livre.

Capítulo 6 – Um modelo de estruturação de conhecimento em interações nas Comunidades Virtuais de *Software* Livre

Este capítulo descreve o modelo proposto para estruturar o conhecimento nas interações em Comunidades Virtuais de *Software* Livre, baseado no ciclo da Gestão do Conhecimento. O modelo de argumentação ArFoLi (Argumentação para Fórum de *Software* Livre) é baseado nos requisitos levantados por meio do Estudo Exploratório e na revisão da literatura.

6.1. Modelo Proposto

O objetivo da proposta é estruturar as interações entre os participantes e facilitar o acesso ao conhecimento gerado pelas interações nas Comunidades Virtuais de *Software Livre*, estimulando o ciclo da Gestão do Conhecimento. Desta forma, o modelo proposto é composto por entidades que representam os elementos presentes no ambiente de apoio a Comunidades Virtuais de *Software Livre*, onde são incorporados os conceitos relacionados à estruturação do conhecimento. O modelo é descrito por meio de:

- um diagrama de classes, onde são apresentadas as principais classes de entidades envolvidas nos ambientes de apoio a Comunidades Virtuais de *Software Livre*, bem como as relações entre elas;
- um diagrama que representa o detalhamento do modelo de argumentação para as interações, bem como a descrição textual da sua forma de uso.

6.2. Diagrama de classes

Um diagrama de classes é um modelo conceitual, que descreve os principais conceitos existentes nos ambientes de apoio a Comunidades Virtuais de *Software Livre*, de acordo com a proposta desta dissertação. Este modelo é centrado no conceito de interação (Figura 21). Esses elementos se relacionam com o foco no desenvolvimento existente nas comunidades, ou seja, o código fonte produzido.

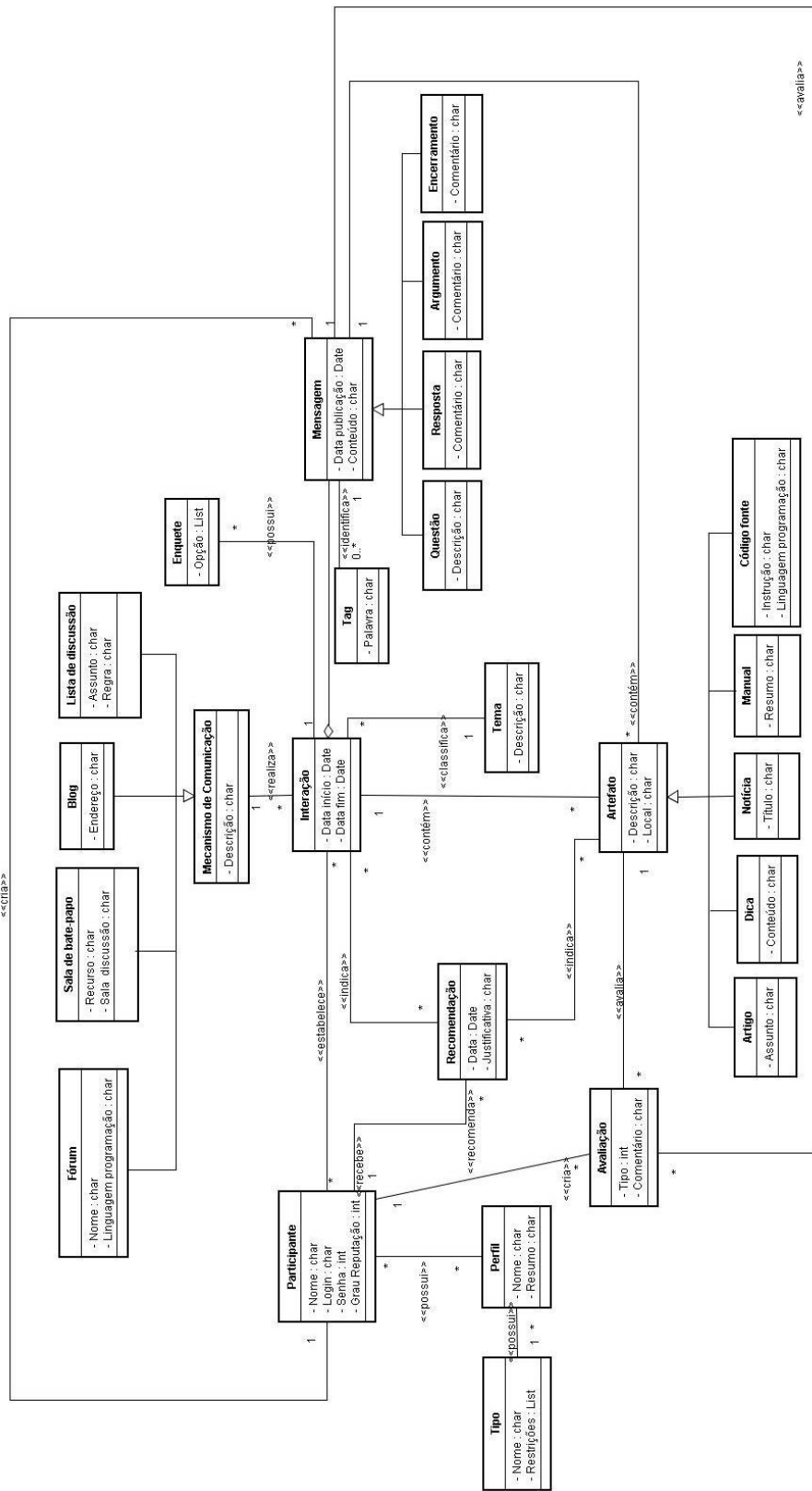


Figura 21 - Modelo Conceitual

No contexto das Comunidades Virtuais de Software Livre, o modelo de classes é composto pelas seguintes classes:

- Participante
- Perfil
- Tipo
- Recomendação
- Avaliação
- Artefato
- Interação
- Mecanismo de Comunicação
- Enquete
- Mensagem
- *Tag*
- Tema

Interação é o ponto central do modelo; a partir dela todas as relações são realizadas. É possível pela Interação que a comunicação, a cooperação, a troca de conhecimento e o desenvolvimento de ferramentas sejam realizados pelos Participantes. A Interação é delimitada no tempo por uma data de início e uma data de fim. Além disso, contém Artefato e possui Enquete.

O Tema classifica Interação e possui uma descrição para caracterizá-lo. Este Tema é a classificação da Interação, um identificador do assunto que está sendo tratado na Interação.

Já a Enquete possui opções de acordo com a realização de uma votação na comunidade e permite a coleta de opiniões dos Participantes.

Participante é um membro da comunidade, caracterizado pelo nome, login, senha e grau de reputação. Os participantes por sua vez podem possuir vários perfis que identificam a atuação do participante na comunidade.

Perfil é caracterizado pelo seu nome e resumo. Tipo possui vários perfis e é identificado pelo nome e suas restrições de atuação na comunidade. Cada Tipo está ligado a cada forma de atuação na comunidade. Os perfis podem ser do tipo: administrador, coordenador, moderador, desenvolvedor e participante informal. O administrador é o responsável pela comunidade, normalmente é quem criou a comunidade. O coordenador é quem organiza os projetos e apoia o administrador. O

moderador é um mediador de uma interação. O desenvolvedor é um participante que trabalha voluntariamente na comunidade apoiando a resolução de um problema seu ou, auxiliando no desenvolvimento de uma ferramenta (produto a ser gerado pela comunidade). O participante informal é aquele que participa nas comunidades sem possuir um papel definido, normalmente está na comunidade e quer tirar uma dúvida ou saber o que está acontecendo.

Cada Participante pode criar uma ou mais Mensagens, as quais são identificadas pela data de criação e o seu conteúdo. Estas mensagens são a forma de capturar o conhecimento dos participantes por meio da sua experiência descrita explicitamente e são de quatro tipos.

As mensagens são classificadas então como: Questão, Resposta, Argumento e Encerramento. Questão é um problema descrito e discutido na comunidade por meio da sua descrição. Resposta é o ponto de vista de cada participante, corresponde ao conhecimento tácito de cada participante sobre o assunto discutido, a experiência que o participante possui e é feito no comentário. Argumento é a opinião descrita de outro participante referente a uma resposta, é realizado pelo comentário. Encerramento é o fechamento descrito de uma interação, um comentário final, uma conclusão.

Cada Mensagem contém vários Artefatos e é identificada por *Tag*. *Tag* é uma espécie de palavra-chave que identifica a Mensagem, ou seja, uma “etiqueta” para ser usada na Questão, na Resposta, no Argumento e no Encerramento

Participante cria Avaliação para avaliar a Mensagem e Artefato. A Avaliação é a opinião sobre a Mensagem ou o Artefato explicitada para a comunidade, é uma apreciação particular do Participante, serve como um agregador de valor ao que está exposto. Avaliações são descritas pelo seu tipo e o comentário em si, que é um texto livre escrito pelo Participante.

Participante também pode indicar Recomendação de Artefatos e mesmo de Interação para outro Participante, ou seja, uma forma de compartilhar e divulgar aos demais participantes Artefatos e Interações. Essa Recomendação poderia ser indicada automaticamente por um Sistema de Recomendação, desde que esta funcionalidade estivesse presente no ambiente de suporte.

Já Artefato é a forma de representar o conhecimento gerado e trocado na comunidade. O Artefato possui descrição, local de armazenamento e, é classificado como: Artigo, Dica, Notícia, Manual e Código fonte. Artigo é caracterizado pelo seu

assunto. Dica contém um conteúdo. Notícia é um artefato caracterizado pelo seu título. Manual é um artefato que contém seu resumo. Já o Código fonte é identificado pela sua instrução e linguagem de programação.

E, por fim, Mecanismo de comunicação é o meio através do qual as interações são realizadas, ou seja, apóiam os participantes a estabelecerem Interações. Mecanismos de comunicação possuem descrição e pode ser do tipo: Fórum, Sala de bate-papo, Blog e Lista de discussão. Fórum possui um nome e uma linguagem de programação. Sala de bate-papo tem seu recurso e a sala de discussão. Enquanto Blog tem endereço (*link*) para sua identificação. Lista de discussão é caracterizada por um assunto e sua regra de utilização.

Todos os elementos do modelo atuam em conjunto para organizar e identificar as interações, para que o conhecimento nela contido fique estruturado.

O modelo conceitual serve para apoiar a Gestão do Conhecimento, pois possibilita:

- Capturar o conhecimento pela Interação realizada por meio dos Mecanismos de comunicação;
- Representar o conhecimento pelas Mensagens e Artefatos;
- Armazenar o conhecimento separado pelo Tema da Interação;
- Compartilhar pela Recomendação e pela Avaliação;
- Reutilizar o conhecimento das Interações.

6.3. ArFoLi (Argumentação para Fórum de *Software* Livre): Modelo de Argumentação

O modelo de argumentação ArFoLi (Argumentação para Fórum de *Software* Livre) é proposto, de forma a atender as seguintes características destacadas no Capítulo 4:

- O trabalho colaborativo na resolução de um problema compartilhado por um desenvolvedor;
- As questões são problemas compartilhados por desenvolvedores;
- As mensagens ou contribuições podem se referir a produtos ou versões de produtos em construção pelo grupo que está discutindo (*software*);
- As mensagens reportam a experiência do participante no tema;

- As mensagens são postadas sem formalidade;
- As mensagens apóiam ou contrariam outras mensagens postadas, o que gera uma contribuição de experiências trocas sobre desenvolvimento;
- A quantidade de mensagens indica o grau de participação na comunidade;
- Os participantes se comunicam ativamente para desenvolver uma solução;
- O resultado da discussão poderá ser de fato implementado por um ou mais participantes e as suas conseqüências serão públicas;
- Não existe na maioria das vezes uma documentação formal sobre o que foi discutido, associada ao produto em questão;
- Existem repositórios para controle de versão dos produtos gerados, mas não das soluções de problemas específicos discutidos nos fóruns;
- A recomendação de artefatos ou mensagens para ajudar os outros em seus problemas ou para indicar uma nova tecnologia;
- A reputação do indivíduo que introduz a contribuição é altamente relevante;
- A reutilização do código fonte desenvolvido é constante gerando evolução tecnológica e inovações;
- Os artefatos gerados são reutilizados de acordo com o problema do participante;
- As interações são a principal fonte geradora de conhecimento nas comunidades.

Em um primeiro nível, as mensagens trocadas durante uma interação são estruturadas em tipos (como mostra o Modelo de Classes), permitindo que se estabeleça um entendimento melhor de seu significado. Este modelo se baseia no IBIS (RITTEL e KUNZ, 1970) e Toulmin (TOULMIN, 1958), tendo como objetivo estruturar a discussão do problema, organizar as respostas inseridas e por meio da estruturação da interação para facilitar a busca do conhecimento pelos participantes, ou seja, pela identificação de cada etapa da discussão e a atuação do participante.

Em um segundo nível de estruturação, é proposto o uso do conceito de *tags* criado no contexto da Web 2.0. A Web 2.0 tem como características: a interatividade e a colaboração. O conceito da Web 2.0 foi criado por O'REILLY (2005) e se disseminou

rapidamente pelo mundo. Na realidade, a Web 2.0 é uma forma de tratar os aplicativos focando na interatividade e na colaboração e não uma mudança tecnológica. Na Web 2.0, (O'REILLY, 2005), os usuários alteram seu papel de simples receptores para compartilhadores de conhecimento. Nessa abordagem da Web 2.0, foram implementadas no fórum as funcionalidades de *tags*, e nuvens de *tags* para facilitar o acesso a conhecimentos-chaves na interação. O conceito de etiqueta ou *tag*, (GOLDER e BERNARDO, 2005), possibilita a inserção de palavras-chaves da discussão pelos participantes, permitindo uma liberdade de expressão nas associações e permitindo enriquecer a discussão. A *tag* tem a função de facilitar o acesso ao conteúdo da mensagem inserida na discussão do fórum.

Desta forma, o modelo propõe a estruturação em termos de relacionamentos entre as mensagens trocadas e sobre o conteúdo das mensagens, sempre do ponto de vista dos participantes da interação.

O modelo é composto pelas categorias: Questão, Resposta, Qualificação, Argumento, *Tag* e Encerramento, conforme a Figura 22.

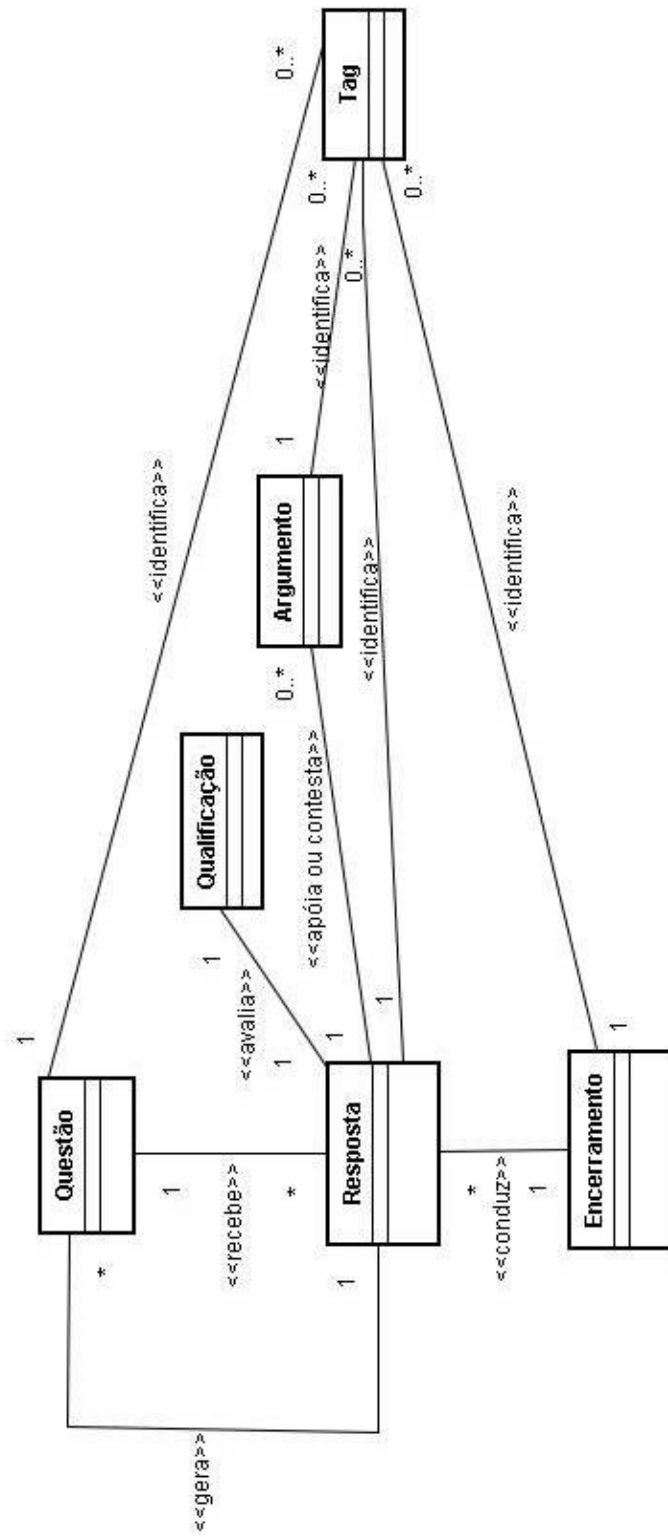


Figura 22 - Modelo ArFoLi

A **Questão** é a pergunta ou dúvida inserida por um participante, que gera a abertura do fórum. Essa questão é normalmente um problema pessoal que o desenvolvedor está enfrentando em alguma etapa de trabalho no produto de *software*, que não consegue resolver sozinho ou esgotou as possibilidades, e acaba sendo compartilhado. Exemplos típicos destes tipos de problemas são: o problema de um componente, um *bug* no teste, como realizar uma atualização de uma versão etc.

A **Resposta** é o ponto de vista de um participante com sua experiência, ou seja, a colaboração e a cooperação de cada participante que indica uma solução do problema existente. As respostas são a fonte de conhecimento implícito particular de cada participante. Geralmente, as respostas são fornecidas em linguagem informal, porém bastante técnica, ou seja, por meio de partes de códigos ou linhas de comando para um *hardware* ficar ativo.

A **Qualificação** é um conceito dado pelo participante para identificar o grau de confiança e propriedade técnica de uma resposta. A Qualificação ajuda os participantes externos a identificarem respostas mais completas pelo seu conceito. Por meio da qualificação é possível identificar o ponto de vista mais adequado pela visão colaborativa da comunidade. Os graus de qualificação podem ser indicados pela pontuação de 1 (mínimo) até 10 (máxima). A pontuação também pode ser positiva (+) ou negativa (-), para indicar se a resposta está adequada ao propósito da discussão, ou seja, se a resposta está coerente com o que está sendo discutido.

O **Argumento** é um ponto de vista de um participante que apóia ou contesta uma determinada resposta inserida no fórum. É a forma por intermédio da qual os participantes podem contribuir com suas experiências práticas e melhor embasarem a solução do problema.

A **Tag** é a identificação das mensagens, isto é, uma forma de criar “etiquetas” para as respostas de uma discussão. As *tags* são complementares as mensagens e promovem a liberdade, pois não são de uso obrigatório, servem como um nível a mais de estruturação. Esse conceito da *tag* ligada à mensagem é também um novo conceito que agiliza o acesso a uma discussão pela sua “etiqueta” identificadora.

O **Encerramento** é a finalização do fórum realizada pelo participante responsável pelo fórum, que pode identificar uma ou mais respostas adequadas para questão levantada. O encerramento possibilita chegar a uma conclusão e não deixar a

discussão sem uma finalização, introduzindo um novo conceito aos modelos de argumentação.

Assim, o modelo ArFoLi possibilita a estruturação da interação para posterior reutilização do conhecimento registrado nas respostas e argumentos, identifica uma discussão pela *tag*, qualifica as respostas mais adequadas, organiza os artefatos e conclui a discussão, o que facilita o acesso as mensagens e recuperação dos artefatos gerados.

6.4. Quadro comparativo dos Modelos de Argumentação e o Modelo ArFoLi

Tabela 4 – Comparação entre os Modelos de Argumentação e o Modelo ArFoLi

CATEGORIAS	IBIS	TOULMIN	GD-IBIS	COLAONTO	ARFOLI
Questão	X	X (Motivo)	X	X	X
Posição	X	X (Justificativa)	X	X	X (Resposta)
Argumentação	X	X (Afirmção)	X	X	X (Argumento)
Qualificador		X		X	X (Qualificação)
Suporte		X		X	
Contra-Prova		X		X	
Tema			X		
Anuência			X		
Dado				X	
Indivíduo				X	
<i>Tag</i>					X
Encerramento					X

O quadro comparativo aponta as diferenças entre os modelos de argumentação encontrados na literatura e o modelo proposto ArFoLi. É possível identificar que o modelo ArFoLi se baseia nas três categorias fundamentais: questão, posição e argumentação e, inseri duas categorias novas: *tag* e encerramento. Com essa proposta o modelo ArFoLi estrutura a interação visando o facilitar o ciclo da Gestão do Conhecimento nas Comunidades Virtuais de *Software Livre*.

A seção seguinte mostra a aplicação do modelo ArFoLi em dois cenários de interação em uma Comunidade Virtual de *Software* Livre.

6.5. Análise de dois Cenários de Aplicação do Modelo

A análise de dois cenários apresenta a aplicação do modelo de argumentação desenvolvido em uma situação real de uma comunidade, ou seja, em uma discussão real. O objetivo é demonstrar a estruturação da discussão em um fórum e suas vantagens, assim são apresentadas duas discussões em cenário real, extraídas do fórum da comunidade VivaOLinux (VIVAOLINUX, 2009). E em seguida essas mesmas discussões são reorganizadas nos moldes do modelo de argumentação proposto.

Cenário 1 – O participante precisa de auxílio na instalação de uma ferramenta.

Participante 1: amigos, eu tinha tentado instalar o hydra por apt-get modificando uns repositórios, e agora sempre q uso o apt-get para instalar algum programa ele apresenta a mensagem: dpkg: aviso sério: arquivo com lista de arquivos do pacote `hydra' está falando, assumindo que pacote não tem arquivos instalados atualmente.140801 arquivos e diretórios atualmente instalados.) Preparando para substituir hydra 5.4-bt1 (usando ../archives/hydra_5.4-bt1_all.deb) ... Descompactando substituto hydra ... dpkg-deb: arquivo `/var/cache/apt/archives/hydra_5.4-bt1_all.deb' contém membros de dados não entendidos data.tar.lzma , desistindo Erro: Sub-process /usr/bin/dpkg received a segmentation fault. em seguida para ao tentar executar o apt-get. Erro: dpkg was interrupted, you must manually run `dpkg --configure -a' to correct the problem. Ae tenho q usar o comando dpkg --configure -a pra poder executar o apt-get, alguem sabe cm resolver? obrigado\

Participante 2: digitando dpkg --configure -a o que dá??

Participante 3: não apresenta nenhuma mensagem apenas vai pra uma nova linha no prompt , depois de executar esse comando consigo usar o apt-get novamente porem no final sempre apresenta a mesma mensagem sobre o hydra.

Participante 4: Comenta as linhas de repositórios que adicionou. o Hydra esta funcionando normalmente? Tenta reinstala-lo!

Participante 5: Kra apaga o pacote defeituoso: sudo rm -rf var/cache/apt/archives/hydra_5.4-bt1_all.deb. reconstrua o banco de dados do dpkg: sudo dpkg --configure -a. Limpa qualquer pacote da memoria do apt-get. sudo apt-get -f install. sudo apt-get -f remove. sudo apt-get clean. sudo apt-get autoremove. Assim deve resolver seus problemas...

Participante 1: amigo tentei seguir os passos q falou e a resposta foi a seguinte: megatron:/home/insigne# sudo rm -rf var/cache/apt/archives/hydra_5.4-bt1_all.deb megatron:/home/insigne# sudo dpkg --configure -a . megatron:/home/insigne# sudo apt-get -f install. Lendo Lista de Pacotes... Pronto. Construindo Árvore de Dependências... Pronto. Os pacotes extra a seguir serão instalados: hydra. Os pacotes a seguir serão atualizados : hydra. 1 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 199 não atualizados. 1 pacotes não totalmente instalados ou removidos. É preciso fazer o download de 0B/83,3kB de arquivos. Depois de desempacotamento, 81,9kB adicionais de espaço em disco serão usados. Quer continuar [S/n] ? AVISO : Os pacotes a seguir não podem ser autenticados ! hydra. Aviso de autenticação sobrescrito. Selecionando pacote previamente não selecionado hydra. (Lendo banco de dados ... dpkg: aviso sério: arquivo com lista de arquivos do pacote `hydra' está falando, assumindo que pacote não tem arquivos instalados atualmente. 140801 arquivos e diretórios atualmente instalados.)Preparando para substituir hydra 5.4-bt1 (usando ../archives/hydra_5.4-bt1_all.deb) ... Descompactando substituto hydra ... dpkg-deb: arquivo `/var/cache/apt/archives/hydra_5.4-bt1_all.deb' contém membros de dados não entendidos data.tar.lzma , desistindo E: Sub-process /usr/bin/dpkg received a segmentation fault. megatron:/home/insigne# sudo apt-get clean. megatron:/home/insigne# sudo apt-get autoremove. E: Operação autoremove inválida. quando tento abrir o synaptic apresenta o erro: E: dpkg was interrupted, you must

manually run 'dpkg --configure -a' to correct the problem. Quando executo dpkg --configure -a consigo abrir, mas qdo seleciono algum pacote pra instalar ele tb tenta instalar o hydra e acaba dando erro. Quando seleciono o pacote a instalar e depois vou em detalhes ta o maldito hydra lá. hydra (versão 5.4-bt1) será atualizado para a versão 5.4-bt1. desde já agradeço sua ajuda!

Participante 1: uma força alguém? parece q esse pacote corrompeu o synaptic

Participante 6: Deixa no seu sources.list somente os repositórios de segurança e oficial, remove os arquivos do cache por completo: #apt-get clean; ou entra na pasta e apaga todo o cache; #cd /var/cache/apt/archives/; #rm *.deb; #dpkg --configure -a; #aptitude update; #aptitude safe-upgrade; #aptitude --purge remove hydra; #aptitude -f install

Participante 2: tenta remover o pacote hydra do cache: rm -rf /var/cache/apt/archives/ hydra_5.4-bt1_all.deb. Se voce achar mais facil: cd /var/cache/apt/archives; rm -rf hydra_5.4-bt1_all.deb. Depois baixa ele e instala de novo.

Participante 1: agora complicou tudo uauuaha copieie o comando rm -rf /var/cache/apt/archives/ hydra_5.4-bt1_all.deb e excluiu a pasta archives. tinha espaço e não vi.

Participante 2: só recriar ela de novo: mkdir /var/cache/apt/archives/. Ve se o hydra ta la ainda.

Participante 1: coloquei o cd e recuperei o sistema graças a Deus o hydra foi embora e o apt-get voltou ao normal, mas infelizmente o repositório do backtrack que havia colocado no repositório nao esta mais e agora não sei que repositório colocar pra conter ele =/

Participante 8: Quando um pacote não instala e a sua instalação é tentada cada vez que se roda o APT, a solução é esta: dpkg --remove --force-remove-reinstreq pacote-com-problema. Fica a dica aqui para quem precisar.

A discussão exemplificada não acaba e não é identificada uma resposta que seja a mais coerente ou a que solucionou o problema ou parte deste (mais qualificada), nem o tema discutido. Assim, se qualquer outro participante estiver com o mesmo problema terá que, primeiro encontrar esta discussão no repositório de interações, depois, ler todas as mensagens, tentando seguir a lógica e o raciocínio (não explícito) de seus participantes, e por fim, testar as dicas (boas ou ruins) e escolher a melhor. O modelo ArFoLi permite que um ciclo organizado de discussão exista pela possibilidade de responder uma questão, além de gerar novas questões para resolução de um problema. A questão segue com as respostas e os argumentos delineados para que qualquer outra pessoa ao ler a discussão a interprete adequadamente e o encerramento conclui a discussão.

A mesma discussão estruturada pelo modelo ArFoLi é a seguinte:

Questão <Participante 1>: amigos, eu tinha tentado instalar o hydra por apt-get modificando uns repositórios, e agora sempre q uso o apt-get para instalar algum programa ele apresenta a mensagem: dpkg: aviso sério: arquivo com lista de arquivos do pacote 'hydra' está faland, assumindo que pacote não tem arquivos instalados atualmente.140801 arquivos e diretórios atualmente instalados.) Preparando para substituir hydra 5.4-bt1 (usando ../archives/hydra_5.4-bt1_all.deb) ... Descompactando substituto hydra ... dpkg-deb: arquivo '/var/cache/apt/archives/hydra_5.4-bt1_all.deb' contém membros de dados não entendidos data.tar.lzma , desistindo Erro: Sub-process /usr/bin/dpkg received a segmentation fault. em seguida para ao tentar executar o apt-get. Erro: dpkg was interrupted, you must manually run 'dpkg --configure -a' to correct the problem. Ae tenho q usar o comando dpkg --configure -a pra poder executar o apt-get, alguém sabe cm resolver? obrigado\

Resposta <Participante 2>: digitando dpkg --configure -a o que dá??

Qualificação: + 5

Questão <Participante 1>: não apresenta nenhuma mensagem apenas vai pra uma nova linha no prompt ,

depois de executar esse comando consigo usar o apt-get novamente porem no final sempre apresenta a mesma mensagem sobre o hydra.

Resposta <Participante 4>: Comenta as linhas de repositórios que adicionou. o Hydra esta funcionando normalmente? Tenta reinstala-lo! **Qualificação: + 1**

Argumento <Participante 5> apóia: Kra apaga o pacote defeituoso: sudo rm -rf var/cache/apt/archives/hydra_5.4-bt1_all.deb. reconstrua o banco de dados do dpkg: sudo dpkg --configure -a. Limpa qualquer pacote da memoria do apt-get. sudo apt-get -f install. sudo apt-get -f remove. sudo apt-get clean. sudo apt-get autoremove. Assim deve resolver seus problemas... **Qualificação: + 1**

Questão <Participante 1>: amigo tentei seguir os passos q falou e a resposta foi a seguinte:
megatron:/home/insigne# sudo rm -rf var/cache/apt/archives/hydra_5.4-bt1_all.deb
megatron:/home/insigne# sudo dpkg --configure -a . megatron:/home/insigne# sudo apt-get -f install. Lendo Lista de Pacotes... Pronto. Construindo Árvore de Dependências... Pronto. Os pacotes extra a seguir serão instalados: hydra. Os pacotes a seguir serão atualizados : hydra. 1 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 199 não atualizados. 1 pacotes não totalmente instalados ou removidos. É preciso fazer o download de 0B/83,3kB de arquivos. Depois de desempacotamento, 81,9kB adicionais de espaço em disco serão usados. Quer continuar [S/n] ? AVISO : Os pacotes a seguir não podem ser autenticados ! hydra. Aviso de autenticação sobrescrito. Selecionando pacote previamente não selecionado hydra. (Lendo banco de dados ... dpkg: aviso sério: arquivo com lista de arquivos do pacote `hydra' está faland, assumindo que pacote não tem arquivos instalados atualmente. 140801 arquivos e diretórios atualmente instalados.)Preparando para substituir hydra 5.4-bt1 (usando .../archives/hydra_5.4-bt1_all.deb) ... Descompactando substituto hydra ... dpkg-deb: arquivo `var/cache/apt/archives/hydra_5.4-bt1_all.deb' contém membros de dados não entendidos data.tar.lzma , desistindo E: Sub-process /usr/bin/dpkg received a segmentation fault.
megatron:/home/insigne# sudo apt-get clean. megatron:/home/insigne# sudo apt-get autoremove.
E: Operação autoremove inválida. quando tento abrir o synaptic apresenta o erro: E: dpkg was interrupted, you must manually run 'dpkg --configure -a' to correct the problem. Quando executo dpkg --configure -a consigo abrir, mas qdo seleciono algum pacote pra instalar ele tb tenta instalar o hydra e acaba dando erro. Quando seleciono o pacote a instalar e depois vou em detalhes ta o maldito hydra lá. hydra (versão 5.4-bt1) será atualizado para a versão 5.4-bt1. desde já agradeço sua ajuda!
uma força alguem? parece q esse pacote corrompeu o synaptic

Resposta <Participante 6>: Deixa no seu sources.list somente os repositorios de segurança e oficial, remove oo arquivos do cache por completo: #apt-get clean; ou entra na pasta e apaga todo o cache; #cd /var/cache/apt/archives/; #rm *.deb; #dpkg --configure -a; #aptitude update; #aptitude safe-upgrade; #aptitude --purge remove hydra; #aptitude -f install **Qualificação: + 5**

Resposta <Participante 2>: tenta remover o pacote hydra do cache: rm -rf /var/cache/apt/archives/ hydra_5.4-bt1_all.deb. Se voce achar mais facil: cd /var/cache/apt/archives; rm -rf hydra_5.4-bt1_all.deb. Depois baixa ele e instala de novo. **Qualificação: + 4**

Questão <Participante 1>: agora complicou tudo uahuaha copiei o coamndo rm -rf /var/cache/apt/archives/ hydra_5.4-bt1_all.deb e excluiu a pasta archives. tinha espaço e não vi.

Resposta <Participante 2>: só recriar ela de novo: mkdir /var/cache/apt/archives/. Ve se o hydra ta la ainda.

Qualificação: - 1

Argumento <Participante 1> apóia: coloquei o cd e recuperei o sistema graças a Deus o hydra foi embora e o apt-get voltou ao normal, mas infelizmente o repositório do backtrack que havia colocado no repositório nao esta mais e agora não sei que repositório colocar pra conter ele =/

Argumento <Participante 8> apóia: Quando um pacote não instala e a sua instalação é tentada cada vez que se roda o APT, a solução é esta: `dpkg --remove --force-remove-reinstreq pacote-com-problema`. Fica a dica aqui para quem precisar.

Encerramento <Participante 1>: A resposta do participante 8 funcionou corretamente.

Cenário 2 – O participante precisa desenvolver um script.

A discussão não estruturada pelo modelo é a seguinte:

Participante 1: Fazer um script shell para listar os arquivos de um diretório especificado e dizer se o arquivo é do tipo diretório ou arquivo comum. Como faço isso? Sou iniciante no linux [usá-lo por código] e queria saber como faço esse script! Abraços!

Participante 2: Da uma olhada neste scrip e ve se consegue entender o que eu fiz.

```
#!/bin/sh
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha; do
if [ -d /home/backup/$linha ]; then
echo $linha "eh pasta"
elif [ -f /home/backup/$linha ]; then
echo $linha "eh um arquivo"
else
echo $linha "sei la o que eh"
fi
done
ABS
```

Participante 1: Não entendi isso

```
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha;
```

Participante 3: O que o amigo fez acima:

1. `ls /home/backup/ > /home/backup/lista`

A listagem do conteúdo do diretório `/home/backup/` é jogada no arquivo `/home/backup/lista`

2. `cat /home/backup/lista | while read linha;`

É lido o arquivo criado no passo 1, e linha a linha é passada em laço (while), sendo posteriormente analisada testando-se ser diretório ou arquivo.

Participante 4 –

```
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha;
```

=====

```

1° linha
ls "lista arquivos e diretorios"
/home/backup "seria o diretorio escolhido"
/home/backup/lista "seria um arquivo que esta sendo criado, e o conteudo dele eh tudo oq o comando ls listou dentro do diretorio
/home/backup"
2° linha
cat /home/backup/lista "ira listar todo o conteudo do arquivo lista"
while "testa continuamente (um loop) uma expressão até que ela se torne falsa"
read linha "nao entendo muito de script, mas parece que ele esta criando uma variavel chamada linha, e linha ($linha) vai significar
todo o conteudo do arquivo lista"
=====

vcs devem ter visto que nao sei muito de script, mas foi isso que entendi
rodei aki, mas fiz algumas alteraçoes
=====

#/bin/sh
ls /home/douglas/ > /home/douglas/lista # lista um caminho e joga para um arquivo
cat /home/douglas/lista | while read linha; do
if [ -d /home/douglas/$linha ]; then
echo $linha "eh pasta"
elif [ -f /home/douglas/$linha ]; then
echo $linha "eh um arquivo"
else
echo $linha "sei la o que eh"
fi
done
=====

e o resultado foi esse
=====

douglas@CASA:~$ sh teste.sh
Desktop eh pasta
FrostWire eh pasta
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Musica Electronica Here We Go-Kaylab.mp3 sei la o que eh
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Sem título 1.odt sei la o que eh
backup eh pasta
download eh pasta
imagens eh pasta
jogos eh pasta
lista eh um arquivo
outros eh pasta
programas eh pasta
programação eh pasta
scripts eh pasta
sons eh pasta
teste.sh eh um arquivo
=====

soh nao sei oq foi isso

```

```

teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Musica Electronica Here We Go-Kaylab.mp3 sei la o que eh
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Participante 2: Vamos destrinchar então, estas linhas querem dizer:
"Não entendi isso
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha;"
ls /home/backup/ # é o diretório que vc especifica, no meu caso esta usando a pasta home do usuario backup
# o sinal de maior ">" joga qualquer saída para um determinado arquivo, subscrevendo caso exista informações neste arquivo, o >>
( 2 vezes) acrescenta, e 2> caso exista algum erro, são os mais usuais.
#depois do maior (no meu caso /home/backup/lista) é o caminho que vai salvar a saída do ls e o nome lista, o nome que eu dei para
o meu arquivo. /caminho/nome_do_arquivo
#a proxima linha cat /home/backup/lista | while read linha;"
o comando é uma estrutura que le o arquivo linha a linha através do comando cat + read le com o read e joga temporariamente na
variavel linha, que é utilizada algumas vezes com $( cifrão, dolar, cifra, cada um chama de um jeito) e a "saída" temporaria linha,
assim $linha.
espero não ter complicado, qualquer coisa responde ai.
Participante 2: boa tarde. até entendi o que esta acontecendo, mas não tenho conhecimento para te explicar o pq, mas o problema é
causado pelo espaço em branco no nome do arquivo, Musica Electronica Here We Go-Kaylab.mp3 ABS
Participante 4: hum, tendi. vlw msm. su script esta muito bem explicado =)
Participante 5: Bem, não sei se é isso que você queria, mas, segue um script feito por mim, com alguma formatação.
#!/bin/bash
ARQ=$(mcookie)
for i in $(ls ./ | sed 's/ /%/g'); do
if [ -d $(echo $i | sed 's/ /%/g') ]; then
echo "$i : é um diretório"
else
echo "$i : é um arquivo comum ou um link"
fi
done >>/tmp/${ARQ}.txt
column -c 2 -t -s ':' /tmp/${ARQ}.txt
rm -f /tmp/${ARQ}.txt
Participante 1: Vlw galera! Entendi galera! Valeu pela força!!! Qualquer outra dúvida que surja estarei aqui! Estão de parabéns :D
Participante 4: xD. marca como melhor resposta, aquela que vc usou para resolver seu problema. xD. flw

```

A discussão exemplificada novamente não é terminada e, desta vez, um participante solicita inclusive a melhor resposta. A necessidade de identificação da melhor resposta fica evidente. Nesse segundo cenário, ainda existe os scripts, que se não forem organizados por respostas ou argumentos, não é possível identificar a melhor forma de aplicar tal script ou qual script está correto ou o que está faltando na resposta de um participante.

A mesma discussão estruturada pelo modelo é a seguinte:

Questão <Participante 1>: Fazer um script shell para listar os arquivos de um diretório especificado e dizer se o arquivo é do tipo diretório ou arquivo comum. Como faço isso? Sou iniciante no linux [usá-lo por código] e queria saber como faço esse script! Abraços!

Resposta <Participante 2>: Da uma olhada neste scrip e ve se consegue entender o que eu fiz.

```
#!/bin/sh
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha; do
if [ -d /home/backup/$linha ]; then
echo $linha "eh pasta"
elif [ -f /home/backup/$linha ]; then
echo $linha "eh um arquivo"
else
echo $linha "sei la o que eh"
fi
done
ABS
```

Qualificação: + 8

Questão <Participante 1>: Não entendi isso

```
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
cat /home/backup/lista | while read linha;
```

Resposta <Participante 3>: O que o amigo fez acima:

1. ls /home/backup/ > /home/backup/lista

A listagem do conteudo do diretório /home/backup/ é jogada no arquivo /home/backup/lista

2. cat /home/backup/lista | while read linha;

É lido o arquivo criado no passo 1, e linha a linha é passada em laço (while), sendo posteriormente analisada testando-se ser diretório ou arquivo.

Participante 4 –

```
ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo
```

```
cat /home/backup/lista | while read linha;
```

=====

1º linha

```
ls "lista arquivos e diretorios"
```

```
/home/backup "seria o diretorio escolhido"
```

```
/home/backup/lista "seria um arquivo que esta sendo criado, e o conteudo dele eh tudo oq o comando ls listou dentro do diretorio /home/backup"
```

2º linha

```
cat /home/backup/lista "ira listar todo o conteudo do arquivo lista"
```

```
while "testa continuamente (um loop) uma expressão até que ela se torne falsa"
```

```
read linha "nao entendo muito de script, mas parece que ele esta criando uma variavel chamada linha, e linha ($linha) vai significar todo o conteudo do arquivo lista"
```

=====

ves devem ter visto que nao sei muito de script, mas foi isso que entendi

rodei aki, mas fiz algumas alteraçoes

=====

```
#!/bin/sh
```

```
ls /home/douglas/ > /home/douglas/lista # lista um caminho e joga para um arquivo
```

```
cat /home/douglas/lista | while read linha; do
```

```

if [ -d /home/douglas/$linha ]; then
echo $linha "eh pasta"
elif [ -f /home/douglas/$linha ]; then
echo $linha "eh um arquivo"
else
echo $linha "sei la o que eh"
fi
done
=====

e o resultado foi esse
=====

douglas@CASA:~$ sh teste.sh
Desktop eh pasta
FrostWire eh pasta
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Musica Electronica Here We Go-Kaylab.mp3 sei la o que eh
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Sem título 1.odt sei la o que eh
backup eh pasta
download eh pasta
imagens eh pasta
jogos eh pasta
lista eh um arquivo
outros eh pasta
programas eh pasta
programação eh pasta
scripts eh pasta
sons eh pasta
teste.sh eh um arquivo
=====

soh nao sei oq foi isso
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments
Musica Electronica Here We Go-Kaylab.mp3 sei la o que eh
teste.sh: line 5: [: too many arguments
teste.sh: line 7: [: too many arguments

```

Qualificação: + 3

Argumentação <Participante 2>: Vamos destrinchar então, estas linhas querem dizer:

"Não entendi isso

ls /home/backup/ > /home/backup/lista # lista um caminho e joga para um arquivo

cat /home/backup/lista | while read linha;"

ls /home/backup/ # é o diretório que vc especifica, no meu caso esta usando a pasta home do usuário backup

o sinal de maior " > " joga qualquer saída para um determinado arquivo, subscrivendo caso exista informações neste arquivo, o >> (2 vezes) acrescenta, e 2> caso exista algum erro, são os mais usuais.

#depois do maior (no meu caso /home/backup/lista) é o caminho que vai salvar a saída do ls e o nome lista, o nome que eu dei para o meu arquivo. /caminho/nome_do_arquivo

#a proxima linha cat /home/backup/lista | while read linha;"

o comando é uma estrutura que lê o arquivo linha a linha através do comando `cat + read` e com o `read` joga temporariamente na variável `linha`, que é utilizada algumas vezes com `$` (cifrão, dólar, cifra, cada um chama de um jeito) e a "saída" temporária linha, assim `$linha`.
espero não ter complicado, qualquer coisa responde aí.

Resposta <Participante 2>: boa tarde. até entendi o que está acontecendo, mas não tenho conhecimento para te explicar o pq, mas o problema é causado pelo espaço em branco no nome do arquivo, Musica Electronica Here We Go-Kaylab.mp3 ABS **Qualificação: + 5**

Resposta <Participante 4>: hum, tendi. vlw msm. su script está muito bem explicado =) **Qualificação: + 4**

Argumentação <Participante 5>: Bem, não sei se é isso que você queria, mas, segue um script feito por mim, com alguma formatação.

```
#!/bin/bash
ARQ=$(mcookie)
for i in $(ls ./ | sed 's/ /*g'); do
if [ -d $(echo $i | sed 's/ /*g') ]; then
echo "$i : é um diretório"
else
echo "$i : é um arquivo comum ou um link"
fi
done >>/tmp/${ARQ}.txt
column -c 2 -t -s ' ' /tmp/${ARQ}.txt
rm -f /tmp/${ARQ}.txt
```

Qualificação: + 6

Encerramento <Participante 1>: Vlw galera! Entendi galera! Valeu pela força!!! Qualquer outra dúvida que surja estarei aqui!
Estão de parabéns :D. A resposta do participante 2 com o script funcionou corretamente.

Os dois cenários reais apresentados e analisados evidenciam a necessidade de estruturação, pois as questões e respostas estão sempre misturadas, dificultando o entendimento da discussão e indicam que o encerramento é primordial, para que uma discussão tenha uma conclusão. Se os cenários fossem identificados pelos seus temas e suas *tags*, a identificação do que estava sendo debatido seria mais clara e as palavras-chave da discussão apontariam também sobre qual é o assunto discutido, identificando o problema debatido ou a versão da ferramenta em desenvolvimento ou o *bug* encontrado.

O modelo de argumentação proposto, ArFoLi, permite que a estruturação da interação aconteça da seguinte forma:

- questão é criada, ou seja, o problema é exposto;
- resposta é inserida para explicar a questão;
- argumentos são colocados como os pontos de vista a favor ou contra uma resposta;
- *tag* inseridas identificam o conteúdo das mensagens;

- qualificação apoia na discussão encontrar as respostas que estão mais adequadas por meio dos pontos inseridos;
- encerramento possibilita concluir uma discussão, fechá-la de forma que fique claro o que aconteceu e se existiu a melhor resposta.

O modelo de argumentação ArFoLi proposto foi implementado em uma ferramenta de fórum pra fins de avaliação, como pode ser visto no próximo capítulo.

Capítulo 7 – Fórum Estruturado para Comunidades Virtuais de *Software* Livre - Fórum Livre

Este capítulo descreve a implementação de um fórum estruturado para apoiar as interações entre os participantes das Comunidades Virtuais de *Software* Livre, visando a facilitar o acesso às mensagens trocadas e aos artefatos disponibilizados nas interações, estruturando o conhecimento gerado. A escolha do fórum foi feita devido à evidência encontrada no Estudo Exploratório nas Comunidades Virtuais de *Software* Livre onde a maioria das interações acontece através de fóruns. O fórum estruturado é baseado no Modelo de Argumentação para Fórum de *Software* Livre, ArFoLi, que atende às particularidades das Comunidades Virtuais de *Software* Livre.

Por meio do Estudo Exploratório, foi possível evidenciar os problemas de estruturação de conhecimento nas Comunidades Virtuais de *Software* Livre e desenvolver o modelo de argumentação, ArFoLi. Para avaliar o modelo proposto, a ferramenta a ser construída deveria ser utilizada pelas comunidades, sendo assim, o tipo de ferramenta escolhida foi o fórum, pois de acordo com o Estudo Exploratório é uma das ferramentas mais utilizadas nas interações das Comunidades Virtuais de *Software* Livre.

A ferramenta de fórum, chamada Fórum Livre, tem como objetivo estruturar as interações realizadas nas Comunidades Virtuais de *Software* Livre, aplicando o modelo de argumentação, ArFoLi e possibilitar que os participantes discutam seus problemas, desenvolvam suas ferramentas e acessem o conhecimento gerado mais facilmente.

7.1. Fórum estruturado

O modelo de argumentação, ArFoLi, foi incorporado na ferramenta *Free Open Source* de fórum chamada Vanilla. A ferramenta Vanilla foi selecionada porque poderia ser customizada (código fonte disponível), atendia aos princípios de liberdade do *Software* Livre.

Além do modelo de argumentação embutido, a ferramenta inclui o conceito de etiqueta ou *tag* (GOLDER e BERNARDO, 2005), ou seja, a possibilidade de inserção de palavras-chaves da discussão pelos participantes, permitindo uma liberdade de expressão nas associações e permitindo enriquecer a discussão. A *tag* tem a função de facilitar o acesso ao conteúdo da resposta inserida na discussão do fórum. Ela é baseada no conceito da Web 2.0 (O'REILLY, 2005), no qual a colaboração e a interação nos ambientes virtuais são fundamentais. A *tag* é inserida na resposta e pode ser visualizada como uma nuvem de palavras-chaves ao lado da discussão. Esse recurso promove o acesso ao conhecimento gerado por uma resposta, pois o participante acessa a discussão em que a palavra de interesse está. Um exemplo é a palavra Linux, o sistema exibirá todas as discussões em que um participante a usou como *tag* em contribuição (NASCIMENTO, SANTORO e VELOSO, 2009).

O fórum estruturado contém um repositório de artefatos separando-os por discussão e facilitando o acesso de participantes externos aos artefatos inseridos em uma discussão. Esse repositório apóia o participante na busca por um determinado artefato de uma discussão específica (NASCIMENTO, SANTORO e VELOSO, 2009).

7.2. Características técnicas do fórum estruturado

A ferramenta de fórum livre Vanilla foi customizada na versão 1.18, rodando no Sistema Operacional Linux Ubuntu versão 9.04 (<http://www.ubuntu.com>), com banco de dados My SQL versão 5.075 (<http://www.ubuntu.com>), com servidor web Apache versão 2.2.11 e desenvolvida em PHP versão 5.2.6. Os browsers que a ferramenta suporta são: Firefox, Safari, Camino, Opera e Internet Explorer.

A ferramenta Vanilla, encontra-se disponível em <http://getvanilla.com/>, na versão 1.1.8 e *release* May de 2009. Ela é uma ferramenta *free open-source* para a Web, com padrões abertos (os módulos podem alterados, licença GPL) e pode ser adquirida livremente para modificações.

A comunidade que discute e desenvolve a ferramenta Vanilla se chama *Lussumo Community*, Figura 23.

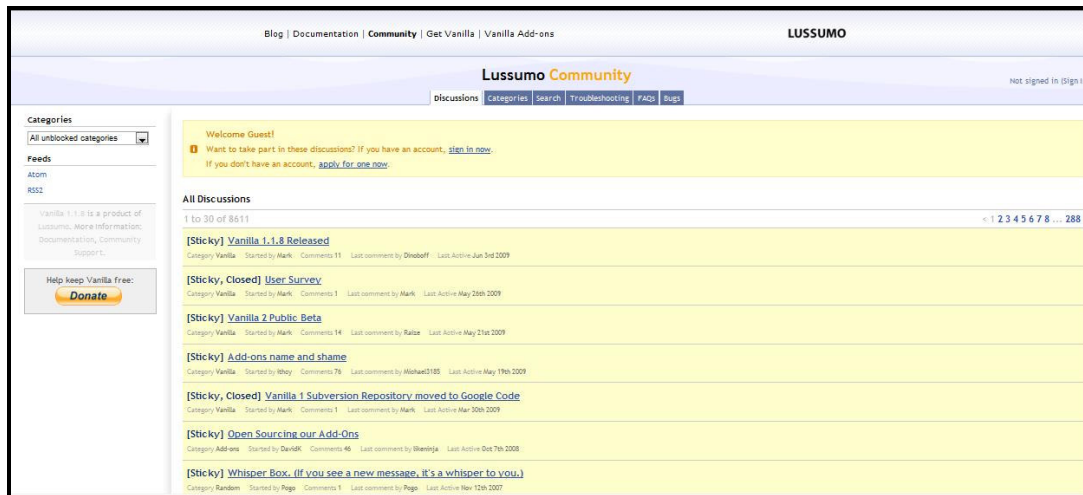


Figura 23 - Comunidade Lussumo

Os módulos customizados foram: *Attachments* versão 2.1, *Community Moderation* versão 1.3, *Dojo Files* versão 0.941, *Subcategories* versão 0.2.3, *Tag This* 1.08. Esses módulos foram alterados para possibilitar a estruturação da discussão no

fórum, ou seja, a inserção do modelo de argumentação, ArFoLi. O fórum com as novas funcionalidades de estruturação será disponibilizado na comunidade Vanilla como uma nova *release*.

7.3. Funcionalidades do fórum estruturado

O fórum tem como características a categorização dos temas e dos tipos de questões a serem exploradas, as mensagens são organizadas como respostas ou argumentos, as avaliações qualificam as respostas e o encerramento é a conclusão da interação. O Fórum Livre possibilita que os usuários sejam designados a dois papéis: o do moderador e o do participante. Nesse caso o moderador é o responsável pela interação, abre a interação e encerra quando achar conveniente, isto é, quando seu problema estiver solucionado. O participante é o contribuinte, o que apresenta possíveis formas de solucionar o problema em questão. O administrador realiza as configurações necessárias no fórum, ou seja, pode criar novas categorias, aceitar novos membros, criar uma nova categoria.

A categorização proposta no Fórum Livre está ligada ao tema discutido no fórum e a questão debatida, é uma forma de organizar a interação por tema. Como por exemplo, a categoria Sistema Operacional, na qual todas as interações sobre Sistema Operacional ficam organizadas nesta categoria.

As mensagens da interação podem ser classificadas como respostas ou argumentos, conforme o modelo de argumentação apresentado no capítulo anterior, o ArFoLi. As respostas são as soluções dadas à questão pelos participantes, enquanto o argumento é uma crítica a resposta, podendo ser contra ou a favor da resposta.

As *tags* são as palavras-chaves referentes à resposta de cada participante na interação do fórum, gerando novo conhecimento. As *tags* são exibidas por interação e são sempre exibidas na tela como uma nuvem de *tags*, ou seja, um conjunto de palavras-chaves daquela interação. Essas *tags* são inseridas pelo participante no momento em que ele está escrevendo sua resposta. As *tags* identificam a discussão, ou seja, permitem que os participantes sejam levados à discussão identificada por elas. Esse recurso promove o acesso ao conhecimento gerado por uma resposta, pois o participante acessa a discussão em que a palavra de interesse está.

A qualificação é a funcionalidade que ajuda a avaliação da resposta por meio de uma pontuação dada, onde o participante adiciona ou retira um ponto. No encerramento

da discussão no fórum é possível identificar uma pontuação para as respostas, apoiando o moderador na escolha de respostas mais adequadas a seu problema.

Os artefatos são documentos (código fonte, artigo, dica, notícia, manual) inseridos como anexo à mensagem (resposta ou argumento) e ficam no repositório referente à interação categorizado no fórum. Os artefatos podem ser recomendados por *e-mail* para outros participantes.

O encerramento do fórum é realizado pelo moderador e permite que o mesmo faça um comentário sobre a interação e indique respostas que achar mais convenientes para a solução do problema levantado.

A estruturação do Fórum Livre para atingir o objetivo de categorizar as mensagens introduzindo o conceito de *tags* para facilitar a busca e a reutilização do conhecimento e ainda organizar os artefatos disponibilizados, é a seguinte:

- **Tema** da discussão, iniciada com o nome do moderador, a data e a hora da criação da interação.
- **Questão** contém o problema discutido. Quando durante a discussão é necessário criar uma questão, essa será chamada de nova questão.
- **Resposta** é o ponto de vista de participante, possibilita a inclusão de **Tag**, mostra o participante que respondeu, a data e a hora da resposta.
- **Argumento** é o apoio ou a contestação da resposta de um participante por outro participante, possibilita a inclusão de *tags*, mostra o participante que a respondeu, a data e a hora da argumentação.
- **Pontuação** é a avaliação da resposta que receberá pontos dos participantes (qualificação). Cada participante pontua em respostas diferentes da sua e uma vez por login.
- **Encerramento** é a finalização da interação, ou seja, quando o moderador conclui a discussão, insere um comentário onde indica a(s) melhor(es) resposta(s), possui a data e a hora do encerramento. Quando a discussão estiver encerrada, na página inicial do fórum, discussão aparece como “FECHADA”.

Assim, o Fórum Livre e sua estruturação podem ser visualizados na Figura 24.

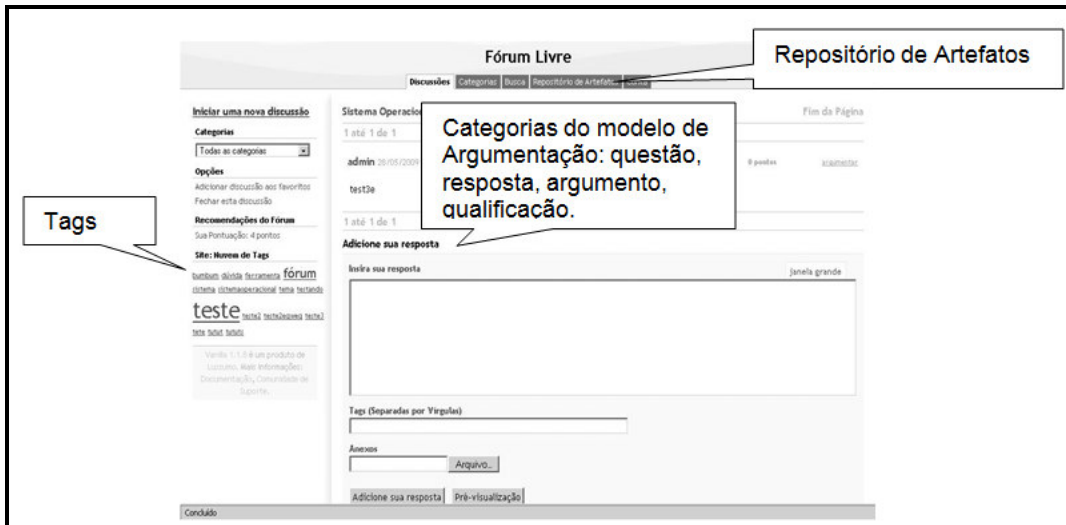


Figura 24 - Estruturação do Fórum Livre

O Fórum Livre possibilita ao participante visualizar todas as discussões de uma categoria ou todas as discussões do fórum, identificando as encerradas (Fechada) e as *tags* das discussões, conforme a Figura 25.

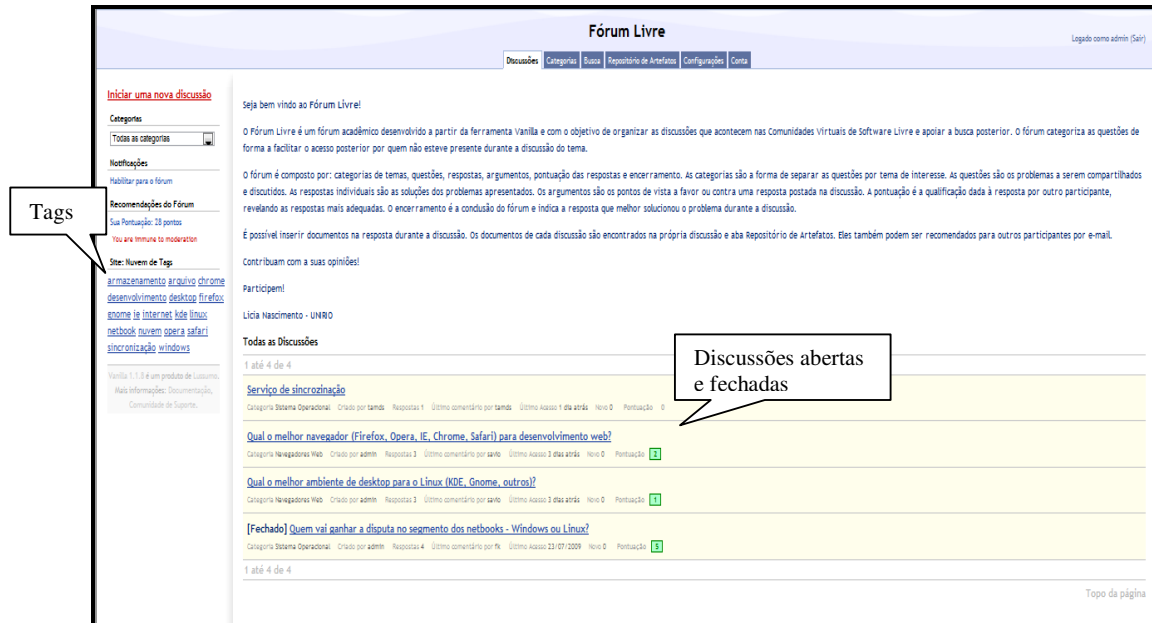


Figura 25 - Tela de discussões

As categorias são identificadas por uma aba específica no Fórum Livre, para organizar as discussões e conseqüentemente o acesso as discussões de uma determinada categoria, conforme a Figura 26.

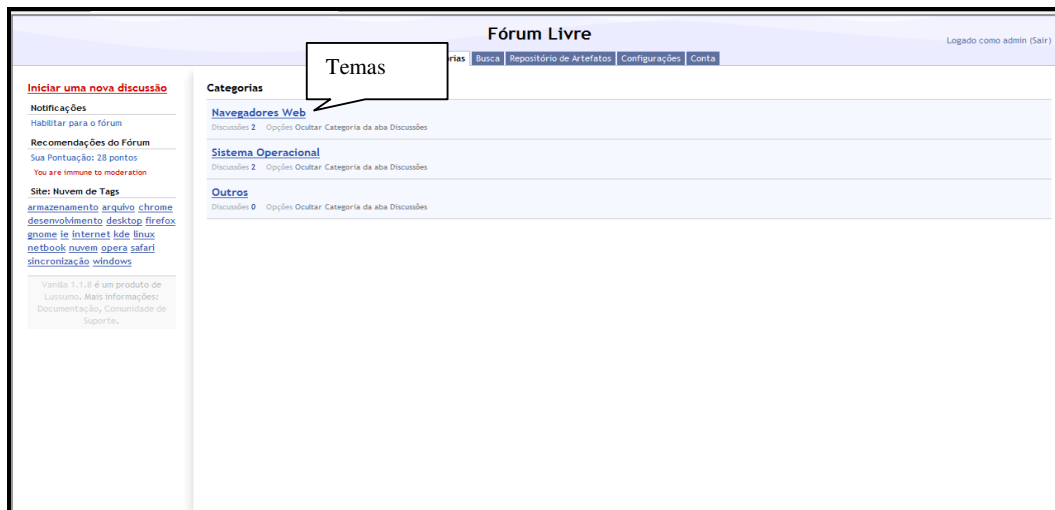


Figura 26 - Tela das categorias

Uma discussão realizada no Fórum Livre está categorizada e estruturada de acordo com o modelo ArFoLi (questão, resposta, argumentos, novas questões e pontuação). Na Figura 27 é apresentada a questão em destaque para a discussão, as respostas, a qualificação da resposta e a quantidade de argumentos e novas questões que resposta possui. A pontuação permite a qualificação da resposta pelos participantes.

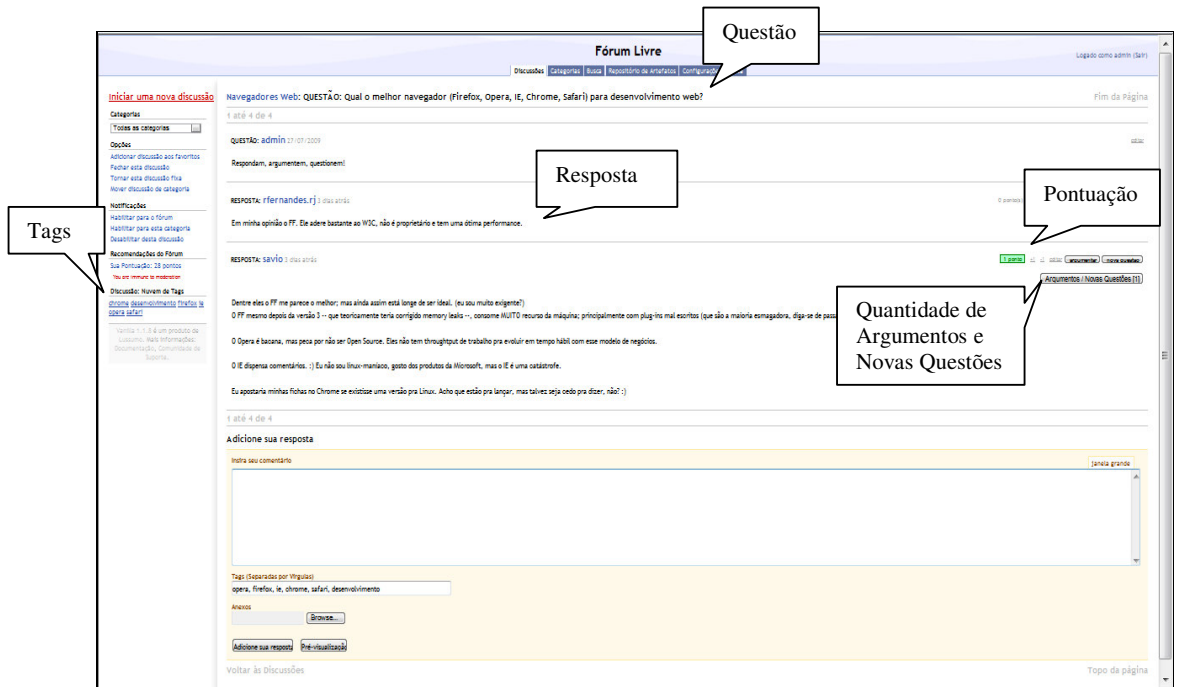


Figura 27 - Tela de uma discussão

A Figura 28 apresenta uma resposta contendo uma nova questão e um argumento, ou seja, o encadeamento de ideias referentes a uma resposta postada no fórum, podendo estar apoiando ou não a resposta e gerando mais debate pela nova questão inserida. Isso mostra que uma resposta gera uma nova questão, seja por falta de informação ou pela necessidade de detalhamento da mesma.

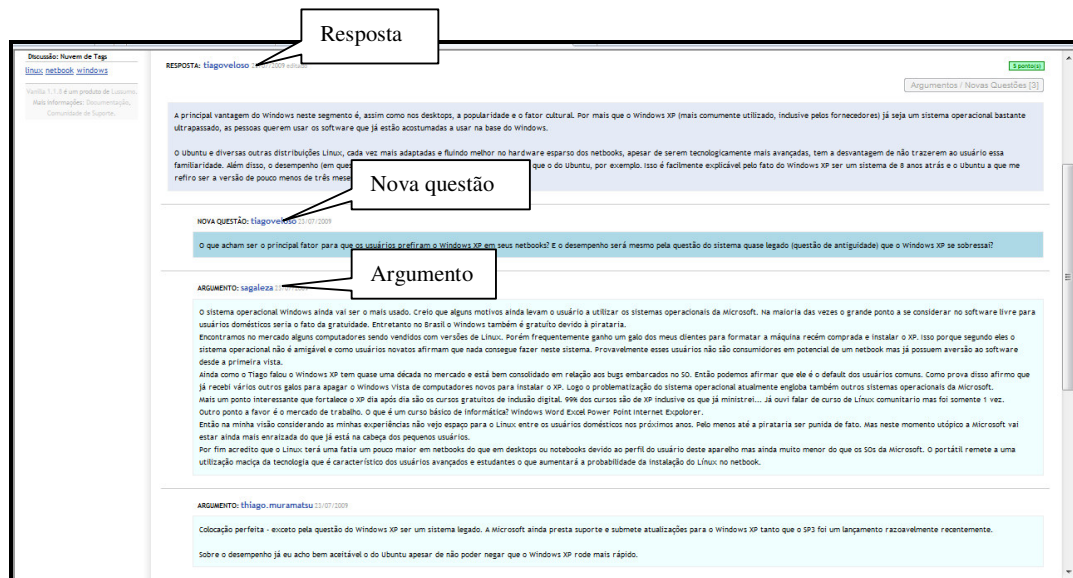


Figura 28 – Tela com resposta, nova questão e argumento

Quando a discussão é encerrada, o moderador do fórum insere uma mensagem identificando a melhor ou melhores respostas e finaliza-a. Na tela de discussões, aquela interação é mostrada como “Fechada”. A Figura 29 mostra a discussão com o encerramento.

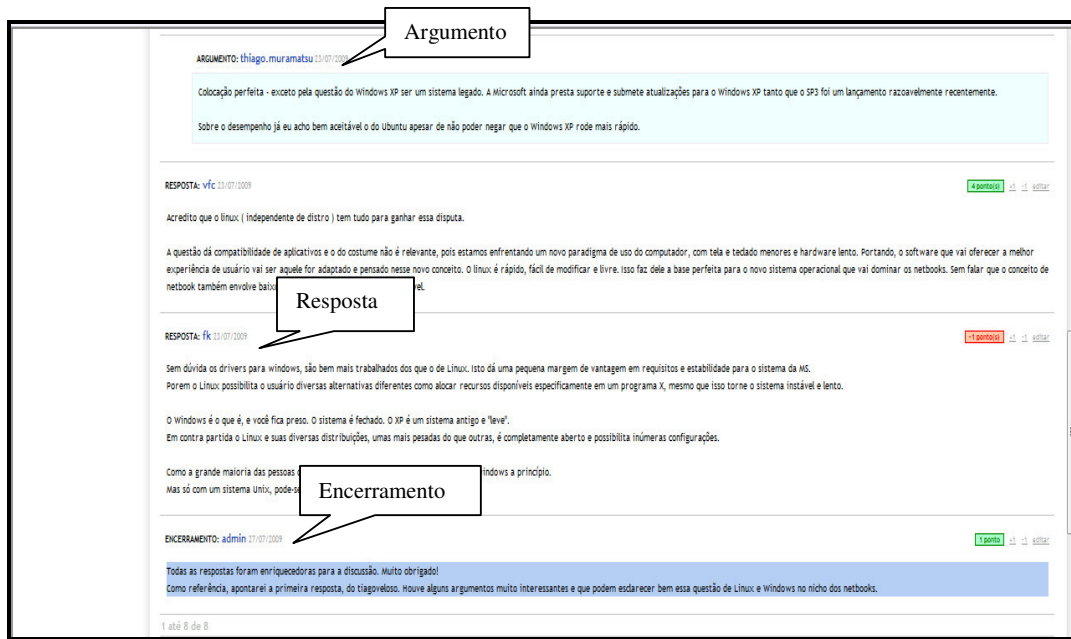


Figura 29 - Tela de encerramento da discussão

No Apêndice B, são descritos os Requisitos Funcionais do Fórum Livre que serviram como base para customizar funcionalidades da ferramenta de fórum Vanilla de acordo com o modelo de argumentação, ArFoLi. No Apêndice C contém os Casos de Uso de como o fórum estruturado se comportaria e como os atores envolvidos utilizariam o fórum estruturado.

A estruturação do fórum possibilita que participantes da comunidade que não estavam presentes na discussão ao acessá-la identifiquem qual a questão discutida, as respostas e suas pontuações, os argumentos das respostas, as novas questões que surgiram e como foi o desfecho da discussão, o que facilita a recuperação do conhecimento gerado durante a discussão. Com isso, é possível que o problema identificado em acessar e buscar o conhecimento nas interações seja minimizado de forma que mesmo que não participou da interação identifique o ocorrido e encontre o que deseja para resolver o seu problema. Já para quem participou da interação, a organização facilita a participação e orienta a discussão. As Comunidades Virtuais de

Software Livre possuem esse princípio de colaboração e liberdade presente nas funcionalidades do Fórum Livre.

O Fórum Livre desenvolvido a partir do modelo de argumentação ArFoLi então cria um encadeamento desde o tema escolhido, a questão aberta até o encerramento, permitindo ao participante identificar as respostas, os argumentos, as *tags* e a qualificação das respostas. Esse encadeamento é comparado ao ciclo da Gestão do Conhecimento, onde se captura a interação, representam-se as respostas, argumentos e qualificações, armazena-se a interação e seus artefatos, permitindo recuperar e compartilhar pela estruturação da discussão (NASCIMENTO, SANTORO e VELOSO, 2009). A reutilização do conhecimento se torna possível pela estruturação e pela facilitação ao acesso da discussão.

A implementação do modelo também foi avaliada em cenários de uso reais. No próximo capítulo, são descritos dois Estudos de Caso realizados com o uso da ferramenta Fórum Livre.

Capítulo 8 – Estudos de Caso

Este capítulo apresenta os Estudos de Caso realizados com dois grupos, um de alunos de Graduação da UNIRIO e outro na comunidade VivaOLinux. Esses estudos permitiram a avaliação da proposta de estruturação de interações através do uso da ferramenta de fórum desenvolvida, Fórum Livre. A avaliação foi realizada em duas etapas: uma com a participação dos grupos em discussões no fórum e outra por meio de um questionário respondido pelos participantes das discussões.

8.1. Estudo de caso

Estudo de Caso, de acordo com YIN (2005), “é uma investigação empírica que investiga um fenômeno contemporâneo dentro de seu contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos”.

YIN (2005) afirma ainda que para se definir o método de pesquisa a ser usado é preciso analisar as questões que são colocadas pela investigação. O Estudo de Caso é apropriado para responder às questões "como" e "por que" que são questões explicativas e tratam de relações operacionais que ocorrem ao longo do tempo mais do que frequências ou incidências (BREISSAN, 2000).

YIN (2005) apresenta quatro aplicações para o Método do Estudo de Caso:

1. Para explicar ligações causais nas intervenções na vida real que são muito complexas para serem abordadas pelos *Surveys* ou pelas estratégias experimentais;
2. Para descrever o contexto da vida real no qual a intervenção ocorreu;
3. Para fazer uma avaliação, ainda que de forma descritiva, da intervenção realizada;
4. Para explorar aquelas situações onde as intervenções avaliadas não possuam resultados claros e específicos.

De acordo com MARTINS e PINTO (2009) “um Estudo de Caso é uma pesquisa empírica que:

- 1 Investiga um fenômeno contemporâneo dentro de seu contexto real;
- 2 As fronteiras entre o fenômeno e o contexto não são claramente evidentes;
- 3 Múltiplas fontes de evidências são utilizadas.

Os Estudos de Caso desta dissertação têm como objetivo observar e avaliar se a estruturação apoiou os participantes durante a discussão e se foi possível encontrar o conhecimento gerado sem dificuldades por participantes externos à discussão, isto é, que não acompanharam a discussão. A escolha de realizar Estudos de Caso foi para tornar viável uma observação e análise sobre como os participantes se comportariam na prática utilizando uma ferramenta diferente da utilizada no cotidiano e se todas as funcionalidades seriam utilizadas, ou seja, responder, argumentar, criar novas questões,

pontuar a resposta de outro participante. Com isso, o Estudo de Caso possibilita refletir sobre a realidade de uma interação.

Os Estudos de Caso foram realizados em duas etapas. Na primeira etapa foram realizadas as interações nos dois grupos selecionados (graduandos da UNIRIO e participantes da comunidade VivaOLinux). A segunda etapa foi realizada por meio de um questionário enviado para alguns participantes das interações.

Foram realizados dois casos com dois grupos diferentes para analisar o comportamento de visões diferentes, uma de graduandos da UNIRIO e outra de pessoas (profissionais, estudantes) que convivem com o *Software* Livre. O primeiro estudo contou com a participação de um grupo de alunos de Graduação da UNIRIO engajados em *Software* Livre, e o segundo com participantes da comunidade VivaOLinux.

O questionário foi escolhido como ferramenta de coleta de dados complementares, pois, de acordo PARASURAMAN (1991), um questionário é um conjunto de questões, feito para gerar os dados necessários para se atingir os objetivos do projeto. Nesta dissertação, o questionário serviu como uma fonte coletora das impressões dos participantes sobre a ferramenta Fórum Livre em relação às funcionalidades e se a estruturação do conhecimento proposta foi realizada.

8.2. Descrição dos Estudos de Caso

8.2.1. Caso 1 – Grupo de alunos da UNIRIO envolvidos com *Software* Livre

O caso 1 aconteceu de 27 de julho de 2009 até 04 de agosto de 2009.

1 Planejamento:

O planejamento foi realizado com foco de disponibilizar a discussão e depois analisar a interação ocorrida e as avaliações dos alunos (moderador e participante).

- Foram convidados os alunos da Graduação do curso de Bacharelado em Sistemas de Informação da UNIRIO, engajados em *Software* Livre. Um aluno foi o moderador e avaliador da discussão. Um aluno participante também foi avaliador da discussão.
- Uma discussão foi criada no fórum, isto é, uma questão para ser debatida entre os alunos de Graduação.
- O fórum foi divulgado com a indicação da questão para os participantes.

- O fórum foi ativado para que todos interagissem sem interferências externas.
- O moderador realizou o encerramento da interação.
- Os avaliadores acessaram a discussão para responder o questionário. O questionário se encontra no Apêndice D – Questionário 2.

2 Execução:

A divulgação do fórum aconteceu pela lista de *e-mails* dos alunos de Graduação da UNIRIO e a ferramenta ficou ativa com a questão aberta pelo moderador por uma semana. Todos os participantes eram alunos de Graduação da UNIRIO e ao todo foram seis alunos que atuaram no Fórum Livre. Um dos alunos foi o moderador da discussão e os outros somente participantes. Todos os alunos que participaram são envolvidos com o tema *Software* Livre, seja estudando ou desenvolvendo ferramentas.

A discussão era monitorada diariamente e foi verificado que os alunos se cadastravam, liam a questão e só respondiam, em geral, dois dias depois. Cada participante inseriu uma resposta, surgiu uma nova questão e dois argumentos sobre uma determinada resposta. Essa resposta gerou debate e foi a mais pontuada com cinco pontos positivos. Assim acabou sendo a escolhida na hora do encerramento.

Um dos alunos abriu uma nova interação no fórum, com outro tema. Essa questão recebeu três respostas, houve uma nova questão em cima de uma das respostas e uma argumentação.

Os alunos também incluíram nas suas respostas *tags* tais como: netbook, Linux, Windows. Isso contribuiu para a discussão de forma a identificar sobre o que era a discussão, o que estava sendo tratado, porque depois a discussão ficou com sua nuvem de *tags* e proporcionou encontrar sobre o que estava sendo tratado naquela discussão com mais facilidade.

3 Análise:

A ferramenta Fórum Livre foi utilizada na sua totalidade, ou seja, os participantes responderam a questão, argumentaram e criaram novas questões, pontuaram as respostas de outros participantes e o encerramento trouxe a conclusão da interação. A interação refletiu uma discussão real entre os participantes, como é ilustrado na Figura 30.

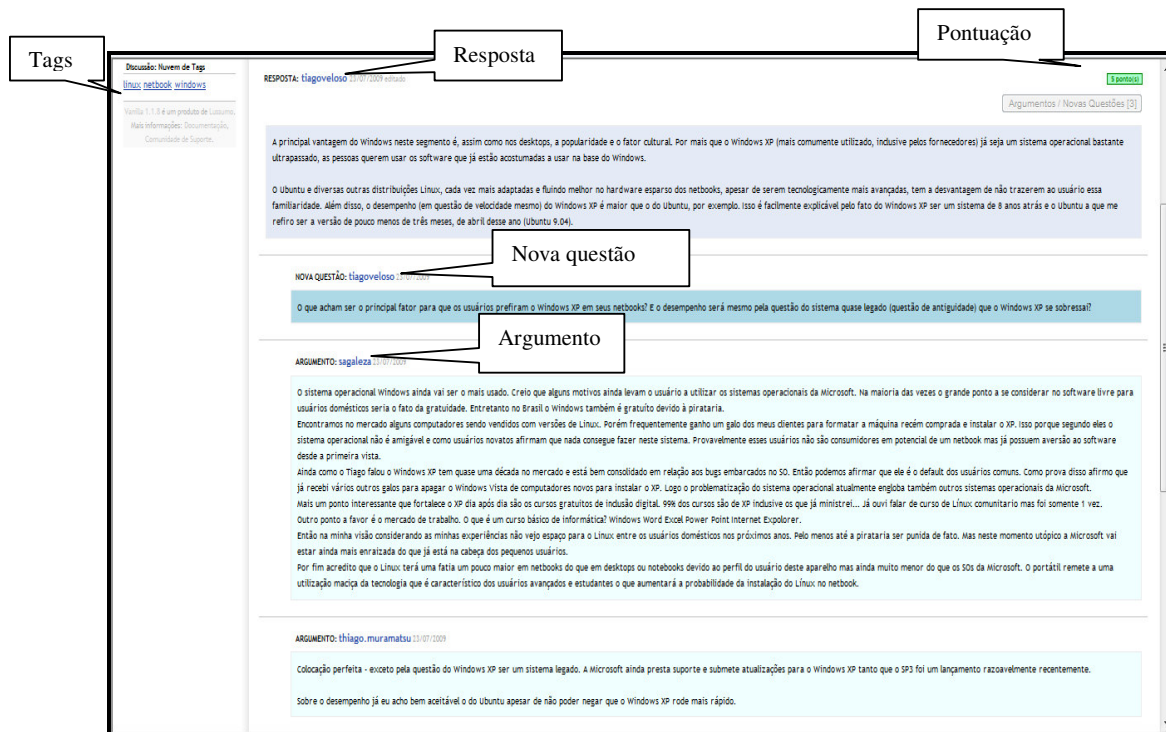


Figura 30 – Tela de discussão (resposta, nova questão e argumento)

Desta interação:

- A resposta mais pontuada e que gerou debate foi: “A principal vantagem do Windows neste segmento é, assim como nos desktops, a popularidade e o fator cultural. Por mais que o Windows XP (mais comumente utilizado, inclusive pelos fornecedores) já seja um sistema operacional bastante ultrapassado, as pessoas querem usar os softwares que já estão acostumadas a usar na base do Windows. O Ubuntu e diversas outras distribuições Linux, cada vez mais adaptadas e fluído melhor no hardware esparso dos *netbooks*, apesar de serem tecnologicamente mais avançadas, têm a desvantagem de não trazerem ao usuário essa familiaridade. Além disso, o desempenho (em questão de velocidade mesmo) do Windows XP é maior que o do Ubuntu, por exemplo. Isso é facilmente explicável pelo fato do Windows XP ser um sistema de 8 anos atrás e o Ubuntu a que me refiro ser a versão de pouco menos de três meses, de abril desse ano (Ubuntu 9.04).”
- A nova questão inserida nessa resposta: “O que acham ser o principal fator para que os usuários prefiram o Windows XP em seus *netbooks*? E

o desempenho será mesmo pela questão do sistema quase legado (questão de antiguidade) que o Windows XP se sobressai?”. Isso ressaltou que o participante continuou tendo a necessidade de solicitar informações nessa resposta, demonstrando que o problema ainda persistia.

- Um argumento da resposta: “O sistema operacional Windows ainda vai ser o mais usado. Creio que alguns motivos ainda levam o usuário a utilizar os sistemas operacionais da Microsoft. Na maioria das vezes o grande ponto a se considerar no software livre para usuários domésticos seria o fato da gratuidade. Entretanto no Brasil o Windows também é gratuito devido à pirataria. Encontramos no mercado alguns computadores sendo vendidos com versões de Linux. Porém frequentemente ganho um gallo dos meus clientes para formatar a máquina recém comprada e instalar o XP. Isso porque segundo eles o sistema operacional não é amigável e como usuários novatos afirmam que nada consegue fazer neste sistema. Provavelmente esses usuários não são consumidores em potencial de um *netbook* mas já possuem aversão ao software desde a primeira vista. Ainda como o Tiago falou o Windows XP tem quase uma década no mercado e está bem consolidado em relação aos bugs embarcados no SO. Então podemos afirmar que ele é o default dos usuários comuns. Como prova disso afirmo que já recebi vários outros galos para apagar o Windows Vista de computadores novos para instalar o XP. Logo a problematização do sistema operacional atualmente engloba também outros sistemas operacionais da Microsoft. Mais um ponto interessante que fortalece o XP dia após dia são os cursos gratuitos de inclusão digital. 99% dos cursos são de XP inclusive os que já ministrei... Já ouvi falar de curso de Linux comunitário mas foi somente 1 vez. Outro ponto a favor é o mercado de trabalho. O que é um curso básico de informática? Windows Word Excel Power Point Internet Explorer. Então na minha visão considerando as minhas experiências não vejo espaço para o Linux entre os usuários domésticos nos próximos anos. Pelo menos até a pirataria ser punida de fato. Mas neste momento utópico a Microsoft vai estar ainda

mais enraizada do que já está na cabeça dos pequenos usuários. Por fim acredito que o Linux terá uma fatia um pouco maior em *netbooks* do que em desktops ou notebooks devido ao perfil do usuário deste aparelho mas ainda muito menor do que os SOs da Microsoft. O portátil remete a uma utilização maciça da tecnologia que é característico dos usuários avançados e estudantes o que aumentará a probabilidade da instalação do Linux no *netbook*". Esse argumento demonstrou que o participante estava expondo seu ponto de vista e que explicitamente não concordava com o que a resposta indicava, ou seja, o conteúdo da mensagem fica mais claro para os participantes da discussão.

A discussão se estruturou conforme os participantes interagiam sem nenhum tipo de interferência externa.

O Fórum Livre pode também ter uma questão aberta por outro participante e gerar uma nova interação com um problema real daquele participante. As *tags* foram utilizadas, o que mostra que são uma forma de destacar o que interessa ao participante.

Os avaliadores da discussão foram o participante moderador e o participante que criou uma nova discussão. Os dois participantes não tiveram comunicação para preencher o questionário enviado.

Os avaliadores concordaram que o argumento ajudou a entender a opinião de outro participante sobre uma resposta, que o repositório de artefatos facilitou o acesso aos artefatos disponibilizados nas discussões, a recomendação de artefatos ajudou no envio para outro participante, que o encerramento facilitou muito para o entendimento do resultado final da discussão. Em relação à pontuação, eles concordaram parcialmente que como indicador de resposta mais adequada.

Uma divergência entre os participantes aconteceu, pois um deles indicou que a estruturação facilitou muito, enquanto o outro disse facilitou pouco. Outra divergência é que um concordou totalmente que a estruturação facilitou o acesso ao conhecimento gerado nas comunidades em relação ao cenário existente atualmente e o outro concordou parcialmente. As opiniões divergiram possivelmente devido à percepção de cada participante em relação a uma nova ferramenta de fórum e à sua experiência em utilizar essa ferramenta. Um fato importante é que nenhum dos participantes avaliadores disse que a estruturação não facilitou em nada. Desta forma, assume-se que os benefícios foram identificados, ainda que não plenamente.

Em relação a esse caso analisado, foi possível evidenciar que a estruturação facilitou a interação e a posterior busca, o argumento apoiou a discussão, o encerramento foi um grande facilitador para um posterior acesso. A pontuação apoia a discussão, mas não totalmente, pois os avaliadores concordam parcialmente. As *tags* foram inseridas pelos participantes, sem obrigatoriedade (poderia ou não ser inserida na resposta), evidenciaram que uma discussão fica enriquecida com palavras-chave.

Esse Estudo de Caso teve como objetivo verificar entre estudantes de Graduação da UNIRIO engajados em *Software* Livre se uma ferramenta estruturada facilitaria o fluxo de informações na discussão e o acesso ao conhecimento gerado, pois foi evidenciado no Estudo Exploratório que encontrar o conhecimento era um dos problemas existentes nas Comunidades Virtuais de *Software* Livre. Um problema que ocorreu nesse Estudo de Caso foi estimular os alunos a participarem, dado que estes se encontravam em um momento de recesso acadêmico. Mas mesmo assim, apesar da pequena quantidade de participantes e respostas, as interações foram focadas no tema discutido e geraram um conhecimento novo, e a estruturação mostrou-se potencialmente como um facilitador.

8.2.2. Caso 2 – Comunidade VivaOLinux

O caso 2 aconteceu de 10 de agosto até 24 de agosto de 2009.

1 Planejamento:

- Foi realizada uma reunião com o administrador da comunidade, para alinhar qual seria a proposta de discussão no fórum.
- Duas questões foram disponibilizadas no Fórum Livre.
- O link do Fórum Livre ficou disponível na comunidade VivaOLinux para avisar aos participantes sobre um novo fórum.
- O fórum ficou ativo por quinze dias.
- As discussões foram encerradas com as conclusões inseridas.
- Dois participantes das interações responderam ao questionário. O questionário se encontra no Apêndice D – Questionário 2.

2 Execução:

A ferramenta Fórum Livre ficou disponível com duas questões ativas em temas diferentes. A comunidade VivaOLinux divulgou também o fórum. Os participantes

pertenciam a comunidades de *Software* Livre. Os avaliadores selecionados foram dois participantes de comunidades de *Software* Livre e o administrador da comunidade VivaOLinux.

A questão 1 foi sobre qual o melhor navegador para desenvolvimento web. Essa questão pertenceu ao tema Navegadores Web e obteve quatro respostas, foi postada um nova questão e quatro argumentos em duas respostas diferentes. Em relação à pontuação, somente duas respostas obtiveram pontos (dois pontos e três pontos positivos). A Figura 31 ilustra estruturação de resposta, nova questão, resposta e argumento, e a pontuação atribuída.

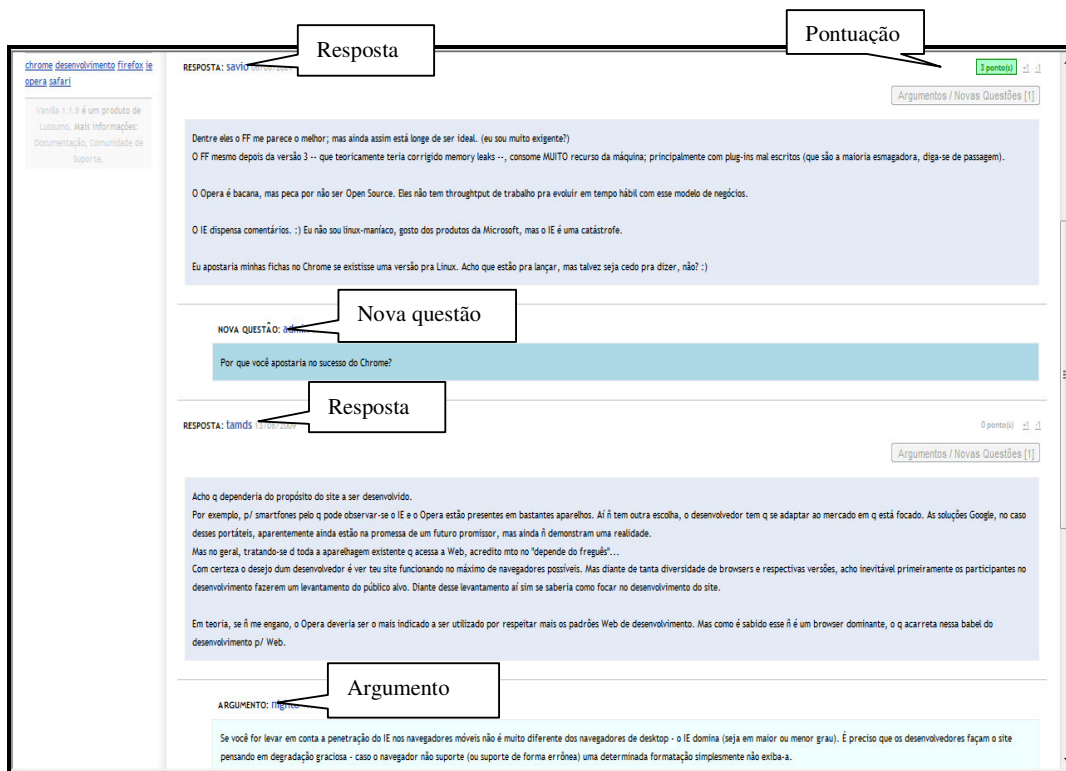


Figura 31 – Estruturação da questão sobre navegadores para desenvolvimento web.

A questão dois foi referente ao melhor ambiente de desktop para o Linux. Foram inseridas quatro respostas e quatro argumentos em duas respostas diferentes. Em relação à pontuação, somente duas respostas obtiveram pontos (um ponto e dois pontos).

3 Análise:

A avaliação dos dois participantes de comunidades apontou que:

- A estruturação do fórum (tema, resposta, argumento, pontuação, *tag* e encerramento) facilitou muito o entendimento da discussão;
- O argumento ajudou muito a entender a opinião de outro participante;
- A pontuação das respostas foi parcialmente um indicador de uma resposta adequada;
- O repositório de artefatos pode facilitar o acesso aos artefatos disponibilizados do fórum;
- A recomendação de artefatos que pode ser enviada por *e-mail* foi aceita parcialmente;
- O encerramento facilitou muito entendimento do resultado final da discussão;
- A estruturação facilitou o acesso ao conhecimento gerado nas comunidades em relação ao cenário existente atualmente nas comunidades.

A avaliação do administrador da comunidade VivaOLinux apontou que ele considera importante toda a estruturação do fórum proposta, ou seja, temas, questão, resposta, argumentos, pontuação, *tag* e encerramento. E que a pontuação foi um indicador de resposta adequada. Ele também avaliou como importante o repositório de artefatos e a recomendação por *e-mail* para outro participante. Em relação à estruturação, ele apontou que facilita o acesso ao conhecimento gerado nas comunidades em relação ao cenário existente atualmente nas comunidades. O administrador da VivaOLinux, não participou da discussão, ele foi um avaliador externo, como um participante que acessou posteriormente a discussão no fórum em busca de um conhecimento gerado.

A quantidade de mensagens trocadas foi pequena, o que acarretou uma discussão com poucas contribuições (do ponto de vista quantitativo), mas apesar disso, os elementos propostos no modelo de argumentação ArFoLi foram utilizados enriquecendo a discussão e permitindo aos avaliadores formarem uma visão total da proposta de estruturação.

No Apêndice E, as telas do Fórum Livre encontram-se mais detalhadas.

8.3. Conclusão geral dos Estudos de Caso

Os participantes dos dois casos utilizaram a ferramenta na discussão com o objetivo de expor seu ponto de vista sobre a questão debatida, utilizando as respostas, os argumentos, as *tags*, as novas questões para ajudá-los a se expressar e explicitar seu conhecimento e experiência e, o encerramento concluiu a interação. Desta forma, como esperado, as respostas ao questionário apontaram que a estruturação da interação é uma demanda para facilitar o acesso ao conhecimento.

Os elementos do modelo de argumentação ArFoLi apoiaram a estruturação da interação na ferramenta de Fórum Livre, pois fica evidenciada pela utilização das funcionalidades propostas no fórum e pelas respostas dos avaliadores, inclusive frente ao cenário existente atualmente nas comunidades.

O repositório de artefatos é uma funcionalidade interessante para os participantes acessarem os artefatos disponibilizados na interação e a recomendação também é um ponto de interesse, porém foram pouco explorados pelos participantes.

Com isso, as funcionalidades existentes na ferramenta de fórum, Fórum Livre, se mostram úteis na estruturação de uma interação. A pontuação das respostas se mostrou válida, no entanto, observou-se que provavelmente é preciso que seja mais explicada para que os participantes pontuem as respostas de forma mais expressiva e que esta pontuação possa ser levada em consideração na discussão e na identificação do valor técnico das respostas.

8.4. Limitações dos Estudos de Caso

Os dois Estudos de Caso realizados representam avaliações pontuais para a validação do modelo ArFoli e para verificar se a ferramenta de fórum desenvolvida, Fórum Livre, estava de acordo com o modelo proposto. A quantidade de participantes e de mensagens trocadas foi pequena em ambos os casos, e não foi possível realizar uma interação com foco no desenvolvimento de uma ferramenta em andamento.

Porém, as discussões mostram-se interessantes, pois os participantes utilizaram os elementos: *tag*, resposta, argumento e nova questão, isto é, esses elementos possibilitaram que uma pequena interação se tornasse estruturada e com informações gerando um conhecimento técnico profundo e de qualidade. Os participantes expuseram adequadamente suas experiências e pontos de vista em relação à questão debatida, quando a resposta não era suficiente uma nova questão surgia.

Assim, existem indícios de que os elementos do modelo de argumentação ArFoLi atenderam o propósito de organizar de uma discussão e facilitar o acesso ao conhecimento gerado. Isso demonstra que mesmo pequenas discussões são enriquecedoras e geradoras de conhecimento nas Comunidades Virtuais de *Software* Livre, visto a qualidade das respostas, argumentos e novas questões que surgiram.

Capítulo 9 – Conclusões e Trabalhos Futuros

Este capítulo apresenta as considerações finais da dissertação e os trabalhos futuros apontados pela pesquisa realizada nas Comunidades Virtuais de *Software* Livre para apoiar a Gestão do Conhecimento.

9.1. Pesquisa realizada

As Comunidades Virtuais de *Software* Livre são ricas em conhecimento e com a característica de gerar um produto final para o uso não só dessa comunidade como também da sociedade. O produto final é proposto a partir de um problema pessoal de um participante da comunidade e que todos colaboram e cooperam para a resolução desse problema. As Comunidades Virtuais de *Software* Livre são formadas por participantes dispersos geograficamente, com objetivos distintos para resolver seus problemas compartilhados, com as premissas de liberdade e apoiados na colaboração entre os participantes. A sistematização da Gestão do Conhecimento é uma demanda evidente nas comunidades, de acordo com o Estudo Exploratório realizado e com a revisão de literatura. Por intermédio do estudo realizado e observando critérios voltados para o ciclo da Gestão do Conhecimento (captura, armazenamento e recuperação de conhecimento), concluiu-se que, apesar de o conhecimento ser divulgado por meio de mensagens de *e-mail*, dicas ou notícias, a recuperação desse conhecimento ainda não é tão simples ou objetiva, já que o potencial do conteúdo das interações não é explorado.

A pesquisa realizada focou em analisar as interações e o conhecimento nelas existente para se propor modelos que apoiassem a Gestão do Conhecimento no contexto das Comunidades Virtuais de *Software* Livre. Para discutir com mais embasamento os problemas nas comunidades, realizou-se um Estudo Exploratório, onde se participou das comunidades para entender como elas funcionavam e, também os participantes foram questionados sobre como o conhecimento é tratado na comunidade. O Estudo Exploratório então pode mostrar uma visão interna e uma externa das comunidades e identificar os problemas tais como: a existência de uma grande quantidade material disponível, mas nem sempre é fácil encontrar um determinado assunto; a falta de organização nas interações que acontecem no fórum (mecanismo de suporte à interação mais utilizado). Além disso, observou-se que os participantes confiam uns nos outros mesmo sem muitas vezes se conhecerem pessoalmente.

Com os problemas identificados, a interação pode ser focada para resolver a falta de estruturação das interações realizadas nas comunidades e propor um modelo de argumentação que apoiasse as interações, então foi desenvolvido o modelo ArFoLi. A aplicação do modelo de argumentação foi realizada em um fórum estruturado,

customizando a partir da ferramenta livre de fórum Vanilla. Para avaliar o fórum estruturado, Fórum Livre, algumas interações foram realizadas no contexto de dois Estudos de Caso. Nesses estudos, os participantes, dos dois grupos, interagiram sem interferência e puderam explorar a ferramenta Fórum Livre e depois avaliaram a ferramenta. As participações e as avaliações dos participantes apontaram indícios de que a estruturação da interação é válida e realmente facilita o acesso ao conhecimento.

9.2. Considerações Finais

Os modelos de argumentação têm como objetivo estruturar as discussões, categorizando-as para encontrar as melhores soluções para problemas e gerar a conclusão. Visto que no contexto das Comunidades Virtuais de *Software* Livre encontramos os problemas sendo discutidos em fóruns desestruturados, o modelo proposto visa apoiar as discussões, organizando-a de forma que o participante identifique o que ocorreu durante uma discussão e facilitando o acesso ao conhecimento gerado mais tarde.

Os avaliadores dos Estudos de Caso evidenciam que a estruturação do conhecimento é necessária e principalmente o encerramento da discussão para que exista um resultado final do que as respostas e os argumentos geraram. Então os elementos propostos no modelo de argumentação ArFoLi se mostraram convenientes para as Comunidades Virtuais de *Software* Livre e que a ferramenta de fórum, o Fórum Livre conseguiu organizar as discussões propostas.

Os benefícios do modelo de argumentação ArFoLi para as comunidades são: a organização das discussões com os temas separados, as respostas e os argumentos proporcionando uma discussão encadeada, o acesso às discussões facilitado pelas *tags*, os artefatos separados por discussão. A qualificação da resposta é válida mas precisa ser mais explicitada para os participantes, sendo esse um ponto a ser melhorado e podendo mudar a visão do participante ao identificar como a resposta está qualificada. A *tag* também deve ser expandida para todo o tipo de mensagem, como questão, argumento, nova questão e encerramento.

9.3. Contribuições

A contribuição central desta pesquisa está na construção de um modelo que representa a forma como as Comunidades Virtuais de *Software* Livre se comportam e os elementos que devem estar presentes para que seja possível prover suporte à Gestão do Conhecimento. Além disso, destacamos as seguintes contribuições:

- Os resultados de um Estudo Exploratório que foi capaz de caracterizar estas comunidades do ponto de vista da produção e uso de conhecimento.
- O modelo conceitual, que provê uma representação formal da caracterização das comunidades do ponto de vista da interação entre seus membros.
- O modelo de argumentação ArFoLi que possibilitou uma proposta para estruturação de uma interação e a organização do conhecimento gerado por ela.
- A implementação da ferramenta Fórum Livre, que conseguiu aplicar os conceitos do modelo de argumentação ArFoLi e estruturou a interação de forma que as ideias da discussão fossem encadeadas e concluídas.
- Os resultados dos Estudos de Caso realizados em contextos reais de uso da ferramenta, que apresentaram indícios de que o modelo implementado através da ferramenta Fórum Livre estruturou a interação e o conhecimento gerado pode ficar organizado e disponível para os participantes.

9.4. Trabalhos Futuros

Os trabalhos futuros desta dissertação podem ser organizados em três grupos: relacionados ao modelo, relacionados à ferramenta e relacionados à obtenção de mais resultados.

Em relação ao modelo proposto, é preciso explorar mais a questão do conhecimento embutido nos produtos gerados, quer dizer, no código implementado pelos grupos de participantes das comunidades e a forma como eles interagem também através destes. Uma vez que existem comunidades focadas em desenvolver projetos e se utilizam das interações para desenvolver suas ferramentas. A estruturação de acordo

com o modelo de argumentação proposto facilitaria aos desenvolvedores discutirem a ferramenta de forma estruturada.

Outros conceitos importantes que não foram explorados são os de reputação e recomendação. A recomendação é uma funcionalidade que permite compartilhar e disseminar o conhecimento existente na interação e no artefato. O conceito de reputação permite que seja explicitada a qualificação do participante e a sua contribuição se tornem mais transparente durante o desenvolvimento de um produto ou serviço de *software*, ou seja, a reputação gerada pela sua participação nas interações realizadas na comunidade pode ser levada em consideração de forma clara. Esse conceito pode ser expandido no modelo conceitual, para que o grau de reputação que está como atributo da classe Participante se torne uma classe Reputação.

O modelo de argumentação também pode evoluir de forma a incorporar novos elementos. Para isso, são necessários mais estudos na literatura para tornar possível a evolução do modelo para discussões com foco em um consenso ou para contextos de aprendizado, de forma a possibilitar cada vez mais a estruturação da discussão e do conhecimento gerado nas discussões de grupo assíncronas e facilitar o seu posterior acesso.

Em relação à ferramenta de fórum, poderiam ser incorporadas funcionalidades para recomendação automática a partir das categorias de interesse do participante, ou seja, recomendar artefatos e discussões a partir das características dos participantes.

A ferramenta de fórum pode ser um serviço disponibilizado para ser executado pelas aplicações da plataforma de qualquer comunidade, isto é, a comunidade pode inserir a ferramenta na sua plataforma sem alterações e utilizando os princípios do modelo de argumentação ArFoLi. Isso é interessante, pois tornaria a ferramenta compatível e integrada com os diversos tipos de ambientes existentes nas comunidades. Os ambientes das comunidades são heterogêneos e desenvolvidos de acordo com o interesse do seu fundador.

Finalmente, em relação aos resultados obtidos, é importante realizar mais Estudos de Caso, focados em desenvolvimento de *software* utilizando recomendações automáticas de códigos e apontando para possíveis participantes qualificados com uma reputação para participarem desses projetos sendo potenciais contribuidores.

Referências Bibliográficas

- ABECKER, A., DECKER, S., 1999, “Organizational Memory: Knowledge Acquisition, Integration and Retrieval Issues: In: Proceedings of XPS-99. Springer LNAI 1570, pp.113-124.
- ALVES, A.M. e PESSÔA, M.S.P., 2008, “Gestão do Conhecimento e Melhoria de Processo de Software - Pesquisa-Ação”. In: Conferência Iadis Ibero-Americana WWW/INTERNET, Lisboa, Portugal, pp. 43-50.
- ANDRADE, L. C. V., SAMPAIO, F.F, ROCHA, L., 2002, “O Modelo GD-IBIS: Grupo de Discussão para Educação a Distância”. In: XXIX Seminário Integrado de Software e Hardware (SEMISH), Anais do XXII Congresso da Sociedade Brasileira de Computação, Florianópolis, Santa Catarina, Brasil, Julho de 2002, v. 3, pp. 183-292.
- BR-Linux. Disponível em: <<http://br-linux.org>>. Acessado em Junho de 2008.
- BRESSAN. F., 2000, “O Método do Estudo de Caso”. Disponível em: <http://www.fecap.br/adm_online/art11/flavio.htm>.
- CAMPBELL-KELLY, M., 2008, “Will the Future of Software be Open Source?”, In: Communications of the ACM, October 2008, v. 51, n. 10.
- CONKLIN, J. e BEGEMAN, M., 1988, “gIBIS: A hypertext tool for exploratory policy discussion”, ACM Transactions on Office Information Systems, v. 3, n. 3.
- BUNTINE W., ABERER K., PODNAR I., RAJMAN M., 2005, “Opportunities from open source search”. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 2–8.
- CAMPOS, A., 2006, “O que é software livre”, BR-Linux. Florianópolis, Março de 2006. Disponível em: <<http://br-linux.org/linux/faq-softwarelivre>>. Acessado em Junho de 2008.
- CMMI, 2008. Disponível em: < <http://www.sei.cmu.edu/cmmi/>>. Acessado em Dezembro de 2008.

- CONKLIN, E. J., 1992, “Capturing Organizational Memory”, In: Proceedings of GroupWare '92, D. Coleman (Ed.), Morgan Kaufmann, San Mateo, CA, pp.133-137.
- CONKLIN, J., 1998, “The Ibis Manual – Short Course in IBIS methodology”. Touchstone Whitepapers.
- DAVENPORT, T., PRUSAK, L., 1998, Working Knowledge: How Organizations Manage What They Know. Boston, USA, Harvard Business School Press.
- DESOUZA, K C., 2003, “Barriers to effective use of knowledge management systems in software engineering”. In: Communications of the ACM, v.46(1) pp. 99-101. Disponível em: <<http://dx.doi.org/doi:10.1145/602421.602458>>
- DEDRICK, J., WEST J., 2004, “An Exploratory Study into Open Source Platform Adoption”, In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, Waikoloa, Hawaí, v.8, pp. 80265b.
- DEMPSEY, B. J., WEISS, D., JONES, P., GREENBERG, J., 2002, “Who Is an Open Source Software Developer?”, In: Communication of the ACM, v. 45 n. 2, pp. 67-72.
- DRAKOS, N., 2007, “What We Can Learn About Collaboration From Open-Source Communities”, Gartner. Disponível em: <<http://www.gartner.com>>. Acessado em Junho 2008.
- DU PLESSIS M., 2008, “The strategic drivers and objectives of communities of practice as vehicles for knowledge management in small and medium enterprises”, International Journal of Information Management, v.28, pp. 61-67.
- EDEN, C., 1989, “Strategic options development and analysis (SODA)”. J. Rosenhead (Ed.), Rational Analysis in a Problematic World, Chichester: Wiley, Londres.
- ELLIOTT, M. S., SCACCHI, W., 2003, “Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration”, In: Proceedings of the ACM International Conference on Supporting Group Work, Sanibel Island, FL, November, pp. 21–30.
- FALBO, R.A. e NATALI, A.C.C., 2002, “Knowledge Management in Software Engineering Environments”, In: XVI Simpósio Brasileiro de Engenharia de Software, Gramado, RS, Brasil, v.1, pp.238-253.

- FELLER, J., FITZGERALD, B., 2000, “A framework analysis of the open source software development paradigm”, In: Proceedings of the International Conference on Information Systems (ICIS), Brisbane, Australia, pp. 58-69.
- FENG J., 2006, “A Knowledge Management Maturity Model and Application”. In: PICMET Proceedings, Istanbul, Turquia, v.3, pp 1251-1255.
- FOLTZ, P. W. DUMAIS, S. T., 1992, “Personalized Information Delivery: Na Analysis of Information Filtering Methods”, In: Communications of the ACM, New York, v.35, n.12 (Dec), pp. 51-60.
- FSF, 2008. Disponível em <<http://www.fsf.org>>. Acessado em Abril 2008.
- GE X., DONG, Y., HUANG, K., 2006, “Shared Knowledge Construction Process in an Open-Source Software Development Community: An Investigation of the Gallery Community”, In: Proceedings of the 7th international conference on Learning sciences, Bloomington, Indiana, Estados Unidos, pp. 189 – 195.
- GNU. Disponível em: <<http://www.gnu.org>>. Acessado em Junho de 2008.
- GOLDER, S. e BERNARDO A. H., 2005, “The Structure of Collaborative Tagging Systems.” HP Labs, Journal of Information Science. Disponível em: <<http://www.hpl.hp.com/research/idl/papers/tags/tags.pdf>>
- GOUVEA, M. T. A., 2005, *Um Modelo para Fidelização em Comunidades de Prática*. Dissertação de M.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.
- GOUVEA, M. T.A., MOTTA, C.L.R, SANTORO, F.M., 2006, “Scoring Mechanisms to Encourage Participation in Communities of Practice.”. In: The 10th International Conference on Conference on Computer Supported Coopertative Work in Design, Nanjing, China , v. 1, pp. 516-521
- HARS, A., OU, S., 2002, “Working for free? Motivations for participating in Open-Source projects. International Journal of Electronic Commerce, v. 6, n. 3, pp. 25–39.
- HENNINGER S., 1997, “Case-based knowledge management tools for software development”. Journal Automated Software Engineering, v. 4, pp. 319-340.
- HERLOCKER, J. L., 2000, *Understanding and Improving Automated Collaborative Filtering Systems*. Ph.D. dissertation, Computer Science Department, University of Minnesota, Minnesota, USA.

- HOUAISS da Língua Portuguesa. Consultado em Junho de 2008.
- HOU, J., SU, C., SU Y. e WANG W., 2008, “Integration of Knowledge Management and Collaborative Design”, In: International Conference on Intelligent Computation Technology and Automation, v. 2, pp. 743-747.
- IEEE, 1993, IEEE Standard Glossary of Software Engineering Terminology.
- KIMIECK, J. L., 2002, *Consolidação de comunidades de prática: um estudo de caso no PROINFO*. Tese de M.Sc., Centro Federal de Educação Tecnológica do Paraná, Curitiba, PR, Brasil.
- KING, W., MARKS P. e MCCOY S., 2002, “The Most Important Issues in Knowledge Management”, IN: Communications of the ACM, v. 45 n. 9, pp. 93–97.
- KOCH, S., 2004, “Free/Open Source Software Development”, Idea Group Inc., pp. 7–8.
- KOMI-SIRVIÖ S., MÄNTYNIEMI A. e SEPPÄNEN V., 2002, “Toward a Practical Solution for Capturing Knowledge for Software Projects”, IEE Software v. 19, n. 3.
- KOPONEN, T., HOTTI, V., 2005, “Open source software maintenance process framework”, In: Proceedings of the Fifth Workshop on Open Source Software Engineering. ACM Press, New York, pp. 1-5.
- KOURGANOFF, W. , 1990, A face oculta da universidade. Tradução Cláudia Schilling; Fátima Murad. São Paulo, Editora da Universidade Estadual paulista.
- KPMG Consulting, 1999, “Knowledge Management Assessment Exercise”. Disponível em: <http://cgi.nedecan.fi/kpmg/consulting/knowledge_managment>.
- KULKARNI, U. e FREEZE, R., 2004, “Development and Validation of a Knowledge Management Capability Assessment Model”, In: Proceedings of the 25th International Conference on Information Systems, Washington, pp. 657-670.
- LINSTONE H. e TUROFF M., 1975, “The Delphi Method, Techniques and Applications, Addison-Wesley, MA.
- MALEEWONG, K., ANUTARIYA, C. e WUWONGSE, V., 2008, "A Collective Intelligence Approach to Collaborative Knowledge Creation", In:

2008 Fourth International Conference on Semantics, Knowledge and Grid (SKG), Beijing: China, pp.64-70.

- MARTINS, G. e PINTO, R., 2009, Metodologia do Trabalho: Estudo de Caso. Disponível em: <http://www.eac.fea.usp.br/metodologia/estudo_caso.htm>.
- MICHAELIS on line, 2009. Disponível em: <<http://michaelis.uol.com.br/moderno/portugues/>> . Acessado em Abril. 2009.
- MOCKUS, A., FIELDING, R. T., HERBSLEB, J., 2002, “Two case studies of open source software development: Apache and Mozilla”. In: ACM Transactions on Software Engineering and Methodology, v. 11, n. 3, pp. 309-346.
- MOTTA, C. L. R, 1999, *Um ambiente de recomendação e filtragem cooperativas para apoio aequipes de trabalho*. Tese de D.Sc., Programa de Pós-graduação de Engenharia, Universidade Federal do Rio de Janeiro, RJ, Brasil.
- NAKAGAWA, E. Y., 2004, “An Investigation of the Open Source Development Process”, In: Actas de las IV Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, IIISIC'2004, Madrid, Espanha.
- NASCIMENTO, L.C. e SANTORO, F.M., 2009, “Análise de interações nas Comunidades Virtuais de Software Livre”, In: Simpósio Brasileiro de Sistemas de Informação, Goiás, Brasília, Brasil, v.1., pp.12-23.
- NASCIMENTO, L.C., SANTORO, F.M. e VELOSO, T.N., 2009, “Um Modelo de Argumentação para apoiar as Comunidades Virtuais de Software Livre”, In: Simpósio Brasileiro de Sistemas de Colaborativos, Fortaleza, Ceará, Brasil, v.1., pp.1.
- NEVES, A., 2001, “Community of Practice”, entrevista com Etienne Wenger. Disponível em: <http://www.kmol.online.pt/pessoas/WengerE/entrev_e.html>. Acessado em Setembro de 2008.
- NONAKA, I., TAKEUCHI, H., 1997, Criação do Conhecimento na Empresa: como as empresas japonesas geram a dinâmica da inovação. 18^a ed., Editora Campus.
- NONAKA, I., TOYAMA, R., HIRATA, T., 2008, Managing flow: a process theory of the knowledge-based firm, Palgrave Macmillan, New York.
- NOVAK, J., GOWIN, B., 1984, “Learning how to learn”. Cambridge University Press, UK.

- O'REILLY, T., 1999, "Lessons from open source development", In: Communications of the ACM, v. 42 n. 4, pp. 33–37.
- O'REILLY, T., 2005, "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software". Disponível em: < <http://oreilly.com/web2/archive/what-is-web-20.html>>. Acessado em Maio de 2009.
- OSI. Disponível em: < <http://www.opensource.org>>. Acessado em Agosto de 2008.
- PARASURAMAN, A., 1991, Marketing research. 2^a. ed., Addison Wesley Publishing Company.
- PAULZEN, O., e PERC, P., 2002, "A Maturity Model for Quality Improvement in Knowledge Management", In: Proceedings of the 13th Australasian Conference on Information Systems (ACIS 2002), Melbourne, Australásia, pp. 243-253.
- PFANFESELLER, M., PFANFESELLER, M., KROTH, E., 2001, "Uma ferramenta de apoio ao desenvolvimento de software baseado em componentes", In: XV Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas, Rio de Janeiro, RJ, Brasil, pp. 88-98.
- PLONE. Disponível em < <http://plone.org>>. Acessado em Janeiro de 2009.
- POLANCIC, G., HORVAT R. V., ROZMAN T., 2004, "Comparative assessment of open source software using easy accessible data". In: Proceedings of the 26th International Conference on Information Technology Interfaces, v. 1, pp. 673–678, Slovenia.
- RAYMOND, E.S., 1998, "A *Catedral e o Bazar*", Disponível em: <<http://www.geocities.com/CollegePark/Union/3590/pt-cathedral-bazaar.html>>. Acessado em Junho de 2008.
- RAYMOND, E.S., 2004, "Open Minds, Open Source". Disponível em: <http://www.catb.org/~esr/writings/analog.html>. Acessado em Maio de 2008.
- REIS C.R., 2003, *Caracterização de um Processo de Software para Projetos de Software Livre*. Dissertação de M.Sc., USP, São Carlos, SP, Brasil.
- REIS, C., FORTES, R.P.M., 2002, "An Overview of the Software Process and Tools in the Mozilla Project", In: Proceedings of the Open Source Software Development Workshop. Newcastle, pp. 155–175.

- RELVAS, S. e ANTUNES, P., 2006, “Análise da Aplicabilidade dos Modelos de Argumentação na Resolução Alternativa de Conflitos”, In: 1ª Conferência Ibérica de Sistemas e Tecnologias de Informação, Portugal.
- RESNICK, P. e VARIAN, H. R., 1997, “Recommender Systems”, In: Communications of the ACM, v. 40, nº 3 (Marc), pp. 56-58.
- RESNICK, P., ZECKHAUSER, R., FRIEDMAN, E., KUWABARA, K., 2000, “Reputation Systems”. In: Communications of the ACM, v. 43(12), p. 45-48.
- RITTEL, H. e KUNZ, W., 1970, “Issues as Elements of Information Systems”. Institute of Urban and Regional Development, University of California, Berkeley, California, Working Paper 131.
- RITTEL, H. e WEBBER, M., 1973, “Dilemmas in a general theory of planning.” Policy Sciences, v.4, n. 2, p.155-169.
- RUDZKI, M.P. and JONSON F., 2003, *Identifying and Analyzing Knowledge Management Aspects of Practices in Open Source Software Development*. Tese de M.Sc., School of Engineering at Blekinge Institute of Technology, Ronneby, Suécia.
- SALTON, G., 1983, Introduction to Modern Information Retrieval. MCGRAW-HILL.
- SCACCHI, W., 2002, “Understanding the Requirements for Developing Open Source Software Systems”, In: IEE Proceedings Software v. 149, pp. 24-39.
- SCACCHI, W., FELLER, J., FITZGERALD, B., HISSAM, S., LAKHANI, K., 2006, “Understanding Free/Open Source Software Development Processes”, In: Software Process- Improvement and Practice, v. 11 n. 2, pp. 95-105.
- SCACCHI W., 2007, “Free/Open Source Software Development: Recent Research Results and Emerging Opportunities”. In: Proceedings European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, Dubrovnik, Croatia, pp. 459-468.
- SILVA, B.C.C., FALBO, R.A., 2006, “Definição de um Processo Padrão para Software Livre”, In: V Simpósio Brasileiro de Qualidade de Software – SBQS’2006, Vila Velha, ES, pp.159-173.
- SOFTEX, 2005, “O impacto do software livre e de código aberto na indústria de software do Brasil”, Softex Campinas. Disponível em <<http://www.softex.br>>.

- SOMMERVILLE, I., 1995, *Software engineering*. 5ª. ed., Editora Addison-Wesley, pp. 7.
- SOWE, S.K., STAMELOS I., ANGELIS L., 2007, “Understanding knowledge sharing activities in free/open source software projects: An empirical study”, *The Journal of Systems and Software*, v. 81, n. 3, pp. 431 -446.
- TAURION, C., 2004, *Software Livre: potencialidades e modelos de negócios*. 1ª. ed, Editora Brasport.
- TERRA, J.C.C., 2000, *Gestão do Conhecimento: O Grande Desafio Empresarial*. Rio de Janeiro, Editora Negócio.
- TERRA, J.C.C., 2001, “Fortalecendo Cadeias Produtivas através de Portais do Conhecimento”. Disponível em: <<http://www.terraforum.com.br>>. Acessado em Julho de 2009.
- TERRA, J.C.C., 2005, *Gestão do Conhecimento*. Disponível em: <<http://www.terraforum.com.br>>. Acessado em Dezembro de 2007.
- TOULMIN, S., 1958, *The Uses of Argument*. Cambridge: Cambridge University Press.
- TOULMIN, S., 2009, “The Toulmin Model of Argumentation”. Disponível em : < http://www-rohan.sdsu.edu/~digger/305/toulmin_model.htm>. Acessado em Abril. 2009.
- TOUCHSTONE Tools, 2000, *The Ibis Manual – Short Course in Touchstone tools and resources*.
- VIVAOLINUX (2009). Disponível em < <http://www.vivaolinux.com.br>>. Acessado em Maio de 2009.
- VIXIE, P., 1995, *Software Engineering em Open Sources: Voices from the revolution*. 1ª. ed., Editora.O’Reilly, pp. 91-100.
- VON KROGH, G., SPAETH, S., HAEFLIGER, S., 2005, “Knowledge reuse in open source software: an exploratory study of 15 open source projects”, In: *Proceedings of the 38th Hawaii International Conference on System Sciences*, (IEEE, HICSS ’05-Track 7), January 03–06, Big Island, Hawaii, p. 198b.
- WANG, C., CHEN Y. e CHEN Y., 2002, “A Distributed Knowledge Model for Collaborative Engineering Knowledge Management in Allied Concurrent Engineering”. In: *IEEE International Engineering Management Conference*, v. 2, pp. 701- 707.

- WIELAND, T., 2004, “Open Source As a Knowledge Management Instrument”. In: 2nd Workshop on Open Source Software in an Industrial Environment, Ulm, Alemanha
- WENGER, E., McDermott, R., Snyder, W. M., 2002, “Cultivating Communities of Practice: a guide to managing knowledge”. Harvard Business Scholl Press.
- WENGER, E ,McDermott, R.; Snyder, W. M., 2002, “Cultivating Communities of Practice: a guide to managing knowledge”. Harvard Business Scholl Press.
- WENGER, E., 2003, “Cultivando comunidades de prática – Liderança e Valor”. Workshop realizado na Petrobrás, Rio de Janeiro, em outubro de 2003.
- WENGER, E., 2004, “Knowledge management is a donut: shaping your knowledge strategy with communities of practice”, Ivey Business Journal.
- WENGER, E., 2006, Communities of practice: a brief introduction. Disponível em: <<http://www.ewenger.com/theory/index.htm>>. Acessado em Setembro de 2008.
- WU, M.-W. e LIN, Y.-D., 2001, “Open source software development: An overview”. In: IEEE Computer, v. 34 n. 6, pp. 33–38.
- YAMAUCHI, Y., YOKOZAWA, M., SHINOHARA, T., e ISHIDA, T., 2000, “Collaboration with Lean Media: how open-source software succeeds”, In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (Philadelphia, Pennsylvania, United States). CSCW '00. ACM Press, New York, NY, 329-338. DOI= <http://doi.acm.org/10.1145/358916.359004>
- YANG J. e WANG J., 2008, “Review on Free and Open Source Software”, In: IEEE International Conference on Service Operations and Logistics, and Informatics, Beijing, China, v. 1, pp. 1044-1049.
- YE Y. e KISHIDA K., 2003, “Toward an understanding of the motivation of open source software developers”. In: 25ª. International Conference on Software Engineering, Portland, Oregon. pp. 419-429.
- YIN, R. K., 2005, Estudo de caso: planejamento e métodos. Tradução: Daniel Grassi. 3ª. ed., Editora Brookman, Porto Alegre.

Apêndice A – Questionário 1

Survey realizado com os participantes de Comunidades *Virtuais de Software Livre*.

Colméia é um projeto de parceria entre o PESC-COPPE/UFRJ e o PPGI-CCET/UNIRIO e o “desafio deste projeto de cooperação está em buscar caminhos para ampliar as vantagens e a cultura de trabalho preconizado pelos processos de desenvolvimento de software livre através da compreensão de sua colaboração intrínseca, da proposta de ferramental de apoio à sua realização eficaz”.

Esta pesquisa tem como objetivo identificar as necessidades das Comunidades Virtuais de Software Livre em acessar o conhecimento gerado durante as interações ocorridas entre seus membros, ou disponibilizada por meio de documentos nos repositórios.

E onde pretendemos chegar com esta pesquisa?

O objetivo específico da pesquisa é propor um modelo de Gestão do Conhecimento para as Comunidades Virtuais de Software Livre, apoiando e facilitando o acesso ao conhecimento.

Por que acreditamos nesta pesquisa?

A Gestão do Conhecimento visa capturar, representar, armazenar, recuperar e compartilhar o conhecimento. Identifica-se nas Comunidades Virtuais de Software Livre muito conhecimento a ser compartilhado com foco em tecnologia livre, filosofia e desenvolvimento de ferramentas de software livre / open source. Muitas pesquisas são desenvolvidas visando a disseminar os projetos de software livre e o conhecimento nessas comunidades virtuais.

Interessa-me participar?

Claro! Afinal, você diretamente e sua comunidade serão potencialmente beneficiados com os resultados apresentados dessa pesquisa.

Quero participar! Como faço?

A melhor forma de participar agora é respondendo ao questionário a seguir.

Suas informações e opiniões são fundamentais.

Desde já, obrigada!

ACESSE O QUESTIONÁRIO!

Questões:

1. Nome

Este campo refere-se aos seus dados pessoais e é de preenchimento opcional.

2. E-mail

Este campo refere-se aos seus dados pessoais e é de preenchimento opcional.

3. Cidade/Estado onde mora

Este campo refere-se aos seus dados pessoais e é de preenchimento opcional.

4. Formação

Este campo refere-se aos seus dados pessoais, e é de preenchimento opcional. Cursos técnicos, Graduação, Pós-Graduação.

Curso Técnico.

Graduação.

Pós-graduação.

Mestrado.

Doutorado.

5. Qual a comunidade virtual de *Software* Livre / Open Source que você faz parte?

Este campo refere-se aos seus dados pessoais e é de preenchimento opcional.

6. Qual seu papel na comunidade?

Administrador.

- Moderador.
- Coordenador.
- Desenvolvedor.
- Participante Informal.

7. Como você interage com os outros participantes na comunidade?

- Fórum.
- Listas de discussão.
- Chat.
- Blog.
- Outra ferramenta Qual? _____

8. Que tipo de material os participantes de sua comunidade costumam compartilhar:

- Artigos.
- Notícias.
- Manuais.
- Código fonte.
- Comentários sobre determinada tecnologia.
- Outro material. Qual? _____

9. Como você identificaria a quantidade de material compartilhada pela sua comunidade:

- Muita quantidade de material é compartilhada.
- Média quantidade de material é compartilhada.
- Pouca quantidade de material é compartilhada.
- Nenhum material é compartilhado.

10. Onde ficam disponíveis os materiais da comunidade:

- Pastas de arquivos.
- Mensagens.
- Fórum de discussões.

- Links.
- Blog.
- Outra forma. Qual? _____

11. Quando você tem alguma dúvida ou deseja alguma informação:

- Manda uma mensagem (através da lista ou fórum).
- Procura alguém na comunidade que já conhece.
- Procura material sobre o assunto no repositório.
- Faz uma busca sobre o assunto usando ferramentas disponíveis na comunidade.
- Faz outras ações. Quais? _____

12. Que tipo de mensagens você costuma trocar quando interage com outros participantes da comunidade:

- Resolução de problemas.
- Envio de uma dúvida e suas explicações pelos participantes da comunidade.
- Formas de utilização de uma tecnologia livre no seu dia a dia.
- Divulgação de informações sobre a tecnologia livre.

13. Na sua opinião, as mensagens trocadas na sua comunidade respondem às questões debatidas?

- Respondem totalmente.
- Respondem parcialmente.
- Nunca respondem por completo.

14. Quando você procura por um determinado assunto na comunidade em alguma das modalidades, considera:

- Sempre fácil e rápido encontrar.
- Nem sempre é fácil e rápido encontrar.
- Nunca encontra.

15. É possível indicar um material da comunidade para outro participante acessar?

Sim.

Não.

16. É possível confiar nas informações disponibilizadas por um participante da comunidade?

Sim, de maneira geral todas as informações são úteis para mim, independente de quem a disponibilizou.

Sim, quando já conheço este participante da comunidade.

Sim, mas costumo verificar a procedência das informações.

Não confio nas informações.

Apêndice B – Requisitos Funcionais do Fórum Livre

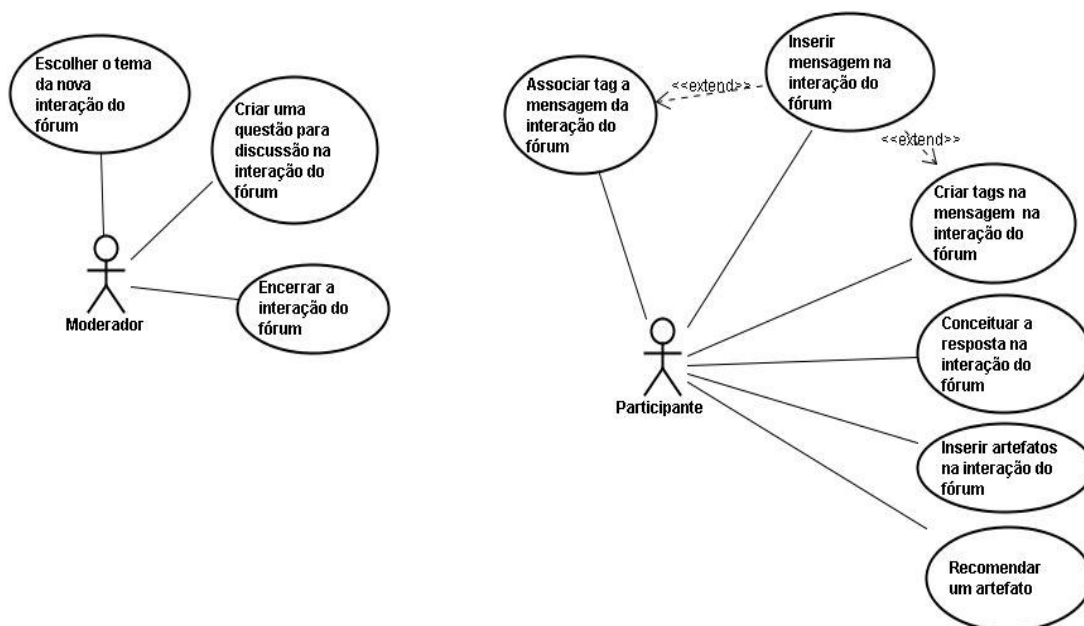
Os Requisitos Funcionais do Fórum Livre para estruturar a interação. Eles foram os seguintes:

- 1 O sistema permite somente que os participantes da comunidade (cadastrados) criem questões.
- 2 O sistema permite a criação de uma nova interação do fórum.
- 3 O sistema permite o encerramento da interação do fórum pelo participante que o criou.
- 4 O sistema exibe o status do fórum de fechado ao ser encerrado.
- 5 O sistema exibe a data da última resposta inserida e a quantidade de respostas inseridas.
- 6 O sistema exibe o moderador da interação do fórum, junto com a descrição da questão.
- 7 O sistema exibe o fórum graficamente, encadeando: tema, questão, resposta, argumento, qualificação e encerramento.
- 8 O sistema solicita o tema (categoria) da nova interação do fórum.
- 9 O sistema solicita a questão a ser discutida na interação.
- 10 O sistema solicita as *tags* da resposta do participante.
- 11 O sistema cria a tabela de artefatos no banco de dados da comunidade para da nova interação do fórum.
- 12 O sistema identifica o participante, a data e a hora da criação da interação no fórum.
- 13 O sistema controla as respostas dos participantes, identificando-as como respostas ou argumentos ou novas questões.
- 14 O sistema identifica que é uma resposta, quando é relativo à questão do fórum.

- 15 O sistema identifica que é um argumento, quando é relativo a uma resposta do fórum.
- 16 O sistema identifica que é uma nova questão argumento, quando é relativo a uma resposta do fórum.
- 17 O sistema coloca o argumento ligado à resposta.
- 18 O sistema insere na tabela de artefatos do banco de dados, os disponibilizados no fórum. Os artefatos são ordenados por interação de fórum e por categoria.
- 19 O sistema permite que um artefato seja recomendado pelo participante para outro participante, facilitando e divulgando os artefatos, visando à reutilização.
- 20 O sistema, quando o fórum é encerrado, permite ao participante responsável inserir uma mensagem de encerramento.
- 21 O sistema identifica os participantes que integraram as interações do fórum.
- 22 O sistema busca os artefatos das interações do fórum.
- 23 O sistema busca um participante do fórum.

Apêndice C – Casos de Uso

Os Casos de Uso descritos a seguir foram utilizados para analisar o comportamento da ferramenta desenvolvida e a atuação dos atores na mesma.



Escolher o tema da nova interação do fórum (CSU01)

- Ator: Moderador
- Fluxo Principal:
 1. O participante responsável solicita a escolha do tema da nova interação no fórum das Comunidades Virtuais de *Software Livre*.
 2. O sistema exibe a lista com as categorias existentes.
 3. O moderador escolhe a categoria desejada.
 4. O sistema registra a categoria escolhida para a nova interação no fórum.

Fluxo Alternativo:

1. O moderador escolhe o tema Outros, quando o tema desejado não existe.

Criar uma questão para discussão na interação do fórum (CSU02)

- Ator: Moderador
- Fluxo Principal:
 1. O sistema exibe o campo para a inclusão da questão.
 2. O moderador insere a questão na interação.
 3. O sistema registra a questão.
 4. O sistema exibe a questão, o nome do moderador, a data e a hora da criação.

Inserir mensagem na interação do fórum (CSU03)

- Ator: Participante
- Fluxo Principal:
 1. O participante insere a mensagem na interação.
 2. O sistema registra a mensagem.
 3. O sistema exibe a mensagem, o nome do participante, a data e a hora da criação.
 4. O sistema exibe uma mensagem ao participante informando que em até 2 (duas) horas é possível alterar a mensagem e inserir *tags*.
- Fluxo Alternativo:
 1. O sistema identifica se a mensagem da interação do fórum é uma resposta ou um argumento ou uma nova questão. A mensagem é uma resposta quando está ligada à questão, e um argumento ou nova questão quando ligado à resposta.

Criar *tags* na resposta na interação do fórum (CSU04)

- Ator: Participante
- Fluxo Principal:
 1. O participante insere a(s) *tag(s)* desejada na resposta.
 2. O sistema identifica se o tempo limite para a inserção da *tag* é válido.
 3. O sistema registra a(s) *tag(s)* junto com a mensagem.

Fluxo Alternativo:

1. O sistema verifica se a(s) *tag(s)* é repetida.

2. Se a(s) *tag(s)* for repetida o sistema envia a mensagem avisando ao participante.

Associar *tag* a resposta da interação do fórum (CSU05)

- Ator: Participante
- Fluxo Principal:
 1. O participante insere a(s) *tag(s)* desejada na resposta.
 2. O sistema identifica se o tempo limite para a inserção da *tag* é válido.
 3. O sistema registra a(s) *tag(s)* junto com a mensagem.

Fluxo Alternativo:

1. O sistema verifica se a(s) *tag(s)* é repetida.
2. Se a(s) *tag(s)* for repetida o sistema envia a mensagem avisando ao participante.

Pontuar a resposta na interação do fórum (CSU06)

- Ator: Participante
- Fluxo Principal:
 1. O sistema solicita um ponto a mais um ou menos na resposta da interação do fórum.
 2. O sistema registra a pontuação.

Inserir artefatos na interação do fórum (CSU07)

- Ator: Participante
- Fluxo Principal:
 1. O participante solicita a inserção de artefatos na interação do fórum.
 2. O sistema insere nos artefatos na tabela de artefatos.
 3. O sistema ordena os artefatos por interação no fórum e por categoria

Fluxo Alternativo:

1. O sistema permite inserir os seguintes artefatos: código fonte, artigo, dica, notícia, manual.

Recomendar um artefato (CSU08)

- Ator: Participante

- Fluxo Principal:
 1. O participante solicita a recomendação de um artefato para outro participante.
 2. O sistema solicita a identificação ou *e-mail* do participante que vai receber o artefato.
 3. O participante insere os dados do participante que vai receber o artefato.
 4. O sistema envia o artefato.

Encerrar a interação do fórum (CSU09)

- Ator: Moderador
- Fluxo Principal:
 1. O moderador solicita o encerramento da interação do fórum.
 2. O sistema solicita a inserção de uma mensagem, onde indica a(s) melhor(es) resposta(s).
 3. O sistema exibe a mensagem de encerramento.
 4. O sistema coloca o status da interação como fechado, não permitindo inserir mais mensagens.

Apêndice D – Questionário 2

Survey utilizado para coletar impressões de participantes da interação realizada no Fórum Livre.

As Comunidades Virtuais de Software Livre são ricas em conhecimento criado e disseminado coletivamente pelas interações que acontecem entre seus participantes, principalmente nos fóruns.

O presente questionário visa, a partir da análise das discussões realizadas no Fórum Livre, identificar se a estruturação ficou adequada para apoiar a Gestão do Conhecimento, ou seja, capturou a discussão, representou de forma estruturada, armazenou as respostas/argumentos e artefatos, recuperou sem dificuldades para compartilhar e foi possível reutilizar o conhecimento gerado na discussão.

1. Quanto à estruturação do fórum (temas, respostas, argumentos, pontuação e encerramento) facilitou no entendimento da discussão?
 - Facilitou muito o entendimento e esclareceu a discussão.
 - Facilitou pouco e não esclareceu a discussão.
 - Não fez diferença.

2. As contribuições do tipo ARGUMENTO, inseridas junto às respostas, ajudaram a entender a opinião de outro participante sobre uma resposta?
 - Muito.
 - Pouco.
 - Nada.

3. A pontuação das respostas foi um indicador de uma resposta adequada?

- Concordo Totalmente.
 - Concordo Parcialmente.
 - Não Concordo.
4. O repositório de artefatos facilitou o acesso aos artefatos disponibilizados nas discussões do fórum?
- Facilitou Muito.
 - Facilitou Pouco.
 - Não Facilitou.
5. A recomendação de artefatos inseridos durante a discussão ajudou no envio para outro participante?
- Concordo Totalmente.
 - Concordo Parcialmente.
 - Não Concordo.
6. O encerramento explícito da discussão no fórum, com apresentação de uma melhor resposta para a questão levantada, facilita o entendimento do resultado final da discussão?
- Facilita Muito.
 - Facilita Pouco.
 - Não Facilita.
7. O cenário de hoje nas comunidades não contém os elementos do fórum estruturado proposto. Você concorda que essa estruturação facilita o acesso ao conhecimento gerado nas comunidades (busca pelas *tags*, categorização)?
- Concordo Totalmente.
 - Concordo Parcialmente.
 - Não Concordo.

Apêndice E – Telas do Fórum Livre

Tela de uma questão:



Figura 32 – Tela do Fórum Livre de uma questão

Telas de uma discussão com resposta, argumento, nova questão e pontuação:

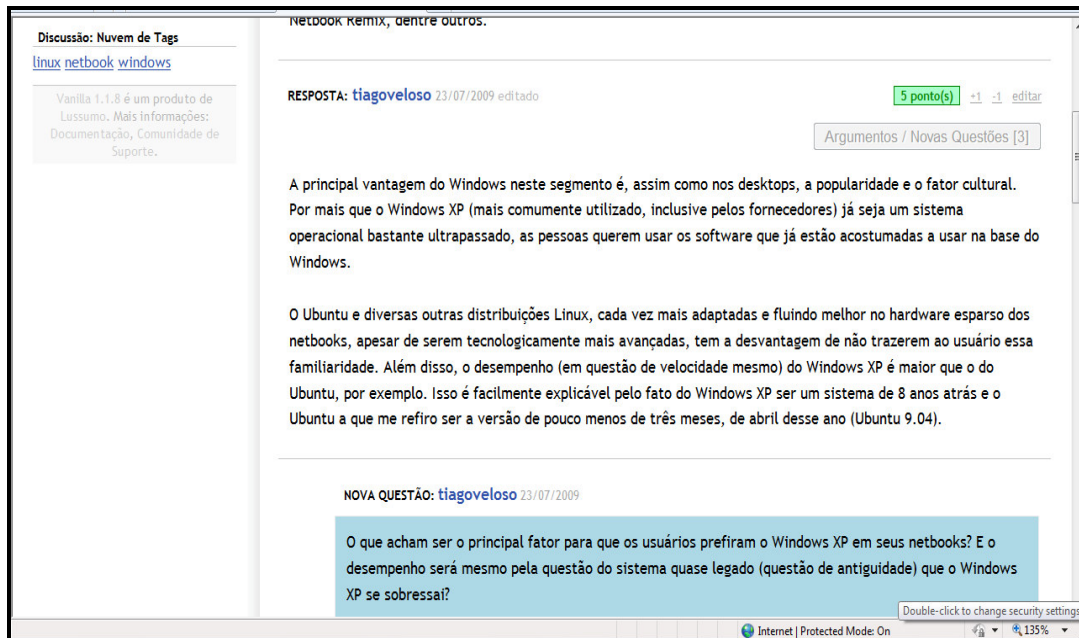


Figura 33 – Tela do Fórum Livre com resposta, nova questão e pontuação

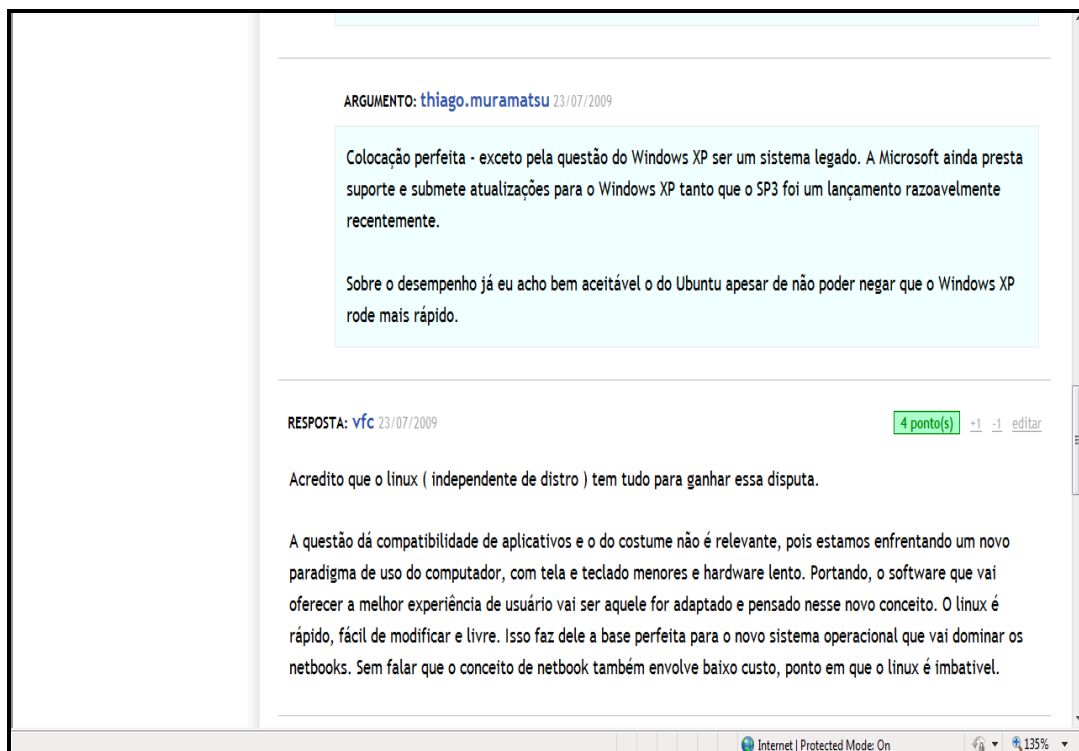


Figura 34 – Tela do Fórum Livre com argumento, resposta e pontuação

Tela com o encerramento da discussão:

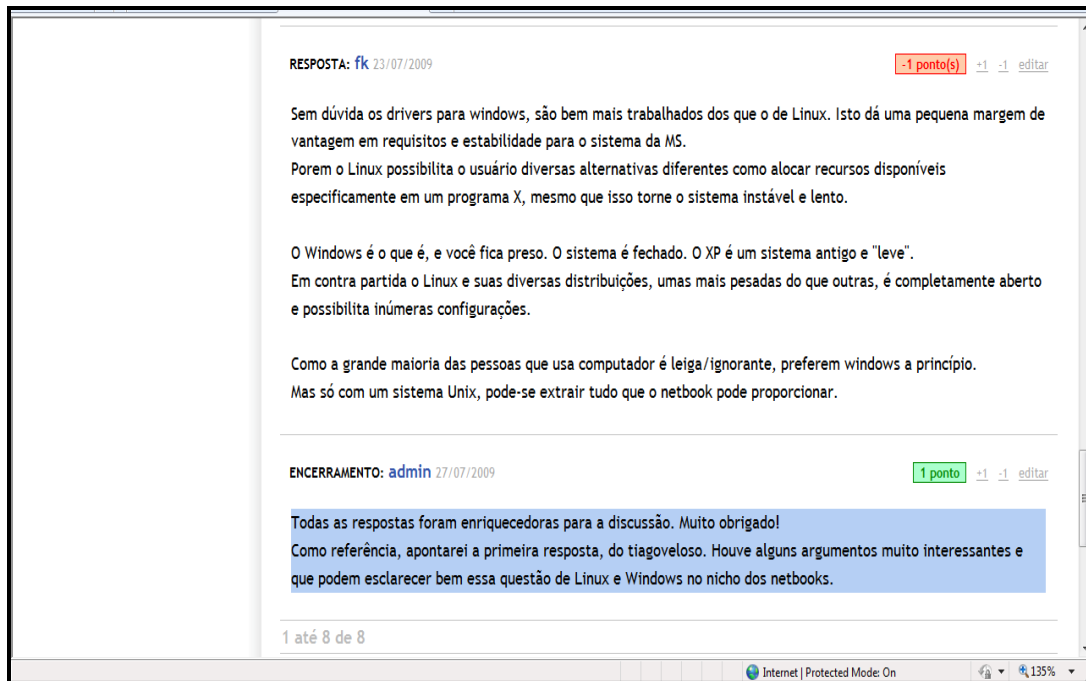


Figura 35- Tela do Fórum Livre com resposta, pontuação e encerramento