

INSTITUTO DE TECNOLOGIA PARA O DESENVOLVIMENTO

EDSON YABUSHITA

SISTEMAS MULTIAGENTES PARA GERENCIAMENTO ACADÊMICO

CURITIBA

2017

EDSON YABUSHITA

SISTEMAS MULTIAGENTES PARA GERENCIAMENTO ACADÊMICO

Dissertação apresentada ao Programa de Pós-Graduação em Desenvolvimento de Tecnologia, Área de Concentração Geração e Transferência de Tecnologia (GTT), do Instituto de Tecnologia para o Desenvolvimento, em parceria com Faculdade Cidade Verde, como parte das exigências para a obtenção do título de Mestre em Desenvolvimento de Tecnologia.

Orientador: Prof. Dr. Sebastião Ribeiro Júnior
Coorientador: Prof. Dr. Alexandre Rasi Aoki

CURITIBA

2017

Y11s Yabushita, Edson

Sistemas multiagentes para gerenciamento acadêmico / Edson Yabushita. – Curitiba, 2017.
158 f. il. ; 30 cm.

Orientador: Prof. Dr. Sebastião Ribeiro Junior
Dissertação (Mestrado) – Instituto de Tecnologia para o Desenvolvimento, Institutos Lactec – Programa de Pós-Graduação em Desenvolvimento de Tecnologia, 2017.

Inclui Referências bibliográficas.

1. Agentes inteligentes. 2. Sistemas multiagentes. 3. AUML. I. Ribeiro Junior, Sebastião. II. Instituto de Tecnologia para o Desenvolvimento, Institutos Lactec – Programa de Pós-Graduação em Desenvolvimento de Tecnologia. III. Título.

CDD 005.7

TERMO DE APROVAÇÃO

EDSON YABUSHITA

SISTEMAS MULTIAGENTES PARA GERENCIAMENTO ACADÊMICO

Trabalho de conclusão de curso aprovado como requisito para obtenção do grau de Mestre, no Mestrado Profissional do Programa de Pós-graduação em Desenvolvimento de Tecnologia, realização do Instituto de Tecnologia para o Desenvolvimento (Institutos Lactec) em parceria com a Faculdade Cidade Verde (FCV), pela seguinte banca examinadora:



ORIENTADOR: Dr. Sebastião Ribeiro Junior

Instituto de Tecnologia para o Desenvolvimento (Institutos Lactec)



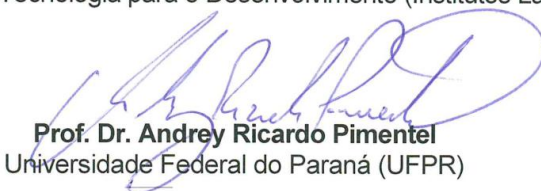
Prof. Dr. Renato De Arruda Penteado Neto

Instituto de Tecnologia para o Desenvolvimento (Institutos Lactec)



Prof.ª Dr.ª Débora Cíntia Marcilio

Instituto de Tecnologia para o Desenvolvimento (Institutos Lactec)



Prof. Dr. Andrey Ricardo Pimentel

Universidade Federal do Paraná (UFPR)

Curitiba, 27 de janeiro de 2017.

AGRADECIMENTO

Agradeço a Deus pela oportunidade de realizar mais um sonho em minha vida, pela bênção e proteção nas viagens que me concedeu.

A minha esposa Andréia e aos meus filhos Yan e Yago, pela paciência, apoio e amor que sempre dedicaram quando estive envolvido com os estudos.

Ao meu orientador Sebastião Ribeiro Junior e ao meu coorientador Alexandre Rasi Aoki pela paciência e apoio aos novos conhecimentos que adquiri durante os estudos.

Aos meus pais Minoru e Helóisa pela criação, educação, amor e bons exemplos que sempre dedicaram.

A todos os professores e funcionários do Programa de Pós-Graduação em Desenvolvimento de Tecnologia do Instituto de Tecnologia para o Desenvolvimento. Em especial ao Ricardo Salton Rosek, que comanda de forma competente a secretaria do Programa, sempre respondendo as dúvidas e prestando um ótimo serviço.

Ao diretor do Centro Técnico Educacional Superior do Oeste Paranaense (CTESOP) Fabricio Jacob Begosso e em especial a diretora Fabiany Politi Begosso Alves pelo apoio e pelo incentivo prestados.

A Secretaria Estadual de Educação (SEED) pela oportunidade oferecida.

Aos funcionários do RH do Núcleo Regional de Educação de Toledo e aos professores e funcionários do Colégio Estadual Santo Agostinho, meus sinceros agradecimentos.

Aos meus colegas de mestrado, que faziam parte do grupo de estudos, pelos esclarecimentos que me ajudaram nas disciplinas.

“Para se ter sucesso, é necessário amar de verdade o que se faz. Caso contrário, levando em conta apenas o lado racional, você simplesmente desiste. É o que acontece com a maioria das pessoas.”

Steve Jobs

RESUMO

Com a disseminação das instituições de ensino que ofertam cursos superiores, um problema cada vez mais frequente é a falta de sistemas para um acompanhamento personalizado do aluno realizado pelas coordenações de curso. Várias instituições têm necessidade em armazenar informações sobre alunos relacionadas a seus conhecimentos e aproveitamento educacional, sempre procurando inseri-los no mercado de trabalho. Existem várias informações sobre os acadêmicos, uma delas importante para classificar as competências, é o nível de conhecimentos. Um sistema automatizado para realizar tal tarefa precisa coletar características dos envolvidos, efetuar a extração dos dados mais importantes e armazená-los em uma base de dados, aguardando que aconteça uma interação para dar a resposta. A arquitetura baseada em Sistemas Multiagentes (SMA) é adequada para tal sistema, essas aplicações são projetadas em termos de entidades de software autônomos chamados agentes que de forma flexível alcançam os seus objetivos, interagindo entre si utilizando protocolos de comunicação. Neste contexto, este trabalho explora o uso de SMA focado no conceito de autonomia e interação, atuando em um sistema de gerenciamento acadêmico, capaz de realizar o acompanhamento da trajetória profissional e rendimento educacional dos alunos. Assim, o objetivo desse trabalho consiste em utilizar uma abordagem que tem sido proposta para a análise e projeto, a Metodologia para a Análise e Projeto de SMA utilizando *Framework* de Desenvolvimento de Agentes Java (JADE) que, considera a etapa de análise genérica, enquanto a fase de concepção concentra-se nas construções fornecidas pelo JADE, a Modelagem Unificada de Agentes (AUML) é o componente principal da metodologia proposta, os diagramas desenvolvidos são apresentados neste trabalho. Todos os aspectos mais importantes analisados durante o projeto do SMA estão descritos, evidenciando a importância de cada um deles. Cada protocolo proposto foi implementado e avaliado, os resultados obtidos durante as experimentações realizadas considerando cenários variados demonstram a importância do uso da metodologia utilizada e a funcionalidade dos protocolos propostos.

Palavras-chave: Agentes Inteligentes. Sistemas Multiagentes. AUML. JADE. FIPA.

ABSTRACT

With the dissemination of educational institutions offering higher education, a frequent problem is the lack of systems for a personalized follow-up of the students, carried out by the coordination. Several institutions need to store information about students regarding their knowledge and educational achievement, always seeking to insert them in the labor market. There is a lot of information about the academics and, one of them, important to classify skills is the level of knowledge. An automated system to accomplish this task needs to collect characteristics of those involved, extract the most important data and store them in a database, waiting for an interaction to occur to provide the answer. The architecture based on Multi-Agent system (M.A.S.) is suitable for such system; these applications are designed in terms of autonomous software entities called agents which, in a flexible way, reach their goals, interacting with each other using communication protocols. In this context, this study explores the use of M.A.S focused on the concept of autonomy and interaction, acting in an academic management system, capable to follow the professional trajectory and the educational achievement of the students. Using an approach that has been proposed for analysis and design, the Methodology for the Analysis and Design of M.A.S. using Java Agent Development Framework (JADE); this methodology considers the generic analysis step, while the conception phase focuses on the constructions provided by JADE. The Unified Modeling of Agents (U.M.L.) is the main component of the proposed methodology; the diagrams developed are presented in this work. Every important aspect analyzed during the M.A.S project is described, highlighting the importance of each one of them. Each proposed protocol was implemented and evaluated; the results obtained during the performed experiments, considering a variety of scenarios, demonstrate the importance of using this methodology and the functionality of the proposed protocols.

Key-words: Smart Agents. Multi-Agent Systems. U.M.L. JADE. FIPA.

LISTA DE FIGURAS

FIGURA 1 – INTERAÇÃO DO AGENTE COM O MEIO AMBIENTE	31
FIGURA 2 – ABORDAGEM EVOLUTIVA DA TECNOLOGIA DE AGENTES	33
FIGURA 3 – INTERAÇÃO ENTRE AGENTE E O AMBIENTE	35
FIGURA 4 – MODELO DE REFERÊNCIA DE AGENTES	38
FIGURA 5 – CICLO DE VIDA DE UM AGENTE	40
FIGURA 6 – CICLO DE EXECUÇÃO DE UM AGENTE.....	42
FIGURA 7 – <i>FIPA REQUEST INTERACTION PROTOCOL</i>	44
FIGURA 8 – <i>FIPA CONTRACT NET INTERACTION PROTOCOL</i>	46
FIGURA 9 – RAIZES DA METODOLOGIA ORIENTADA A AGENTES	50
FIGURA 10 – RESUMO DA FASE DE ANÁLISE	53
FIGURA 11 – EXEMPLO DE DIAGRAMA DE CLASSES	55
FIGURA 12 – CASO DE USO FUNÇÕES E RECURSOS DO ATOR.....	56
FIGURA 13 – MODELO DE CASO DE USO INTERNO	56
FIGURA 14 – RECOMENDAÇÃO DE DIAGRAMA DE CASO DE USO	57
FIGURA 15 – DESCRIÇÃO DE CASO DE USO AUML	58
FIGURA 16 – TABELA DE RESPONSABILIDADES.....	59
FIGURA 17 – DIAGRAMA DE SEQUÊNCIA AUML COM EXTENSÕES.....	60
FIGURA 18 – CRONOLOGIA ESTADO DA ARTE	61
FIGURA 19 – ARQUITETURA DO SISTEMA MULTIAGENTE.....	68
FIGURA 20 – DIAGRAMA DE CASO DE USO SMA PARA GESTÃO ESCOLAR....	70
FIGURA 21 – DIAGRAMA DE CASO DE USO MANTER AGENTE ALUNO	71
FIGURA 22 – DIAGRAMA DE CASO DE USO MANTER AGENTE EMPRESA.....	72
FIGURA 23 – CASO DE USO INTERAÇÃO AGENTES EMPRESA E ALUNO	73
FIGURA 24 – CASO DE USO MANTER AGENTE REGISTRA MÉDIA.....	74
FIGURA 25 – CASO DE USO MANTER AGENTE BUSCA MÉDIA.....	75
FIGURA 26 – CASO DE USO MANTER AGENTE MONITOR	76
FIGURA 27 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE ALUNO.....	77
FIGURA 28 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE EMPRESA.....	77
FIGURA 29 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE ALUNO E AGENTE EMPRESA.....	78

FIGURA 30 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE REGISTRA MÉDIAS.....	78
FIGURA 31 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE BUSCA MÉDIAS	79
FIGURA 32 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE MONITOR.....	79
FIGURA 33 – DIAGRAMA DE CLASSES AUML SMA	80
FIGURA 34 – INTERFACE <i>POWER DESIGNER</i> 15.1	81
FIGURA 35 – INTERFACE <i>NETBEANS</i> IDE 8.0.2	82
FIGURA 36 – INTERFACE FRAMEWORK JADE	83
FIGURA 37 – PRINCIPAIS ELEMENTOS ARQUITETÔNICOS DO JADE	83
FIGURA 38 – RELAÇÃO ENTRE COMPONENTES DA ARQUITURA JADE	84
FIGURA 39 – INTERFACE <i>MYSQL WORKBENCH</i> 5.2	85
FIGURA 40 – INTERFACE LANÇAR PERFIL CONHECIMENTOS DO ALUNO.....	88
FIGURA 41 – INTERFACE LANÇAR SOLICITAÇÕES DE EMPRESAS	89
FIGURA 42 – INTERFACE ACOMPANHAMENTO DE SOLICITAÇÕES.....	90
FIGURA 43 – INTERFACE BUSCAS REALIZADAS	91
FIGURA 44 – INTERFACE FRAMEWORK JADE	91
FIGURA 45 – MENU DE ACESSO AGENTE SNIFFER	92
FIGURA 46 – INTERFACE DO AGENTE SNIFFER.....	92
FIGURA 47 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 1.....	94
FIGURA 48 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 2.....	97
FIGURA 49 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 3.....	100
FIGURA 50 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 4.....	103
FIGURA 51 – DIÁRIO DE CLASSE	106
FIGURA 52 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 5.....	108
FIGURA 53 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 6.....	110
FIGURA 54 – DIÁRIO DE AULA CURSO DE SISTEMAS DE INFORMAÇÃO.....	112
FIGURA 55 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 7.....	113

LISTA DE GRÁFICOS

GRÁFICO 1 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 1	96
GRÁFICO 2 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 2	99
GRÁFICO 3 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 3	102
GRÁFICO 4 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 4	105
GRÁFICO 5 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 5	109
GRÁFICO 6 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 6	111
GRÁFICO 7 – MÉDIAS BAIXAS POR DISCIPLINA EXPERIMENTO 7	115
GRÁFICO 8 – MÉDIA POR ALUNO EXPERIMENTO 7	115
GRÁFICO 9 – FALTAS POR ALUNO EXPERIMENTO 7	116

LISTA DE QUADROS

QUADRO 1 – RESPONSABILIDADES DO AGENTE	69
QUADRO 2 – DETALHAMENTO CASO DE USO MANTER ALUNO	71
QUADRO 3 – DETALHAMENTO CASO DE USO MANTER AGENTE EMPRESA ..	72
QUADRO 4 – DETALHAMENTO CASO DE USO INTERAÇÃO ENTRE AGENTE ALUNO E AGENTE EMPRESA.....	73
QUADRO 5 – DETALHAMENTO CASO DE USO MANTER AGENTE REGISTRA MÉDIA.....	74
QUADRO 6 – DETALHAMENTO CASO DE USO MANTER BUSCA MÉDIA.....	75
QUADRO 7 – DETALHAMENTO CASO DE USO MANTER AGENTE MONITOR...	76
QUADRO 8 – PERFIS DE CONHECIMENTOS NO EXPERIMENTO 1.....	93
QUADRO 9 – PERFIS SOLICITADOS NO EXPERIMENTO 1	93
QUADRO 10 – PERFIS DE CONHECIMENTOS EXPERIMENTO 2	96
QUADRO 11 – PERFIS SOLICITADOS EXPERIMENTO 2.....	96
QUADRO 12 – PERFIS DE CONHECIMENTOS EXPERIMENTO 3	99
QUADRO 13 – PERFIS SOLICITADOS EXPERIMENTO 3.....	99
QUADRO 14 – PERFIS DE CONHECIMENTOS EXPERIMENTO 4	102
QUADRO 15 – PERFIS SOLICITADOS EXPERIMENTO 4.....	102
QUADRO 16 – PERFIS DE CONHECIMENTOS EXPERIMENTO 5	106
QUADRO 17 – PERFIS SOLICITADOS EXPERIMENTO 5.....	107
QUADRO 18 – MÉDIAS DE ALUNOS EXPERIMENTO 6	110

LISTA DE TABELAS

TABELA 1 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 1	95
TABELA 2 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 2	98
TABELA 3 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 3	101
TABELA 4 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 4	104
TABELA 5 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 5	109
TABELA 6 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 6	111
TABELA 7 – RESULTADOS DO AGENTE <i>SNIFFER</i> EXPERIMENTO 7	114

LISTA DE SIGLAS

AAII	– <i>Australian Artificial Intelligence Institute</i>
ACL	– <i>Agent Communication Language</i>
AIDs	– <i>Agents Identifiers</i>
AMAS	– <i>Adaptative Multi-Agent Systems</i>
AMS	– <i>Agent Management System</i>
AP	– <i>Agent Platform</i>
AUML	– <i>Agent Unified Modeling Language</i>
AVA	– <i>Ambiente Virtual de Aprendizagem</i>
BDI	– <i>Belief Desire Intention</i>
BPM	– <i>Business Process Management</i>
CASE	– <i>Computer-Aided Software Engineering</i>
CFP	– <i>Call for Proposal</i>
CT	– <i>Managing the Container Table</i>
DBA	– <i>DataBase Administrator</i>
DF	– <i>Directory Facilitator</i>
EaD	– <i>Educação a Distância</i>
FIPA	– <i>Foundation for Intelligent Physical Agents</i>
HMS	– <i>Holonic Multi-Agent Systems</i>
IMS	– <i>Information Management System</i>
JADE	– <i>Java Agent Development Framework</i>
JDK	– <i>Java Development Kit</i>
KQML	– <i>Knowledge Query and Manipulation Language</i>
MARDS	– <i>Multi-Agent Reactive Decisional System</i>
MASE	– <i>Multiagent System Engineering</i>
MTS	– <i>Message Transport Service</i>
OA	– <i>Orientado a Agente</i>
OMT	– <i>Object Modeling Technique</i>
OO	– <i>Orientado a Objetos</i>

RUP – *Rational Unified Process*
SMA – Sistema Multiagente
SQL – *Structured Query Language*
UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	27
1.1 MOTIVAÇÃO	28
1.2 JUSTIFICATIVA	28
1.3 OBJETIVOS	29
1.4 ORGANIZAÇÃO DO TRABALHO	29
2 FUNDAMENTAÇÃO TEÓRICA	31
2.1 AGENTES INTELIGENTES	31
2.1.1 Arquitetura de Agentes.....	33
2.1.2 Ambientes de Agentes	34
2.1.3 Linguagens de Comunicação de Agentes	35
2.2 FUNDAÇÃO PARA AGENTES INTELIGENTES FIPA.....	36
2.2.1 Modelo de Referência de Gerência de Agentes.....	37
2.2.2 Diretório Facilitador (DF)	38
2.2.3 Sistema de Gestão de Agentes (AMS).....	39
2.2.4 Ciclo de Vida de um Agente	39
2.2.5 Execução de Comportamentos	41
2.2.6 Protocolos de Interação entre Agentes	43
2.2.6.1 FIPA <i>request interaction protocol</i>	43
2.2.6.2 FIPA <i>contract net interaction protocol specification</i>	45
2.3 SISTEMAS MULTIAGENTES	46
2.3.1 Aplicações de Sistemas Multiagentes	47
2.4 METODOLOGIAS DE ANÁLISE E PROJETOS BASEADOS EM AGENTES.....	49
2.4.1 Metodologia para a Análise e Projeto de SMA Utilizando Jade.....	52
2.4.1.1 Fase de análise	53
2.5 MODELAGEM AUML	54
2.5.1 Extensão do Diagrama de Classe UML para Agentes	55
2.5.2 Diagrama de Caso de Uso AUML	55
2.5.3 Detalhamento de Caso de Uso AUML	58
2.5.4 Tabela de Responsabilidades de Agentes	59
2.5.5 Extensão do Diagrama de Sequência para AUML	59
3 ESTADO DA ARTE E TRABALHOS CORRELATOS.....	61
3.1 TRABALHOS CORRELATOS	65

4 ANÁLISE DO SISTEMA MULTIAGENTE	67
4.1 FASE DE ANÁLISE DO SMA	67
4.1.1 Arquitetura do SMA.....	67
4.1.2 Responsabilidades de Agentes.	69
4.1.3 Diagramas de Caso de Uso AUML	69
4.1.3.1 Diagrama de caso de uso manter agente aluno	71
4.1.3.2 Diagrama de caso de uso manter agente empresa	71
4.1.3.3 Diagrama de caso de uso interação agentes empresa e agentes aluno	72
4.1.3.4 Diagrama de caso de uso manter agente registra média	73
4.1.3.5 Diagrama de caso de uso manter agente busca e registra média.....	74
4.1.3.6 Diagrama de caso de uso manter agente monitor	75
4.1.4 Diagramas de Sequência AUML.....	76
4.1.4.1 Diagrama de sequência manter agente aluno	76
4.1.4.2 Diagrama de sequência manter agente empresa	77
4.1.4.3 Diagrama de sequência interação agente aluno e agente empresa.....	78
4.1.4.4 Diagrama de sequência manter agente registra médias.....	78
4.1.4.5 Diagrama de sequência manter agente busca médias	79
4.1.4.6 Diagrama de sequência manter agente monitor	79
4.1.5 Diagrama de Classes AUML.....	80
5 TECNOLOGIAS EMPREGADAS	81
5.1 FERRAMENTA DE MODELAGEM UML	81
5.1.1 Ferramenta de Desenvolvimento	82
5.1.2 Ambiente de Desenvolvimento Integrado	82
5.1.3 Framework JADE.....	82
5.1.4 Ferramenta para Desenvolvimento de Banco de Dados	85
5.1.5 Gerenciador de Banco de Dados.....	85
6 EXPERIMENTOS E ANÁLISE DOS RESULTADOS	87
6.1 DETALHAMENTO DOS EXPERIMENTOS	87
6.1.1 Experimentos Agente Aluno e Agente Empresa.....	87
6.1.2 Experimentos Agente Registra Médias e Agente Busca Médias	87
6.2 EXPERIMENTOS AGENTE ALUNO E AGENTE EMPRESA.....	88
6.2.1 Lançar Perfil de Conhecimentos do Agente Aluno.....	88
6.2.2 Lançar Solicitação de Empresas	89
6.2.3 Interface Acompanhamento Agentes Ativos	90

6.2.4 Interface Acompanhamento Buscas Realizadas	90
6.2.5 Interface Gráfica do Jade	91
6.2.6 Interface Gráfica do Agente Jade <i>Sniffer</i>	92
6.3 EXPERIMENTO 1 AGENTE ALUNO E AGENTE EMPRESA.....	93
6.3.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 1	93
6.3.2 Análise dos Resultados Experimento 1	95
6.4 EXPERIMENTO 2 AGENTE ALUNO E AGENTE EMPRESA.....	96
6.4.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 2	97
6.4.2 Análise dos Resultados Experimento 2	98
6.5 EXPERIMENTO 3 AGENTE ALUNO E AGENTE EMPRESA.....	99
6.5.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 3	100
6.5.2 Análise dos Resultados Experimento 3	101
6.6 EXPERIMENTO 4 AGENTE ALUNO E AGENTE EMPRESA.....	102
6.6.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 4	103
6.6.2 Análise dos Resultados Experimento 4	104
6.7 EXPERIMENTO 5 AGENTE ALUNO E AGENTE EMPRESA.....	105
6.7.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 5	107
6.7.2 Análise dos Resultados Experimento 5	108
6.8 EXPERIMENTO 6 AGENTE REGISTRA MÉDIAS E BUSCA MÉDIAS	109
6.8.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 6	110
6.8.2 Análise dos Resultados Experimento 6	111
6.9 EXPERIMENTO 7 AGENTES REGISTRA MÉDIAS E BUSCA MÉDIA	112
6.9.1 Resultados do <i>Jade Agente Sniffer</i> Experimento 7	113
6.9.2 Análise dos Resultados Experimento 7	114
7 CONCLUSÕES	117
7.1 CONCLUSÕES	117
7.2 TRABALHOS FUTUROS	118
APÊNDICE A – DIAGRAMAS DE CASO DE USO E SEUS DETALHAMENTOS.	125
APÊNDICE B – DIAGRAMAS DE SEQUÊNCIA UML	143
APÊNDICE C – DIAGRAMA HIERÁRQUICO TELAS DO SISTEMA	157
APÊNDICE D – DIAGRAMA DE CLASSES.....	158

1 INTRODUÇÃO

O acompanhamento dos alunos é uma tarefa de grande valor para as instituições de ensino, principalmente nos estabelecimentos que ofertam o ensino superior, todos os acadêmicos já são possíveis candidatos ao mercado de trabalho. Cada indivíduo tem suas próprias características, conhecimentos e preferências. Ferreira (1986) define competência como qualidade de quem é capaz de apreciar e resolver certo assunto, fazer determinada coisa, capacidade, habilidade, aptidão. Neste contexto uma empresa pode buscar informações sobre acadêmicos de acordo com seus conhecimentos e formação para preencher uma vaga de trabalho ou estágio, a inserção no mercado de trabalho e o sucesso profissional dos acadêmicos para estas empresas podem ser atribuídos, em parte, à formação recebida nessas instituições.

Além do acompanhamento profissional é muito importante para as instituições um acompanhamento pedagógico referente aos acadêmicos, principalmente para as coordenações de curso, um acompanhamento diferenciado, onde as informações dos alunos com problemas de notas ou faltas possam ser realizadas frequentemente. A ausência dessas informações impede um planejamento de atividades para suprir necessidades como, por exemplo, cursos extracurriculares e monitorias.

Diante da inexistência de sistemas que realizem esses processos em particular e por ser crescente o número de pesquisas envolvendo Sistemas Multiagentes (SMA) nas mais diversas áreas de conhecimento, surgiu o SMA para Gerenciamento Acadêmico como uma proposta para auxiliar o acompanhamento das atividades realizadas pelas coordenações de curso. Utilizando cinco agentes: o agente aluno, responsável por disponibilizar perfis e nível de experiência de cada aluno, através de pesquisas no banco de dados; o agente empresa, responsável por disponibilizar as solicitações das empresas por perfis de conhecimento, através de pesquisas no banco de dados; o agente registra médias, responsável por realizar pesquisa no banco de dados por alunos e suas médias; o agente busca médias, responsável por interagir com o agente registra médias em busca de alunos com notas abaixo da média; e o agente monitor, responsável por verificar quais agentes estão ativos na plataforma de agentes. Todo processo de negociação e interação

entre os agentes ocorre utilizando os protocolos *FIPA request interaction protocol* e *FIPA contract net interaction protocol specification*.

Esta dissertação apresenta um modelo de SMA desenvolvido com uso da metodologia para a análise e projeto utilizando *Java Agent Development Framework* (JADE), proposta por Nikraz, Caire e Bahri em 2006 que trabalha utilizando os padrões de comunicação da *Foundation for Intelligent Physical Agents* (FIPA) utilizando a extensão da *Unified Modeling Language* (UML). Segundo Booch, Rumbaugh e Jacobson (2000 p.13), UML é uma linguagem usada para especificar, visualizar, construir e documentar os artefatos de sistema através da modelagem visual. Já a *Agent Unified Modeling Language* (AUML) foi definida como extensão da UML em 2001 por Bauer, Müller e Odell. Para o desenvolvimento foi utilizada a linguagem de programação Java, com banco de dados Mysql e uma plataforma de agentes o Framework (JADE).

1.1 MOTIVAÇÃO

O elemento motivador deste trabalho são as crescentes pesquisas envolvendo SMA nas mais diversas áreas de conhecimento.

Promover melhorias no processo de acompanhamento educacional e profissional é fundamental para o rendimento dos acadêmicos e considerando as dificuldades encontradas pelos mesmos. É importante a criação de um sistema que possa auxiliar as coordenações de curso, fornecendo informações sobre o andamento das atividades desenvolvidas pelos alunos e que realize de forma autônoma a filtragem de perfis de cada aluno.

1.2 JUSTIFICATIVA

Os sistemas atuais de controle acadêmico apresentam apenas os dados finais, não permitem um acompanhamento em tempo real das atividades desenvolvidas pelos acadêmicos, também não permitem a criação de um perfil de conhecimentos de cada acadêmico, que serve como indicação para o mercado de trabalho.

A pesquisa apresentada nesta dissertação não tem a pretensão de fornecer soluções definitivas, mas sim contribuir com uma parcela de conhecimento prático

de controle acadêmico autônomo, por meio de utilização da tecnologia de agentes inteligentes, apresentando uma arquitetura que forneça subsídios para o apoio na elaboração de estratégias elaboradas pelas coordenações de curso a fim de auxiliar os alunos com problemas detectados.

1.3 OBJETIVOS

O objetivo geral deste trabalho foi o desenvolvimento de uma plataforma para acompanhar a vida acadêmica de estudantes do ensino superior, com o diferencial de filtrar perfis de competência de cada aluno para o mercado de trabalho além de realizar um acompanhamento pedagógico, ambos utilizando agentes inteligentes, oportunizando aos que possuem melhor nível a preferência na indicação.

No contexto das proposições deste trabalho, os seguintes objetivos específicos podem ser assim descritos:

- Documentar o projeto do software através do uso de diagramas da UML.
- Utilizar os conceitos da metodologia para análise e projeto utilizando JADE para a documentação do projeto do software envolvendo agentes inteligentes.
- Definir e desenvolver a arquitetura do SMA.
- Apresentar ferramentas para análise e desenvolvimento de SMA.
- Validar e testar o sistema, com os protocolos de comunicação de agentes utilizados, apresentando o resultado da comunicação entre os agentes envolvidos.

1.4 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado em 7 capítulos correlacionados. O Capítulo 1, Introdução, apresentou por meio de sua contextualização o tema proposto neste trabalho, motivação, justificativa e os objetivos.

O Capítulo 2 apresenta a fundamentação teórica, os conceitos relevantes sobre agentes inteligentes abrangendo a arquitetura, o ambiente, a comunicação, o

modelo FIPA, conceitos de SMA, metodologias de análise e projetos baseados em agentes e conceitos e diagramas da UML.

O Capítulo 3 apresenta o estado da arte sobre SMA, permitindo uma visão dos trabalhos realizados e as técnicas abordadas e os trabalhos correlatos que apresentam as seguintes características afins com o SMA proposto: a) definição de perfil e competência; b) filtragem de perfil; c) acompanhamento pedagógico; d) autonomia; e) comunicação; f) negociação entre agentes.

O Capítulo 4 apresenta a análise do SMA, destacando os métodos utilizados no desenvolvimento do sistema proposto.

O Capítulo 5 apresenta as tecnologias (materiais) que auxiliaram na elaboração dos diagramas e no desenvolvimento do SMA proposto.

O Capítulo 6 apresenta os experimentos e análise dos resultados com o propósito de validar a proposta.

O Capítulo 7 apresenta as conclusões e trabalhos futuros. Por último, apresentam-se as referências que nortearam o estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os referenciais teóricos referentes aos temas principais do trabalho, visando destacar os aspectos teóricos em relação aos principais conceitos envolvidos, propondo um alinhamento entre os diversos assuntos.

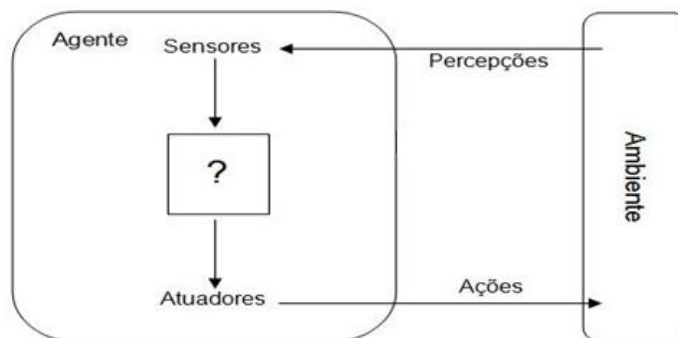
Temas abordados: a) agentes inteligentes; b) FIPA; c) SMA; d) metodologias de análise e projeto baseados em agentes; e) AUML; f) conceitos de UML.

2.1 AGENTES INTELIGENTES

“Um agente é um sistema de computador que é capaz de realizar uma ação independente” (WOOLDRIDGE; JENNINGS, 1995). Em outras palavras, um agente pode descobrir sozinho o que ele precisa fazer para alcançar seus objetivos, ao invés de ter que ser dito o que ele deve fazer.

Agente, para Russell e Norvig (1995, p. 31), é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores, conforme ilustra a FIGURA 1.

FIGURA 1 – INTERAÇÃO DO AGENTE COM O MEIO AMBIENTE



FONTE: Adaptado de: Russell e Norvig (1995, p. 32).

Wooldridge e Jennings (1995) adotaram duas definições gerais: noção fraca e noção forte de agentes. A classificação da noção fraca é feita como sendo sistemas de computador que tem o hardware ou software com autonomia, habilidade social, reatividade e proatividade. A classificação da noção forte abrange noções relacionadas ao comportamento humano tais como, crença, intenção e a obrigação.

O trabalho de Wooldridge e Jennings (1995) mostra algumas propriedades que caracterizam o agente, como:

- a) Autonomia é a capacidade de tomar decisões de forma independente.
- b) Habilidade social proporciona aos agentes interação com outros agentes para alcançar objetivos.
- c) Reatividade é quando os agentes podem perceber mudanças que ocorrem em seu ambiente, em tempo hábil para satisfazer seus objetivos.
- d) Proatividade é quando os agentes têm a capacidade de tomar iniciativas próprias para satisfazer seus objetivos.

Para Brenner, Zarnekow e Wittig (1998), os agentes são classificados segundo várias características destacando:

- Autonomia: um agente possui a capacidade de realizar seus objetivos automaticamente, sem intervenção externa, o agente tem controle no ambiente sobre suas ações e percepção sobre seu estado.
- Inteligência: a inteligência de um agente é composta por vários componentes destacando-se a base de conhecimento, a capacidade de raciocínio, a capacidade de aprender ou de se adaptar às mudanças ocorridas no ambiente.

Ainda de acordo com Brenner, Zarnekow e Wittig (1998), além das já citadas, os agentes podem também ser: a) estáticos: o processamento é feito em um único computador; b) móveis: se movem através de algum tipo de rede de computadores; c) reativos: reagem a estímulos do ambiente; d) deliberativos: conseguem planejar, negociar e coordenar suas atividades; e) sociáveis: conseguem interagir com outros agentes a fim de alcançar seus objetivos; f) pró-ativos: possuem iniciativa própria não reagindo apenas a mudanças no ambiente.

Brenner, Zarnekow e Wittig (1998) e Wooldridge (2002), utilizam várias características comuns para classificar os agentes, mas a principal delas é a autonomia que proporciona ao agente a capacidade de aprender. Sendo assim, pode-se concluir que, para classificar um programa como agente ele precisa ser autônomo. Portanto, precisa possuir inteligência que é a capacidade de aprender.

Segundo Odell (2010), “o que torna revolucionária a tecnologia de agentes é a nossa forma de pensar e usar agentes para projetar sistemas”. Odell apresenta na

FIGURA 2 a evolução das linguagens de programação. Com o aumento do nível de abstração ocorrido foi possível o desenvolvimento da linguagem orientada a agentes.

FIGURA 2 – ABORDAGEM EVOLUTIVA DA TECNOLOGIA DE AGENTES



FONTE: Adaptado de: Odell (2010, p. 3).

2.1.1 Arquitetura de Agentes

Wooldridge e Jennings (1995) dividem a arquitetura em três categorias de acordo com a construção dos agentes envolvidos:

- **Arquiteturas Deliberativas:** Os agentes atuam com pouca autonomia com modelos simbólicos exclusivos de seus ambientes.
- **Arquiteturas Reativas:** Tentam não utilizar algum tipo de modelo complexo e tomar decisões em tempo real.
- **Arquiteturas Híbridas:** Apresentam características combinadas das arquiteturas deliberativa e reativa.

Destaca-se ainda por Wooldridge e Jennings (1995) o conceito de arquitetura por camadas (horizontais ou verticais), onde os vários subsistemas dos agentes são organizados em hierarquias e interagem por diversos níveis.

Já Russell e Norvig (1995, p.40), consideram as seguintes arquiteturas de agentes:

- Agentes de reflexo simples: verificam quais ações devem utilizar com base na percepção do ambiente, sendo sempre baseadas em regras de condições.
- Agentes baseados em objetivos: os agentes através da percepção do ambiente analisam a situação atual, estabelecendo um objetivo e com base nele selecionam a ação.
- Agentes baseados em utilidade: os agentes através da percepção do ambiente identificam qual a situação desejada de acordo com sua utilidade e só a partir disto, definem a ação a ser tomada.

Segundo ChaiB-Draa e Dignum (2002), a arquitetura de agentes é dividida em quatro principais grupos: arquitetura baseada em lógica, arquitetura reativa, arquitetura *Belief-Desire-Intention* (BDI) e arquitetura de camadas.

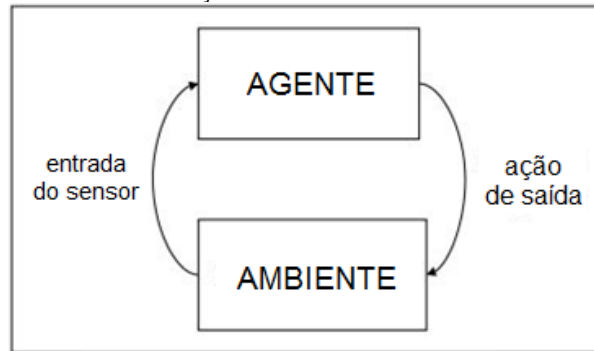
Arquitetura baseada em lógica utiliza mecanismos de raciocínio para manipulação e representação do ambiente. A arquitetura reativa utiliza mecanismo de estímulo e resposta para implementar a tomada de decisão. Na arquitetura BDI, o agente tem informação de seu ambiente através do uso de crenças, os objetivos ou metas que um agente tem que alcançar são representados pelos desejos enquanto que as intenções representam o que agente quer alcançar utilizando desejos para chegar a seus objetivos. Por fim a arquitetura de camadas sendo dividida em dois tipos: a horizontal e vertical, que interagem entre si sendo formadas por no mínimo duas camadas, as reativas e as pró-ativas.

Na camada horizontal, de acordo com Woodridge (2002, p.98), os softwares são ligados diretamente à saída de entrada e ação sensorial de cada camada. Com efeito, cada camada em si age como um agente, dando sugestões sobre qual ação deve ser executada. Já na camada vertical a entrada sensorial e a saída da ação são tratados por no máximo uma camada.

2.1.2 Ambientes de Agentes

Através do ambiente ocorre a dispersão do controle, dos dados e do conhecimento pela comunidade de agentes. Os ambientes fornecem informações, as quais são captadas pelas percepções do agente. A FIGURA 3 ilustra a definição de Wooldridge (2002) sobre o agente, que está situado em algum ambiente.

FIGURA 3 – INTERAÇÃO ENTRE AGENTE E O AMBIENTE



FONTE: Adaptado de: Wooldridge (2002, p. 16).

Segundo Russel e Norvig (1995, p. 46), os ambientes podem ser classificados quanto a suas propriedades. São elas:

- Acessível ou inacessível: Um ambiente acessível é aquele no qual o agente tem acesso a informações completas, precisas e atualizadas sobre o seu estado. Caso contrário, ele é inacessível.
- Determinístico ou não determinístico: Um ambiente é determinístico quando a ação tem um único efeito possível sobre ele. Quando não é possível determinar o estado do ambiente após uma ação o ambiente é não determinístico.
- Estático ou dinâmico: Um ambiente é estático permanece inalterado até o momento em o agente executa alguma ação. Em um ambiente dinâmico, não são só os agentes que influenciam, existem processos, alterando o estado do mesmo.
- Discreto ou contínuo: Um ambiente é discreto se houver um fixo e finito número de ações e percepções na mesma.

2.1.3 Linguagens de Comunicação de Agentes

De acordo com Bellifemine, Caire e Greenwood (2007, p. 3), um dos principais componentes de sistemas agentes é a comunicação. O agente precisa ser capaz de se comunicar e interagir com os usuários, com recursos do sistema, cooperando, negociando e assim por diante, esta interação é feita utilizando linguagens de comunicação especiais, denominadas linguagens de comunicação agente.

Segundo ChaiB-Draa e Dignum (2002), a primeira linguagem de comunicação entre agentes de uso em larga escala foi *Knowledge Query and Manipulation Language* (KQML), que possuía dois componentes: uma linguagem de representação para o conteúdo das mensagens (chamada conhecimento) e uma linguagem de comunicação (conjunto de primitivas) com o objetivo de apoiar as interações entre os agentes. Ainda sobre KQML, em 1995 Wooldridge e Jennings classificam como uma linguagem externa para comunicações entre agentes.

Além do KQML existem várias linguagens de comunicação em SMA. Destando-se a *Agent Communication Language* (ACL) controlada pela FIPA.

A FIPA é responsável por definir padrões para a implementação de comunicação entre agentes. É uma organização sem fins lucrativos com o objetivo de garantir a interoperabilidade entre agentes heterogêneos. (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2014).

2.2 FUNDAÇÃO PARA AGENTES INTELIGENTES FIPA

Segundo Bellifemine, Caire e GreenWood (2007, p.10), a FIPA foi fundada em 1996 como uma associação internacional sem fins lucrativos para desenvolver uma coleção de normas relativas à tecnologia de agentes de software. A associação elaborou especificações padrões para tecnologias de agentes de software com os seguintes princípios:

- Tecnologias de agentes fornecem um novo paradigma para resolver problemas antigos e novos;
- Algumas tecnologias de agentes chegaram a um considerável grau de maturidade;
- Para serem úteis algumas tecnologias de agentes exigem padronização;
- Normalização das tecnologias genéricas;
- A padronização dos mecanismos internos de agentes em si não é a principal preocupação, mas sim a infra-estrutura e linguagem necessários para interoperação ente agentes.

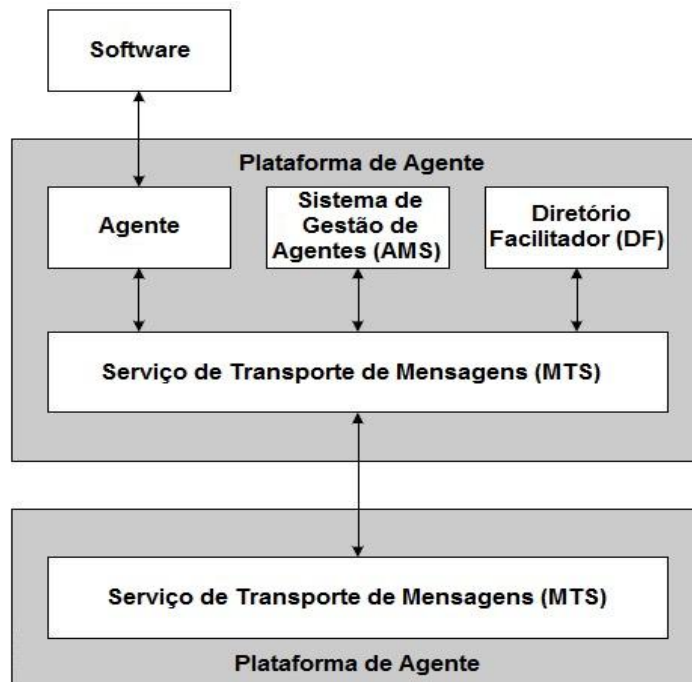
2.2.1 Modelo de Referência de Gerência de Agentes

A FIPA estabelece o modelo de referência lógica para a criação, registro, localização, comunicação, migração e retirada de agentes, representando um ambiente onde eles podem existir e atuar. Este modelo apresenta os seguintes componentes lógicos, (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2004, p. 2–3):

- **Agente:** um agente é um processo computacional que implementa a comunicação autônoma de um aplicativo, ele é um componente fundamental em uma plataforma de agentes que combina um ou mais serviços, publicados em um repositório, e um modelo de execução unificado e integrado.
- **Directory Facilitator (DF):** componente opcional da plataforma de agentes. O DF fornece serviços que ficam disponíveis para outros agentes. Os agentes podem registrar os seus serviços no DF ou consultar o DF para descobrir que serviços são oferecidos por outros agentes. Podem existir diversos DFs em uma plataforma de agentes.
- **Agent Management System (AMS):** componente obrigatório da plataforma de agentes responsável pelo controle e supervisão do acesso e uso da plataforma de agentes. Existe apenas um AMS numa plataforma de agentes. O AMS mantém um diretório que contém *Agent Identifiers* (AIDs) e endereços de transporte (entre outras coisas) para os agentes registrados na plataforma. Cada agente deve se registrar com uma AMS a fim de obter AID válido para atuar no sistema.
- **Message Transport Service (MTS):** componente que permite a comunicação entre agentes de plataformas diferentes.
- **Agent Platform (AP)** fornece a infra-estrutura física onde os agentes podem ser implantados representando as máquina(s), sistema operacional, software de suporte do agente, os componentes de gerenciamento agente FIPA (DF, AMS e MTS) e agentes.
- **Software:** coleções executáveis de instruções acessíveis através de agentes.

A FIGURA 4 mostra o modelo de referência de agentes proposto pela FIPA.

FIGURA 4 – MODELO DE REFERÊNCIA DE AGENTES



FONTE: Adaptado de: FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (2004, p. 2).

2.2.2 Diretório Facilitador (DF)

Como já visto na FIGURA 4, os agentes se comunicam através de troca de mensagens, mas para que esta comunicação aconteça eles precisam localizar os outros agentes com quem necessitam trocar informações. Para que isto possa ser feito, a plataforma JADE implementa o serviço de páginas amarelas: o DF, seguindo as especificações do padrão FIPA. Agentes que desejam divulgar seus serviços registram-se no DF, e os demais podem então buscar algum serviço desejado para que possam realizar a comunicação. (FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, 2004, p.6)

O DF pode executar as seguintes funções:

- Informações sobre um agente e os serviços que deseja publicar.
- Cancelamento parcial ou total do registro de um agente
- Modificação do registro e serviços de agente.
- Procura por agente que preste um serviço específico.

2.2.3 Sistema de Gestão de Agentes (AMS)

O AMS é responsável por gerenciar as operações de uma plataforma de agentes, tais como a criação, eliminação e também a supervisão da migração de agentes entre plataformas diferentes. (*FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS*, 2004, p. 9).

Ele representa a autoridade de controle de uma plataforma de agentes. Se ela possui várias máquinas então o AMS controla todas elas. Um AMS pode solicitar que um agente execute uma função de controle específica, como parar toda a execução na plataforma de agentes, podendo inclusive agir a força se o pedido for ignorado pelo agente. O AMS mantém um registro de todos os agentes presentes na plataforma, o registro contém o AID de todos os agentes. A descrição de um agente podem ser modificada em qualquer momento e por qualquer motivo desde que tenha autorização do AMS.

A vida de um agente em uma plataforma termina com o seu cancelamento, depois disso ele tem seu AID removido do diretório.

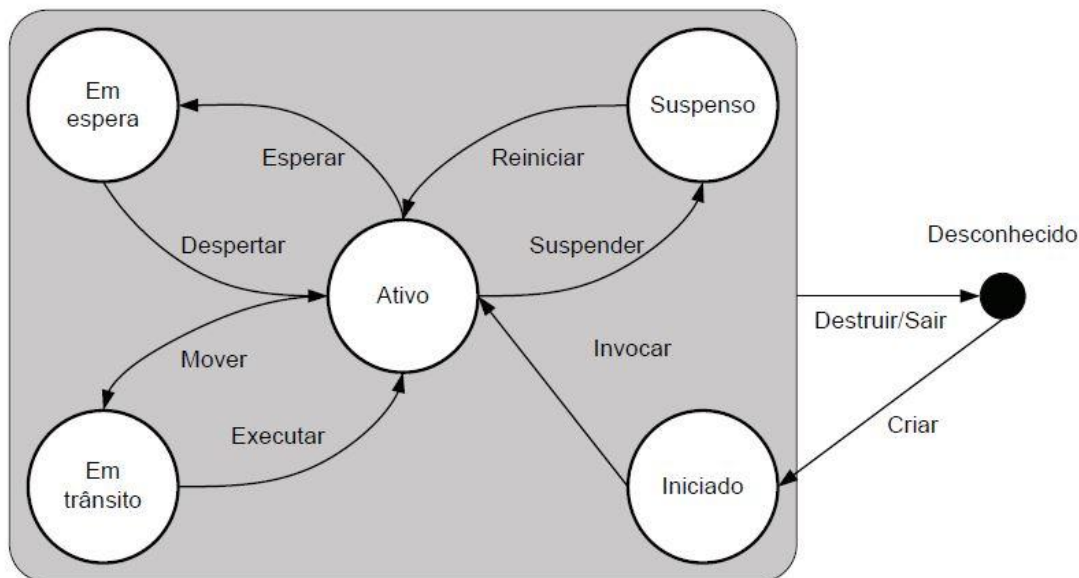
2.2.4 Ciclo de Vida de um Agente

De acordo com a *FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS* (2004, p. 9) agentes existem fisicamente em uma plataforma. Neste contexto, um agente é visto como um processo, tem um ciclo de vida gerido na plataforma sobre responsabilidade do AMS. Este ciclo de vida possui as seguintes características:

- Limitado a uma plataforma de agentes. Um agente é fisicamente gerenciado dentro de uma plataforma, como seu ciclo de vida é estático ele é limitado a uma plataforma de agentes específica.
- Independente de Aplicação. O modelo de ciclo de vida define apenas os estados e as transições do serviço do agente ao longo de sua vida, independente de qualquer aplicação de sistema.
- Orientada a Instância. O agente descrito no modelo de ciclo de vida é assumido como sendo uma instância, isto é, cada agente tem um nome único e é executado de forma independente.
- Único. Cada agente possui somente um estado de ciclo de vida

A FIGURA 5 ilustra o ciclo de vida de um agente descrito.

FIGURA 5 – CICLO DE VIDA DE UM AGENTE



FONTE: Adaptado de: FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (2004, p. 11).

A seguir são descritos os possíveis estados de um agente durante o seu ciclo de vida:

- Ativo: O agente está apto ao envio e recebimento de mensagens a outros agentes de forma normal.
- Iniciado: O agente foi iniciado mas ainda não consegue interagir com os outros agentes.
- Espera: O agente fica inativo temporariamente aguardando que um evento ocorra.
- Suspendido: O agente não é capaz de executar suas atividades, esta ação é feita pelo próprio agente ou pelo AMS.
- Trânsito: Somente agentes móveis podem entrar neste estado. Isso garante que um agente estacionário execute todas as suas instruções no local onde foi invocado.
- Desconhecido: O agente não foi reconhecido e não é capaz de interagir com o ambiente.

As transições de estado de agentes podem ser descritas como:

- Criar: A criação ou instalação de um novo agente no ambiente.
- Invocar: Invocar de um novo agente para realizar alguma tarefa.

- Destruir: Encerramento forçado de um agente, só pode ser iniciado pela AMS e não pode ser ignorado pelo agente.
- Encerrar: Encerramento não forçado de um agente, podendo ser ignorado por ele.
- Suspender: Colocar um agente em um estado suspenso. Isso pode ser iniciada pelo agente ou pela AMS.
- Prosseguir: Tirar o agente do estado suspenso. Isso só pode ser iniciado pela AMS.
- Esperar: Coloca um agente em um estado de espera. Isso só pode ser iniciado por um agente.
- Acordar: Trazer o agente de um estado de espera. Isso só pode ser iniciado pela AMS.
- Mover: Coloca o agente em um estado transitório, para ser movido de uma plataforma para outra. Isso só pode ser iniciada pelo agente.
- Executar: Trazer o agente a partir de um estado transitório. Isso só pode ser iniciado pela AMS.

Vale ressaltar que as transições mover e executar são usados apenas por agentes móveis.

2.2.5 Execução de Comportamentos

De acordo com Bellifemine, Caire e Greenwood (2007), um agente pode executar vários comportamentos concorrentemente com o uso de um escalonador que controla a execução dos comportamentos de um agente. Um comportamento é executado até que seu método de ação do agente (*action*) chegue ao fim de sua execução. O escalonador utiliza as seguintes estruturas de dados:

- Uma fila de comportamentos ativos;
- Uma fila de comportamentos bloqueados.

Bellifemine, Caire e Greenwood (2007) afirmam que execução de um agente na plataforma JADE é constituída por três níveis básicos:

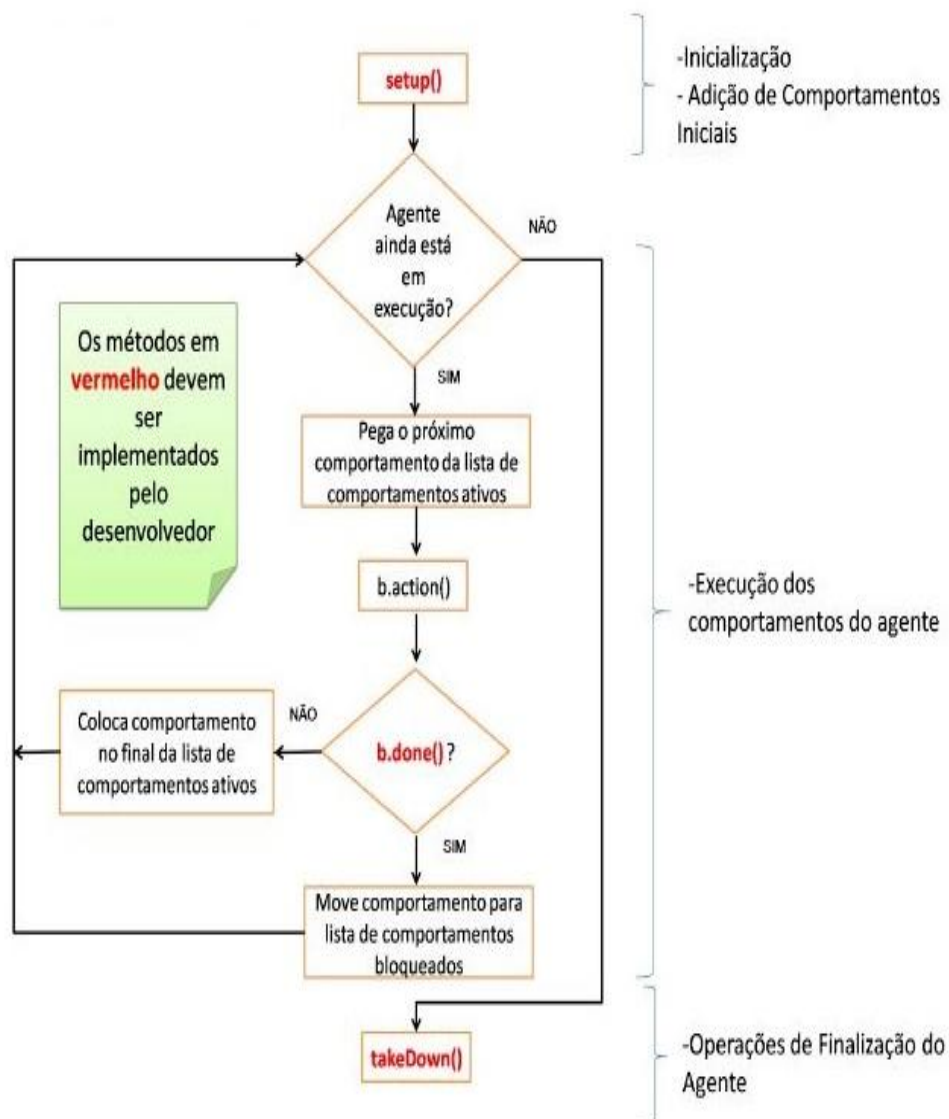
- a) Inicialização - Consiste na execução do método *setup()*;
- b) Realização da tarefa - Representa o nível de execução dos comportamentos do agente.

- c) O escalonador seleciona o primeiro comportamento da fila e executa seu método *action()*. Após a execução deste método, verifica-se a finalização do comportamento no método *done()*. Caso este comportamento ainda não esteja finalizado, o escalonador captura o próximo comportamento da lista de comportamentos ativos, colocando este comportamento ainda não finalizado no final da fila, bloqueando até que chegue sua vez de ser executado novamente;

Limpeza e finalização - Consiste na execução de métodos específicos para finalização do agente *takeDown()*.

A FIGURA 6, ilustra a interação de um agente com estas filas com o uso de uma análise no ciclo de execução de um agente.

FIGURA 6 – CICLO DE EXECUÇÃO DE UM AGENTE



FONTE: Adaptado de: Bellifemine, Caire e GreenWood (2007, p. 59).

2.2.6 Protocolos de Interação entre Agentes

O padrão FIPA especifica um conjunto de protocolos que podem ser utilizados para padronizar as conversas entre os agentes. Para cada conversa, sempre existem dois papéis: o papel do iniciador, o agente que inicia a conversa, e o papel do participante, o que responde ao iniciador. (*FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS*, 2015)

Segundo Bauer, Müller e Odell (2001), as conversas são feitas através de mensagens que devem satisfazer uma comunicação padronizada, que definem o tipo e o teor das mensagens, exemplos de mensagens padronizadas são FIPA (ACL) e KQML.

Para demonstrar a interação entre agentes de software são utilizados diagramas de protocolos de interação, que definem o exato comportamento de um grupo de agentes que se cooperam, combina-se diagramas de sequência com a notação de diagramas de estado para a especificação de protocolos de interação.

Exemplos desses protocolos são FIPA-Request, o FIPA-Propose, FIPA-Query, nestes protocolos de comunicação JADE define dois papéis, um que inicia a conversa e o outro que recebe à mesma o iniciador e o participante. (*FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS*, 2002)

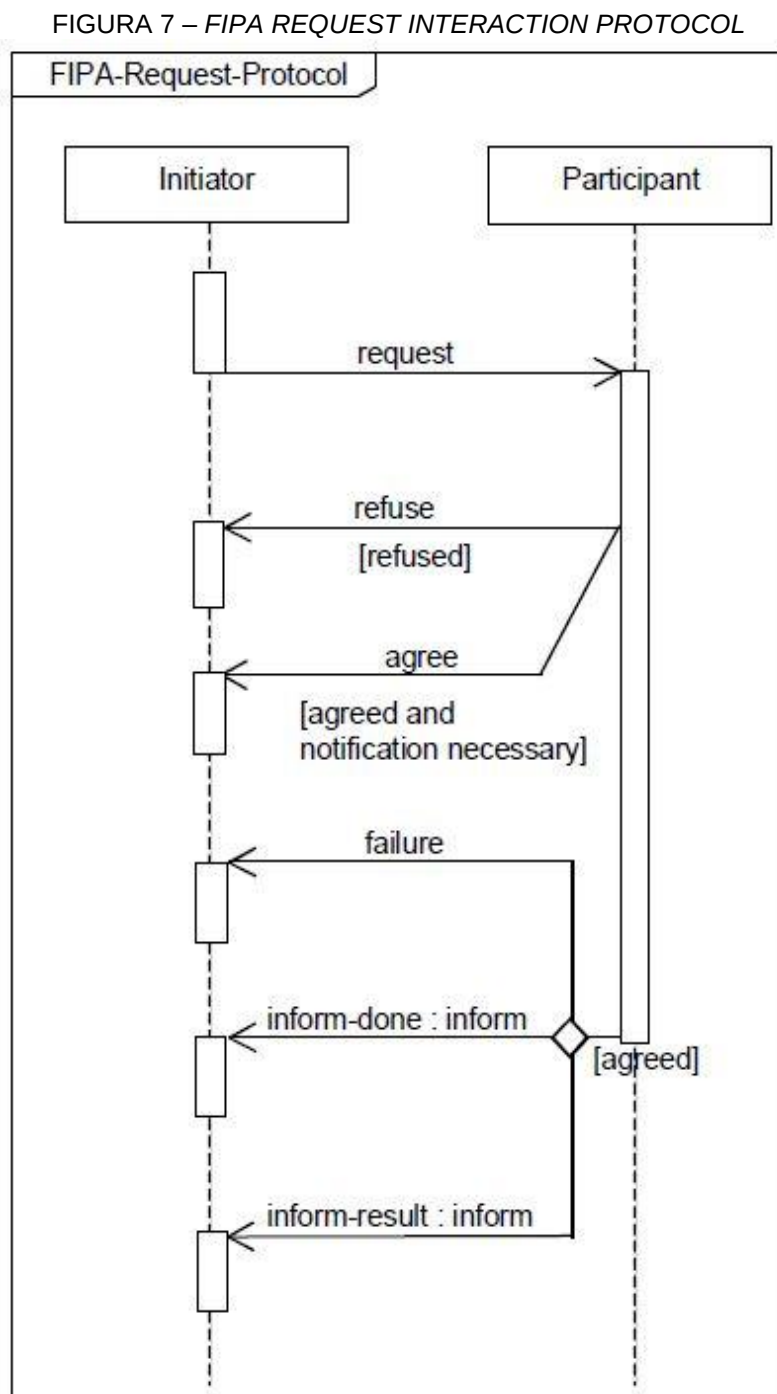
Como a FIPA apresenta um conjunto de onze protocolos de interação, a seguir serão descritos sucintamente somente os que foram utilizados no desenvolvimento do trabalho.

2.2.6.1 FIPA request interaction protocol

Este protocolo permite que um agente solicite a outro a execução de alguma ação, neste tipo de protocolo o agente solicitador sabe a qual agente solicitar. (*FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS*, 2002, p. 1)

Um agente Iniciador (*Initiator*) solicita (*request*) a execução de uma ação para o agente Participante (*Participant*), que processa o pedido e faz uma decisão de aceitar (*I agree*) ou recusar (*refuse*) o pedido. Se uma decisão de recusar é feita, então "Recusou" torna-se verdadeiro e o participante comunica a recusa caso contrário "concordou" torna-se verdade.

Quando a solicitação é processada, pode ocorrer duas respostas por parte do participante, uma resposta informando que concluiu com êxito o pedido (*inform-done*), ou uma resposta que terminou e informa ao iniciador os resultados (*inform-result*). Quando a solicitação falhar por algum motivo retorna (failure). A FIGURA 7 ilustra este protocolo.



FONTE: FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (2002, p.1).

2.2.6.2 FIPA contract net interaction protocol specification

Neste protocolo o agente iniciador precisa solicitar uma tarefa, mas não sabe para qual agente participante fazer o pedido, além disso, ele pode negociar para obter a melhor proposta. (*FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS*, 2002, p.1)

O agente iniciador envia uma mensagem a vários participantes, solicitando as propostas oferecidas por eles. Esta mensagem é chamada de *call for propose* (cfp) que inclui as pré-condições que o participante deseja, os participantes que recebem esta mensagem, precisam verificar se possuem ou não as pré-condições solicitadas. Pode acontecer de um participante que receber a mensagem cfp não responda, isto ocorre porque o agente recusou o serviço ou não teve tempo hábil para a resposta. O tempo de resposta é definido pelo agente iniciador quando isto ocorre houve um *deadline*, ou seja, o agente não conseguiu enviar uma resposta no prazo definido.

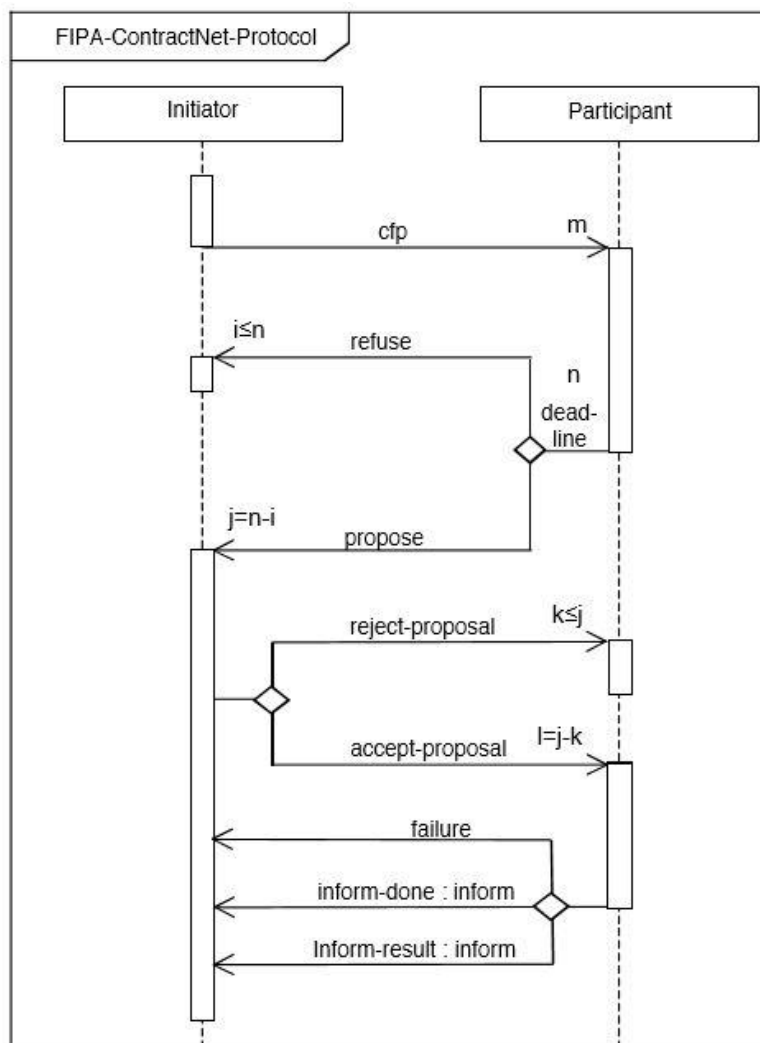
Uma quantidade (j) de agentes envia a proposta (*propose*) ao agente iniciador, este número é igual ao número total de agentes que responderam a proposta (n) subtraindo-se os agentes que recusaram a resposta, representados pela letra i , na FIGURA 8 é representado por $j=n-i$.

O agente iniciador após receber as propostas responde ao agente cuja proposta foi aceita uma mensagem (*accept-proposal*) e uma mensagem (*reject-proposal*) aos agentes que não tiveram a proposta aceita, o agente iniciador pode aceitar mais de uma proposta e para isto ele pode enviar mais uma mensagem (*accept-proposal*).

Toda negociação termina quando o participante que recebeu a mensagem (*accept-proposal*) envia uma mensagem de confirmação dizendo que a tarefa foi realizada como esperado (*inform-done*), ou uma mensagem com algum resultado (*inform-result*). Caso o agente que teve a proposta aceita não consiga realizar a tarefa, ele envia uma mensagem de falha (*failure*) com o motivo de não ter conseguido realizar a tarefa.

A FIGURA 8 ilustra o processo de negociação envolvido entre dois agentes utilizando o protocolo *Fipa contract net interaction protocol specification* e suas interações.

FIGURA 8 – FIPA CONTRACT NET INTERACTION PROTOCOL



FONTE: FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (2002, p.2).

2.3 SISTEMAS MULTIAGENTES

Russel e Norvig (1995) definem SMA como sendo uma subárea da Inteligência Artificial, junto com a resolução de problemas distribuídos.

Segundo Jennings (2000), um SMA pode ser definido como: um conjunto de agentes, que interagem uns com os outros agindo em favor de usuários com diferentes objetivos e motivações. Para terem sucesso nas interações, eles precisam ter habilidades de cooperação e negociação uns com os outros, exibindo um comportamento autônomo e ao mesmo tempo trabalhando em conjunto de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos.

SMA podem ser utilizados nos domínios que consistem as seguintes características (WOOLDRIDGE, 2002 p. 105):

- Conhecimento é distribuído em locais diferentes.
- Diversas entidades, unem suas habilidades de resolução de problemas para ser capaz de resolver um problema complexo, mantendo o seu comportamento autônomo.
- Os problemas no domínio podem ser decompostos em diferentes sub-problemas, mesmo tendo algum tipo de interdependências.

Para Luck, McBurney e Preist (2003, p. 18), SMA são entidades autônomas e inteligentes de software capazes de realizar ações autônomas em diferentes ambientes.

De acordo com Bellifemine, Caire e Greenwood (2007, p.3), SMA são sistemas que possuem vários agentes. São utilizados para modelar sistemas complexos possibilitando que agentes que têm objetivos comuns ou conflitantes possam interagir cooperando uns com os outros, sobre o ambiente ou por comunicação e negociação para benefício mútuo ou ainda competindo para benefício próprio.

Para Maalal e Addou (2011), a concepção de SMA é complexa, porque exige a inclusão de várias partes do sistema que podem ser abordadas a partir de diferentes ângulos.

2.3.1 Aplicações de Sistemas Multiagentes

Os SMAs estão sendo cada vez mais utilizados numa grande variedade de aplicações, desde pequenos sistemas de assistência pessoal, a complexos sistemas essenciais para aplicações industriais. (JENNINGS; WOOLDRIDGE, 1995)

As principais áreas de aplicação de SMAs segundo Bellifemine, Caire e Greenwood (2007, p. 9-10) são:

- Aplicações industriais: foi onde surgiram as primeiras técnicas e onde foram realizados experimentos que demonstraram o potencial dos SMA em ambientes industriais como: processos de controle, diagnóstico de sistemas, fabricação, logística de transportes e gerenciamento de redes.
- Internet: devido a natureza do grande volume de informações disponíveis, a Internet tem contribuído com o uso de tecnologias de

agentes no comércio e nos campos de gerenciamento de processos de negócios, eles oferecem oportunidades para melhorar de forma significativa a maneira pela qual as entidades envolvidas, clientes e empresas interagem nos processos de negociação.

- Trânsito e Transporte: a natureza distribuída do tráfego e processos de transporte e a forte independência entre as entidades envolvidas em tais processos fazem dos SMAs uma ferramenta valiosa para a realização de soluções comerciais verdadeiramente eficazes. Um exemplo é o sistema de controle de tráfego aéreo implantado e utilizado com sucesso no aeroporto de Sydney, na Austrália.
- Sistemas de telecomunicações: onde grandes redes de componentes distribuídos interconectados precisam ser monitorados e gerenciados em tempo real, no mercado competitivo as empresas de telecomunicações e provedores de serviços precisam fornecer melhores serviços, mais rápidos ou mais confiáveis. Portanto, SMAs são utilizados tanto para a gestão de tais redes distribuídas e para a realização de serviços avançados de telecomunicações, fornecendo autonomia para essas aplicações.
- Sistemas multi-robóticos: usam técnicas de planejamento distribuídos para os multi agentes e a coordenação entre os diferentes robôs.
- Cuidados de saúde: SMAs para tratar diversos tipos de problemas no domínio de cuidados de saúde, incluindo agendamento de pacientes, acesso a informações médicas e de gestão e de apoio à decisão. Várias aplicações realizadas têm mostrado que SMAs podem ser a solução certa para a construção sistemas de decisão médica realizados pelo uso de agentes inteligentes.
- Aplicações Comerciais: gerência de Informação (mecanismos de busca na *Web*), comércio eletrônico e gerência de processos de negócios.
- Computação: engenharia de *software*, interfaces adaptativas, bancos de dados dedutivos e ativos e sistemas distribuídos.
- Entretenimento: jogos de computador, cinema interativo e aplicações de realidade virtual.

- Matemática: demonstração de teoremas, resolução simbólica de equações e geometria.
- Reconhecimento de linguagem e resolução de problemas: dispositivos para reconhecimento de escrita a mão e reconhecimento de voz.
- Sistemas Tutores: modelagem do aluno, escolha de estratégias pedagógicas.

2.4 METODOLOGIAS DE ANÁLISE E PROJETOS BASEADOS EM AGENTES

A Engenharia de Software Orientada a Agentes (OA) trata do uso do paradigma de agentes inteligentes no desenvolvimento de software para solução de sistemas autamente complexos. (GIORGINI, 2005)

Uma metodologia de análise e projeto pretende primeiramente auxiliar a compreensão de um sistema em particular, e, em segundo, na concepção de que metodologias geralmente consistem de um conjunto de modelos que seguem um conjunto de orientações. (WOODRIDGE, 2002, p. 226)

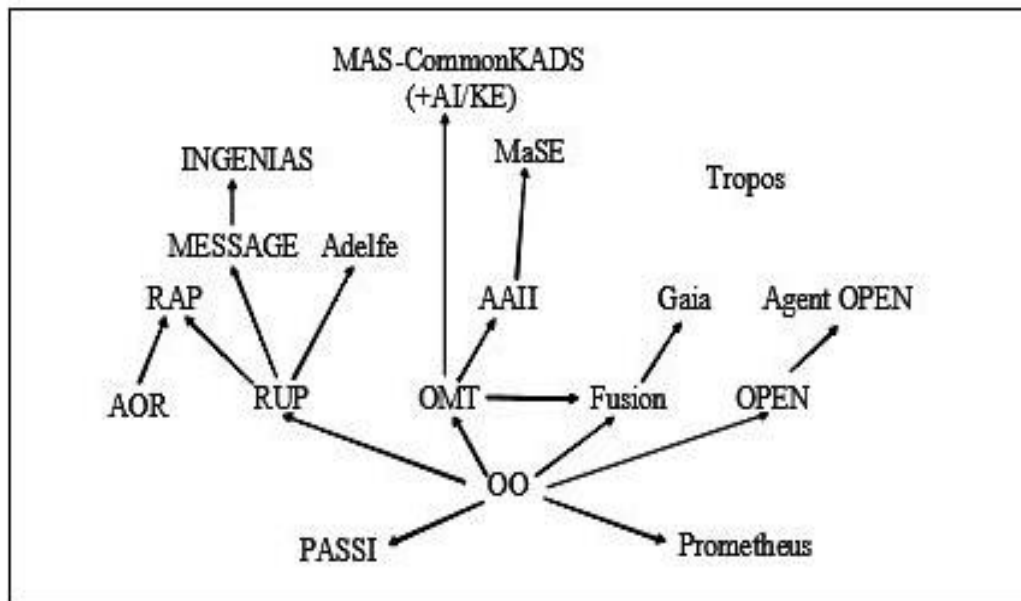
Os modelos servem para compreender um sistema que está sendo proposto. Normalmente, os modelos começam abstratos, e com o processo contínuo de análise e projeto, tornam-se cada vez mais concretos, detalhados, e mais perto da implementação.

Woodridge (2002, p. 226), destaca que as metodologias para a análise e projeto de sistemas baseados em agentes podem ser amplamente divididas em dois grupos:

- Aqueles que têm sua inspiração de desenvolvimento orientado a objeto, e que utilizam ou adaptam Metodologias Orientadas a Objetos (OO) existentes.
- Aqueles que adaptam a engenharia do conhecimento ou outras técnicas.

Para Giorgini (2005), existem as metodologias que utilizam uma mescla de conceitos das duas já citadas por Woodridge (2002) e ainda algumas que são derivadas de outras metodologias OA. A FIGURA 9 mostra as raízes das metodologias OA.

FIGURA 9 – RAIZES DA METODOLOGIA ORIENTADA A AGENTES



FONTE: Giorgini (2005, p. 7).

Algumas das muitas metodologias para análise e projeto de SMA são citadas a seguir:

- *Australian Artificial Intelligence Institute (AAIL)* foi desenvolvida com base na experiência acumulada durante a construção de sistemas BDI, definem as especificações de agentes, tais como desejos, crenças e intenções. (MAALAL; ADDOU, 2004)
- *GAIA* onde a entidade agente, é um elemento central, podendo desempenhar uma ou mais funções. O agente pode desempenhar papéis definidos em termos de permissões, responsabilidades, atividades e interações com outros papéis. (ISERN; SANCHEZ; MORENO, 2011)
- *MESSEGE* surgiu no domínio das aplicações de Telecomunicações. Utiliza a UML como ponto de partida, adicionando conceitos de entidades e relacionamentos necessários à AUML. Descrever a forma como um SMA funciona para a realização de uma meta coletiva.
- *MAS-CommonKADS*, tornou-se uma referência no desenvolvimento de SMAs baseados em conhecimento, principalmente na Europa. O Modelo de Experiência é o central da metodologia *CommonKADS* ele

tem a meta de modelar o conhecimento para resolução de problemas a ser adotado. (SCHREIBER, 1999)

- *ADELFE* foi desenvolvida com propósito de consolidar a teoria de SMAs Adaptativos, *Adaptive Multi-Agent Systems* (AMAS) e trabalhar alguns aspectos ainda não considerados pelas metodologias OA. Trabalha baseado num processo específico adaptado de uma interpretação do *Rational Unified Process* (RUP), além do RUP utiliza conceitos da UML e também os princípios para expressar protocolos de interação entre agentes da AUML. (GIORGINI, 2005)
- Tropos é uma metodologia de desenvolvimento de sistemas OA que utiliza os conceitos de nível de conhecimento, tais como, meta, plano e outros, nas fases de desenvolvimento do software. Noções de agente, objetivo de tarefas e de dependência (social) são usados para modelar e analisar requisitos de software precoces e tardios. (TROPOS PROJETO, 2015)
- MASE, *Multiagent System Engineering* (MASE) é uma metodologia de uso geral para o desenvolvimento de SMA que utiliza os princípios básicos de engenharia de software. MASE não vê os agentes com propósito de existências autônomas, criativas e reativas, os vê como processos de software puros. Nesta metodologia os agentes são vistos como uma abstração mais profunda do paradigma de OO, onde os agentes são especializações de objetos possuindo métodos que podem ser invocados por outros objetos, para alcançar uma meta individual ou do sistema, os agentes conversam entre si e agem proativamente. (SOLEIMANIAN; ZEBARDAST; AMINI, 2013)
- Metodologia para a Análise e Projeto de SMA utilizando Jade concentra-se sobre as principais questões em análise e projeto de SMA. A etapa de análise é genérica, enquanto a fase de concepção concentra-se nas construções fornecidas pelo Framework de Desenvolvimento de Agentes Java (JADE), utilizando o padrão FIPA. (NIKRAZ; CAIRE; BAHRI, 2006)

A Engenharia de software baseada no agente é um campo relativamente novo e pode ser pensado como um evolução da programação orientada a objetos. (NIKRAZ, CAIRE; BAHRI, 2006).

Em 2003 Luck, Mcburney e Preist, apontavam que uma das principais barreiras para a utilização em larga escala da tecnologia de agentes era a falta de metodologias de desenvolvimento de agentes. Next contexto segundo Werneck et. al. (2008), o crescimento da demanda de aplicações que trabalhavam de forma autônoma e integrada e para a construção desses sistemas a modelagem OA em conjunto com a modelagem OO, supria a necessidade na construção desses sistemas complexos. Já em 2011, Werneck, Costa e Cysneiros afirmam que, apesar da crescente utilização de SMA os paradigmas da época não consideravam requisitos para lidar com propriedades do agente como autonomia, sociabilidade e pro atividade, e que, eram muitas as metodologias que aplicavam os conceitos orientados a agente o que dificultava a sua avaliação.

Para Maalal e Addol (2011), mesmo com muitas metodologias para abordagens orientadas a agente, nossa capacidade de modelar, projetar e construir sistemas distribuídos complexos aumentou.

2.4.1 Metodologia para a Análise e Projeto de SMA Utilizando Jade

Para Nikraz, Caire e Bahri (2006 p. 4), a metodologia para a análise e projeto de SMA utilizando JADE, procura formalizar as fases de análise e projeto de software baseados em agentes e o ciclo de vida do desenvolvimento. O foco da metodologia é sobre o processo e os artefatos.

É nas fases iniciais do ciclo de desenvolvimento de software, isto é na fase de projeto, onde é decidido qual ferramenta será utilizada. A metodologia para análise e projeto utilizando Jade aproveita as funcionalidades que o Framework Jade oferece para servir como base na fase de concepção. No final da fase de concepção, já é possível progredir diretamente para o aplicação, que é onde ocorre a codificação real.

Tal como acontece com qualquer metodologia, alguns pressupostos devem ser seguidos e merecem destaque:

- A plataforma JADE é a plataforma de escolha para a implementação.
- Há um número relativamente pequeno de agentes.

2.4.1.1 Fase de análise

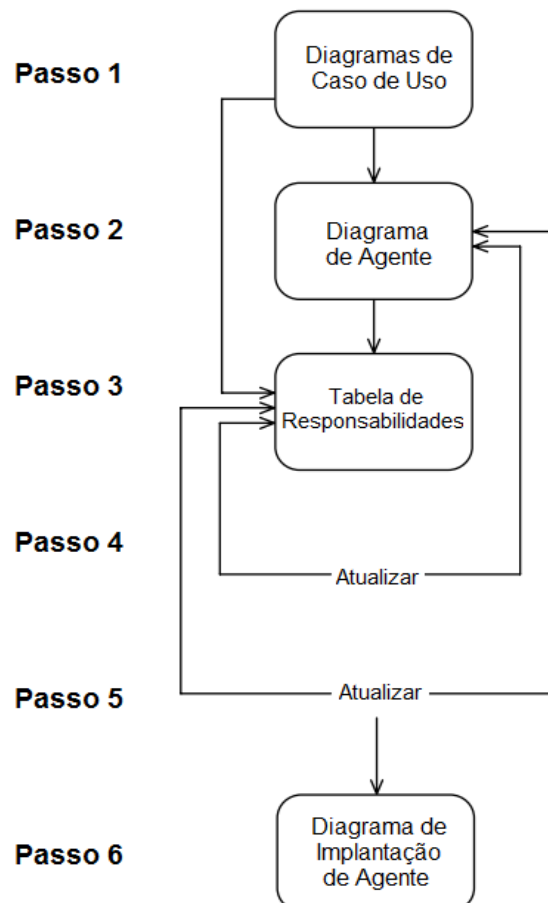
Segundo Nikraz, Caire e Bahri (2006 p. 7), a fase de análise visa identificar o problema sem qualquer preocupação sobre a solução. A fase de análise é realizada seguindo os seguintes passos:

- a) Casos de uso;
- b) Identificação inicial de tipos de agentes;
- c) Identificação das responsabilidades dos agentes;
- d) Identificação da interação entre agentes;
- e) Refinamento do agente;
- f) Gestão e monitorização.
- g) Informações de implantação do agente;

Os elementos importantes obtidos com a realização da fase de análise são os diagramas. Eles formam a base para a fase de concepção.

A FIGURA 10 ilustra os passos realizados em cada etapa da fase de análise.

FIGURA 10 – RESUMO DA FASE DE ANÁLISE



FONTE: Adaptado de: Nikraz, Caire e Bahri (2006, p.17).

2.5 MODELAGEM AUML

Segundo Bauer, Müller e Odell (2001), a AUML ou Agente UML é uma extensão da linguagem UML, um padrão para análise e design OO com extensão de métodos conhecidos e confiáveis, fornecendo ferramentas de engenharia para apoiar métodos comprovados de implantação de tecnologia. É fundamental detalhar as premissas dos conceitos UML aplicados aos agentes:

- a) extensão de objetos ativos: introduz agentes como um objeto que pode ter autonomia como capacidade de iniciar a ação, sem interferência externa e capacidade de recusar ou alterar um pedido externo.
- b) uso de representações padrão: métodos e ferramentas para apoiar a análise, especificação e concepção de software agente.

Ao adotar uma abordagem em camadas, AUML oferece os diagramas de protocolo de interação de agentes para representar a interação dos agentes. Os diagramas de protocolo de interação são definidos pelo FIPA.

A principal vantagem de modelagem com AUML é a sua representação gráfica intuitiva da arquitetura e processos.

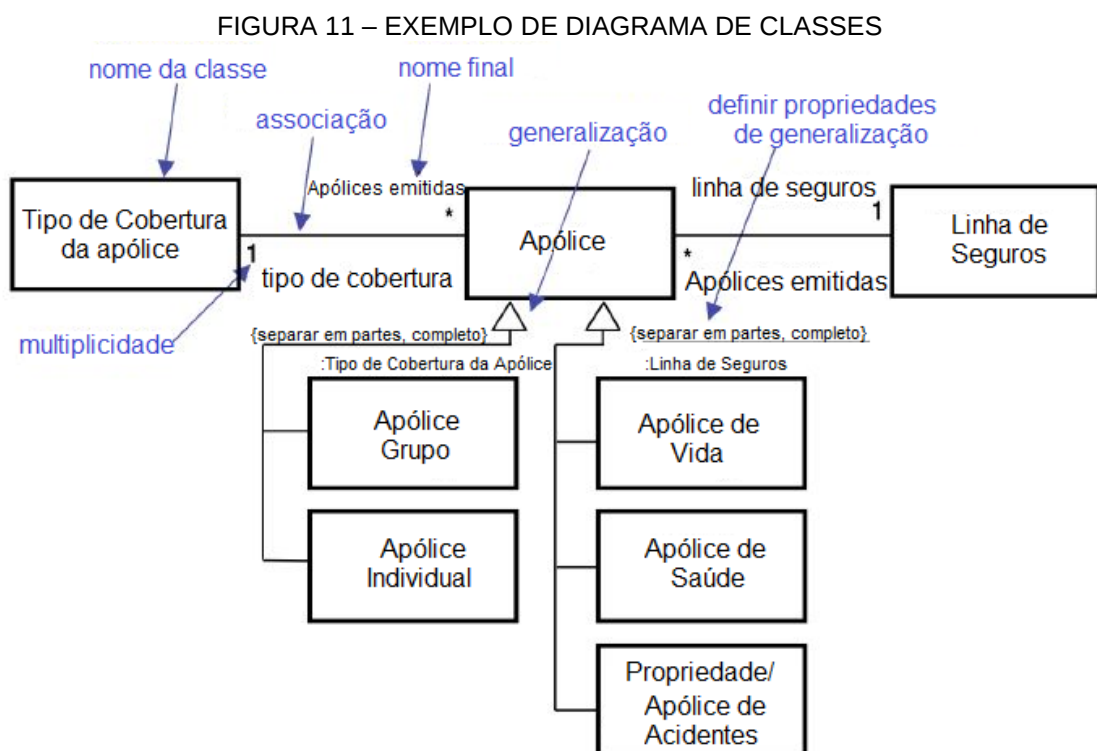
De acordo com Sellers (2013), tem sido proposto um objetivo de padronização da comunidade de linguagens de modelagem, apoiando a metodologia Tropos, com os diagramas divididos em dois grupos: diagramas estáticos e dinâmicos, recomendado um conjunto formado por três tipos de diagramas estáticos e dois dinâmicos, são eles:

- a) Extensão do Diagrama de Classe da UML.
- b) Diagrama de Organização.
- c) Diagrama de Papéis.
- d) Extensão do Diagrama de Sequência da UML.
- e) Extensão do Diagrama de Atividades da UML.

Outras metodologias usam também os diagramas de casos de uso da UML como por exemplo, MASE, ADELFE e MAS-CommonKADS onde ele é chamado de descritor de domínio diagrama.

2.5.1 Extensão do Diagrama de Classe UML para Agentes

Diagramas de classe da UML foram adaptados para descrever relacionamentos entre os papéis, relacionamentos entre organizações e papéis e relacionamentos entre agentes. Diagramas de classe descrevem classes e interfaces com os seus atributos e operações, bem como associações entre eles incluindo agregação, composição, generalização e dependências (Bauer, Odell 2005 p 5). A FIGURA 11 ilustra uma extensão do diagrama de classe para Agentes.



FONTE: Adaptado de: Bauer e Odell (2005, p. 5).

2.5.2 Diagrama de Caso de Uso AUML

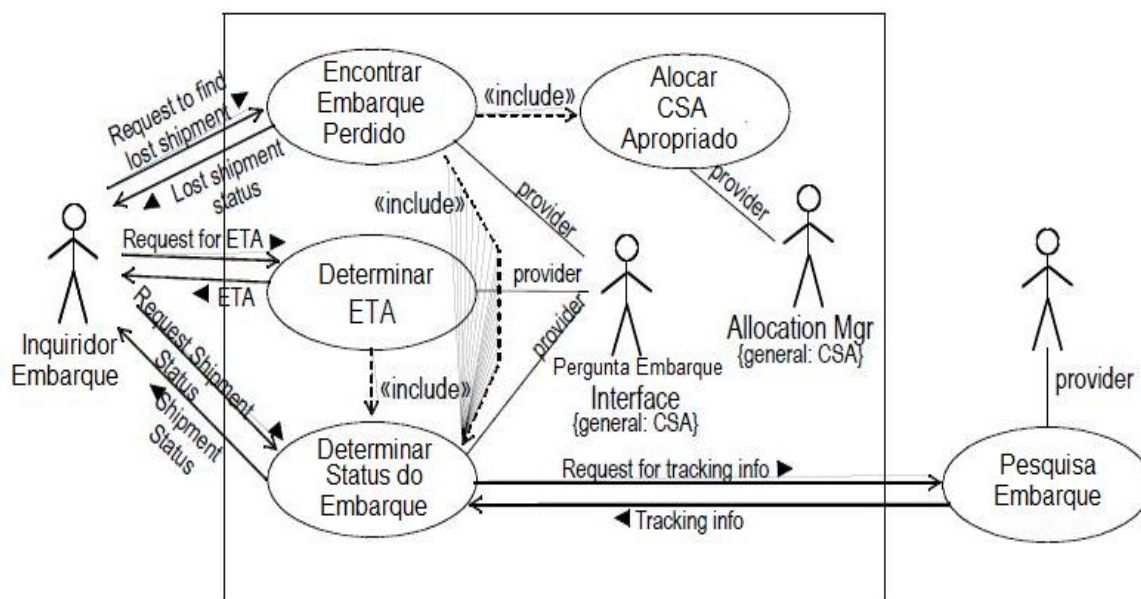
Segundo Bauer e Odell (2005), em comparação com os casos de uso da UML alguns conceitos novos foram adicionados:

- novos eventos conhecidos como ações indicando funções e serviços, afetam um ator de alguma forma e são indicados com nomes e direções nas associações entre atores e casos de uso.
- adição gráfica indicando o ator fornecendo serviço definido pelo caso de uso, chamado de *provider*.

- c) o ator pode interagir com o sistema, ou dentro do sistema e, portanto, pode ser interno ou externo ao sistema.

A FIGURA 12 ilustra os conceitos de Bauer e Odell (2005), sobre diagrama de caso de uso.

FIGURA 12 – CASO DE USO FUNÇÕES E RECURSOS DO ATOR



FONTE: Adaptado de: Bauer e Odell (2005).

Segundo Dantas et. al. (2007) a definição de casos de uso é essencial para definir os requisitos iniciais do sistema, descreve os agentes que participam da solução dos problemas e principalmente os diagramas de casos de uso interno. A FIGURA 13 identifica as habilidades e restrições de cada agente.

FIGURA 13 – MODELO DE CASO DE USO INTERNO

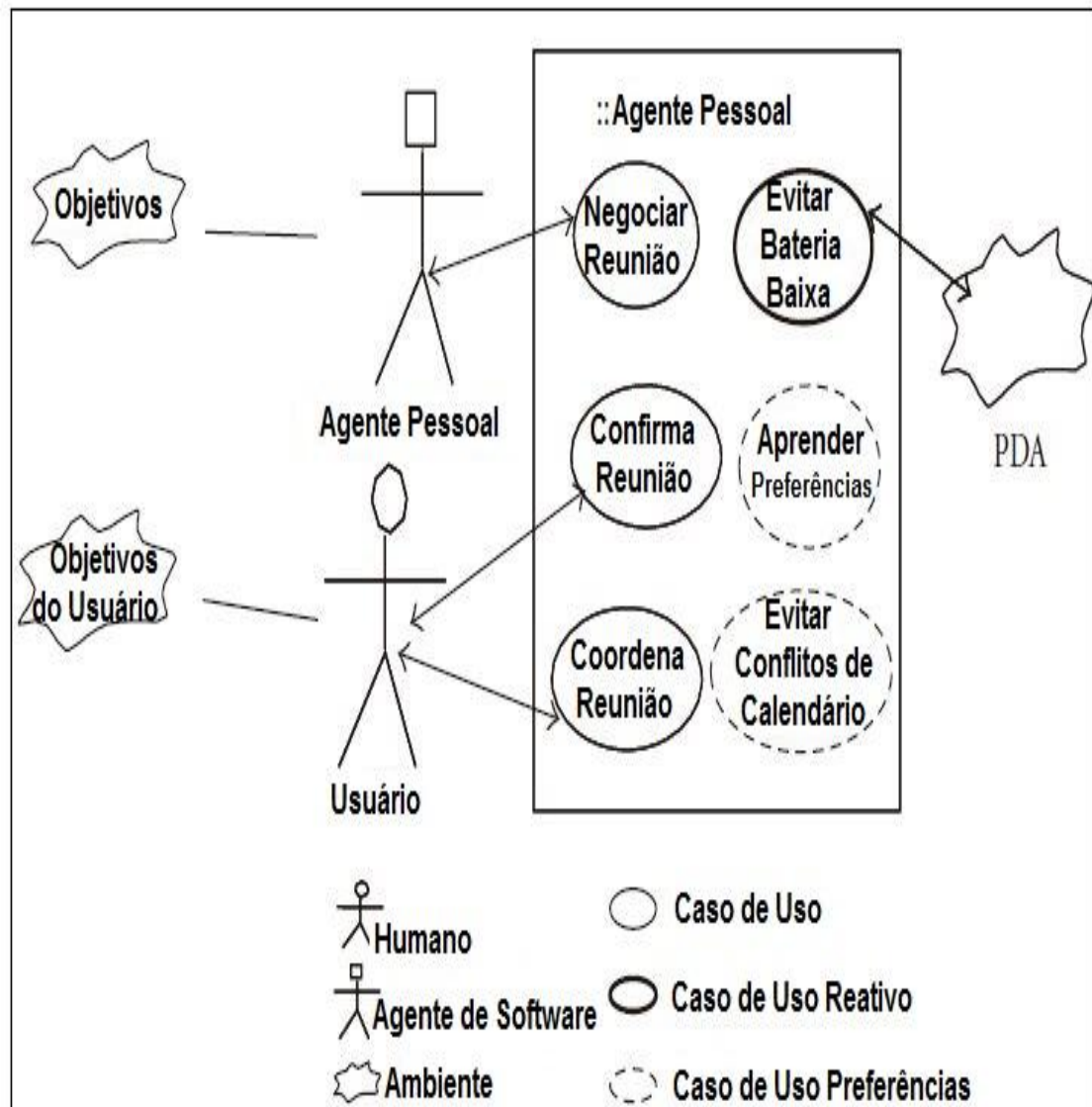


FONTE: Sellers (2013).

Seller (2013) apresenta um modelo de casos de uso com atores externos diferenciados, o ator agente é representado como uma pessoa com cabeça em formato de um quadrado, os atores humanos seguem a mesma linha de notação da UML tradicional. Já os atores ambientes apresentam a notação gráfica ilustrada na FIGURA 14.

Surgiram também dois novos conceitos, o conceito de preferências e o conceito reativo. O modelo de preferências é responsável por uma das características do agente a aprendizagem, já o modelo reativo responde a percepções. Interpreta a entrada, verifica a regra correspondente e atua buscando cumprir a tarefa proposta.

FIGURA 14 – RECOMENDAÇÃO DE DIAGRAMA DE CASO DE USO



FONTE: Adaptado de: Sellers (2013).

2.5.3 Detalhamento de Caso de Uso AUML

De acordo com Dantas et. al. (2007) o detalhamento de caso de uso serve para destacar as atividades desempenhadas pelos agentes. A FIGURA 15 apresenta um exemplo de detalhamento de um Caso de Uso, destacando as atividades desempenhadas pelo agente cadastro e seu modelo de tarefas, destacando o ator principal, e o agente, apresenta as ações realizadas pelo ator principal e a resposta do agente. O plano de ação representa o modelo de tarefas do agente cadastro.

FIGURA 15 – DESCRIÇÃO DE CASO DE USO AUML

Fazer Cadastro de um Novo Aluno	
Ator Principal	Aluno
Agente	Agente Cadastro
Resumo	Este caso de uso define o cadastramento do aluno no sistema.
Ações	01. Novo aluno fornece seus dados: nome, endereço, bairro, cidade, estado, CEP, telefone, <i>e-mail</i> , <i>login</i> e senha. 02. Aluno solicita cadastro após fornecer os dados. 03. Agente valida dados. 04. Agente efetua cadastro. 05. Aluno recebe confirmação do cadastramento.
Fluxo Alternativo	O aluno poderá sair do sistema a qualquer momento.
Modelo de Tarefas - Agente Cadastro	
Plano de Ação	02a. Recebe os dados do aluno. 02b. Verifica os campos vazios, além dos campos CEP, telefone, <i>login</i> e senha. 03a. Enquanto houver dados inválidos, solicita novos dados. 04a. Gera número de registro para o aluno. 04b. Trata os dados digitados para inserir no BD e armazena os mesmos. 05a. Envia mensagem de confirmação para o aluno, apresentando os dados armazenados. 05b. Direciona o aluno para página inicial do aluno, repassando seu nome e registro.
Interação	AlunoPagPrincipal.jsp

FONTE: Dantas et. al., (2007).

2.5.4 Tabela de Responsabilidades de Agentes

De acordo com Nikraz, Caire e Bahri (2006, p.11), para cada tipo de agente identificado, uma lista inicial é feita de suas principais responsabilidades. O artefato de modelagem resultante deste processo é a tabela de responsabilidade ilustrada na FIGURA 16.

Para criar a tabela de responsabilidades aplica-se as seguintes regras:

- Extrair o conjunto inicial de responsabilidades dos casos de uso.
- Considerar que as responsabilidades dos agentes mais importantes devem ser desenvolvidas primeiro, podendo atrasar a identificação de responsabilidades de agentes de etapas posteriores. Para que esta funcionalidade seja desenvolvida é necessário definir as prioridades de cada agentes em relação ao sistema.

FIGURA 16 – TABELA DE RESPONSABILIDADES

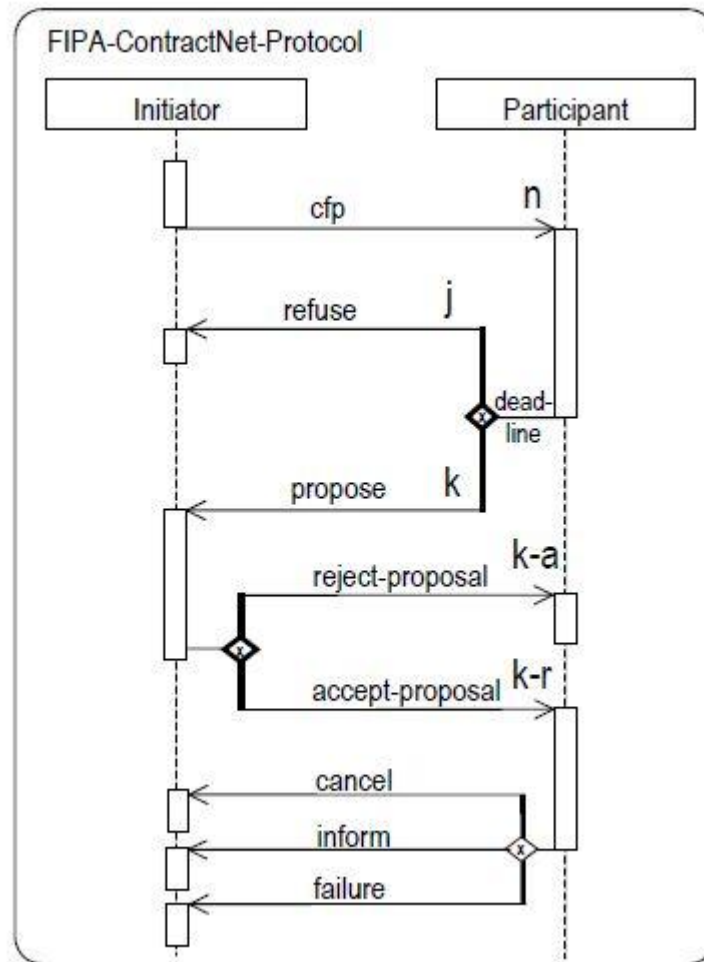
Tipo de Agente	Responsabilidades
Agente Organizador Cinema	Servir pedidos de início de convites do usuário organizador cinema. Deixe o usuário cinema organizador selecionar amigos para convidar. Deixe o usuário inserir as preferências sobre filmes e cinemas. Apresenta o Calendário do Cinema. Responder a convites de outros agentes organizador cinema.

FONTE: Adaptado de: Nikraz, Caire e Bahri (2006, p.11).

2.5.5 Extensão do Diagrama de Sequência para AUML

Segundo Odell (2010) o primeiro diagrama adaptado da UML para agentes foi o diagrama de sequência para descrever os protocolos definidos no contexto de papéis e interação entre agentes, apresentando um modelo com um agente iniciador e um participante, a interação sempre começa com o agente iniciador. A FIGURA 17 ilustra um exemplo adaptado do diagrama de sequência da UML para agentes.

FIGURA 17 – DIAGRAMA DE SEQUÊNCIA AUML COM EXTENSÕES



FONTE: Bauer e Odell (2005, p.16).

3 ESTADO DA ARTE E TRABALHOS CORRELATOS

Esta revisão do estado da arte em um formato cronológico (FIGURA 18) busca apresentar os estudos e abordagens sobre SMA fornecendo um contexto para melhor compreender o estado da arte atual.

FIGURA 18 – CRONOLOGIA ESTADO DA ARTE



FONTE: O autor (2016).

Endo (2010) apresenta um panorama de tendência do uso da tecnologia adaptativa para desenvolvimento de SMA, com capacidade de racionalização utilizando tabelas de decisão dando foco principalmente a resolução de problemas de tomada de decisão. Para consolidação dos conceitos Endo apresentou um estudo de caso com o objetivo de priorização de atendimentos a pacientes com

problemas nefrológicos, desenvolvendo agentes racionais para cálculo de pontuação para classificação das prioridades em filas de atendimento a pacientes.

De acordo com Alkhateeb et. al. (2010), uma rede de sensores sem fio é uma rede que consiste em dispositivos autônomos distribuídos utilizando sensores para monitorar cooperativamente condições físicas ou ambientais, tais como temperatura, som, vibração, pressão, ou movimento em diferentes locais, utilizando estes conceitos propõem o desenvolvimento de um sistema de segurança eficaz e de baixo custo feita pela rede de sensores sem fio e com base em multiagentes para proteger os edifícios e faculdades de um campus da Universidade.

Lermen (2011) utiliza uma arquitetura baseada em ontologias e agentes de software para o acoplamento e interoperabilidade de dados como base para construir um sistema de avaliação de competências acoplável a sistemas de gestão da aprendizagem agregando funcionalidades de avaliação de competências a ferramentas de gestão da aprendizagem já utilizadas atualmente. A utilização crescente da noção de competência nas organizações tem reafirmado o interesse sobre esse conceito, tanto sob uma perspectiva mais estratégica, onde predomina competências organizacionais e competências essenciais, quanto sob uma perspectiva mais específica de gestão de recursos humanos.

Khan e Saqib (2011) destacam que a falta dos alunos é uma grande preocupação para professores de instituições de ensino superior, o uso da identificação Biométrica seria a mais segura ferramenta para controle de faltas, ela não pode ser emprestada, roubada ou esquecida. Diante disso propõem um SMA com três agentes que possuem um mecanismo capaz de adquirir conhecimentos de seu ambiente e aprender a partir de sua experiência trabalhando em conjunto para automaticamente receber entradas, manipular os dados, e manter o registro de presença dos alunos de uma forma automática, utilizando sensores desenvolvidos através da adaptação da tecnologia biométrica, podendo também ser usado para registro de frequência dos funcionários e para fins de segurança.

Benavides (2012) destaca que nos últimos anos é crescente o interesse em sistemas de composição e de performance musical que utilizam a tecnologia de agentes, sendo maior o interesse por sistemas que integram técnicas de composição algorítmica, componentes de inteligência artificial e interatividade. Propõem um ambiente que utilize apenas componentes já existentes tanto para movimentação dos agentes como para propagação sonora. O raciocínio do agente

permite ao músico controlar ações dentro do mundo virtual tratando as movimentações como sequências programáveis de trajetórias.

Ren et. al. (2012) propõem uma estratégia de coordenação descentralizada para superar limitações de SMA com coordenação centralizada utilizados para restaurar sistemas energéticos quando ocorre uma interrupção ou falha. Os sistemas com coordenação centralizada, utilizam um único coordenador central responsável por todo o sistema de gestão, manutenção e restauração. No entanto, desvantagens, como deficiências na robustez, abertura e flexibilidade impedem o uso em aplicações online. Neste contexto o SMA proposto possui uma estrutura de coordenação flexível, de modo a equilibrar a eficácia e eficiência do SMA e gerenciar dinamicamente agentes no sistema.

Segundo Vieira (2013), a licitação pública é um processo administrativo com o objetivo de comprar bens ou serviços para órgãos pertencentes a administração pública, nesse processo há necessidade de tomadas de decisões como a escolha do tipo e da modalidade da licitação. Diante dessa necessidade propõem o desenvolvimento de SMA para o processo de licitação pública, com o objetivo de otimizar a tomada de decisão utilizando a arquitetura de troca de mensagens assíncronas, feitas pelo protocolo de conversação que dita as regras e impõem o formalismo para que as mensagens sejam encaminhadas e compreendidas pelos agentes.

A arquitetura empresarial tem a finalidade fornecer uma imagem clara e abrangente da estrutura de uma organização. Neste contexto, Berrada e Bounabat (2013) propõem uma nova abordagem para a arquitetura empresarial a modelagem de negócios que usa dois conceitos: a) *Multi-Agent Reactive Decisional System* (MARDS), modelo para especificar e implementar a arquitetura os componentes e suas relações; b) *Business Process Management* (BPM), conceitos de processo para a modelagem e simulação de processos da arquitetura empresarial. O SMA reativo decisional é uma estrutura de software caracterizado por um conjunto de Agentes Reativos Decisional, interligados por uma comunicação de interfaces,

Akhtar, Ghori e Salamat (2014) apresentam uma abordagem baseada na análise, projeto e verificação formal de multiagentes com base no *Information Management System* (IMS), a arquitetura possui um agente orquestrador que controla a coordenação entre todos os agentes. Ele também gerencia a conectividade de banco de dados para todo o sistema. O IMS é proposto com base

na arquitetura de agentes BDI, que modelam o sistema baseado na crença, desejo e intenções. É um sistema baseado em agentes que trabalha em computadores e em dispositivos móveis; fornece aos alunos informações sobre os seus horários de aula, datas do exame, resultados, as datas de pagamento das taxas, horários de ônibus universitários e rotas.

Rodrigues (2014) destaca que instituições de ensino precisam coletar e armazenar informações e acompanhar a evolução profissional de seus egressos. Rodrigues propõem um SMA utilizando técnicas de coleta de informações e processamento de linguagem natural com frames semânticos e ontologias, para coletar informações em bases de textos em linguagem natural, efetuar a extração de informação por meio de técnicas de processamento, além de identificar entidades e papéis exercidos, integrando informações heterogêneas.

Para Frazão (2015), o crescimento exponencial da informação disponível na internet torna cada vez mais difícil ao usuário obter as informações quando dela necessitam, justificando as abordagens baseadas na tecnologia dos agentes inteligentes autônomos e SMA, permitindo combinar múltiplos algoritmos de recomendação, aumentando as chances das recomendações serem do interesse do usuário, os algoritmos de recomendação são baseados em memória e incrementais.

As tecnologias móveis surgiram para facilitar o processo de aprendizagem, ampliando as atividades tradicionais de sala de aula. Neste contexto, Iglesia et. al. (2015) destacam a importância dos aspectos da arquitetura de software com o objetivo de enfrentar os desafios relacionados a compartilhamento de recursos em atividades de aprendizagem móveis colaborativas, oferecendo evidências qualitativas dos benefícios da auto-adaptação para aplicações de aprendizagem colaborativa via celular.

Kalton et. al. (2016) propõem a criação de uma simulação baseada em agentes inteligentes modelando a introdução de capacidades de coordenação de cuidados em um complexo sistema para pacientes com doença mental grave e persistente. Os resultados preliminares de aplicação da simulação são fornecidos para demonstrar as potenciais melhorias decorrentes da introdução da tecnologia a coordenação dos cuidados.

Segundo Fiorini (2000), “Sistemas Holônicos são aqueles que tem por objetivo alcançar características inerentes a organismos vivos e organizações sociais”. Neste contexto, Hsieh e Lin (2016) enfatizam que, a capacidade de se

adaptar dinamicamente às mudanças é um fator chave de sucesso para *Holonic multi-agent systems* (HMS), eles fornecem uma arquitetura flexível e reconfigurável para aceitar as mudanças com base na organização dinâmica e colaboração de agentes autônomos. Uma metodologia para projetar sistemas de software auto-adaptativos baseados na arquitetura HMS é proposta pelos autores.

A diferença e relevância deste trabalho proposto para os demais apresentados no estado da arte consiste em utilizar a metodologia para a análise e projeto de SMA utilizando Jade, a metodologia proposta pela FIPA.

3.1 TRABALHOS CORRELATOS

Antes de analisar os trabalhos correlatos, é importante ressaltar que o foco de desenvolvimento do trabalho foi realizado levando em consideração as características principais do SMA de gerenciamento acadêmico proposto que são: a) definição de perfil e competência; b) filtragem de perfil; c) acompanhamento pedagógico; d) autonomia; e) comunicação; f) negociação entre agentes.

O trabalho de Mahdi e Attia (2008) apresenta três agentes com destaque para dois principais: o agente estudante responsável por gerenciar o perfil do estudante e acompanhar suas ações, fazendo recomendações a ele quando necessário e o agente instrutor que fornece materiais didáticos e avalia a participação do aluno durante o curso.

Cazella et al. (2009) utiliza o conceito de competências e filtragem colaborativa realizado por agentes, apresentando um modelo de sistema de recomendação de objetos de aprendizagem.

Uma proposta de um ambiente adaptativo que utiliza modelos pedagógicos baseados nas competências do aluno proposto por Faddouli et. al. (2011), utiliza as notas das avaliações dos alunos para realizar o acompanhamento, recomendando estratégias com objetivo de aumentar o nível de aprendizado do estudante.

O trabalho de Bremgartner e Netto (2012) descreve uma experiência usando agentes inteligentes que realizam a recomendação personalizada de alunos que estão aptos a ajudar em função de suas habilidades e competências. O projeto auxilia as atividades dos alunos pela indicação personalizada de algum estudante do mesmo curso, que possui habilidades e competências desejáveis para tratar de dúvidas sobre determinado assunto. O trabalho mostrou que o uso de alunos com

perfil adequado para tirar dúvidas de seus colegas aumentou a motivação e promoveu a aprendizagem colaborativa entre os alunos, sendo uma importante estratégia pedagógica.

Uma ferramenta para a recomendação pedagógica foi criada por Paiva, Bittencourt e Silva em (2013), para auxiliar o professor no processo de tomada de decisões pedagógicas, por meio de técnicas de mineração de dados. Ao final, os autores constataram que a ferramenta é aplicável e traz benefícios ao processo de ensino e aprendizagem, oferecendo recomendações pedagógicas para as situações identificadas.

Alencar e Netto (2013) desenvolveram um sistema acadêmico integrado a base de dados do Ambiente Virtual de Aprendizagem (AVA) *Moodle*, para facilitar o acompanhamento das atividades e notas dos alunos. Os autores destacam que com a integração dos sistemas, foi possível gerar vários relatórios que auxiliaram na tomada de decisões dos profissionais de Educação a Distância (EaD).

Os trabalhos correlatos foram fundamentais para compreender o problema norteador do trabalho defendido. A relevância deste trabalho para os demais apresentados consiste em identificar e apresentar alunos com perfis de conhecimento específicos para determinadas solicitações de empresas, oportunizando ao aluno a inserção no mercado de trabalho por intermédio de um Sistema Multiagente, possibilitando a geração de resultados em tempo real. Além disso, há também a proposta de acompanhamento sobre quais alunos estão com problemas de notas e faltas. Entende-se que este processo seria um facilitador do trabalho desempenhado pelas coordenações de um curso.

4 ANÁLISE DO SISTEMA MULTIAGENTE

Neste capítulo serão apresentados os métodos utilizados no desenvolvimento do sistema. Primeiro, a arquitetura do sistema explicando em detalhes sobre a integração entre os agentes, o banco de dados e a aplicação desenvolvida em java. Em seguida, a tabela de responsabilidades de agentes. Depois são ilustrados os diagramas de caso de uso AUML e seus detalhamentos, os diagramas de sequência AUML, o diagrama de classe AUML, e o diagrama hierárquico do SMA proposto.

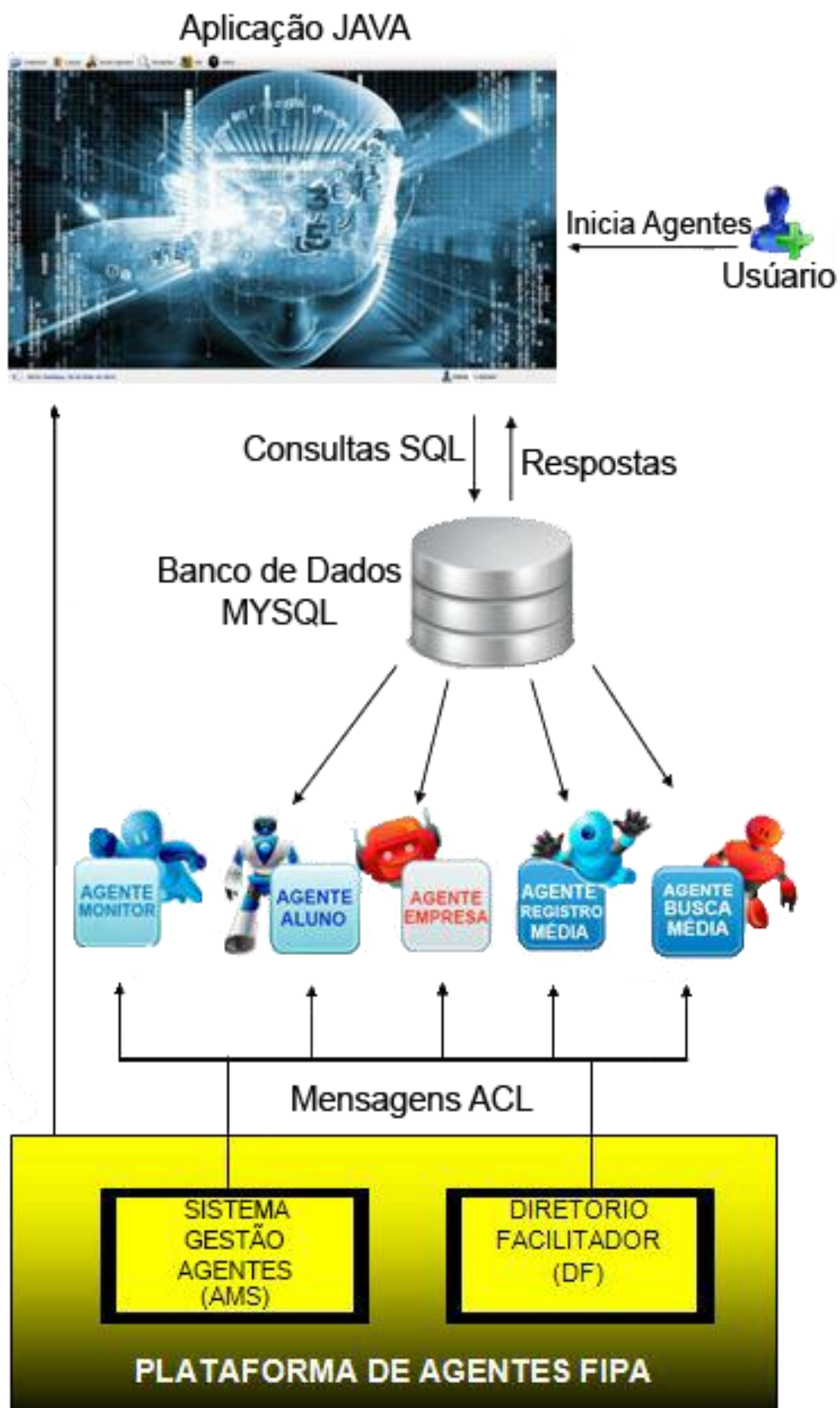
4.1 FASE DE ANÁLISE DO SMA

Para a fase de análise do sistema foram utilizados os conceitos da metodologia para a análise e projeto de SMA utilizando Jade. Esta metodologia é uma proposta disponível no site oficial do Framework Jade e procura formalizar as fases de análise e projeto de software baseados em agentes e o ciclo de vida do desenvolvimento focalizando o processo e os artefatos é a metodologia proposta pela FIPA.

4.1.1 Arquitetura do SMA

A arquitetura do SMA é formada pelos seguintes elementos: a Aplicação Interface do usuário desenvolvida utilizando a linguagem programação Java, utilizada para visualização dos resultados das interações entre os agentes desenvolvidos, o Banco de Dados MySQL que alimenta os argumentos para ativar o agente com seus dados, os cinco agentes desenvolvidos (Agente Monitor, o Agente Aluno, o Agente Empresa, o Agente Registro Média e o Agente Busca Média) e a plataforma de agentes Jade e seus componentes (MAS e DF) que permitem a comunicação entre os agentes realizada através de trocas de mensagens do tipo ACL, o banco de dados e o sistema. A FIGURA 19 ilustra a arquitetura do SMA e seus componentes adaptado da *FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS* (2012).

FIGURA 19 – ARQUITETURA DO SISTEMA MULTIAGENTE



FONTE: O Autor (2016).

4.1.2 Responsabilidades de Agentes.

Na etapa de análise e projeto do SMA é muito importante que as responsabilidades e comportamentos de cada agente sejam bem definidos, isto é fundamental para que a colaboração entre eles ocorra corretamente e suas tarefas sejam concluídas com êxito. No QUADRO 1 são destacados os tipos de agentes e suas responsabilidades.

QUADRO 1 – RESPONSABILIDADES DO AGENTE

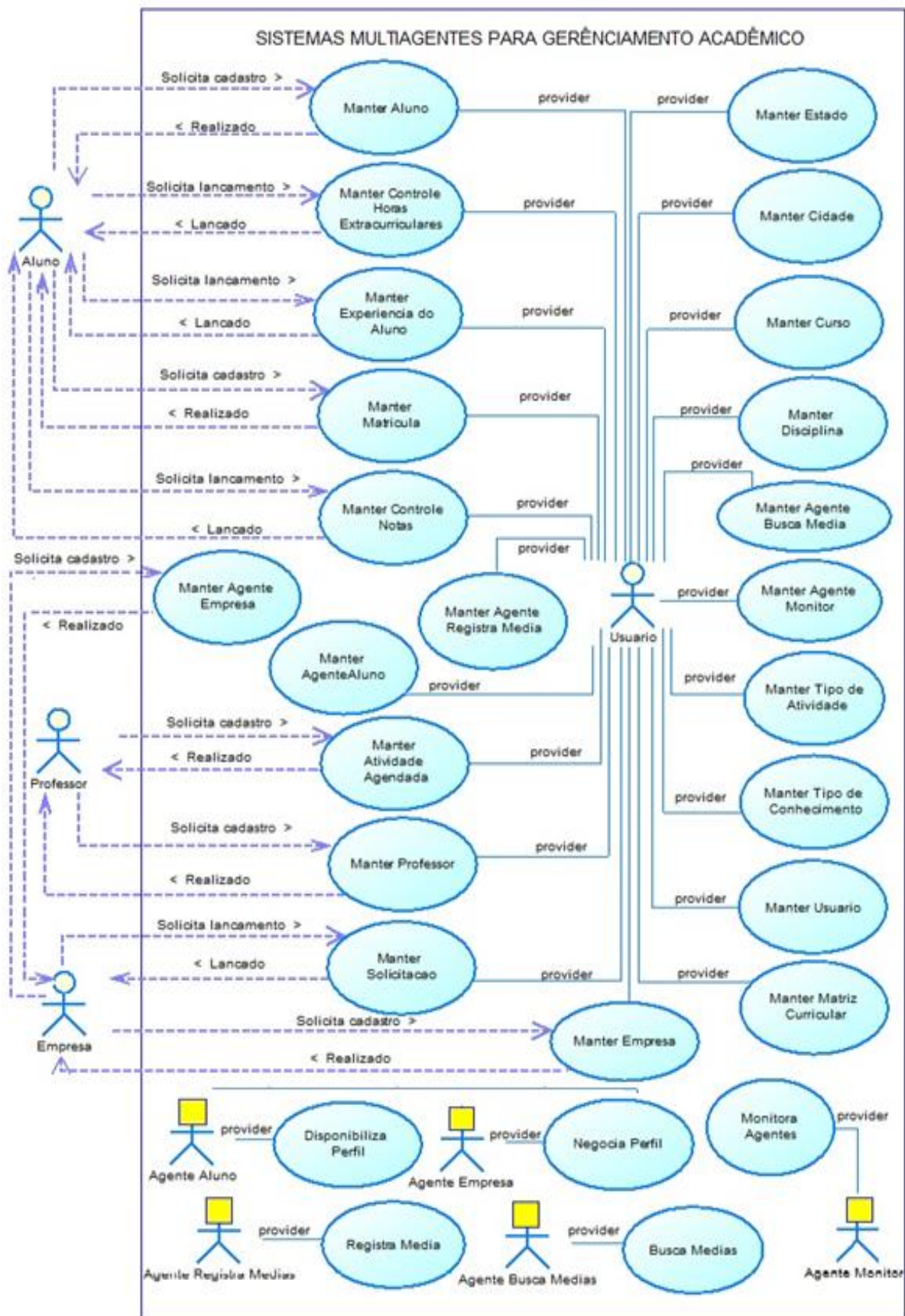
Tipo de Agente	Responsabilidades
Agente Aluno	1. Registrar entrada do agente no serviço de páginas amarelas.
	2. Servir as solicitações dos agentes empresas.
	3. Recusar a solicitação enviando uma resposta de recusa.
	4. Aceitar a solicitação enviando uma proposta.
	5. Processar a mensagem de retorno após a proposta.
	6. Informar que a solicitação foi concluída com sucesso.
	7. Informar que terminou e retorna ao iniciador os resultados
	8. Informar uma falha caso a solicitação falhar por algum motivo.
	9. Finalizar a missão do agente.
	10. Cancelar o registro das páginas amarelas.
Agente Empresa	1. Enviar para cada agente aluno uma mensagem de solicitação.
	2. Receber as propostas dos agentes alunos encontrados.
	3. Aceitar a proposta, enviando uma mensagem aceitando a proposta
	4. Rejeitar a proposta, enviando uma mensagem rejeitando a proposta.
	5. Receber a resposta com a informação final.
Agente Registra Médias	1. Registrar entrada do agente no serviço de páginas amarelas.
	2. Recusar o pedido do agente busca médias, enviando uma mensagem recusando o pedido.
	3. Aceitar o pedido do agente busca medias, enviando uma mensagem aceitando o pedido junto com uma notificação.
	4. Informar uma falha caso o pedido falhar por algum motivo.
	5. Informar que o pedido foi concluído com sucesso.
	6. Informar que o pedido terminou e retornar os resultados.
Agente Busca Médias	1. Enviar pedidos a agentes registra médias.
	2. Receber os pedidos recusados dos agentes registra médias
	3. Receber os pedidos aceitos dos agentes registra médias.
	4. Receber os pedidos que falharam por algum motivo.
	5. Receber os pedidos com informação que foram concluídos com sucesso.
	6. Receber os pedidos que terminaram e seus resultados.
Agente Monitor	1. Solicitar ao AMS os agentes ativos naquele momento.

FONTE: O Autor (2016).

4.1.3 Diagramas de Caso de Uso AUML

A FIGURA 20 ilustra o diagrama de caso de uso AUML de acordo com os conceitos de Sellers (2013) já citados no Capítulo 2 página 55.

FIGURA 20 – DIAGRAMA DE CASO DE USO SMA PARA GESTÃO ESCOLAR

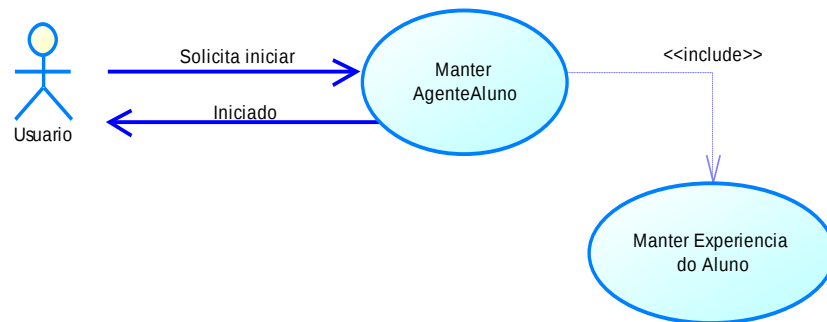


FONTE: O Autor (2016).

4.1.3.1 Diagrama de caso de uso manter agente aluno

Por meio dos casos de uso e seus relacionamentos, a FIGURA 21 apresenta como iniciador o ator Usuário. Quando um agente aluno é iniciado seu caso de uso precisa executar um método de busca do caso de uso manter experiência do aluno, então o esteriótipo de inclusão (*include*) é utilizado.

FIGURA 21 – DIAGRAMA DE CASO DE USO MANTER AGENTE ALUNO



FONTE: O Autor (2016).

Toda ação realizada na FIGURA 21 é detalhada no QUADRO 2.

QUADRO 2 – DETALHAMENTO CASO DE USO MANTER ALUNO

Manter Agente Aluno	
Ator Principal	Usuário
Agente	Agente Aluno
Resumo	Este caso de uso define a ativação do Agente Aluno no Sistema.
Ações	01. O Usuário abre a interface Histórico Contratação. 02. O Usuário clica no botão iniciar Agente Aluno. 03. O Agente Aluno é ativado com os conhecimento incluídos do caso de uso Manter Experiência do Aluno. 04. O Usuário recebe a confirmação dos Agentes Alunos ativos.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agente Aluno	
Plano de Ação	02a. Caso não houver nenhum argumento recebido do caso de uso Manter Experiência o agente Aluno não tem argumentos para ativar.
Interação	FrmHistoricoContratacao.java

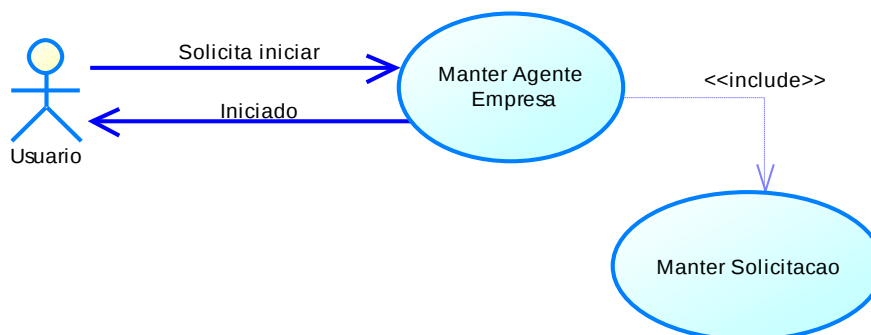
FONTE: O Autor (2016).

4.1.3.2 Diagrama de caso de uso manter agente empresa

A FIGURA 22 ilustra o diagrama de caso de uso manter agente empresa apresenta como iniciador o ator Usuário, o esteriótipo de inclusão (*include*) é

utilizado quando um agente empresa é iniciado, seu caso de uso precisa executar um método de busca do caso de uso manter solicitação.

FIGURA 22 – DIAGRAMA DE CASO DE USO MANTER AGENTE EMPRESA



FONTE: O Autor (2016).

Toda ação realizada na FIGURA 22 é detalhada no QUADRO 3.

QUADRO 3 – DETALHAMENTO CASO DE USO MANTER AGENTE EMPRESA

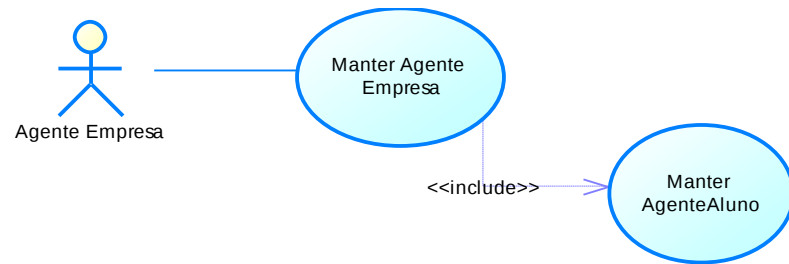
Manter Agente Empresa	
Ator Principal	Usuário
Agente	Agente Empresa
Resumo	Este caso de uso define a inicialização do Agente Empresa.
Ações	01. O Usuário abre a interface Histórico Contratação. 02. O Usuário clica no botão iniciar Agente Empresa. 03. O Agente Empresa é ativado com os argumentos incluídos do caso de uso manter solicitação. 04. O Usuário recebe a confirmação dos Agentes Empresas ativos e suas solicitações. 05. Após o tempo do comportamento de busca estiver terminado, serão apresentadas as buscas realizadas, solicitações encontradas e nenhuma solicitações pendentes. 06. O Agente Empresa é encerrado.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agente Empresa	
Plano de Ação	02a. Se o Agente Aluno não estiver ativo, uma mensagem de erro é informada. 06a. Caso houver alguma solicitação pendente retorna ao passo 05.
Interação	Agente Aluno

FONTE: O Autor (2016).

4.1.3.3 Diagrama de caso de uso interação agentes empresa e agentes aluno

A FIGURA 23 ilustra a interação entre agentes empresa e aluno, o esteriótipo de inclusão (*include*) é utilizado quando um agente empresa precisa interagir com um agente aluno.

FIGURA 23 – CASO DE USO INTERAÇÃO AGENTES EMPRESA E ALUNO



FONTE: O Autor (2016).

Toda ação realizada na FIGURA 23 é detalhada no QUADRO 4.

QUADRO 4 – DETALHAMENTO CASO DE USO INTERAÇÃO ENTRE AGENTE ALUNO E AGENTE EMPRESA

Interação entre Agente Empresa e Agente Aluno	
Ator Principal	Agente Empresa
Agente	Agente Empresa (iniciador)
Resumo	Este caso de uso define a comunicação entre os Agentes Aluno e Empresa.
Ações	01. O Agente Empresa envia ao Agente Aluno uma mensagem (cpf) solicitando propostas. 02. O Agente Aluno envia as propostas encontradas ao Agente Empresa. 03. O Agente Empresa envia uma mensagem aceitando a proposta recebida. 04. O Agente Aluno envia uma mensagem informando o resultado. 05 O Agente Empresa é finalizado.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agentes Aluno e Empresa	
Plano de Ação	02a. O Agente Aluno envia uma mensagem recusando a proposta feita pelo Agente Empresa. 03a. O Agente Empresa envia uma mensagem rejeitando a proposta. 04b O Agente Aluno envia uma mensagem cancelando a proposta recebida. 04c. O Agente Aluno envia uma mensagem de falha.
Interação	Agente Aluno (participante)

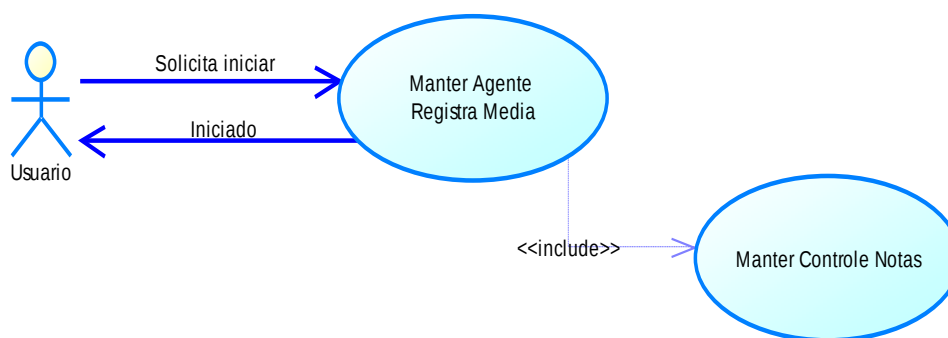
FONTE: O Autor (2016).

4.1.3.4 Diagrama de caso de uso manter agente registra média

A FIGURA 24 ilustra o diagrama de caso de uso manter agente registra média, apresenta como iniciador da ação o ator Usuário, o esteriótipo de inclusão

(*include*) é utilizado quando um agente registra média é iniciado, seu caso de uso precisa executar um método de busca do caso de uso manter controle notas.

FIGURA 24 – CASO DE USO MANTER AGENTE REGISTRA MÉDIA



FONTE: O Autor (2016).

Toda ação realizada na FIGURA 24 é detalhada no QUADRO 5.

QUADRO 5 – DETALHAMENTO CASO DE USO MANTER AGENTE REGISTRA MÉDIA

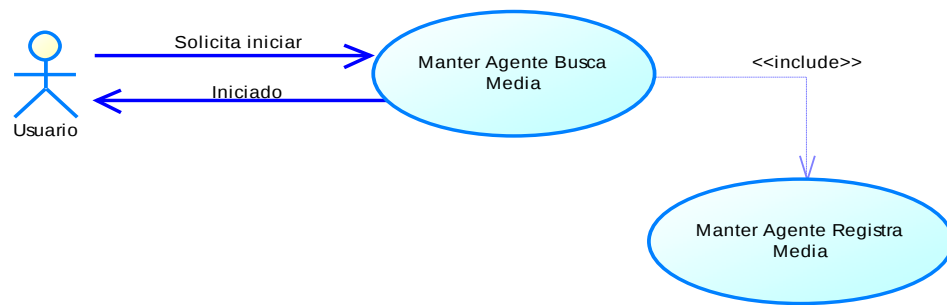
Manter Agente Registra Média	
Ator Principal	Usuário
Agente	Agente Registra Média
Resumo	Este caso de uso define a inicialização do Agente Registra Média no Sistema.
Ações	01. O Usuário abre a interface Histórico Notas. 02. O Usuário clica no botão iniciar Agente Registra Média. 03. O Agente Registra Média é iniciado. 04. O Usuário recebe a confirmação dos Agentes Registra Média ativos na interface Histórico Notas.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agente Registra Média	
Plano de Ação	02a. Caso ocorrer algum problema com a plataforma Jade, uma mensagem de erro é apresentada e o agente não é inicializado.
Interação	FrmHistoricoNotas.java

FONTE: O autor (2016).

4.1.3.5 Diagrama de caso de uso manter agente busca e registra média

Por meio dos casos de uso e seus relacionamentos, a FIGURA 25 apresenta como iniciador o ator Usuário. Quando um agente busca média é iniciado seu caso de uso precisa executar um método de busca do caso de uso manter agente registra média, então o esteriótipo de inclusão (*include*) é utilizado.

FIGURA 25 – CASO DE USO MANTER AGENTE BUSCA MÉDIA



FONTE: O Autor (2016).

Toda ação realizada na FIGURA 25 é detalhada no QUADRO 6.

QUADRO 6 – DETALHAMENTO CASO DE USO MANTER BUSCA MÉDIA

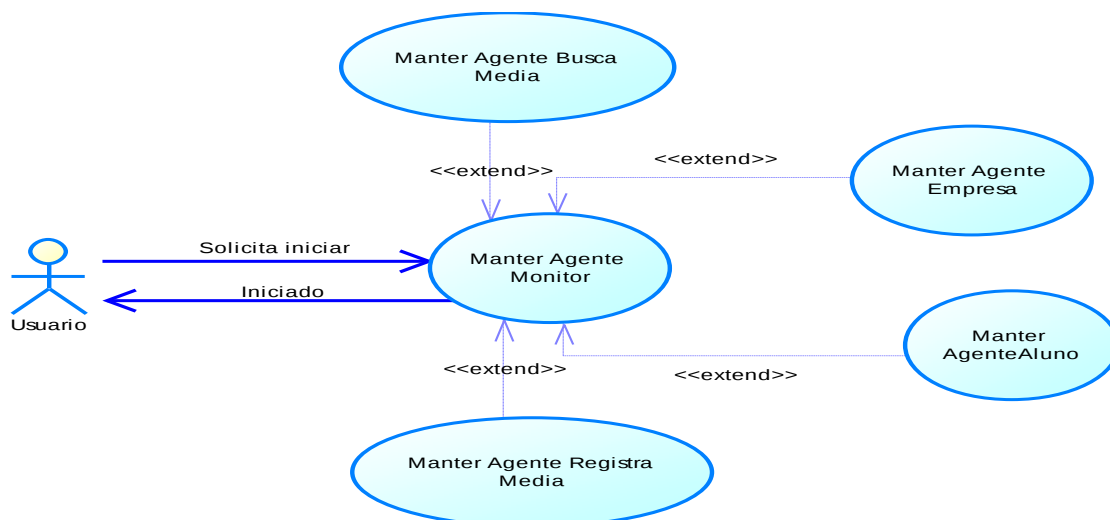
Manter Agentes Busca Média	
Ator Principal	Agente Busca Média
Agente	Agente Busca Média (iniciador).
Resumo	Este caso de uso define a comunicação entre os Agentes Registra Média e Busca Média.
Ações	01. O Agente Busca Média envia uma mensagem de pedido ao Agente Registra Média. 02. O Agente Registra Média envia uma mensagem aceitando o pedido. 03. O Agente Registra Média envia uma mensagem informando os resultados encontrados ao Agente Busca Média.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agentes Registra Média e Busca Média	
Plano de Ação	02a. O Agente Registra Média envia uma mensagem ao Agente Busca Média recusando o pedido feito pelo Agente Busca Médias. 03a. O Agente Registra Média envia uma mensagem de falha ao Agente Busca Média. 03b. O Agente Registra Média envia uma mensagem informando que fez o pedido ao Agente Busca Média.
Interação	Registra Médias (participante).

FONTE: O autor (2016).

4.1.3.6 Diagrama de caso de uso manter agente monitor

A FIGURA 26 ilustra o diagrama de caso de uso manter agente monitor, apresenta como iniciador do caso de uso o ator Usuário, o esteriótipo de estende (*extend*) é utilizado toda vez que o agente monitor precisa verificar quais agentes estão ativos na plataforma. Toda ação realizada é detalhada no QUADRO 7.

FIGURA 26 – CASO DE USO MANTER AGENTE MONITOR



FONTE: O Autor (2016).

QUADRO 7 – DETALHAMENTO CASO DE USO MANTER AGENTE MONITOR

Manter Agente Monitor	
Ator Principal	Usuário
Agente	Agente Monitor
Resumo	Este caso de uso define a comunicação entre o Agente Monitor e os demais agentes ativos.
Ações	01. O Usuário abre a interface Histórico Monitor. 02. O Usuário clica no botão iniciar Agente Monitor. 03. O Agente Monitor é Iniciado. 04. Através do serviço de páginas brancas do AMS o Agente Monitor consegue verificar todos os agentes que estão ativos. 04. Os resultados são apresentados na Interface Histórico Monitor.
Fluxo Alternativo	O Usuário poderá sair do Sistema a qualquer momento.
Modelo de Tarefas – Agente Monitor	
Plano de Ação	02a. Caso ocorrer algum problema com a plataforma Jade, uma mensagem de erro é apresentada e o agente não é inicializado. 04. Se não houver nenhum agente ativo nada será apresentado.
Interação	Agentes: Aluno, Empresa, Registra Medias e Busca Medias

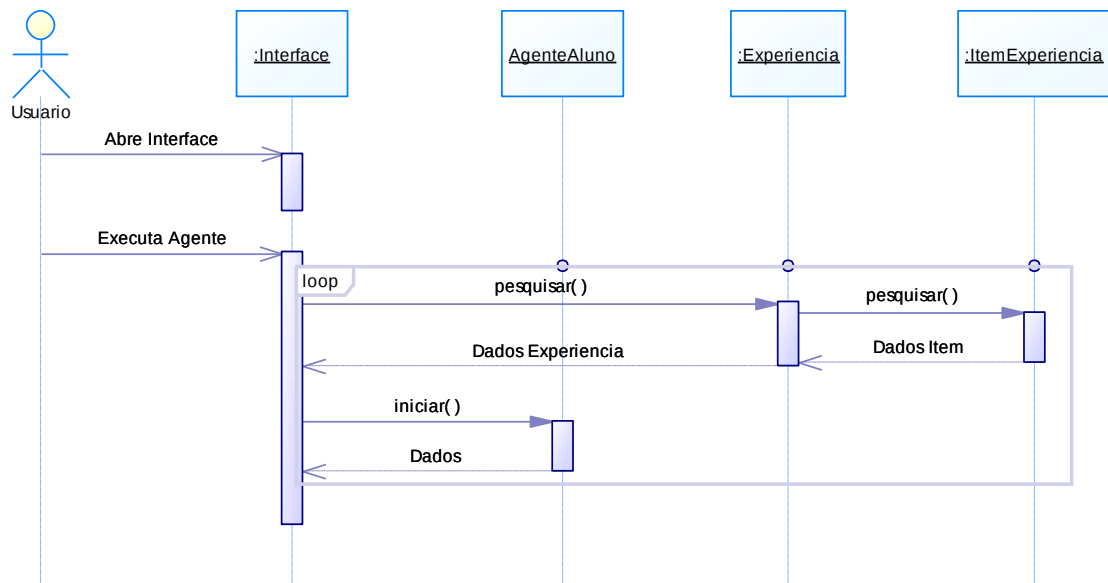
FONTE: O autor (2016).

4.1.4 Diagramas de Sequência AUML

4.1.4.1 Diagrama de sequência manter agente aluno

A FIGURA 27 apresenta a sequência de ativação do Agente aluno no sistema e as classes que fornecem dados para que o agente.

FIGURA 27 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE ALUNO

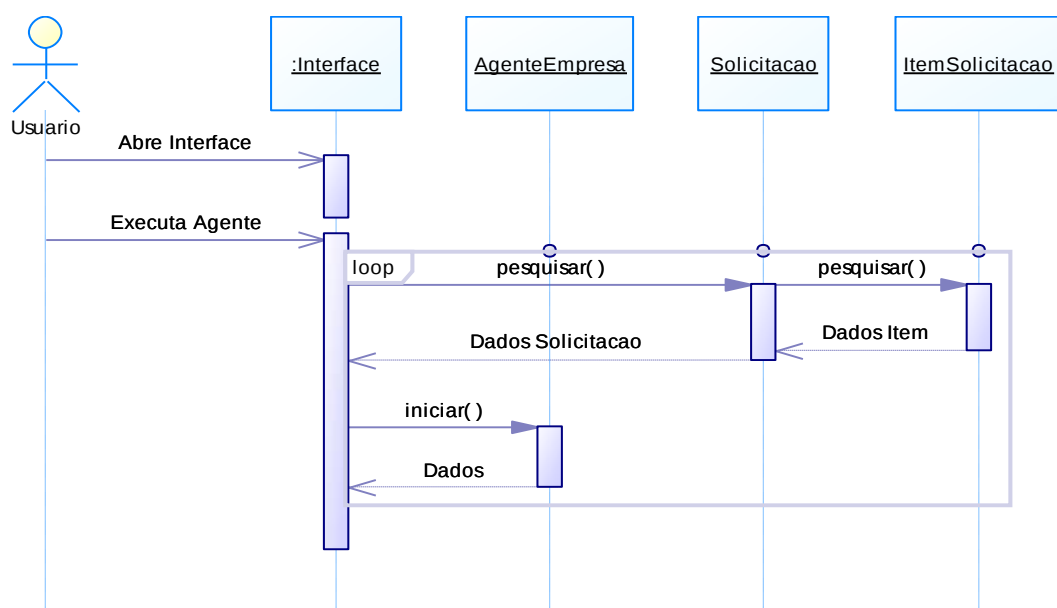


FONTE: O Autor (2016).

4.1.4.2 Diagrama de sequência manter agente empresa

A FIGURA 28 apresenta a sequência de ativação do agente empresa no sistema e as classes que fornecem dados para o agente.

FIGURA 28 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE EMPRESA

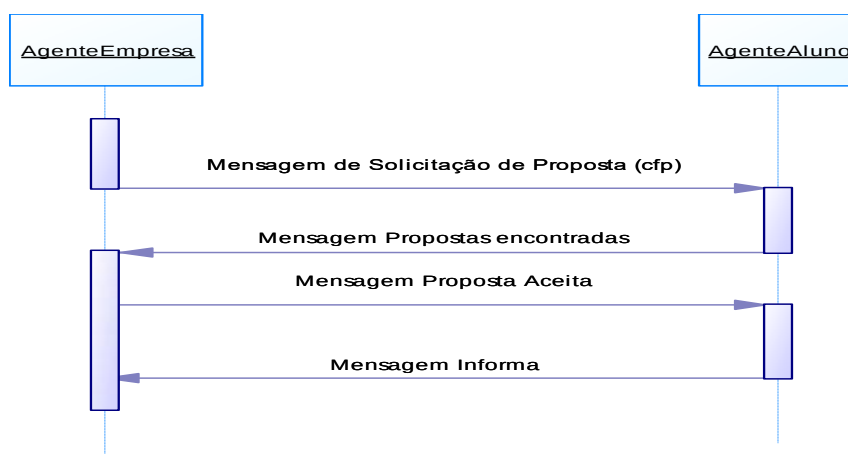


FONTE: O Autor (2016)

4.1.4.3 Diagrama de sequência interação agente aluno e agente empresa

A FIGURA 29 apresenta a sequência de comunicação entre agentes com base no protocolo FIPA *request interaction protocol* já citado no Capítulo 2 página 43.

FIGURA 29 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE ALUNO E AGENTE EMPRESA

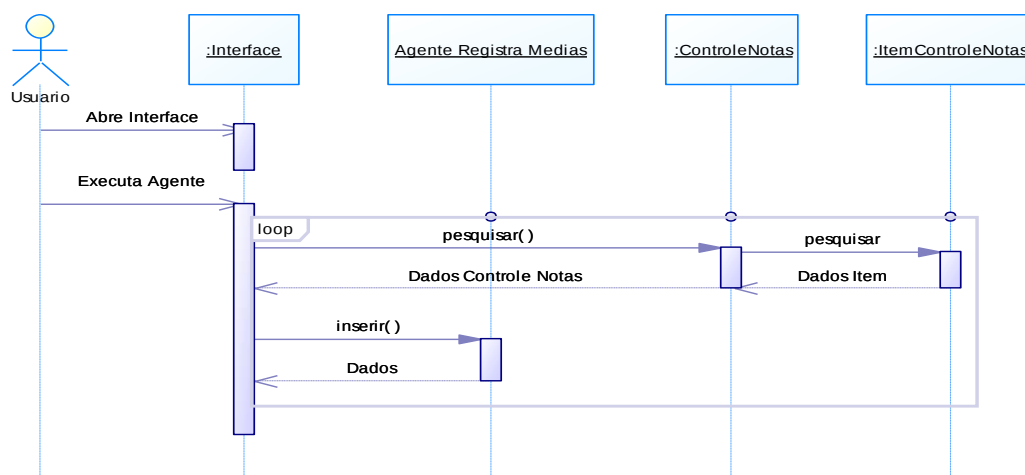


FONTE: O Autor (2016).

4.1.4.4 Diagrama de sequência manter agente registra médias

A FIGURA 30 apresenta a sequência de ativação do agente registra médias no sistema e as classes que fornecem dados para o agente.

FIGURA 30 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE REGISTRA MÉDIAS

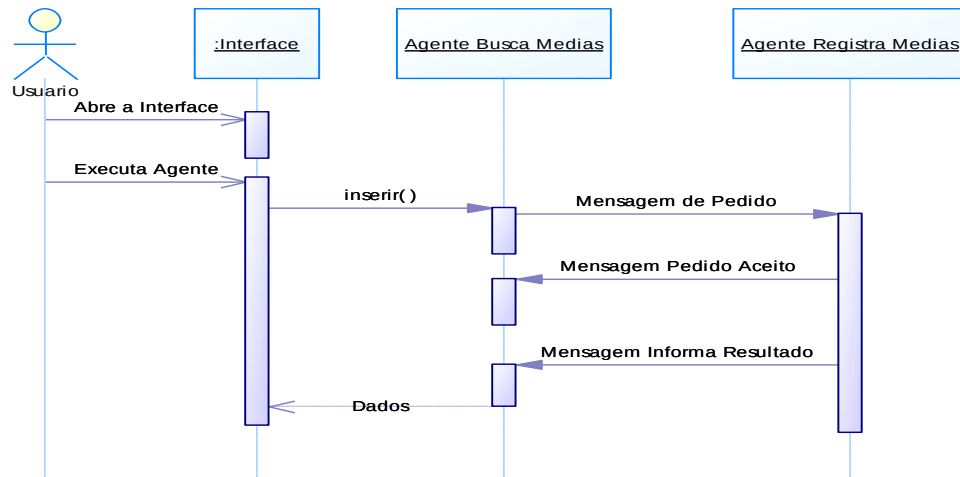


FONTE: O Autor (2016).

4.1.4.5 Diagrama de sequência manter agente busca médias

A FIGURA 31 apresenta a sequência de comunicação entre agentes com base também no protocolo FIPA *request interaction protocol*.

FIGURA 31 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE BUSCA MÉDIAS

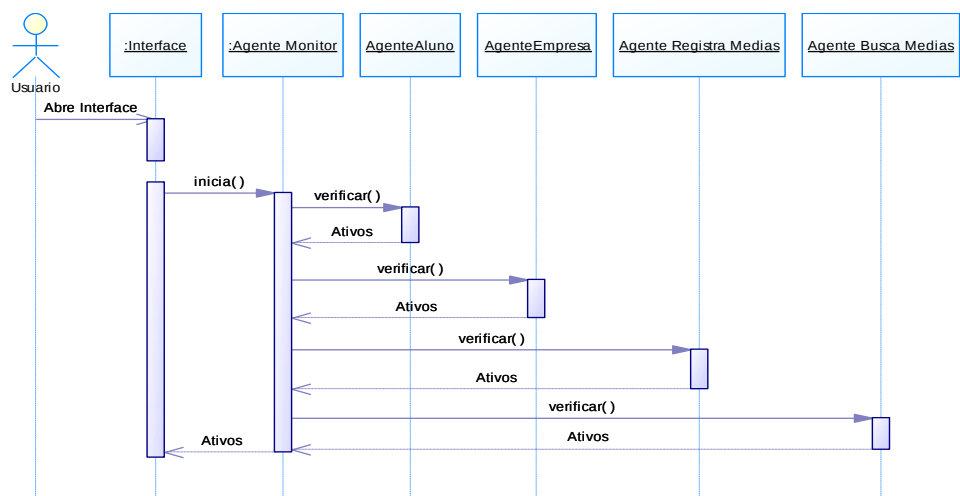


FONTE: O Autor (2016)

4.1.4.6 Diagrama de sequência manter agente monitor

A FIGURA 32 apresenta a sequência de ativação do agente monitor e as classes de agentes envolvidas. O agente monitor verifica em cada classe de agente quais estão ativos.

FIGURA 32 – DIAGRAMA DE SEQUÊNCIA MANTER AGENTE MONITOR



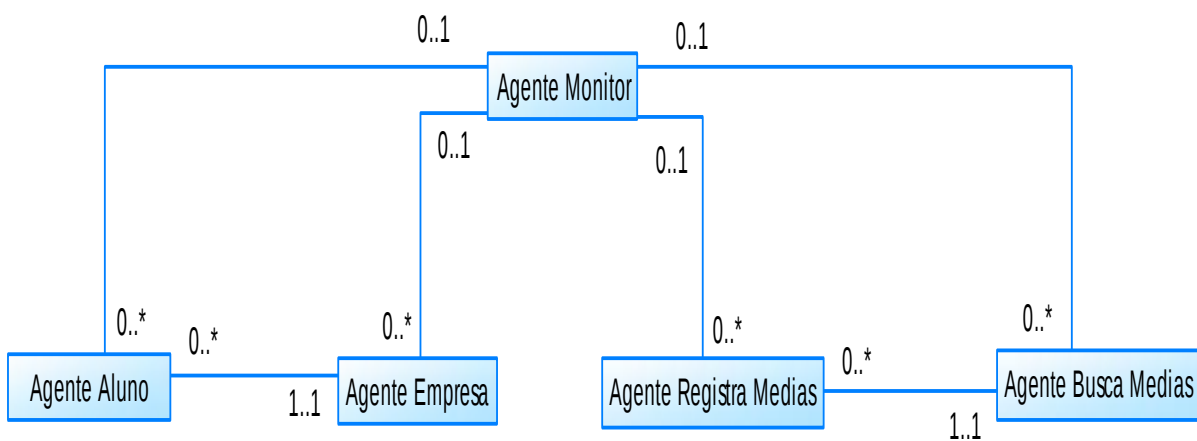
FONTE: O Autor (2016).

4.1.5 Diagrama de Classes AUML

O Diagrama de classes AUML foi desenvolvido com base no modelo proposto por Bauer e Odell (2005) já mencionado no Capítulo 2 página 55. As Classes são representadas por retângulos, é possível verificar as associações que ocorrem entre os agentes e suas multiplicidades.

A FIGURA 33 ilustra a extensão do diagrama de classes para agente do modelo proposto.

FIGURA 33 – DIAGRAMA DE CLASSES AUML SMA



FONTE: O Autor (2016).

5 TECNOLOGIAS EMPREGADAS

Neste capítulo são apresentadas as tecnologias, que foram utilizadas para o desenvolvimento do SMA proposto.

Os diagramas UML e AUML que formam a modelagem do SMA foram construídos na ferramenta de modelagem Power Designer 15.1. No desenvolvimento do SMA foi utilizada a linguagem de programação Java, através da plataforma de desenvolvimento Java Se Development Kit 8 (64 Bits), juntamente com o ambiente de desenvolvimento integrado *NETBEANS* IDE 8.0.2. Para criar os agentes e realizar sua execução, utilizou-se o Framework Jade 4.4.0. Um Framework já adaptado ao padrão de comunicação de agentes FIPA.

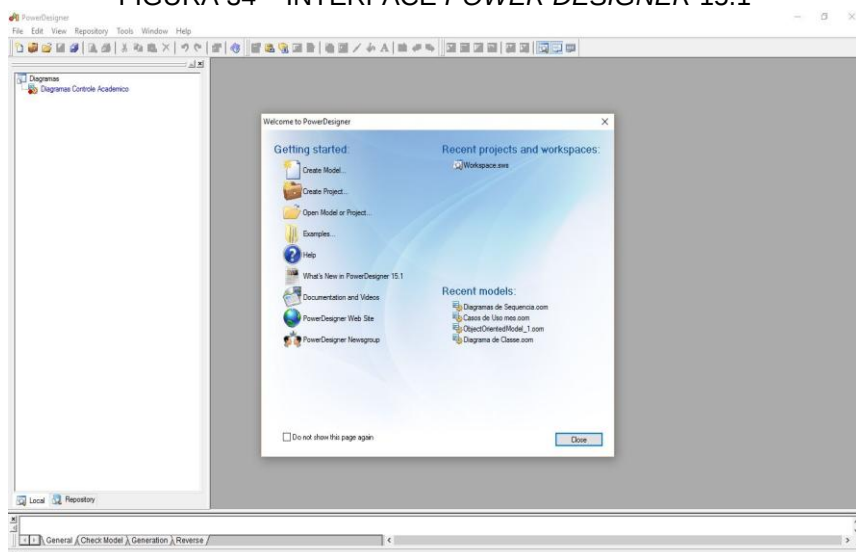
O banco de dados utilizado foi o *MySQL Server* 5.5, sendo construído na ferramenta visual para desenvolvimento de SQL *MySQL Workbench* 5.2, que possibilita a criação de e manipulação de tabelas de uma forma visual.

5.1 FERRAMENTA DE MODELAGEM UML

O *PowerDesigner* foi utilizado para desenvolver todos os diagramas UML e AML que fazem parte da proposta. Sua interface é intuitiva e possibilitou a adaptação dos diagramas AUML. (SYBASE, 2016)

A FIGURA 34 apresenta a interface do *PowerDesigner*.

FIGURA 34 – INTERFACE POWER DESIGNER 15.1



FONTE: O autor (2016).

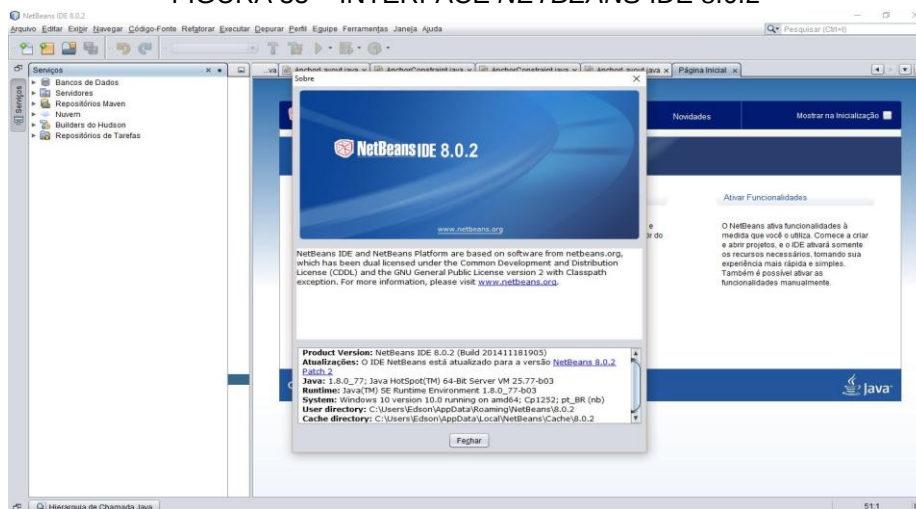
5.1.1 Ferramenta de Desenvolvimento

Segundo ORACLE (2015), o Java Development Kit (JDK) é um ambiente de desenvolvimento para a criação de aplicativos, applets, e componentes usando a linguagem de programação Java. O JDK inclui ferramentas úteis para desenvolver e testar programas escritos na linguagem de programação Java e em execução na plataforma Java.

5.1.2 Ambiente de Desenvolvimento Integrado

Netbeans é o primeiro IDE a oferecer suporte a JDK 8 gratuito e de código aberto para desenvolvedores de software nas linguagens Java, C, C++, PHP, Groovy, Ruby, entre outras. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. (NETBEANS 2015). A FIGURA 35 ilustra a interface da IDE *NETBEANS*.

FIGURA 35 – INTERFACE *NETBEANS* IDE 8.0.2



FONTE: O autor (2016).

5.1.3 Framework JADE

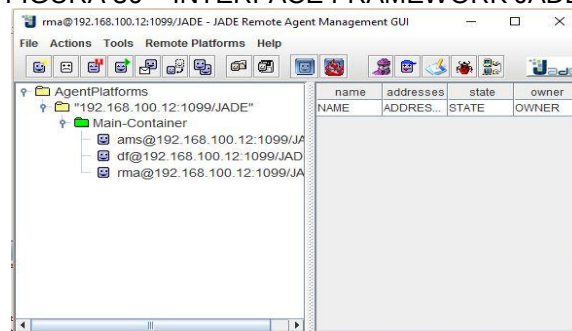
De acordo com Bellifemine, Caire e GreenWood (2007, p. 29) a plataforma JADE, surgiu no final de 1998 desenvolvida pela Telecom Itália, motivada pela necessidade de validar o início das especificação da FIPA. É uma plataforma de software que fornece funcionalidades básicas que são independentes de uma

aplicação específica o que possibilita o uso em aplicações distribuídas, possibilitando sua utilização em dispositivos fixos ou móveis.

O JADE é de natureza dinâmica, onde os agentes podem ser criados e finalizados a qualquer momento. Para a aplicação, cada agente só pode ser identificado por um nome único, não podem existir dois agentes com o mesmo número de identificação (AID). A plataforma pode registrar ou procurar agentes, controlar o ciclo de vida dos agentes e proporciona comunicação entre agentes por troca de mensagens assíncronas.

Na comunicação entre agentes, enquanto um agente envia uma mensagem ele não pode receber nenhuma comunicação enquanto não estiver finalizado o envio. A FIGURA 36 ilustra a interface da ferramenta.

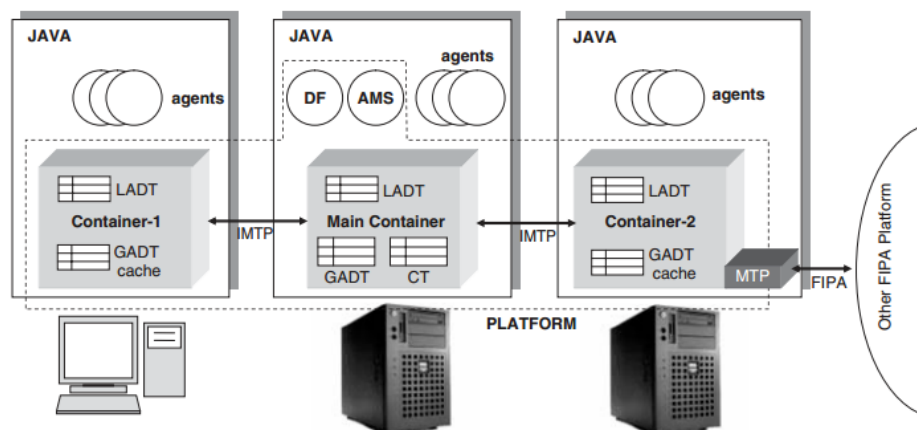
FIGURA 36 – INTERFACE FRAMEWORK JADE



FONTE: O autor (2016).

O JADE utiliza o modelo de referência de gerência de agentes padrão FIPA já citado no Capítulo 2 na página 37. Os componentes da arquitetura JADE são ilustrados na FIGURA 37.

FIGURA 37 – PRINCIPAIS ELEMENTOS ARQUITETÔNICOS DO JADE

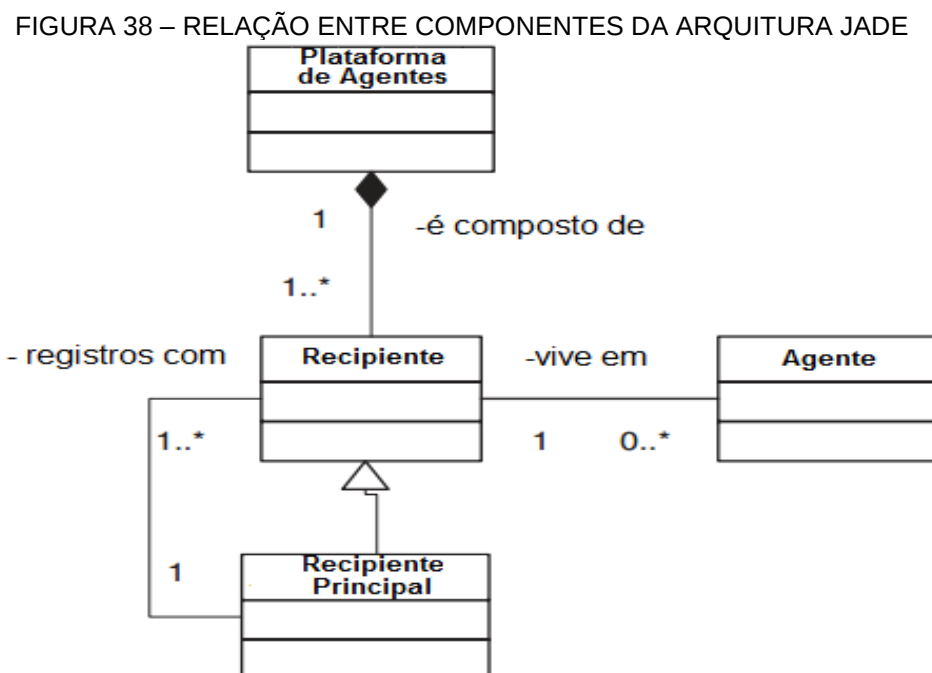


FONTE: Bellifemine, Caire e Greenwood (2007, p.32).

Para Bellifemine, Caire e GreenWood (2007, p. 33-34) a arquitetura da plataforma JADE é composta por:

- O Managing the Container Table (CT) é o registro das referências de objetos e transportes endereços de todos os nós de recipientes que compõem a plataforma.
- O AMS é o agente que supervisiona toda a plataforma, todo agente é obrigado a realizar seu registro no AMS para conseguir interagir com outros agentes e gerenciar seu ciclo de vida, o AMS mantém uma lista de identificadores de agentes (AID) e seus estados.
- O Diretório Facilitador (DF) é o agente que implementa o serviço de páginas amarelas, usado por qualquer agente que pretende registrar os seus serviços ou procurar outros serviços disponíveis.
- O Serviço de Transporte de Mensagens (MTS) provê a comunicação entre os agentes. Esta comunicação é realizada através da Linguagem de Comunicação de Agentes (ACL).
- Linguagem de Comunicação de Agentes (ACL) é usada para codificar as mensagens trocadas entre os agentes, independentemente da linguagem de programação utilizada pela aplicação.

A FIGURA 38 ilustra os principais componentes da arquitetura do JADE.

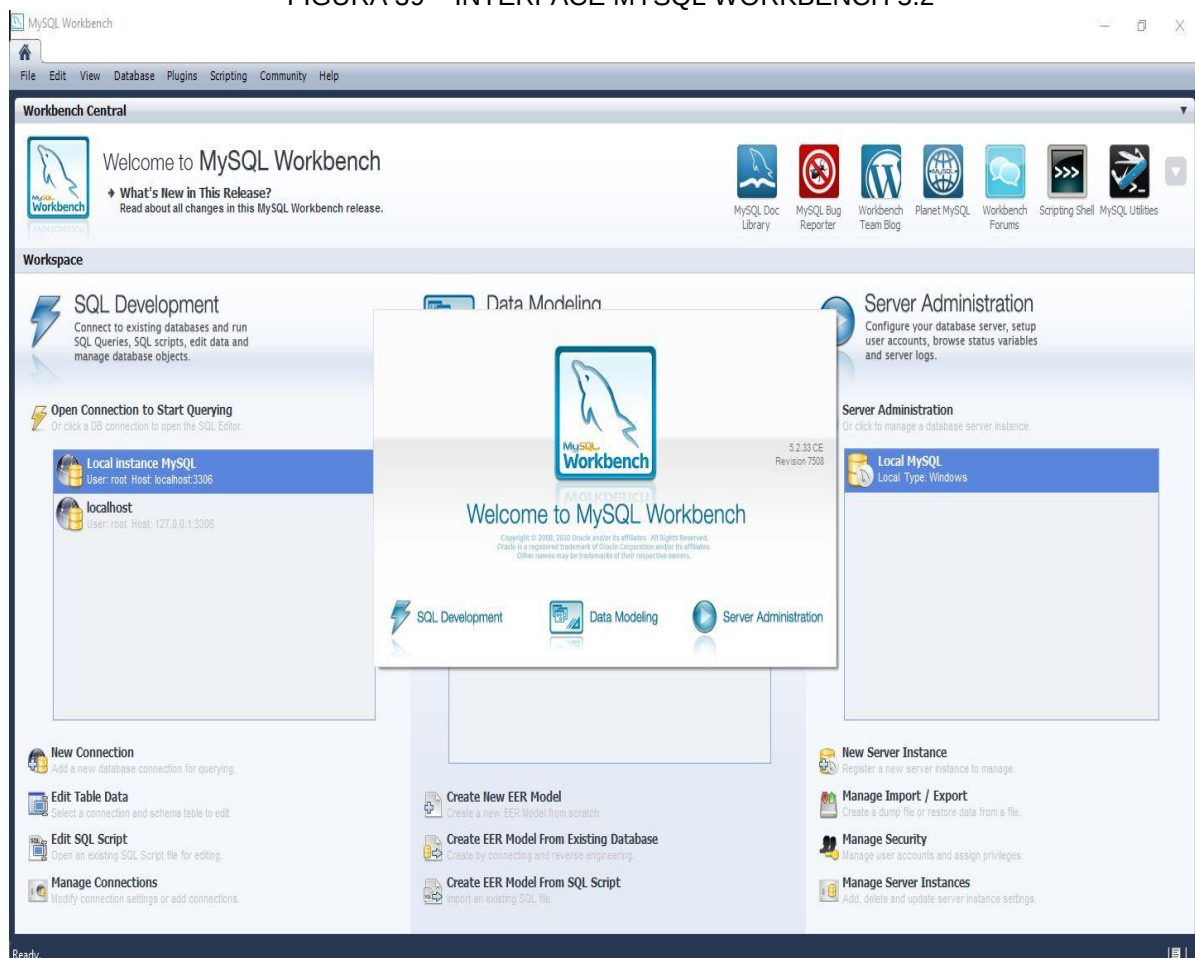


FONTE: Adaptado de: Bellifemine, Caire e GreenWood (2007 p.33).

5.1.4 Ferramenta para Desenvolvimento de Banco de Dados

MySQL Workbench é uma ferramenta visual para desenvolvedores de banco de dados e *DataBase Administrator (DBA)*. *MySQL Workbench* fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração para configuração do servidor de banco de dados. *MySQL Workbench* está disponível no *Windows*, *Linux* e *Mac OS X*. (MYSQL, 2016b). A FIGURA 39 ilustra a interface da ferramenta.

FIGURA 39 – INTERFACE MYSQL WORKBENCH 5.2



FONTE: O autor (2016).

5.1.5 Gerenciador de Banco de Dados

O MySQL é um SGBD, um Sistema de Gerenciamento de Banco de Dados, que usa a linguagem SQL como interface. É um banco de Dados *Open Source* cujo

sucesso deve-se em grande medida à fácil integração com o PHP, incluído quase que obrigatoriamente, nos pacotes de hospedagem de sites da Internet oferecidos atualmente. (MYSQL, 2016a)

6 EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Este capítulo apresenta os sete experimentos e seus respectivos resultados, que serviram para avaliar as funcionalidades e interações dos agentes no sistema proposto e validar os protocolos de agentes FIPA utilizados. Todos os experimentos foram realizados com dados específicos para o curso de “Sistemas de Informação”, entretanto o sistema pode ser adaptado para qualquer curso, para isto basta alterar os dados que compõem o perfil de conhecimentos do aluno.

6.1 DETALHAMENTO DOS EXPERIMENTOS

6.1.1 Experimentos Agente Aluno e Agente Empresa

Foram realizados cinco experimentos com os agentes aluno e empresa, que serviram para avaliar os resultados do SMA proposto e validar a utilização o protocolo *FIPA Contract Net Iteration*.

Os quatro primeiros experimentos foram realizados a priori utilizando bases de dados simuladas em um cenário utilizando nove participantes, dos quais cinco são alunos (ALUNO 1, ALUNO 2, ALUNO 3, ALUNO 4 e ALUNO 5) e três empresas (EMPRESA 1, EMPRESA 2 e EMPRESA 3). Cada simulação ativa os agentes alunos e agentes empresas com argumentos diferentes para testar a resposta dada por eles. No quinto experimento, foram usados dados reais de uma turma do terceiro ano do curso de “Sistemas de Informação” de uma instituição de ensino superior.

6.1.2 Experimentos Agente Registra Médias e Agente Busca Médias

Foram realizados dois experimentos com os agentes registra médias e busca médias, que serviram para avaliar os resultados do SMA proposto e validar a utilização o protocolo *FIPA Contract Net Iteration*.

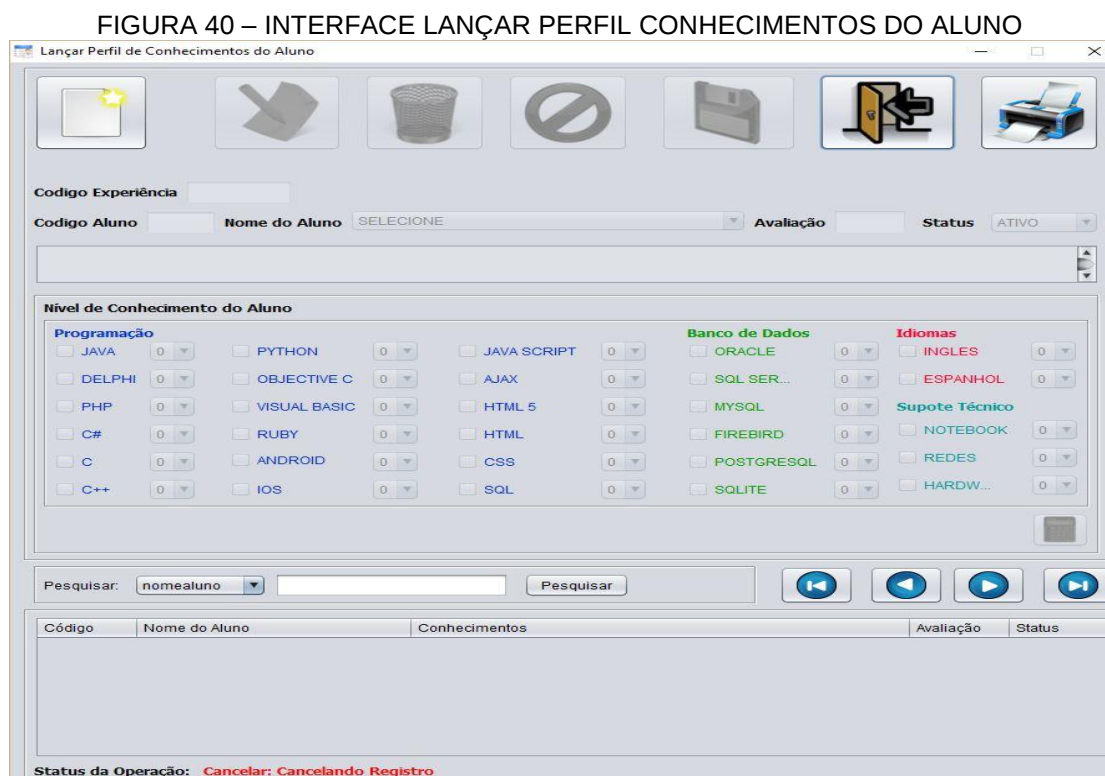
O primeiro experimento foi realizado a priori utilizando bases de dados simulados em um cenário composto por cinco alunos (ALUNO 1, ALUNO 2, ALUNO 3, ALUNO 4 e Aluno 5), onde cada aluno recebeu duas notas que formaram a média bimestral, esta média foi utilizada para testar a resposta dada pelos agentes

que registram as médias baixas disponibilizando um serviço, que depois é utilizado pelo agente busca médias. Utilizando dados para testar o protocolo *FIPA Contract Net Iteration*. No segundo experimento, foram usados dados reais de uma turma do terceiro ano do curso de “Sistemas de Informação” de uma instituição de ensino superior.

6.2 EXPERIMENTOS AGENTE ALUNO E AGENTE EMPRESA

6.2.1 Lançar Perfil de Conhecimentos do Agente Aluno

A interface lançar perfil de conhecimentos do aluno é utilizada para lançar os conhecimentos do aluno e o nível de cada conhecimento, ela é dividida em duas partes principais: a) os campos para preenchimento dos dados do Aluno; b) os campos para selecionar os conhecimentos e o nível de cada um, o nível de conhecimento vai de 0 (zero) a 5 (cinco), após os dados serem preenchidos é realizado o cálculo da média dos conhecimentos informados e seus resultados são salvos em uma tabela do Banco de Dados.. A FIGURA 40 ilustra a interface utilizada para lançar os perfis de conhecimento dos alunos e calcular sua média.



FONTE: O autor (2016)

6.2.2 Lançar Solicitação de Empresas

A interface lançar solicitações de empresas é utilizada para preencher o perfil de conhecimentos solicitados em uma vaga de emprego ou estágio ofertada por uma empresa, ela é dividida em duas partes principais: a) os campos para preenchimento dos dados da Empresa; b) os campos para selecionar os conhecimentos para definir o perfil procurado, após os dados serem preenchidos são salvos em uma do Banco de Dados, para criar os agentes empresa o SMA proposto utiliza essas informações A FIGURA 41 ilustra esta interface.

FIGURA 41 – INTERFACE LANÇAR SOLICITAÇÕES DE EMPRESAS

Lançar Solicitação de Empresas

Codigo
 Empresa
 Seleccione
 Data Cadastro
 Status: ATIVO

Conhecimentos Requeridos

Nível de Conhecimento do Aluno

Programação			Banco de Dados	Idiomas
☐ JAVA	☐ PYTHON	☐ JAVASCRIPT	☐ ORACLE	☐ INGLES
☐ DELPHI	☐ OBJECTIVE C	☐ AJAX	☐ SQL SER...	☐ ESPANHOL
☐ PHP	☐ VISUAL BASIC	☐ HTML 5	☐ MYSQL	Supote Técnico
☐ C#	☐ RUBY	☐ HTML	☐ FIREBIRD	☐ NOTEBOOK
☐ C	☐ ANDROID	☐ CSS	☐ POSTGRESQL	☐ REDES
☐ C++	☐ IOS	☐ SQL	☐ SQLITE	☐ HARDWARE

Pesquisar: empresa

Codigo	Empresa	Data Cadastro	Conhecimentos Solicitados

Status da Operação: **Navegar: Está no Primeiro Registro**

FONTE: O autor (2016).

6.2.3 Interface Acompanhamento Agentes Ativos

A interface de acompanhamento de solicitações de empresas apresenta todos os resultados do SMA. Ela é dividida em abas, cada aba recebe uma informação do agente podendo ser salva em arquivos txt ou no caso das solicitações encontradas, podem ser enviadas para qualquer email.

Os agentes alunos ativos estão registrados na plataforma disponibilizando um serviço do tipo “perfil de conhecimentos”, os agentes empresa são configurados para procurar agentes que fornecem este tipo serviço para iniciar a interação. A FIGURA 42 ilustra a interface de acompanhamento de solicitações das empresas. No quadro agente alunos são informados os ativos e no quadro agentes empresa os agentes empresas ativos. É possível parar qualquer um dos agentes a qualquer momento.

FIGURA 42 – INTERFACE ACOMPANHAMENTO DE SOLICITAÇÕES



FONTE: O autor (2016).

6.2.4 Interface Acompanhamento Buscas Realizadas

As buscas realizadas retornam sempre os mesmos resultados, pois em todas as simulações serão utilizados sempre os mesmos agentes, alterando apenas as informações que serão utilizadas para inicia-los, desenvolvemos uma interface para acompanhamento de buscas realizadas, são apresentadas três solicitações que compreendem ao número de agentes empresa ativos, o intervalo de busca foi configurado no comportamento do agente empresa para ser repetido a cada 1

minuto, podendo ser interrompido a qualquer momento, ou finalizado quando todas as solicitações são encontradas. A FIGURA 43 ilustra os resultados encontrados.

FIGURA 43 – INTERFACE BUSCAS REALIZADAS



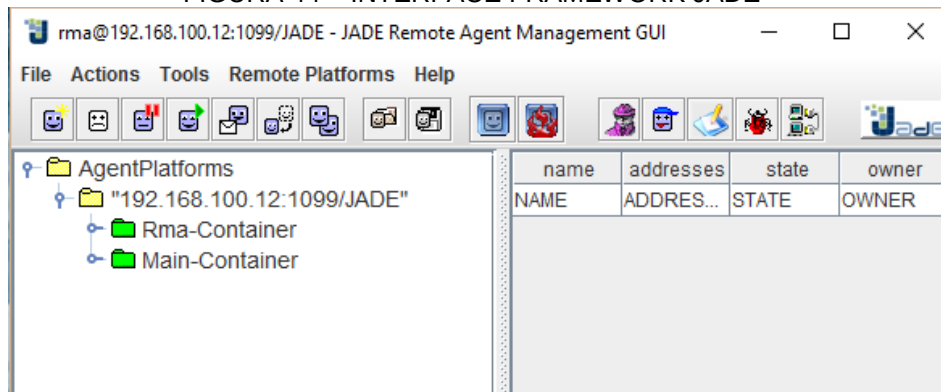
FONTE: O autor (2016).

6.2.5 Interface Gráfica do Jade

A plataforma de agentes Jade é iniciada automaticamente sem o modo gráfico quando o sistema é iniciado, para poder receber os agentes e possibilitar a interação entre eles, desenvolvemos uma opção para abrir a interface gráfica através da Interface principal do sistema, esta opção está disponível no menu iniciar agentes, opção iniciar a plataforma gráfica do Jade.

A FIGURA 44 ilustra a plataforma JADE padrão FIPA e seus componentes.

FIGURA 44 – INTERFACE FRAMEWORK JADE



FONTE: O autor (2016)

6.2.6 Interface Gráfica do Agente *Jade Sniffer*

Toda comunicação entre agentes pode ser monitorada por um componente da plataforma Jade o Agente *Sniffer*. Foi desenvolvida uma opção para acessar o agente diretamente do sistema, sem precisar acessar a plataforma Jade. Ele é acessado da interface principal do sistema, através do Menu Agentes Inteligentes, na opção Iniciar Agente *Sniffer* conforme ilustra a FIGURA 45.

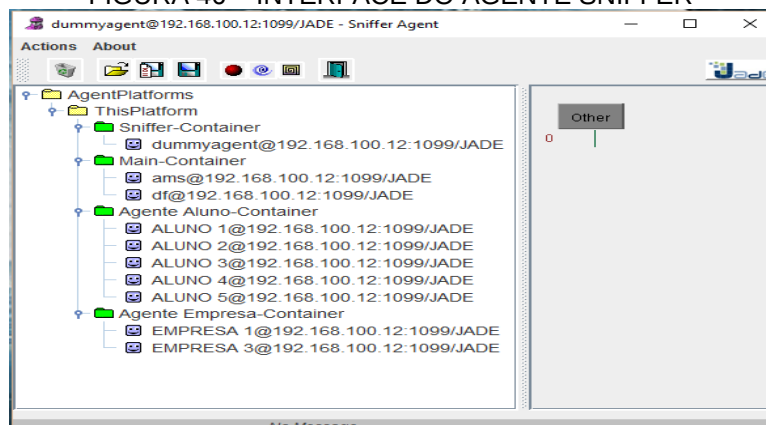
FIGURA 45 – MENU DE ACESSO AGENTE SNIFFER



FONTE: O autor (2016).

A FIGURA 46 ilustra a interface do agente Sniffer com os agentes ativos.

FIGURA 46 – INTERFACE DO AGENTE SNIFFER



FONTE: O autor (2016).

6.3 EXPERIMENTO 1 AGENTE ALUNO E AGENTE EMPRESA

O experimento 1, apresenta um cenário onde nenhum agente aluno possui os perfis de conhecimentos solicitados pelos agentes empresa. No QUADRO 8 são apresentados o perfil de conhecimentos de cada agente aluno e seu nível de conhecimento. As informações estão armazenadas em um banco de dados.

QUADRO 8 – PERFIS DE CONHECIMENTOS NO EXPERIMENTO 1

Agente	Perfil de Conhecimentos	Nível
ALUNO 1	JAVA – ORACLE	3,2
ALUNO 2	DELPHI - SQL SERVER – INGLES	4,5
ALUNO 3	PHP – MYSQL – INGLES	3,4
ALUNO 4	C# - FIREBIRD	3,4
ALUNO 5	C - SQL SERVER	4,0

FONTE: O autor (2016).

Os agentes empresa e o perfil de conhecimentos são apresentados no QUADRO 9, às informações também estão armazenadas em uma base de dados.

QUADRO 9 – PERFIS SOLICITADOS NO EXPERIMENTO 1

Agente	Perfil Conhecimentos
EMPRESA 1	C++ - SQLITE
EMPRESA 2	PYTHON - JAVA SCRIPT – ORACLE
EMPRESA 3	OBJECTIVE C - AJAX - SQL SERVER

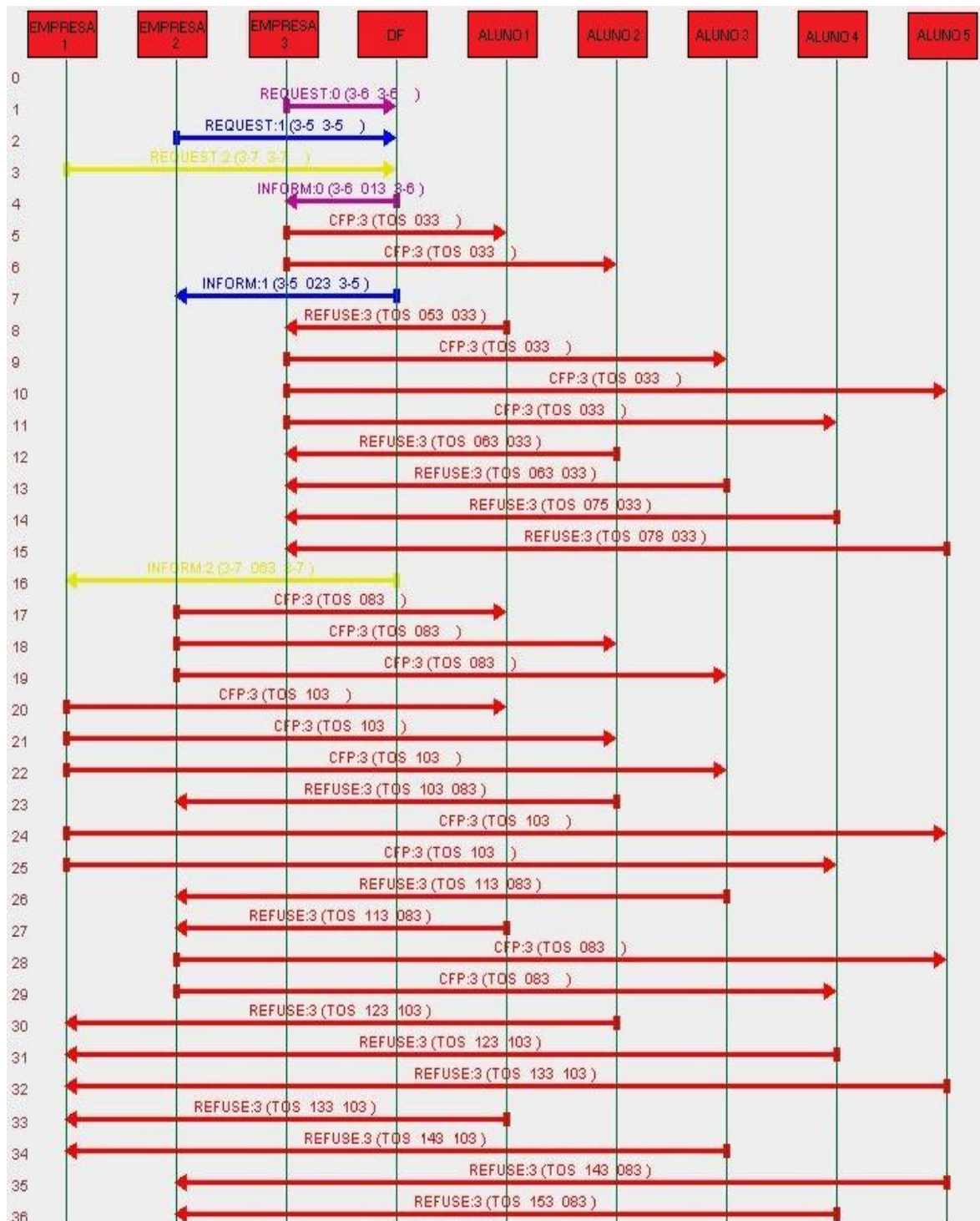
FONTE: O autor (2016).

6.3.1 Resultados do *Jade Agente Sniffer* Experimento 1

Para iniciar o acompanhamento com o agente *Sniffer*, cada agente aluno registra o tipo de serviço que fornece no serviço de páginas amarelas da plataforma Jade o DF. A comunicação é iniciada pelos agentes empresa, cada agente empresa envia um pedido (*request*) ao DF solicitando quais agentes fornecem o serviço “perfil de conhecimentos”, o DF informa para cada agente empresa os agentes que fornecem o tipo de serviço solicitado, ao receber a informação os agentes empresa enviam para cada agente aluno uma mensagem (*cfp*), esta mensagem possui o perfil de conhecimentos solicitados juntamente com uma solicitação de proposta. No experimento 1, os agentes aluno não possuem o perfil de conhecimentos solicitados, então todos respondem recusando a solicitação com uma mensagem (*recuse*).

A FIGURA 47 ilustra todas as interações entre os agentes aluno e agentes empresa monitoradas pelo agente *Sniffer*. A linhas são representadas por números e nelas o tipo de interação que ocorreu entre os agentes.

FIGURA 47 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 1



FONTE: O Autor (2016).

6.3.2 Análise dos Resultados Experimento 1

Na TABELA 1, são apresentados os resultados quantificados das interações entre os agentes aluno e empresa no experimento 1.

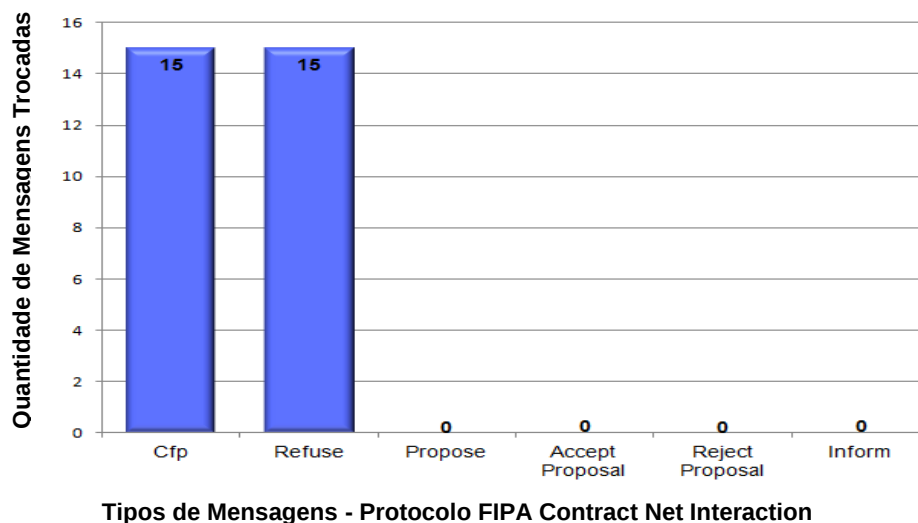
Os resultados utilizados foram os apresentados pelo agente *Sniffer* na FIGURA 47.

TABELA 1 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 1

Agente Empresa	Cfp	Refuse	Propose	Accept Proposal	Reject Proposal	Inform	Agente Aluno
EMPRESA 1	20	33	0	0	0	0	ALUNO 1
	21	30	0	0	0	0	ALUNO 2
	22	34	0	0	0	0	ALUNO 3
	25	32	0	0	0	0	ALUNO 4
	24	31	0	0	0	0	ALUNO 5
EMPRESA 2	17	27	0	0	0	0	ALUNO 1
	18	23	0	0	0	0	ALUNO 2
	19	25	0	0	0	0	ALUNO 3
	29	35	0	0	0	0	ALUNO 4
	28	36	0	0	0	0	ALUNO 5
EMPRESA 3	5	8	0	0	0	0	ALUNO 1
	6	12	0	0	0	0	ALUNO 2
	9	13	0	0	0	0	ALUNO 3
	11	14	0	0	0	0	ALUNO 4
	10	12	0	0	0	0	ALUNO 5
	15	15	0	0	0	0	

FONTE: O Autor (2016).

Para uma melhor compreensão da TABELA 1 consideramos o primeiro agente empresa da primeira coluna, que tem o nome de EMPRESA 1. A EMPRESA 1 enviou uma mensagem ao ALUNO 1 do tipo *cfp* na linha 20, na linha 21 ao ALUNO 2, na linha 22 ao ALUNO 3, na linha 25 ao ALUNO 4 e na linha 24 ao ALUNO 5, conforme ilustra a FIGURA 47, a linha onde aconteceu a interação entre os agentes é numerada. Como foram 15 mensagens do tipo *refuse* recebidas pelos agentes empresa e nenhuma mensagem do tipo *propose*, podemos concluir que não foi encontrado nenhum agente com os conhecimentos solicitados. O GRÁFICO 1 ilustra as quantidades apresentadas.

GRÁFICO 1 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 1

FONTE: O autor (2016).

6.4 EXPERIMENTO 2 AGENTE ALUNO E AGENTE EMPRESA

No experimento 2 é apresentado um cenário onde são encontrados todos os perfis solicitados pelos agentes empresa. As interfaces seguem o mesmo padrão anterior e serão comentadas somente quando uma informação diferente ou relevante surgir no decorrer da apresentação. No QUADRO 10 são apresentados o perfil de conhecimentos de cada agente aluno e seu nível de conhecimento.

QUADRO 10 – PERFIS DE CONHECIMENTOS EXPERIMENTO 2

Agente	Perfil de Conhecimentos	Nível
ALUNO 1	C++ - SQLITE	3,4
ALUNO 2	DELPHI - SQL SERVER – INGLES	4,0
ALUNO 3	PHP – MYSQL – INGLES	3,4
ALUNO 4	PYTHON - JAVA SCRIPT – ORACLE	4,5
ALUNO 5	OBJECTIVE C - AJAX - SQL SERVER	3,2

FONTE: O autor (2016).

O QUADRO 11 apresenta o perfil de conhecimentos solicitados pelos agentes empresa.

QUADRO 11 – PERFIS SOLICITADOS EXPERIMENTO 2

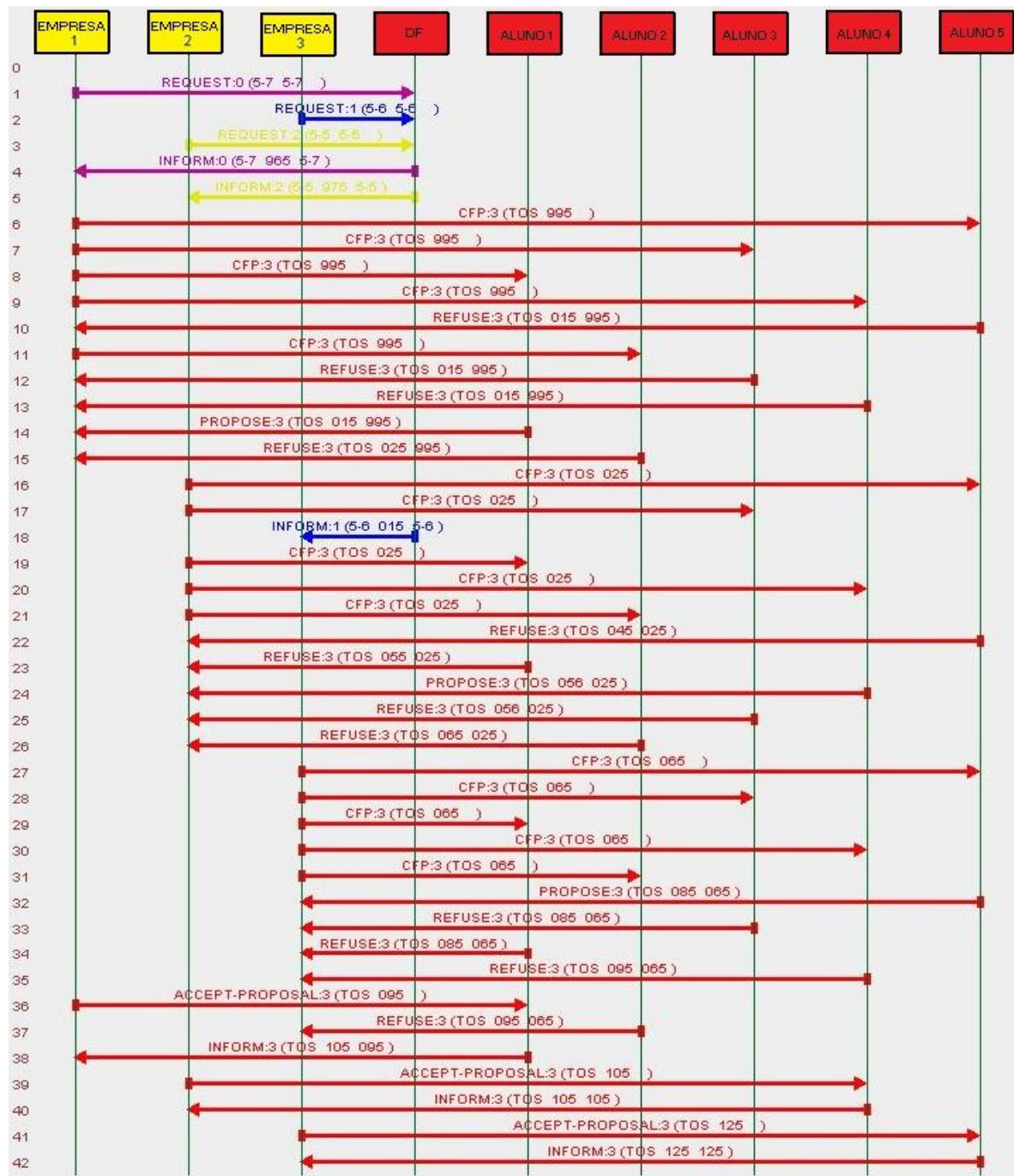
Agente	Perfil Conhecimentos
EMPRESA 1	C++ - SQLITE
EMPRESA 2	PYTHON - JAVA SCRIPT – ORACLE
EMPRESA 3	OBJECTIVE C - AJAX - SQL SERVER

FONTE: O autor (2016).

6.4.1 Resultados do *Jade Agent Sniffer* Experimento 2

A FIGURA 48 ilustra todas as interações entre os agentes aluno e agentes empresa monitoradas pelo agente *Sniffer*. As linhas são representadas por números e nelas o tipo de interação que ocorreu entre os agentes.

FIGURA 48 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 2



FONTE: O Autor (2016)

6.4.2 Análise dos Resultados Experimento 2

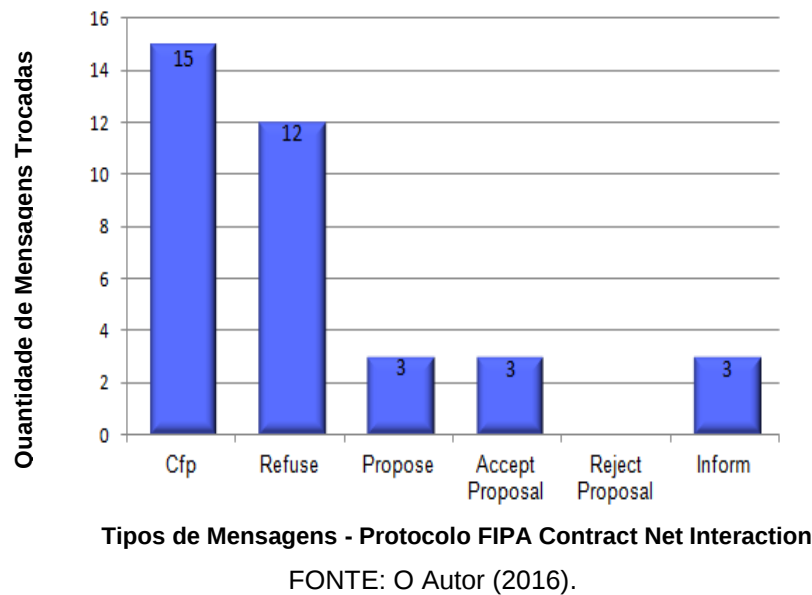
A TABELA 2 apresenta os dados quantitativos das interações entre os agentes apresentadas na FIGURA 48. Foram enviadas 15 solicitações pelos agentes empresa (*cfp*), das quais 12 foram recusadas (*refuse*) e 3 aceitas (*accept*) pelos agentes aluno, para verificar quais agentes aceitaram a proposta é só observar por exemplo o agente aluno 1 enviou uma proposta (*propose*) ao agente empresa 1 captada pelo agente *Sniffer* na linha 14. Como as três empresas aceitaram a proposta e comunicaram os agentes aluno com uma mensagem do tipo proposta aceita (*accept proposal*), 3 mensagens do tipo *inform* foram enviadas pelos agentes alunos.

TABELA 2 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 2

Agentes Empresa	<i>Cfp</i>	<i>Refuse</i>	<i>Propose</i>	<i>Accept Proposal</i>	<i>Reject Proposal</i>	<i>Inform</i>	Agentes Aluno
EMPRESA 1	8	0	14	36	0	38	ALUNO 1
	11	15	0	0	0	0	ALUNO 2
	7	12	0	0	0	0	ALUNO 3
	9	13	0	0	0	0	ALUNO 4
	6	10	0	0	0	0	ALUNO 5
EMPRESA 2	19	23	0	0	0	0	ALUNO 1
	21	26	0	0	0	0	ALUNO 2
	17	25	0	0	0	0	ALUNO 3
	20	0	24	39	0	40	ALUNO 4
	16	22	0	0	0	0	ALUNO 5
EMPRESA 3	29	34	0	0	0	0	ALUNO 1
	31	37	0	0	0	0	ALUNO 2
	28	33	0	0	0	0	ALUNO 3
	30	35	0	0	0	0	ALUNO 4
	27	0	32	41	0	42	ALUNO 5
	15	12	3	3	0	3	

FONTE: O Autor (2016).

GRÁFICO 2 ilustra as quantidades apresentadas na TABELA 2. Um total de 15 propostas foram enviadas pelos agentes empresa, 12 foram recusadas e 3 tiveram uma proposta como retorno. As 3 propostas enviadas foram aceitas pelos agentes empresa.

GRÁFICO 2 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 2

6.5 EXPERIMENTO 3 AGENTE ALUNO E AGENTE EMPRESA

No experimento 3 é apresentado um cenário onde somente dois perfis de conhecimento solicitados foram encontrados, um perfil fica pendente aguardando até um aluno com as características solicitadas aparecer.

No QUADRO 12 são apresentados o perfil de conhecimentos de cada agente aluno e seu nível de conhecimento.

QUADRO 12 – PERFIS DE CONHECIMENTOS EXPERIMENTO 3

Agente	Perfil de Conhecimentos	Nível
ALUNO 1	C++ - SQLITE	3,2
ALUNO 2	JAVA ORACLE	4,5
ALUNO 3	PYTHON - JAVA SCRIPT – ORACLE	4,0
ALUNO 4	PHP – MYSQL – INGLES	3,4
ALUNO 5	OBJECTIVE C - AJAX - SQL SERVER	3,4

FONTE: O autor (2016).

Os agentes empresa e o perfil de conhecimentos solicitados são apresentados no QUADRO 13.

QUADRO 13 – PERFIS SOLICITADOS EXPERIMENTO 3

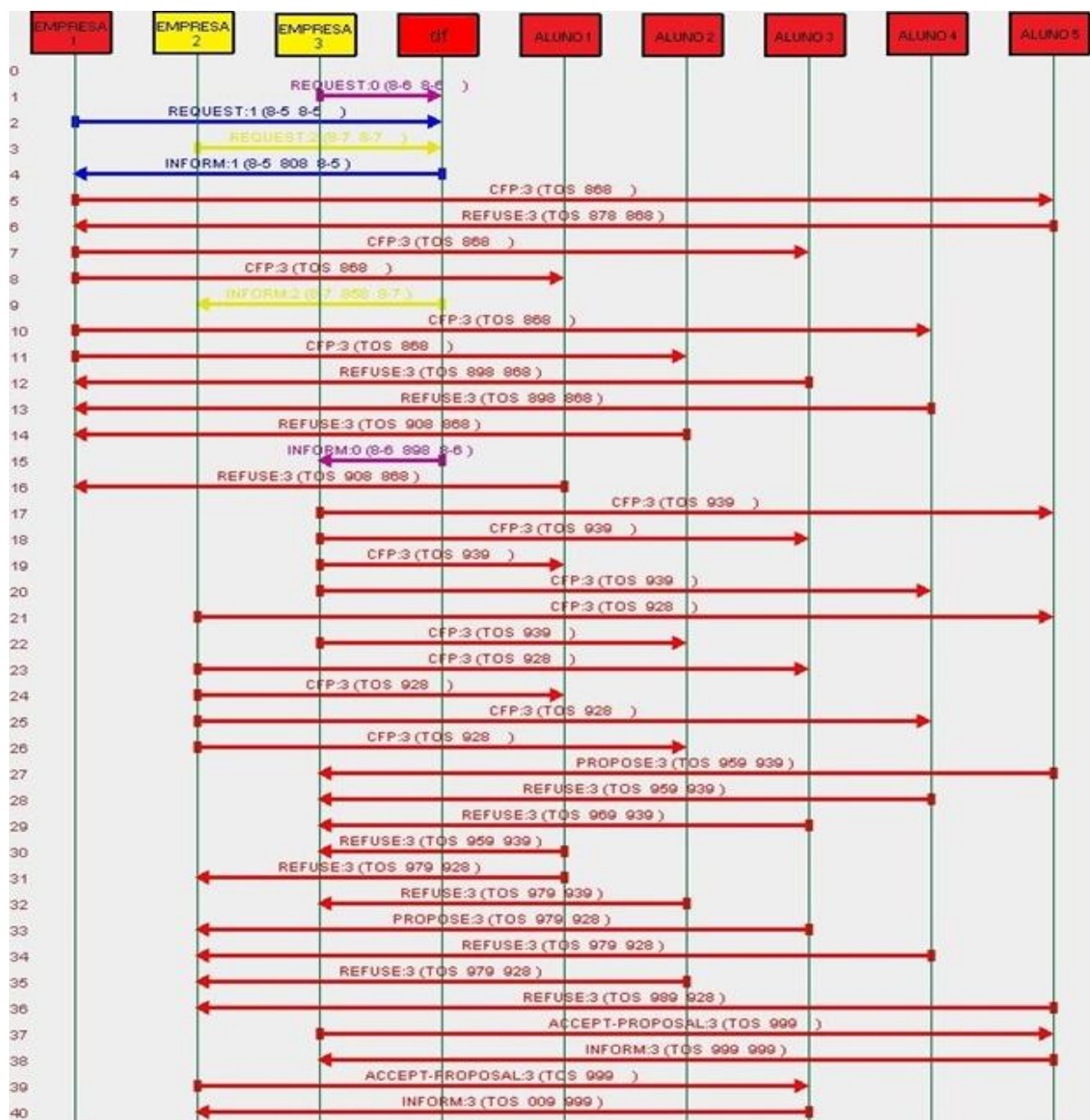
Agente	Perfil Conhecimentos
EMPRESA 1	RUBY ORACLE
EMPRESA 2	PYTHON - JAVA SCRIPT – ORACLE
EMPRESA 3	OBJECTIVE C - AJAX - SQL SERVER

FONTE: O autor (2016)

6.5.1 Resultados do *Jade Agente Sniffer* Experimento 3

A FIGURA 49 ilustra todas as interações entre os agentes aluno e empresa monitoradas pelo agente *Sniffer*. As linhas são representadas por números e nelas o tipo de interação que ocorreu entre os agentes. Cada agente ativo na plataforma é representado como um retângulo, dos 3 agentes empresa os 2 que tiveram sua solicitação encontrada estão representados na cor amarelo, são eles: empresa 2 e 3. A empresa 1 não teve sua solicitação encontrada por isso permaneceu com a cor em vermelho.

FIGURA 49 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 3



FONTE: O Autor (2016).

6.5.2 Análise dos Resultados Experimento 3

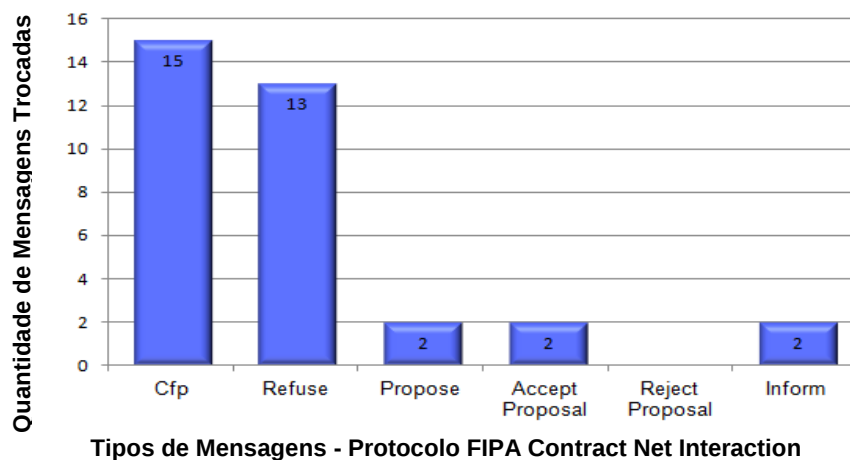
A TABELA 3 apresenta os dados estatísticos das interações entre os agentes apresentadas na FIGURA 49. Um total de 15 solicitações foram enviadas pelos agentes empresa, das quais 13 foram recusadas e 2 aceitas. Os agentes alunos que aceitaram a solicitação enviaram uma proposta para os agentes empresa que aceitaram a proposta com uma mensagem do tipo proposta aceita (*accept proposal*), duas mensagens do tipo (*inform*) foram retornadas aos agentes empresa que aceitaram a proposta. Ainda de acordo com a TABELA 3 o agente empresa 1 não recebeu nenhuma proposta e 5 recusas, foi o único agente que não teve a solicitação encontrada.

TABELA 3 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 3

Agentes Empresa	<i>Cfp</i>	<i>Refuse</i>	<i>Propose</i>	<i>Accept Proposal</i>	<i>Reject Proposal</i>	<i>Inform</i>	Agentes Aluno
EMPRESA 1	8	16	0	0	0	0	ALUNO 1
	11	14	0	0	0	0	ALUNO 2
	7	12	0	0	0	0	ALUNO 3
	10	13	0	0	0	0	ALUNO 4
	5	6	0	0	0	0	ALUNO 5
EMPRESA 2	24	31	0	0	0	0	ALUNO 1
	26	35	0	0	0	0	ALUNO 2
	23	0	33	39	0	40	ALUNO 3
	25	34	0	0	0	0	ALUNO 4
	21	36	0	0	0	0	ALUNO 5
EMPRESA 3	10	30	0	0	0	0	ALUNO 1
	22	37	0	0	0	0	ALUNO 2
	18	29	0	0	0	0	ALUNO 3
	20	28	0	0	0	0	ALUNO 4
	17	0	27	37	0	38	ALUNO 5
	15	13	2	2	0	2	

FONTE: O Autor (2016).

O GRÁFICO 3 ilustra os dados da TABELA 3. Um total de 15 propostas foram enviadas pelos agentes empresa, 13 foram recusadas e 2 tiveram proposta como retorno. Todas propostas enviadas foram aceitas pelos agentes empresa.

GRÁFICO 3 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 3

FONTE: O Autor (2016).

6.6 EXPERIMENTO 4 AGENTE ALUNO E AGENTE EMPRESA

No experimento 4 é apresentado um cenário onde o de perfil de conhecimentos solicitado é encontrado em mais de um agente aluno. Neste cenário o agente que tiver o maior nível de conhecimentos será o escolhido pela empresa. No QUADRO 14 são apresentados o perfil de conhecimentos de cada agente aluno e seu nível de conhecimento.

QUADRO 14 – PERFIS DE CONHECIMENTOS EXPERIMENTO 4

Agente	Perfil de Conhecimentos	Nível
ALUNO 1	PYTHON - JAVA SCRIPT – ORACLE	3,2
ALUNO 2	PYTHON - JAVA SCRIPT – ORACLE	4,5
ALUNO 3	PYTHON - JAVA SCRIPT – ORACLE	4,0
ALUNO 4	PHP – MYSQL – INGLES	3,4
ALUNO 5	JAVA – ORACLE	3,5

FONTE: O autor (2016).

Os agentes empresa e o perfil de conhecimentos solicitados são apresentados no QUADRO 15.

QUADRO 15 – PERFIS SOLICITADOS EXPERIMENTO 4

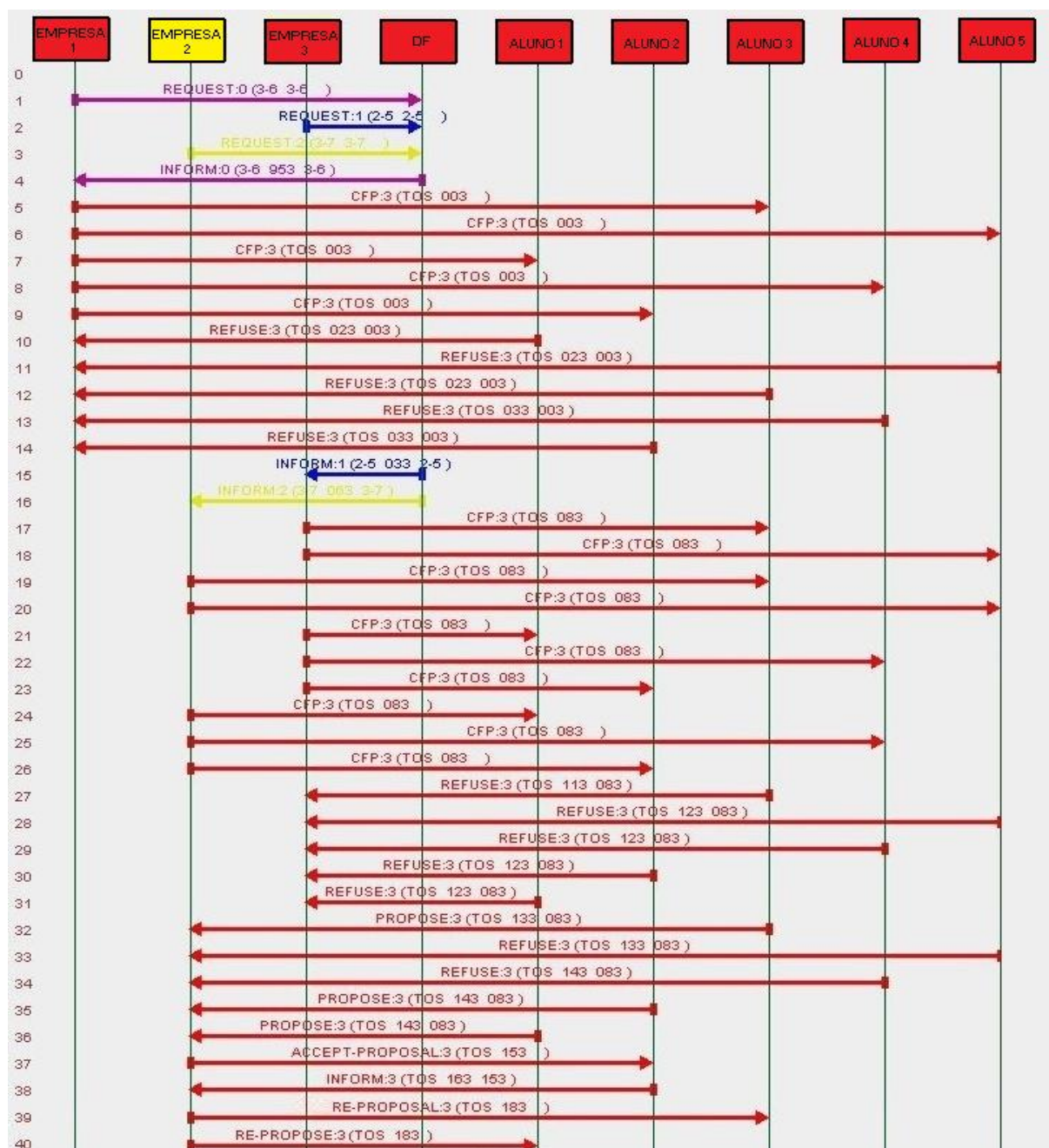
Agente	Perfil Conhecimentos
EMPRESA 1	RUBY – ORACLE
EMPRESA 2	PYTHON – JAVA SCRIPT – ORACLE
EMPRESA 3	OBJECTIVE C - AJAX - SQL SERVER

FONTE: O autor (2016).

6.6.1 Resultados do *Jade Agent Sniffer* Experimento 4

A FIGURA 50 ilustra todas as interações entre os agentes aluno e agentes empresa monitoradas pelo agente *Sniffer*. As linhas são representadas por números e nelas o tipo de interação que ocorreu entre os agentes. Cada agente ativo na plataforma de agentes é representado como um retângulo. Dos 3 agentes empresa 1 teve sua solicitação encontrada e esta representado na cor amarelo.

FIGURA 50 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 4



FONTE: O Autor (2016).

6.6.2 Análise dos Resultados Experimento 4

A TABELA 4 apresenta os dados estatísticos das interações entre os agentes apresentadas na FIGURA 50. Foram enviadas 15 solicitações pelos agentes empresa das quais 12 foram recusadas (*refused*) e 3 aceitas (*accept proposal*), as propostas aceitas foram da empresa 2, os agentes alunos que aceitaram a solicitação enviaram uma proposta. O agente empresa recebeu as 3 propostas e aceitou a proposta com maior nível de conhecimento com uma mensagem do tipo (*accept proposal*), as propostas rejeitadas foram enviadas com mensagens do tipo (*reject proposal*). Apenas uma mensagem do tipo *inform* foi enviada pelo agente aluno que teve a proposta aceita.

TABELA 4 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 4

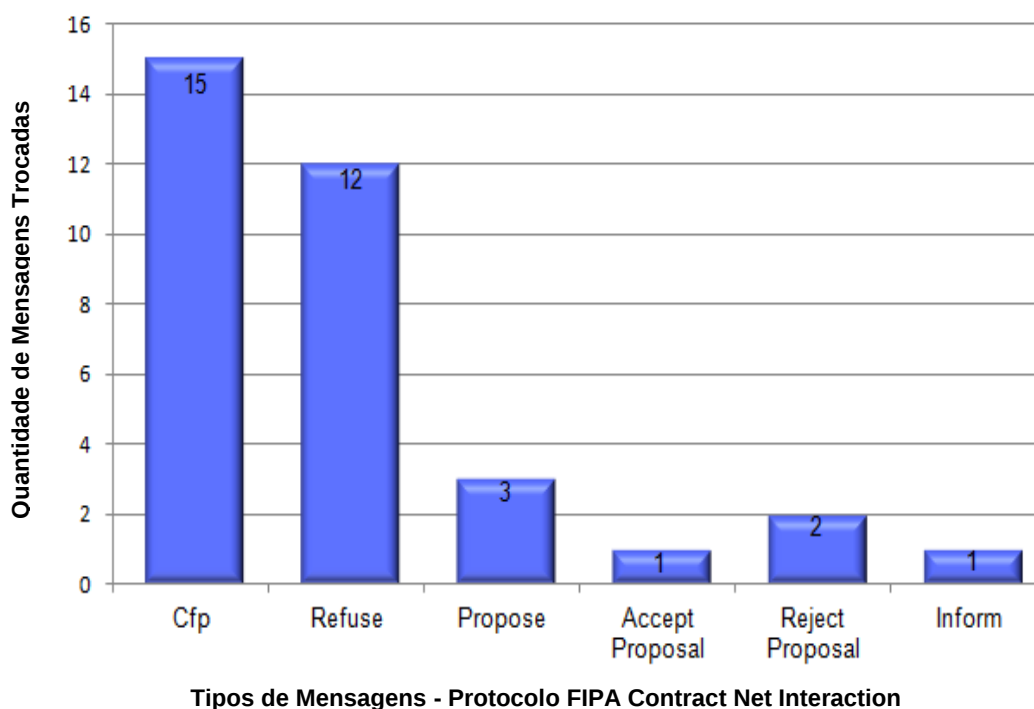
Agentes Empresa	Cfp	Refuse	Propose	Accept Proposal	Reject Proposal	Inform	Agentes Aluno
EMPRESA 1	7	10	0	0	0	0	ALUNO 1
	9	14	0	0	0	0	ALUNO 2
	5	12	0	0	0	0	ALUNO 3
	8	13	0	0	0	0	ALUNO 4
	6	11	0	0	0	0	ALUNO 5
EMPRESA 2	24	0	36	0	40	0	ALUNO 1
	26	0	35	27	0	38	ALUNO 2
	19	0	32	0	39	0	ALUNO 3
	25	34	0	0	0	0	ALUNO 4
	20	33	0	0	0	0	ALUNO 5
EMPRESA 3	21	31	0	0	0	0	ALUNO 1
	23	30	0	0	0	0	ALUNO 2
	17	27	0	0	0	0	ALUNO 3
	22	29	0	0	0	0	ALUNO 4
	18	28	0	0	0	0	ALUNO 5
	15	12	3	1	2	1	

FONTE: O autor (2016).

O GRÁFICO 4 ilustra as quantidades apresentadas na TABELA 4 para melhor compreensão. Um total de 15 propostas foram enviadas pelos agentes empresa, 12 foram recusadas e 3 tiveram uma proposta como retorno. As propostas

recebidas a escolhida foi a de maior nível de conhecimento cujo agente recebeu uma mensagem do tipo *accept proposal* e enviou uma mensagem do tipo *inform*. As 2 propostas com nível de conhecimento menor receberam uma mensagem do tipo *reject proposal*.

GRÁFICO 4 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 4



FONTE: O autor (2016).

6.7 EXPERIMENTO 5 AGENTE ALUNO E AGENTE EMPRESA

O quinto experimento, foi realizado com dados reais de uma turma do terceiro ano do curso de “Sistemas de Informação” de uma instituição de ensino superior, situada no oeste do Paraná a qual será denominada de “Instituição A”, por ser um curso de Bacharel e o mercado de trabalho oferecer oportunidade de colocação, tanto para estágio quanto para vaga permanente foi o curso escolhido. A turma é composta por 24 alunos que estudam no período noturno podendo trabalhar no período da manhã e da tarde, cada um forneceu detalhes de seus conhecimentos para compor o Perfil.

A FIGURA 51 ilustra uma parte do diário de classe com os alunos que fazem parte de turma do terceiro ano.

FIGURA 51 – DIÁRIO DE CLASSE

Matrícula	Nome do Aluno
1400282	ANDREA S.
1400354	BRUNA C.
1400643	DIEGO C.
1401020	DIEGO S.
1401094	DOUGLAS K.
1401414	EDMO A.
1401456	EDSON G.
1401464	EVERTON G.
1401472	FABIANA A.
1401480	FABIO A.
1401497	GIANCARLO G.
1401536	IGOR C.
1401544	JONATHAN S.
1401841	JORGE R.
1401858	LEANDRO A.
1401911	LUIS F.
1402008	LUIS M.
1402168	MARCELO K.
1402186	MARCIO T.
1402298	MARCOS B.
1402354	MICHEL R.
1402555	PEDRO G.
1402869	RENAN G.
1402885	WELINGTON C.

FONTE: Adaptado de: Instituição A, (2016)

O QUADRO 16 apresenta os níveis de conhecimento e a média dos conhecimentos de cada aluno, os dados foram informados pelos próprios alunos.

QUADRO 16 – PERFIS DE CONHECIMENTOS EXPERIMENTO 5

(continua)

Agente	Perfil de Conhecimentos	Nível
ANDREA S.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	3.4
BRUNA C.	DELPHI FIREBIRD INGLES	4.0
DIEGO C.	DELPHI FIREBIRD INGLES	3.4
DIEGO S.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	4.5
DOUGLAS K.	DELPHI FIREBIRD SQL INGLES	3.2
EDMO A.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	3
EDSON G.	DELPHI FIREBIRD SQL INGLES	3.5
EVERTON G.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	3.8
FABIANA A.	DELPHI FIREBIRD SQL INGLES	3.1

QUADRO 17 – PERFIS DE CONHECIMENTOS EXPERIMENTO 5

(conclusão)

Agente	Perfil de Conhecimentos	Nível
FABIO A.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	3
GIANCARLO G.	DELPHI FIREBIRD SQL INGLES	3.5
IGOR C.	DELPHI FIREBIRD SQL INGLES PHP HTML MYSQL	3.2
JONATHAN S.	DELPHI FIREBIRD SQL INGLES	3.4
JORGE R.	DELPHI FIREBIRD SQL INGLES HTML	3.7
LEANDRO A.	DELPHI FIREBIRD SQL INGLES	3
LUIS F.	DELPHI FIREBIRD SQL INGLES	3.5
LUIS M.	DELPHI FIREBIRD SQL INGLES	3
MARCELO K.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES PHP HTML	3.8
MARCIO T.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES PHP HTML	3.5
MARCOS B.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES PHP HTML	3.5
MICHEL R.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES PHP HTML	3.7
PEDRO G.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES	3.5
RENAN G.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES	3
WELLINGTON C.	DELPHI FIREBIRD JAVA MYSQL SQL INGLES	3,4

FONTE: O autor (2016).

Os perfis de conhecimentos solicitados são apresentados no QUADRO 17.

QUADRO 18 – PERFIS SOLICITADOS EXPERIMENTO 5

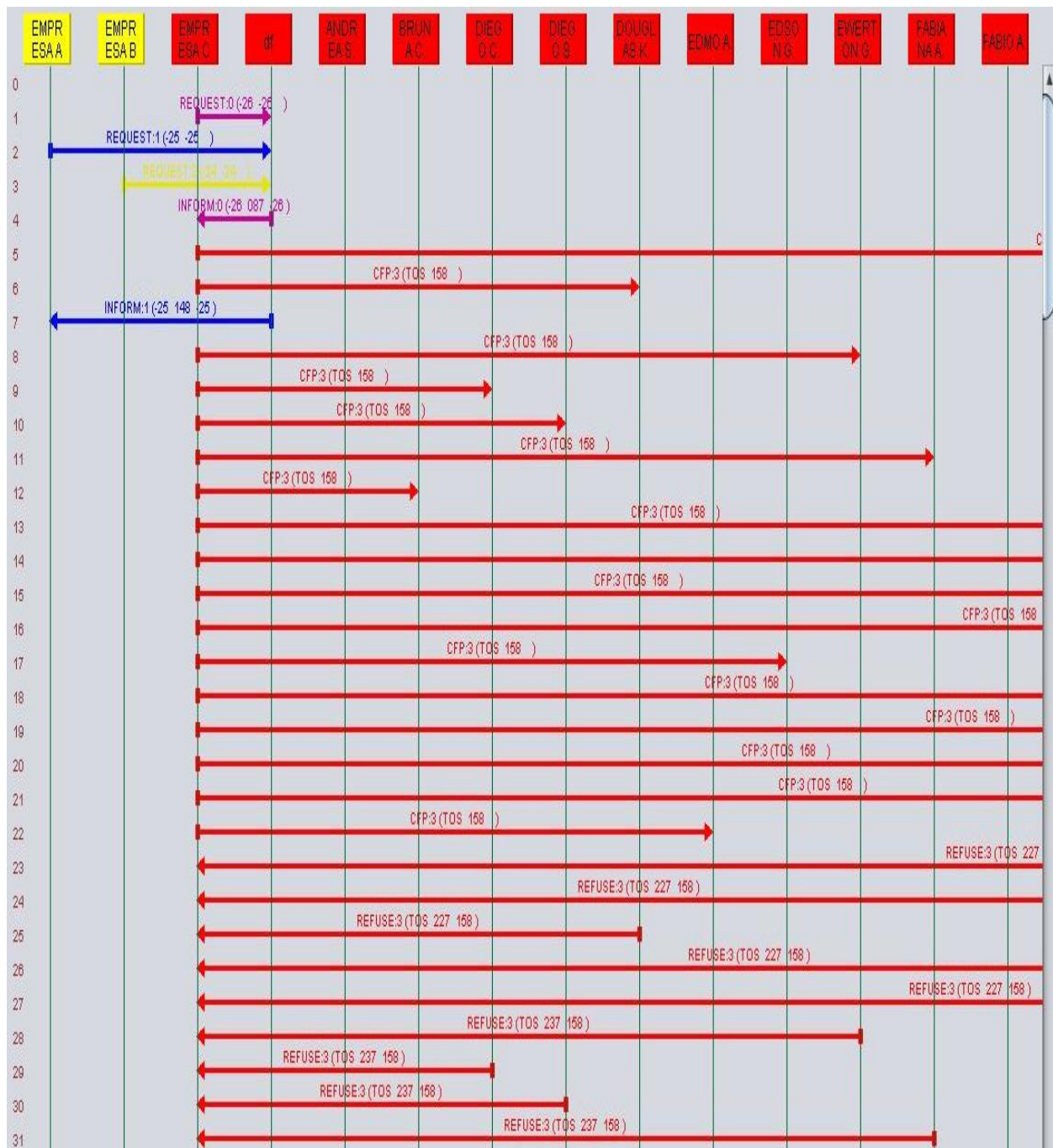
Agente	Perfil Conhecimentos
EMPRESA A	DELPHI FIREBIRD JAVA MYSQL SQL INGLES PHP HTML
EMPRESA B	DELPHI FIREBIRD JAVA MYSQL SQL INGLES
EMPRESA C	OBJECTIVE C - AJAX - SQL SERVER

FONTE: O autor (2016).

6.7.1 Resultados do *Jade Agente Sniffer* Experimento 5

A FIGURA 52 apresenta apenas uma parte das informações, uma vez que o experimento foi aplicado com todos os matriculados da turma, proporciona uma representação das interações entre agentes empresa e agentes aluno. As linhas são representadas por números e nelas o tipo de interação que ocorreu entre os agentes, as duas empresas que aparecem com a cor amarelo a Empresa A e B encontraram alunos com perfil de conhecimento desejado.

FIGURA 52 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 5



FONTE: O Autor (2016).

6.7.2 Análise dos Resultados Experimento 5

A TABELA 5 apresenta somente os resultados finais, cada coluna representa o total de tipos de mensagens resultantes da interação entre os dois agentes aluno e empresa. Foram 360 solicitações enviadas pelos agentes empresa, das quais 353 foram recusadas (*refuse*). A empresa A recebeu 4 propostas

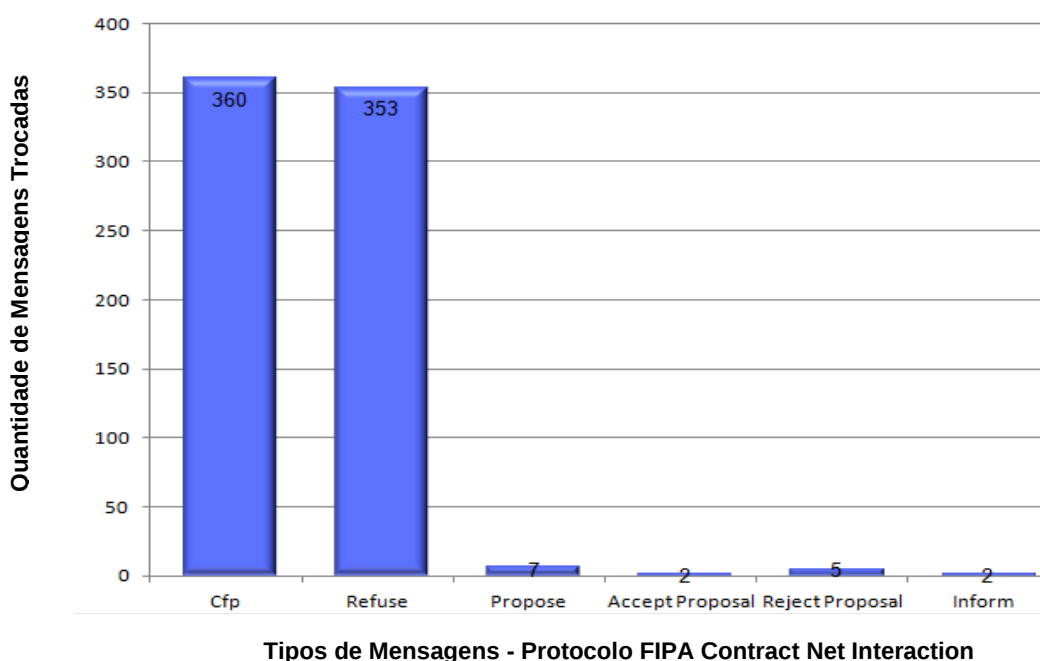
selecionou a melhor e rejeitou as demais, a empresa B recebeu 3 propostas selecionou a melhor e rejeitou as demais, já a Empresa C não recebeu nenhuma proposta. Os agentes alunos que tiveram a proposta aceita enviaram 2 mensagens do tipo (*inform*). O GRÁFICO 5 ilustra os dados da TABELA 5.

TABELA 5 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 5

Agente Empresa	Cfp	Refuse	Propose	Accept Proposal	Reject Proposal	Inform
EMPRESA A	120	116	4	1	3	1
EMPRESA B	120	117	3	1	2	1
EMPRESA C	120	120	0	0	0	0
	360	353	7	2	5	2

FONTE: O Autor (2016).

GRÁFICO 5 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 5



FONTE: O autor (2016).

6.8 EXPERIMENTO 6 AGENTE REGISTRA MÉDIAS E BUSCA MÉDIAS

No QUADRO 18 são apresentados os dados dos alunos do experimento 6 a disciplina e média de cada um.

QUADRO 19 – MÉDIAS DE ALUNOS EXPERIMENTO 6

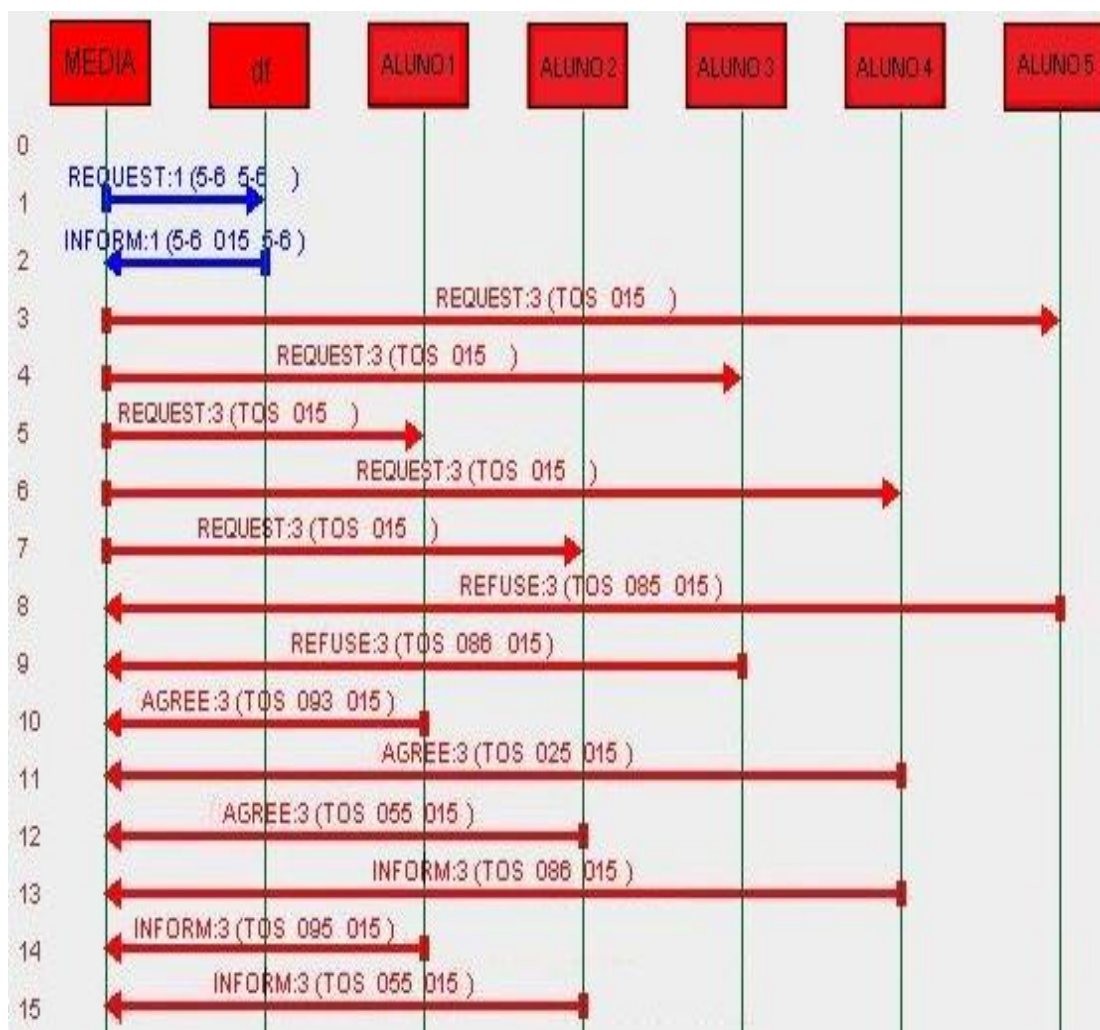
Agente	Disciplina	Média
ALUNO 1	DISCIPLINA A	3.2
ALUNO 2	DISCIPLINA A	4.5
ALUNO 3	DISCIPLINA A	8
ALUNO 4	DISCIPLINA A	3.4
ALUNO 5	DISCIPLINA A	9

FONTE: O autor (2016).

6.8.1 Resultados do Jade Agente *Sniffer* Experimento 6

A FIGURA 53 ilustra todas as interações entre o agente busca médias e os agentes alunos. As linhas são representadas por números e em cada uma delas o que o agente está fazendo naquele momento.

FIGURA 53 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 6



FONTE: O Autor (2016).

6.8.2 Análise dos Resultados Experimento 6

A TABELA 6 apresenta os dados estatísticos das interações entre os agentes apresentados na FIGURA 53, o agente registra médias enviou 5 mensagem do tipo (*request*) para os agentes aluno, recebeu 2 mensagens do tipo (*refuse*) do Aluno 3 e Aluno 5. Três mensagens do tipo (*agree*) e 3 do tipo (*inform*) foram enviadas por agentes com médias abaixo de 7.0 são eles: Aluno 1, 2 e 4.

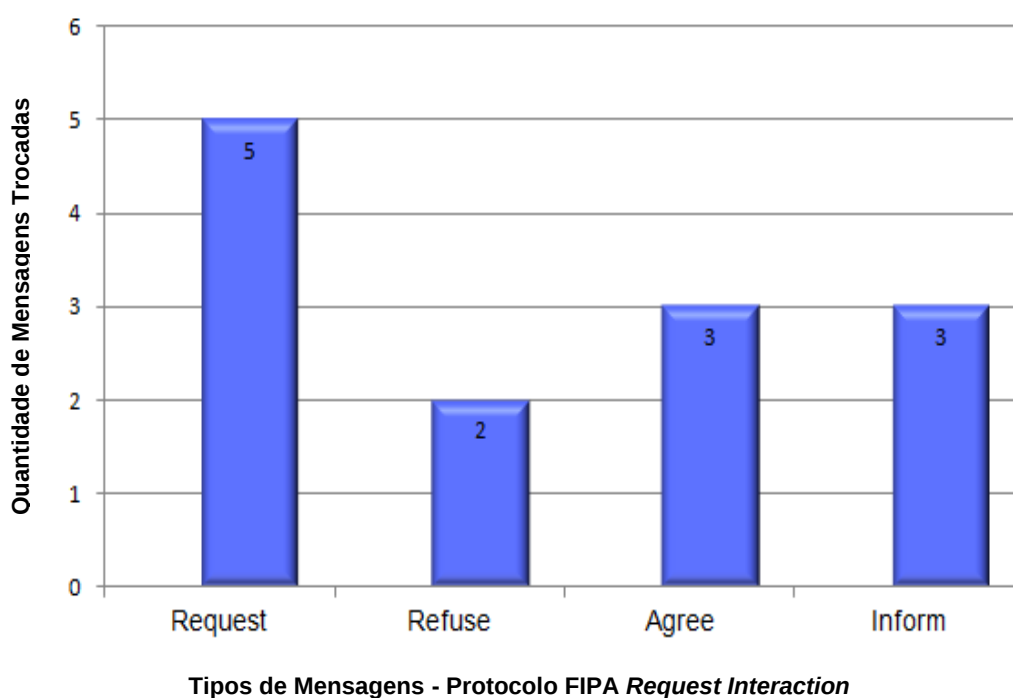
TABELA 6 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 6

Agentes Registra Médias	<i>Request</i>	<i>Refuse</i>	<i>Agree</i>	<i>Inform</i>	<i>Aluno</i>
REGISTRA	5	0	10	14	ALUNO 1
REGISTRA	7	0	12	15	ALUNO 2
REGISTRA	4	9	0	0	ALUNO 3
REGISTRA	6	0	11	13	ALUNO 4
REGISTRA	3	8	0	0	ALUNO 5
	5	2	3	3	

FONTE: O Autor (2016).

O GRÁFICO 6 apresenta os dados da TABELA 6.

GRÁFICO 6 – RESULTADOS DO AGENTE *SNIFFER* EXPERIMENTO 6



FONTE: O autor (2016).

6.9 EXPERIMENTO 7 AGENTES REGISTRA MÉDIAS E BUSCA MÉDIA

O sétimo experimento dos agentes registra médias e busca médias, foi aplicado utilizando dados de uma turma do terceiro ano do curso de “Sistemas de Informação” em uma instituição situada no Paraná a qual denominamos de Instituição A. A turma é composta por 24 alunos conforme ilustra a FIGURA 54. Sua matriz curricular possui 7 disciplinas, para cada disciplina foram aplicadas duas avaliações, com os dados encontrados pelos agentes é possível detectar alguns problemas que serão apresentados ao final do experimento.

FIGURA 54 – DIÁRIO DE AULA CURSO DE SISTEMAS DE INFORMAÇÃO

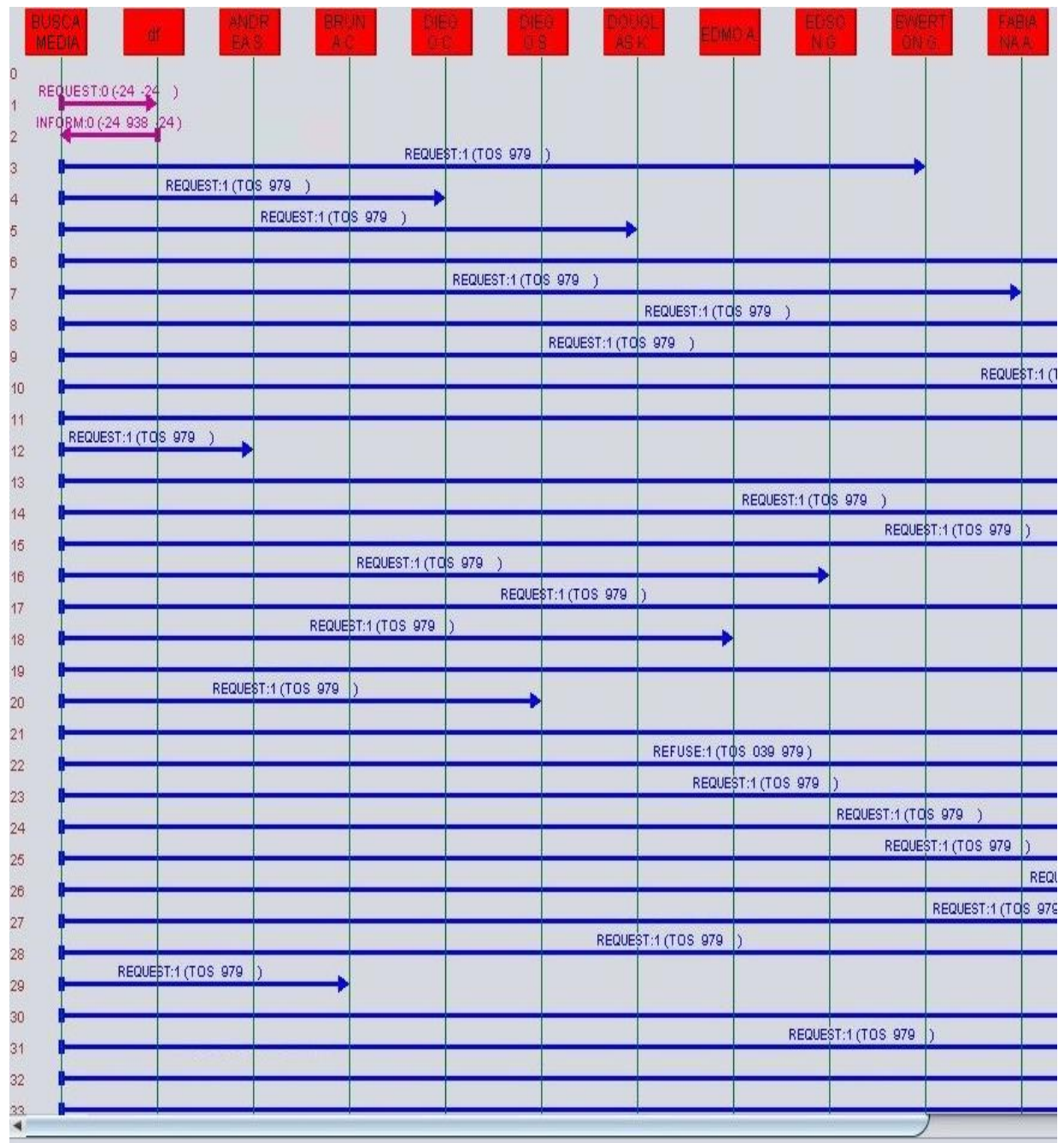
DIÁRIO DE CLASSE RESUMO ANUAL 2016			Curso							Turma	Aulas Prev.	Aulas Dadas	Lim. Faltas	Visto Professor		Visto Coordenador	
			Sistemas de Informação							3 Ano	80 H	20					
			Disciplina							Professor							
Banco de Dados II																	
Matrícula	Nome do Aluno	Num	FALTAS							NOTAS							
			Fev/Mar	Abr/Ma	Jun/Jul	Ago/Set	Out/Nov	Dez	Total	1º	2º	3º	4º	M Anual	Exame	M.Final	RESUL
1400282	ANDREA S.	1	3	6	0	0	0	0	9	4,5	6,5			5,5			
1400354	BRUNA C.	2	2	0	0	0	0	0	2	7,0	7,0			7,0			
1400643	DIEGO C.	3	2	2	0	0	0	0	4	6,0	8,0			7,0			
1401020	DIEGO S.	4	1	2	0	0	0	0	3	8,5	7,0			7,7			
1401094	DOUGLAS K.	5	4	6	0	0	0	0	10	5,0	6,0			5,5			
1401414	ÉDMO A.	6	5	3	0	0	0	0	8	4,5	5,5			5,0			
1401456	EDSON G	7	1	2	0	0	0	0	3	9,0	9,0			9,0			
1401464	EVERTON G.	8	3	2	0	0	0	0	5	8,5	8,0			8,2			
1401472	FABIANA A.	9	4	2	0	0	0	0	6	7,0	8,0			7,5			
1401480	FABIO A.	10	4	4	0	0	0	0	8	5,0	6,0			5,5			
1401497	GIANCARLO G.	11	4	4	0	0	0	0	8	8,0	8,0			8,0			
1401536	IGOR C.	12	9	0	0	0	0	0	9	4,5	6,5			5,5			
1401544	JONATHAN S.	13	5	0	0	0	0	0	5	8,0	7,0			7,5			
1401841	JORGE R.	14	7	1	0	0	0	0	8	5,0	4,0			4,5			
1401858	LEANDRO A.	15	2	0	0	0	0	0	2	4,5	4,5			4,5			
1401911	LUIS F.	16	0	0	0	0	0	0	0	8,0	7,0			7,5			
1402008	LUIS M.	17	0	3	0	0	0	0	3	7,0	9,0			8,0			
1402168	MARCELO K.	18	0	0	0	0	0	0	0	8,5	7,5			8,0			
1402186	MARCIO T.	19	5	7	0	0	0	0	12	3,0	4,5			3,7			
1402298	MARCOS B.	20	2	0	0	0	0	0	2	8,0	8,0			8,0			
1402354	MICHEL R.	21	0	1	0	0	0	0	1	8,5	8,5			8,5			
1402555	PEDRO G.	22	0	0	0	0	0	0	0	7,0	6,0			6,5			
1402869	RENAN G	23	1	3	0	0	0	0	4	5,0	3,0			8,0			
1402885	WELINGTON C.	24	4	4	0	0	0	0	8	5,5	7,0			6,2			

FONTE: Faculdade A (2016).

6.9.1 Resultados do *Jade Agente Sniffer* Experimento 7

A FIGURA 55 ilustra todas as interações entre o agente busca média e os agentes alunos. As linhas são representadas por números e em cada uma delas qual ação o agente está fazendo naquele momento.

FIGURA 55 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 7



FONTE: O Autor (2016).

6.9.2 Análise dos Resultados Experimento 7

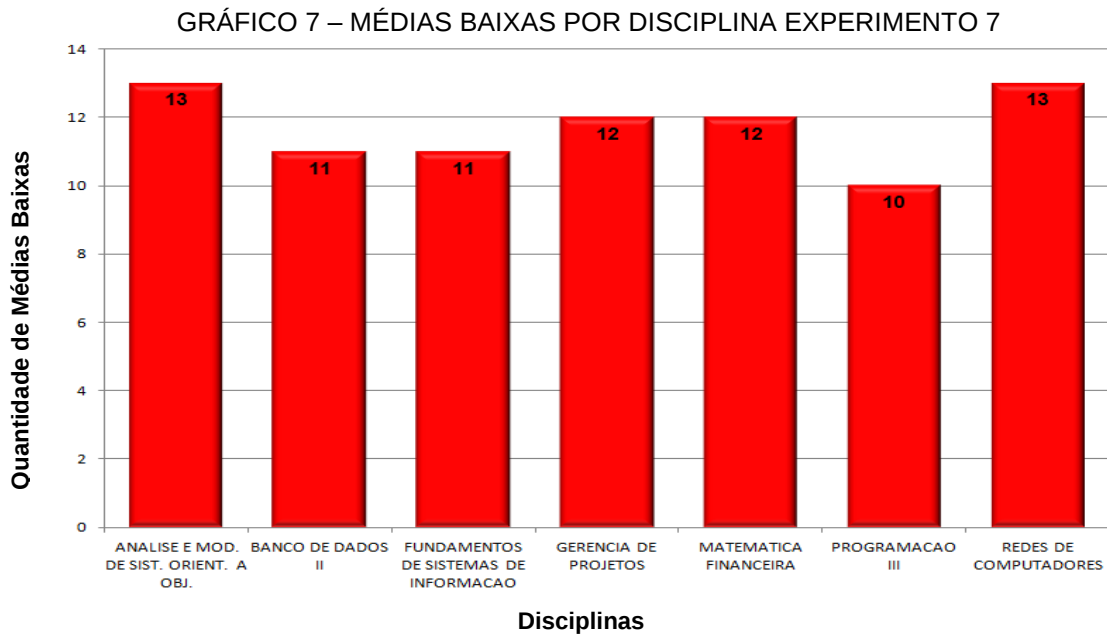
A TABELA 7 apresenta parte dos dados estatísticos reais das interações entre os agentes apresentadas na FIGURA 55, o agente registro médias filtrou todos os alunos com médias abaixo de 7.

TABELA 7 – RESULTADOS DO AGENTE SNIFFER EXPERIMENTO 7

ALUNO	DISCIPLINA	MÉDIA	FALTAS
ANDREA S.	BANCO DE DADOS II	5.5	9
ANDREA S.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	5	7
ANDREA S.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	4.2	0
BRUNA C.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	6.5	1
BRUNA C.	GERENCIA DE PROJETOS	6	0
BRUNA C.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	5	4
DIEGO S.	REDES DE COMPUTADORES	6.5	5
DIEGO S.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	5	0
DIEGO S.	GERENCIA DE PROJETOS	5	5
DOUGLAS K.	GERENCIA DE PROJETOS	6.5	2
DOUGLAS K.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	6	6
DOUGLAS K.	MATEMATICA FINANCEIRA	6.5	3
DOUGLAS K.	BANCO DE DADOS II	5.5	10
DOUGLAS K.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	5.5	3
DOUGLAS K.	PROGRAMACAO III	6.5	8
EDMO A.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	6.5	5
EDMO A.	PROGRAMACAO III	5	8
EDMO A.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	4	4
EDMO A.	REDES DE COMPUTADORES	5.5	4
EDMO A.	BANCO DE DADOS II	5	8
EDSON G.	GERENCIA DE PROJETOS	4.5	2
EDSON G.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	5.5	6
EDSON G.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	6	6
EVERTON G.	FUNDAMENTOS DE SISTEMAS DE INFORMACAO	5	7
EVERTON G.	GERENCIA DE PROJETOS	4	0
EVERTON G.	ANALISE E MOD. DE SIST. ORIENT. A OBJ.	6.6	3

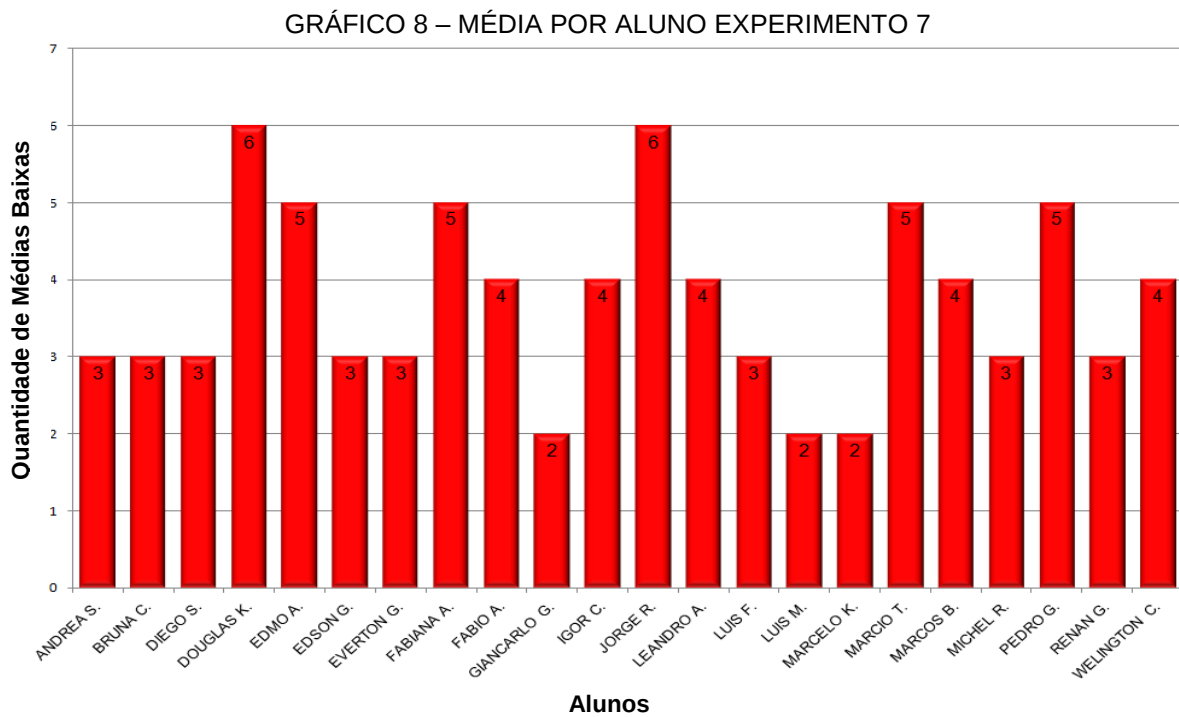
FONTE: O Autor (2016).

O GRÁFICO 7 apresenta a quantidade de alunos abaixo da média por disciplina.



FONTE: O autor (2016).

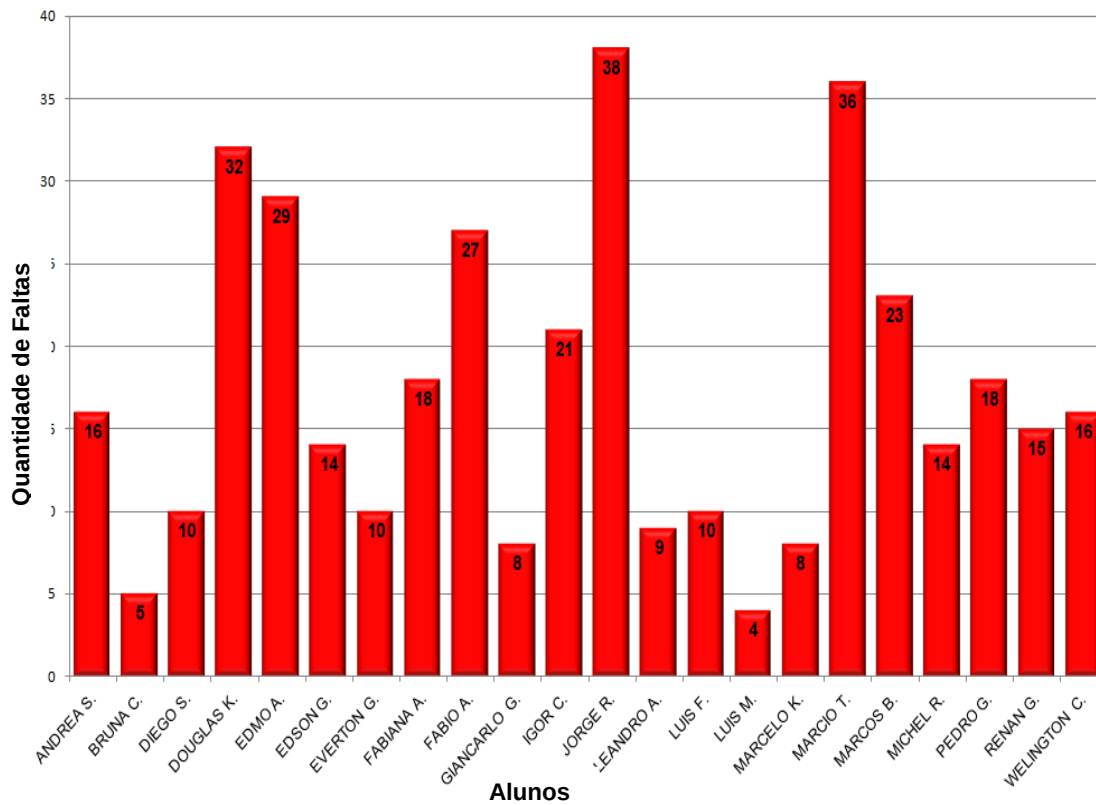
O GRÁFICO 8 apresenta os alunos e suas médias, ali pode-se verificar quais alunos precisam de mais atenção.



FONTE: O autor (2016).

No GRÁFICO 9 são apresentadas as faltas de todos os alunos, o que é muito útil para a coordenação de curso buscar os motivos que ocasionaram essas faltas.

GRÁFICO 9 – FALTAS POR ALUNO EXPERIMENTO 7



FONTE: O autor (2016).

7 CONCLUSÕES

7.1 CONCLUSÕES

Nesta dissertação foi apresentada uma abordagem que utiliza um sistema multiagentes nos padrões de protocolos de comunicação FIPA, desenvolvidos no Framework JADE. O sistema apresentou uma arquitetura composta por cinco agentes inteligentes integrada a um sistema de gerenciamento acadêmico, desenvolvido na linguagem de programação Java com um banco de dados MySQL, atuando com foco principal nas competências que os alunos possuem, apoiadas nos conhecimentos que o mercado de trabalho atualmente procura. Por sua vez, alunos que possuem perfil com maiores níveis de conhecimento, foram os recomendados no ambiente, para indicação a uma solicitação realizada por uma empresa em busca de conhecimentos específicos. Agentes de acompanhamento pedagógico também foram implementados, monitorando o ambiente e apresentando alunos com desempenho abaixo da média.

O processo de tomada de decisão dos agentes mediante os eventos que ocorreram no ambiente, como verificar se um aluno tem os conhecimentos desejados, avaliar entre os alunos selecionados qual tem o maior nível de conhecimentos e quais alunos têm problemas de desempenho pedagógico, foi realizado utilizando dois protocolos de comunicação. O protocolo FIPA *request interaction protocol* e o protocolo FIPA *contract net interaction protocol specification* e produziram os resultados esperados.

Este trabalho produziu duas contribuições principais. A primeira trata-se da implantação e validação do recurso de recomendação de estudantes com melhores perfis, baseados em seus conhecimentos realizado pelos agentes de forma autônoma, desenvolvidos utilizando a Metodologia para a Análise e Projeto de SMA utilizando Jade e seus diagramas AUML e UML adaptados, por serem estas as características dessa metodologia. A segunda diz respeito à parte da implementação que faz integração com um sistema de gerenciamento acadêmico desenvolvido, devido a falta deste tipo de sistema de uso comercial, buscando evidenciar as técnicas utilizadas.

Os experimentos simulados com dados fictícios foram utilizados para explorar a eficácia da comunicação entre agentes e validar as respostas dadas por

eles. Os experimentos realizados com dados reais também obteve resultados favoráveis nos dois tipos de experimentos. Além de validar os protocolos utilizados, foram constatados e identificados, pela análise dos resultados obtidos, elementos que podem aumentar a eficiência dos trabalhos das coordenações de curso.

Cabe ressaltar que a pesquisa, implementações e experimentos possibilitaram o domínio de várias ferramentas e representam uma contribuição para as comunidades científicas.

7.2 TRABALHOS FUTUROS

Para trabalhos futuros recomendam-se os seguintes temas:

- a) Aumentar a arquitetura proposta, incluindo novos agentes e aprimorar os agentes já existentes, objetivando o melhoramento do acompanhamento das atividades e implementando novas ações sobre o sistema;
- b) Incluir características de mobilidade dos agentes entre plataformas, acrescentando funcionalidades para dispositivos móveis;
- c) Implementar novos protocolos de comunicação entre os agentes;

REFERÊNCIAS

ALENCAR, M. A. S.; NETTO, J. F. M. Facilitando a tutoria EAD utilizando o SISACAD. In: CIAED - CONGRESSO INTERNACIONAL ABED DE EDUCAÇÃO A DISTÂNCIA, 3., 2013, Manaus. **Anais do CIAED**, Brasília, 2013. Disponível em: <www.abed.org.br/congresso2013/trabalhos/36.doc>. Acesso em: 22 set. 2015.

AKHTAR, N.; GHORI, A. S.; SALAMAT, N. Requirement analysis, architectural design and formal verification of a multi-agent based university information management system. **International Journal of Computer Science & Information Technology**. v. 6, n. 5, dez 2014. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1501/1501.01273.pdf>>. Acesso em: 25 set. 2015.

ALKHATEEB, F., et. al. Multi-agent-based system for securing university campus. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS, MODELLING AND SIMULATION, 2010, Liverpool. **Anais ICISMS**, Nova York: IEEE, 2010. p. 75–79.

BAUER, B.; MÜLLER, P.; ODELL, J.. Agent UML: a formalism for specifying multiagent interaction. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 2001, Toronto. **Anais ICSE** Disponível em:<<http://www.icse-conferences.org/2001/print/fp.pdf>>. Acesso em: 25 jun. 2015.

BAUER B.; ODELL J. UML 2.0 and agents: how to build agent-based systems with the new UML standard, **Journal of Engineering Applications of Artificial Intelligence**. Amsterdam: Elsevier, v.18, n. 2, mar. 2005.

BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. **Developing multi-agent systems with Jade**. Nova York: Wiley, 2007.

BENAVIDES, S. D. D. **Raciocínio de agentes musicais composição algorítmica, vida artificial e interatividade em sistemas multiagentes musicais**. 2012. 95 f. Dissertação (Mestrado em Ciências) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/45/45134/tde-17082013-154859/pt-br.php>>. Acesso em: 22 set. 2015.

BERRADA, M.; BOUNABAT, B. Business modeling of enterprise architecture based on multi-agent system. **International Journal of e-Education, e-Business, e-Management and e-Learning**. v. 3, n.6, dez 2013. Disponível em: <<http://www.ijeeee.org/Papers/281-A0054.pdf>>. Acesso em: 22 set. 2015.

BEZERRA, E. **Princípios de análise e projeto de sistemas com Uml**. 2. ed. Rio de Janeiro: Campus, 2007.

BREMGARTNER, V., NETTO, J. F. M. Relato de uma experiência de auxílio personalizado a alunos em ambiente virtual de aprendizagem utilizando agentes. In: WORKSHOP DE INFORMÁTICA NA EDUCAÇÃO, 18., 2012, Rio de Janeiro. **Anais do XVIII WIE**, Porto Alegre: Sociedade Brasileira de Computação – SBC, 2012.

BRENNER, W.; ZARNEKOW, R.; WITTIG, H.. **Intelligent software agents**. Nova York: Springer, 1998.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML guia do usuário**. 1. ed. Rio de Janeiro: Elsevier, 2000.

DANTAS, T. C. et. al. AprendEAD: Ambiente para educação à distância apoiado em agentes. . **Cadernos do IME Série Informática**. v. 23, jun. 2007. Disponível em: <<http://www.e-publicacoes.uerj.br/index.php/cadinf/article/view/6549/4667>>. Acesso em: 27 set. 2015.

CAZELLA, S. C. et al. Recomendação de objetos de aprendizagem empregando filtragem colaborativa e competências In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 20., 2009, Florianópolis. **Anais do XX Simpósio Brasileiro de Informática na Educação**, Porto Alegre: Sociedade Brasileira de Computação – SBC, 2009. p. 1147-1156.

CHAIB-DRAA, B.; DIGNUM F. Trends in agent communication language **Computational Intelligence**., New York: Wiley, v. 18, n.2, p. 89-101, mai. 2002.

ENDO, E. **Arquitetura de agentes racionais utilizando a tecnologia adaptativa**. 2010. 146 f. Dissertação (Mestrado em Tecnologia) – Programa de Mestrado em Tecnologia, Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2010. Disponível em: <http://lta.poli.usp.br/lta/publicacoes/teses-e-dissertacoes/2010/endo-2010-arquitetura-de-agentes-rationais-utilizando-a-tecnologia-adaptativa/at_download/file>. Acesso em: 20 ago. 2015.

FERREIRA, A. B. H. **Novo Dicionário da Língua Portuguesa**. Rio de Janeiro: Nova Fronteira. 1986.

FIORINI, M. M. **Estudo e análise comparativa de algumas características de arquitetura entre os sistemas holônicos de manufatura e a manufatura ágil**. 2000. 126 f. Dissertação (Mestrado em Engenharia Mecânica) – Programa de Pós Graduação em Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, 2000. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000218940&fd=y>>. Acesso em: 20 fev. 2016.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **FIPA request interaction protocol specification**. Genebra, mar. 2002. Disponível em: <<http://www.fipa.org/specs/fipa00026/SC00026H.html>>. Acesso em: 11 jun. 2015.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **FIPA agent management specification**. Genebra, mar. 2004. Disponível em: <<http://www.fipa.org/specs/fipa00023/SC00023K.html>>. Acesso em: 10 jun. 2015.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. **Welcome to fipa**. Disponível em: <<http://www.fipa.org>>. Acesso em: 07 de jul. 2015.

FRAZÃO, J. A. Q. N. **Abordagem multiagente em sistemas de recomendação web**. 2015. 79 f. Dissertação (Mestrado em Tecnologias e Sistemas Informáticos

Web) – Programa Tecnologias e Sistemas Informáticos Web, Universidade Aberta de Lisboa, Lisboa, 2015. Disponível em: <http://repositorioaberto.uab.pt/bitstream/10400.2/3896/1/TMTSIW_Joaquim%20Neto.pdf>. Acesso em: 27 set. 2015.

GIORGINI, P.; SELLER, B. H. **Agent-oriented methodologies**. Hershey: Idea Group Publishing, 2005.

HSIEH, F. S.; LIN J. B. A self-adaptation scheme for workflow management in multi-agent systems. **Journal of Intelligent Manufacturing**. v. 27, n.1, p.131-148, fev. 2016.

IGLESIA, G. D. et. al. A self-adaptive multi-agent system approach for collaborative mobile learning. **IEEE Transactions on Learning Technologies**. v. 8 n.2, 2015 p.158-172. Disponível em: < <https://www.computer.org/csdl/trans/lt/preprint/06948376.pdf>>. Acesso em: 25 set. 2015.

ISERN D.; SANCHEZ D.; MORENO A. **Organizational structures supported by agent-oriented methodologies**, The journal of Systems and Software, v. 84, n. 2, Oxford, Nova York: Elsevier, 2011, p.169-184.

JENNINGS, N. R. On agent-based software engineering. **Artificial Intelligence** v. 117, n. 2, p. 277-296, mar. 2000.

KALTON, A. Multi-agent-based simulation of a complex ecosystem of mental health care. **Jornal of Medical System**. v. 40 n. 39, fev. 2016.

KHAN, A. R.; SAQIB, S. M. Blind multi-agent system for department teaching schedule. **International Journal of Advanced Research in Computer Science**. Udaipur, v.2, n. 2, mar-abr. 2011. Disponível em: < <http://www.ijarcs.info/?wicket:interface=:2:::>>. Acesso em: 25 set. 2015.

LUCK, M.; MCBURNEY, P.; PREIST, C. **Agent technology: enabling ... generation computing**. 1. ed. Liverpool: AgentLink, 2003.

MAHDI, H.; ATTIA, S. S. MASCE: a multi-agent system for collaborative e-Learning. In: IEEE/ACS INTERNATIONAL CONFERENCE ON COMPUTER SYSTEM AND APPLICATIONS, 2008, Doha. **Anais ACS**, Nova York: IEEE, 2008. p. 925-926.

MAALAL, S.; ADDOU, M. A new approach of designing multi-agent systems with a practical sample. **(IJACSA) International Journal of Advanced Computer Science and Applications**, Nova York, v. 2, n.11, nov. 2011. Disponível em: <<http://thesai.org/Publications/ViewPaper?Volume=2&Issue=11&Code=IJACSA&SerialNo=26>>. Acesso em: ago. 2015.

MYSQL. **MySQL Server 5.5**. Disponível em: < <https://www.mysql.com/>>. Acesso em: 04 fev. 2016a.

MYSQL. **MySQL WorkBench**. Disponível em:< <https://www.mysql.com/products/workbench/>>. Acesso em: 04 fev. 2016b.

NETBEANS. **IDE netbeans**. Disponível em: <https://netbeans.org/features/index_pt_BR.html>. Acesso em: 03 fev. 2016.

NIKRAZ, M.; CAIRE B. G.; BAHRI, A.P. A methodology for the analysis and design of multi-agent systems using Jade. **International Journal of Computer Systems Science & Engineering**. Leicester, v. 21, n. 2, mar. 2006.

NOUREDDINE, F. E. et. al. Towards an competency-based learning system using assessment. **IJCSI International Journal of Computer Science Issues**, v. 8, n. 1, 2011. p. 265-274. Disponível em: <<http://ijcsi.org/papers/IJCSI-8-1-265-274.pdf>>. Acesso em: 21 set. 2015.

ODELL, J. **Agent technology: an overview**. out. 2010. Disponível em:<http://www.jamesodell.com/Agent_Technology-An_Overview.pdf>. Acesso em: 20 jun. 2015.

ORACLE. **Java platform JDK 8**. Disponível em: <<http://www.oracle.com/technetwork/pt/java/javase/overview/index.html>>. Acesso em: 03 fev. 2016.

PAIVA, R.; BITTENCOURT, I. I.; SILVA, A. P. Uma ferramenta para recomendação pedagógica baseada em mineração de dados educacionais. Congresso Brasileiro de Informática na Educação, 3, 2013, Campinas. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**, Porto Alegre: Sociedade Brasileira de Computação – SBC, 2013.

REN, F. et. al. **Conceptual design of A multi-agent system for interconnected power systems restoration**. IEEE Transactions on Power Systems. Nova York: IEEE, v.27, n. 2, jan. 2012. p.732-740.

RODRIGUES, D. F. **Acompanhamento do desenvolvimento profissional de egressos por meio de sistemas multiagentes**. 2014. 86 f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática, Universidade Federal de Viçosa, Viçosa, 2013. Disponível em: < http://www.dpi.ufv.br/arquivos/ppgcc/dissertacoes/2014-ms-Diego_Fialho_Rodrigues.pdf>. Acesso em: 24 set. 2015.

RUSSEL, S.; NORVIG, P. **Artificial intelligence a modern approach**. Nova Jersey: Prentice Hall, 1995.

SCHREIBER, G, et al, **Knowledge engineering and management: the commonKADS methodology**. Londres: Cambridge, 1999.

SELLERS, H. B. Towards the consolidation of a diagramming suite for agent-oriented modelling languages. **ISRN Software Engineering**, v. 2013, p.53, 2013.

SOLEIMANIAN F.; ZEBARDAST B.; AMINI E.. Analysis and design by agent based MaSE methodology: a case Study. **International Journal of Computer Applications**. v. 63, n. 4, fev. 2013. Disponível em: <<http://research.ijcaonline.org/volume63/number4/pxc3885156.pdf>>. Acesso em: 15 abr. 2015.

SYBASE. **Power Designer**. Disponível em: <<http://powerdesigner.de/en/>>. Acesso em: 03 fev. 2016.

TROPOS PROJECT. **The tropos methodology**. Disponível em: <<http://www.troposproject.org/node/93>> Acesso 15 de abril de 2015.

VIEIRA, F. L. **Um sistema multiagente para apoio as decisões no processo de licitação pública**. 2013. 114 f. Dissertação (Mestrado em Engenharia de Eletricidade) – Centro de Ciências Exatas e Tecnologia, Universidade Federal do Maranhão, Maranhão, 2013. Disponível em: <http://www.tedebc.ufma.br/tde_arquivos/10/TDE-2013-05-09T100045Z-773/Publico/dissertacao%20Fabio%20Lopes.pdf>. Acesso em: 24 set. 2015.

WERNECK, V. M. et.al. Metodologias orientada a agentes. **Cadernos do IME Série Informática**, Rio de Janeiro, v. 26, dez. 2008.

WERNECK, V., M. B.; COSTA, R. M. E. M.; CYSNEIROS, L. M. **Mode multiagent system using different methodologies**. Croacia: InTech, 2011.

WOOLDRIDGE, M.; JENNINGS N. R. Intelligent agents: theory and practice. **The Knowledge Engineering Review**, Cambridge, v. 10, n.2, p. 115-152, Jun. 1995.

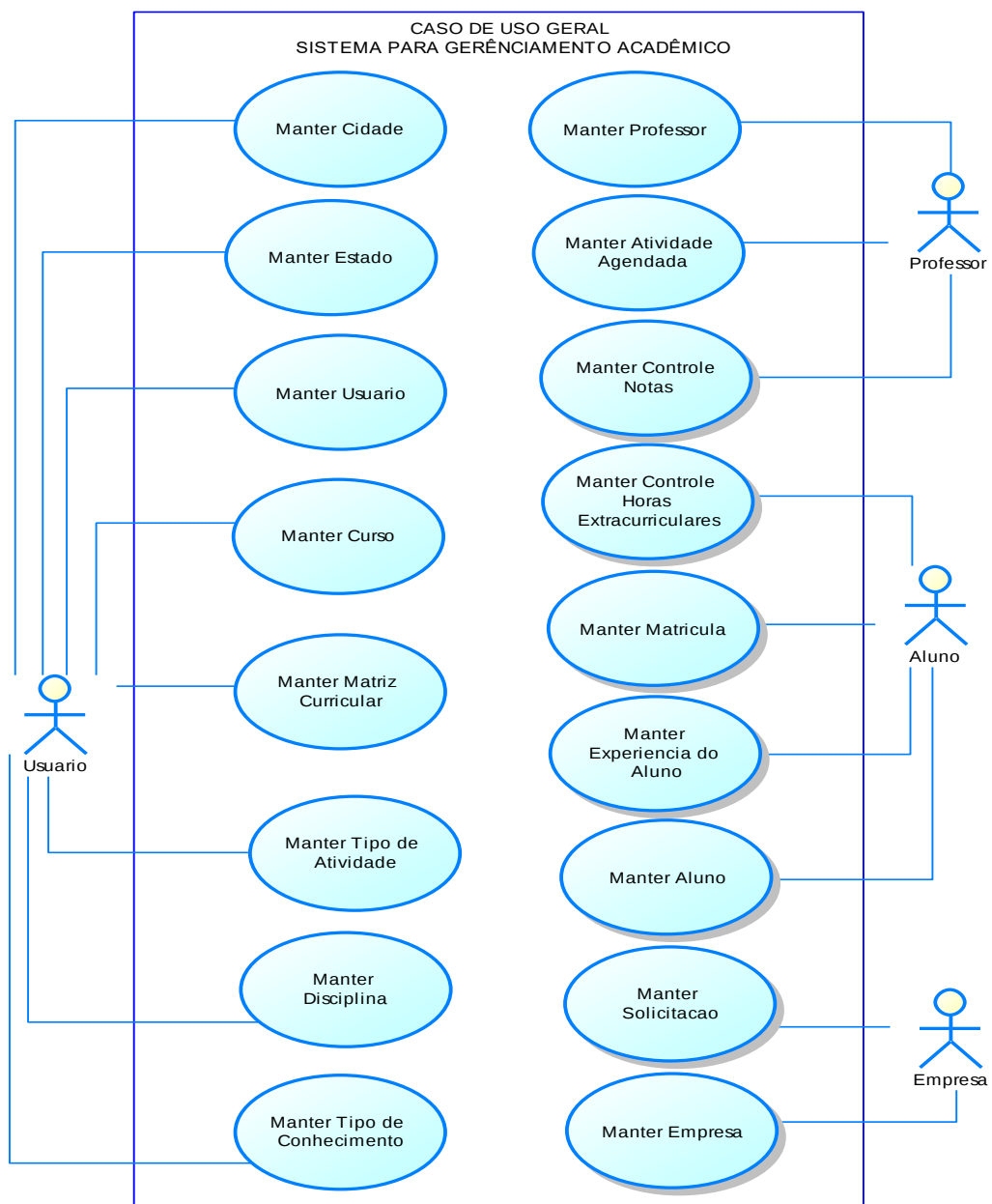
WOOLDRIDGE, M. **An introduction to multiagent systems**. 1. ed. Nova York: Wiley, 2002.

APÊNDICE A – DIAGRAMAS DE CASO DE USO E SEUS DETALHAMENTOS

Os diagramas a seguir foram utilizados para o desenvolvimento do sistema proposto.

Diagrama de Caso de Uso Geral

Foi desenvolvido de acordo com Bezerra (2007).



Fonte: O autor (2016).

Descrição de caso de Uso

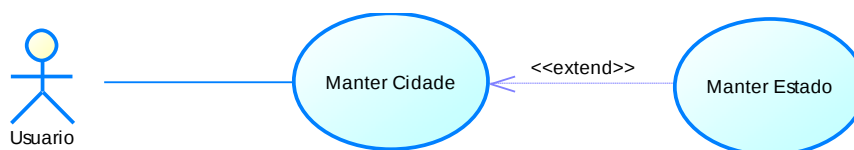
Foi desenvolvido no estilo de descrição narrativa particionada numerada conforme Bezerra (2007).

Caso de Uso Manter Estado



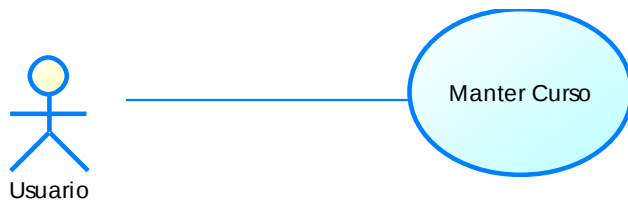
Nome do Caso de Uso: Manter Estado	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar, Navegar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro do Estado então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Navegar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do Estado
5. O Usuário insere todas as informações do Estado	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema valida os dados
	9. O sistema verifica se o Estado já Existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, estado registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Estado já esteja cadastrado o sistema emite mensagem: "Este Estado já está cadastrado!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Cidade



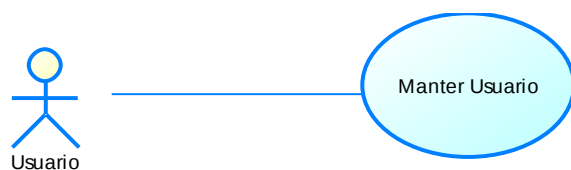
Nome do Caso de Uso: Manter Cidade	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s): Manter Estado	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro da Cidade então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro da cidade.
5. O Usuário insere todas as informações da Cidade.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se a Cidade já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.2 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.3 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Cidade registrada em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1 Caso o Estado para a Cidade não esteja cadastrado o sistema entende para o caso de uso Manter Estado e possibilita o cadastro do Estado.	
7.1. Caso a opção escolhida seja Cancelar o sistema cancela a operação e vai para o item 2 da descrição de caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso a Cidade já esteja cadastrada o sistema emite mensagem: "Esta Cidade já está cadastrada!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Curso



Nome do Caso de Uso: Manter Curso	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro do Curso então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do Curso.
5. O Usuário insere todas as informações do Curso.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se o Curso já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Curso registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Curso já esteja cadastrado o sistema emite mensagem: "Este Curso já está cadastrado!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Usuário



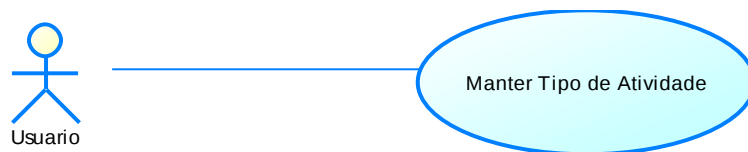
Nome do Caso de Uso: Manter Usuário	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Usuário solicita seu cadastro, então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o seu cadastro
5. O Usuário insere todas as informações.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se o Usuário já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: “não foi possível excluir, Usuário registrado em outra tabela” e vai para o item 2 da descrição do caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: “dados inválidos ou campo ‘nome do campo’ em branco” e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Usuário já esteja cadastrado o sistema emite mensagem: “Este Usuário já está cadastrado!” e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Disciplina



Nome do Caso de Uso: Manter Disciplina	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro da Disciplina então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro da disciplina.
5. O Usuário insere todas as informações da Disciplina.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se a Disciplina já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: “não foi possível excluir, Disciplina registrado em outra tabela” e vai para o item 2 da descrição do caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: “dados inválidos ou campo ‘nome do campo’ em branco” e vai para o item 5 da descrição do caso de uso.	
9.1. Caso a Disciplina já esteja cadastrado o sistema emite mensagem: “Esta Disciplina já está cadastrado!” e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Tipo Atividade



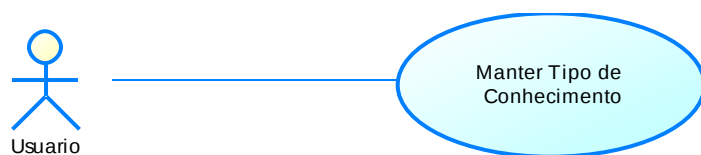
Nome do Caso de Uso: Manter Tipo Atividade	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro de um Tipo de Atividade então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do tipo de atividade
5. O Usuário insere todas as informações do Tipo de Atividade	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se o Tipo de atividade já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Tipo de Atividade registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Tipo de Atividade já esteja cadastrado o sistema emite mensagem: "Este Tipo de Atividade já está cadastrada!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Professor



Nome do Caso de Uso: Manter Professor	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Professor	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Cidade	
Curso Normal	
Ator	Sistema
1. O Professor solicita o seu cadastro ao Usuário então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do Professor
5. O Usuário insere todas as informações do Professor.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema valida os dados
	9. O sistema verifica se o Professor já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Professor registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1. Caso a Cidade para o Professor não esteja cadastrada, o sistema estende para o caso de uso Manter Cidade e possibilita o cadastro da Cidade.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Professor já esteja cadastrado o sistema emite mensagem: "Este Professor já está cadastrado!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Tipo de Conhecimento



Nome do Caso de Uso: Manter Tipo de Conhecimento	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s):	
Curso Normal	
Ator	Sistema
1. O Professor solicita o seu cadastro ao Usuário então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do Professor
5. O Usuário insere todas as informações do Professor.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema valida os dados
	9. O sistema verifica se o Professor já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Professor registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1. Caso a Cidade para o Professor não esteja cadastrada, o sistema estende para o caso de uso Manter Cidade e possibilita o cadastro da Cidade.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Professor já esteja cadastrado o sistema emite mensagem: "Este Professor já está cadastrado!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Aluno



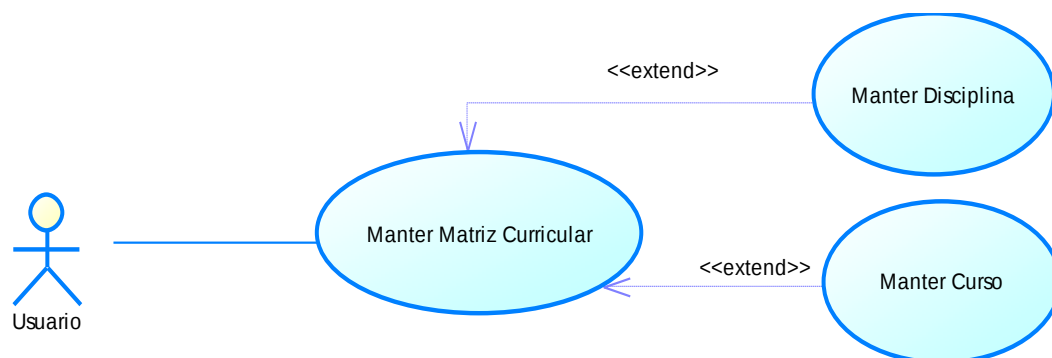
Nome do Caso de Uso: Manter Aluno	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Aluno	
Ator Secundário: Usuário (Secundário)	
Caso(s) de Uso envolvido(s): Manter Cidade	
Curso Normal	
Ator	Sistema
1. O Aluno solicita o seu cadastro ao Usuário então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro do Aluno
5. O Usuário insere todas as informações do Aluno.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema validará os dados
	9. O sistema verifica se o Aluno já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Aluno registrado em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1. Caso a Cidade para o Aluno não esteja cadastrada, o sistema estende para o caso de uso Manter Cidade e possibilita o cadastro da Cidade.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Aluno já esteja cadastrado o sistema emite mensagem: "Este Aluno já está cadastrado!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Empresa



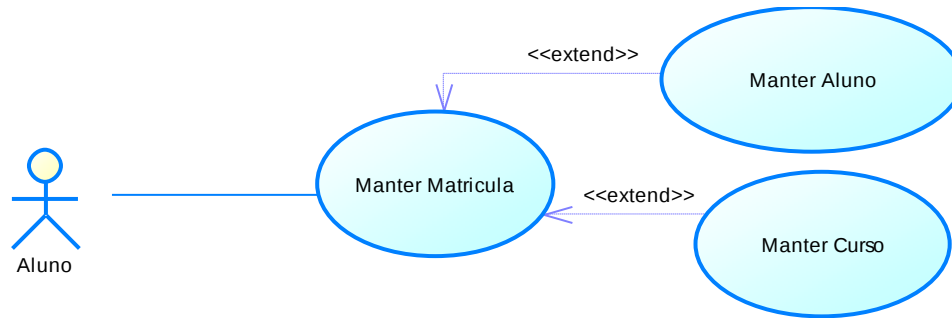
Nome do Caso de Uso: Manter Empresa	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Empresa	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Cidade	
Curso Normal	
Ator	Sistema
1. A Empresa solicita o seu cadastro ao Usuário então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar e Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o Usuário realizar o cadastro da Empresa.
5. O Usuário insere todas as informações da Empresa.	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
7. O Usuário clica em Salvar	
	8. O sistema valida os dados
	9. O sistema verifica se a Empresa já existe.
	10. O Sistema registra os dados informados imediatamente, assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Empresa registrada em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1. Caso a Cidade para a Empresa não esteja cadastrada, o sistema estende para o caso de uso Manter Cidade e possibilita o cadastro da Cidade.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso a Empresa já esteja cadastrado o sistema emite mensagem: "Esta Empresa já está cadastrada!" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Matriz Curricular



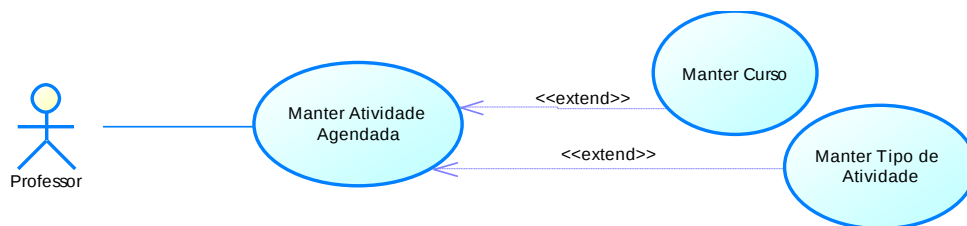
Nome do Caso de Uso: Manter Matriz Curricular	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário:	
Caso(s) de Uso envolvido(s): Manter Disciplina, Manter Curso	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro de uma Matriz Curricular, então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados da matriz.
5. O Usuário fornece os dados da matriz	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens da matriz
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Funcionário clica em Salvar	
	12. O sistema valida os dados da Matriz.
	13. O sistema registra os dados da Matriz. Assim, este caso de uso está encerrado.
Curso Alternativo	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 9 da descrição do caso de uso.	

Caso de Uso Manter Matrícula



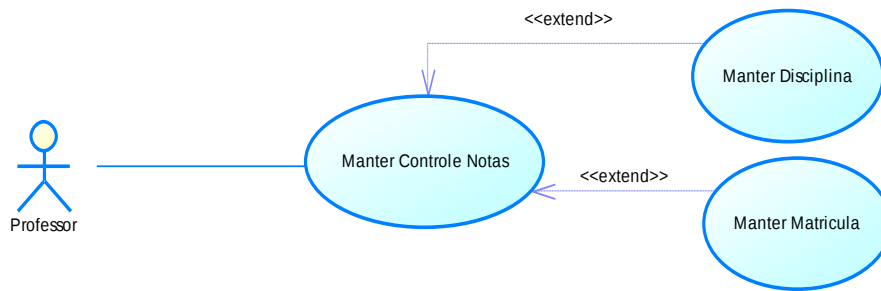
Nome do Caso de Uso: Manter Matrícula	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Usuário	
Ator Secundário: Aluno	
Caso(s) de Uso envolvido(s): Manter Aluno, Manter Curso	
Curso Normal	
Ator	Sistema
1. O Usuário solicita o cadastro de uma Matrícula então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados da matrícula.
5. O Usuário fornece os dados da matrícula	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados da Matrícula.
	9. O sistema registra os dados da Matrícula. Assim, este caso de uso está encerrado.
Curso Alternativo	
3.1. Caso a opção escolhida seja a de Excluir, o sistema verifica se o registro a ser excluído já esteja relacionado em alguma outra tabela.	
3.1.1 Caso não esteja relacionado, a Exclusão é realizada e o sistema vai para o item 2 da descrição do caso de uso.	
3.1.2 Caso esteja relacionado, o Sistema emite mensagem: "não foi possível excluir, Matrícula registrada em outra tabela" e vai para o item 2 da descrição do caso de uso.	
5.1. Caso o Aluno não esteja cadastrado, o sistema estende para o caso de uso Manter Aluno e possibilita o cadastro do Aluno.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	

Caso de Uso Manter Atividade Agendada



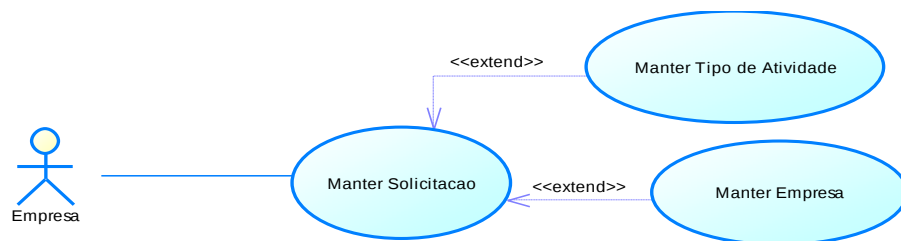
Nome do Caso de Uso: Manter Atividade Agendada	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Professor	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Cidade, Manter Curso, Manter Tipo de Atividade	
Curso Normal	
Ator	Sistema
1. O Professor solicita ao usuário o cadastro de uma Atividade Agendada, então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados das Atividades Agendadas.
5. O Usuário fornece os dados das Atividades Agendadas	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens das Atividades.
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Funcionário clica em Salvar	
	12. O sistema valida os dados das Atividades.
	13. O sistema registra os dados das Atividades, este caso de uso está encerrado.
Curso Alternativo	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1 Caso o Tipo de Atividade não esteja cadastrada, o sistema estende para o caso de uso Manter Tipo de Atividade e possibilita o cadastro do Tipo de Atividade.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 9 da descrição do caso de uso.	

Caso de Uso Manter Controle Notas



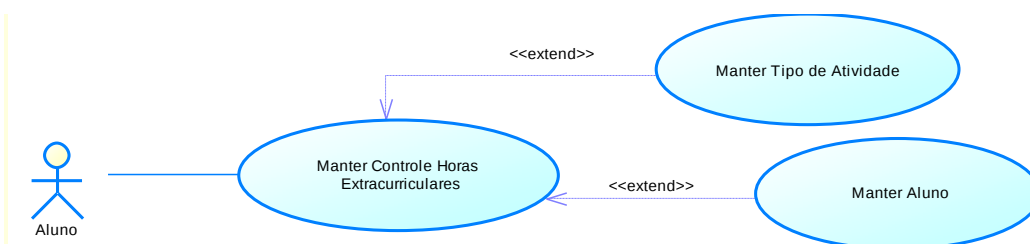
Nome do Caso de Uso: Manter Controle Notas	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Professor	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Matrícula, Manter Disciplina	
Curso Normal	
Ator	Sistema
1. O Professor solicita ao usuário o controle de uma nota, então o caso de uso é iniciado.	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados do controle notas.
5. O Usuário fornece os dados da solicitação	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens de controle notas.
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Usuário clica em Salvar	
	12. O sistema valida os dados do Controle
	13. O sistema registra os dados do Controle este caso de uso está encerrado.
Curso Alternativo	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos" e vai para o item 5 da descrição do caso de uso.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos" e vai para o item 9 da descrição do caso de uso.	

Caso de Uso Manter Solicitação



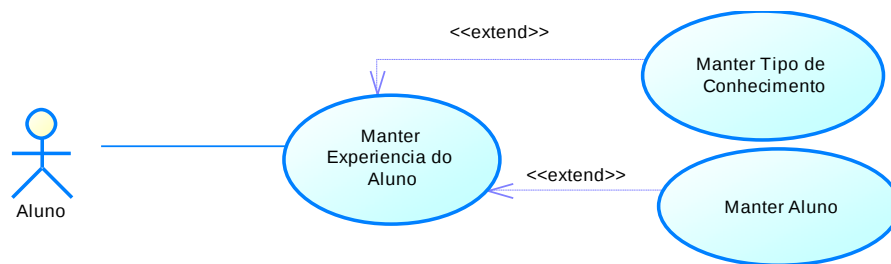
Nome do Caso de Uso: Manter Controle Solicitação	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Empresa	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Tipo de Atividade, Manter Empresa	
Curso Normal	
Ator	Sistema
1. A Empresa solicita ao usuário o cadastro de uma solicitação, então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados da solicitação.
5. O Usuário fornece os dados da solicitação	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens da solicitação
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Funcionário clica em Salvar	
	12. O sistema valida os dados da Solicitação.
	13. O sistema registra os dados da Solicitação este caso de uso está encerrado.
Curso Alternativo	
5.1. Caso a Empresa não esteja cadastrada, o sistema estende para o caso de uso Manter Empresa e possibilita o cadastro da Empresa.	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1. Caso o Tipo de Atividade não esteja cadastrado, o sistema estende para o caso de uso Manter Tipo de Atividade e possibilita o cadastro do Tipo de Atividade.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos" e vai para o item 9 da descrição do caso de uso.	

Caso de Uso Controle Horas Extracurriculares



Nome do Caso de Uso: Manter Controle Solicitação	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Aluno	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Tipo de Atividade, Manter Empresa	
Curso Normal	
Ator	Sistema
1. O Aluno solicita ao usuário o cadastro de uma Hora Extracurricular, então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados das Horas extracurriculares.
5. O Usuário fornece os dados das Horas	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens das Horas.
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Funcionário clica em Salvar	
	12. O sistema valida os dados das Horas.
	13. O sistema registra os dados das Horas, este caso de uso está encerrado.
Curso Alternativo	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 5 da descrição do caso de uso.	
9.1 Caso o Tipo de Atividade não esteja cadastrada, o sistema estende para o caso de uso Manter Tipo de Atividade e possibilita o cadastro do Tipo de Atividade.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: "dados inválidos ou campo 'nome do campo' em branco" e vai para o item 9 da descrição do caso de uso.	

Caso de Uso Manter Experiência do Aluno



Nome do Caso de Uso: Manter Experiência do Aluno	
Descrição: Este caso de uso permite a Novo, Alterar, Salvar, Cancelar, Excluir, Fechar, Pesquisar.	
Ator Primário: Aluno	
Ator Secundário: Usuário	
Caso(s) de Uso envolvido(s): Manter Aluno, Manter Tipo de Conhecimento	
Curso Normal	
Ator	Sistema
1. O Aluno solicita ao usuário lança suas experiências então o caso de uso é iniciado	
	2. O Sistema apresenta as seguintes operações a serem realizadas: Novo, Alterar, Salvar, Excluir, Pesquisar, Fechar.
3. O Usuário escolhe a opção Novo	
	4. O sistema possibilita o cadastro dos dados dos Conhecimentos.
5. O Usuário fornece os dados dos Conhecimentos	
	6. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar
7. O Usuário clica em Salvar	
	8. O sistema valida os dados e abre a janela para inserção de itens das Atividades.
9. O Usuário insere os dados do item	
	10. O Sistema possibilita nesse momento as operações: Cancelar ou Salvar.
11. O Funcionário clica em Salvar	
	12. O sistema valida os dados das experiências
	13. O sistema registra os dados das Experiências, este caso de uso está encerrado.
Curso Alternativo	
8.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: “dados inválidos ou campo ‘nome do campo’ em branco” e vai para o item 5 da descrição do caso de uso.	
9.1 Caso o Tipo de Conhecimento não esteja cadastrado, o sistema estende para o caso de uso Manter Tipo de Conhecimentos e possibilita o cadastro do Tipo de Conhecimento.	
12.1. Caso algum dado obrigatório esteja em branco ou inválido o sistema exibe mensagem: “dados inválidos ou campo ‘nome do campo’ em branco” e vai para o item 9 da descrição do caso de uso.	

APÊNDICE B – DIAGRAMAS DE SEQUÊNCIA UML

Diagrama de Sequência Manter Estado

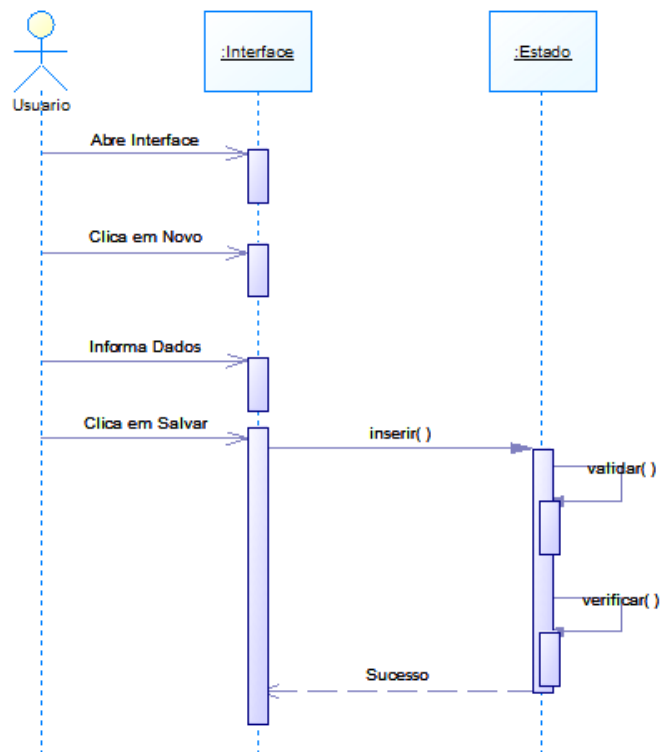


Diagrama de Sequência Cidade

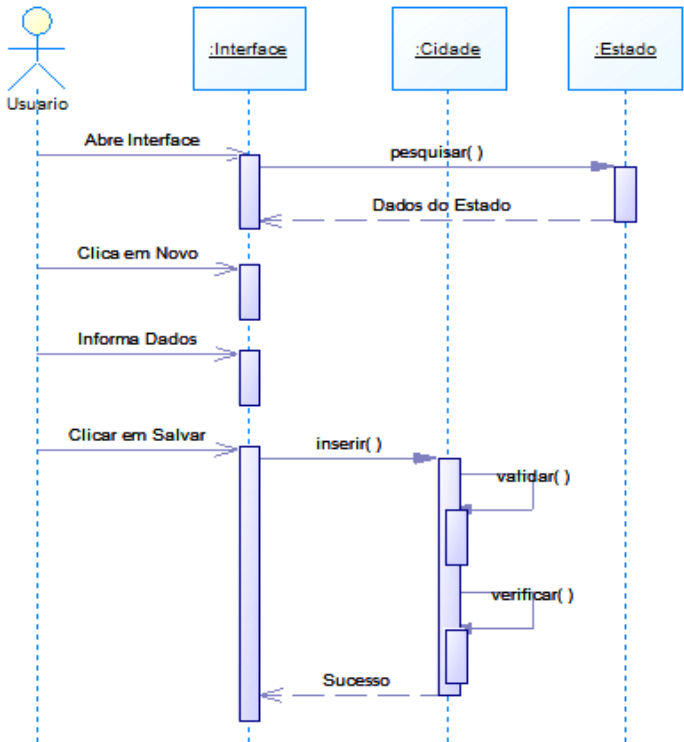


Diagrama de Sequência Manter Curso

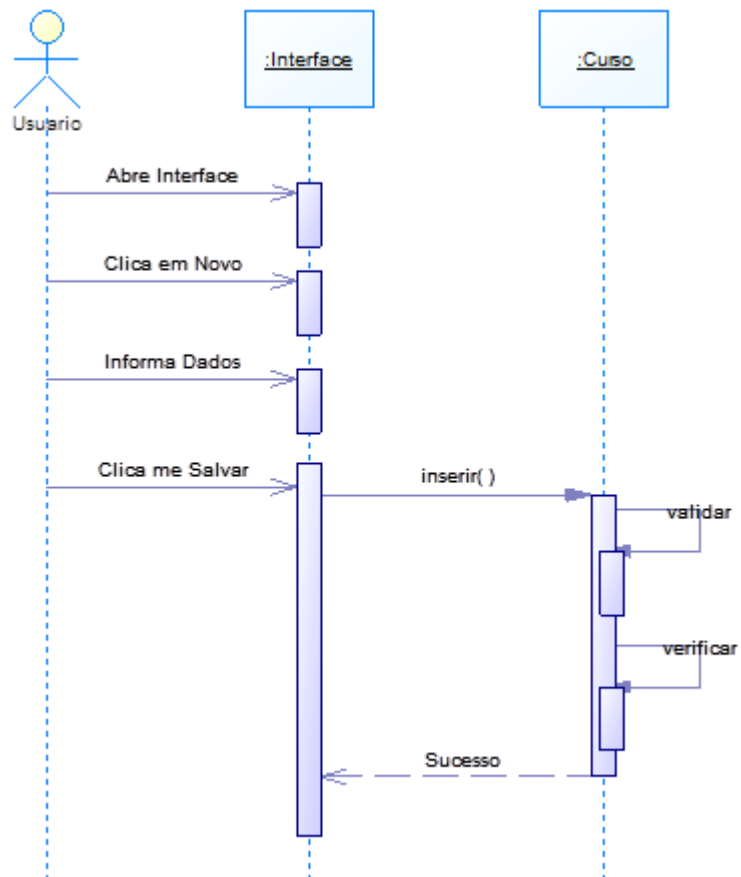


Diagrama de Sequência Manter Usuário

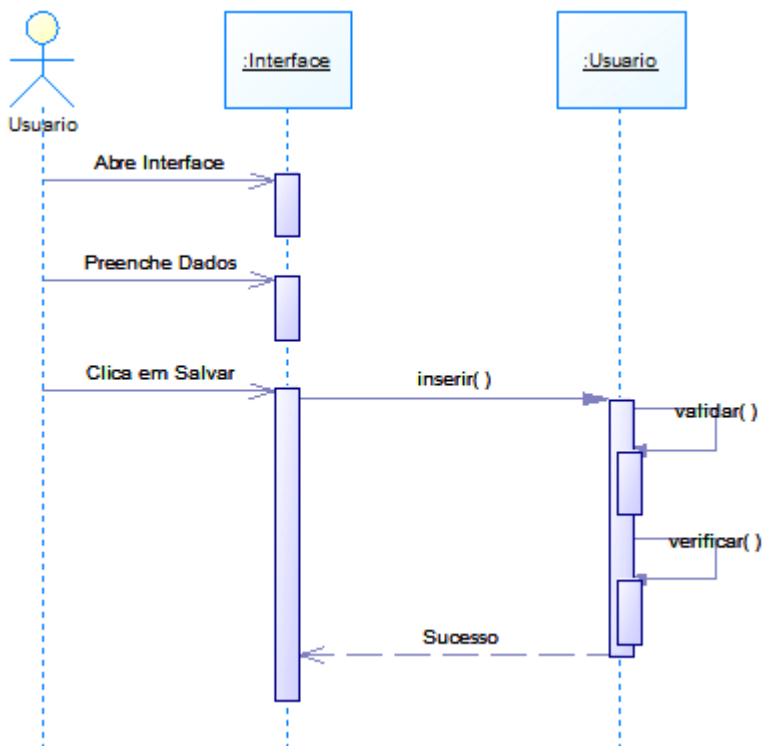


Diagrama de Sequência Manter Disciplina

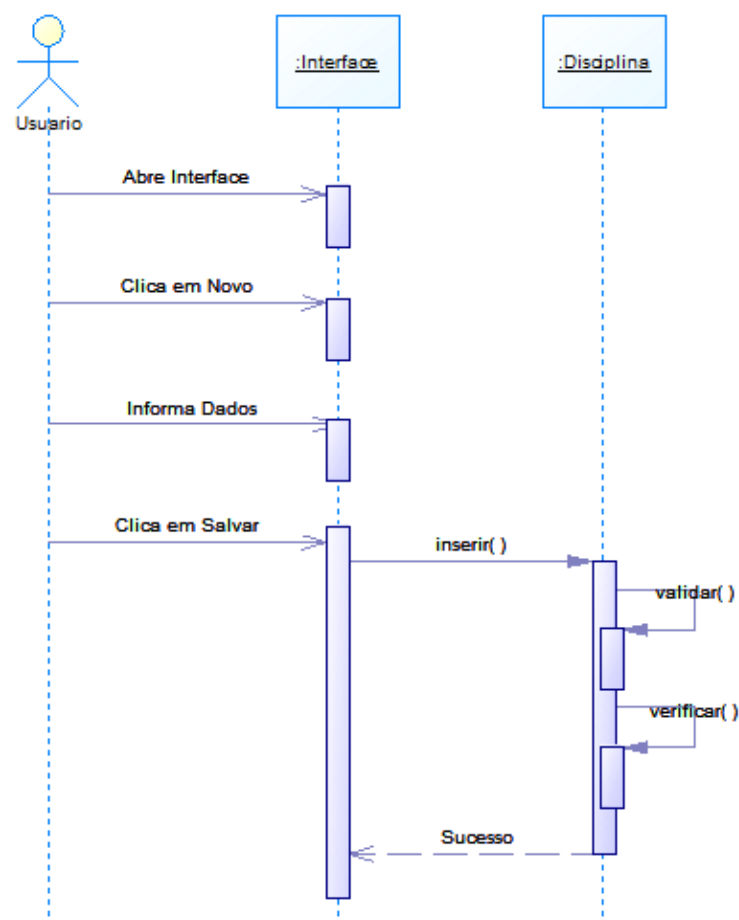


Diagrama de Sequência Manter Tipo Atividade

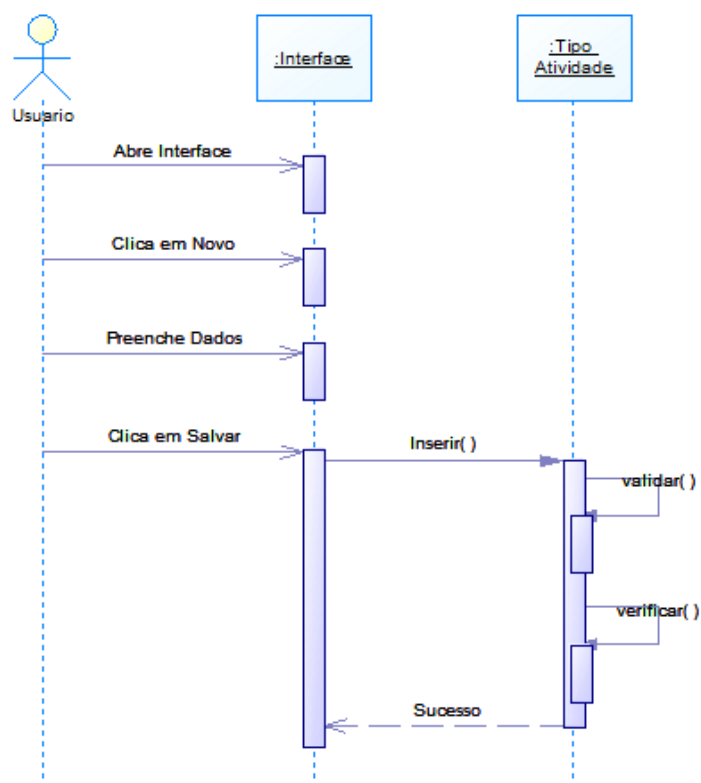


Diagrama de Sequência Manter Professor

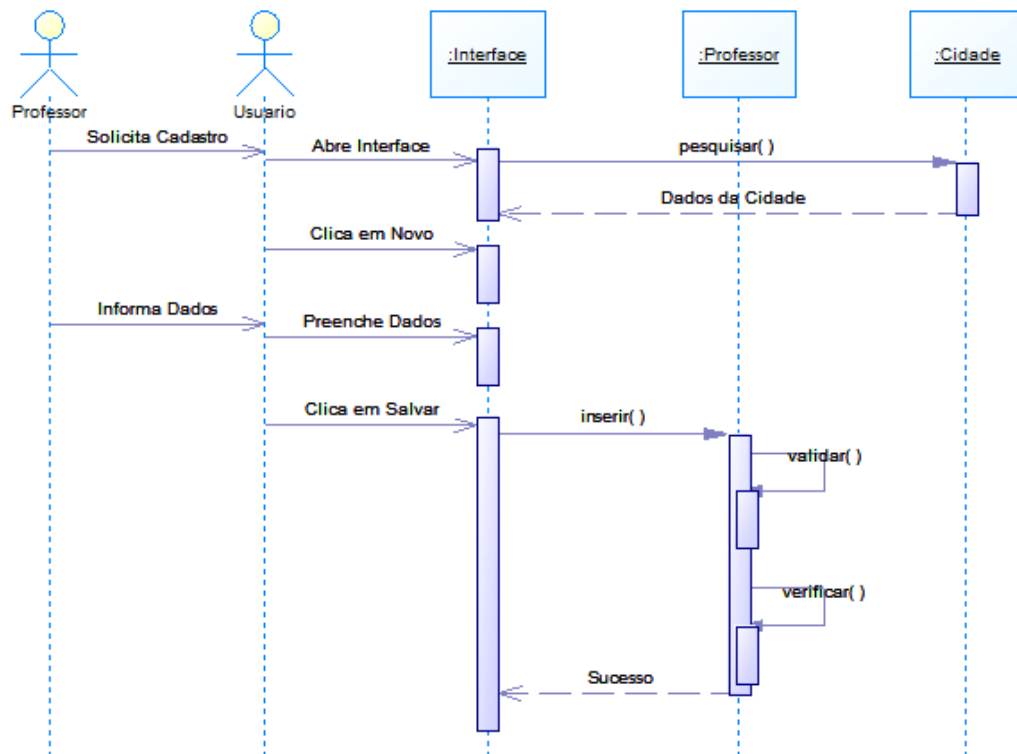


Diagrama de Sequência Manter Tipo de Conhecimento

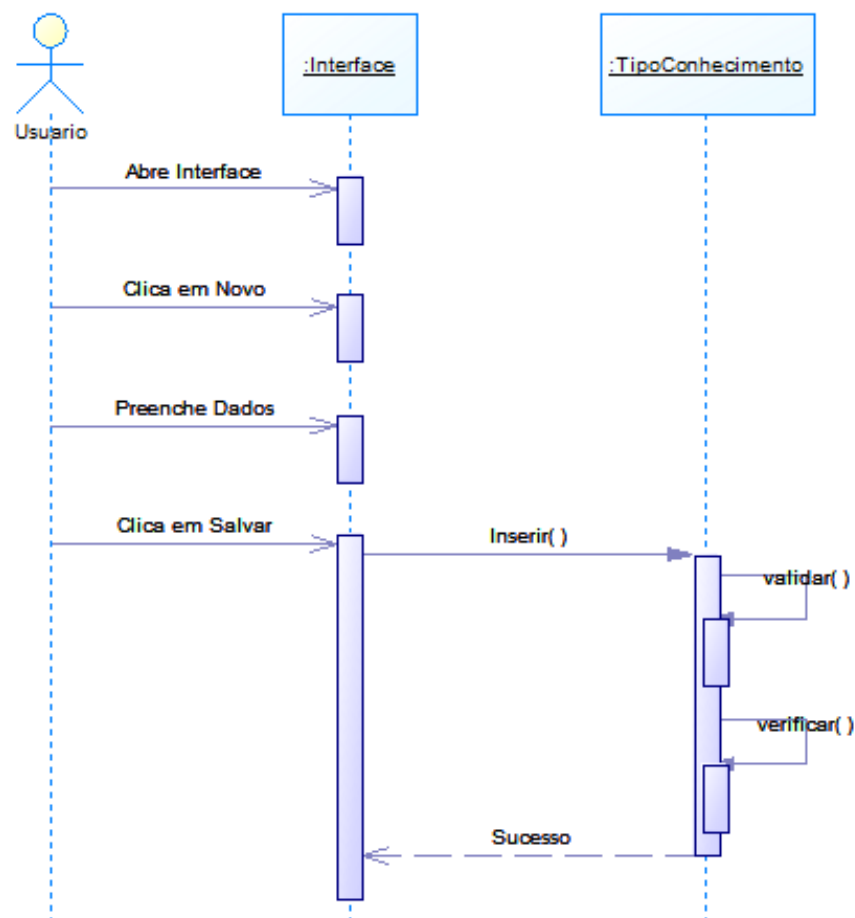


Diagrama de Sequência Manter Aluno

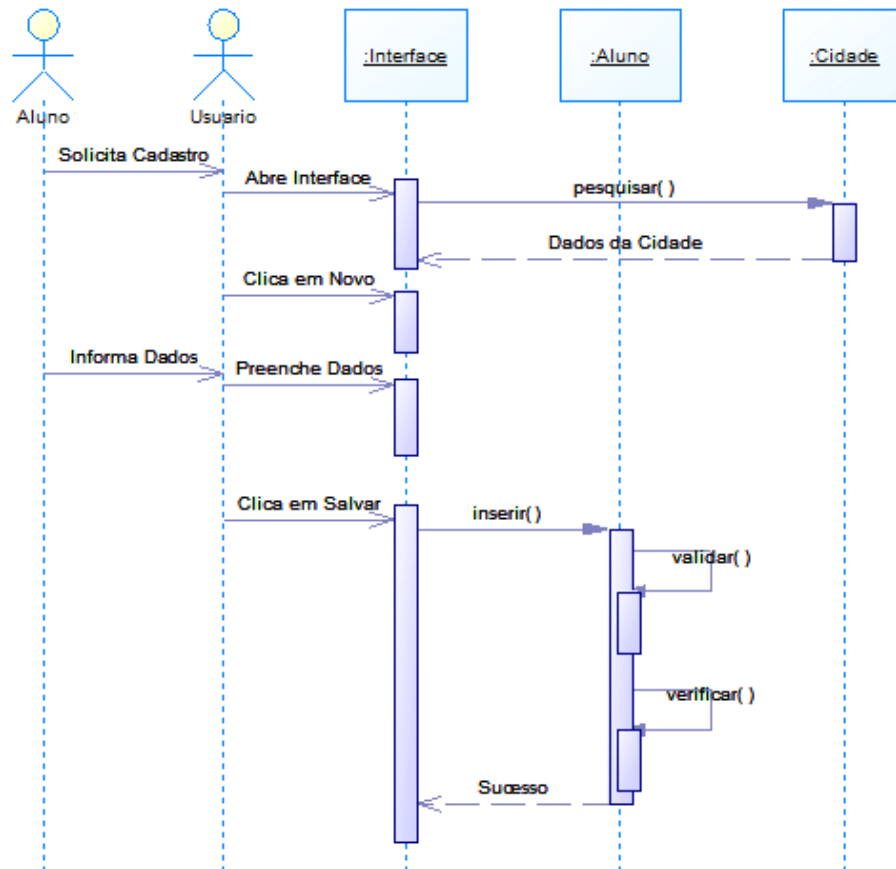


Diagrama de Sequência Manter Empresa

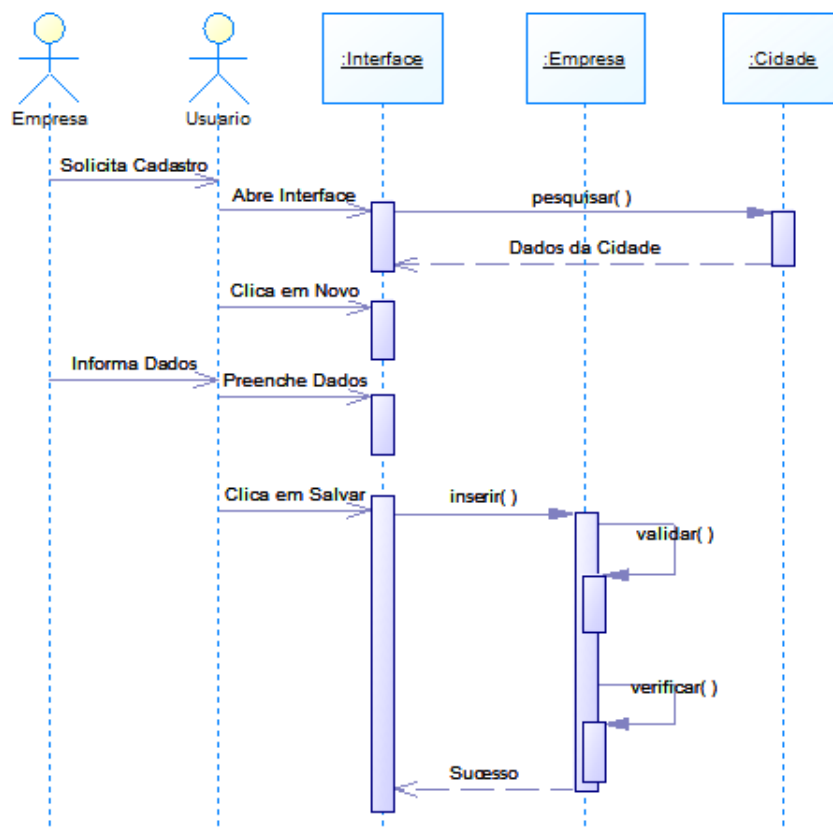


Diagrama de Sequência Manter Matrícula

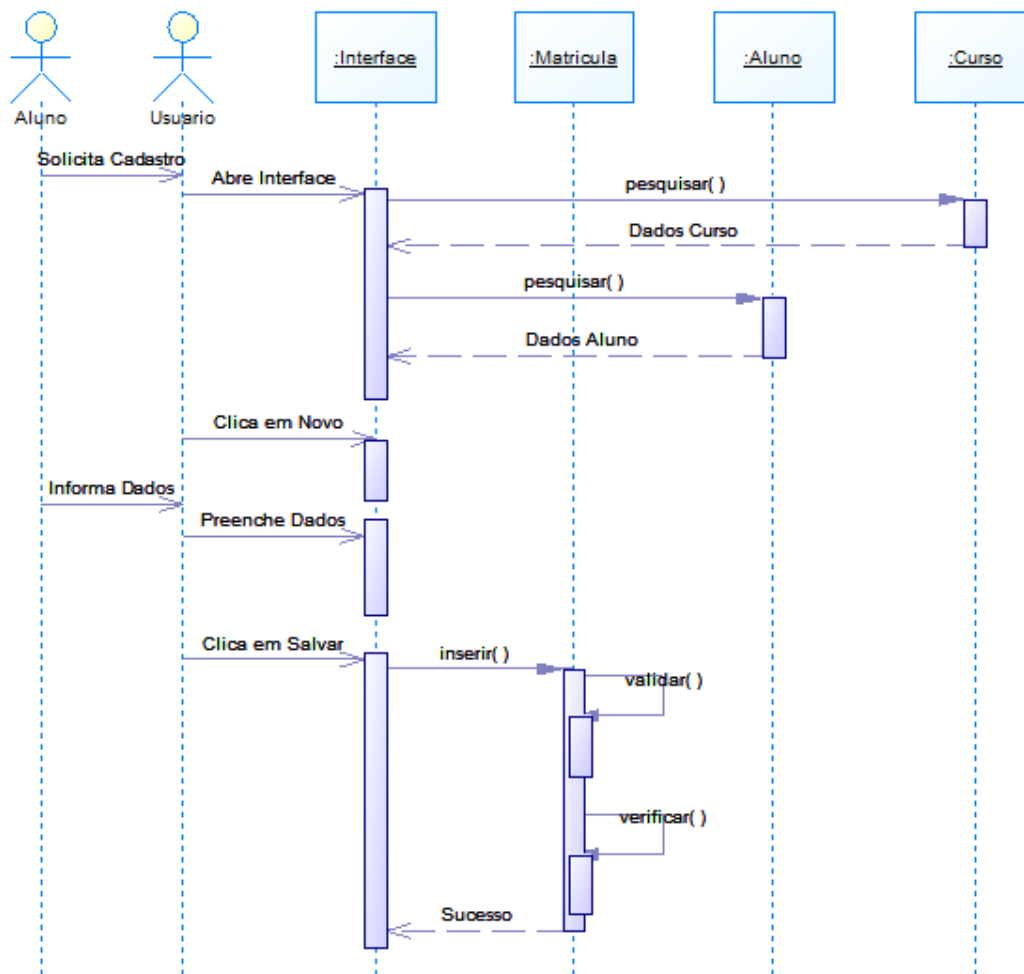


Diagrama de Sequência Manter Atividade Agendada

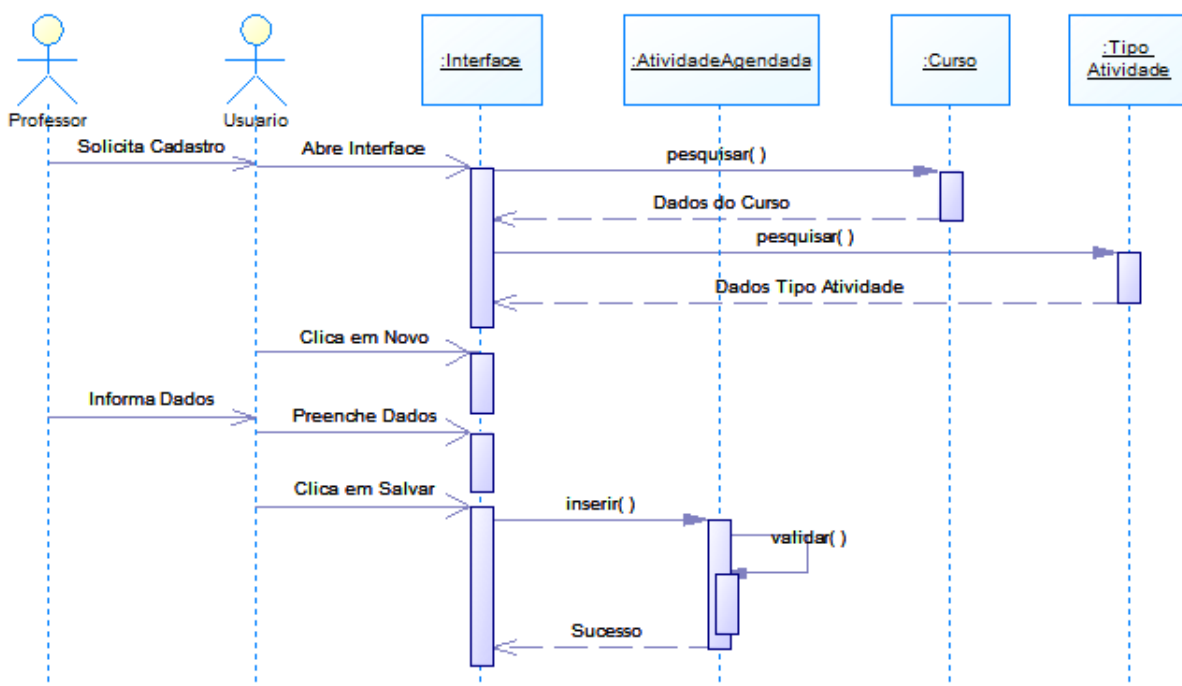


Diagrama de Sequência Manter Matriz Extracurricular

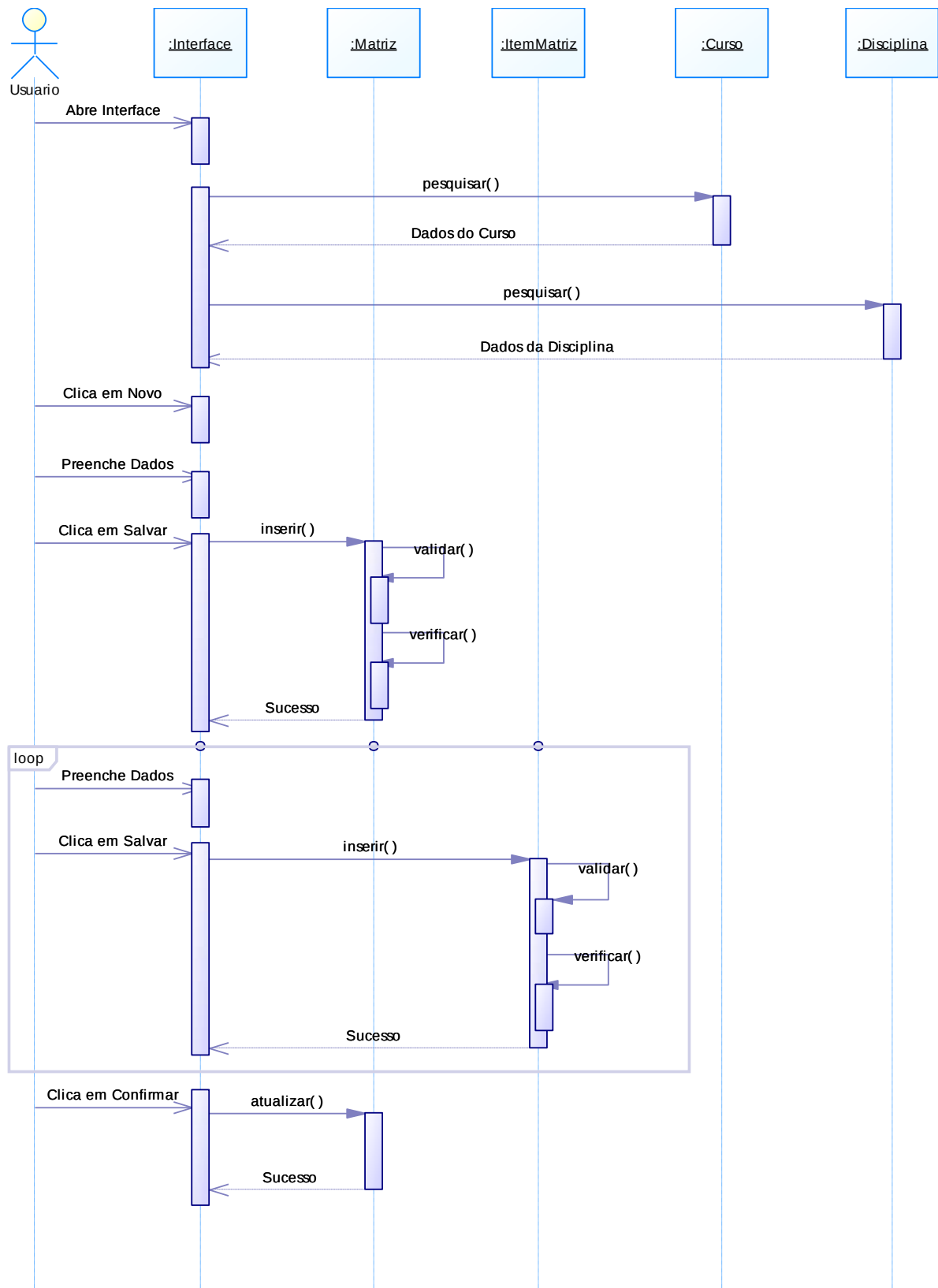
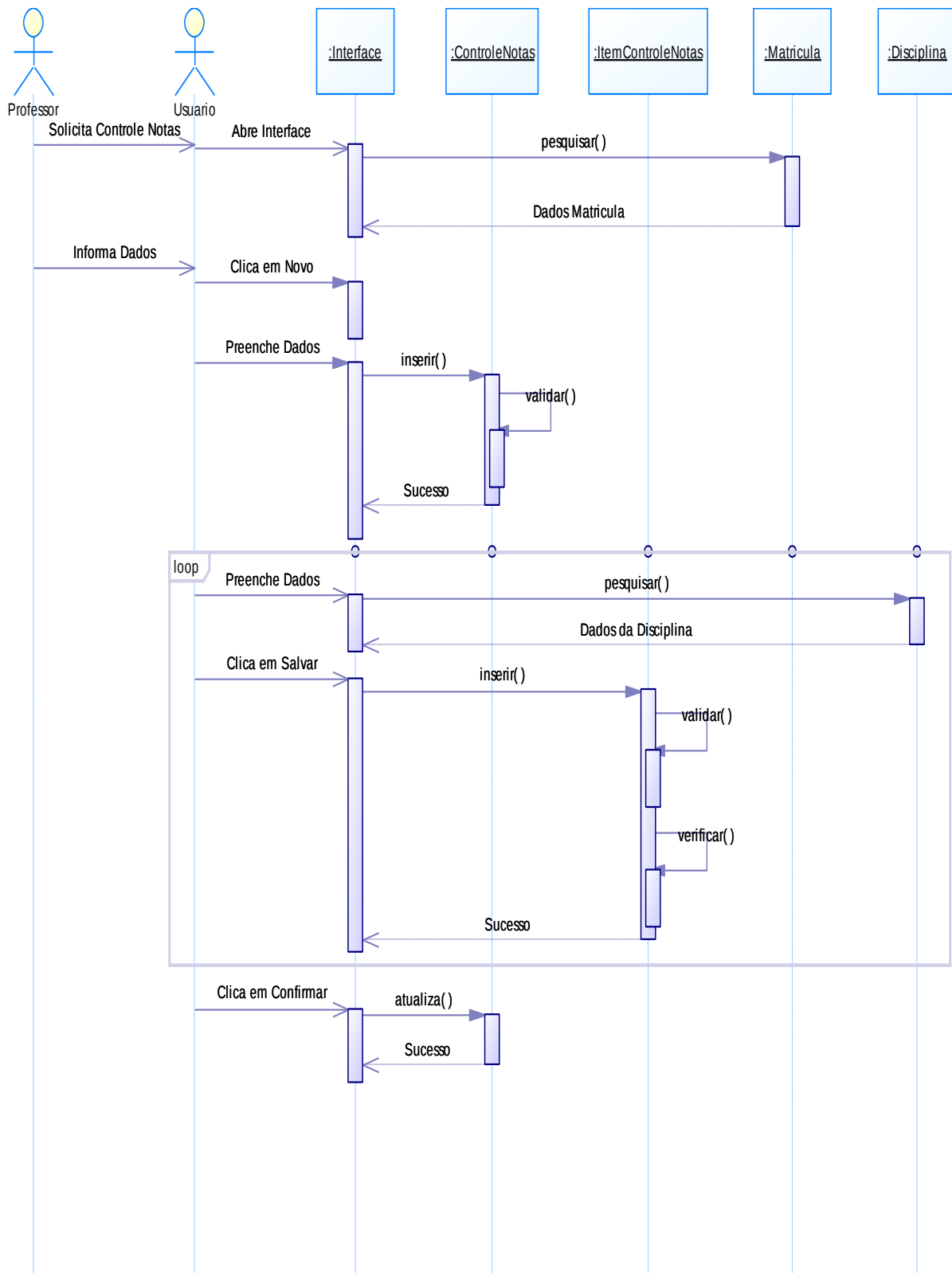
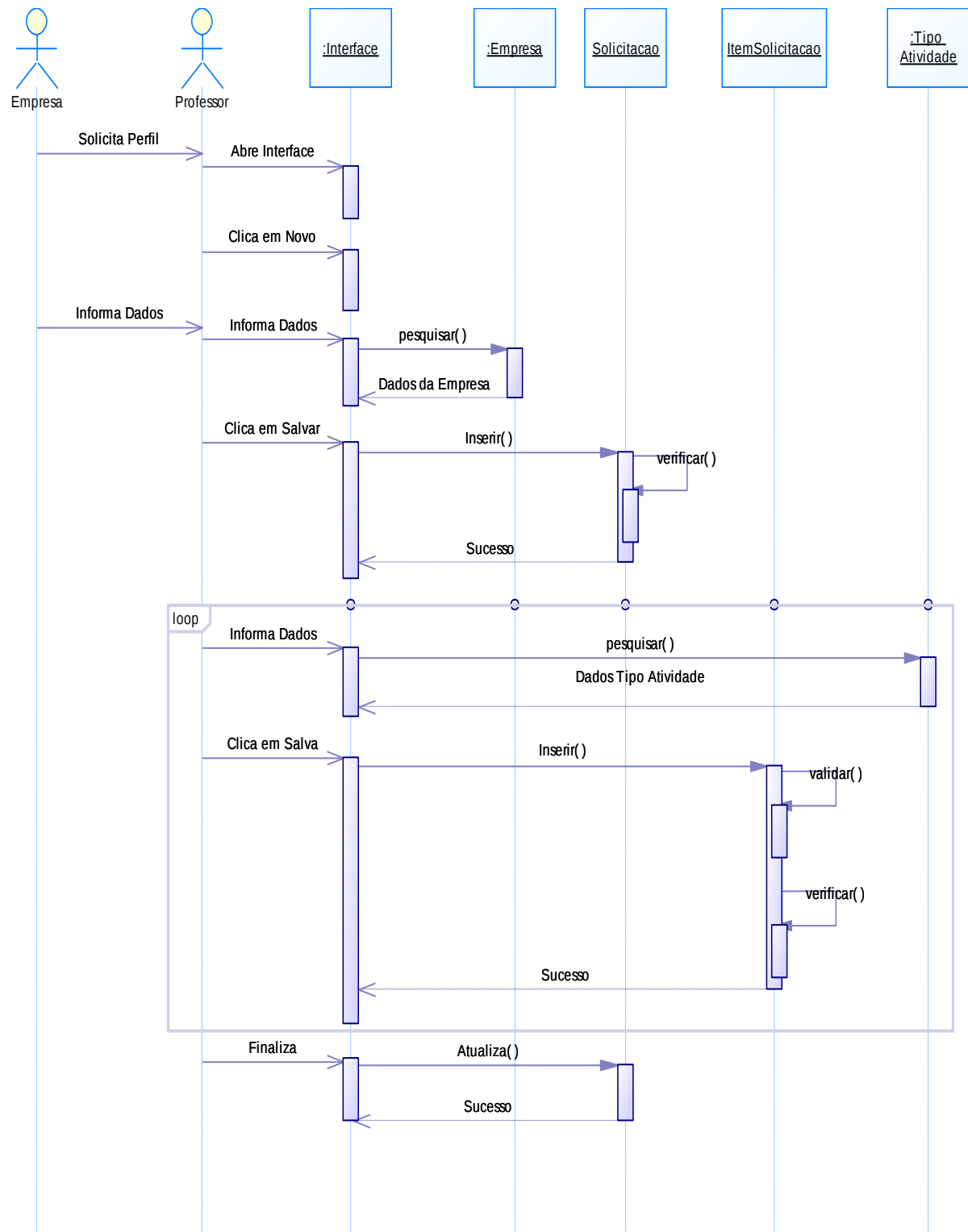


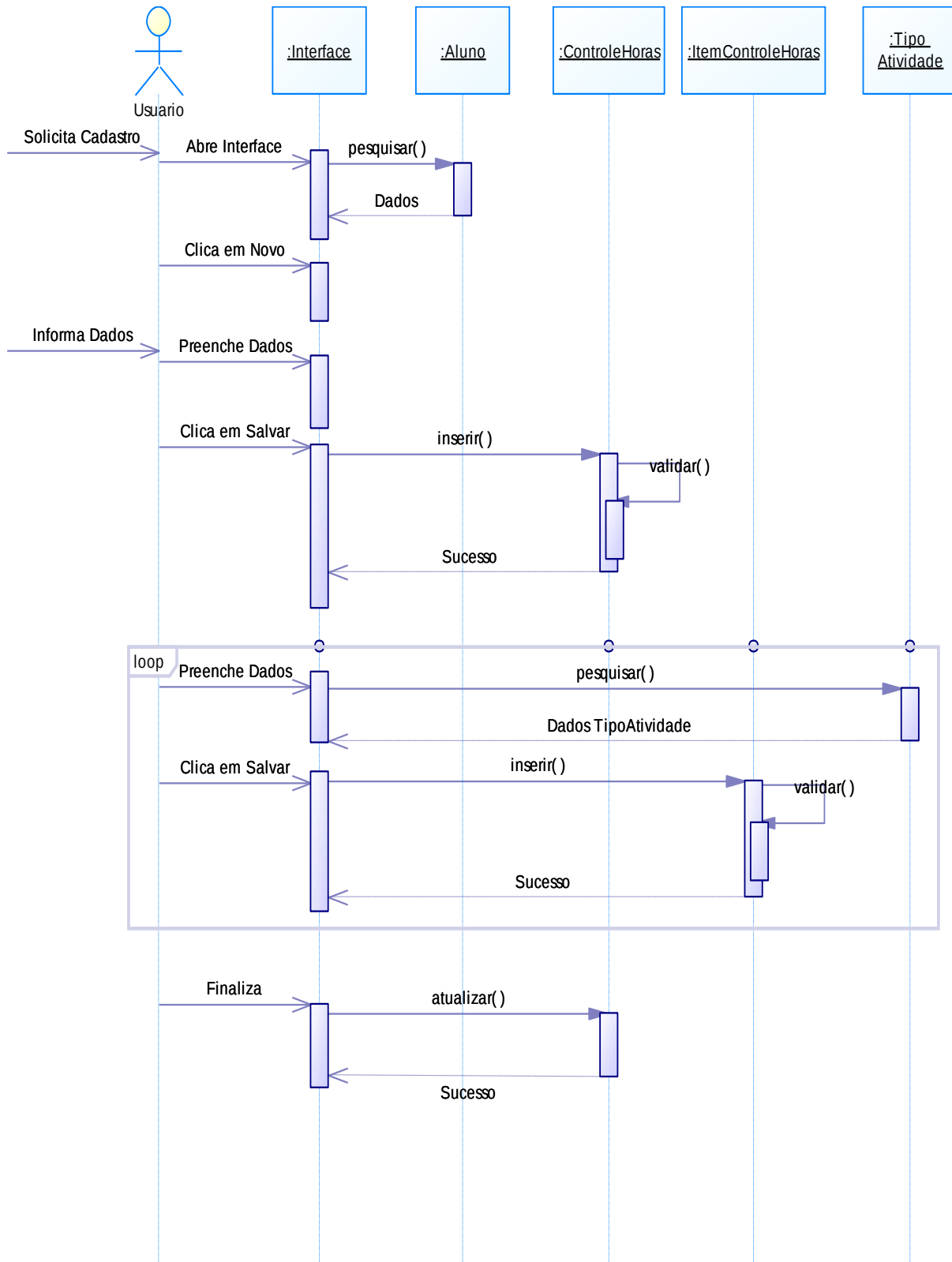
Diagrama de Sequência Manter Controle Notas



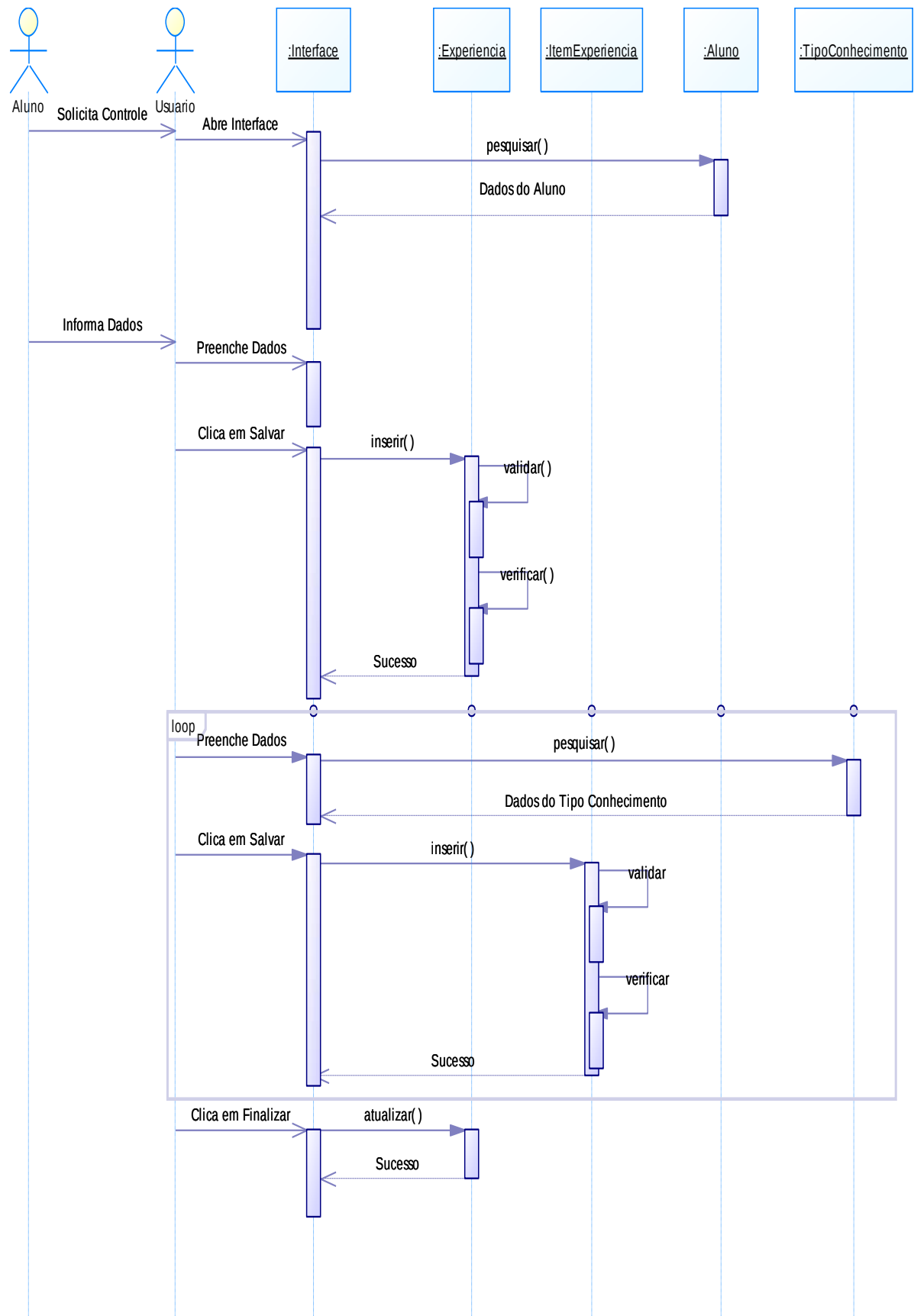
Caso de Uso Manter Solicitação



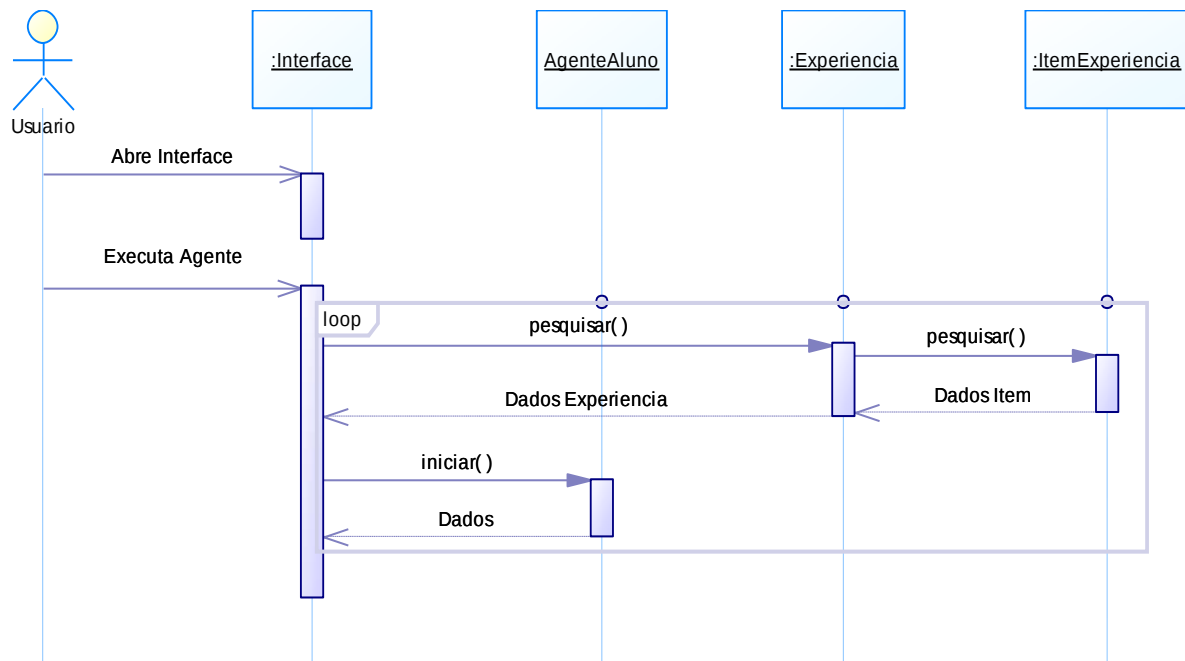
Caso de Uso Controle Horas



Caso de Uso Manter Experiência



Caso de Uso Manter Agente Aluno



Caso de Uso Manter Agente Empresa

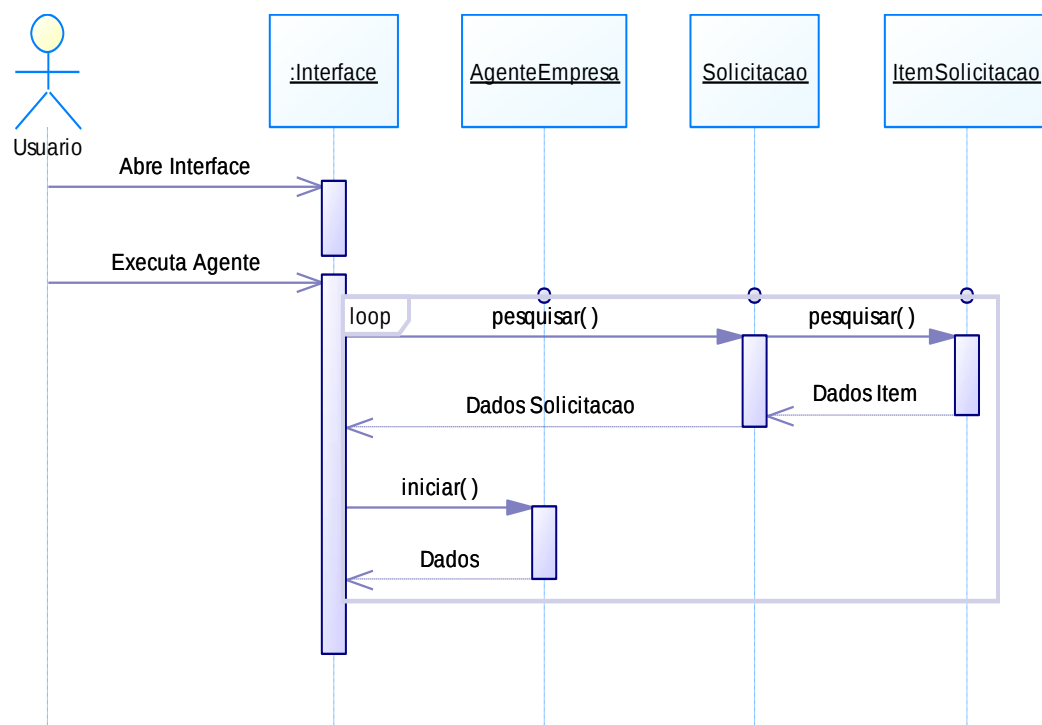


Diagrama de Sequência Agente Registra Médias

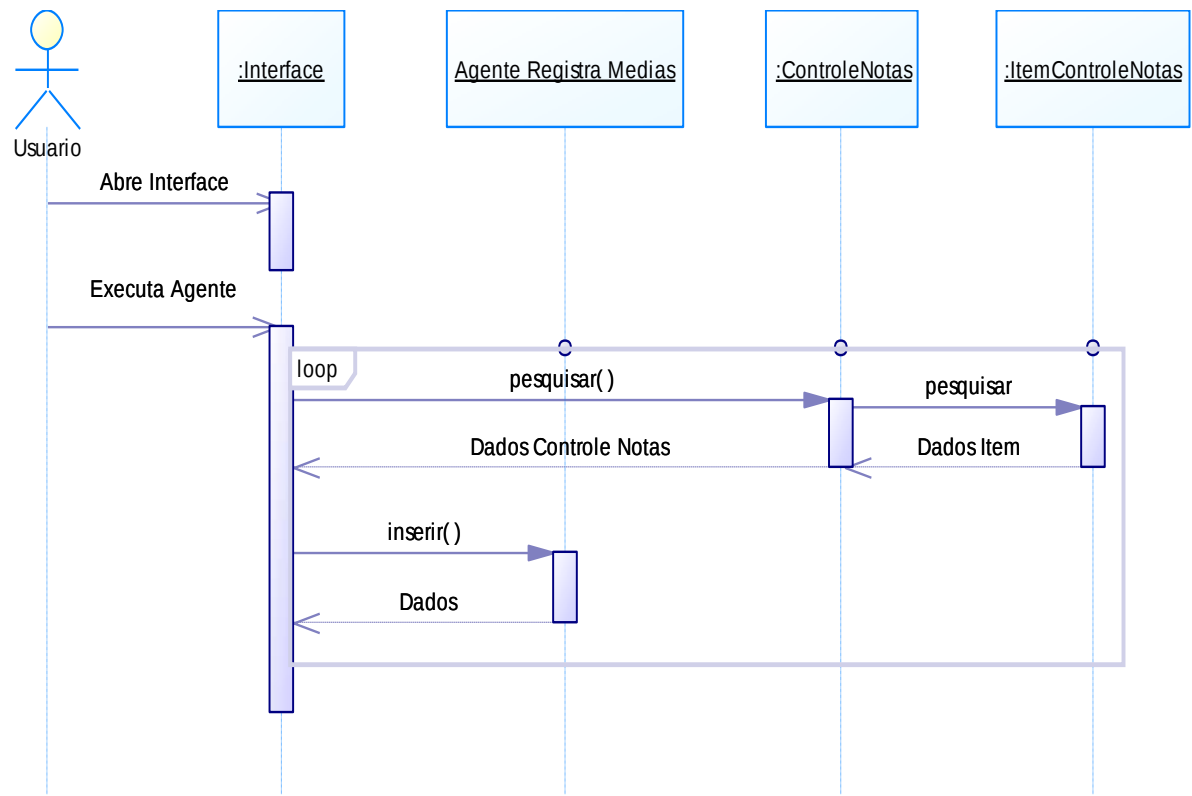


Diagrama de Sequência Agente Busca Médias

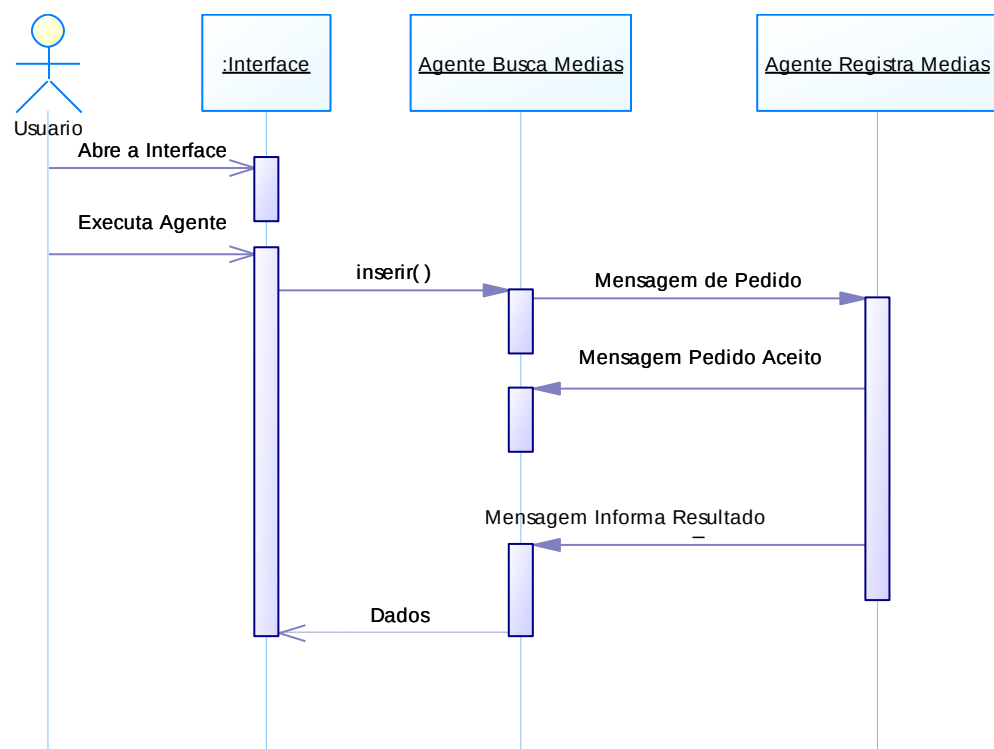
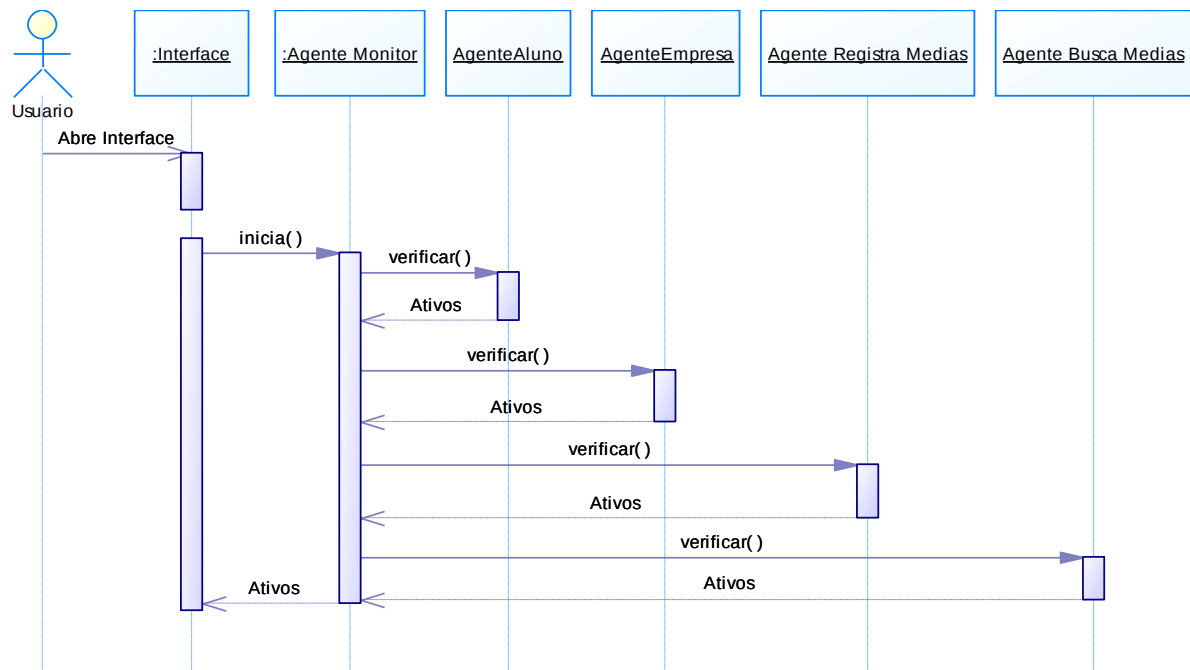


Diagrama de Sequência Agente Monitor

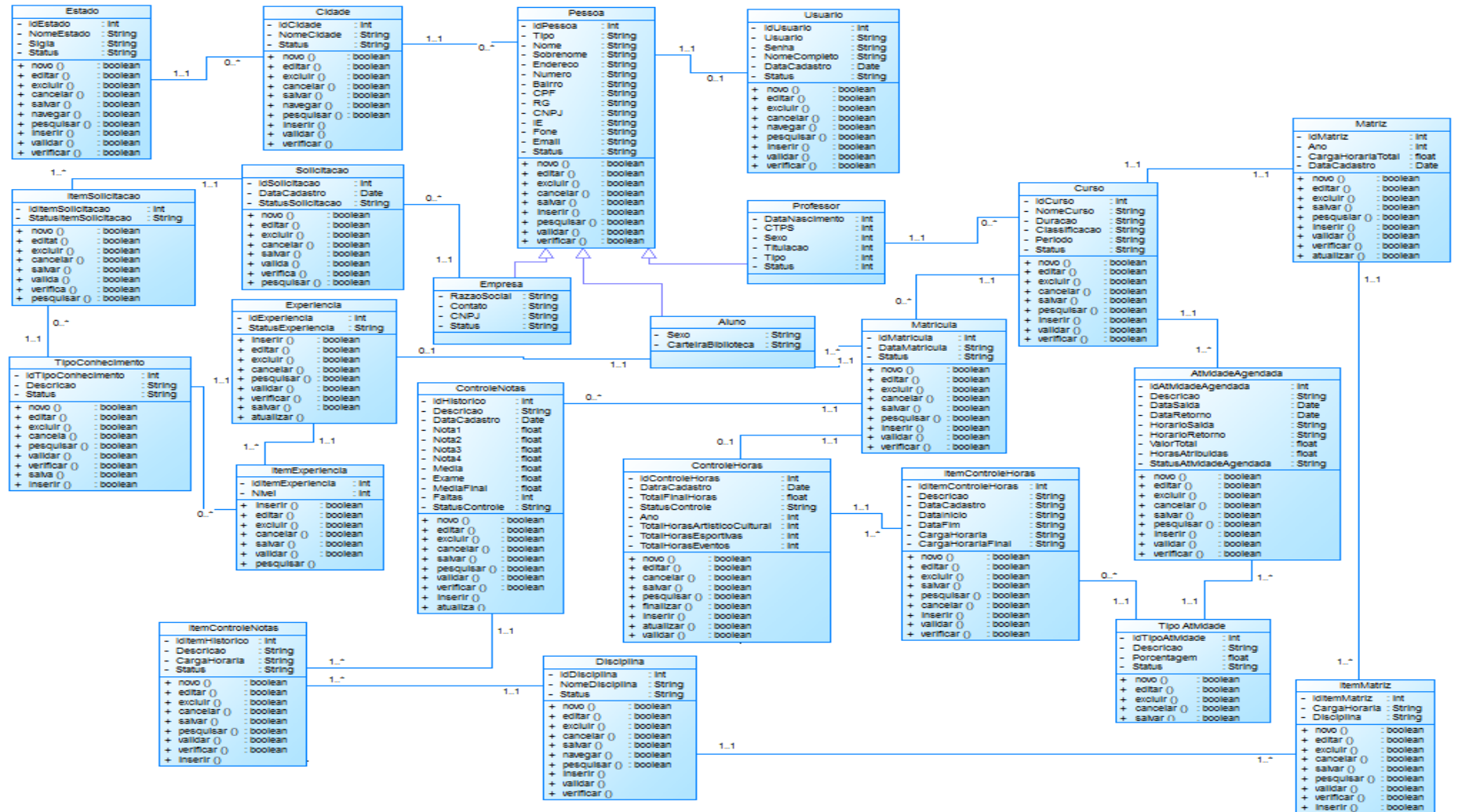


APÊNDICE C – DIAGRAMA HIERÁRQUICO TELAS DO SISTEMA



Fonte: O Autor (2016)

APÊNDICE D – DIAGRAMA DE CLASSES



Fonte: O autor (2016)