



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
BACHARELADO SISTEMAS DE INFORMAÇÃO

**BRUNO LEANDRO GOMES DE CASTRO**

**UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES  
ABERTOS UTILIZANDO O FRAMEWORK INSTITUTIONAL ENVIRONMENTS  
BASEADO NA METODOLOGIA GAIA**

**QUIXADÁ  
2014**



**BRUNO LEANDRO GOMES DE CASTRO**

**UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES  
ABERTOS UTILIZANDO O FRAMEWORK INSTITUTIONAL ENVIRONMENTS  
BASEADO NA METODOLOGIA GAIA**

Trabalho de Conclusão de Curso submetido à  
Coordenação do Curso Bacharelado em Sistemas de  
Informação da Universidade Federal do Ceará como  
requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Orientador Prof<sup>o</sup> Dr. Marcos Antonio de Oliveira

**QUIXADÁ  
2014**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

- 
- C35p Castro, Bruno Leandro Gomes de  
Um processo para desenvolvimento de sistemas multiagentes abertos utilizando o Framework Institutional Environments baseado na metodologia Gaia / Bruno Leandro Gomes de Castro. – 2014.  
179 p. : il. color., enc. ; 30 cm.
- Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2014.  
Orientação: Prof. Dr. Marcos Antonio de Oliveira  
Área de concentração: Computação

1. Inteligência Artificial Distribuída 2. Sistemas Multiagentes 3. Metodologia Gaia I. Título.

**BRUNO LEANDRO GOMES DE CASTRO**

**UM PROCESSO PARA DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES  
ABERTOS UTILIZANDO O FRAMEWORK INSTITUTIONAL ENVIRONMENTS  
BASEADO NA METODOLOGIA GAIA**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Aprovado em: 03 / Dezembro / 2014.

**BANCA EXAMINADORA**

---

Prof<sup>o</sup>. Dr. Marcos Antonio de Oliveira (Orientador)  
Universidade Federal do Ceará-UFC

---

Prof<sup>o</sup>. Msc. Bruno Góis Mateus  
Universidade Federal do Ceará-UFC

---

Prof<sup>o</sup>. Msc. Enyo José Tavares Gonçalves  
Universidade Federal do Ceará-UFC



## **AGRADECIMENTOS**

Gostaria de expressar aqui minha gratidão a todos que de alguma forma me auxiliaram na conclusão deste trabalho e fizeram parte na minha caminhada durante a graduação.

Primeiramente a Deus por ter me dado a oportunidade de chegar ao final do meu curso e estar concluindo este trabalho.

Quero agradecer também aos professores Enyo Gonçalves, Bruno Góis por terem integrado a banca examinadora e também aos professores Diego Andrade, Lincoln Rocha. Agradeço a esses professores por todas as suas contribuições, e em especial ao professor Marcos de Oliveira, por ter sido meu orientador e pacientemente ter prestado toda assistência durante a execução deste trabalho.

Quero agradecer também a minha família, por todo auxílio que tive deles em todos os sentidos. Por sempre estarem me incentivando a continuar buscando meus objetivos e por nunca ter medido esforços para estarem ao meu lado sempre que precisei. Ao meu pai Francisco Alberto, a minha mãe Maria Lerenice e a minha irmã Ariana Gomes.

Quero fazer um agradecimento especial também aos meus grandes amigos Alexandra Valesca, José Diógenes, Pedro Henrique, Salomão Silva, Thalita Maria e Micaelly Soares. Por terem me dado o maior incentivo e apoio para concluir este trabalho, por sempre estarem ao meu lado e serem os melhores amigos que já tive.





## RESUMO

A criação de um processo visa padronizar uma sequência de atividades que devem ser executadas para alcançar um objetivo. Em Engenharia de Software, um processo de desenvolvimento é usado com a pretensão de alcançar maior qualidade nos produtos gerados no desenvolvimento, por meio de uma sequência de passos que compõem atividades, as quais compõem fases, como, por exemplo, definição do domínio da aplicação, levantamento de requisitos, análise do design, codificação e testes do sistema. Cada paradigma de desenvolvimento possui suas especificidades e portanto requer processos que tratem estas especificidades. O desenvolvimento de sistemas multiagentes vem crescendo rapidamente, assim sendo, diversas ferramentas têm sido propostas para auxiliar o desenvolvimento desses sistemas. Entretanto, desenvolver esse tipo de sistema ainda representa um grande desafio, principalmente quando tratamos de sistemas multiagentes abertos. O *Institutional Environments* é um *framework* para desenvolvimento de sistemas multiagentes abertos, cujo principal objetivo é promover uma abordagem institucional com mecanismos que mantenham a autonomia dos agentes, sem ter que obrigá-los a seguir a risca as regras do sistema. Este trabalho visa definir e apresentar um processo de desenvolvimento de sistemas multiagentes abertos utilizando o *framework Institution Environments* como recurso para a fase de desenvolvimento. O processo aqui apresentado baseia-se na metodologia Gaia, que apresenta um contexto baseado na atribuição de papéis aos agentes para poderem interagir no sistema. Apresenta também a ideia de organização dentro do ambiente do sistema, onde os agentes membros deverão seguir regras para realizar suas atividades. Entretanto, na sua forma original, Gaia não se adéqua ao desenvolvimento de sistemas multiagentes abertos justamente por tais regras, que restringem demais o comportamento dos agentes. Também é apresentada uma demonstração de uso do processo para o desenvolvimento do Mercado Livre como um sistema multiagente aberto com abordagem institucional.

Palavras chave: Processo de Software, Sistemas Multiagentes, Metodologia Gaia, Ambientes Institucionais.



## **ABSTRACT**

The creation of a process aimed at standardizing a sequence of tasks that must be performed to reach a goal. In software engineering, the development process is used with claim of achieving higher quality in products made it in the development stage, through a sequence of steps that comprise activities, which comprise stages such as, definition of the application domain, requirements gathering, design analysis, development and system's test. Each development paradigm has its specificities. Therefore requires processes that deal these specificities. The development of multi-agent systems has been growing quickly, thus, several tools have been proposed to aid the development of these systems. However, developing such a system is still a challenge, especially when dealing with open multi-agent systems. The Institutional Environments is a framework for developing open multi-agent systems whose main goal is to promote an institutional approach with mechanisms that maintain the autonomy of the agents, without having to force them to follow strictly the rules of the system. This work aims to define and present a development process for open multi-agent systems using the framework Institution Environments as a resource for the development stage. The process presented here is based on the Gaia methodology, which provides a context based on the assignment of roles to agents in order to interact with the system. Also introduces the idea of organization within the system environment, where agent members must follow the rules to accomplish their activities. However, in its original way, Gaia does not fit the development of open multi-agent systems precisely due these rules, which restrict the behavior of other agents. Is also presented a demonstration of using this process for the development of the Mercado Livre as an open multi-agent system with institutional approach.

Keywords: Multiagent Systems, Gaia Methodology, Institutional Environments, Software Development Process.



## LISTA DE FIGURAS

Figura 1: Representação da visão organizacional de um SMA na Metodologia Gaia .....	28
Figura 2: Representação dos modelos e fases da metodologia Gaia e suas relações. ....	29
Figura 3: Representação das forças que influenciam a identificação da estrutura organizacional.....	33
Figura 4 - Espaço Normativo do Institutional Environment .....	41
Figura 5: Representação do fluxo principal do processo.....	49
Figura 6: Representação do fluxo das tarefas no levantamento de requisitos. ....	50
Figura 7: Representação do fluxo das tarefas da fase de análise. ....	53
Figura 8: Representação do fluxo da Fase de Projeto .....	63
Figura 9: Representação do fluxo das tarefas do Projeto de Arquitetura. ....	64
Figura 10: Representação do fluxo de tarefas do Projeto de Designer.....	70
Figura 11: Representação do fluxo da fase de Implementação das CPNs.....	73
Figura 12: Representação do fluxo da fase de testes do sistema. ....	75
Figura 13: Representação da arquitetura do Mercado Livre .....	90
Figura 14: Representação dos papéis do Mercado Livre usando CPNs.....	91
Figura 15: Representação da atividade do papel Visitante usando CPN.....	96
Figura 16: Representação das atividades do papel Vendedores usando CPN.....	97
Figura 17: Representação das atividades do papel Compradores usando CPN. ....	97
Figura 18: Representação da CPN das atividades de um agente visitante no ambiente.....	99
Figura 19: Representação dos cenários das atividades de um deputy agent no ambiente. ....	100
Figura 20: Representação do cenário de Anunciar um produto de um Vendedor.....	101
Figura 21: Representação do cenário de Cancelar Anúncio de um Vendedor .....	101
Figura 22: Representação do cenário de Aceitar Oferta de um Vendedor. ....	102
Figura 23: Representação do cenário de Rejeitar Oferta de um Vendedor .....	102
Figura 24: Representação do cenário de Confirmar Pagamento de um Vendedor.....	103
Figura 25: Representação do cenário de Enviar Produto .....	103
Figura 26: Representação do cenário de Fazer Oferta por um Comprador. ....	104
Figura 27: Representação do cenário de Cancelar Oferta por um Comprador.....	105
Figura 28: Representação do cenário de Confirmar Compra .....	105
Figura 29: Representação do cenário de Realizar Pagamento.....	106
Figura 30: Representação do cenário de Confirmar Recebimento de um Produto/Serviço ...	106
Figura 31: Representação do fluxo de trabalho de um Visitante.....	107
Figura 32: Representação do fluxo de trabalho do anúncio de um produto/serviço por um Vendedor. ....	108
Figura 33: Representação do fluxo de trabalho de cancelar o anúncio de um produto/serviço por um Vendedor. ....	108
Figura 34: Representação do fluxo de trabalho de aceitar uma oferta por um Vendedor. ....	109
Figura 35: Representação do fluxo de trabalho de rejeitar uma oferta por um Vendedor. ....	109
Figura 36: Representação do fluxo de trabalho de confirmar um pagamento por um Vendedor. ....	110
Figura 37: Representação do fluxo de trabalho do envio de um produto/serviço por um Vendedor. ....	110
Figura 38: Representação do fluxo de trabalho de fazer uma oferta por um Comprador. ....	111
Figura 39: Representação do fluxo de trabalho de Cancelar uma Oferta por um Comprador. ....	111
Figura 40: Representação do fluxo de trabalho de confirmar compra por um Comprador...	112



Figura 41: Representação do fluxo de trabalho de Realizar o Pagamento de por um Comprador.....	112
Figura 42: Representação do fluxo de trabalho de Confirmar Recebimento do produto pelo Comprador.....	113
Figura 43: Representação do fluxo de trabalho do cancelamento de um cadastro por um papel.....	114
Figura 44: Representação do fluxo de trabalho de um comentário no sistema por um papel.	114
Figura 45: Representação do fluxo de trabalho da mudança de papel feita por um papel. ....	115





## LISTA DE QUADROS

Tabela 1: Diferença entre sistemas baseados em agentes e sistemas baseados em objetos .....	44
Tabela 2: Principais diferenças entre as metodologias para sistemas multiagentes .....	45
Tabela 3: Representação da tarefa Elicitação e Especificação dos Requisitos.....	51
Tabela 4: Representação da tarefa Validar Requisitos .....	51
Tabela 5: Representação da tarefa Identificar as Organizações do Sistema.....	54
Tabela 6: Representação da tarefa Definir modelo ambiental do sistema.....	55
Tabela 7: Representação da tarefa para Definir os papéis preliminares do sistema.....	56
Tabela 8: Representação da tarefa Definir os atos institucionais .....	58
Tabela 9: Representação da tarefa para definir o modelo preliminar de interações.....	59
Tabela 10: Representação da tarefa para definir os compromissos.....	60
Tabela 11: Representação da tarefa Especificar a Reputação .....	62
Tabela 12: Representação para a tarefa de definir a estrutura organizacional .....	65
Tabela 13: Representação da tarefa Representar a Estrutura Organizacional .....	67
Tabela 14: Representação da tarefa Conclusão dos modelos de papéis e Interações.....	68
Tabela 15: Representação da tarefa Definição do Modelo de Agente .....	70
Tabela 16: Representação da tarefa Definição do Modelo de Serviços .....	71
Tabela 17: Representação da tarefa Representar Modelos de Agentes e Serviços Usando CPNs.....	72
Tabela 18: Representação da tarefa de Implementar as CPNs .....	74
Tabela 19: Representação da tarefa Realizar testes das CPNs .....	75
Tabela 20: Detalhes do Papel de Visitantes.....	79
Tabela 21: Detalhes do Papel de Vendedor.....	79
Tabela 22: Detalhes do Papel Comprador .....	80
Tabela 23: Representação do modelo de ambiente do Mercado Livre.....	81
Tabela 24: Representação das interações do papel Visitante .....	84
Tabela 25: Representação das interações do papel Comprador .....	84
Tabela 26: Representação das interações do papel Vendedor.....	85
Tabela 27: Compromissos assumidos pelo papel Visitante .....	85
Tabela 28: Compromissos assumidos pelo papel Vendedor .....	86
Tabela 29: Compromissos assumidos pelo papel Comprador.....	87
Tabela 30: Detalhes do papel Visitante .....	92
Tabela 31: Detalhes do papel Comprador .....	92
Tabela 32: Detalhes do papel Vendedor.....	93
Tabela 33: Detalhes dos serviços do papel Visitante .....	95
Tabela 34: Detalhes do papel Vendedor.....	95
Tabela 35: Detalhes do papel Comprador .....	96



## LISTA DE ABREVIATURAS E SIGLAS

SMA	Sistema Multiagentes Abertos
SMA <sub>s</sub>	Sistemas Multiagentes Abertos
IE	Institutional Environments
CPNs	Redes de Petri Coloridas
FIPA	Foundation for Intelligent Physical Agents
ACL	Agent Communication Language
EPF	Eclipse Process Framework Project
AUML	Agent Unified Modeling Language
UML	Unified Modeling Language



## SUMÁRIO

1. INTRODUÇÃO.....	16
1.1. Objetivos.....	18
1.2. Metodologia.....	18
1.3. Estrutura do Trabalho.....	19
2. REVISÃO BIBLIOGRÁFICA.....	20
2.1. Principais Problemas no Desenvolvimento de Sistemas Multiagentes Abertos.....	20
2.1.1. Definir padrão de comunicação entre os agentes.....	21
2.1.2. Desenvolver um sistema que abstraia as características das entidades do mundo real e de fácil divisão de tarefas e responsabilidades.....	21
2.1.3. Como garantir que os agentes cumpram suas tarefas sem ter que restringir demais o comportamento dos agentes, mantendo o desempenho do sistema aceitável...	23
2.2. Metodologias de Desenvolvimento para Sistemas Multiagentes.....	24
2.3. Metodologia Gaia.....	27
2.3.1. Estrutura de Gaia.....	28
2.3.1.1. Fase de Análise.....	29
2.3.1.2. Fase de Projeto.....	31
2.3.2. Limitações.....	34
2.4. Institutional Environments.....	35
2.4.1. Desenvolvimento de um Ambiente Institucional.....	36
2.4.2. Instituições, Papéis e os Atos Institucionais.....	37
2.4.3. Mecanismos de Controle Social.....	37
2.4.4. Métodos e Modelagem da Interação dos Agentes.....	39
2.5. Conclusão.....	41
3. ETAPAS PARA A DEFINIÇÃO DO PROCESSO DO INSTITUTIONAL ENVIRONMENTS.....	43
3.1. Primeira Etapa.....	44
3.2. Segunda Etapa.....	45
3.3. Terceira Etapa.....	46
3.4. Quarta Etapa.....	46
3.5. Quinta Etapa.....	47
3.6. Sexta Etapa.....	48
4. PROCESSO DO INSTITUTIONAL ENVIRONMENTS.....	48
4.1. Fase de Levantamento dos Requisitos.....	50
4.1.1. Elicitação e Especificação dos Requisitos.....	51
4.1.2. Validar os Requisitos.....	51
4.2. Fase de Análise.....	52
4.2.1. Identificar As Organizações Do Sistema.....	54
4.2.2. Modelo Ambiental do Sistema.....	55
4.2.3. Modelo Preliminar de Papéis.....	56
4.2.4. Definir os Atos Institucionais.....	57
4.2.5. Modelo Preliminar de Interações.....	58
4.2.6. Definir os Compromissos.....	60
4.2.7. Especificar a Reputação.....	61
4.3. Fase de Projeto.....	62



4.3.1.	Projeto De Arquitetura.....	63
4.3.1.1.	Definir Estrutura Organizacional.....	64
4.3.1.2.	Representar a Estrutura Organizacional .....	67
4.3.1.3.	Conclusão dos Modelos de Papéis e de Interações .....	68
4.3.2.	Projeto de Designer .....	69
4.3.2.1.	Definição do Modelo de Agente .....	70
4.3.2.2.	Definição do Modelo de Serviços.....	71
4.3.2.3.	Representar Modelos de Agentes e Serviços Usando CPNs .....	72
4.4.	Fase de Implementação das CPNs .....	73
4.4.1.	Implementação das CPNs.....	73
4.5.	Fase de Testes do Sistema.....	74
4.5.1.	Realizar testes das CPNs .....	75
5.	DEMONSTRAÇÃO DE USO .....	76
5.1.	Introdução .....	76
5.2.	Fase de Levantamento dos Requisitos .....	77
5.2.1.	Elicitar os Requisitos .....	77
5.2.2.	Validar os Requisitos.....	78
5.3.	Fase de Análise.....	78
5.3.1.	Identificação das Organizações do Sistema.....	78
5.3.2.	Modelo Preliminar de Papéis.....	79
5.3.3.	Modelo Ambiental do Sistema .....	80
5.3.4.	Definir os Atos Institucionais .....	82
5.3.5.	Modelo Preliminar de Interações.....	84
5.3.6.	Definir os Compromissos .....	85
5.3.7.	Especificar a Reputação.....	87
5.4.	Fase de Projeto .....	89
5.4.1.	Projeto de Arquitetura .....	89
5.4.1.1.	Definir Estrutura Organizacional.....	90
5.4.1.2.	Representar Estrutura Organizacional .....	91
5.4.1.3.	Conclusão dos Modelos de Papéis e Interações.....	92
5.4.2.	Projeto de Designer .....	94
5.4.2.1.	Definição do Modelo de Agente .....	94
5.4.2.2.	Definição do Modelo de Serviços.....	95
5.4.2.3.	Representar Modelo de Agente e Serviços usando CPNs .....	96
5.5.	Fase de Implementação das CPNs .....	98
5.5.1.	Implementar as CPNs .....	98
5.6.	Fase de Testes do Sistema.....	115
5.6.1.	Realizar Testes das CPNs .....	115
6.	CONSIDERAÇÕES FINAIS .....	116
6.1.	Conclusão.....	116
6.2.	Trabalhos Futuros .....	117
	REFERÊNCIAS .....	118
	APÊNDICES .....	124





## 1. INTRODUÇÃO

O desenvolvimento de software necessita de métodos e de ferramentas que integrem os diferentes elementos de um sistema, de forma a criar um ambiente estável de execução. Quando se está desenvolvendo um sistema complexo, o uso dessas ferramentas é ainda mais necessário. Um processo de software pode ser uma dessas ferramentas.

A criação de um processo visa padronizar uma sequência de atividades que devem ser executadas para alcançar um objetivo (FALBO; BARCELLOS, 2011). Em Engenharia de Software, um processo de desenvolvimento é usado com a pretensão de alcançar maior qualidade nos produtos gerados durante o desenvolvimento ou até mesmo para definir uma sequência de passos para realizar uma atividade. “Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software” (FALBO; BARCELLOS, 2011, p. 5).

Podemos citar algumas vantagens na utilização de um processo de software: (FALBO; BARCELLOS, 2011).

- Produzir conhecimento de como realizar as atividades e quais artefatos gerar;
- Diminuição dos erros que poderiam ocorrer e do retrabalho;
- Produzir um sistema de maior qualidade, conseguindo assim a satisfação do cliente.

Atualmente um novo paradigma para desenvolvimento de software começou a ganhar notoriedade tanto em aplicações simples, quanto em aplicações na indústria. Trata-se do desenvolvimento orientado a agentes. O uso de sistemas multiagentes pode ser defendido pela capacidade de abstração e de distribuição de responsabilidades que esses sistemas possuem. E também pelo fato das aplicações serem projetadas e desenvolvidas baseadas em entidades de software autônomas (agentes), onde as mesmas podem atingir seus objetivos de forma flexível. Ou seja, da maneira que acharem conveniente, mantendo interações com os demais agentes e com o próprio sistema. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Contudo, este paradigma introduz uma série de novas abstrações e questões de concepção e desenvolvimento que devem ser consideradas ao se projetar um sistema. Como a representação da autonomia e do comportamento reativo dos agentes. Isso significa que, durante o desenvolvimento de um SMA (Sistemas Multiagentes Abertos) é importante tratar a

capacidade de decisão que os agentes devem ter no ambiente e o seu comportamento reativo sobre uma determinada situação.

Essas tarefas tornam-se ainda mais complexas ao se desenvolver SMAs abertos, onde agentes desenvolvidos por terceiros podem ser integrados ao sistema para realizar alguma tarefa ou utilizar algum serviço. Os agentes que compõem esses sistemas, muitas vezes são desenvolvidos com diferentes objetivos e tornam-se difíceis de gerenciar para que o sistema se mantenha estável. O que torna o desenvolvimento desse tipo de sistemas um grande desafio. Trata-se de como conseguir manter um desempenho considerado aceitável do sistema como um todo, sem ter que limitar demasiadamente o comportamento dos agentes envolvidos no ambiente de trabalho (OLIVEIRA, 2013).

Essa questão sobre o desenvolvimento de SMAs abertos é nova na comunidade de pesquisa relacionada a SMAs. A maioria dos trabalhos existentes são relacionados a sistemas fechados, onde a quantidade e tipos de agentes são conhecidos e limitados.

Em Oliveira (2013), é proposto um *framework* para desenvolvimento de sistemas multiagentes abertos, cujo principal objetivo é promover mecanismos que mantenham a autonomia dos agentes, sem que seja necessário obrigar-lhes a seguir a risca as regras do sistema. Esse framework foi denominado *Institutional Environments* (IE).

A ideia abordada pelo *Institutional Environments* é utilizar uma abordagem institucional para desenvolver SMAs, onde os sistemas seriam modelados à imagem de uma instituição tradicional, onde os agentes representam papéis que devem seguir regras estabelecidas pela própria instituição para que suas ações no sistema sejam consideradas válidas. E que permita também que possam escolher a melhor maneira de realizar suas ações.

Dessa forma os agentes assumiriam papéis dentro do sistema e poderiam interagir com o sistema por meio de atos e regras (representadas por compromissos entre agentes), pré-definidas pelo designer do sistema. Existindo também a possibilidade de o próprio agente criar seus próprios atos e/ou compromissos.

Diversas abordagens e metodologias já foram propostas a fim de facilitar a representação e o desenvolvimento de SMAs abertos, podemos citar Prometheus (DÁRIO, 2005), Tropos (ZANUZ, 2007), Zeus (BRITO; GAVA; TAVARES; MENEZES, 2001) e Gaia (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). No entanto, nenhuma destas dá apoio ao desenvolvimento de sistemas multiagentes abertos. Surgindo a necessidade de uma proposta aderente a esse tipo de sistemas.

Com o surgimento dessa necessidade propomos desenvolver neste trabalho um processo de desenvolvimento aderente ao *framework Institutional Environments* que fosse de

acordo com o contexto de SMAs abertos e que utilizasse métodos já existentes em outras metodologias.

A seguir são apresentados os objetivos, metodologias e a estrutura do trabalho.

### **1.1. Objetivos**

O objetivo principal deste trabalho é apresentar um processo de desenvolvimento onde seja feita uma adaptação da metodologia Gaia com abordagem institucional defendida pelo *Institutional Environments*. Dessa maneira conseguiremos aplicar os princípios metodológicos defendidos por Gaia a um contexto de SMAs Abertos, estendendo-a assim para o gerenciamento da autonomia dos agentes, e ampliando a gama de sistemas aos quais ela pode ser aplicada.

Os objetivos específicos desse trabalho foram:

- Descrever a problemática relacionada ao desenvolvimento de SMAs.
- Selecionar e analisar algumas metodologias para desenvolvimento de SMAs já existentes identificando sua relevância para a criação do processo.
- Descrever a metodologia que mais se adeque ao contexto do IE para servir como base para criação do processo.
- Identificar características do IE para a definição do processo.
- Definir um processo para a implementação de sistemas multiagentes abertos baseado na metodologia escolhida e nas características do *Institutional Environments*.
- Demonstrar o uso do processo criado em um cenário onde pode-se aplicar SMA Abertos.

### **1.2. Metodologia**

A primeira fase desse trabalho teve como objetivo fazer um levantamento bibliográfico sobre técnicas utilizadas para a definição de um processo de desenvolvimento de sistemas, e de técnicas que conseguissem explorar as características típicas do paradigma orientado a agentes e aplicá-las em um projeto.

A segunda fase consistiu em um estudo sobre as metodologias propostas em trabalhos anteriores para identificação de pontos que devem ser levados em consideração no

momento de se criar um processo de desenvolvimento para sistemas multiagentes. Tanto pontos que explorem as características próprias do paradigma, como pontos que explorem características específicas de cada tipo de sistema. Assim, pode-se entender melhor como adequar o desenvolvimento de sistemas para determinados tipos de ambientes a uma dada metodologia.

A terceira fase consistiu em um estudo sobre o *framework Institutional Environments*, para identificação das técnicas utilizadas por essa ferramenta para desenvolver SMAs com abordagem institucional.

A quarta fase visou escolher a metodologia que mais se adequasse a abordagem defendida pelo IE e que tornasse viável a elaboração de um processo de desenvolvimento.

A quinta fase consistiu na representação do processo de desenvolvimento da metodologia Gaia, escolhida como sendo a mais adequada, e na criação do processo de desenvolvimento do *Institutional Environments* baseando-se na metodologia Gaia.

E por fim, a sexta e última fase teve como objetivo realizar uma demonstração do uso do processo. Para isso aplicamos o processo no desenvolvimento no sistema de e-commerce Mercado Livre.

### **1.3. Estrutura do Trabalho**

Este trabalho está dividido em sete seções, incluindo esta introdução.

A seção 2, Revisão Bibliográfica, apresenta todos os conceitos importantes estudados durante este trabalho. Apresentando suas definições e principais lições aprendidas com cada conceito para ser usada na definição do processo.

A seção 3, Procedimentos Metodológicos, apresenta como se deu a execução deste trabalho, citando quais foram os passos executados.

A seção 4, Definição do Processo do Institutional Environments, apresenta detalhadamente como se deu a elaboração de todas as etapas do processo.

A seção 5, Processo do Institutional Environments, apresenta as etapas do processo, detalhando suas atividades, tarefas e passos.

A seção 6, Demonstração de Uso, apresenta um exemplo de aplicação do processo proposto na modelagem do sistema do Mercado Livre.

Por fim, a seção 7 apresenta as Considerações Finais deste trabalho, propondo a realização de trabalhos futuros.

## **2. REVISÃO BIBLIOGRÁFICA**

Esta seção apresenta todos os conceitos importantes para a realização deste trabalho. Está composta por quatro subseções, sendo: A subseção 2.1, Principais Problemas no Desenvolvimento de Sistemas Multiagentes Abertos, onde são abordados os principais problemas encontrados para o desenvolvimento de SMAs abertos e possíveis soluções encontradas hoje na comunidade de pesquisa. A subseção 2.2, Metodologia Gaia, onde é abordada as principais características da metodologia Gaia e os principais pontos que a fizeram adequada para a criação do processo deste trabalho. A subseção 2.3, *Institutional Environments*, onde é apresentado os principais conceitos e técnicas utilizadas pelo *framework* a fim da definição do seu processo. E por fim, a subseção 2.4, contendo as conclusões finais da seção e faz um comparativo entre as características da metodologia Gaia e entre o *framework Institutional Environments*.

### **2.1. Principais Problemas no Desenvolvimento de Sistemas Multiagentes Abertos**

Em uma organização do mundo real, independente da sua natureza, diversas ideias são usadas para se resolver um mesmo problema. Muitas vezes isso pode acarretar divergências ou até mesmo conflitos entre os funcionários (ROSA; COSER, 2005). Do mesmo modo, em um sistema multiagentes aberto os agentes podem decidir por realizar uma tarefa de diferentes maneiras. Muitas vezes isso acarretará em problemas nas interações entre os agentes e, conseqüentemente, no andamento das atividades.

Assim, é importante desenvolver sistemas abertos que estejam preparados para lidar com tais divergências de ideias, implementando métodos e mecanismos para isso (OLIVEIRA, 2013).

A seguir citamos alguns problemas que podem ser considerados como as principais dificuldades no desenvolvimento de sistemas abertos. Sobre cada problema é feita uma abordagem de como o *Institutional Environments* se comporta diante de tais problemas.

### **2.1.1. Definir padrão de comunicação entre os agentes.**

No mundo real, quando nós humanos viajamos para outro país, é costume que comecemos a nos comunicar usando o idioma padrão do país em questão. O contexto de comunicação entre os agentes de um SMA pode seguir a mesma ideia. O fato de sistemas multiagentes abertos serem desenvolvidos pensando-se na chance de agentes externos virem a ingressar no sistema faz com que a comunicação entre esses agentes se torne um dos principais problemas na hora de se implementar um sistema desse tipo (RIBAS, 2003).

Esta diversidade de tipos de agentes que compõem um SMA acaba por influenciar na execução do sistema, pois o comportamento desses agentes acaba sendo distinto do que é esperado pelo ambiente, tanto com relação aos seus objetivos quanto a reações diferentes aos mesmos estímulos do ambiente. (GLUZ; VICCARI, 2003).

Algumas abordagens já foram feitas para se resolver isso, mas a maioria era focada apenas no comportamento do agente no ambiente, não em como ele irá realizar suas atividades. Deixando o ambiente do sistema muito restrito. Esteva (2003) e Brafman (1996) são dois exemplos desses trabalhos. Essa restrição é mais adequada para sistemas pequenos, pois a complexidade das atividades é menor. Para sistemas abertos, a tarefa de implementar métodos padrões de comunicação fica mais complexa por poder haver agentes no sistema que foram desenvolvidos por terceiros (OLIVEIRA, 2013).

No IE a modelagem dos protocolos que definem as relações entre os agentes é feita usando-se CPNs (Redes de Petri Coloridas), onde a troca de mensagens entre os agentes deve ser do formato FIPA-ACL (CPN GROUP, 2014).

Um agente que venha a ingressar no sistema terá que assumir um papel. Este papel por sua vez, é representado por uma própria CPN. As relações e a união dessas CPNs formam no IE um ambiente distribuído chamado de Ambiente Normativo.

### **2.1.2. Desenvolver um sistema que abstraia as características das entidades do mundo real e de fácil divisão de tarefas e responsabilidades.**

A teoria organizacional tem sido aplicada com sucesso no desenvolvimento de sistemas, pois quando se está desenvolvendo um sistema complexo é interessante a utilização de métodos e de ferramentas que integrem os elementos do sistema (COSER; ROSA, 2004).

Segundo essa teoria, qualquer membro de uma organização, seja um ser humano ou um componente eletrônico e tecnológico, deverá ter habilidades limitadas e não será capaz de realizar várias tarefas individualmente para se chegar a um objetivo comum. (SIMON, 1957 *apud* ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Pode-se entender de instituição como sendo uma representação de organizações de software onde os agentes irão representar as entidades do mundo real através de papéis assumidos e pré-definidos para interagir dentro do ambiente. Após assumir um papel no ambiente os agentes irão interagir com o ambiente através de atos institucionais específicos para o seu papel. (OLIVEIRA, 2013).

A utilização de sistemas multiagentes com uma abordagem institucional pode ser utilizada para implementação de sistemas que precisem abstrair ao máximo as relações entre as entidades do mundo real que estão sendo representadas e ao mesmo tempo ser de fácil divisão de tarefas.

No entanto, o desenvolvimento desse tipo de sistemas, principalmente sistemas multiagentes abertos, ainda sofrem com a complexidade de manter um ambiente que opere de maneira aceitável e de manter uma estrutura homogênea entre as relações dos agentes dentro do sistema. Não somente isso, em SMAs abertos deve-se também manter a independência e autonomia dos agentes (OLIVEIRA, 2013)

A ideia abordada *pele Institutional Environments* é de modelar os sistemas a imagem de uma instituição tradicional, onde os agentes representam papéis que devem cumprir compromissos estabelecidos com a própria instituição, ou criar seus próprios compromissos, para que suas ações no sistema sejam consideradas válidas.

Assim sendo, a utilização do IE como *framework* para desenvolver SMAs abertos pode ser defendida por algumas características próprias do IE (OLIVEIRA, 2013):

- Proporciona o desenvolvimento de SMA aberto de forma que se mantenha um desempenho aceitável do sistema gerenciando o comportamento dos seus agentes de maneira mais ou menos severa através do uso de conceitos de reputação e compromissos entre os agentes.
- O conceito de reputação de agentes é usado para garantir a aplicação das regras do sistema, de forma que se mantenha a autonomia dos agentes para seguir ou não seus compromissos. Todas as interações entre os agentes serão então baseadas nos dados de reputação dos agentes envolvidos;

- Todo agente externo que venha a ingressar no sistema terá um papel e um monitor responsável em monitorar suas atitudes dentro do sistema para atualização de sua reputação.
- Proporciona ao sistema desenvolvido uma maneira de lidar com a heterogeneidade do sistema, pois o mesmo pode ser formado por agentes desenvolvidos por terceiros.
- O uso de CPNs para representar a interação entre os agentes. Isso possibilita que a troca de mensagens entre os agentes possa ser feita de maneira concorrente entre vários agentes. Diferentemente de outras abordagens que utilizam Máquinas de Estado Finitos, onde as interações são representadas de maneira sequencial.

### **2.1.3. Como garantir que os agentes cumpram suas tarefas sem ter que restringir demais o comportamento dos agentes, mantendo o desempenho do sistema aceitável.**

Diversos conceitos são utilizados na comunidade de pesquisa para representar as regras de sistema em um SMA. Conceitos como normas e regras estão entre os mais comuns e mais aceitos. O problema é que esses conceitos, normalmente, restringem demais o comportamento dos agentes e o seu poder de decisão sobre suas atividades (BRITO; GAVA; TAVARES; MENEZES, 2001). O que não é adequado para sistemas abertos, que podem receber novos agentes membros desenvolvidos por terceiros para compor o sistema.

Para manter o controle global de um sistema aberto é necessária a cooperação mútua entre os agentes. Para isso é necessário que se construa um esquema de confiança entre os agentes para que eles possam interagir e atingirem um mesmo objetivo (OLIVEIRA, 2013).

No IE é usado conceito de confiança entre o agente que representa a instituição e os agentes externos ao sistema, firmando-se compromissos, de modo que as restrições do sistema possam ser implementadas, mas de forma que o próprio agente decida como irá proceder diante de tais restrições e também de forma que se mantenha uma reputação do agente em questão, para torná-lo um parceiro desejável nas atividades de cooperação entre agentes do SMA (OLIVEIRA, 2013).

Estes compromissos delimitam como cada papel exercido pelo agente deve se comportar no sistema. A análise feita sobre o cumprimento ou não dos compromissos firmados pelos agentes irá influenciar diretamente na reputação dos mesmos, pois irá informar



se os mesmos são confiáveis para assumirem ou dividirem responsabilidades com outros com agentes dentro do sistema (OLIVEIRA, 2013).

Através da análise dos dados de reputação dos agentes, as interações entre os mesmos só poderão ser realizadas por agentes que mantenham uma reputação considerada boa pela instituição. Dessa forma, será mantido o ciclo de interações. Caso contrário, outros agentes serão procurados para a troca de mensagens (OLIVEIRA, 2013).

O modelo de confiança no IE é então formado por esses compromissos firmados entre os agentes, por dados de reputação mantidos para cada agente no ambiente e agentes monitores, responsáveis em monitorar tais compromissos e informar aos agentes de reputação como está se comportando o agente em questão (OLIVEIRA, 2013).

## **2.2. Metodologias de Desenvolvimento para Sistemas Multiagentes**

Com a necessidade de se definir pontos importantes em uma metodologia para desenvolvimento de SMA, faz-se necessário o estudo preliminar de algumas das metodologias já abordadas em outras obras. A seguir são apresentadas as principais características de algumas das abordagens mais utilizadas e conhecidas atualmente no desenvolvimento de sistemas multiagentes. Na tabela apresentada na seção 3.2 podemos ver de forma resumida uma comparação entre as mesmas seguindo um mesmo aspecto.

- **Metodologia Prometheus**

O aspecto fundamental da metodologia Prometheus é a associação de tarefas a agentes. Ela especifica também protocolos de interação modelados em AUML. O que destaca Prometheus de outras metodologias é o intuito que esta metodologia tem de obter um conjunto de agentes que se comportem no sistema para alcançar um objetivo definido, e que interagem quando for necessário seguindo os protocolos de comunicação. (DÁRIO, 2005).

Prometheus suporta o desenvolvimento de agentes inteligentes que possuem objetivos, crenças, planos e eventos. Proporciona mecanismos de estruturação hierárquica que permite ser executado em diferentes níveis de abstração. A metodologia utiliza o conceito de capacidade, a qual pode ser composta por planos, eventos, crenças e outras capacidades que dão habilidades ao agente. O agente é composto por várias capacidades e cada uma delas tem uma função específica (DÁRIO, 2005).

Dois aspectos sobre a metodologia Prometheus foram definidos em Padgham e Winikoff (2003) e é importante descrevê-los aqui. São eles:

- Seu processo de aplicação é interativo e não sequencial, como na maioria das metodologias. Embora sua definição seja feita de maneira sequencial, o foco a cada interação muda, dependendo do estado de evolução do projeto.
- Prevê níveis hierárquicos entre os elementos do sistema. O que permite que os projetos sejam modelados em diferentes níveis de abstração.

É importante destacarmos também que essa metodologia está em constante evolução, pois é usada pelos próprios desenvolvedores em seus projetos e em práticas pedagógicas por alunos de graduação da RMIT University em Melbourne Austrália (PADGHAM; WINIKOFF, 2003).

Com isso fica claro que Prometheus tem uma abordagem centrada no agente. Um agente terá capacidades sem qualquer noção de organização. Interagindo quando for necessário. Embora sejam úteis no desenvolvimento de sistemas fechados ela não é adequada para implementação de SMAs abertos e também não se adéqua ao contexto do IE.

- **Metodologia Tropos**

Tropos é uma metodologia de desenvolvimento de SMAs que abrange todas as atividades de um processo de desenvolvimento de software, desde as suas fases iniciais até a sua implementação, dando ênfase nas atividades de especificação do sistema (BRESCIANI *et al.* 2004). Sua aplicação é indicada para sistemas baseados em atores, objetivos e dependências sociais (ZANUZ, 2007). Apesar de prover atividades para todo o processo de desenvolvimento, sua maior ênfase é na atividade de levantamento de requisitos. A ideia defendida é que a qualidade dos documentos de especificação do sistema irá influenciar diretamente no sucesso do projeto. As atividades da metodologia Tropos são (BRESCIANI *et al.* 2004):

- Requisitos Primários: seu objetivo é especificar o problema a ser resolvido. Se preocupa com a organização do sistema;
- Requisitos Atrasados: seu foco é em descrever o ambiente operacional do sistema, ou seja, suas funções e seus atributos relevantes. Definir seus requisitos funcionais e não-funcionais;
- Projeto Arquitetônico: se preocupa em definir a arquitetura geral do sistema. Em definir como seus elementos irão ser interligados.
- Projeto Detalhado: se preocupa em definir o comportamento de cada elemento da arquitetura.

Em Zanuz (2007), é feita uma especificação formal sobre TROPOS, onde descreve os elementos relevantes a metodologia (atores, objetivos, dependências) de um domínio e os relacionamentos entre eles. A descrição de cada elemento é estruturada em duas camadas:

- A camada externa é similar a uma declaração de classe. Faz a associação de um elemento a um conjunto de atributos que definem a sua estrutura.
- A camada interna expressa limitações sobre o tempo de vida dos objetos.

Assim como a metodologia Prometheus, Tropos também especifica padrões de interação modelados em AUML e preocupa-se com a identificação de atividades, sendo aqui, identificadas desde a fase de análise de requisitos dos sistemas. Também não se preocupa com noção de organização e tem uma abordagem focada no agente. Tropos também não se adequa ao desenvolvimento de SMAs abertos e ao contexto do IE.

- **Metodologia Zeus**

Zeus é um framework para análise, projeto e construção de agentes. O objetivo desta ferramenta é prover o desenvolvimento de SMA de forma colaborativa. O contexto de Zeus prevê que a ferramenta seja capaz de prover os seguintes princípios (COLLINS; NDUMU, 1999):

- A ferramenta deverá ser capaz de oferecer mecanismos para a resolução de um problema, independentemente da aplicação. Oferecendo elementos de coordenação, execução, monitoramento das atividades.
- Deve prover mecanismos que ajudem na modelagem do sistema através da criação de artefatos.
- A ferramenta fosse capaz de estar sempre atualizada com os padrões das tecnologias utilizadas no desenvolvimento de SMAs. Por exemplo, prover mecanismos para implementar a troca de mensagens FIPA-ACL entre os agentes, que é um padrão bastante utilizado para a comunicação entre os agentes.

Assim como Gaia, Zeus também é orientada a papéis e pode ser aplicada em vários domínios. (BRITO; GAVA; TAVARES; MENEZES, 2001).

O que difere Zeus e Gaia é o suporte de ferramentas geradoras de agentes presente em Zeus, que permite a geração de partes dos agentes em código Java (BRITO; GAVA;

TAVARES; MENEZES, 2001). Zeus também fornece suporte para testes dos agentes implementados.

No processo de modelagem dos papéis em Zeus, são elaborados: Diagrama de Papéis, Diagrama de Interação e a Descrição dos Papéis.

Logo, Prometheus, Tropos e Zeus são metodologias centrada no agente, impõe restrições rígidas sobre o comportamento dos agentes dentro do sistema, modelando o seu modo de agir. Isso a torna inadequada para tratar a autonomia e reatividade dos agentes de um SMA aberto, não se adequando ao IE.

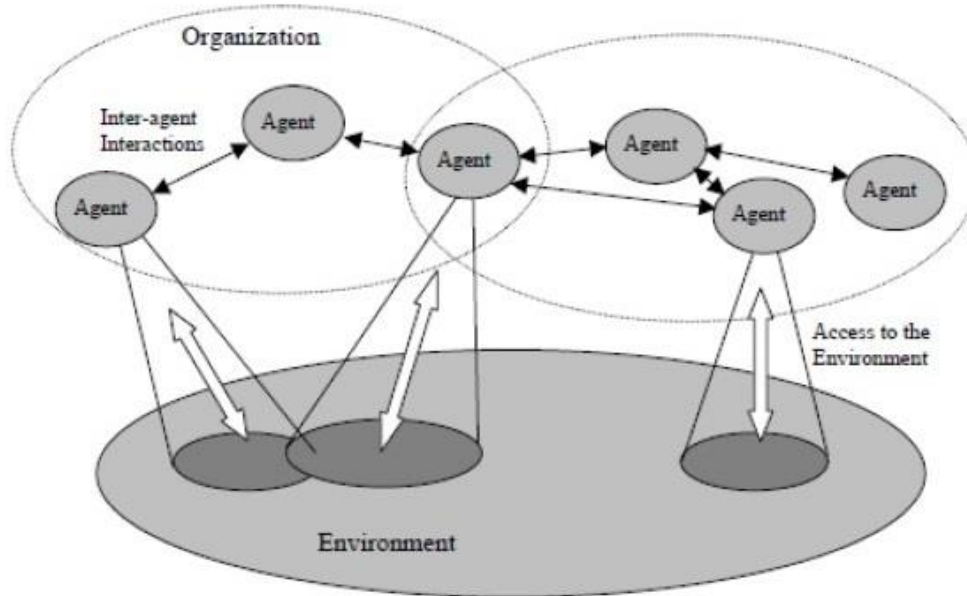
### **2.3. Metodologia Gaia**

Antes de falar com mais detalhes sobre como essa metodologia serviu como base para o desenvolvimento do processo do *Institutional Environments*, vamos apresentar uma visão geral de sua abordagem, seus principais estágios e artefatos gerados.

Gaia é uma metodologia projetada para o desenvolvimento de sistemas multiagentes com o objetivo de proporcionar maior qualidade na organização dos elementos do sistema. Baseando-se na ideia de que um SMA pode ser projetado como uma organização computacional, onde os agentes teram que exercer papéis para interagir no ambiente, Gaia pôde ser adaptada ao contexto do IE para a criação do processo proposto neste trabalho (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

A ideia defendida em Gaia é que um SMA pode ser dividido em suborganizações menores, compostas por grupos de agentes com características e objetivos parecidos . Cada grupo de agentes dentro de uma organização irá assumir um papel pré-definido. E será por meio desse papel que um agente poderá interagir dentro do ambiente com os demais agentes do sistema (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). A figura 1 demonstra a subdivisão das organizações de um sistema e da participação dos agentes nessas organizações.

Figura 1: Representação da visão organizacional de um SMA na Metodologia Gaia



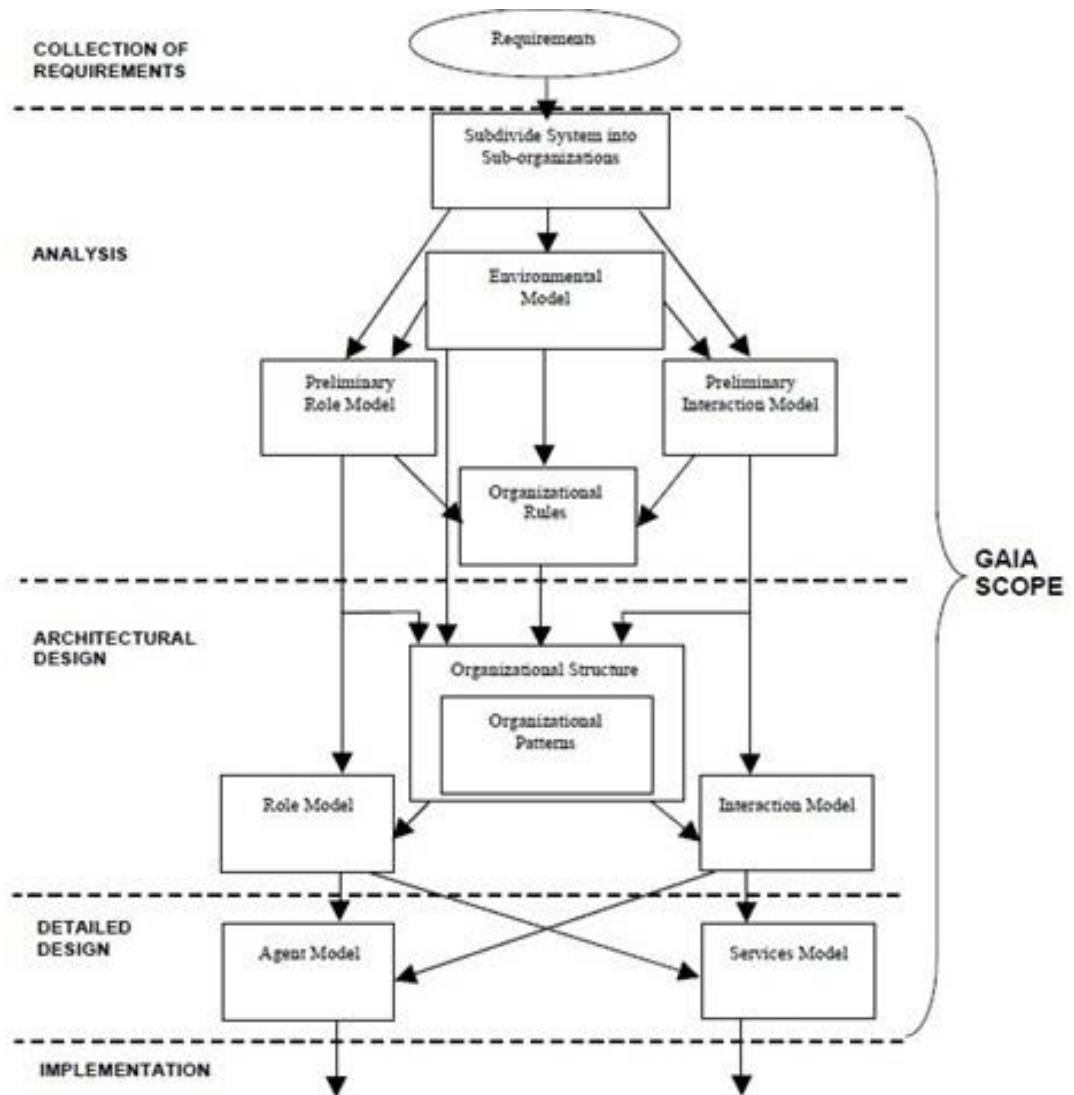
Fonte: Copiado de Zambonelli, Jennings e Wooldridge(2003).

### 2.3.1. Estrutura de Gaia

GAIA se divide em duas fases principais, Fase de Análise e Fase de Projeto. Sendo a fase de projeto subdividida em Projeto de Arquitetura e Projeto de Designer. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Durante essas fases, serão definidos vários modelos descrevendo os elementos do SMA. Esses modelos irão ser aperfeiçoados conforme o andamento do processo. Partindo de modelos conceituais até chegar em modelos mais concretos, que possam ser implementados e modelados usando alguma linguagem de programação e modelagem. A figura 2 abaixo mostra a estrutura das fases e atividades da metodologia e como elas se relacionam entre si.

Figura 2: Representação dos modelos e fases da metodologia Gaia e suas relações.



Fonte: Copiado de Zambonelli, Jennings e Wooldridge(2003).

### 2.3.1.1. Fase de Análise

Na Fase de Análise, Gaia se preocupa em entender o que será o sistema. Entender quais serão seus componentes e como será sua estrutura. Sem levar em consideração qualquer aspecto de implementação. Analisar e documentar tudo que for básico para modelar o sistema como uma estrutura organizacional. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). Para isso, esta fase inclui:

- Identificar os atores e demais componentes do sistema para identificar as organizações do sistema.
- Identificar os recursos do sistema. Como serão usados e por quem serão usados. Para criação de um modelo que represente o ambiente onde será instalado o SMA.
- Criação de um modelo preliminar de papéis, identificando as habilidades básicas necessárias pelo sistema.
- Criação de um modelo preliminar de interações, identificando as relações básicas entre os papéis no sistema, suas dependências e seus objetivos.
- Definição das regras organizacionais que os agentes terão que seguir dentro do ambiente do sistema.

A metodologia começa com a atividade de Identificar as Organizações do Sistema e seus respectivos atores, o que irá ajudar a dividir a complexidade do sistema em conjuntos menores.

A metodologia define que essa tarefa seja realizada logo nas etapas iniciais da metodologia por tornar a modelagem das interações do ambiente mais simples e também por facilitar a divisão das tarefas entre os membros do ambiente (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). Nessa atividade será possível ainda identificar os papéis principais que irão compor o sistema, principalmente se for levado em consideração que os papéis que participarem de uma mesma organização tendem a ter objetivos parecidos.

Após identificar as organizações do sistema, Gaia sugere a criação de um modelo que descreva o ambiente onde o sistema será implantado. Esse modelo é chamado de Modelo de Ambiente. Este irá conter uma descrição dos recursos do sistema, informando, por exemplo, quais recursos estariam disponíveis para os agentes e o que cada agente poderá fazer com esses recursos (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). Definir este modelo será importante para informar o tipo de ambiente que o sistema será implantado.

Ainda na Fase de Análise teremos uma atividade para identificar os papéis básicos do sistema. Esses papéis são identificados após uma análise pouco detalhada sobre a documentação dos requisitos (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). Sobre essa atividade é importante observarmos que a documentação de especificação do sistema será proveniente de uma fase do desenvolvimento que Gaia considera que já foi realizada, a fase de levantamento de requisitos. Isso foi um ponto importante a ser observado para a criação

do processo proposto nesse trabalho, pois o objetivo aqui é propor um processo contendo todas suas fases.

Após definir os papéis iniciais do sistema, é possível capturar as relações entre os vários papéis na organização, com suas respectivas dependências. Isso será importante para definir os protocolos para cada tipo de interação entre os papéis. Essas informações serão documentadas no modelo preliminar de interações, onde Gaia defini alguns atributos importantes como padrão para a definição de um protocolo de interação. Como por exemplo, quem inicia o protocolo (que será o papel responsável em iniciar a interação) e para quem se destina a interação (papel responsável pela resposta ao iniciador) (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Com relação às regras organizacionais do sistema, Gaia defende que a identificação prévia das regras do sistema irá ajudar na definição da estrutura do sistema na atividade posterior, pois com as regras já definidas, o designer poderá formar uma estrutura que funcione de forma coerente e que também seja eficiente (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

### **2.3.1.2. Fase de Projeto**

Após a Fase de Análise inicia-se a Fase de Projeto. É nessa fase que decisões serão tomadas sobre as características reais do sistema. Ela pode ser subdividida em Projeto de Arquitetura e Projeto de Designer. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

A Fase de Projeto irá definir como será o funcionamento real do sistema e terá como entrada os resultados obtidos na Fase de Análise, onde as funções e características do sistema são identificadas.

É também na fase de projeto que serão terminados os modelos de papéis e de interações, iniciados na Fase de Análise. Esses modelos só podem ser terminados após a definição de como será a estrutura do sistema, pois dependem de decisões que serão tomadas sobre ela. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

O Projeto de Arquitetura inclui:

- *A definição e representação da estrutura organizacional do sistema.* Visa definir como serão organizado e explorado os recursos do sistema levando-se em conta três fatores principais: (I) a eficiência organizacional, (II) a organização do mundo



real que o SMA se baseia, (III) a necessidade de cumprir as regras organizacionais (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

- . *A conclusão dos modelos de papéis e de interações.* Visa completar os modelos preliminares da fase de análise com informações sobre a estrutura do sistema (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

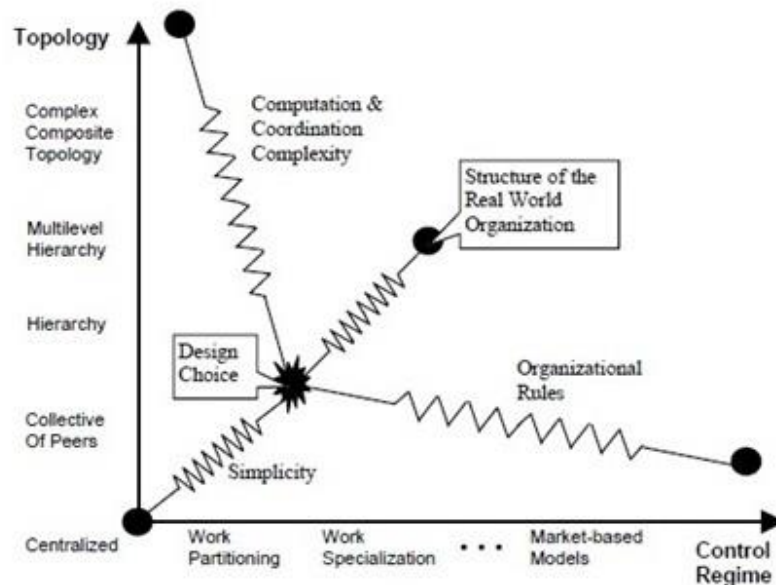
Com relação aos fatores citados acima que podem ajudar a definir a estrutura organizacional, Gaia defende a ideia do dividir para conquistar, levando em consideração que a estrutura deve ser a mais simples possível sem deixar de ser produtiva (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Outro fator que deve ser levado em consideração na escolha de uma estrutura organizacional é a capacidade do sistema de seguir as regras do ambiente durante sua execução. Um sistema muito restrito pode acarretar em perda de representatividade da autonomia dos agentes no sistema, pois irá limitar a maneira que seus membros irão se comportar no ambiente. Em contrapartida, um sistema aberto, onde os membros são movidos por objetivos próprios e podem querer se comportar da maneira que acharem melhor, pode ocasionar em maior trabalho de coordenação do ambiente.

Desse modo, a estrutura do sistema deve ser projetada para estar preparada com eventos não programados dentro do ambiente, mas sem ter que restringi-los ao ponto de perder a autonomia dos agentes (OLIVEIRA, 2013). Assim, a imposição de regras deverá ser ajustada conforme a necessidade de distribuir as tarefas na organização, mas sempre levando em conta as restrições do ambiente.

Com relação a influência das organizações do mundo real na definição da estrutura de um SMA, o fato de identificar essas semelhanças entre as duas estruturas pode ajudar a revelar inconsistências no desenvolvimento, pois a estrutura de um SMA normalmente é projetada baseando-se em alguma organização do mundo real. (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

Figura 3: Representação das forças que influenciam a identificação da estrutura organizacional.



(Fonte: Copiado de Zambonelli, Jennings e Wooldridge(2003).

Após a identificação da estrutura organizacional do sistema o designer deve decidir por representá-la da maneira mais legível possível. Normalmente, isso é feito integrando a documentação textual e seus modelos contendo uma representação gráfica (CABAC, 2003).

Em Zambonelli, Jennings e Wooldridge (2003) é sugerida a junção das duas representações, a textual e a gráfica, mas não menciona qual deve ser usada. Ficando a critério de quem utilizar a metodologia escolher como irá representar a estrutura do sistema em desenvolvimento.

Após o término da atividade para definir a estrutura do sistema, pode-se concluir os modelos de papéis e de interações. De maneira geral, depois de definir a estrutura do sistema o designer poderá identificar quais papéis irão interagir com que outros papéis e quais protocolos terão de ser executados.

Assim, o modelo completo de papéis e dos protocolos inclui:

- Definição de todos os papéis do sistema, documentando todas as atividades que um papel irá fazer no sistema e suas relações com os demais agentes (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003);

- Definição completa dos protocolos de interação dos agentes. Contendo também informações a respeito da estrutura organizacional (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

A fase de Projeto de Designer é responsável em definir os modelos de agentes do sistema e o modelo de serviços desses agentes. Abordando o que será necessário para cada agente exercer seu papel (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

- *A definição de um Modelo de Agente.* Visa identificar as classes de agentes para compor o sistema.
- *A definição de um Modelo de Serviços.* Identifica os serviços que serão prestados por cada classe de agente.

Sobre o modelo de agente proposto por Gaia, é importante destacar que deve ser usado na atribuição estática de papéis. Assim que um papel é assumido dentro do sistema, o agente não poderá mudá-lo (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003).

### **2.3.2. Limitações**

Este trabalho refere-se à nova versão de Gaia, também conhecida como Gaia 2.0, onde o modelo de papéis tornou-se mais completo, referenciando as regras do sistema e as interações que os agentes realizam no ambiente. Assim, é possível que ao final da fase de análise seja especificado modelos de agentes contendo todas as características dos protocolos de interação que os agentes participam.

Após a execução do processo de Gaia, os desenvolvedores terão a disposição a documentação da modelagem de classes dos agentes, dos papéis, dos protocolos de interações e da estrutura que deverá compor o sistema.

De forma resumida, Gaia se preocupa mais em como a sociedade de agentes de um SMA irá cooperar para realizar suas atividades e alcançar seus objetivos, do que em como isso será feito por eles. A maneira que cada agente realiza um serviço fica além do contexto da metodologia e irá depender da aplicação.

Na versão anterior de Gaia, os resultados da Fase de Análise, por exemplo, era um modelo de papéis sem qualquer referência sobre as regras do sistema, tornando a metodologia

inadequada para controlar o comportamento espontâneo de agentes externos (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003)..

Durante o estudo da metodologia Gaia foi possível percebermos que mesmo depois dela ter sofrido algumas alterações, ela continua sendo uma metodologia limitada quando aplicada no desenvolvimento de sistemas multiagentes abertos. Dentre elas:

- Dificuldade em tratar eventos não previstos no ambiente, pois assume que todos os agentes são cooperativos e que vão se comportar de acordo com o especificado (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). O que nem sempre acontece com agentes externos;
- Não utiliza nenhuma notação gráfica, apesar de defender o uso, para representar a ligação entre os modelos de ambiente, modelo preliminar de papéis e de interações. Fazer isso somente com documentos textuais pode ser uma tarefa difícil tanto pra ser executada, como pra ser compreendida pelos desenvolvedores;
- Não lida diretamente com a parte de implementação do sistema. O resultado do processo de Gaia é uma especificação detalhada, mas sem muitos subsídios para ser diretamente implementada. Uma análise sobre quais tecnologias, adequadas ao paradigma de agentes, seria necessária antes de iniciar o desenvolvimento;
- Não lida diretamente também com questões de testes do sistema. Não menciona o procedimento de testes que os desenvolvedores poderiam utilizar para validar e verificar o sistema desenvolvido.

Concluimos então, que a versão atual de Gaia continua deficiente para tratar a questão da autonomia dos agentes e sobre o tratamento de eventos anômalos no ambiente. Podemos concluir também que Gaia continua mais adequada para sistemas estáticos, onde, por exemplo, as habilidades dos papéis são pré-definidas e a atribuição de papéis são feitas de forma estática.

#### **2.4. Institutional Environments**

Como já foi citado acima, um dos principais desafios na implementação de SMA abertos é como manter um bom desempenho do sistema como um todo sem ter que restringir

o comportamento dos agentes impondo-lhes regras de conduta e sem tirar sua autonomia. Em Oliveira (2013), é proposto um framework para desenvolvimento de sistemas multiagentes abertos, cujo principal objetivo é promover mecanismos que mantenham a autonomia dos agentes, sem que seja necessário obrigar-lhes a seguir a risca as regras do sistema. Esse framework foi chamado de *Institutional Environment*.

#### **2.4.1. Desenvolvimento de um Ambiente Institucional**

Podemos definir o ambiente de um sistema multiagentes como sendo uma organização composta por agentes, cujas atividades serão distribuídas entre os agentes que irão compor o sistema. Estes por sua vez, devem exercer papéis e interagir com o restante do sistema através de trocas de mensagens ou através da realização de tarefas. (OLIVEIRA, 2013).

Na tentativa de conseguir o desenvolvimento de SMAs mais abertos, aspectos de uma organização do mundo real podem ser movidos para modelar os SMAs. Para isso, uma abordagem institucional pode ser usada para modelar os sistemas.

Quando observarmos o ambiente de sistema como sendo um ambiente institucional, vemos um ambiente onde os objetivos podem ser delegados e onde os papéis estarão interligados, de forma que os agentes que ingressarem no sistema serão monitorados para que sigam as regras impostas pelo ambiente e para que obedeçam aos protocolos impostos (ROSA; COSER, 2005). Portanto, SMAs poderão ser modelados como um conjunto de papéis relacionados, onde os papéis são definidos em termos de obrigações, permissões e protocolos acrescidos de seus objetivos e de suas habilidades.

As organizações serão compostas de agentes de interface, cujo objetivo é se comunicar com agentes externos que queiram ingressar no sistema. E agentes internos, que são os agentes que já estão ingressos no sistema.

Essa interface do sistema deve responder a pedidos de informação dos agentes externos, fiscalizar as obrigações dos agentes internos e informar os eventos que acontecerem no sistema e que os agentes internos precisem saber.

Sobre o desenvolvimento do sistema é esperado que os agentes que ingressarem no ambiente obtenham o conhecimento necessário sobre o funcionamento da organização. Ou seja, devem saber exatamente o que devem fazer para cumprir suas atividades. Os agentes

externos, por sua vez, normalmente são desenvolvidos por terceiros e possuem pouco conhecimento sobre o ambiente em que irão trabalhar. É necessário então mecanismos para manter o controle de suas ações dentro do sistema.

#### **2.4.2. Instituições, Papéis e os Atos Institucionais**

No IE o conceito de papéis de agentes será abordado no momento que um agente externo ingressa no ambiente do sistema. Os papéis irão representar a função dos agentes dentro do sistema. Todo agente externo que ingressar no sistema, irá solicitar permissão a um agente representante da instituição do sistema, chamado de *Institutional Agent*. Este irá atribuir um papel do sistema ao novo agente. Logo, a partir do momento em que um papel é delegado a um agente, o agente herda as obrigações e permissões especificadas para o papel herdado (BRITO; GAVA; TAVARES; MENEZES, 2001).

Os Atos Institucionais representam as ações que os agentes realizam para interagir com o ambiente institucional. Essas ações serão definidas a partir das informações contidas nos protocolos de interações que ditam como deve ocorrer a interação entre os papéis.

Fazendo uma pequena analogia a sociedade humana, podemos comparar tais conceitos a um time de futebol. Um jogador (agente) que joga em um time (instituição) tem a habilidade de fazer os gols, portanto joga na posição de atacante (papel) e irá exercer seu papel chutando a bola para fazer os gols (ato institucional).

#### **2.4.3. Mecanismos de Controle Social**

Agentes externos ao ingressarem no sistema deverão de alguma forma garantir que irão cumprir com suas atividades. O sistema, por sua vez, deve de alguma forma exigir do agente recém ingresso o cumprimento dessas atividades. Isso pode acarretar na restrição das ações que um agente pode exercer dentro de um SMA aberto, prejudicando a sua autonomia e o seu grau de abstração de problemas do mundo real.

No momento que um agente ingressa no sistema, o mesmo concorda em seguir as regras impostas para interagir no ambiente. Para outras abordagens, essa imposição acaba gerando uma restrição no comportamento dos agentes, fazendo com que só possam se

comportar da maneira imposta. Para sistemas desenvolvidos usando o IE, essa imposição de regras é menos restritiva com o uso de compromissos entre os agentes. O agente também tem que seguir regras de comportamento do sistema, mas não necessariamente são obrigados a segui-las a risca. Eles têm a autonomia de decidir a forma com que iram realizar a tarefa especificada, podendo ser gratificados caso seja uma escolha positiva, ou punidos caso seja uma escolha negativa.

No IE, quando um agente externo ingressar no ambiente institucional, o mesmo terá que firmar um compromisso com o agente instituição e terá liberdade de exercer seu papel da maneira que achar mais viável, de acordo com suas permissões. O *monitor agent* (agente responsável em monitorar as ações dos agentes no sistema) é informado sobre o novo agente do sistema. Este irá conter informações sobre o compromisso firmado com o agente instituição no momento do ingresso no ambiente institucional e sobre os outros compromissos firmados com os demais agentes da instituição.

Oliveira (2013) acredita que o uso de compromissos torna o processo de definição e modelagem das regras de conduta dos SMAs abertos mais simples de serem representados. Compromissos podem ser entendidos como acordos firmados entre um devedor (agente externo) e um credor (instituição).

Apesar de possuir um agente regulamentador de tarefas, o agente instituição, o controle global do sistema é obtido com a cooperação mútua entre os agentes. Para isso é necessário que se construa um esquema de confiança entre os agentes para que eles possam interagir para atingirem um mesmo objetivo (OLIVEIRA, 2013). Em Castelfranchi (2000) métodos para implementar o controle social nos sistemas e distribuir a gestão do ambiente entre os agentes membros foram adaptadas para o domínio de SMA abertos.

Modelos de confiança e de reputação são então usados pelo IE para garantir o controle social e construir esse elo de confiança entre os agentes no SMA (OLIVEIRA, 2013). A análise feita sobre o cumprimento ou não dos compromissos firmados pelos agentes irá influenciar diretamente na reputação dos mesmos, pois irá informar se os agentes são confiáveis para assumirem ou dividirem responsabilidades com outros agentes dentro do sistema (OLIVEIRA, 2013).

Esses recursos usados no IE podem ser visto como um mecanismo de auxílio para garantir a aplicação das regras de conduta dentro do sistema.

O designer do sistema é quem irá delimitar no desenvolvimento do sistema o grau de liberdade que os agentes externos podem ter para que possam manter uma reputação boa e

honrar seu compromisso. Ele deverá especificar claramente quais serão os critérios utilizados para premiar ou punir os agentes com pontos de reputação. Especificar o que será feito e como será feito, caso seja identificado que um agente quebrou um compromisso ou cumpriu um compromisso.

Com o compromisso firmado, o agente institucional disponibiliza então um papel no sistema para o mais novo agente no SMA. Esses papéis são representados por *deputy agents* (agentes representantes). Todo agente externo registrado no sistema terá o seu *deputy* correspondente, assim como todo agente que deixar o sistema, terá seu *deputy* desalocado, e o *deputy* ficará então ocioso até o próximo agente externo que irá ser alocado. O agente monitor irá informar então ao agente reputação sobre o novo agente ingresso no sistema e este irá manter dados sobre a reputação do agente em questão. O agente reputação por sua vez, tem o papel de controlar o cálculo da reputação de cada agente registrado no sistema impondo as regras do sistema através de controle social, e considerá-las como critério de análise ao se punir ou aceitar um agente do sistema. A reputação de um agente será alterada a cada ação do agente. Se o comportamento do agente ser considerado aceitável pelo agente instituição, então sua reputação ganhará pontos positivos. Do contrário, se o agente tem um comportamento dito como inadequado pela instituição constantemente, isso fará sua reputação perder pontos, podendo isso prejudicar posteriormente a comunicação com os demais agentes.

#### **2.4.4. Métodos e Modelagem da Interação dos Agentes**

Dentro de uma instituição qualquer, a troca de informações e a interação deve ser restrita e de uma forma que seja entendida por todos os integrantes do ambiente (ROSA; COSER, 2005). No caso específico de um ambiente institucional de um SMA, as trocas de mensagens devem ser feitas de maneira que sigam um protocolo de comunicação onde cada membro pode se comunicar. Esse protocolo de comunicação visa estabelecer um padrão onde todos os agentes membros do sistema poderão se comunicar. Isso é necessário, pois um SMA aberto pode conter vários agentes desenvolvidos por terceiros, e ter seus próprios meios de comunicação e podem não compreender como se dá a comunicação entre os agentes do sistema (OLIVEIRA, 2013).

Algumas abordagens já foram desenvolvidas para padronizar a comunicação entre os agentes de um SMA. Entretanto, boa parte dos trabalhos desenvolvidos sobre esse assunto visam o comportamento próprio do agente, não em padronizar um comportamento no



ambiente mais geral do sistema. Em Esteva (2003), por exemplo, é feita uma abordagem mais focada no comportamento do agente, não no ambiente. Isso é mais adequado para sistemas fechados, pois a complexidade das atividades é menor. Para sistemas abertos, a tarefa de implementar métodos padrões de interação e de cooperação fica mais complexa justamente pelo fato de haver agentes desenvolvidos por terceiros (OLIVEIRA, 2013). Tal complexidade levou a modelagem de organizações de agentes.

Para se entender as diferentes formas com que os padrões de interação entre os agentes são definidos, analisemos as metodologias citadas na seção 2.2. Pode-se perceber que as principais preocupações são em relação a identificação de papéis que os agentes devem exercer e suas relações com o ambiente e com a atribuição de funcionalidades aos agentes.

Na obra de Esteva (2003), por exemplo, Máquinas de Estado Finito são utilizadas para definir os protocolos de interação. Essa modelagem caracteriza-se por representar sistemas sequenciais, onde cada processo de interação entre os agentes é um estado. Esse tipo de modelagem tem dificuldade em representar a comunicação concorrente entre agentes.

Já nas metodologias de Tropos e Prometheus os protocolos de interação são modelados usando AUML, que é uma linguagem estendida da UML, mas que visa descrever os aspectos relacionados às especificações dos agentes. Pode-se usar, por exemplo, diagramas de caso de uso semelhantes ao usado na UML, para capturar requisitos do sistema e diagramas de interação para modelar as interações entre os agentes e com o próprio sistema (CABAC, 2003).

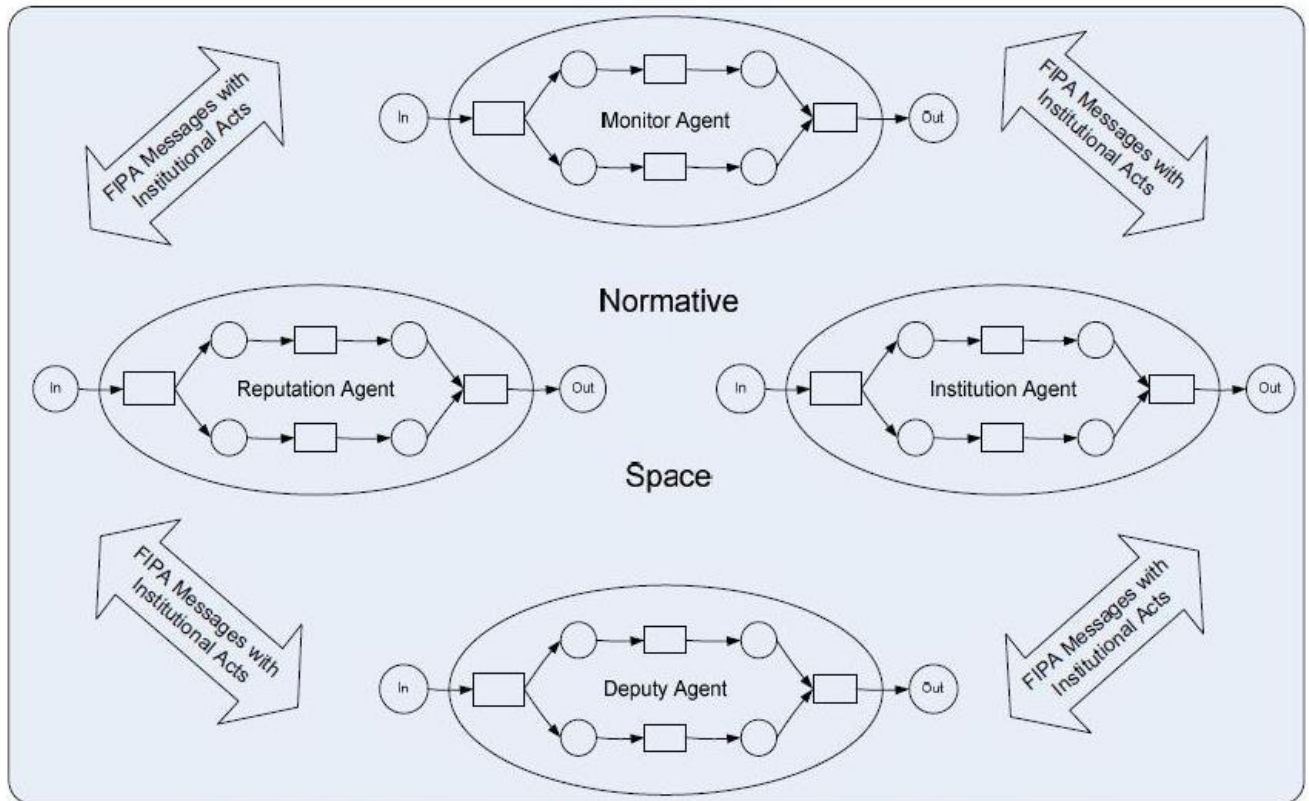
Já em Oliveira (2013), é utilizado Redes de Petri Coloridas (CPN). CPN é uma ferramenta utilizada para representar graficamente os papéis e as interações entre os mesmos em um SMA Aberto, de forma que seja fácil de compreender as relações entre eles. CPN também é uma alternativa para solucionar o problema de representar interações concorrentes ou paralelas entre as entidades do sistema. No IE, as CPNs formam um ambiente distribuído chamado de espaço normativo, representado na Figura 4 (OLIVEIRA, 2013).

Com o uso das CPNs no IE, conseguiu-se uma maior flexibilidade ao representar fluxos de trabalho no Ambiente Institucional. Oliveira (2013) acredita que o uso de CPNs, oferece uma ferramenta mais adequada para o desenvolvimento de ambientes institucionais com diferentes níveis de abstração, tornando os SMAs mais fáceis de serem analisados como espaços normativos.

Para modelar os protocolos de comunicação, e as trocas de mensagens foram utilizadas mensagens baseadas em FIPA - ACL. FIPA (*Foundation for Intelligents Physical*

*Agents*) é uma entidade que vêm especificando padrões, principalmente padrões de comunicação para o desenvolvimento de sistemas multiagentes. O termo ACL vem do inglês *Agent Communication Languages*, ou seja, é uma linguagem de comunicação para as trocas de mensagens entre os agentes de um SMA.

Figura 4 - Espaço Normativo do Institutional Environment



Fonte: Copiado de Oliveira (2013).

## 2.5. Conclusão

A seguir, destacamos os principais pontos de semelhanças e de divergências entre o contexto de Gaia e do IE, diferenciando como uma mesma característica de SMAs é abordada nas duas ferramentas e como foram adaptadas para a criação do processo.

A abordagem defendida em Oliveira (2013) aborda algumas características defendidas na metodologia Gaia, ao mesmo tempo em que propõe melhorias para quando ela for utilizada para desenvolver sistemas abertos. Defende a modelagem baseada em instituição e atribuição de papéis e aborda também a questão de como o comportamento dos agentes irá afetar o desempenho do sistema.

Gaia não foi projetada para lidar com mudanças na habilidade social e nas atividades desempenhadas pelos agentes em tempo de execução, ou seja, não trata a questão da autonomia dos agentes. Portanto, é esperado que os agentes que ingressarem no sistema sempre irão se comportar de acordo como foi especificado para o papel que assumirem no sistema. Resumidamente, não está preparada para lidar com eventos anômalos ao seu contexto e nem com as diferenças de ideais de agentes desenvolvidos por terceiros no sistema.

Os agentes externos deverão de alguma forma garantir que irão cumprir com suas atividades quando ingressarem no sistema. O sistema, por sua vez, deve de alguma forma exigir do agente recém ingresso o cumprimento dessas atividades.

Em Gaia, isso é resolvido com a definição das regras organizacionais do sistema, que ditam como os agentes devem se comportar ao assumir determinado papel. O problema é que isso pode acarretar na restrição das ações que um agente pode exercer dentro de um SMA aberto, prejudicando a percepção do agente do mundo real.

Para tratar essa questão o IE utiliza o conceito de compromissos entre os agentes e entre a instituição. Oliveira (2013) acredita que o uso de compromissos torna o processo de definição e modelagem das regras de conduta nos SMAs abertos mais simples de serem representados. Compromissos podem ser entendidos como acordos firmados entre um devedor (agente externo) e um credor (instituição ou até mesmo outros agentes).

É importante destacar que no IE os agentes poderão realizar atividades pré-definidas e atividades definidas em tempo de execução. Assim sendo, poderão também criar compromissos em tempo de execução para realizar suas tarefas e contatar outros agentes registrados na instituição.

Com isso, a imposição das regras nos sistemas desenvolvidos usando o IE é menos restritiva. O agente também terá que cumprir regras de comportamento do sistema, no caso, os compromissos, mas não necessariamente é obrigado a segui-las a risca. Eles têm a autonomia de decidir a forma com que irão realizar sua tarefa, podendo ser gratificados caso seja uma boa escolha, ou punidos caso seja uma escolha ruim.

Isso é feito atribuindo-se permissões aos papéis que os agentes irão assumir. Essas permissões irão manter a autonomia dos agentes, sem ter que restringir demais o seu comportamento. Dando-lhes a possibilidade de criação de novos atos institucionais e, conseqüentemente, de novos compromissos, os agentes poderão escolher a melhor maneira de agir no sistema.

O IE defende a ideia de que usando uma abordagem institucional, é possível desenvolver um SMA de modo a tornar fácil o gerenciamento do sistema devido a implementação de formas de recompensar ou punir os agentes, tendo como base o seus comportamentos e sua reputação. Para isso, utiliza o conceito de agentes monitores e de reputação para manter um ambiente estável no sistema, mesmo contendo divergências de ideais entre seus agentes.

A facilidade de gerenciamento fica mais evidente quando se pretende desenvolver um sistema aberto onde os agentes participantes têm alto grau de autonomia. Assim sendo, a utilização do IE como framework para o desenvolvimento de SMAs pode ser defendida pelas seguintes características:

- Proporciona o desenvolvimento de SMA aberto, de forma que se mantenha um desempenho aceitável do sistema sem limitar o comportamento dos seus agentes.
- Proporciona ao sistema desenvolvido uma maneira de lidar com a heterogeneidade do sistema, pois o mesmo pode ser formado por agentes desenvolvidos por terceiros.
- O uso de CPNs para representar a interação entre os agentes. Isso possibilita que a troca de mensagens entre os agentes pode ser feita de maneira concorrente entre vários agentes. Diferentemente de outras abordagens que utilizam Máquinas de Estado Finitos, onde as interações são representadas de maneira sequencial.

Assim sendo, o desenvolvimento de um processo para o framework IE juntamente com os conceitos já defendidos por Gaia para sistemas abertos, explora pontos fracos da metodologia, podendo ser visto como uma contribuição para a aplicação de Gaia em sistemas abertos, visto que obtemos uma melhoria na metodologia a partir do controle social proposto pelo IE. Ampliando a gama de aplicações em que essa metodologia pode ser aplicada. Mais precisamente em sistemas abertos.

### **3. ETAPAS PARA A DEFINIÇÃO DO PROCESSO DO INSTITUTIONAL ENVIRONMENTS**

Esta seção visa apresentar todas as etapas para a definição do processo do IE. Esta composta por seis subseções.

### 3.1. Primeira Etapa

A primeira etapa teve início com um estudo sobre as técnicas utilizadas para desenvolvimento de SMAs a fim de compará-las com um processo de desenvolvimento de sistemas orientados a objetos. Para isso foram utilizadas como fonte de pesquisa obras de outros autores que abordam o tema SMAs. Por exemplo, podemos citar as obras de Esteva (2003), de Brafman (1996), Castelfranchi (2000) e Arcos *et al.* (2005).

Foram identificadas diferenças no desenvolvimento de sistemas orientados à agentes com os orientados à objetos juntamente com características típicas de SMAs que devem ser exploradas.

Podemos observar na tabela abaixo como agentes e objetos se diferenciam com relação ao seu comportamento, estado e relacionamento com os demais membros do sistema.

Tabela 1: Diferença entre sistemas baseados em agentes e sistemas baseados em objetos

	<b>Estado</b>	<b>Comportamento</b>	<b>Relacionamento</b>
<b>Sistemas Baseados em Agentes</b>	Para representar o estado de um agente podemos dizer que ele é formado por suas <i>Crenças, Objetivos, Planos e Ações</i> . Onde as crenças será composta por todo o conhecimento que o agente possui, os objetivos serão os desejos que o agente quer satisfazer, as ações serão o que um agente poderá fazer e seus planos será como deve fazer.	O comportamento dos agentes será a própria execução dos seus planos de ações. Cada plano possui um objetivo e cada objetivo tem uma prioridade.	O relacionamento entre os agentes de um sistema se dá através de troca de mensagens. Para isso, utilizam de uma linguagem padrão para manter o entendimento. E ao receber uma mensagem, um agente pode escolher por não respondê-la.
<b>Sistemas Baseados em Objetos</b>	O estado dos objetos será representado por seus atributos. Estes irão armazenar informações sobre o objeto e sobre as demais entidades do sistema.	O comportamento dos objetos será representado por seus métodos. São tarefas que podem ser executadas e que podem modificar o estado do objeto.	O relacionamento entre objetos é representado por chamadas de métodos. Sempre que um objeto recebe uma chamada de método, este irá responder após a execução do método.

Fonte: elaborado pelo autor

### 3.2. Segunda Etapa

A segunda etapa teve início com um estudo das diferentes metodologias existentes para o desenvolvimento de SMAs. Esse estudo foi necessário para que pudéssemos entender como adequar o desenvolvimento de sistemas para determinados tipos de ambientes a uma dada metodologia. E como uma metodologia específica tratava características próprias de SMAs. A tabela mostra como as diferentes metodologias estudadas abordam esses pontos.

Tabela 2: Principais diferenças entre as metodologias para sistemas multiagentes

	<b>Gaia</b>	<b>Prometheus</b>	<b>Tropos</b>	<b>Zeus</b>
<b>Características de Domínio</b>	Pode ser aplicada em vários domínios.	Assim como GAIA, pode ser aplicada em vários domínios.	Diferentemente das demais que focam no projeto, TROPOS foca na análise de requisitos.	É direcionada para desenvolvimento utilizando a ferramenta Zeus. E assim como Gaia os papéis trabalham para realizar suas responsabilidades.
<b>Papéis</b>	A determinação de papéis é feita até a fase de projeto.	Funcionalidades são atribuídas aos agentes na fase de projeto de arquitetura.	Define atividades para todo o processo de desenvolvimento tendo como base os requisitos coletados na fase inicial do projeto.	Sugerem que os papéis devem ser modulares, possuir alta coesão, serem completos, e ter pouca dependência entre papéis.
<b>Configuração Social</b>	Baseia-se na visão do sistema como uma organização computacional que consiste na interação de vários papéis.	Obtém um conjunto de agentes que se comportem no sistema para alcançar um objetivo definido, e que interagem quando for necessário seguindo os protocolos de comunicação.	A construção do MAS é dividido em cinco fases: Requisitos Iniciais, Pós Requisitos, Projeto Arquitetônico, Projeto Detalhado e Implementação	Assim com Gaia, baseia-se na visão do sistema como uma organização computacional que consiste na interação de vários papéis, mas que possuam pouca dependência.

<b>Métodos de controle</b>	Não trata objetivos conflitantes entre os agentes.	São modelados usando AUML.	Assim como Prometheus é usado a linguagem de modelagem AUML.	Oferece suporte de ferramentas geradoras de agentes presente em Zeus, que permite a geração de partes dos agentes em código Java e também mecanismos de testes dos agentes. Permitindo a aplicação direta da metodologia.
<b>Aplicação</b>	Tem o foco no comportamento do agente e na organização dos agentes no ambiente do sistema.	Não é adequada para sistemas abertos por não ter uma noção de organização.	Também não se preocupa com noção de organização e tem uma abordagem focada no comportamento do agente. Não adequada para SMAs abertos.	Também é centrada no agente, impõe restrições rígidas sobre o comportamento dos mesmos dentro do sistema. O que a torna inadequada para SMAs abertos

Fonte: elaborado pelo autor

### 3.3. Terceira Etapa

A terceira fase teve início com o estudo do *framework Institutional Environments*, onde identificamos como essa ferramenta aborda o desenvolvimento de SMAs de forma a se tornar uma alternativa para os problemas enfrentados por sistemas abertos.

Destacamos na seção 2.3 deste trabalho os principais aspectos do IE relacionados aos principais problemas enfrentados no desenvolvimento de SMAs abertos, também citados com mais detalhes na seção 2.1. Logo após fizemos a relação de como cada aspecto identificado na ferramenta se relacionava com as metodologias estudadas para dar continuidade ao trabalho no passo seguinte.

### 3.4. Quarta Etapa

A partir dos resultados obtidos com o estudo de diferentes metodologias existentes e do estudo do framework IE, podemos analisar e escolher qual metodologia seria

mais adequada para adaptarmos seus passos para a criação do processo. Assim sendo, a quarta etapa consistiu na escolha dessa metodologia.

Podemos relacionar abaixo os principais aspectos que nos levou a escolher Gaia como metodologia base para criação do processo:

- Os agentes devem exercer papéis para se relacionarem com o ambiente do sistema, algo que seria adequado para manter o controle social imposto pelo IE. Essa característica ainda sofreu uma pequena influência da metodologia Tropos, onde a definição de papéis é feita a partir da definição do sistema. Portanto, utilizamos a ideia de atribuição de papéis proposta em Gaia, mas que seja feita desde os estágios iniciais do desenvolvimento.
- Pode ser aplicada em qualquer domínio de aplicação
- Foca na visão do projeto do sistema como sendo uma organização computacional que pode ser decomposta em papéis e no comportamento dos agentes que representam esses papéis.
- Não especifica ferramentas ou linguagens específicas para representação gráfica da modelagem, deixando para livre escolha.
- Não lida com objetivos conflitantes entre os agentes. O que torna interessante aplicar o contexto do IE para resolução desse problema.
- Apesar de não ser adequada para SMAs abertos, acreditamos que adaptações feitas para o processo podem ser úteis para que torne a metodologia adequada a esse tipo de sistemas.

### **3.5. Quinta Etapa**

Na quinta etapa começamos a definir o processo, tendo como entrada os resultados obtidos nas etapas anteriores. Para isso, utilizamos a ferramenta EPF Composer (ECLIPSE FOUNDATION, 2011), onde conseguimos representar de maneira gráfica todas as atividades do processo.

A partir daí, foram desenvolvidas diversas versões do processo, sempre acrescentando ou excluindo pontos. A primeira versão estava muito baseada na definição de Gaia. Foi então preciso adaptar as atividades a maneira como é abordada no IE. A primeira



versão do processo deveria apresentar uma sequência de passos onde fossem explorados todos os diferenciais do IE.

A partir daí, foram incluídas as fases de levantamento de requisitos, de implementação das CPNs e de testes das CPNs. Isso foi necessário porque este trabalho tem o objetivo de apresentar como seria o desenvolvimento de um SMA utilizando o IE em todas suas fases. Algo que não estava sendo mostrado em versões anteriores e nem fazia parte do contexto apresentado em Gaia. Após várias versões, chegou-se a uma versão final do processo que será apresentado em detalhes na seção 5.

### **3.6. Sexta Etapa**

A sexta e última etapa deste trabalho consistiu na demonstração de uso do processo, onde o aplicamos para modelar o sistema do Mercado Livre.

Inicialmente foram identificados todos os aspectos que seriam importantes definir de antemão sobre o Mercado Livre, seguindo as atividades propostas no processo. Logo após, fizemos a representação gráfica das CPNs que iriam representar os papéis do sistema, juntamente com o fluxo de trabalho de cada ato institucional dos papéis. Para isso utilizamos diagramas de atividades.

## **4. PROCESSO DO INSTITUTIONAL ENVIRONMENTS**

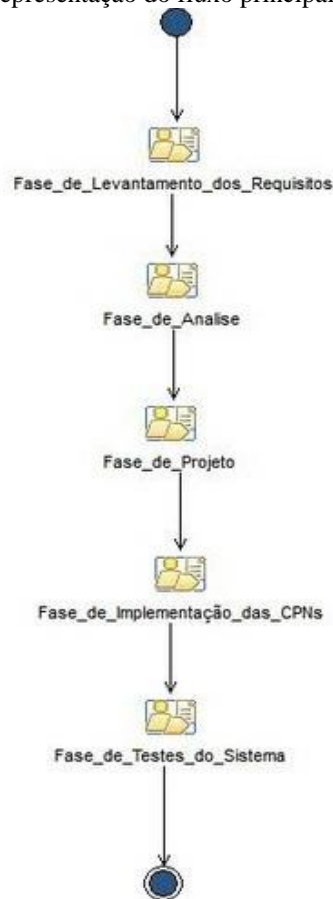
Esta seção tem como objetivo apresentar o processo desenvolvido, descrevendo todas suas atividades e passos necessários.

Para criação do processo do *Institutional Environments* foram aproveitados muitos conceitos que estão definidos em Gaia e que foram apresentados no capítulo 2. Com isso, conseguimos adequar o processo ao contexto do IE fazendo algumas adaptações.

O processo foi modelado utilizando a ferramenta EPF Composer. A figura 4 apresenta o fluxo principal do processo com todas as suas fases principais.

Logo após, serão apresentadas detalhadamente cada uma das fases juntamente com suas atividades.

Figura 5: Representação do fluxo principal do processo.



Fonte: elaborado pelo autor.

A figura acima representa o fluxo principal do processo contendo as fases principais. O processo se inicia com o levantamento dos requisitos do sistema. É nessa fase que será criado o documento de requisitos que será importante para o restante do processo. Após essa fase, terá início a Fase de Análise cujo principal objetivo foi mantido como está definido em Gaia. Trata-se de organizar as informações iniciais sobre o sistema para que possa ser entendido com mais facilidade. Terminada a Fase de Análise, terá início a Fase de Projeto, onde será definido como as coisas irão funcionar no sistema. Enquanto que na Fase de Análise é definido o que o sistema terá, é na Fase de Projeto que será definido como elas vão funcionar. Logo após a Fase de Projeto, irá iniciar a Fase de Implementação das CPNs. Nela as Redes de Petri serão codificadas de acordo como foram definidas na Fase de Projeto. Por fim, temos a Fase de Testes do sistema.

Nas próximas subseções, serão apresentadas detalhadamente cada uma das fases juntamente com tabelas detalhando cada uma das atividades, onde:

**Objetivo:**

- Visa definir qual será o objetivo de cada atividade descrevendo os resultados esperados.

<b>Papéis principais:</b> - Visa definir quais <i>stakeholders</i> são essenciais para realizar a atividade.	<b>Papéis Secundários:</b> - Visa definir quais <i>stakeholders</i> podem dar auxílio para realizar a atividade.
<b>Entradas:</b> - Visa definir quais artefatos servirão como entrada para realizar a tarefa.	<b>Saída:</b> - Visa definir quais artefatos serão gerados após a realização da atividade.
<b>Passos:</b> - Visa definir quais serão os passos necessários para realizar a atividade.	

Fonte: elaborado pelo autor.

#### 4.1. Fase de Levantamento dos Requisitos

A fase de levantamento de requisitos é o ponto inicial do processo proposto, sendo que esta não baseou-se em Gaia, uma vez que Gaia não trata a definição dos requisitos do sistema. O levantamento de requisitos do processo proposto foi definido a fim de atribuir uma atividade para a especificação do sistema a ser desenvolvido, pois de acordo com técnicas da Engenharia de Software todo processo de desenvolvimento de software tem início como a fase de levantamento dos requisitos.

Ela está dividida em duas tarefas, como pode ser observado na figura 5, sendo elas a tarefa para Elicitação e Especificação dos Requisitos e outra tarefa para a validação dos requisitos junto ao cliente.

Figura 6: Representação do fluxo das tarefas no levantamento de requisitos.



Fonte: elaborado pelo autor.

#### 4.1.1. Elicitação e Especificação dos Requisitos

Esta tarefa visa identificar quais serão os requisitos do sistema a ser desenvolvido. As informações sobre esta tarefa estão na Tabela 3 abaixo.

Tabela 3: Representação da tarefa Elicitação e Especificação dos Requisitos

<b>Objetivo:</b> - Esta tarefa consiste em identificar e documentar os requisitos do sistema a ser desenvolvido.	
<b>Papéis principais:</b> - Analista de requisitos	<b>Papéis Secundários:</b> - Cliente
<b>Entradas:</b> -	<b>Saída:</b> - Documento de Especificação dos Requisitos (APÊNDICE J)
<b>Passos:</b> - Identificar os Requisitos - Documentar os Requisitos	

Fonte: elaborado pelo autor.

A seguir, serão detalhados os passos para a execução da tarefa de *Elicitar Requisitos*.

**Passo 1 - Identificar os Requisitos:** O analista de requisitos irá fazer um levantamento dos requisitos através de reuniões e entrevistas com os *stakeholders* para posteriormente documentá-los.

**Passo 2 - Documentar os Requisitos:** Após identificar os requisitos, o analista de requisitos irá criar o Documento de Especificação dos Requisitos. Este deverá conter todos os requisitos identificados no passo anterior com uma descrição clara de seus objetivos.

#### 4.1.2. Validar os Requisitos

A partir da identificação dos requisitos iniciais do sistema, será preciso fazer sua validação junto ao cliente para verificarem a relevância de cada requisito. Após essa validação, será necessário atualizar a versão do documento de requisitos, caso haja alguma mudança por parte do cliente.

Tabela 4: Representação da tarefa Validar Requisitos

<b>Objetivo:</b> - Esta tarefa objetiva validar os requisitos identificados na tarefa anterior.	
<b>Papéis principais:</b> -Analista de Requisitos	<b>Papéis Secundários:</b> -Cliente, Desenvolvedores

<b>Entradas:</b> - Documento de Especificação dos Requisitos (APÊNDICE J)	<b>Saída:</b> - Documento de Especificação dos Requisitos (APÊNDICE J)
<b>Passos:</b> - Validação dos Requisitos - Atualizar documento de especificação dos requisitos	

Fonte: elaborado pelo autor

A seguir, serão detalhados os passos para a execução da tarefa de *Validação dos Requisitos*.

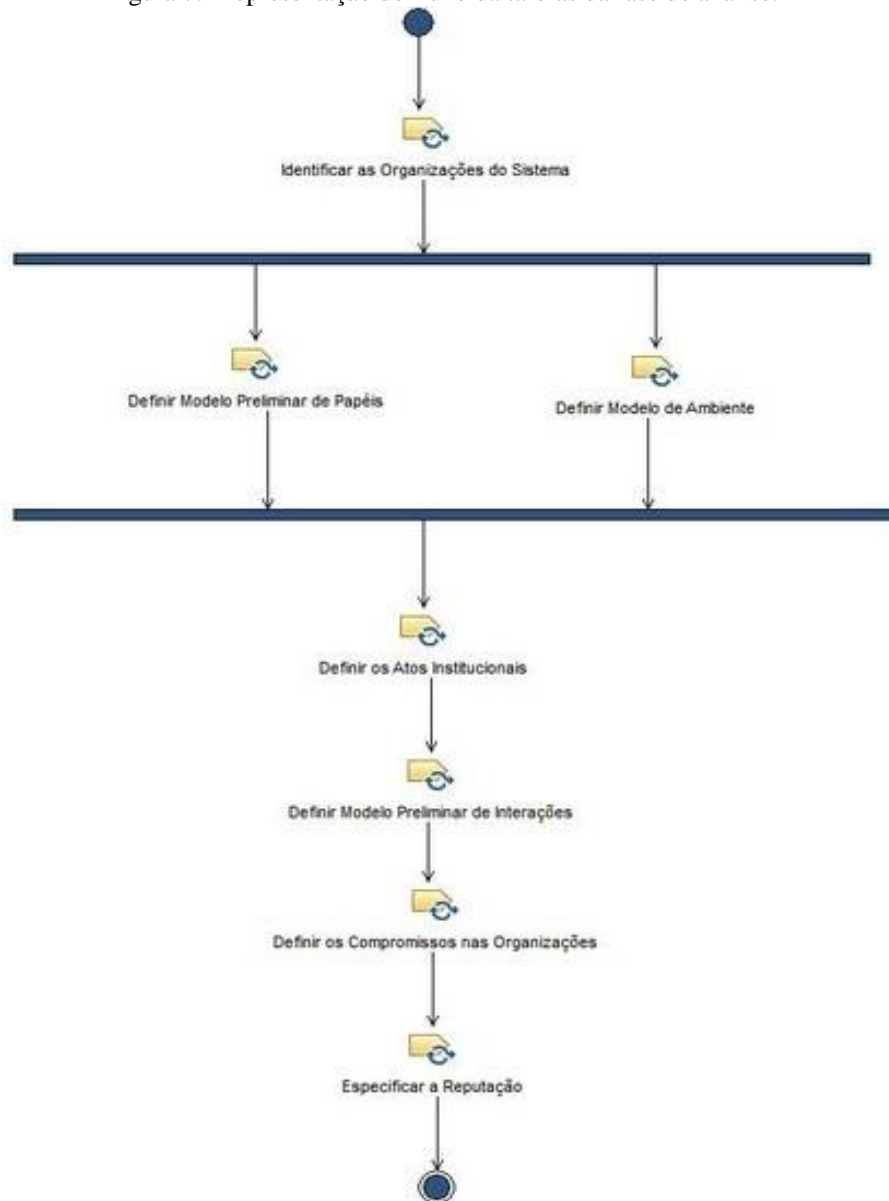
**Passo 1 - Validação dos Requisitos:** Os requisitos contidos no Documento de Especificação dos Requisitos devem ser validados pelo cliente. Dessa forma, o cliente poderá adicionar, modificar ou até mesmo excluir requisitos.

**Passo 2 - Atualizar documento de especificação dos requisitos:** Após o cliente validar os requisitos contidos no documento, o mesmo deverá ser atualizado para uma versão final contendo todas as modificações feitas pelo cliente, caso existam.

#### 4.2. Fase de Análise

Na fase de análise o objetivo é coletar e organizar as informações a respeito do sistema. A figura 6 apresenta o fluxo das tarefas da fase de análise.

Figura 7: Representação do fluxo da tarefas da fase de análise.



Fonte: elaborado pelo autor.

Nesta fase foram adicionadas algumas tarefas além das que já constavam em Gaia (identificação das organizações do sistema, definir modelos preliminares de papéis e interações e documentar as regras organizacionais) para representar o controle social imposto pelo IE. São elas: definir os atos institucionais, definir os compromissos e definir critérios para a reputação dos agentes. Portanto, os modelos dos artefatos gerados nas atividades dessa fase foram aproveitados das atividades de Gaia, sendo criados apenas os modelos das atividades novas.

De forma resumida, essa fase visa identificar o que o sistema terá que ser. Isso inclui a identificação:

- Dos objetivos das organizações que constituem o sistema e seu esperado comportamento.
- Dos papéis que serão necessários ao sistema.
- De uma representação do ambiente em que o SMA ficará situado.
- Definição das ações que os agentes realizaram dentro do sistema.
- Das interações básicas que serão necessárias dentro do sistema.
- Dos compromissos que os agentes devem assumir em seu comportamento global.
- Das informações que serão levadas em consideração na hora de atribuir dados de reputação aos agentes.

Vale ressaltar também que nessa fase incluímos uma pequena característica da metodologia Tropos, que diz que a identificação dos elementos que irão compor o sistema (organizações, papéis, interações) deverão ser identificados com base na definição inicial do sistema. Ou seja, a partir dos documentos gerados na fase de levantamento de requisitos de um sistema.

#### 4.2.1. Identificar As Organizações Do Sistema

Essa atividade não sofreu adaptações ao processo. Portanto, mantemos os mesmos objetivos definidos em Gaia.

Seu objetivo principal é identificar as diversas organizações que poderão compor o sistema, dividindo a complexidade do problema em conjuntos menores e melhor manejáveis.

Tabela 5: Representação da tarefa Identificar as Organizações do Sistema

<b>Objetivo:</b> - Visa identificar as diversas organizações que poderão compor o sistema.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entradas:</b> - Documento de especificação dos requisitos. (APÊNDICE J)	<b>Saída:</b> - Relatório com as organizações do sistema e seus objetivos. (APÊNDICE H)
<b>Passos:</b> - Analisar documentos de especificação do sistema - Verificar organizações que já foram identificadas no processo de definição dos requisitos. - Verificar semelhanças do sistema com alguma organização do mundo real - Documentar as organizações identificadas.	

Fonte: elaborado pelo autor

A seguir, serão detalhados os passos para cada passo para a Identificação das Organizações do Sistema.

**Passo 1 - Analisar documentos de especificação do sistema:** Visa analisar os documentos de especificação do sistema a fim de identificar informações relevantes para a definição das organizações. Por exemplo: Documento de Caso de Uso, Documento de Requisitos.

**Passo 2 - Verificar organizações que já foram identificadas no processo de definição:** Pode acontecer que durante o processo de especificação do sistema, algumas organizações que tenham sido identificadas. Esse passo visa documentá-las mais detalhadamente.

**Passo 3 - Verificar semelhanças do sistema com alguma organização do mundo real:** Identificar alguma semelhança entre as organizações identificadas no sistema, com alguma organização do mundo real a fim de facilitar na definição da estrutura do sistema.

**Passo 4 - Documentar as organizações identificadas:** Detalhar textualmente cada organização identificada no sistema.

#### 4.2.2. Modelo Ambiental do Sistema

Não houve necessidade de modificarmos também a atividade de definir um modelo ambiental do sistema, sendo ela definida como está em Gaia.

Tabela 6: Representação da tarefa Definir modelo ambiental do sistema

<b>Objetivo:</b> - Visa definir um modelo para representar o ambiente onde o SMA irá ser implantado.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entrada:</b> - Documentos de especificação do sistema. (APÊNDICE J)  - Relatório com as organizações do sistema e seus objetivos. (APÊNDICE H)	<b>Saída:</b> - Modelo de Ambiente. (APÊNDICE A)
<b>Passos:</b> - Analisar quais são os recursos do sistema - Adicionar uma descrição do recurso e suas permissões - Documentar a lista de recursos com seus respectivos comentários	

Fonte: elaborado pelo autor



Abaixo são explicados os passos para a elaboração do modelo de ambiente do sistema.

**Passo 1 - Analisar quais são os recursos do sistema:** Este passo visa analisar os documentos de especificação e o relatório das organizações a fim de identificar quais serão os recursos do sistema.

**Passo 2 - Adicionar uma descrição do recurso e suas permissões:** Visa descrever o recurso identificado e definir quais agentes terá permissão de utilizá-lo o que cada irá poder fazer.

**Passo 3 - Documentar a lista de recursos com seus respectivos comentários:** Unir os resultados dos passos anteriores e criar o modelo de ambiente.

#### 4.2.3. Modelo Preliminar de Papéis

Esta atividade não teve muitas modificações com relação a definição em Gaia. A principal delas é com relação às permissões que devem ser definidas para os papéis. Enquanto que em Gaia essas permissões se referem, principalmente, aos recursos do sistema que os papéis terão acesso, no contexto do IE é importante especificar as permissões que os agentes terão sobre as regras do sistema, no caso os compromissos, e sobre seus atos institucionais.

Vale lembrar ainda que esta atividade só será concluída na fase de projeto, quando se sabe como e com quem os papéis irão interagir. E quais serão as permissões necessárias para eles exercerem suas atividades no sistema. Além disso, a fase de designer é susceptível de introduzir funções adicionais logo após a definição da estrutura do sistema e que, portanto, não foram identificadas na fase de análise.

Tabela 7: Representação da tarefa para Definir os papéis preliminares do sistema

<b>Objetivo:</b> - Esta atividade visa identificar as "habilidades básicas" que são necessários pela organização para alcançar os seus objetivos, bem como as interações básicas que são necessárias para essas habilidades.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de Requisitos
<b>Entrada:</b> - Documento de requisitos do sistema. (APÊNDICE J)	<b>Saída:</b> - Modelo preliminar de papéis (APÊNDICE C)

- Relatório com as organizações do sistema e seus objetivos. (APÊNDICE H)	
<p><b>Passos:</b></p> <ul style="list-style-type: none"> <li>- Analisar documentos de especificação do sistema, o relatório das organizações e o modelo de ambiente.</li> <li>- Adicionar uma descrição do papel</li> <li>- Definir as responsabilidades</li> <li>- Definir as permissões</li> </ul>	

Fonte: elaborado pelo autor

A seguir são detalhados os passos para executar a tarefa para definir os papéis preliminares do sistema.

**Passo 1 - Analisar documentos de especificação do sistema, o relatório das organizações e o modelo de ambiente:** Fazer uma análise em cima dos documentos de especificação do sistema, do relatório com as organizações do sistema e do modelo de ambiente a fim de encontrar informação para possíveis papéis.

**Passo 2 - Adicionar uma descrição do papel:** Descrever qual a finalidade do papel.

**Passo 3 - Definir as responsabilidades:** Visa identificar o comportamento esperado dos agentes. Como ele irá se comportar dada uma determinada situação.

**Passo 4 - Definir as permissões:** Especificar quais serão as permissões que os papéis terão para poderem realizar suas atividades no sistema. Cada habilidade do papel pode necessitar de uma permissão diferente, que pode não ser aceita pelos agentes externos. Por isso, deve-se especificar também as permissões que os papéis terão para poderem realizar suas tarefas da maneira que acharem conveniente, caso não cumpram com uma permissão pré-definida. Com isso, devemos permitir também que agentes sejam capazes de criar seus próprios compromissos com o sistema.

Um ponto importante na hora de atribuir permissões aos agentes é que Gaia não permitiu a atribuição estática de papéis. Como no IE isso é permitido, faz-se necessário também atribuir permissões aos papéis para que possam mudar de papel caso desejem.

#### 4.2.4. Definir os Atos Institucionais

A criação desta atividade no processo foi importante, pois o IE permite que os agentes criem seus próprios atos para poderem realizar suas atividades e Gaia não dá apoio à atos institucionais.

Os atos institucionais são definidos como as ações identificadas e necessárias para os papéis interagirem entre si e para a gestão do ambiente institucional. São, portanto, termos identificados no ambiente institucional que quando utilizados trazem efeitos institucionais, podendo mudar o estado do ambiente institucional (OLIVEIRA, 2013). Eles são utilizados em protocolos de interação que levam às conversas que possivelmente irá mudar o estado atual do da instituição.

Tabela 8: Representação da tarefa Definir os atos institucionais

<b>Objetivo:</b> - Identificar os atos institucionais necessários para os papéis interagirem entre si e para a gestão do ambiente institucional.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entrada:</b> - Modelo preliminar de papéis (APÊNDICE C)  - Modelo de ambiente (APÊNDICE A)	<b>Saída:</b> - Relatório com os atos institucionais. (APÊNDICE L)
<b>Passos:</b> - Identificar e descrever os atos preliminares de cada papel	

Fonte: elaborado pelo autor

Abaixo detalhamos o único passo para Definir os Atos Institucionais

**Passo 1 - Identificar e descrever os atos preliminares de cada papel:** Visa identificar quais serão os atos que cada agente que assumir determinado papel irá ter que realizar para interagir com o sistema.

#### 4.2.5. Modelo Preliminar de Interações

Para definir o modelo preliminar de interações não precisamos realizar muitas adaptações. Mantivemos então os objetivos principais da atividade, sendo a única modificação feita foi a inclusão do Relatório com os Atos Institucionais como entrada da atividade. Este será importante para que os responsáveis saibam como os agentes irão agir no momento da interação, ou seja, quais atos institucionais irão utilizar.

Tabela 9: Representação da tarefa para definir o modelo preliminar de interações

<b>Objetivo:</b> - Este modelo captura as dependências e as relações entre os vários papéis na organização, em termos de uma definição de protocolo para cada tipo de interação entre papéis.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entrada:</b> - Relatório com as organizações que irão compor o sistema. (APÊNDICE H)  - Modelo de ambiente (APÊNDICE A)  - Modelo preliminar de papéis (APÊNDICE C)  - Relatório com os atos institucionais	<b>Saída:</b> - Modelo preliminar de interações e protocolos. (APÊNDICE B)
<b>Passos:</b> - Analisar relatório contendo as organizações do sistema e o modelo de ambiente - Definir quais serão as interações de cada organização e dos agentes no ambiente e suas respectivas dependências - Especificar as mensagens utilizadas no tipo FIPA-ACL - Definir o nome, o iniciador, o interlocutor, as entradas, as saídas e a descrição de cada interação.	

Fonte: elaborado pelo autor

Abaixo estão detalhados os passos para Definir as Interações.

**Passo 1 - Analisar relatório contendo as organizações do sistema e o modelo de ambiente:** Fazer uma análise em cima das organizações identificadas no sistema e sob o modelo de ambiente a fim de encontrar informações relevantes para possíveis interações entre os agentes e entre as próprias organizações.

**Passo 2 - Definir quais serão as interações de cada organização e dos agentes no ambiente e suas respectivas dependências:** Identificar e documentar quais serão as interações entre as organizações e/ou entre agentes, identificando também possíveis dependências de outras partes do sistema.

**Passo 3 - Especificar as mensagens utilizadas no tipo FIPA-ACL:** FIPA é uma entidade que vêm especificando padrões, principalmente de comunicação, para o desenvolvimento de sistemas multiagentes. O termo ACL (*Agent Communication Languages*) refere-se a uma linguagem usada para modelar a estrutura das mensagens trocadas entre os agentes que compõem um SMA. Este passo visa modelar a mensagens geradas nas interações usando FIPA-ACL.

**Passo 4 - Definir o nome, o iniciador, o interlocutor, as entradas, as saídas e a**

**descrição de cada interação:** Visa nomear a interação, identificar quem serão os iniciadores, os interlocutores, quais serão as entradas, as saídas e descrever sobre o que se trata a interação.

#### 4.2.6. Definir os Compromissos

A abordagem de Gaia não trata da autonomia dos agentes, pois os obrigam a seguir as regras organizacionais impostas pelo sistema. O uso de compromissos pode ser visto como contratos firmados entre os agentes e a sociedade do sistema, podendo estes serem criados ou quebrados. Fatos que irão afetar diretamente na reputação do agente com os demais agentes do sistema.

A definição de compromissos para representar as regras do sistema foi mais uma adaptação importante feita para processo se adequar ao IE. Os agentes ao criarem seus próprios atos institucionais, também deveriam firmar um compromisso com a instituição sobre o compromisso criado. Com isso, viu-se a necessidade de dar permissão aos agentes de criarem também novos compromissos.

Compromissos são utilizados pelo IE para representar as normas sociais do sistema por ser uma maneira mais simples de representar tais componentes e por não impor tantas restrições como a definição de regras. Esses compromissos expressam restrições sobre as atividades de execução de papéis e protocolos e são de importância primordial na promoção da eficiência no projeto.

Tabela 10: Representação da tarefa para definir os compromissos

<b>Objetivo:</b> - Visa especificar os compromissos preliminares que os agentes devem firmar ao ingressar no sistema e assumir um papel. Compromissos são utilizados pelo IE para representar as normas sociais do sistema.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entrada:</b> - Modelo preliminar de papéis (APÊNDICE C)  - Modelo preliminar de interações (APÊNDICE B)  - Relatório com os atos institucionais. (APÊNDICE L)	<b>Saída:</b> - Documento de especificação dos compromissos. (APÊNDICE M)
<b>Passos:</b>	

- Analisar modelo preliminares de papéis e interações e modelo de ambiente
- Definir critérios de sobrevivência do sistema
- Definir critérios de segurança
- Especificar critérios de avaliação e de tolerância para cada compromisso

Fonte: elaborado pelo autor

Abaixo estão detalhados os passos para a Definição dos Compromissos.

**Passo 1 - Analisar modelo preliminar de papéis e interações e modelo de ambiente:** A análise desses modelos pode ajudar a identificar quais serão os compromissos dos papéis.

**Passo 2 - Definir critérios para funcionamento equilibrado do sistema:** Especificar as regras que definem como a dinâmica da organização deve evoluir ao longo do tempo. Definições de como as ações devem ser realizadas.

**Passo 3 - Definir critérios de segurança:** Definir as regras que devem ser respeitadas pela organização. Definições que devem ser seguidas para executar as ações.

**Passo 4 - Especificar critérios de avaliação e de tolerância para cada compromisso:** Visa especificar quais serão os critérios para se avaliar um compromisso. Especificar como um compromisso será considerado cumprido ou violado.

#### 4.2.7. Especificar a Reputação

A restrição imposta no ambiente institucional pelo IE é flexível. Os agentes que aderirem ao sistema poderão seguir ou não os compromissos firmados com o ambiente e ser recompensado ou punido por isso, ganhando ou perdendo pontos de reputação. Esta tarefa visa especificar como serão atribuído esses pontos de reputação aos agentes e seguindo quais critérios. Fatos esses que devem ser definidos claramente pelo designer do sistema para que possam ser implementados.

Uma observação importante sobre essa atividade é que compromissos diferentes podem acarretar recompensas ou punições diferentes para os agentes. E que é justamente o uso da reputação que irá ajudar a manter um ambiente estável na execução do sistema e ajudar a convencer os agentes a cumprirem seus compromissos, ao mesmo tempo em que o deixam livre para agir de forma autônoma.

Tabela 11: Representação da tarefa Especificar a Reputação

<b>Objetivo:</b> - Esta tarefa visa especificar como será atribuído esses pontos de reputação aos agentes e baseado em que.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Analista de requisitos
<b>Entrada:</b> - Documento de especificação dos compromissos. (APÊNDICE M)  - Relatório com os atos institucionais. (APÊNDICE L)	<b>Saída:</b> - Especificação dos dados de reputação. (APÊNDICE I)
<b>Passos:</b> - Analisar os atos institucionais e os compromissos pré-definidos - Definir critérios para os pontos de reputação para cada compromisso	

Fonte: elaborado pelo autor

**Passo 1 – Analisar os atos institucionais e os compromissos pré-definidos:** Visa coletar informações sobre os atos institucionais e sobre os compromissos pré-definidos para que se possa especificar atitudes sobre a reputação dos agentes.

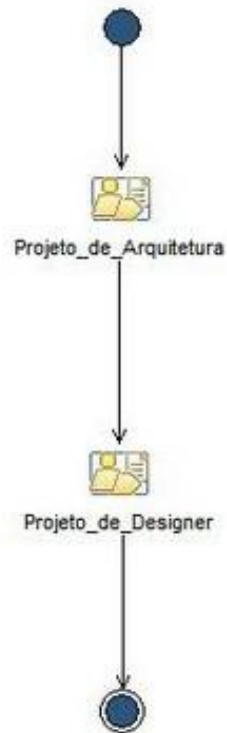
**Passo 2 - Definir critérios para os pontos de reputação para cada compromisso:** O designer do sistema terá que especificar claramente quais serão os critérios utilizados para premiar ou punir os agentes com pontos de reputação. Especificar o que será feito e como será feito, caso seja identificado que um agente quebrou um compromisso ou cumpriu um compromisso.

### 4.3. Fase de Projeto

A fase de projeto utiliza os artefatos gerados na fase anterior como entrada para algumas de suas atividades. Esta fase, assim como em Gaia, foi subdividida em Projeto de Arquitetura e em Projeto de Designer. A figura 6 mostra o fluxo da Fase de Projeto sendo subdividida em Projeto de Arquitetura e Projeto de Designer.

Embora a fase de análise tenha como principal objetivo compreender o que o SMA terá que ser, a fase de projeto é onde decisões têm de ser tomadas sobre as características reais do sistema. Portanto, algumas atividades no projeto (a conclusão dos modelos de papéis e de interações) dependem de decisões sobre a estrutura organizacional. Por isso só podem ser concluídas nessa fase.

Figura 8: Representação do fluxo da Fase de Projeto



Fonte: elaborado pelo autor.

#### 4.3.1. Projeto De Arquitetura

Como já foi citado, a maioria dos SMA projetados atualmente utiliza um modelo muito restritivo a comportamento dos agentes. Obrigando os eles a seguirem a risca as regras organizacionais para manter um funcionamento aceitável do sistema.

A ideia defendida pelo IE é que é possível implementar SMAs em que é utilizada uma arquitetura comum para as estruturas organizacionais. Dessa forma, consegue-se dar mais autonomia aos agentes que se juntam ao ambiente (OLIVEIRA, 2013). Essa é uma alternativa para resolver o principal problema na implementação de SMAs: como manter o desempenho aceitável de um SMA sem que seja necessário restringir demais o comportamento dos agentes.

A Fase de Projeto Arquitetônico podemos utilizá-la da maneira como foi definida em Gaia. Não foi necessário fazer muitas alterações.

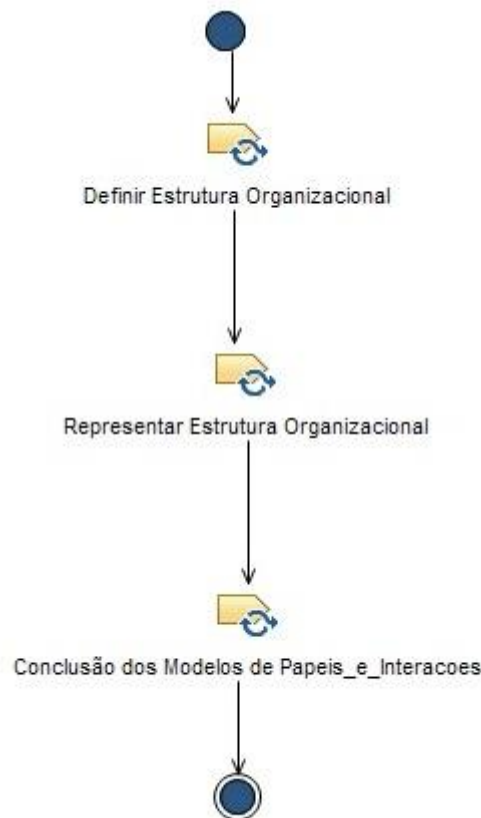
Assim sendo, a fase de projeto arquitetônico inclui:



- A definição da estrutura organizacional do sistema.
- A conclusão do modelo de papéis e do modelo preliminar de interações.

A definição da estrutura do sistema irá auxiliar na conclusão dos modelos de papéis e de interações, pois estes modelos precisam de informações de como será a estrutura do sistema para tomar algumas decisões. Por isso que eles só podem ser concluídos após essa tarefa. A figura 7 representa o fluxo das tarefas do projeto de arquitetura.

Figura 9: Representação do fluxo das tarefas do Projeto de Arquitetura.



Fonte: elaborado pelo autor

#### 4.3.1.1. Definir Estrutura Organizacional

A atividade para definir a estrutura organizacional do sistema pode ser utilizada no processo do IE da mesma forma como foi definida em Gaia. No entanto, é importante fazer algumas observações.

No IE, as regras do sistema são representadas pelos compromissos firmados entre os agentes e a instituição. Estes compromissos são determinados a partir dos atos institucionais necessários nas interações dos agentes no sistema. No entanto, tais atos podem exigir compromissos diferentes aos que foram definidos previamente, ou o próprio agente

pode decidir em realizar a tarefa de outra maneira. Assim, o agente dentro do sistema terá a autonomia de seguir ou não o seu compromisso, podendo quebrar e criar novos compromissos. Dessa forma foram acrescentados como entradas nesta atividade os documentos de compromissos do sistema e o relatório com os atos institucionais dos agentes.

Os fatores de influência para definição da estrutura organizacional citados em Gaia foram os aspectos aproveitados da metodologia para esta atividade. De maneira resumida, podemos descrevê-lo da seguinte forma:

**Eficiência Organizacional e Simplicidade** – visa identificar o quanto o sistema pode ser eficiente sendo o mais simples possível.

**A Influência das Normas Organizacionais** – Trata da necessidade do SMA respeitar as normas de comportamento da organização e de ser capaz de sancioná-las durante a execução.

**A Influência da Organização do Mundo Real** – Visa identificar o quanto uma organização do mundo real foi usada como modelo para a definição da estrutura do sistema e o quanto ela pode ajudar na sua representação.

Tabela 12: Representação para a tarefa de definir a estrutura organizacional

<b>Objetivo:</b> - Esta tarefa irá definir como será organizada a estrutura do sistema.	
<b>Papéis principais:</b> - Arquiteto de Software	<b>Papéis Secundários:</b> - Designer do sistema.
<b>Entrada:</b> - Modelo de ambiente (APÊNDICE A)  - Modelo preliminar de papéis (APÊNDICE C)  - Modelo preliminar de interações (APÊNDICE B)  - Relatório com os atos institucionais. (APÊNDICE L)  - Documento de especificação dos compromissos. (APÊNDICE M)	<b>Saída:</b> - Documento de definição da estrutura organizacional. (APÊNDICE E)

**Passos:**

- Fazer uma análise sobre a eficiência e simplicidade do sistema
- Especificar um regime de controle
- Analisar o quanto as normas organizacionais influenciam no desempenho do sistema
- Definir e documentar a estrutura organizacional.

Fonte: elaborado pelo autor

**Passo 1 - Fazer uma análise sobre a eficiência e simplicidade do sistema:** Significa analisar o quanto o sistema deve ser simples, mas sem deixar de ser eficiente. Para uma organização trabalhar de forma eficiente não é necessário sobrecarregar seus membros. Se isso ocorre deve-se criar novos membros a fim de distribuir melhor a carga de trabalho entre eles para que não haja termos ociosos nem termos sobrecarregados.

Essa simplicidade vai depender da natureza da aplicação e do que for definido pelos *stakeholders* do sistema, pois cada aplicação tem características próprias e o que pode ser considerado simples para uma pode não ser para outra.

**Passo 2 - Especificar um regime de controle:** Visa especificar um regime para o controle das interações entre as organizações. O regime de controle poderá basear-se em um regime de *particionamento da carga de trabalho* (no qual cada um dos colegas tem o mesmo papel e fornece os mesmos serviços) ou em uma *especialização da carga de trabalho* (em que cada um dos membros realiza um serviço/atividade específica).

Quando um grande número de tarefas semelhantes tem de ser realizadas, trabalhar com a partição de tarefas é melhor na redução da complexidade e custos de coordenação, quando a carga de trabalho deriva de algumas atividades complexas, a especialização do trabalho pode incorrer menores custos de coordenação.

Isso também vai depender da natureza de cada aplicação e do que for definido pelos *stakeholders* do sistema.

**Passo 4 - Analisar o quanto as normas organizacionais influenciam no desempenho do sistema:** Fazer uma análise da necessidade de o SMA respeitar as normas de comportamento da organização e de ser capaz de sancioná-las durante a execução.

**Passo 5 - Definir e documentar a estrutura organizacional:** A partir dos resultados dos passos anteriores, especificar detalhadamente como deve ser a estrutura organizacional do sistema.

### 4.3.1.2. Representar a Estrutura Organizacional

Uma vez que a definição de uma estrutura organizacional para o SMA está completa, o designer tem de determinar a forma de representá-la de forma eficaz. Conforme Gaia sugere, adotamos uma notação gráfica para completar a notação textual nessa atividade.

No *Institutional Environments* são usadas CPNs para esse fim, por tratar com mais facilidade de compreensão as interações entre os agentes, que podem muitas vezes ser paralelas entre vários agentes. É a união das CPNs que irá formar o chamado Espaço Normativo do sistema. Para isso sugerimos o uso da ferramenta CPN Tools, ferramenta utilizada para representar as CPNs representadas na demonstração de uso deste trabalho.

Cada agente definido pelo IE para formar o Espaço Normativo (*Monitor Agent*, *Institutional Agent*, *Reputation Agent*, *External Agent*) será representado por uma CPN específica, onde serão interligadas conforme a necessidade de interação.

Tabela 13: Representação da tarefa Representar a Estrutura Organizacional

<b>Objetivo:</b> - Visa fazer uma representação da estrutura organizacional definida.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> - Arquiteto de software.
<b>Entrada:</b> - Documento de Definição da Estrutura Organizacional (APÊNDICE E)	<b>Saída:</b> - Documento de Definição da Estrutura Organizacional (APÊNDICE E).
<b>Passos:</b> - Documentar as relações e funções, com seus respectivos tipos, que existem na organização. - Descrever detalhadamente tais relações e funções	

Fonte: elaborado pelo autor

**Passo 1 – Documentar as relações e funções, com seus respectivos tipos, que existem na organização:** Visa documentar a estrutura definida na atividade anterior. O meio aceito para especificar formalmente uma organização é explicar suas relações e funções que existem dentro dela.

**Passo 2 – Descrever detalhadamente tais relações e funções:** Para todas as representações serem úteis para os desenvolvedores, elas devem ser devidamente detalhadas com a descrição da semântica das relações e com comentários adicionais que poderiam torná-lo mais compreensíveis.

### 4.3.1.3. Conclusão dos Modelos de Papéis e de Interações

Com as funções e protocolos preliminares já identificados, é possível fazer a representação completa dos modelos de interação e de papéis. Isto inclui todas as características de papéis e protocolos (identificação completa de atividades, serviços e protocolos).

De modo geral, esta atividade não sofreu alterações com relação à como ela está definida em Gaia, mas é importante observar que na definição das permissões dos papéis deverá ser dada permissão para que os agentes possam tanto criar seus próprios compromissos como para mudar de papel em tempo de execução. Algo que não acontece em Gaia.

Tabela 14: Representação da tarefa Conclusão dos modelos de papéis e Interações

<p><b>Objetivo:</b> -Esta atividade visa concluir os modelos de papéis e interações. Contendo todos os protocolos identificados após a fase de análise e de definição da estrutura organizacional.</p>	
<p><b>Papéis principais:</b> - Designer do sistema.</p>	<p><b>Papéis Secundários:</b> -</p>
<p><b>Entrada:</b> - Documento de Definição da Estrutura Organizacional (APÊNDICE E)  - Modelo preliminar de papéis (APÊNDICE C)  - Modelo preliminar de interações (APÊNDICE B)</p>	<p><b>Saída:</b> - Modelo completo de papéis. (APÊNDICE C)  - Modelo completo de interações. (APÊNDICE B)</p>
<p><b>Passos:</b> - Analisar a estrutura organizacional definida e os modelos preliminares de papéis e de interações - Definição de todos os papéis do sistema. - Definição de todas as atividades de um papel. - Definir todos os protocolos organizacionais do sistema. - Definir quais serão as permissões que cada papel terá.</p>	

Fonte: elaborado pelo autor

**Passo 1 - Analisar a estrutura organizacional definida e os modelos preliminares de papéis e de interações:** Com a estrutura da organização identificada, o designer pode saber quais papéis terão que interagir com que outros e quais protocolos terão

de ser executados. Assim, os modelos de papéis e de interações podem ser transformados em modelos completos de funções e interações.

**Passo 2 - Definição de todos os papéis do sistema:** Especificar todos os papéis que irão compor o sistema.

**Passo 3 - Definição de todas as atividades de um papel:** Descrição de todas as habilidades que cada agente irá possuir no sistema.

**Passo 4 - Definir todos os protocolos organizacionais do sistema:** Especificar todos os protocolos que cada papel irá participar para interagir no sistema.

**Passo 5 - Definir quais serão as permissões que cada papel terá:** Cada agente que assumir um papel no sistema deverá ter a permissão de criar novos compromissos, ou de criar seus próprios atos institucionais para realizarem suas tarefas. Por isso, o designer do sistema deve especificar claramente quais as permissões atribuídas a cada papel.

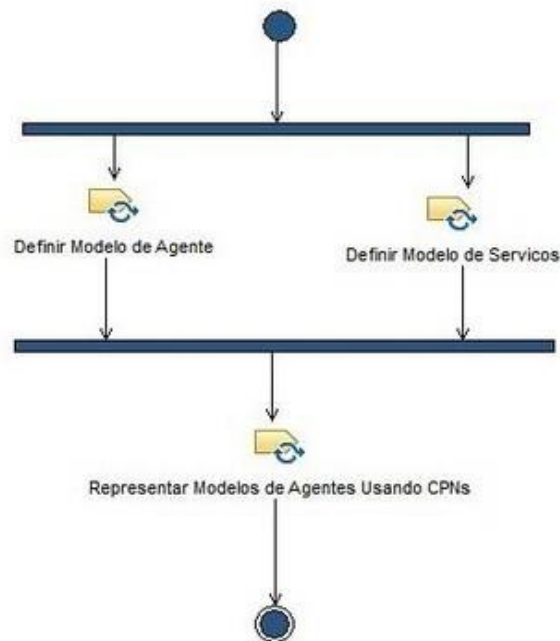
#### 4.3.2. Projeto de Designer

Nesta fase não houve necessidade de fazer muitas adaptações nas atividades. A principal alteração foi a inclusão da atividade para representar os modelos de agentes graficamente utilizando CPNs. A figura 8 representa o fluxo das tarefas do Projeto de Designer.

O projeto de designer passou então a incluir:

- A definição do modelo de agente;
- A definição do modelo de serviços.
- Representar Modelos de Agentes e Serviços Usando CPNs.

Figura 10: Representação do fluxo de tarefas do Projeto de Designer.



Fonte: elaborado pelo autor.

#### 4.3.2.1. Definição do Modelo de Agente

Com relação à definição dos modelos de agentes é importante lembrar que o modelo de agente citado em Gaia é apropriado para a atribuição estática de papéis. O que não acontece no IE. A atribuição de papéis também pode ser de maneira dinâmica.

Para resolver isso, devemos atribuir permissões aos agentes para que possam mudar de papel mesmo depois de ingressarem no sistema.

Tabela 15: Representação da tarefa Definição do Modelo de Agente

<b>Objetivo:</b> - Este modelo irá conter a descrição completa de um agente no sistema. Seu papel desempenhado, protocolos que devem ser seguidos e permissões de interações com os demais agentes do sistema.	
<b>Papéis principais:</b> - Designer do sistema.	<b>Papéis Secundários:</b> -
<b>Entrada:</b> - Modelo completo de papéis. (APÊNDICE C)  - Modelo completo de interações. (APÊNDICE B)	<b>Saída:</b> - Modelo de agente (APÊNDICE F)

**Passos:**

- Identificar quais classes de agentes devem ser definidos para desempenhar os papéis.
- Identificar quantas instâncias dessas classes serão necessárias.

Fonte: elaborado pelo autor

**Passo 1 - Identificar quais classes de agentes devem ser definidos para desempenhar os papéis:** Devem ser especificadas quais classes de agentes serão utilizadas para representar cada papel.

**Passo 2 - Identificar quantas instâncias dessas classes serão necessárias:** Quantas instancias de cada agente serão necessárias para o papel.

#### 4.3.2.2. Definição do Modelo de Serviços

Assim como definido em Gaia, a atividade de definição do modelo de serviços tem o objetivo de identificar e documentar os serviços associados a cada classe de agente ou a cada papel. Assim, o modelo de serviços aplica-se tanto no caso de atribuição estática de papéis como no caso em que agentes podem assumir funções dinamicamente.

Tabela 16: Representação da tarefa Definição do Modelo de Serviços

<b>Objetivo:</b> - Tem o objetivo identificar os serviços associados a cada classe de agente ou, equivalente com cada um dos papéis a serem desempenhados pelas classes de agentes.	
<b>Papéis principais:</b> - Designer do sistema.	<b>Papéis Secundários:</b> -
<b>Entrada:</b> - Modelo completo de papéis. (APÊNDICE C)  - Modelo completo de interações. (APÊNDICE B)	<b>Saída:</b> - Modelo de Serviços (APÊNDICE G)
<b>Passos:</b> - Identificar quais os serviços prestados por cada classe de agente. - Identificar as entradas, as saídas, as pré-condições, e as pós-condições dos serviços.	

Fonte: elaborado pelo autor

**Passo 1 - Identificar quais os serviços prestados por cada classe de agente:** Cada papel assumido por uma classe de agente irá disponibilizar determinados serviços. E



isso irá depender das características dos papéis que esses agentes irão assumir (protocolos, habilidades etc.).

**Passo 2 - Identificar as entradas, as saídas, as pré-condições, e as pós-condições dos serviços:** Para cada um dos serviços que podem ser realizados por um agente, é necessário documentar suas propriedades. Especificamente, temos de identificar as *entradas, saídas, pré-condições e pós-condições*.

#### 4.3.2.3. Representar Modelos de Agentes e Serviços Usando CPNs

Existem diversas formas para se representar a comunicação entre os agentes no ambiente de um sistema multiagentes. Dentre as mais utilizadas estão o uso de AUML e Máquinas de Estado Finitos.

Máquinas de Estado Finitos são geralmente usadas para representar os estados de uma conversa e os fatos que podem acarretar alguma mudança de estado. Mas essa forma de representar a comunicação entre os agentes não é adequada para representar uma comunicação paralela, por exemplo. Como no caso de um agente iniciar comunicação com mais de um agente. Devido a isso, o uso de CPNs (Redes de Petri Coloridas) para esse fim tem crescido bastante.

Redes de Petri oferecem um maior poder de representação de protocolos por permitir especificar tarefas simultâneas entre os agentes. Fato esse que é essencial para abordagem do IE, que pretende manter um ambiente menos restritivo entre os agentes.

Tabela 17: Representação da tarefa Representar Modelos de Agentes e Serviços Usando CPNs

<b>Objetivo:</b> - Esta tarefa tem o objetivo de representar usando Redes de Petri os serviços que serão prestados pelos agentes ou, mais especificamente, pelos papéis que compõe o sistema juntamente com seus protocolos.	
<b>Papéis principais:</b> - Designer do sistema	<b>Papéis Secundários:</b> -
<b>Entrada:</b> - Modelo completo de Papéis.  - Modelo completo de Interações.  - Modelo de serviços	<b>Saída:</b> - Modelo de agentes com CPN.

**Passos:**

- Identificar cada classe de agente com seus respectivos serviços
- Fazer a representação do modelo usando Redes de Petri para cada classe de agente

Fonte: elaborado pelo autor

**Passo 1 - Identificar cada classe de agente com seus respectivos serviços:**

Coletar os dados de cada classe de agentes do sistema para poder representá-los.

**Passo 2 - Fazer a representação do modelo usando Redes de Petri para cada**

**classe de agente:** Modelar os modelos de agentes e de serviços através de CPNs, de forma que represente todas as características dos agentes do sistema.

**4.4. Fase de Implementação das CPNs**

Tendo como entrada os resultados obtidos na fase de projeto, principalmente os modelos completos de agentes e interações. Esta fase tem como atividade a implementação das CPNs definidas na tarefa de representação das CPNs na fase de projeto. A figura

Figura 11: Representação do fluxo da fase de Implementação das CPNs.



Fonte: elaborado pelo autor

**4.4.1. Implementação das CPNs**

Seu objetivo é implementar os modelos de agentes, e suas respectivas interações, que foram representadas usando CPNs.

A tecnologia utilizada nessa fase irá depender da natureza da aplicação e será definida pelo designer do sistema. Mas, a troca de mensagens deverá ser do tipo FIPA-ACL. Vale a pena ressaltar que essas CPNs serão os Deputy Agents (Representantes) dos agentes externos dentro da instituição e eles que realmente descreverão todas as possíveis interações dos agentes externos no contexto do sistema criado.

Tabela 18: Representação da tarefa de Implementar as CPNS

<b>Objetivo:</b> - Visa codificar os modelos de agentes, com suas respectivas interações, representados pelas CPNs.	
<b>Papéis principais:</b> - Desenvolvedores	<b>Papéis Secundários:</b> -
<b>Entrada:</b> - Modelo de agentes com CPN  - Modelo completo de papéis  - Modelo completo de interações  - Modelo de serviços	<b>Saída:</b> - Código fonte das CPNs.
<b>Passos:</b> - Codificação do Sistema.	

Fonte: elaborado pelo autor

**Passo 1 - Codificação do Sistema:** Neste passo a codificação do sistema será iniciada.

#### 4.5. Fase de Testes do Sistema

Após a implementação das CPNs, serão realizados os testes da implementação das CPNs.

Figura 12: Representação do fluxo da fase de testes do sistema.



Fonte: elaborado pelo autor

#### 4.5.1. Realizar testes das CPNs

Será verificado se os modelos de agente estão de acordo com o que foi definido na sua representação gráfica. Se suas interações estão correspondidas e se seus objetivos foram alcançados.

Tabela 19: Representação da tarefa Realizar testes das CPNs

<b>Objetivo:</b> - Visa realizar testes nas CPNs implementadas na fase anterior e verificar se estão de acordo com os modelos definidos.	
<b>Papéis principais:</b> - Testadores	<b>Papéis Secundários:</b> -
<b>Entrada:</b> - Código fonte do sistema	<b>Saída:</b> - Resultados obtidos com a execução do sistema
<b>Passos:</b> - Elaboração e execução de testes no sistema	

Fonte: elaborado pelo autor

**Passo 1 – Elaboração e execução de testes no sistema:** Esta atividade irá verificar se a implementação do sistema está de acordo com o que foi especificado nas CPNs, como também avaliar as CPNs criadas de acordo com suas especificações formais como por exemplo se todos as transições da rede são disparadas em algum momento ou se todos os estados são alcançados em algum momento de sua execução.

Maiores detalhes sobre a execução dessa atividade do processo, como os testes serão realizados por exemplo, são questões que deixamos para trabalhos futuros, pois está mais relacionada com análise de qualidade do sistema, algo que não é o foco deste trabalho.

## **5. DEMONSTRAÇÃO DE USO**

### **5.1. Introdução**

Nos dias atuais o comércio eletrônico passou a ser uma opção a mais para o mundo dos negócios. Essa forma de comércio gerou uma nova forma de lidar com transações que historicamente eram feitas diretamente entre humanos e entre empresas.

Isso acabou por gerar uma maior competitividade entre as empresas prestadoras deste serviço. Quanto mais serviços a empresa oferecer aos usuários, mais complexo acaba sendo o seu sistema, e com isso cresce a necessidade de desenvolver um sistema que consiga representar bem esses serviços.

Um exemplo dessas empresas é o Mercado Livre. Fundado em 1999 por Marcos Galperin, oferece soluções de comércio eletrônico para que pessoas e empresas possam comprar, vender, pagar, anunciar e enviar produtos por meio da Internet.

Em setembro de 2001, o Mercado Livre firmou uma parceria com o eBay, a maior empresa de comércio eletrônico no mundo e pioneira no setor. Com isso, ele incorporou algumas características do eBay que ajudou bastante na disseminação da empresa, principalmente na América Latina, e no amadurecimento de aspectos relacionados ao público usuário MERCADOLIVRE (2014).

Um exemplo dos resultados dessa parceria foi a adição do compartilhamento de experiências entre os usuários no site. Um ambiente onde é possível avaliar e analisar os responsáveis pela venda, os serviços de pagamento, de transporte e sobre a qualidade do produto que é entregue são alguns dos pontos que podem ser compartilhados entre os usuários. O que transforma o Mercado Livre em um sistema de comércio eletrônico bastante popular.

O Mercado livre não vende diretamente nada, apenas é uma forma de conectar vendedores e consumidores de qualquer ponto do Brasil e do mundo MERCADOLIVRE

(2014). Diferente da maioria das empresas de comércio eletrônico, que vendem diretamente o produto aos seus clientes, o Mercado Livre funciona como um ambiente de anúncios onde os usuários pagam para publicar seus produtos e serviços sejam pessoas físicas ou não. Isso possibilita que qualquer usuário, de qualquer lugar do mundo, possa participar do ambiente.

Essa característica acaba por gerar certa desconfiança sob o Mercado Livre, pois assim como acontece no mundo real, existe vendedores de má fé que publicam anúncios falsos para conseguirem interesses próprios. É importante ressaltar isso, pois o Mercado Livre em si, não é responsável pela venda do produto, mas também não deixa de ter uma parcela de responsabilidade no processo de venda. Ele deve ser responsável, por exemplo, pela autenticidade dos anúncios que são publicados no ambiente do seu sistema.

Por ser um sistema que depende basicamente da participação de seus usuários na realização de suas operações e da cooperação dos mesmos para a qualidade dos serviços, o Mercado Livre pode ser facilmente interpretado como sendo um sistema multiagente aberto, visto que permite a participação de qualquer usuário mediante um cadastro inicial.

Essas características do Mercado Livre o qualificaram para uma demonstração de uso deste trabalho, onde seu contexto será adaptado ao processo de desenvolvimento *Institutional Environments* proposto nesse trabalho.

## **5.2. Fase de Levantamento dos Requisitos**

Conforme definido no processo, o primeiro passo foi iniciarmos o levantamento dos requisitos do Mercado Livre através das atividades de Elicitação e Especificação dos Requisitos e a atividade de Validação dos Requisitos.

### **5.2.1. Elicitar os Requisitos**

Nesta tarefa identificamos que o Mercado Livre, por ser um sistema de e-commerce deve por padrão permitir que seus usuários possam comprar, vender, pagar, anunciar e enviar produtos por meio da Internet. Além de permitir a troca de experiências entre eles através de comentários sobre transações antigas que tenham participado. Estes

foram os requisitos identificados através do estudo do site do Mercado Livre e através de entrevistas com o orientador.

### **5.2.2. Validar os Requisitos**

Para se realizar esta tarefa seria necessário reuniões com o solicitante do sistema, para esclarecimento de possíveis dúvidas sobre os requisitos do sistema. Entretanto, não foi possível nos comunicar com os responsáveis do Mercado Livre. Portanto, a validação dos requisitos desta demonstração de uso do processo foi feita através de reuniões com o próprio orientador.

## **5.3. Fase de Análise**

Na fase de análise o objetivo é fazer uma análise inicial de como será o funcionamento do sistema. Identificar quais serão os recursos utilizados, quais os papéis envolvidos, as interações necessárias etc.

### **5.3.1. Identificação das Organizações do Sistema**

Analisando o ambiente do Mercado Livre podemos identificar que um usuário pode tanto exercer ações como vendedor como também de comprador, dependendo da transação em que está participando. Entretanto um usuário externo pode apenas desejar visitar o site, antes de realizar seu cadastro. Com isso esse usuário navegante será representado no ambiente como visitantes.

Independente dos papéis que exercem no momento, os usuários participam de uma mesma organização geral dentro do sistema do Mercado Livre, pois não são formados grupos entre os agentes da organização principal. Podemos identificar essa organização como sendo a própria plataforma do Mercado Livre, onde o principal objetivo é manter transações de compra e venda de produtos e/ou serviços entre os usuários cadastrados. Oferece a possibilidade de anunciarem seus produtos de maneira conjunta (uma empresa que anuncia

seus diversos serviços) ou de maneira independente (uma pessoa comum que realiza o anúncio de algum produto).

### 5.3.2. Modelo Preliminar de Papéis

Os agentes que compõe a abordagem do IE para ambientes institucionais são: Agente Institucional (representando a própria instituição), Agente Monitor (responsável em monitorar os agentes que assumem um papel), Agente de Reputação (armazena informações sobre a reputação dos agentes), Agente Administrador de Banco de Dados (responsável sobre dados dos agentes). Não há necessidade de representá-los nesse trabalho, pois trata-se de recursos já oferecidos pela ferramenta.

Para especificar os papéis presentes na plataforma do Mercado Livre seria interessante especificar papéis baseando-se em cada serviço oferecido pela companhia. Com isso, identificamos os seguintes papéis na plataforma:

Tabela 20: Detalhes do Papel de Visitantes

<b>Nome do Papel</b>	Visitante
<b>Descrição</b>	Usuário que apenas navega no site.
<b>Permissões</b>	Um visitante terá acesso a todos os anúncios no sistema, mas não poderá interagir com os vendedores antes de realizar seu cadastro no sistema.
<b>Responsabilidades de Equilíbrio</b>	Um visitante irá navegar pelo site em busca de informações de seu interesse. Caso resolva realizar alguma transação no sistema, ele terá que se cadastrar e assumir o papel de vendedor ou o papel de comprador.
<b>Responsabilidades de Segurança</b>	Um visitante irá manter o seu comportamento normal no ambiente, até tornar-se um usuário cadastrado.

Fonte: elaborado pelo autor

Tabela 21: Detalhes do Papel de Vendedor

<b>Nome do Papel</b>	Vendedor
<b>Descrição</b>	Usuário cadastrado que quiser vender algum serviço/produto deverá assumir esse papel.
<b>Permissões</b>	Um vendedor poderá fazer anúncios e comentários no ambiente do sistema.
<b>Responsabilidades de Equilíbrio</b>	Um vendedor irá anunciar um produto/serviço e conforme as ofertas recebidas, irá analisar qual será melhor para ele. Quando for feito o



	pagamento pelo comprador, ele irá confirmar o recebimento e irá enviar o produto ao comprador.
<b>Responsabilidades de Segurança</b>	Um vendedor irá manter o seu comportamento normal no ambiente, até que algum comprador deixe de realizar o pagamento de uma compra, ou o próprio vendedor deixe de enviar o produto.

Fonte: elaborado pelo autor

Tabela 22: Detalhes do Papel Comprador

<b>Nome do Papel</b>	Comprador
<b>Descrição</b>	Usuário cadastrado que quiser adquirir algum serviço/produto deverá assumir esse papel.
<b>Permissões</b>	Um comprador poderá fazer ofertas em anúncios, adquirir serviços no site e fazer comentários no ambiente do sistema.
<b>Responsabilidades de Equilíbrio</b>	Um comprador irá fazer uma oferta por algum produto/serviço de seu interesse. Caso seja aceita pelo vendedor, ele terá que realizar o pagamento conforme a definição do vendedor.
<b>Responsabilidades de Segurança</b>	Um comprador irá manter o seu comportamento normal no ambiente, até que algum vendedor deixe de confirmar o recebimento do pagamento ou não receber o produto que comprou.

Fonte: elaborado pelo autor

Vale ressaltar que os usuários cadastrados no sistema poderão assumir papéis em tempo de execução. Ou seja, poderão assumir o papel de vendedor, de comprador ou até mesmo de os dois ao mesmo tempo. Isso explica o fato de um mesmo usuário poder utilizar o sistema para comprar um produto ao mesmo tempo em que anuncia outro para venda.

Para isso, é necessário que, ao se definir um papel, seja dada a ele a permissão de exercer mais de um papel ou que solicite mudança de papel, caso seja necessário em suas atividades no sistema. Tal permissão será concedida ou negada, conforme análise dos dados de reputação do agente.

Os dados de reputação, por sua vez, devem ser claramente especificados critérios de aceitação para que um agente seja considerado confiável ou não. Essa é tarefa do designer do sistema.

### 5.3.3. Modelo Ambiental do Sistema

O Mercado Livre pode ser descrito como sendo um ambiente onde os usuários cadastrados podem realizar transações comerciais e trocar experiências através de comentários sobre outros usuários cadastrados, seja ele uma pessoa física ou alguma prestadora de serviço. Para isso, os agentes que assumirem os papéis de “vendedor” e de “comprador” devem dispor da capacidade de anunciar para venda ou adquirir um produto ou serviço, e também de fazer comentários sobre qualquer aspecto envolvido na transação. Não há sentido o papel de “Visitante” ter algum desses recursos disponíveis dentro do sistema, visto que ele só irá participar efetivamente de alguma atividade no ambiente após ser cadastrado.

Portanto, apenas usuários cadastrados e que assumam o papel de vendedor ou de comprador poderão ter acesso a todos os recursos do sistema.

Tabela 23: Representação do modelo de ambiente do Mercado Livre

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
<b>Qual papel terá acesso ao recurso</b>	<b>O que o agente pode fazer usando o recurso.</b>	<b>Variável que irá representar o recurso legível do ambiente.</b>	<b>Breve descrição da acessibilidade do recurso.</b>
‘Visitante’	‘Leitura’	‘Anúncios’	Os anúncios poderão ser apenas acessados pelos agentes visitantes.

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
‘Comprador’	‘Leitura’	‘Anúncios’	Os anúncios poderão ser acessados livremente por um comprador.

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
‘Comprador’	‘Editar’	‘Comentários’	Um comprador poderá realizar comentários sobre suas experiências no sistema.

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
‘Vendedor’	‘Editar’	‘Anúncios’	Os vendedores poderão editar seus anúncios.

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
‘Vendedor’	‘Leitura’	‘Anúncios’	Vendedores poderão apenas acessar os anúncios dos demais

			vendedores.
--	--	--	-------------

<b>Papel</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
'Vendedor'	'Editar'	'Comentários'	Um vendedor poderá realizar comentários sobre suas experiências no sistema.

Fonte: elaborado pelo autor

#### 5.3.4. Definir os Atos Institucionais

*Cadastro* – O agente instituição irá disponibilizar um papel de vendedor ou um papel de comprador para o novo agente. O mesmo irá ter a liberdade de escolher qual irá assumir, dada a sua necessidade no sistema. É então criado seus dados de reputação pelo agente de reputação.

*Cancelar Cadastro* – O agente deixará de ter permissões de um usuário cadastrado no ambiente, passando a ser apenas um agente visitante. O agente instituição então irá desalocar o seu papel deixando-o disponível no sistema para um possível novo cadastro. O agente monitor do papel irá informar ao agente de reputação a saída do agente para que ele possa atualizar seus dados de reputação.

*Mudança de Papel* – Um agente cadastrado poderá escolher qual papel irá assumir. Para isso, ele solicita permissão ao agente instituição que lhe dará permissão para isso e irá firmar novos compromissos com o agente. O agente reputação então será informado sobre os dados desse agente de acordo com o papel exercido por ele.

*Anunciar Produto/Serviço* – O agente que assumir o papel de vendedor irá informar ao agente instituição o anúncio de um novo produto. O agente instituição então checa seus dados de reputação com o agente de banco de dados para permitir ou não o anúncio do agente.

*Cancelar Anúncio* – O agente de banco de dados irá apagar o anúncio do sistema e reservar o espaço de memória para um próximo anúncio.

*Fazer oferta* – Um agente no papel de comprador poderá fazer uma oferta em anúncio de um vendedor. O agente instituição irá então verificar se a oferta está de acordo com os critérios estabelecidos para a transação e informar ao agente vendedor se a oferta for bem aceita para que ele possa decidir em aceitar ou não a oferta.

*Cancelar Oferta* – O comprador que realizar a oferta poderá solicitar o cancelamento da oferta sem punições na sua reputação desde que a sua oferta não seja aceita pelo comprador ou ele não confirme a compra. Caso contrário, seus dados de reputação serão afetados de acordo com a especificação dada pelo designer do sistema para casos que um comprador não honrar algum compromisso com os vendedores.

*Aceitar oferta* – Se a oferta feita por um comprador estiver de acordo com os dados especificados no anúncio (aceitar oferta acima de determinado valor, aceitar primeira proposta, aceitar maior proposta etc.), o agente vendedor poderá decidir se aceitará ou não.

*Rejeitar oferta* – Se a oferta feita por um comprador não estiver de acordo com os dados especificados no anúncio, a oferta será automaticamente rejeitada pelo sistema.

*Confirmar compra* - Após o agente vendedor aceitar a oferta, o agente comprador poderá confirmar ou não a compra. Caso confirme, será firmado um novo compromisso com o agente instituição para efetuar o pagamento.

*Realizar Pagamento* – Depois de confirmar a compra, o agente comprador terá um compromisso a cumprir. Terá que realizar o pagamento dentro de um prazo estabelecido pelo vendedor no seu anúncio. Caso contrário, o agente reputação será informado para atualizar os dados de reputação do agente atribuindo-lhe pontos negativos pelo atraso ou pelo não pagamento.

*Confirmar Pagamento* – O vendedor por sua vez, terá que confirmar o pagamento feito pelo comprador. Isso fará o agente reputação atualizar os dados de reputação do agente comprador atribuindo-lhe pontos positivos. Caso o pagamento já tenha sido feito, mas o vendedor não confirme o recebimento, o agente reputação irá ser informado para atribuir pontos negativos ao vendedor pelo seu atraso.

*Enviar Produto* – Após confirmar que recebeu o pagamento, o agente vendedor deverá que enviar o produto/serviço ao comprador.

*Confirmar recebimento do produto/serviço* – Como última atividade numa transação de compra, o agente comprador deverá confirmar que recebeu o produto/serviço do vendedor.

*Comentários dentro do sistema* – Os usuários cadastrados no sistema e que assumem o papel de vendedor ou de comprador no sistema poderão comentar sobre os aspectos nas interações que participou.

### 5.3.5. Modelo Preliminar de Interações

No modelo preliminar das interações no Mercado Livre constam as principais interações realizadas pelos papéis de Visitante, Vendedores e Compradores.

Tabela 24: Representação das interações do papel Visitante

<b>NOME DO PROTOCOLO</b>	Pesquisa no site.
<b>INICIADOR</b>	Visitante.
<b>INTERLOCUTOR</b>	Vendedor.
<b>ENTRADAS</b>	Anúncios realizados pelos vendedores cadastrados no sistema.
<b>SAÍDAS</b>	Um visitante pode decidir em se cadastrar no sistema para poder assumir o papel de vendedor ou de comprador.
<b>DESCRIÇÃO</b>	Um visitante irá visitar o site no intuito de buscar informações a respeito de algum assunto de seu interesse. Seja uma busca específica de algum produto/serviço seja com relação ao próprio site.

Fonte: elaborado pelo autor

Tabela 25: Representação das interações do papel Comprador

<b>NOME DO PROTOCOLO</b>	Negociar produto.
<b>INICIADOR</b>	Comprador.
<b>INTERLOCUTOR</b>	Vendedor.
<b>ENTRADAS</b>	Oferta do comprador.
<b>SAÍDAS</b>	Venda concluída – caso o vendedor aceite a proposta do comprador. Venda cancelada – caso o vendedor recuse a proposta do comprador.
<b>DESCRIÇÃO</b>	O comprador irá negociar com o vendedor uma transação no sistema. Para isso, fará ofertas sobre o produto/serviço que o vendedor tenha anunciado. O vendedor por sua vez, irá decidir se aceita ou não essas ofertas para poder fechar negócio.

<b>NOME DO PROTOCOLO</b>	Comentários sobre vendedores
<b>INICIADOR</b>	Comprador.
<b>INTERLOCUTOR</b>	Vendedor.
<b>ENTRADAS</b>	Comentários textuais.
<b>SAÍDAS</b>	Atualização dos dados de reputação do vendedor.
<b>DESCRIÇÃO</b>	Um comprador poderá realizar comentários sobre as transações da qual participou com um vendedor.

Fonte: elaborado pelo autor

Tabela 26: Representação das interações do papel Vendedor

<b>NOME DO PROTOCOLO</b>	Realizar Anúncio.
<b>INICIADOR</b>	Vendedor.
<b>INTERLOCUTOR</b>	Comprador.
<b>ENTRADAS</b>	Descrição do produto/serviço, preço e informações de contato.
<b>SAÍDAS</b>	Anúncio postado no site.
<b>DESCRIÇÃO</b>	Um vendedor irá poder realizar anúncios no site para que compradores possam entrar em contato para uma negociação.

<b>NOME DO PROTOCOLO</b>	Comentários sobre compradores.
<b>INICIADOR</b>	Vendedor.
<b>INTERLOCUTOR</b>	Comprador.
<b>ENTRADAS</b>	Comentários textuais.
<b>SAÍDAS</b>	Atualização dos dados de reputação do comprador.
<b>DESCRIÇÃO</b>	Um vendedor poderá realizar comentários sobre as transações da qual participou com um comprador.

Fonte: elaborado pelo autor

### 5.3.6. Definir os Compromissos

Os compromissos firmados dentro do Mercado Livre são entre os usuários e o sistema e entre os próprios usuários. Portanto os agentes que representam os usuários podem tanto ser os devedores como podem ser os credores.

O primeiro compromisso firmado pelos agentes é com a própria instituição do Mercado Livre. Todo usuário cadastrado no sistema concordará em seguir os termos de uso do sistema, estando propenso a punições caso não as cumpra.

*Compromissos dos visitantes* - Um usuário visitante irá se comprometer ao navegar pelo sistema, que ao se cadastrar irá assumir em algum momento os papéis de vendedores ou de compradores no sistema e realizar alguma ação no ambiente.

Tabela 27: Compromissos assumidos pelo papel Visitante

<b>Identificador</b>	Cadastro
<b>Credor</b>	Mercado Livre
<b>Devedor</b>	Visitante
<b>Estado</b>	O cadastro do usuário está ativo no sistema.
<b>Condição</b>	Após fazer seu cadastro

	no sistema um visitante irá interagir com o ambiente como vendedor ou como comprador.
<b>Conteúdo</b>	Sempre que um usuário estiver cadastrado no sistema, terá que assumir um papel e realizar uma ação no ambiente.

Fonte: elaborado pelo autor

A partir daí, sempre que um agente assumir um papel de vendedor ou de comprador irão herdar os compromissos pré-definidos para esses papéis. São eles:

*Compromissos dos vendedores* – Um vendedor estará compromissado em fazer apenas anúncios verídicos, em confirmar o pagamento feito por um comprador após o seu recebimento, em enviar o produto adquirido pelo comprador.

Tabela 28: Compromissos assumidos pelo papel Vendedor

<b>Identificador</b>	Realizar Anúncio
<b>Credor</b>	Mercado Livre
<b>Devedor</b>	Vendedor
<b>Estado</b>	Um anúncio é postado no sistema.
<b>Condição</b>	A partir do momento que um vendedor anunciar um item este compromisso será avaliado.
<b>Conteúdo</b>	Todos os anúncios são verídicos e não é constatada nenhuma malícia no anúncio.

<b>Identificador</b>	Confirmar pagamentos
<b>Credor</b>	Comprador
<b>Devedor</b>	Vendedor
<b>Estado</b>	Um comprador realizou o pagamento.
<b>Condição</b>	A partir do momento que um comprador confirmar o pagamento.
<b>Conteúdo</b>	Se o pagamento foi confirmado pelo comprador, então o vendedor terá dois dias para confirmar seu recebimento.

<b>Identificador</b>	Envio de produtos/serviços
<b>Credor</b>	Comprador
<b>Devedor</b>	Vendedor
<b>Estado</b>	O pagamento já foi realizado.
<b>Condição</b>	A partir do momento da confirmação do recebimento do pagamento.
<b>Conteúdo</b>	Todos os produtos/serviços anunciados serão enviados dentro do prazo para seus compradores.

Fonte: elaborado pelo autor

*Compromissos dos compradores* – Sempre que um comprador confirmar uma compra estará comprometido em realizar o pagamento dentro dos critérios estabelecidos pelo vendedor.

Tabela 29: Compromissos assumidos pelo papel Comprador

<b>Identificador</b>	Realizar pagamento
<b>Credor</b>	Vendedor
<b>Devedor</b>	Comprador
<b>Estado</b>	Uma compra foi confirmada.
<b>Condição</b>	A partir do momento que um comprador confirmar uma compra.
<b>Conteúdo</b>	O pagamento deverá ser feito de acordo com a especificação do vendedor (prazo, valor etc).

Fonte: elaborado pelo autor

### 5.3.7. Especificar a Reputação

Baseando-se nos compromissos que definidos previamente para os papéis, podemos especificar o quanto um agente irá ser recompensado, ou o quanto ele será punido com relação ao cumprimento de seus compromissos.

Os dados de reputação dependem da especificação dada pelo designer do sistema. Cada ação realizada pelos agentes pode ter um peso na avaliação da sua reputação. Cabe



então ao designer analisar o valor que o sistema pode ganhar ou perder com tais ações dos agentes. Duas ou mais ações bem sucedidas por um agente, por exemplo, podem ter atribuição de pontos de reputação diferentes, dependendo do impacto que essas ações terá no ambiente. Do mesmo modo, ações mal sucedidas podem ter punições diferentes, dependendo de sua importância no ambiente.

Para o compromisso dos visitantes, podemos avaliá-lo a partir do seu cadastro, já que é a partir do cadastro que um usuário poderá interagir no sistema. Após o cadastro, o agente terá uma reputação desconhecida e só será alterada após alguma interação do agente no ambiente. Se o mesmo não realizar nenhuma atividade no sistema depois de cadastrado ou formalmente avaliado pelos demais agentes em suas atividades, ele poderá ser julgado como inativo pelo sistema e não ser relacionado para interagir com os demais agentes.

Para a avaliação dos demais papéis do Mercado Livre (compradores e vendedores), podemos atribuir uma reputação desconhecida para todo usuário recém-cadastrado no sistema, de modo que, os agentes veteranos no ambiente saibam que o agente é novato e decidam se irão interagir com ele ou não. A partir daí, atualizamos seus pontos de reputação a cada compromisso não cumprido por esse agente. Ou até mesmo incrementar seus pontos a cada compromisso cumprido. Se a reputação de um agente chegar a um nível muito baixo ele será julgado como inativo pelo sistema. Cabe ao designer definir o quão baixo deverá ser o nível de reputação para ser considerado baixo.

O modelo real de reputação seguido pelo site do Mercado Livre leva em consideração principalmente a frequência de avaliações positivas e negativas para um usuário. Após uma interação entre vendedores e compradores é possível trocar informação sobre sucessos ou fracassos através de comentários ou avaliações por votação, nível de satisfação por exemplo. Para os demais usuários que quiserem interagir com outros, deverão analisar por conta própria esses dados e decidir se vão interagir ou não com eles.

No modelo proposto para representar a reputação usando o contexto do IE, isso será feito através da análise dos dados de reputação que cada agente irá fazer antes de iniciar uma interação. Ou seja, no momento que um agente interagir com o ambiente é através da análise desses dados que o agente saberá quais outros agentes são aconselháveis para interação. Por exemplo, um comprador que desejar fazer uma compra de um dado produto irá analisar primeiramente a reputação dos vendedores desse produto em interações passadas e só então saberá quais são os agentes aconselháveis de interagir.

## 5.4. Fase de Projeto

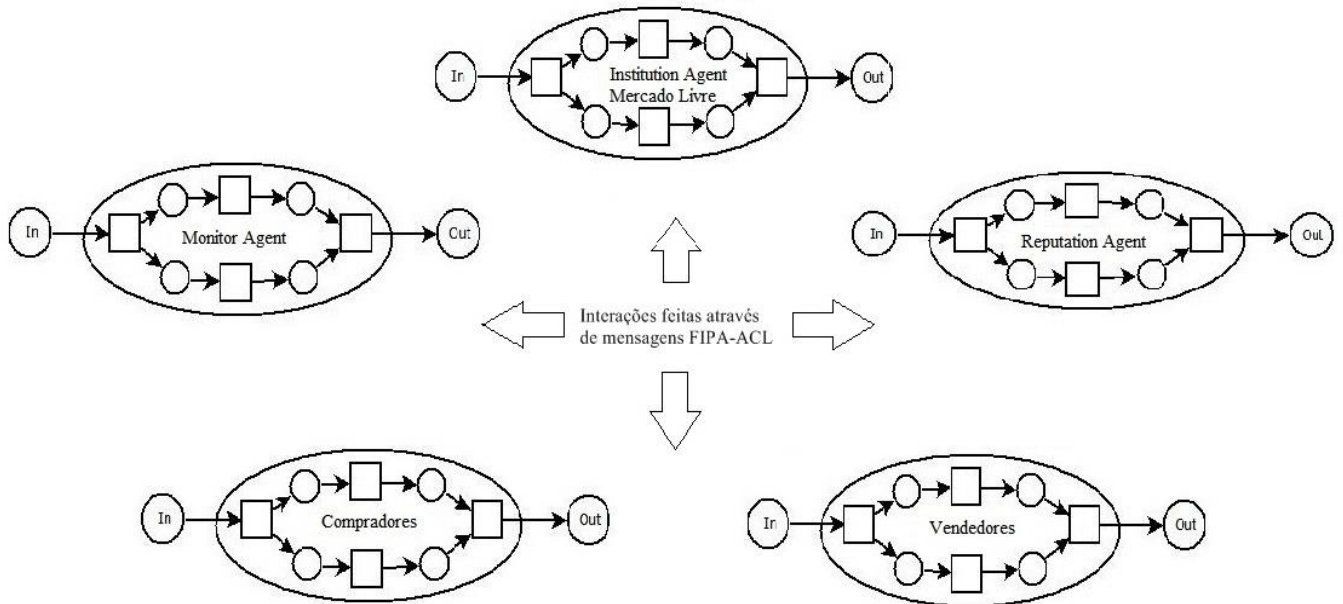
Fase de definições do funcionamento do sistema. Nesta fase serão definidas como as funcionalidades do sistema serão implementadas. Ou seja, como o sistema irá funcionar.

### 5.4.1. Projeto de Arquitetura

Para o Mercado Livre é importante observamos que seu objetivo principal é servir de intermediário entre usuários que desejam realizar alguma transação de comércio eletrônico. Podemos descrever sua arquitetura como sendo composta por usuários vendedores que poderão realizar anúncios de produtos e/ou serviços, onde usuários compradores poderão fazer ofertas por um anúncio e confirmar sua compra. Usuário vendedores e compradores também poderão fazer comentário no ambiente sobre suas experiências no sistema. A figura abaixo demonstra a arquitetura do Mercado Livre modelado conforme o contexto do IE. Podemos ver que os papéis de Compradores e Vendedores irão se relacionar entre si e com os próprios agentes de sistema, que são os Agentes Monitores, Agentes de Reputação e com o próprio agente da instituição, o Agente Mercado Livre.

O Mercado Livre também poderá se relacionar com outros sistemas independentes para fornecer os serviços de pagamento e transporte dos produtos. Mas o objetivo deste estudo de caso é representar um sistema de *e-commerce*, no caso o Mercado Livre, com a abordagem do IE. Os demais sistemas são irrelevantes para nosso objetivo.

Figura 13: Representação da arquitetura do Mercado Livre



Fonte: elaborado pelo autor

#### 5.4.1.1. Definir Estrutura Organizacional

A definição da estrutura usada no Mercado Livre torna-se mais fácil quando analisado os fatores considerados para essa atividade no processo do IE. São eles: Eficiência Organizacional, Regime de Controle, Influência das Normas Organizacionais e a Influência das Organizações do Mundo Real.

Com relação à Eficiência Organizacional e ao Regime de Controle, os agentes visitantes que se cadastrarem no sistema irão interagir no ambiente ou como vendedores ou como compradores, de acordo com sua necessidade. Cada agente irá realizar as tarefas que estão relacionadas ao seu papel.

Quando assumirem o papel de vendedores, irão interagir somente com visitantes e compradores. Visitantes apenas visualizaram suas atividades, enquanto que compradores poderão trocar mensagens com vendedores.

Já quando assumirem o papel de compradores, as interações serão feitas apenas com os vendedores, visto que não faz sentido um agente comprador interagir com um agente que é apenas visitante.

Com relação à Influência das Normas Organizacionais, no ambiente de um sistema desenvolvido seguindo o IE, os agentes podem ingressar no sistema dinamicamente e assumir papéis em tempo de execução. Assim sendo foi dada a permissão para os agentes realizarem suas ações a partir da criação de novos compromissos. Isso inclusive foi uma das

adaptações com relação à metodologia Gaia, pois o contexto de Gaia não permite que agentes assumam papéis em tempo de execução.

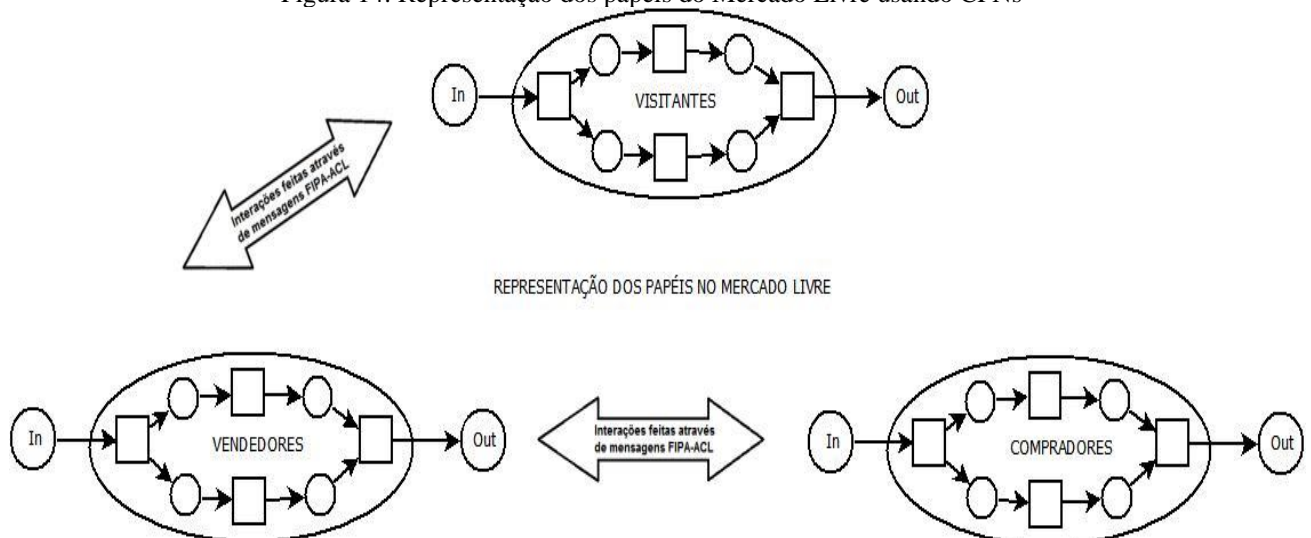
Agentes externos não precisarão saber de antemão como funciona o ambiente do sistema. Ao ingressar no ambiente, eles firmaram compromissos com a instituição, concordando em agregar valor ao ambiente. Caberá a própria instituição avaliar o comportamento do agente. Se ele estará contribuindo com o sistema, ou se está prejudicando.

Com relação à Influência do Mundo Real, o Mercado Livre busca representar no ambiente web, as transações comerciais realizadas comumente entre os humanos. Tentando criar um modelo onde seja possível troca de informações e experiências entre seus usuários, do mesmo modo em que ocorre informalmente entre as pessoas quando comentam entre si sobre determinada loja ou determinado comerciante.

#### 5.4.1.2. Representar Estrutura Organizacional

O processo define que o uso de uma notação gráfica pode ser usada para facilitar a compreensão do sistema pelos desenvolvedores. Dessa forma, com a definição das interações e dos papéis do ambiente do Mercado Livre, podemos representar a estrutura do sistema, utilizando Redes de Petri como na figura abaixo. Para isso sugerimos o uso da ferramenta CPN Tools, ferramenta utilizada para representar as CPNs desta demonstração de uso.

Figura 14: Representação dos papéis do Mercado Livre usando CPNs



Fonte: elaborado pelo autor

### 5.4.1.3. Conclusão dos Modelos de Papéis e Interações

De acordo com o processo, após a definição de como será a estrutura do sistema, podemos terminar o modelo preliminar de papéis, incrementando o mesmo com as interações entre os agentes e com seus protocolos. No caso, os protocolos são vistos pelo IE como sendo uma sequência pré-definida de troca de mensagens (as interações entre os agentes) pelas CPNs que irão representar os papéis dos agentes.

Tabela 30: Detalhes do papel Visitante

<b>Nome do Papel</b>	Visitante
<b>Descrição</b>	Usuário que apenas navega no site.
<b>Permissões</b>	Um visitante terá acesso a todos os anúncios no sistema, mas não poderá interagir com os vendedores antes de realizar seu cadastro no sistema.
<b>Responsabilidades de Equilíbrio</b>	Um visitante irá navegar pelo site em busca de informações de seu interesse. Caso resolva realizar alguma transação no sistema, ele terá que se cadastrar e assumir o papel de vendedor ou o papel de comprador.
<b>Responsabilidades de Segurança</b>	Um visitante irá manter o seu comportamento normal no ambiente, até tornar-se um usuário cadastrado.
<b>Interações</b>	Vendedores.
<b>Compromissos</b>	Um usuário visitante irá se comprometer, ao navegar pelo sistema, que irá se cadastrar no sistema, caso deseje participar de alguma negociação de produto e/ou serviços.
<b>Atos Institucionais</b>	Cadastro.

Fonte: elaborado pelo autor

Tabela 31: Detalhes do papel Comprador

<b>Nome do Papel</b>	Comprador
<b>Descrição</b>	Usuário cadastrado que quiser adquirir algum serviço/produto deverá assumir esse papel.
<b>Permissões</b>	Um comprador poderá fazer ofertas em anúncios, adquirir serviços no site e fazer comentários no ambiente do sistema.
<b>Responsabilidades de Equilíbrio</b>	Um comprador irá fazer uma oferta por algum produto/serviço de seu interesse. Caso seja aceita pelo vendedor, ele terá que realizar o pagamento conforme a definição do vendedor.
<b>Responsabilidades</b>	Um comprador irá manter o seu comportamento

<b>de Segurança</b>	normal no ambiente, até que algum vendedor deixe de confirmar o recebimento do pagamento ou não receber o produto que comprou.
<b>Interações</b>	Vendedores.
<b>Compromissos</b>	Sempre que um comprador confirmar uma compra, estará comprometido em realizar o pagamento dentro dos critérios estabelecidos pelo vendedor.
<b>Atos Institucionais</b>	Cancelar Cadastro, Mudança de Papel, Fazer oferta, Cancelar Oferta, Confirmar compra, Realizar Pagamento, Confirmar recebimento do produto/serviço, Comentários dentro do sistema.

Fonte: elaborado pelo autor

Tabela 32: Detalhes do papel Vendedor

<b>Nome do Papel</b>	Vendedor
<b>Descrição</b>	Usuário cadastrado que quiser vender algum serviço/produto deverá assumir esse papel.
<b>Permissões</b>	Um vendedor poderá fazer anúncios e comentários no ambiente do sistema.
<b>Responsabilidades de Equilíbrio</b>	Um vendedor irá anunciar um produto/serviço e conforme as ofertas recebidas irão analisar qual será melhor para ele. Quando for feito o pagamento pelo comprador, ele irá confirmar o recebimento e irá enviar o produto ao comprador.
<b>Responsabilidades de Segurança</b>	Um vendedor irá manter o seu comportamento normal no ambiente, até que algum comprador deixe de realizar o pagamento de uma compra, ou o próprio vendedor deixe de enviar o produto.
<b>Interações</b>	Irá interagir com os compradores e visitantes.
<b>Compromissos</b>	Um vendedor estará comprometido em fazer apenas anúncios verídicos, em confirmar o pagamento feito por um comprador após o seu recebimento, em enviar o produto adquirido pelo comprador.
<b>Atos Institucionais</b>	Cancelar Cadastro, Mudança de Papel, Anunciar Produto/Serviço, Cancelar Anúncio, Rejeitar oferta, Aceitar oferta, Enviar Produto, Confirmar Pagamento, Comentários dentro do sistema.

Fonte: elaborado pelo autor

## 5.4.2. Projeto de Designer

Aqui são definidos os modelos de agentes e de serviços do Mercado Livre.

### 5.4.2.1. Definição do Modelo de Agente

O objetivo principal dessa atividade é identificar quais agentes e classes que devem ser definidas para desempenhar os papéis específicos e quantas instâncias de cada classe tem que ser instanciado no sistema atual.

Fazendo uma análise da estrutura do Mercado Livre, podemos facilmente identificar que o número de instâncias das classes de agentes que irão representar os papéis de ‘visitantes’, ‘vendedores’ e ‘compradores’ são indeterminados. Podendo ser instanciadas quantas forem necessário.

Segundo o IE, cada agente externo que ingressar no ambiente terá seu respectivo *deputy agent* (agente responsável em representar o papel do agente externo no ambiente) e seu *monitor agent* (agente responsável em monitorar as ações de um *deputy*).

Portanto, no momento em que uma classe de agente é instanciada no sistema e assume um papel, o *institutional agent* irá alocar um *deputy agent* e um *monitor agent* para o agente externo. O agente externo passará então a ser representado no ambiente pelo seu *deputy* e será monitorado pelo seu *monitor*

Vale lembrar também, que o modelo de agentes usado em GAIA é próprio para atribuição de papéis estaticamente. No caso do IE, os agentes têm permissão de alterar suas funções no sistema. Ou seja, podem mudar de papel. Para permitir isso, foi definido que os papéis terão a permissão de alterar seu papel mediante comunicado a própria instituição.

Isso foi necessário para representar o Mercado Livre, pois no sistema um mesmo usuário pode ser ora vendedor, ora comprador, ou até mesmo assumir os dois papéis simultaneamente.

### 5.4.2.2. Definição do Modelo de Serviços

O modelo de serviço visa definir os serviços providos por cada classe de agente usada no sistema.

Estabelecemos aqui quais são os serviços providos pelas classes de agentes que irão representar os papéis de visitantes, vendedores e de compradores. E os seus respectivos atributos.

Segundo o processo, é necessário também especificar as entradas, saídas, pré-condições, e pós-condições dos serviços.

Para as classes dos agentes que irão representar os papéis no Mercado Livre serão necessário:

Tabela 33: Detalhes dos serviços do papel Visitante

<b>Papel</b>	Visitante
<b>Descrição</b>	Realizar buscas no sistema.
<b>Entradas</b>	Especificação da busca: celular, carro etc.
<b>Saídas</b>	O visitante fará cadastro no sistema.
<b>Pré-Condições</b>	Qualquer usuário pode acessar o sistema.
<b>Pós-Condições</b>	Os anúncios estarão disponíveis aos visitantes.  O usuário foi cadastrado com sucesso.

Fonte: elaborado pelo autor

Tabela 34: Detalhes do papel Vendedor

<b>Papel</b>	Vendedor
<b>Descrição</b>	Realizar Vendas
<b>Entradas</b>	Dados de um Anúncio: descrição, preço, contato (telefone, email), endereço.
<b>Saídas</b>	Um produto ou serviço é anunciado no sistema.
<b>Pré-Condições</b>	O usuário deve estar cadastrado e ter assumido o papel de vendedor.  O usuário deve ter uma reputação aceitável no sistema.
<b>Pós-Condições</b>	O anúncio feito está visível para todos os usuários.  O produto do anúncio foi entregue com sucesso ao comprador.

Fonte: elaborado pelo autor



Tabela 35: Detalhes do papel Comprador

<b>Papel</b>	Comprador
<b>Descrição</b>	Realizar compras
<b>Entradas</b>	Especificação da busca: celular, carro etc. Oferta por um produto.
<b>Saídas</b>	O comprador recebe o item adquirido no sistema.
<b>Pré-Condições</b>	O usuário deve estar cadastrado e assumido o papel de comprador.  O usuário deve ter uma reputação aceitável no sistema.  O comprador deve efetuar o pagamento conforme especificado pelo vendedor.
<b>Pós-Condições</b>	O produto deve ser recebido pelo comprador.

Fonte: elaborado pelo autor

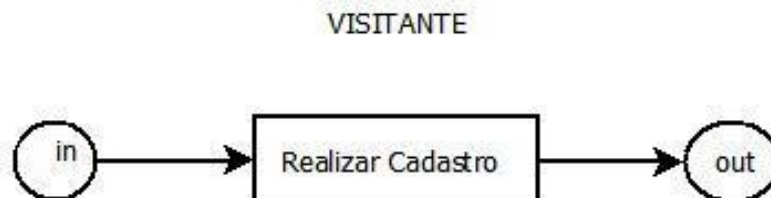
#### 5.4.2.3. Representar Modelo de Agente e Serviços usando CPNs

Nesta atividade do processo, devemos representar os modelos de agentes do sistema utilizando CPNs.

Para fazer isso, representamos cada papel (Deputy Agent) juntamente com as ações que cada um pode realizar no ambiente do Mercado Livre. Observe as figuras abaixo.

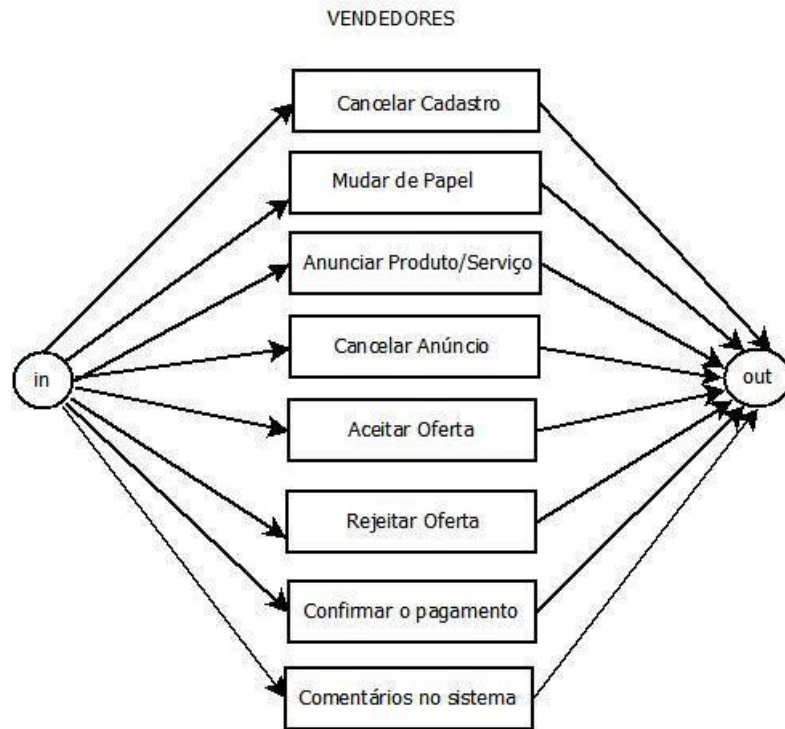
Lembrando que os papéis têm permissões de realizar suas próprias ações. Portanto, essa representação é feita a partir dos atos institucionais pré-definidos.

Figura 15: Representação da atividade do papel Visitante usando CPN.



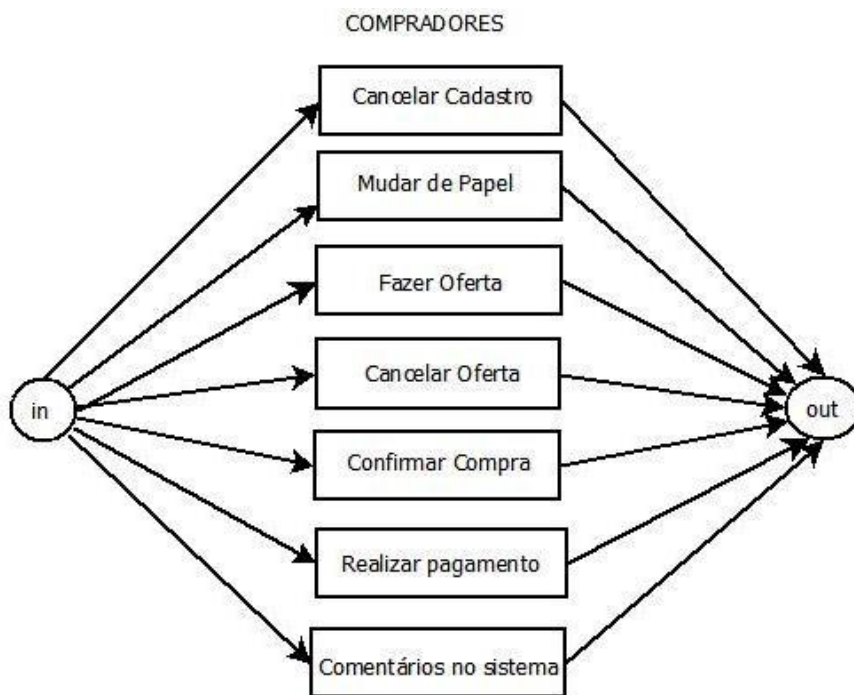
Fonte: elaborado pelo autor

Figura 16: Representação das atividades do papel Vendedores usando CPN.



Fonte: elaborado pelo autor

Figura 17: Representação das atividades do papel Compradores usando CPN.



Fonte: elaborado pelo autor

## 5.5. Fase de Implementação das CPNs

. Os documentos gerados na fase de projeto de designer servirão de entrada para os desenvolvedores terem ideia de como codificar o sistema.

Nesta fase será implementada as Redes de Petri definidas na atividade anterior do projeto de designer (Representar o Modelo de Agentes e Interações usando CPNs). Incluindo também como entrada os modelos completos de agentes e de interações.

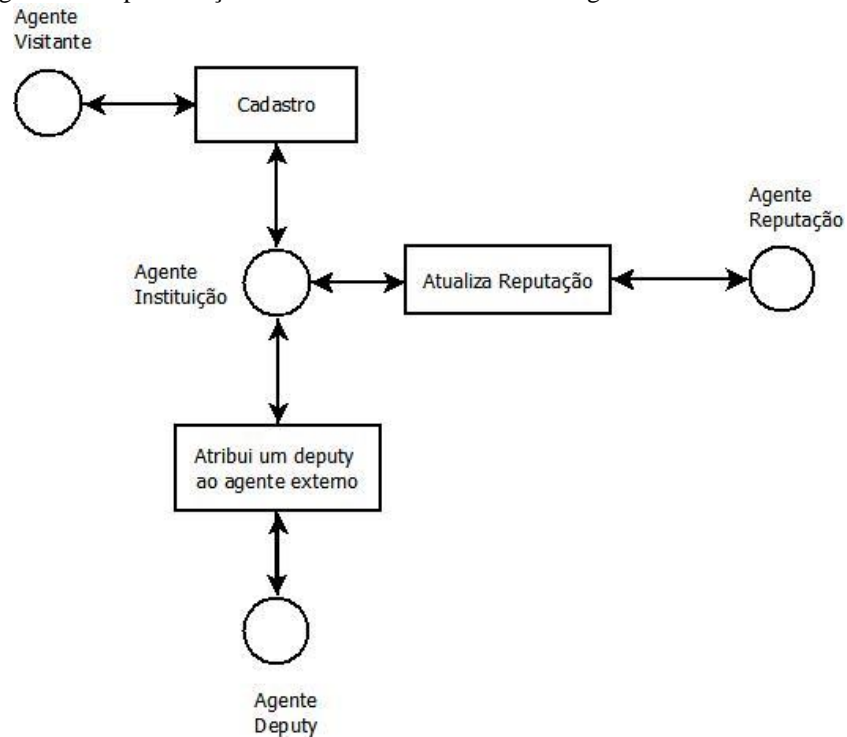
### 5.5.1. Implementar as CPNs

Podemos representar as atividades dos papéis no Mercado Livre conforme as CPNs abaixo. É importante ressaltar que as trocas de mensagens entre os agentes devem ser do tipo FIPA-ACL, segundo o IE.

As figuras abaixo representam o cenário para a interação de um agente externo (agente visitante) quando ele realizar o cadastro no sistema e os cenários para cada interação dos deputys no ambiente. Vale observar que algumas atividades são comuns para Compradores e Vendedores, portanto tratamos a representação como sendo atividades em comum para os deputys.

Primeiramente o agente externo terá que fazer seu cadastro no ambiente junto ao Agente Instituição (representando o sistema). Este irá atribuir um papel (*deputy agent*) ao novo agente e irá também atualizar seus dados de reputação junto ao agente reputação. Assim como visto na figura 17.

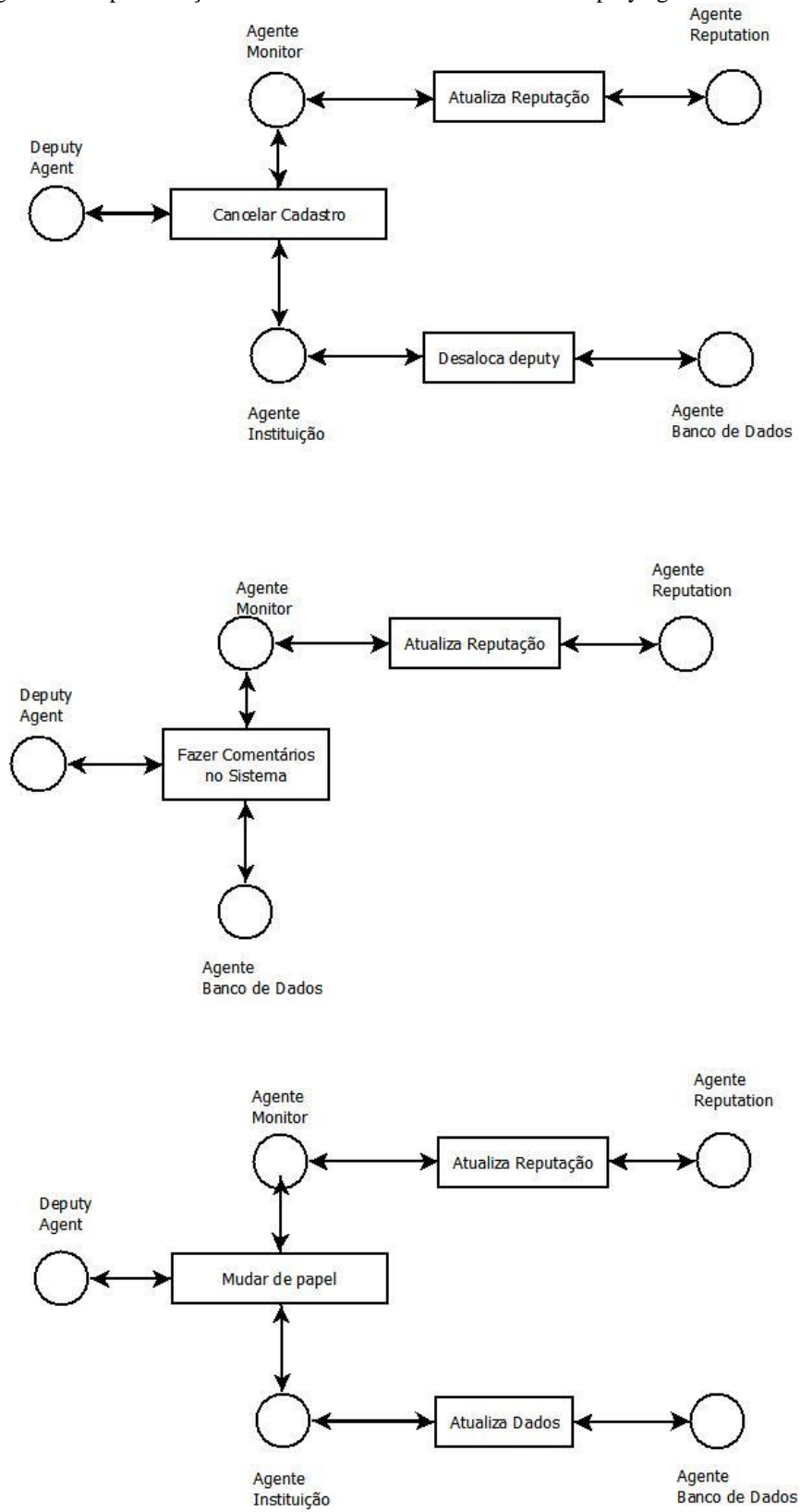
Figura 18: Representação da CPN das atividades de um agente visitante no ambiente



Fonte: elaborado pelo autor.

Tanto agentes vendedores como agentes compradores poderão fazer comentários no sistema, mudar de papel ou até mesmo cancelar seu cadastro no sistema. Para isso, o agente monitor sempre será informado dessas ações e irá atualizar seus dados de reputação junto ao agente reputação. Para mudar de papel, um agente precisará de prévia aprovação do Agente Instituição e para cancelar seu cadastro deverá informar também ao Agente Instituição, para que o mesmo possa atualizar seus dados de cadastro junto ao banco de dados. A figura 18 representa as ações que um deputy agent (seja ele um vendedor ou um comprador) poderá realizar no sistema.

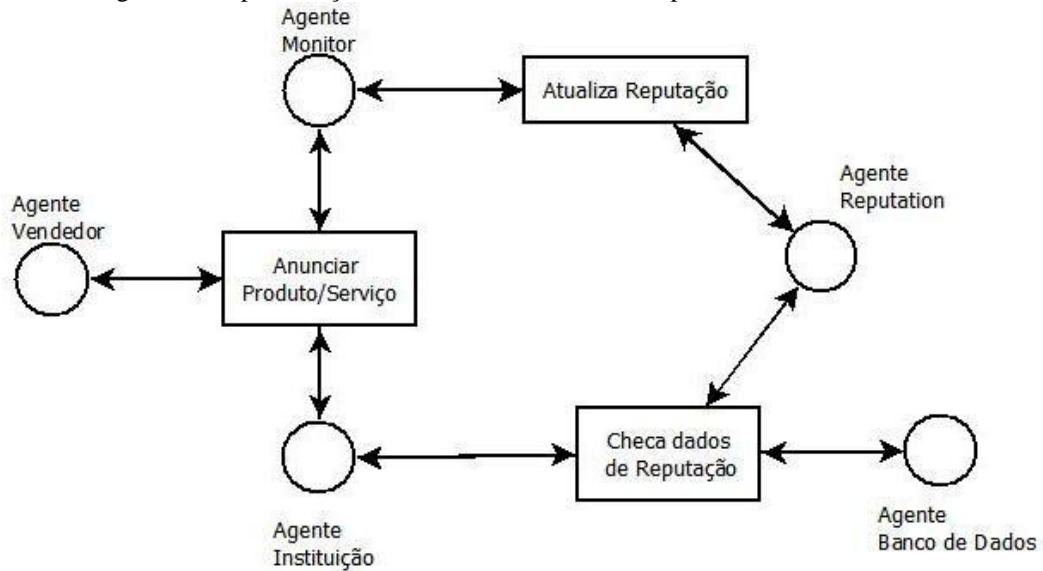
Figura 19: Representação dos cenários das atividades de um deputy agent no ambiente.



Fonte: elaborado pelo autor.

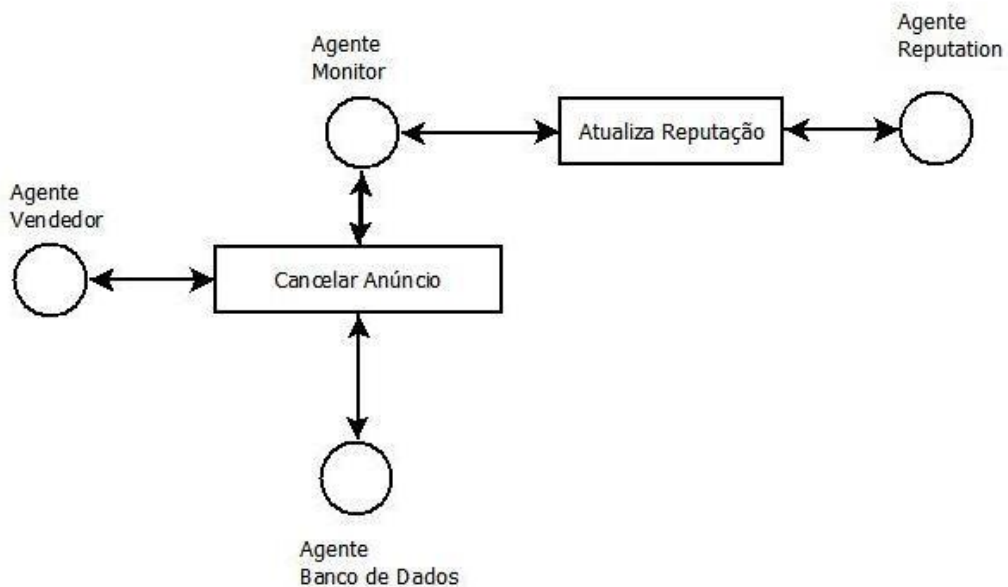
As figuras abaixo representam os cenários das interações de um vendedor no sistema. Sempre que o mesmo realizar uma ação o Agente Monitor será informado para que possa atualizar seus dados de reputação. Em algumas ações o Agente Instituição deverá ser informado para atualização de dados junto ao banco de dados.

Figura 20: Representação do cenário de Anunciar um produto de um Vendedor.



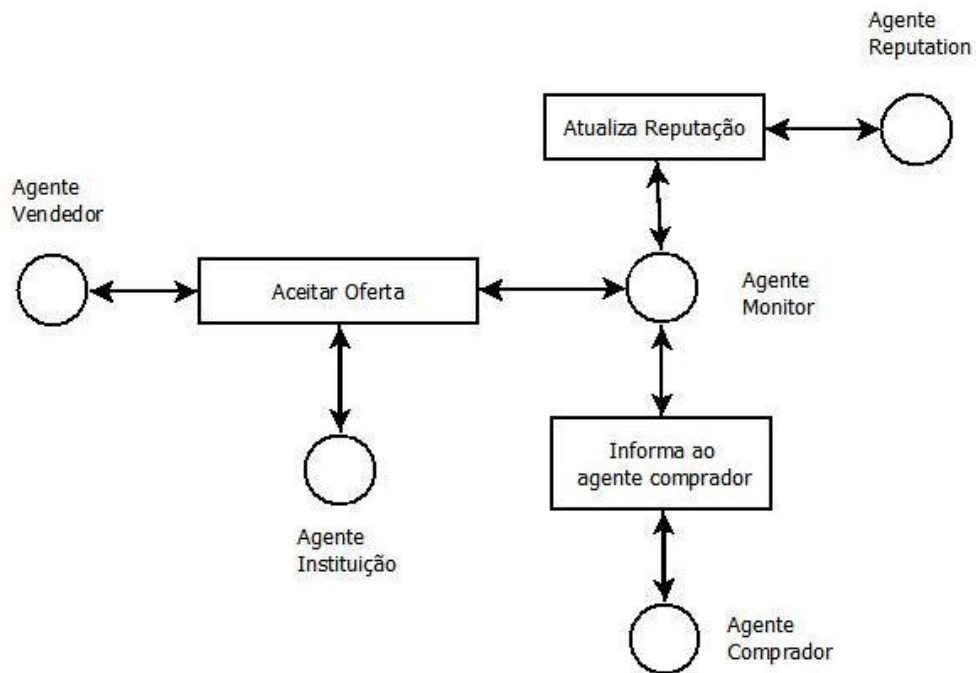
Fonte: elaborado pelo autor.

Figura 21: Representação do cenário de Cancelar Anúncio de um Vendedor



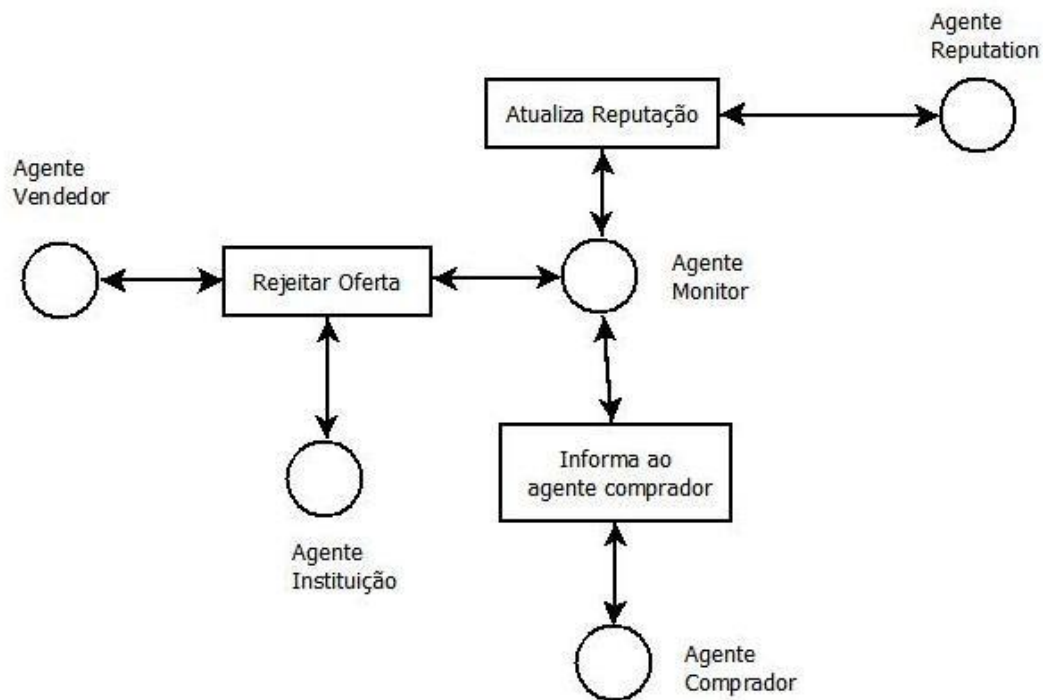
Fonte: elaborado pelo autor.

Figura 22: Representação do cenário de Aceitar Oferta de um Vendedor.



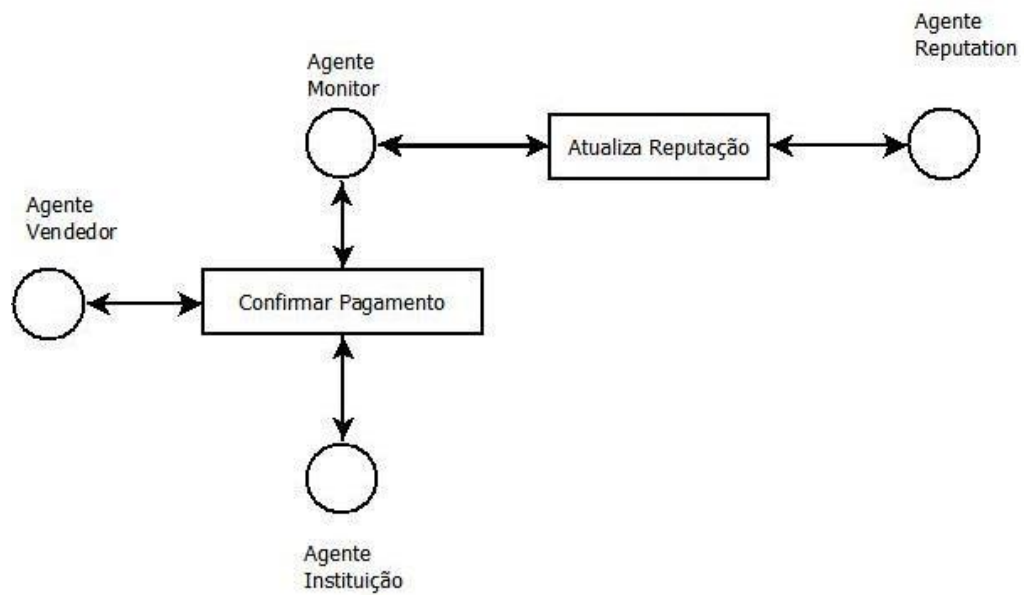
Fonte: elaborado pelo autor.

Figura 23: Representação do cenário de Rejeitar Oferta de um Vendedor



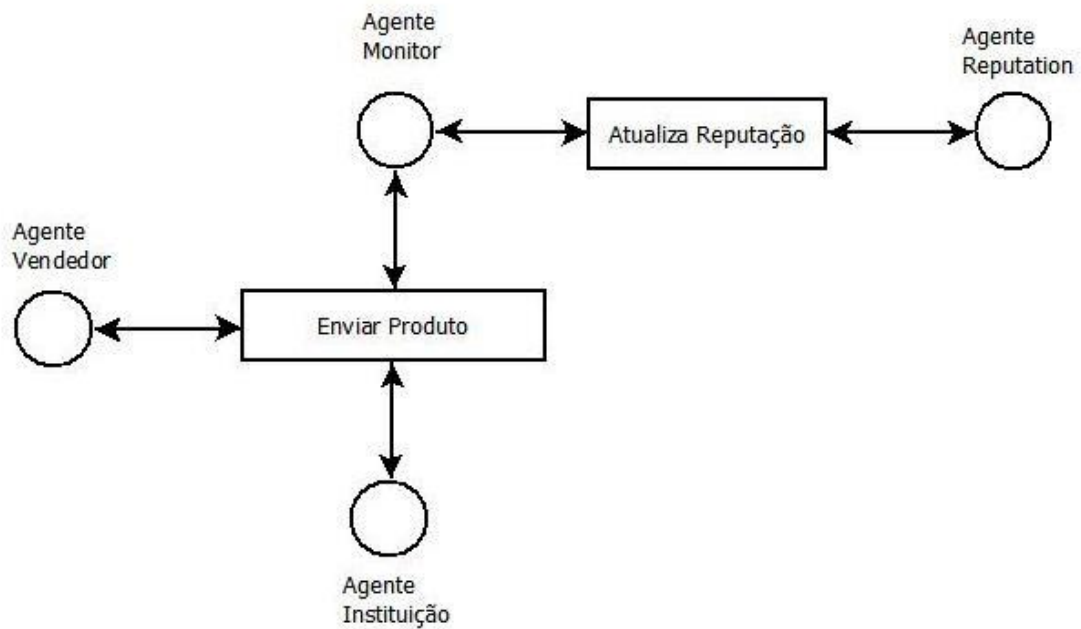
Fonte: elaborado pelo autor.

Figura 24: Representação do cenário de Confirmar Pagamento de um Vendedor.



Fonte: elaborado pelo autor.

Figura 25: Representação do cenário de Enviar Produto de um Vendedor.

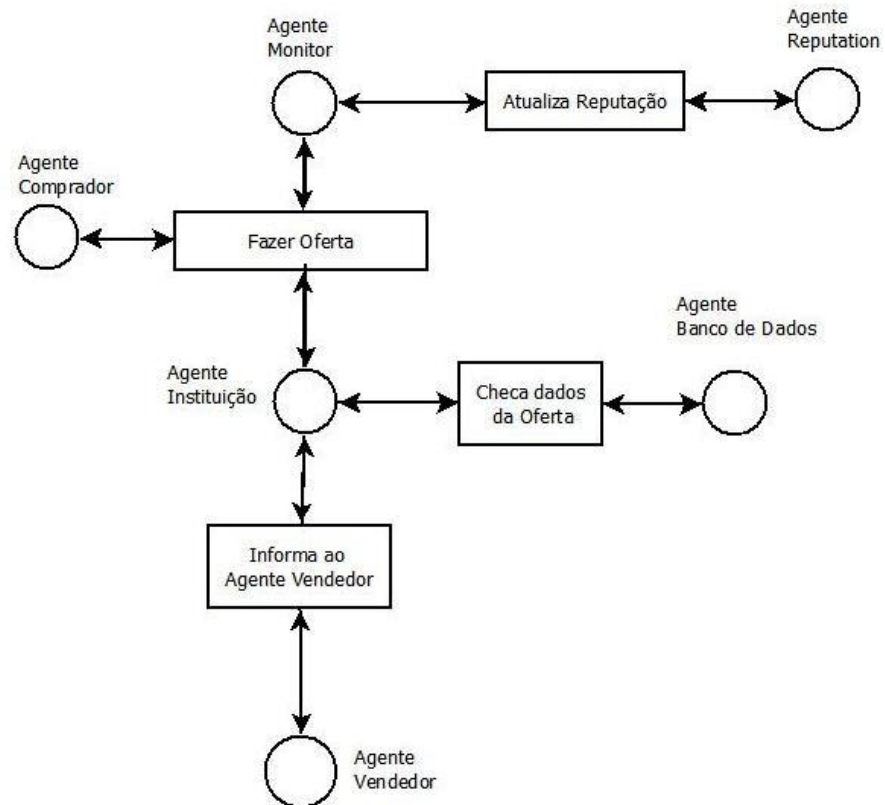


Fonte: elaborado pelo autor.



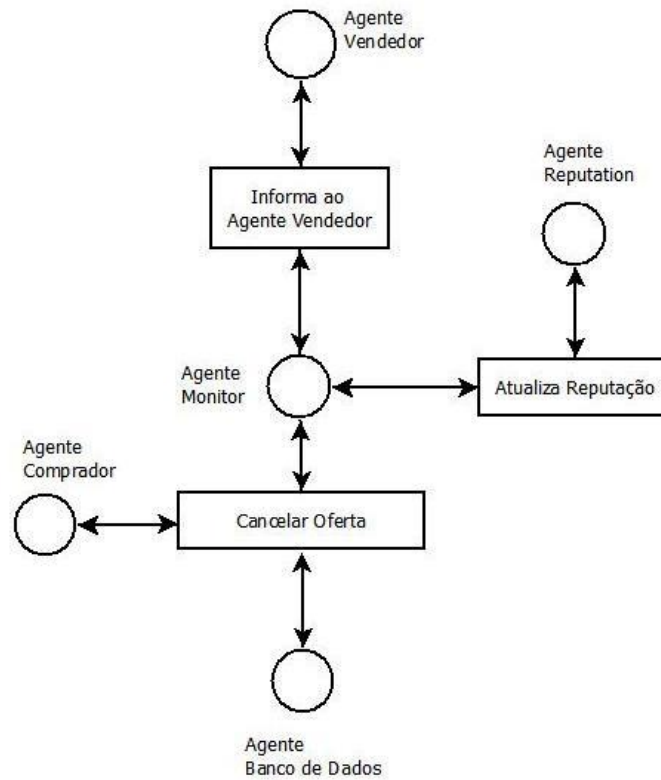
Do mesmo modo acontece com o Agente Comprador. A figura 16 representa todas as suas ações dentro do ambiente. Sempre que realizar uma ações, o seu Agente Monitor será informado para atualização dos dados de reputação. Assim como Agente Instituição deverá ser informado de algumas ações para atualização do dados junto ao banco de dados.

Figura 26: Representação do cenário de Fazer Oferta por um Comprador.



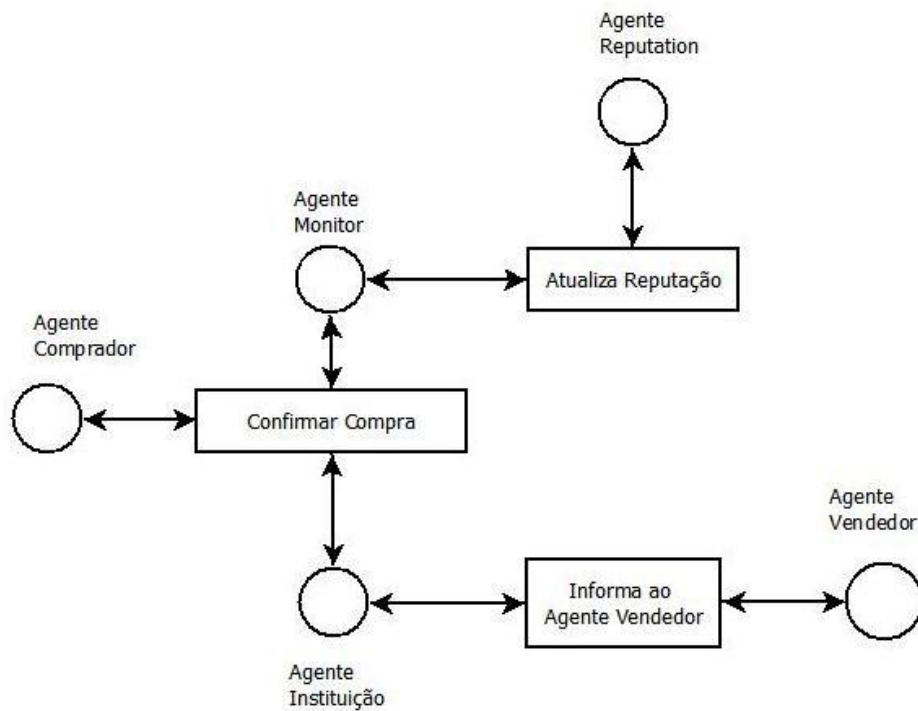
Fonte: elaborado pelo autor.

Figura 27: Representação do cenário de Cancelar Oferta por um Comprador.



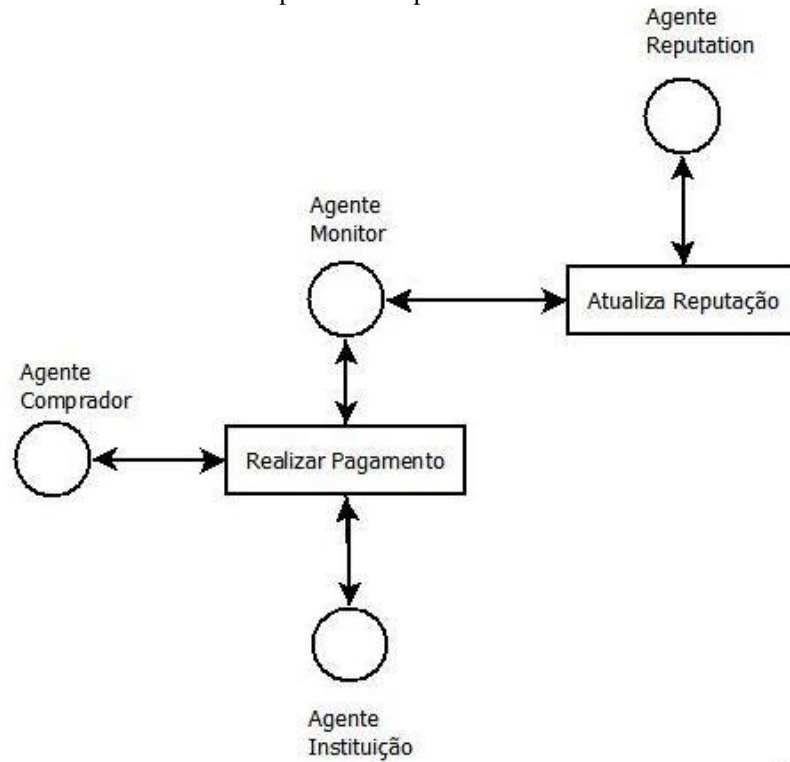
Fonte: elaborado pelo autor.

Figura 28: Representação do cenário de Confirmar Compra por um Comprador.



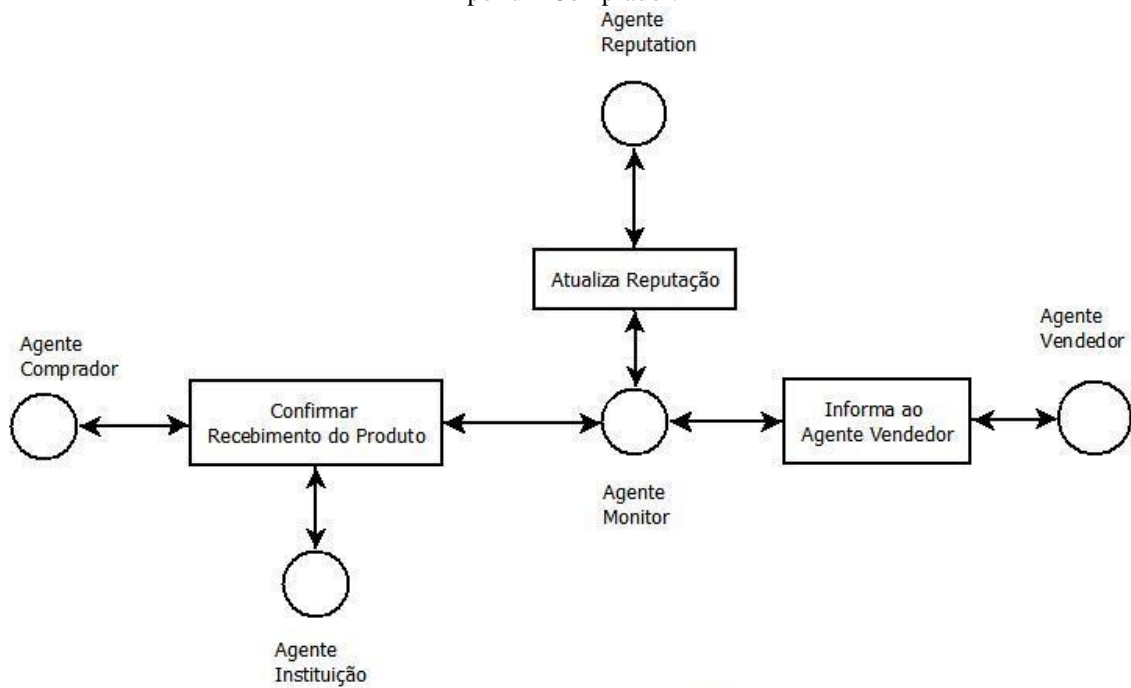
Fonte: elaborado pelo autor.

Figura 29: Representação do cenário de Realizar Pagamento por um Comprador.



Fonte: elaborado pelo autor.

Figura 30: Representação do cenário de Confirmar Recebimento de um Produto/Serviço por um Comprador.

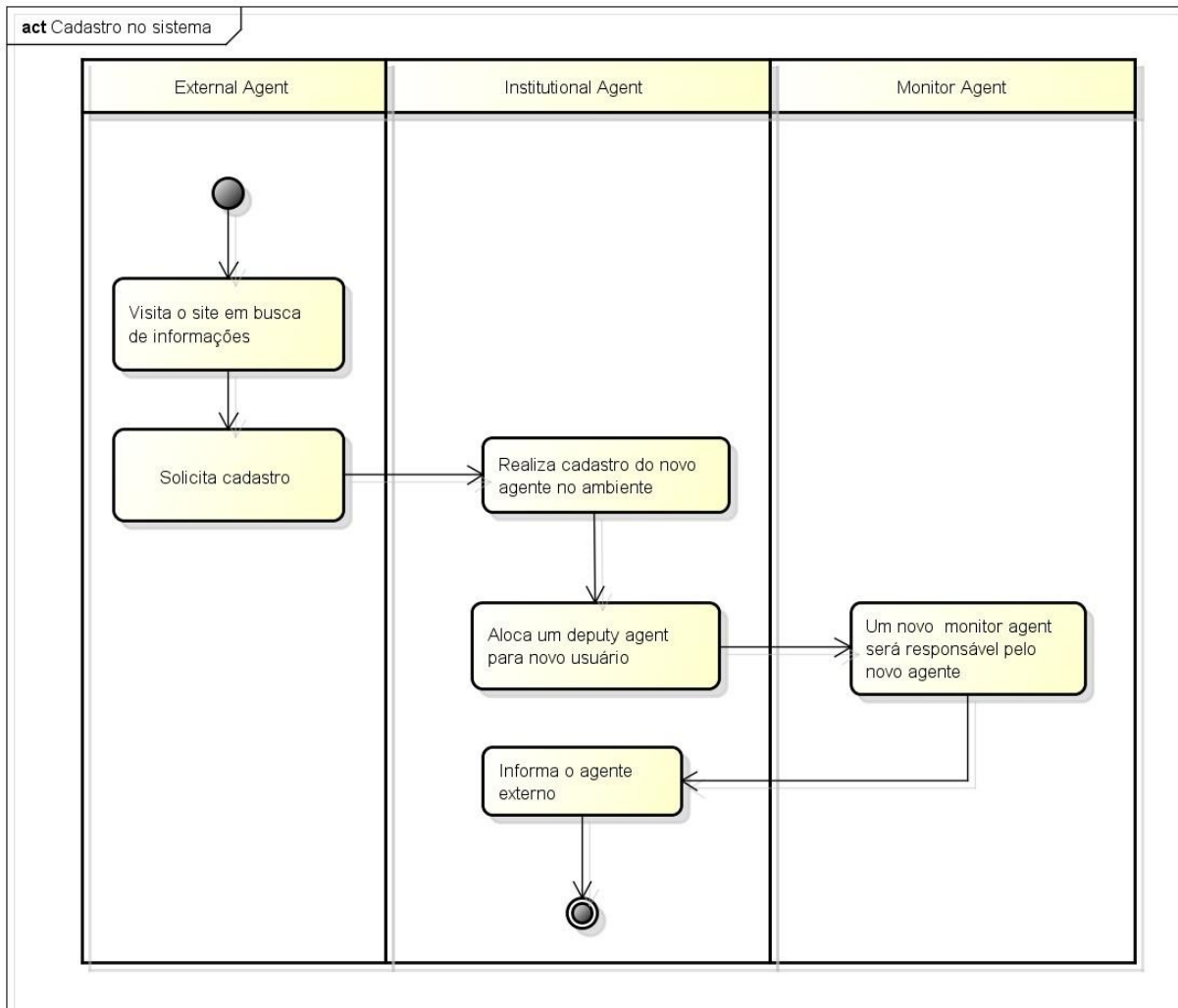


Fonte: elaborado pelo autor.

Cada atividade dos papéis no sistema possui um fluxo de trabalho diferente no ambiente. Para representá-los utilizamos diagramas de atividades, onde são representadas as interações entre as diferentes partes do sistema. Para isso, utilizamos e sugerimos a ferramenta ASTAH. Disponível em diferentes versões, ela dispõe de recursos para modelagem de diagrama de classes, de casos de uso, de atividades etc ASTAH.NET (2014).

Os visitantes podem interagir com o sistema da seguinte maneira:

Figura 31: Representação do fluxo de trabalho de um Visitante.

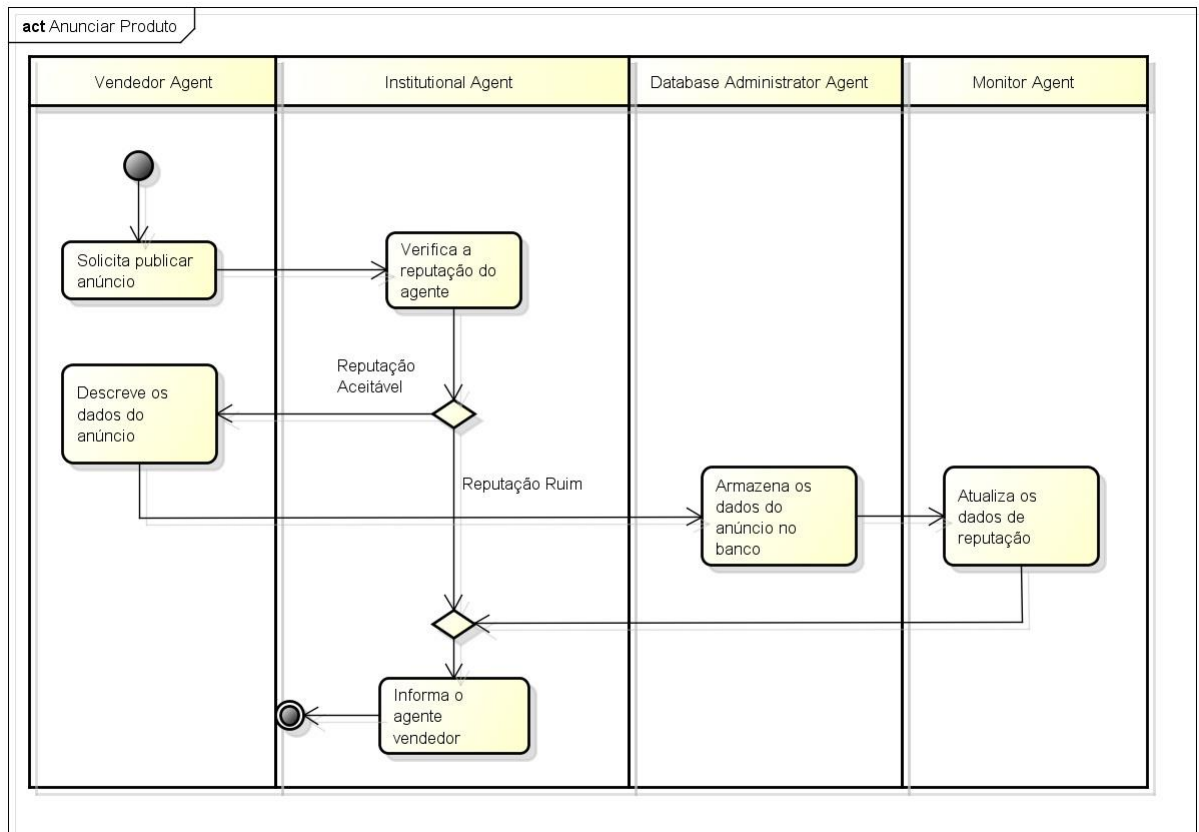


powered by astah®

Fonte: elaborado pelo autor.

Os fluxos de trabalho para as habilidades dos Vendedores são representados abaixo:

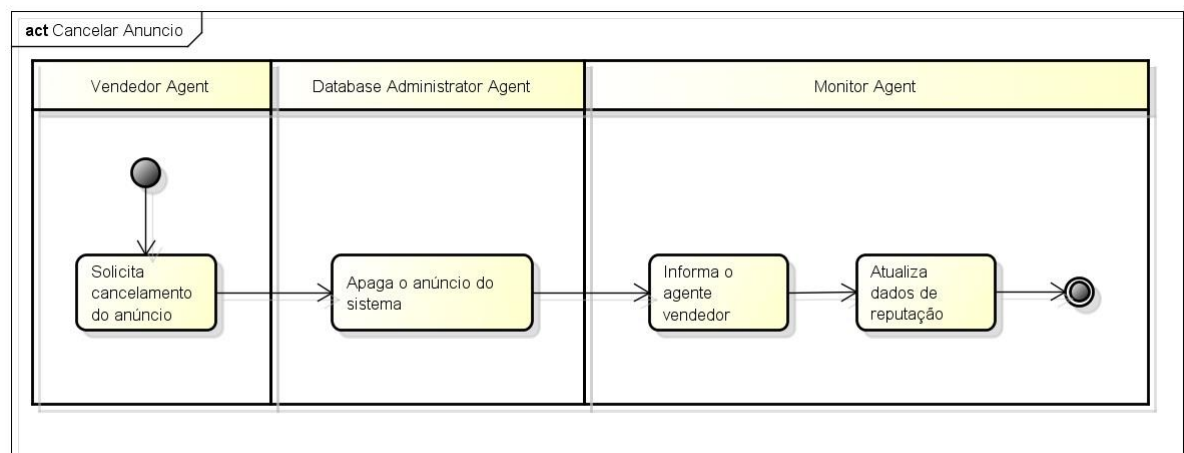
Figura 32: Representação do fluxo de trabalho do anúncio de um produto/serviço por um Vendedor.



powered by astah

Fonte: elaborado pelo autor.

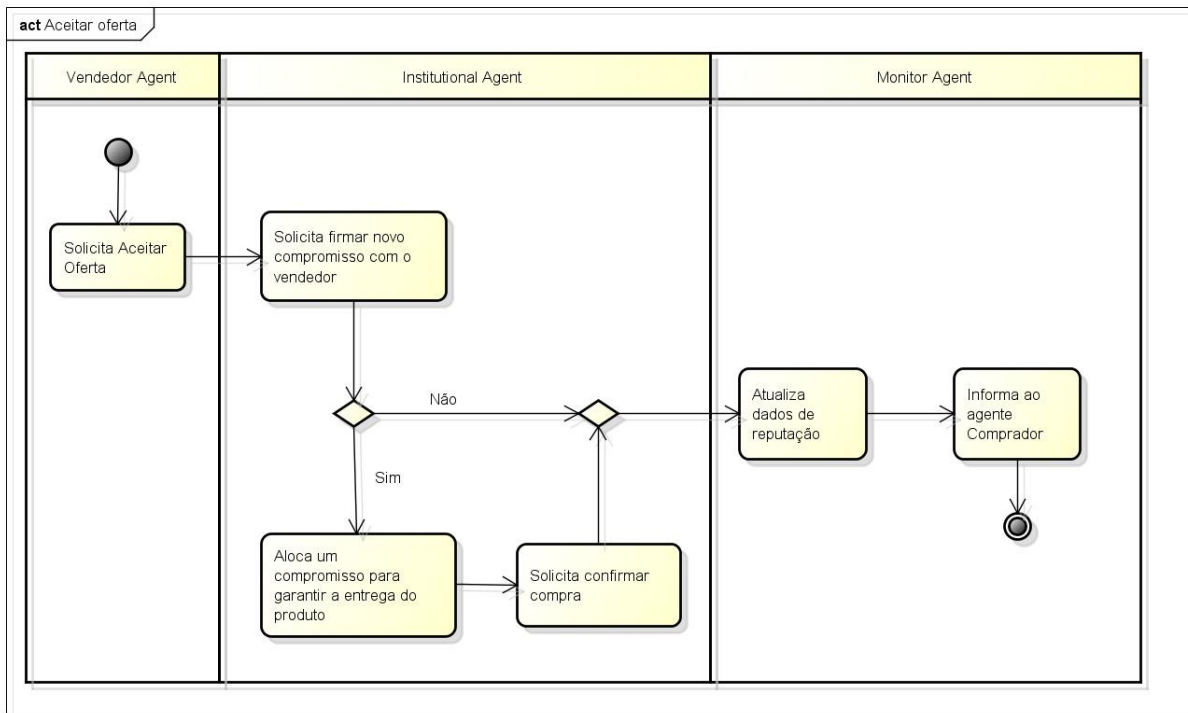
Figura 33: Representação do fluxo de trabalho de cancelar o anúncio de um produto/serviço por um Vendedor.



powered by astah

Fonte: elaborado pelo autor.

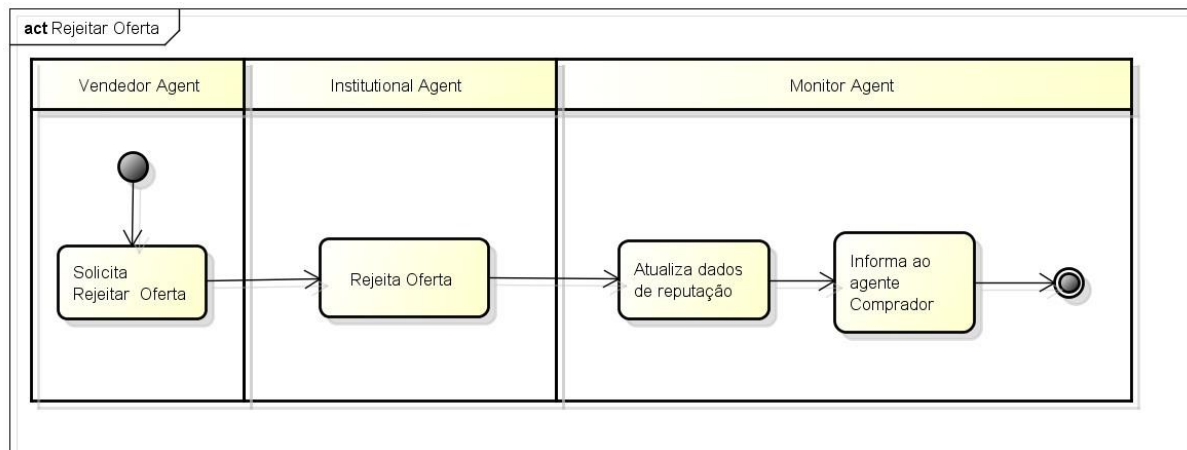
Figura 34: Representação do fluxo de trabalho de aceitar uma oferta por um Vendedor.



powered by astah

Fonte: elaborado pelo autor.

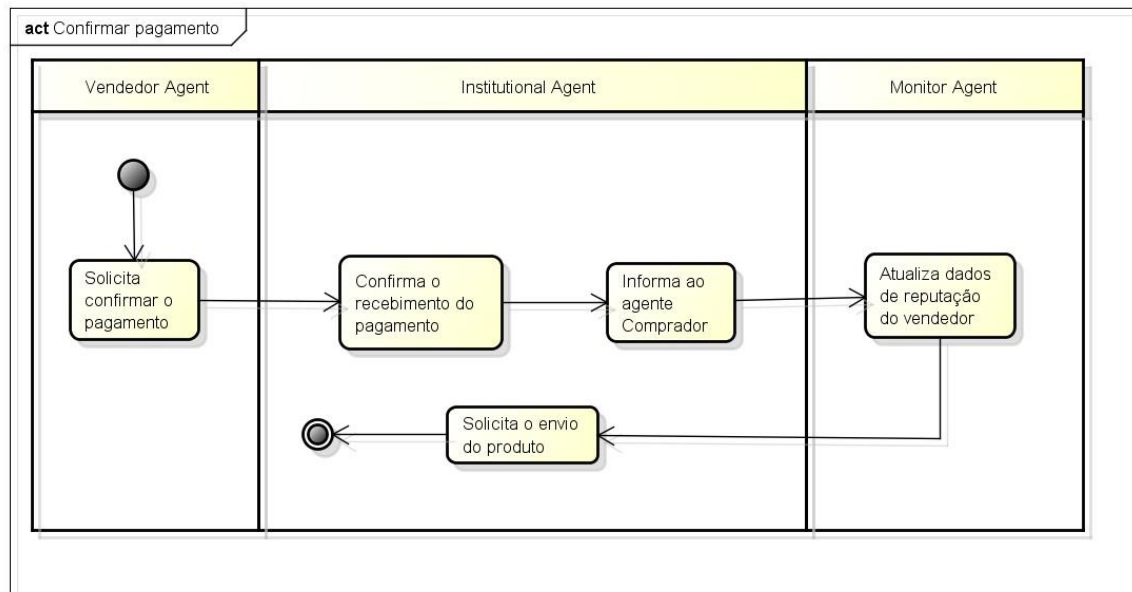
Figura 35: Representação do fluxo de trabalho de rejeitar uma oferta por um Vendedor.



powered by astah

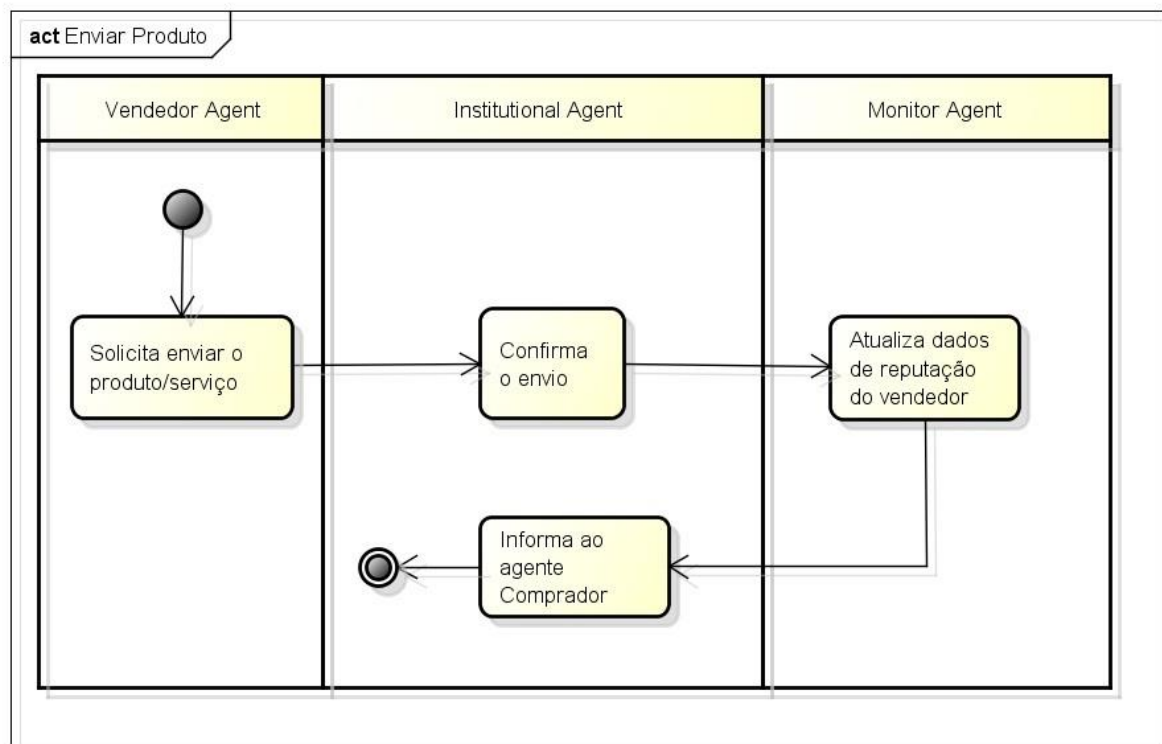
Fonte: elaborado pelo autor.

Figura 36: Representação do fluxo de trabalho de confirmar um pagamento por um Vendedor.



Fonte: elaborado pelo autor.

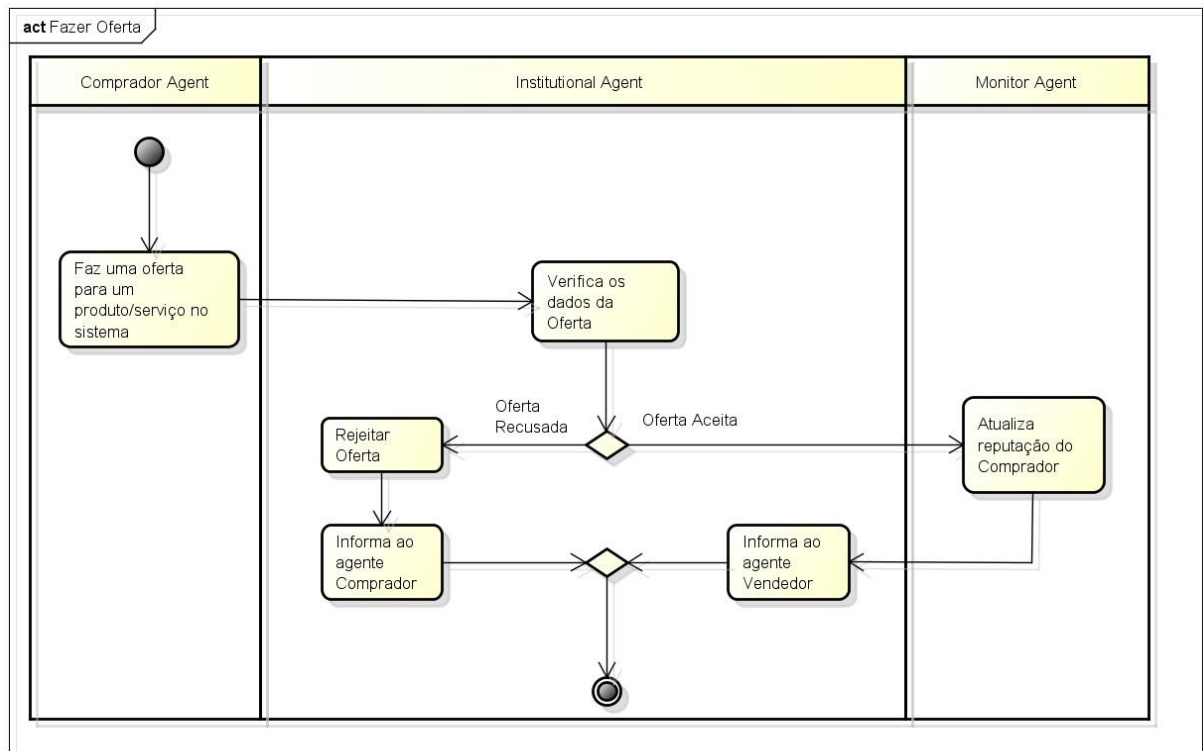
Figura 37: Representação do fluxo de trabalho do envio de um produto/serviço por um Vendedor.



Fonte: elaborado pelo autor.

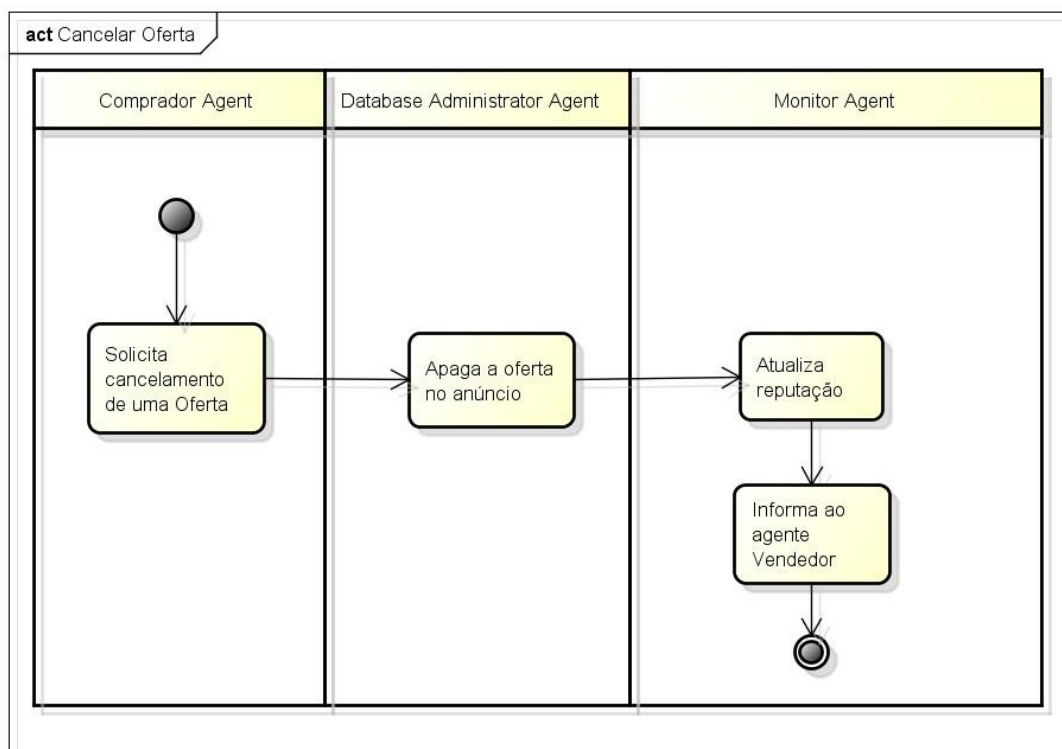
Para as habilidades dos compradores os fluxos são representados abaixo:

Figura 38: Representação do fluxo de trabalho de fazer uma oferta por um Comprador.



Fonte: elaborado pelo autor.

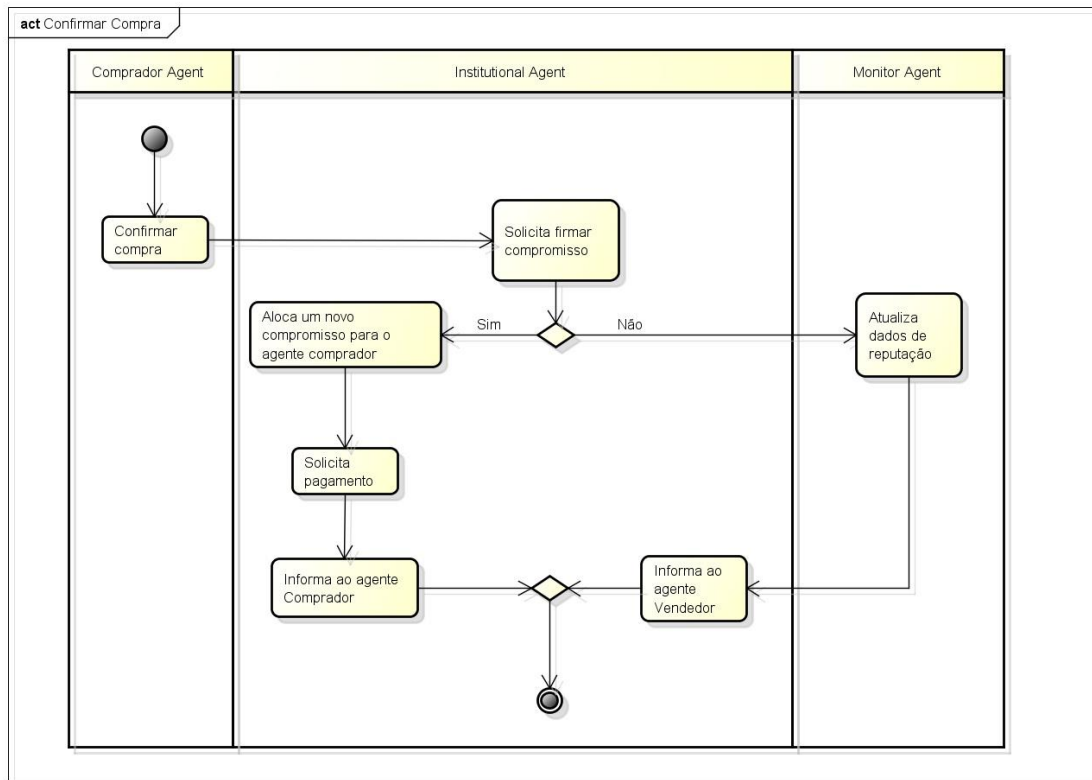
Figura 39: Representação do fluxo de trabalho de Cancelar uma Oferta por um Comprador.



Fonte: elaborado pelo autor.

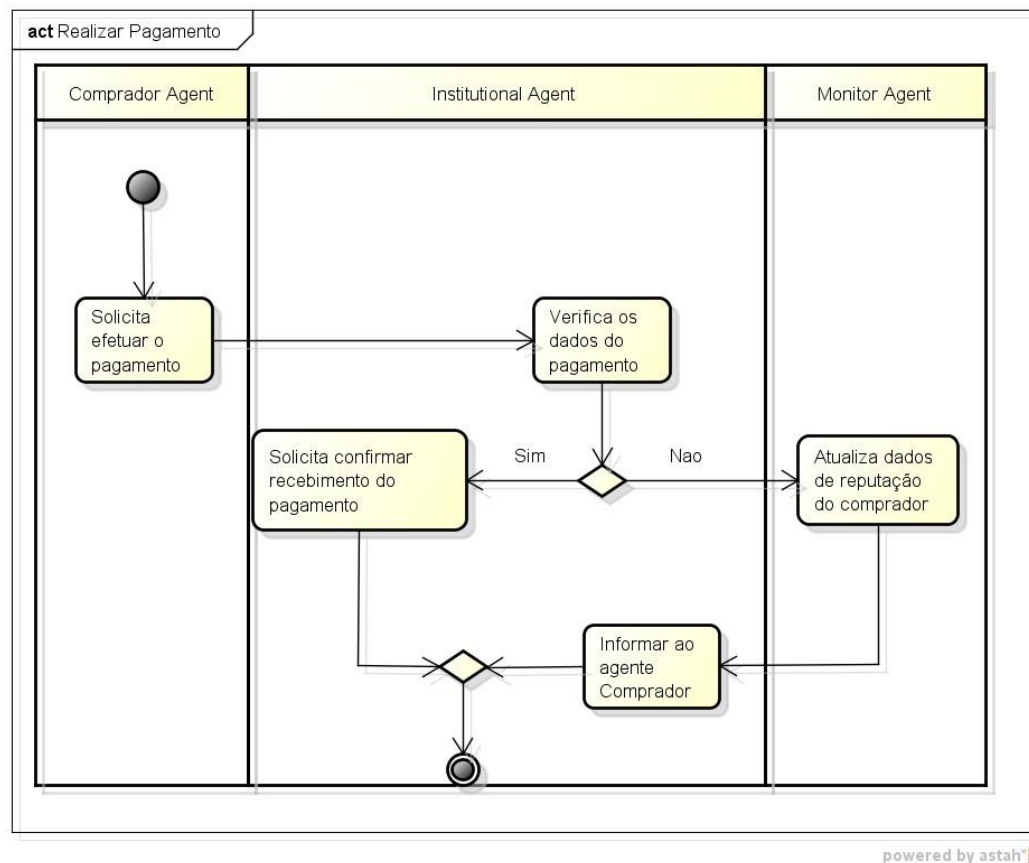


Figura 40: Representação do fluxo de trabalho de confirmar compra por um Comprador.



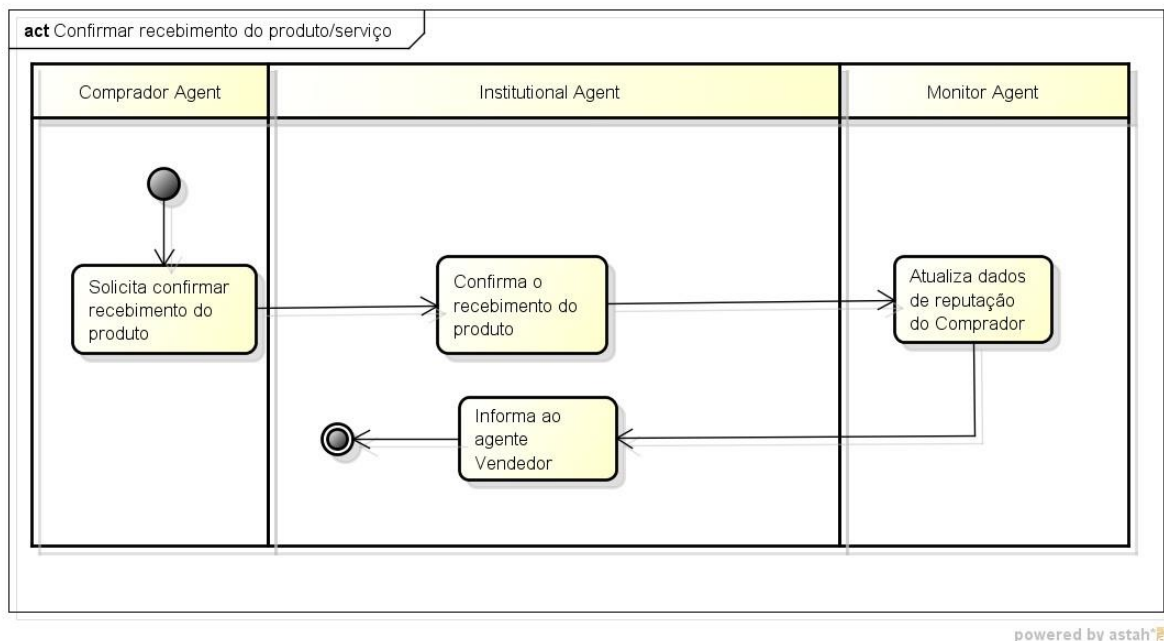
Fonte: elaborado pelo autor.

Figura 41: Representação do fluxo de trabalho de Realizar o Pagamento de por um Comprador.



Fonte: elaborado pelo autor.

Figura 42: Representação do fluxo de trabalho de Confirmar Recebimento do produto pelo Comprador.

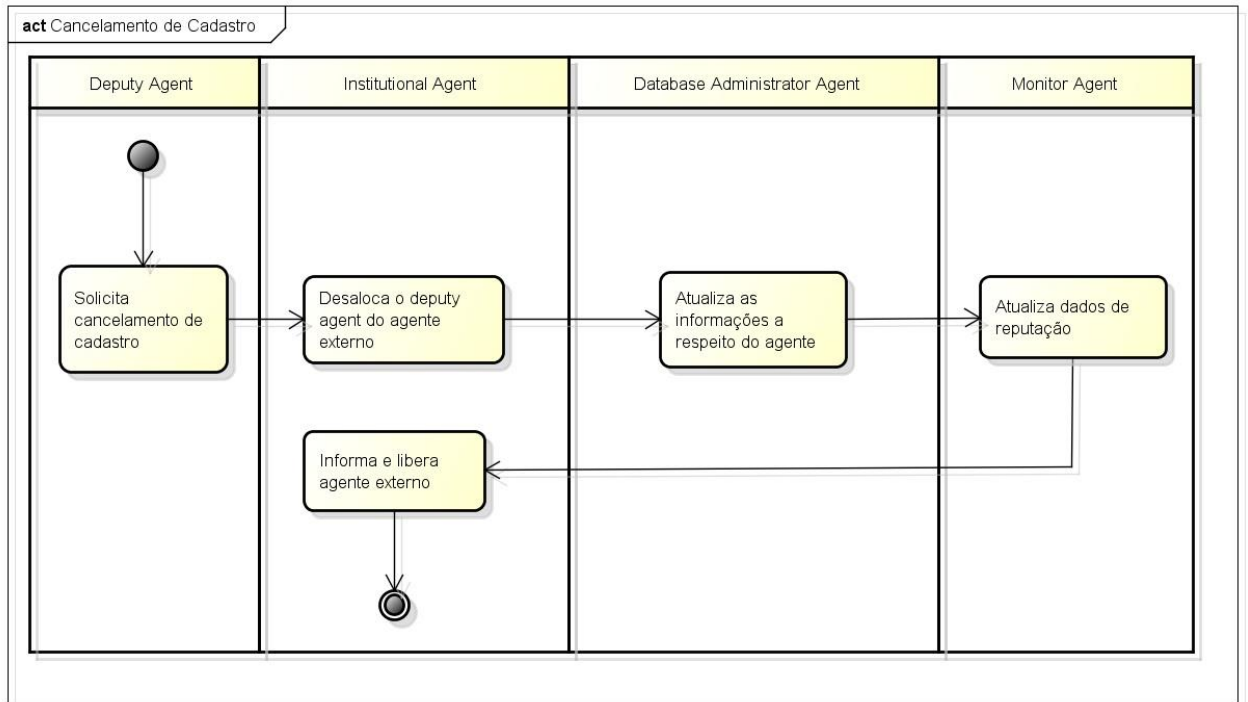


Fonte: elaborado pelo autor.

Algumas habilidades dos vendedores e dos compradores seriam melhor representadas se fossem colocadas em conjunto. Como é o caso de "Realizar Pagamento" de um comprador e a de "Confirmar Pagamento" que deve ser feito por um vendedor.

Outras habilidades são comuns aos dois papéis, portanto é possível representá-las em um único diagrama. No caso, "Cancelamento de Cadastro", "Fazer Comentários no Sistema", "Mudança de Papel". Como podemos ver nas figuras abaixo:

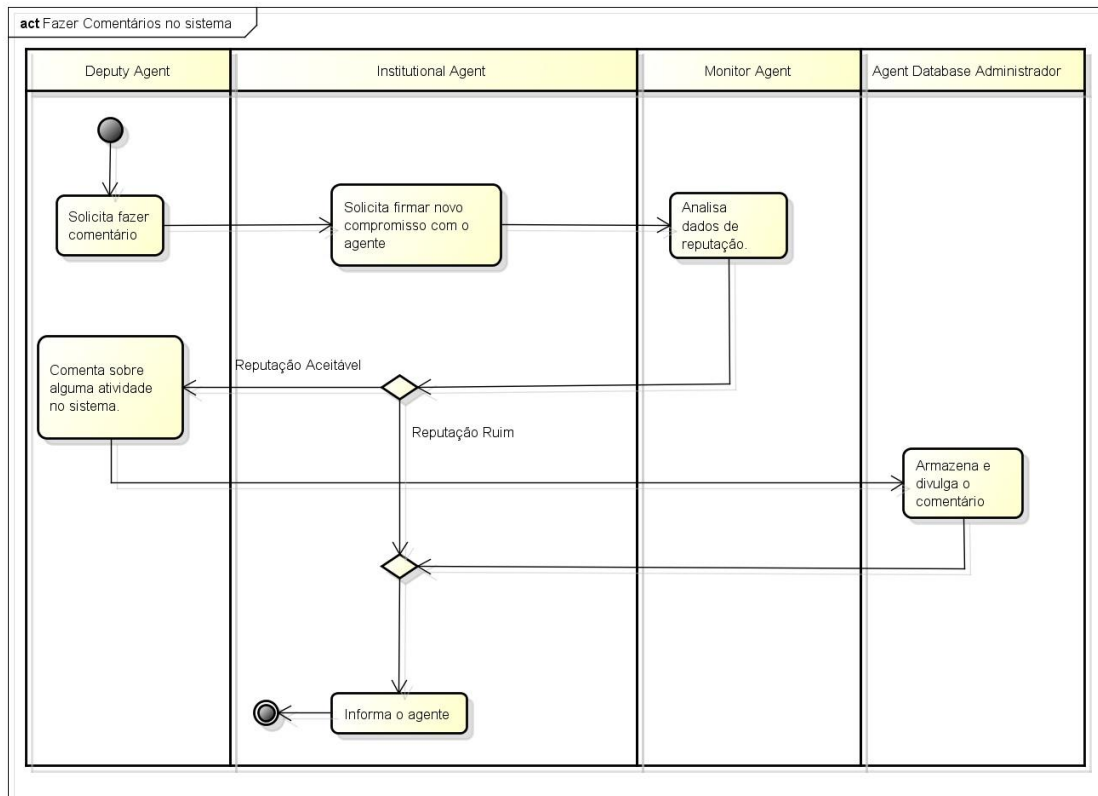
Figura 43: Representação do fluxo de trabalho do cancelamento de um cadastro por um papel.



powered by astah

Fonte: elaborado pelo autor.

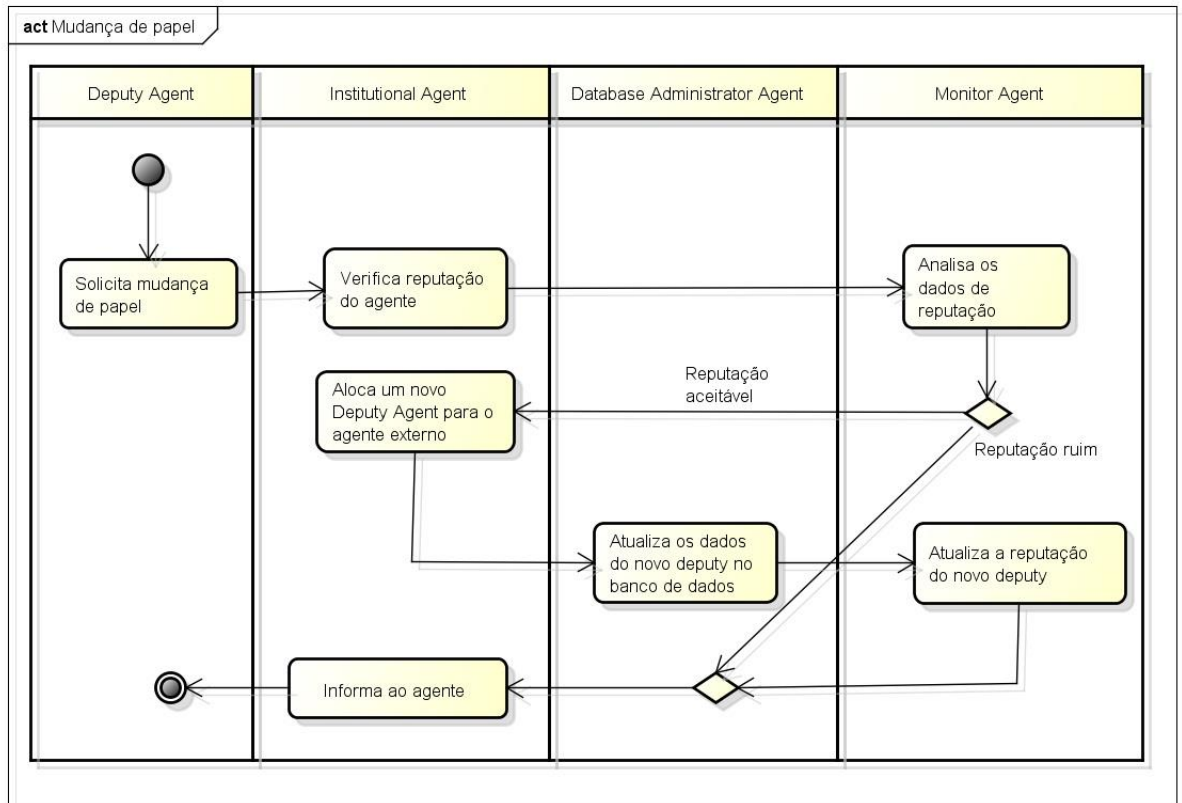
Figura 44: Representação do fluxo de trabalho de um comentário no sistema por um papel.



powered by astah

Fonte: elaborado pelo autor.

Figura 45: Representação do fluxo de trabalho da mudança de papel feita por um papel.



Fonte: elaborado pelo autor.

## 5.6. Fase de Testes do Sistema

Nesta atividade devem ser realizados os testes da execução do sistema para verificar se o comportamento está de acordo com o que foi especificado utilizando as CPNs.

### 5.6.1. Realizar Testes das CPNs

Para representar as saídas dos testes seria necessário a implementação do sistema Mercado Livre, o que foi considerado fora de escopo para este trabalho.

## 6. CONSIDERAÇÕES FINAIS

### 6.1. Conclusão

Este trabalho consistiu na definição de um processo de desenvolvimento baseado na metodologia Gaia, para sistemas multiagentes abertos utilizando a abordagem institucional proposta pelo *framework Institutional Environments*.

Para identificarmos qual metodologia seria utilizada e o que seria necessário adaptar para a criação do processo, identificamos quais os recursos utilizados pelo *framework* para solucionar os problemas apresentados. Logo após, identificamos qual metodologia melhor se adequava para solucionar esses problemas. Então escolhemos adaptar Gaia, pois também defende a tese de que SMAs podem ser desenvolvidos como sendo organizações divididas em papéis. Onde esses papéis seriam assumidos por agentes externos que ingressam no ambiente do sistema.

A partir daí, definimos os passos e atividades do processo baseando-se nas atividades propostas por Gaia para desenvolver SMAs. Com isso, nos deparamos com pontos que precisariam ser adaptados na metodologia, para que pudéssemos definir um processo adequado ao IE.

As principais adaptações feitas para o processo dizem respeito principalmente:

- Ao tratamento da autonomia dos agentes através do controle social implementado por compromissos e relações de confiança;
- Da capacidade de representar e implementar atividades concorrentes no ambiente;
- Sobre a atribuição de papéis em tempo de execução.
- A criação da atividade de representação gráfica utilizando CPNs.
- A criação das fases de implementação das CPNs e de testes do sistema.

Todas essas adaptações tratam de conceitos que não são consideradas na versão atual de Gaia, ou não são considerados diretamente. Sobre a questão da autonomia, Gaia por considera que todos os agentes que ingressarem no ambiente irão se comportar como especificado nas regras do sistema. Portanto, irá restringir demasiadamente o comportamento dos agentes e consequentemente sua autonomia. Gaia também não trata da atribuição de

papéis em tempo de execução . Uma vez assumido um papel, o agente só poderá exercer tal papel enquanto estiver no ambiente, o que não ocorre no IE. O *Institutional Environments* permite que os agentes solicitem mudança de papel, conforme seu desejo ou necessidade.

As atividades de Representação Gráfica Utilizando CPNs e as fases de Implementação das CPNs e de Testes do Sistema foram adicionadas por não serem citadas diretamente na abordagem de Gaia, mas devido a disponibilidade de ferramentas que permitem sua execução no IE, portanto um processo para desenvolvimento de SMAs Abertos utilizando o IE deveria contê-las. Especificamente em relação a testes, devido aos modelos do IE serem implementados e representados por CPNs, e devido a disponibilidade de ferramentas para análise e teste desse formalismo como o JFern (Nowostalski 2010) e o CPN Tools (CPN GROUP, 2014). Apesar de sugerir a criação de uma representação gráfica, por exemplo, para representar os modelos criados durante o processo, Gaia não cita em nenhum momento como isso deve ser feito. Assim como questões de implementação e testes do sistema.

Por fim, para demonstrar a aplicação do processo em um domínio real, realizamos a especificação do sistema de comércio eletrônico Mercado Livre utilizando a abordagem definida neste trabalho.

## **6.2. Trabalhos Futuros**

Como trabalhos futuros podemos citar: A utilização do processo proposto no desenvolvimento de sistemas para análises de resultados e realização de possíveis ajustes no mesmo; O acréscimo de iterações ao processo de maneira que o sistema seja desenvolvido incrementalmente.

**REFERÊNCIAS**

- ARCOS, J. L.; ESTEVA, M.; NORIEGA, P.; RODRÍGUEZ-AGUILAR, J. A.; SIERRA, C. **Engineering open environments with electronic institutions**. Engineering applications of artificial intelligence, 18(2), 191-204, 2005.
- ASTAH.NET. **Astah Community**. Versão 6.4.1 2014. Disponível em: <<http://astah.net/>>. Acesso em: 08 de Setembro de 2014.
- BRAFMAN, M. (1996) **On partially controlled multi-agent systems**, Journal of Artificial Intelligence Research., pp. 477-507.
- BRAGA, B.T. da R.; PEREIRA, J. L. A. **Agentes Inteligentes – Conceitos, Características e Aplicações**. Centro de Ciências Exatas e Tecnologia, Universidade da Amazônia. Belém – PA , 2001.
- BRESCIANI, P.; PERINI, A.; GIORGINI, P.; GIUNCHIGLIA, F.; MYLOPOULOS, J. **Tropos: An agent-oriented software development methodology**. Autonomous Agents and Multi-Agent Systems, 203-236. Netherlands, 2004.
- BRITO, S. R.; GAVA, T. B. ; TAVARES, O. L.; MENEZES, C. S. **Metodologias para desenvolvimento de sistemas multiagentes: Visão geral e comparação**. In: Encontro Nacional de Inteligência Artificial - ENIA, 2001, Fortaleza. ENIA. Fortaleza - CE, 2001.
- CABAC, L. **Modeling Agent Interaction Protocols with AUML Diagrams and Petri Nets**. Department of Computer Science, University of Hamburg. Hamburg – Germany, 2003.
- CASTELFRANCHI, C. **Engineering social order**. Engineering societies in the agents world. Springer Berlin Heidelberg. Berlin, 2000.
- COLLIS, J.; NDUMU, D. **Zeus Technical Manual**. Intelligent Systems Research Group, BT Labs, British Telecommunications, 1999.
- COSER, C.; ROSA, A. R. **A abordagem institucional na administração: a produção científica brasileira entre 1993 e 2003**. I Seminário de Gestão de Negócios -Fae Business School, 2004. Curitiba: FAE, 2004.
- CPN GROUP. **CPN Tools**. Versão 4.0. 2013. Disponível em: <<http://www.cpntools.org/>>. Acesso em: 08 de Setembro de 2014.
- DÁRIO, C. F. B. **Uma Metodologia Unificada para o Desenvolvimento de Sistemas Orientados a Agentes**. 2005. 176 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Departamento de Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual De Campinas, Campinas, 2005.
- DIGNUM, V.; ALDEWERELD, H.; DIGNUM, F. (2011). **On the Engineering of Multi Agent Organizations**, 12th Workshop on Agent Oriented Software Engineering, 53-65. Taipei, Taiwan.
- ECLIPSE FOUNDATION. **Eclipse Process Framework Composer**. Versão 1.5.1.3. 2011. Disponível em: <<http://www.eclipse.org/epf/>>. Acesso em: 08 de Março de 2014.





ESTEVA, M. **Electronic Institutions: from specification to development**. IIIA PhD Monography. Vol. 19, 2003.

FALBO, R. A.; BARCELLOS, M. P. **Engenharia de Software**. 2011. Disponível em: <<http://www.inf.ufes.br/~monalessa/PaginaMonalessa-NEMO/ES/NotasDeAula-EngSoftware-EngComp-Parte-I.pdf>>. Acesso em: 26 de Fevereiro de 2014.

HÜBNER, J. F.; BORDINI, R. H.; VIEIRA, R. **Introdução ao desenvolvimento de sistemas multiagentes com Jason**. *XII Escola de Informática da SBC – Paraná*. Unicentro, Guarapuava - PR, 2004.

JUCHEM, M.; BASTOS, R.M. **Engenharia de Sistemas Multiagentes: Uma Investigação sobre o Estado da Arte**. Relatório Técnico 014/2001 - Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul – Rio Grande do Sul, 2001.

GLUZ, J. C.; VICCARI, R. M. **Linguagens de Comunicação entre Agentes: Fundamentos Padrões e Perspectivas**. Jornada de Mini-Cursos de Inteligência Artificial, 3, 53-102, 2003.

MARIA, B.A.de; SIVA, V.T. da; LUCENA, C.J.P.de. **Usando MDA no Desenvolvimento de Sistemas Multi-Agentes**. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro – Brasil, 2004.

MERCADOLIVRE. **Sobre o Mercado Livre**. Disponível em <http://institucional.mercadolivre.com.br/sobre-mercadolivre/>. Acesso em 07 de outubro de 2014.

OLIVEIRA, M. A. de. **Normative spaces for open electronic institutional environments**. 2013. 246 p. Thesis (Doctor of Philosophy in Information Science) - University of Otago, Dunedin, New Zealand.

PADGHAM, L.; WINIKOFF, M. **Prometheus: A methodology for developing intelligent agents**. Agent-Oriented Software Engineering III. Springer Berlin Heidelberg, p. 174-185. 2003.

RIBAS, I. K. **Um Sistema para Compartilhamento de Arquivos em Redes P2P Utilizando uma Abordagem Multiagentes**. 2003. 85 f. Curso de Ciência da Computação, Universidade Federal De Pelotas, Pelotas - RS, 2003.

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. 2ed. Rio de Janeiro: Elsevier, 2004.

SILVA, L. A. de. M. e. **Estudo e Desenvolvimento de Sistemas Multiagentes usando JADE: Java Agent Development framework**. 2003. 97 f. Curso de Informática, Universidade de Fortaleza, Fortaleza – CE, 2003.

WOOLDRIDGE, M. **An introduction to multiagent systems**. 2nd. United Kingdom: John Wiley and Sons Ltd, 2009.

WEISS, G. **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence**. Cambridge: Gerhard Weiss, 1999.



ZAMBONELLI, F.; JENNINGS, N. R.; WOOLDRIDGE, M. **Developing multiagent systems: The Gaia methodology**. ACM Transactions on Software Engineering and Methodology (TOSEM), v. 12, n. 3, p. 317-370, 2003.

ZANUZ, L. **Visão Geral da Metodologia Tropos**. Universidade do Vale do Rio dos Sinos – UNISINOS. Rio Grande do Sul – RS, 2007.



# APÊNDICES



# APÊNDICES A

## Modelo de Ambiente



### MODELO DE AMBIENTE

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





## 1. Introdução

O modelo de ambiente em GAIA pode ser visto como uma lista de recursos, caracterizada pelo tipo de ações que os agentes podem realizar sobre ele, e associado com comentários adicionais textuais e descrições. Uma maneira simples para representá-lo é a seguinte:

<b>Nome do agente</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
<b>Nome do agente ou do papel que o agente assume no sistema.</b>	<b>O que o agente pode fazer usando o recurso.</b>	<b>Variável que irá representar o recurso legível do ambiente.</b>	<b>Breve descrição da acessibilidade do recurso.</b>
‘Agente Fulano’	‘Leitura’	‘Arquivo x’	O recurso x poderá ser lido pelo agente fulano, e modificado pelo agente cicrano.

<b>Nome do agente</b>	<b>Permissão</b>	<b>Recurso</b>	<b>Descrição</b>
<b>Nome do agente ou do papel que o agente assume no sistema.</b>	<b>O que o agente pode fazer usando o recurso.</b>	<b>Variável que irá representar o recurso legível do ambiente.</b>	<b>Breve descrição da acessibilidade do recurso.</b>
‘Agente Cicrano’	‘Leitura e Escrita’	‘Arquivo y’	O arquivo y contendo consultas de banco de dados poderá ser lido e modificado pelo agente cicrano.



# APÊNDICES B

## Modelo de Interações



### MODELO DE INTERAÇÕES

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



## 1. Introdução

Este modelo captura as dependências e as relações entre os vários papéis na organização, em termos de uma definição de protocolo para cada tipo de interação entre papéis. Em Gaia, um protocolo pode ser visto como um padrão institucionalizado de interação. Isto é, um padrão de interação que foi formalmente definido e abstraído de qualquer determinada sequência de etapas de execução, para se concentrar sobre a natureza essencial e o propósito da interação, em vez de a ordem exata de trocas de mensagens particulares.

<b>NOME DO PROTOCOLO</b>	Breve descrição textual para capturar a natureza da interação. Por exemplo, "pedido de informações", "atividade do cronograma X", "atribuir tarefa Y".
<b>INICIADOR</b>	Papel(s) responsável por iniciar a interação.
<b>INTERLOCUTOR</b>	O papel(s) de resposta com a qual o iniciador interage
<b>ENTRADAS</b>	Informações utilizadas pelo papel iniciador para iniciar o protocolo;
<b>SAÍDAS</b>	Informações fornecidas pelo protocolo de resposta durante a interação;
<b>DESCRIÇÃO</b>	Descrição textual explicando a finalidade do protocolo e as atividades de processamento implicados na sua execução.

### EXEMPLO

<b>NOME DO PROTOCOLO</b>	Encher
<b>INICIADOR</b>	Moça do café
<b>INTERLOCUTOR</b>	Máquina de café
<b>ENTRADAS</b>	Cafeteira
<b>SAÍDAS</b>	NívelCafé
<b>DESCRIÇÃO</b>	Encher a Máquina de Café



# APÊNDICES C

## Modelo de Definição de Papel



### MODELO DE DEFINIÇÃO DE PAPEL

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





## 1. Introdução

Esta atividade visa identificar as "habilidades básicas" que são necessários pela organização para alcançar os seus objetivos, bem como as interações básicas que são necessárias para essas habilidades. Este modelo não irá representar toda a estrutura do sistema, mas baseado em uma análise feita nos documentos de especificação do sistema é possível identificar características que irão se manter as mesmas durante todo o projeto.

<b>Nome do Papel</b>	Nome do papel
<b>Descrição</b>	Descrição do que o papel irá representar no sistema e qual sua função.
<b>Protocolos</b>	Especificar quais protocolos o agente irá seguir para interagir no sistema
<b>Permissões</b>	Tem o objetivo de identificar o recurso que pode ser acessado pelo papel, e os limites pelo qual esse recurso deve operar.
<b>Responsabilidades:</b>	<p>Determinam o comportamento esperado de um papel e como são associados a um recurso. Podem ser divididas em responsabilidades de:</p> <p><i>Sobrevivência</i> - descreve o comportamento de um agente, dadas certas condições.</p> <p><i>Segurança</i> - afirma que nada de ruim acontece, isto é, que um estado se mantém aceitável até que aconteça algo contrário.</p>

### EXEMPLO

<b>Nome do Papel</b>	MoçadoCafezinho
<b>Descrição</b>	Este papel tem o objetivo de garantir que a garrafa de café sempre estará cheia e avisar ao empregados quando o café estiver pronto.
<b>Protocolos</b>	EncherGarrafa; AvisarTrabalhadores; VerificarNívelCafé.
<b>Permissões</b>	Leitura: 'EstadoCafeteira' Escrita: 'NivelCafe'
<b>Responsabilidades:</b>	<ul style="list-style-type: none"> <li>• Segurança NivelCafe &gt; 0</li> <li>Senão:</li> <li>• NivelCafe ++;</li> </ul>



# APÊNDICE D

## Relatório de Testes



### RELATÓRIO DE TESTES

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



## ÍNDICE ANALÍTICO

1.	Introdução.....	1423
1.1.	Finalidade.....	1423
1.2.	Release do Sistema .....	1423
1.3.	Escopo.....	1423
1.4.	Público-alvo .....	1423
2.	Priorização de Requisitos .....	1423
2.1.	Casos de Uso e Requisitos não funcionais.....	1423
2.2.	Critérios de Priorização .....	1423
2.3.	Suítes de Teste .....	1423
3.	Ambiente de Teste – Software & Hardware.....	144
4.	Estratégias de Testes.....	144
5.	Estágios de Testes.....	144
6.	Resultado dos Testes .....	144
6.1.	Registro de Bugs .....	144
7.	Marcos e Cronograma .....	144
8.	Riscos .....	144



## **1. Introdução**

Este documento define o Plano de Testes com o objetivo de registrar o que será testado no sistema e também documentar aspectos globais relacionados a testes. Isto possibilitará uma bem-sucedida coordenação e condução de testes no projeto.

### **1.1. Finalidade**

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias para planejar e controlar o esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

### **1.2. Release do Sistema**

Visa definir prazos para a entrega das releases.

### **1.3. Escopo**

Este tópico tem como objetivo tratar da abordagem de testes caixa preta, tratar dos estágios de teste de sistema e de aceitação.

### **1.4. Público-alvo**

Visa definir todos os envolvidos com o processo de testes do sistema.

## **2. Priorização de Requisitos**

Esta seção contém os requisitos que serão testados em ordem de prioridade. Esses requisitos são divididos por suítes, em casos de uso e requisitos não funcionais conforme descrito abaixo.

### **2.1. Casos de Uso e Requisitos não funcionais**

Especificar quais requisitos não funcionais serão testados.

### **2.2. Critérios de Priorização**

Visa especificar critérios para a priorização dos testes do sistema.

### **2.3. Suítes de Teste**

As suítes foram definidas com base nos requisitos e agrupam os casos de testes relacionados a tais requisitos.





### 3. Ambiente de Teste – Software & Hardware

O ambiente de teste deve reproduzir o ambiente do cliente.

### 4. Estratégias de Testes

A estratégia deve apoiar a identificação do que será testado, sendo assim, utilizaremos a partição por equivalência e os cenários de casos de uso para identificação de casos de testes, além dos requisitos não funcionais.

### 5. Estágios de Testes

Esta seção lista os estágios de testes a serem realizados a partir da entrega do sistema pela equipe de desenvolvimento.

### 6. Resultado dos Testes

Irá apresentar os resultados obtidos com a aplicação dos testes no sistema.

#### 6.1. Registro de Bugs

Os bugs serão registrados na planilha do projeto de testes. Ao final de cada ciclo de testes, será gerado um Relatório de Testes descrevendo o trabalho realizado. Os possíveis status reportados pelo testador são:

Bloqueado: Problema que bloqueia a execução da funcionalidade.

Testado com erro: Problema detectado durante a execução da funcionalidade.

Testado sem erro: Funcionalidade presente como especificado.

### 7. Marcos e Cronograma

A tabela abaixo relaciona os principais marcos da equipe de testes.

Data	Descrição	Artefatos Gerados

### 8. Riscos

Id	Descrição	Impacto	Probabilidade	Plano de Contingência	Plano de Mitigação



# APÊNDICE E

## Documento de Definição da Estrutura Organizacional



**DOCUMENTO DE DEFINIÇÃO DA  
ESTRUTURA ORGANIZACIONAL**

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



**1. Introdução**

*[Inserir uma introdução sobre o documento]*

**2. Definição da Estrutura do Sistema**

<b>Papéis</b>	<b>Interação</b>	<b>Recursos Necessários</b>	<b>Permissões</b>	<b>Protocolo Utilizado</b>
<i>[Definir quais papéis serão envolvidos na interação]</i>	<i>[Especificar qual interação o papel irá participar]</i>	<i>[Definir quais recursos do ambiente serão necessários para a interação]</i>	<i>[Definir quais as permissões dos papéis com relação ao recursos e sobre suas ações no ambiente]</i>	<i>[Especificar quais protocolos serão necessários para cada interação]</i>



# APÊNDICE F

## Modelo de Agente



UNIVERSIDADE  
FEDERAL DO CEARÁ

**MODELO DE AGENTE**

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





## 1. Introdução

*[Inserir uma introdução sobre o documento]*

## 2. Definição do Modelo de Agente

<b>Nome do Papel</b>	<i>[Inserir nome do papel]</i>
<b>Descrição</b>	<i>[Descrição do papel]</i>
<b>Permissões</b>	<i>[Permissões que o agente que exercer um determinado papel terá]</i>
<b>Responsabilidades de Equilíbrio</b>	<i>[Condições para o comportamento do agente manter um fluxo de execução normal no ambiente]</i>
<b>Responsabilidades de Segurança</b>	<i>[Condições para caso o comportamento do agente mude dada uma condição]</i>
<b>Interações</b>	<i>[Com quais outros papéis o papel irá interagir]</i>
<b>Compromissos</b>	<i>[Compromissos que poderão ser assumidos pelo agente]</i>
<b>Atos Institucionais</b>	<i>[Relação com os atos institucionais do papel]</i>



# APÊNDICE G

## Modelo de Serviços



### MODELO DE SERVIÇOS

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



**1. Introdução**

*[Inserir uma introdução sobre o documento]*

**2. Definição do Modelo de Serviços**

<b>Papel</b>	<i>[Papel envolvido]</i>
<b>Descrição</b>	<i>[Descrição do Serviço]</i>
<b>Entradas</b>	<i>[Dados necessários para realização do serviço]</i>
<b>Saídas</b>	<i>[Saídas esperadas do serviço]</i>
<b>Pré-Condições</b>	<i>[Condições necessárias para realização do serviço]</i>
<b>Pós-Condições</b>	<i>[Condições necessárias para o serviço ser considerado realizado]</i>



# APÊNDICE H

## Relatório das Organizações do Sistema



### RELATÓRIO DAS ORGANIZAÇÕES DO SISTEMA

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





**1. Introdução**

*[Inserir uma introdução sobre o documento]*

**2. Relatório com as Organizações do Sistema**

<b>Nome</b>	<b>Descrição</b>	<b>Papéis envolvidos</b>
[Inserir o nome da organização ou organizações do sistema]	[Inserir uma descrição sobre a organização]	[Identificar quais papéis irão participar de cada organização]



# APÊNDICE I

## Relatório de Especificação da Reputação



### RELATÓRIO DE ESPECIFICAÇÃO DA REPUTAÇÃO

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



## 1. Introdução

*[Inserir uma introdução sobre o documento]*

## 2. Especificação dos Dados de Reputação

<b>Papel</b>	<b>Interação</b>	<b>Compromisso</b>	<b>Condição de Equilíbrio</b>	<b>Condição de Segurança</b>
<i>[Papel envolvido na atualização de reputação]</i>	<i>[Interação envolvida]</i>	<i>[Compromisso a ser avaliado]</i>	<i>[Condição a ser considerada para incrementar a reputação]</i>	<i>[Condição a ser considerada para decrementar a reputação]</i>



# APÊNDICE J

## Documento de Especificação dos Requisitos



## DOCUMENTO DE ESPECIFICAÇÃO DOS REQUISITOS

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





## 1. Introdução

*[Inserir uma introdução sobre o documento]*

### 1.1. Objetivos

*[Inserir objetivo do documento]*

### 1.2. Público alvo deste documento

*[Inserir público alvo]*

## 2. Requisitos do Sistema

*[Quaisquer funções, restrições ou propriedade que o sistema realizar, obedecer ou satisfazer de forma a realizar o que seus usuários desejam]*

### 2.1. Requisitos Funcionais

Id	Interessado	Descrição

### 2.2. Regras de Negócio

Id	Interessado	Descrição

### 2.3. Requisitos não funcionais

Id	Interessado	Descrição



# APÊNDICE L

## Relatório com os Atos Institucionais



UNIVERSIDADE  
FEDERAL DO CEARÁ

RELATÓRIO COM OS  
ATOS INSTITUCIONAIS

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	



**1. Introdução**

*[Inserir uma introdução sobre o documento]*

**2. Definição dos Atos Institucionais**

<b>Descrição do Ato</b>	<b>Papel envolvido</b>
<i>[Inserir uma descrição sobre o ato]</i>	<i>[Inserir qual papel irá realizar o ato]</i>



# APÊNDICE M

## Documento de Especificação dos Compromissos



### DOCUMENTO DE ESPECIFICAÇÃO DOS COMPROMISSOS

**RESERVADO**

<b>Responsável:</b>	
<b>Elaboradores:</b>	
<b>Email:</b>	





**1. Introdução**

*[Inserir uma introdução sobre o documento]*

**2. Definição dos Compromissos dos Papéis**

<b>Identificador</b>	<i>[Nome do compromisso]</i>
<b>Credor</b>	<i>[Com quais papéis será firmado o compromisso]</i>
<b>Devedor</b>	<i>[Papel que irá assumir o compromisso]</i>
<b>Estado</b>	<i>[Estado atual do compromisso]</i>
<b>Condição</b>	<i>[Condição para que o compromisso possa ser avaliado]</i>
<b>Conteúdo</b>	<i>[Estado da atividade do agente em que o compromisso irá ser avaliado como cumprido]</i>