



Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Fabiane Flores Penteadó Galafassi

*Agente Pedagógico para Mediação do Processo de Ensino-
Aprendizagem da Dedução Natural na Lógica*

São Leopoldo, 2013

FABIANE FLORES PENTEADO GALAFASSI

**Agente Pedagógico para Mediação do Processo de Ensino-
Aprendizagem da Dedução Natural na Lógica
Proposicional**

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – Unisinos.

Área de Atuação: Ciências Exatas e da Terra

Coordenador: Prof. Dr. Cristiano André da Costa

Orientador: Prof. Dr. João Carlos Gluz

São Leopoldo

2013

Ficha catalográfica

G146a Galafassi, Fabiane Flores Penteadó
Agente pedagógico para mediação do processo de ensino-aprendizagem da dedução natural na lógica / por Fabiane Flores Penteadó Galafassi. – 2013.
127 f.: il., 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2013.

Orientação: Prof. Dr. João Carlos Gluz ; Coordenação: Prof. Dr. Cristiano André da Costa.

1. Objetos de aprendizagem. 2. Sistemas tutores inteligentes.
3. Agentes pedagógicos. 4. Estratégias de mediação. I. Título.

CDU 004:37

Catálogo na Fonte:

Bibliotecária Vanessa Borges Nunes - CRB 10/1556

FABIANE FLORES PENTEADO GALAFASSI

Agente Pedagógico para Mediação do Processo de Ensino- Aprendizagem da Dedução Natural na Lógica Proposicional

**Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada da Universidade do Vale
do Rio dos Sinos – Unisinos.**

Área de Atuação: Ciências Exatas e da Terra

Coordenador: Prof. Dr. Cristiano André da Costa

Orientador: Prof. Dr. João Carlos Gluz

Aprovado em ____ de _____ de 2013.

BANCA EXAMINADORA

Prof. Dr. João Carlos Gluz

Prof. Dr. Jorge Luiz Victória Barbosa

Prof. Dr. Ismar Frango Silveira

São Leopoldo

2013

AGRADECIMENTOS

Em especial,

Ao meu amigo e orientador: Prof. Dr. João Carlos Gluz, por ter acreditado em mim e me dado à oportunidade da realização de um sonho e pelo grande aprendizado. O meu muito obrigado. Valeu mesmo!

Ao professor Ricardo Oliveira pelo apoio, ajuda e compreensão principalmente nas primeiras etapas desta caminhada.

A FINEP pelo apoio financeiro.

Ao meu esposo pela ajuda, paciência, carinho, compreensão e companheirismo.

Aos meus pais, pelo incentivo, apoio, torcida e força.

A todos os amigos que me ajudaram nessa caminhada.

Agradecer a Deus os benefícios da vida e valorizar os recursos do próprio corpo.

Trabalhar e servir além do próprio dever, quanto lhe seja possível.

Observar, ainda mesmo por instantes, a beleza da paisagem que lhe emoldura a presença.

Nada reclamar.

Comentar unicamente os assuntos edificantes.

Refletir nas qualidades nobres de alguma pessoa com a qual os seus sentimentos ainda não se afinem.

Falar sem azedume e sem agressividade na voz.

Ler algum trecho construtivo.

Praticar, pelo menos, uma boa ação, sem contar isso à pessoa alguma.

Cultivar tolerância para com a liberdade dos outros sem atrapalhar a ninguém.

Atendamos diariamente a semelhante receita de atitude e, em breve tempo, realizaremos a conquista da paz.

André Luiz – Psicografia de Chico Xavier

Do livro: Busca e Acharás

RESUMO

A Lógica é uma ciência de índole matemática que está fortemente ligada à Filosofia, cuidando das leis do raciocínio, ou do pensar correto, sendo, portanto, um instrumento do pensar. Assim o aprendizado da lógica se faz necessário para garantir que nossos pensamentos se realizem de forma correta a fim de produzir conhecimentos verdadeiros. A lógica estuda os princípios e métodos usados para distinguir o raciocínio correto do incorreto. Desta forma, a Lógica é uma disciplina fundamental para os cursos acadêmicos de Ciência da Computação. No entanto, uma persistência de altos níveis de reprovação ou desistências prematuras mostra que há um amplo espaço para melhorar o processo de ensino desta disciplina. Nesse contexto, um tema particularmente crítico da Lógica e que causa sérias dificuldades aos alunos é a aprendizagem dos processos de dedução formal. Tendo essa questão em vista, o presente trabalho tem por objetivo propor um modelo computacional de mediação apropriado para o ensino da Dedução Natural para a Lógica Proposicional, incorporado na forma de um agente pedagógico que auxilie o aluno em seu processo de aprendizagem, servindo como ferramenta de apoio para esse processo. Este agente é parte de um projeto de pesquisa, denominado Heráclito, que visa integrar e aplicar as tecnologias de Objetos Inteligentes de Aprendizagem e Agentes Pedagógicos no ensino de Lógica.

Palavras chave: Objetos de Aprendizagem, Sistemas Tutores Inteligentes, Agentes Pedagógicos e Estratégias de Mediação.

ABSTRACT

Logic is a mathematical science of nature that is strongly linked to Philosophy, tending the laws of reasoning, or right thinking, therefore, an instrument of thought. So learning the logic is necessary to ensure that our thoughts are carried correctly to produce true knowledge. Logic studies the methods and principles used to distinguish correct from incorrect reasoning. Thus, the logic is a fundamental discipline for academic courses of Computer Science. However, the persistence of high levels of premature failure or dropouts shows that there is ample room for improving the process of teaching this subject. In this context, a particularly critical of Logic and causing severe difficulties for students is learning the processes of formal deduction. With this question in mind, this paper aims to propose a computational model of mediation appropriate for the teaching of Natural Deduction for Propositional Logic, incorporated as a pedagogical agent to assist students in their learning process, serving as a tool support for this process. This agent is part of a research project, named Heraclitus, which aims to integrate and apply the technologies of Smart Objects Learning and Pedagogical Agents in teaching Logic.

Keywords: Learning Objects, Virtual Learning Environments, Intelligent Tutoring Systems, Software Agents, Mediation Learning.

LISTA DE TABELAS

Tabela 1 – Dados combinados das turmas de Lógica – Modalidade presencial.	16
Tabela 2 – Dados das turmas de Lógica do professor X – Modalidade a distância.	18
Tabela 3 – Expressões em português e operadores lógicos.....	25
Tabela 4 – Regras Básicas de Inferência.....	26
Tabela 5 – Regras de Inferência Derivadas.....	27
Tabela 6 – Comparativo de AVAs nos padrões SCORM e IMS.....	29
Tabela 7 – Parâmetros das mensagens FIPA-ACL.....	40
Tabela 8 – Técnicas de ensino.....	47
Tabela 9 – Táticas de agentes tutores.....	48
Tabela 10 – Trabalhos sobre OAs e AVAs encontrados nos principais eventos de Computação e Informática na Educação.....	55
Tabela 11 – Análise comparativa sobre OAs e AVAs.....	57
Tabela 12 – Comparação entre os sistemas tutores de Lógica pesquisados e o sistema Heráclito.....	62
Tabela 13 – Mensagens trocadas entre os agentes.	89
Tabela 14 – Lista de Exercícios usada no Experimento do Heráclito.....	96
Tabela 15 – Total de alunos aprovados/reprovados com e sem o uso do Heráclito.....	98
Tabela 16 – Total de alunos que usaram/não usaram e aprovaram/reprovaram na disciplina.	98
Tabela 17 – Distribuição de exercícios realizados por alunos no experimento.....	101
Tabela 18 – Sumário dos resultados do quesito efetividade pedagógica da mediação.....	101
Tabela 19 – Sumário dos resultados do quesito qualidade do conteúdo apresentado.....	103
Tabela 20 – Sumário dos resultados do quesito <i>Layout</i> Visual.....	104
Tabela 21 – Sumário dos resultados do quesito usabilidade do editor de provas.....	104

LISTA DE FIGURAS

Figura 1 – Evolução das taxas de aprovação e reprovação, modalidade presencial.	17
Figura 2 – Evolução das taxas de aprovação e reprovação, modalidade EAD.	18
Figura 3 – Processo de estímulo-resposta mediado.	22
Figura 4 – Agentes interagem com ambientes por meio de sensores e atuadores.	33
Figura 5 – Conceito básico do sistema de troca de mensagens.	34
Figura 6 – Ciclo de Raciocínio do Jason.	37
Figura 7 – Modelo de referencia para uma plataforma de agentes FIPA.	39
Figura 8 – Estrutura de mensagens FIPA.	39
Figura 9 – Organização Geral da Infraestrutura MILOS.	43
Figura 11– Caso de uso do Sistema Heráclito.	66
Figura 12 – Arquitetura do Sistema Heráclito.	67
Figura 13 – Integração do Heráclito na MILOS.	69
Figura 14 – Tela inicial do Sistema Heráclito.	70
Figura 15 – Tela de Login do Heráclito.	72
Figura 16 – Tela de edição do Sistema Heráclito.	73
Figura 17 – Tela de edição do Sistema Heráclito selecionando um argumento.	73
Figura 18 – Tela de edição do Sistema Heráclito adicionando hipóteses.	74
Figura 19 – Tela de edição do Sistema Heráclito aplicando a regra Modus Ponens.	75
Figura 20 – Interface da versão móvel do sistema Heráclito.	76
Figura 21 – Interface com mensagem crítica.	77
Figura 22 – Exemplo de uso de regra na versão móvel – Seleção da regra.	78
Figura 23– Exemplo de uso de regra na versão móvel – Aplicação da regra.	78
Figura 24 – Finalização do processo do uso e aplicação das regras.	79
Figura 25 – Diagrama de atividades para o diagnóstico de um novo argumento.	82
Figura 26 – Diagrama de atividades de acompanhamento de ajuda.	83
Figura 27 – Diagrama de atividades de Acompanhamento de Prova.	84
Figura 28 – Diagrama de atividades de Ação Pedagógica de uma Prova.	86
Figura 29 – Diagrama de atividades de Acompanhamento de aluno ocioso.	87
Figura 30 – Diagrama de atividades de Acompanhamento de Status da Prova.	88
Figura 31 – Troca de mensagens entre os agentes quando o aluno inicia um novo argumento	89
Figura 32 – Troca de mensagens entre os agentes em aplicação de uma regra.	90
Figura 33 – Troca de mensagens entre os agentes quando o aluno está ocioso.	91
Figura 34 – Troca de mensagens entre os agentes quando o aluno é informado do status de sua prova.	91
Figura 35 – Exemplo de plano para um novo argumento.	91
Figura 36 – Exemplo de plano para uma aplicação de regra.	92
Figura 37 – Exemplo de plano para aplicação de uma regra prejudicial.	92
Figura 38 – Exemplo de plano para a reaplicação de uma regra redundante.	93
Figura 39 – Exemplo de plano para aplicação de regras redundantes em sequência.	93
Figura 40 – Exemplo de plano para obtenção de uma regra a ser aplicada.	94
Figura 41 – Exemplo de plano para obtenção de uma regra a ser aplicada.	94
Figura 42 – Exemplo de plano para informar o aluno qual regra deve ser aplicada.	94
Figura 43 – Nível de complexidade dos exercícios em %	100

LISTA DE SIGLAS

AAMAS	International Conference on Autonomous Agents and Multiagent Systems
ACL	Agent Communication Language
ADL	Advanced Distributed Learning initiative
AMS	Agent Management System
API	Applications Programmers Interface
ARIADNE	Alliance of Remote Institute of Electrical and Distribution Networks for Europe
AVA	Ambientes Virtuais de Aprendizagem
BDI	Belief – Desire – Intention
CSBC	Congresso da Sociedade Brasileira de Computação
DCMI	Dublin Core Metadata Initiative
DF	Directory Facilitator
FIPA	Foundation for Intelligent Physical Agents
HTTP	HyperText Transport Protocol
IEEE	Institute of Electrical and Electronics Engineers
IEEE-LOM	IEEE Learning Object Metadata
IEEE-LTSC	IEEE Learning Technology Standard Comitee
IES	Instituição de Ensino Superior
IMS	IMS Global Learning Consortium
ITS	Intelligent Tutoring Systems
JADE	Plataforma de Agentes que suporta o padrão FIPA
JASON	Ambiente de Desenvolvimento de SMA para a linguagem AgentSpeak(L)
LACLO	Comunidade Latino-americana de Objetos de Aprendizagem
LOM	Learning Object Metadata
LTSC	Learning Technology Standard Comitee
LGPL	Lesser General Public License
MAS	Multiagent System

MEC	Ministério da Educação e Cultura
MILOS	Multiagent Infrastructure for Learning Object Support
OA	Objetos de Aprendizagem
OBAA	Objetos de aprendizagem Baseados em Agentes Artificiais
OMG	Object Management Group
OWL	Web Ontology Language
RBIE	Revista Brasileira de Informática na Educação
RDF	Resource Description Framework
RENOTE	Revista Novas Tecnologias na Educação
SBC	Sociedade Brasileira de Computação
SBIE	Simpósio Brasileiro de Informática na Educação
SCORM	Sharable Content Object Reference Model
SMA	Sistema Multiagente
TICs	Tecnologias de Informação e Comunicação (TICs)
TISE	Taller Internacional de Software Educativo
TV	Televisão
UML	Unified Modelling Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO.....	13
2.1	Justificativa.....	15
2.2	Apresentação	18
2.3	Objetivos.....	19
2.4	Metodologia de desenvolvimento do Sistema Heráclito	20
3	FUNDAMENTAÇÃO DA PESQUISA.....	21
3.1	Processo sócio histórico: A teoria da Mediação de Vygotsky.....	21
3.1.1	O uso de instrumentos e signos	22
3.1.2	Zona de Desenvolvimento Proximal (ZDP)	23
3.1.3	Aprendizado, desenvolvimento e ensino	23
3.2	Dedução Natural na Lógica Proposicional (DNLP).....	25
3.2.1	Regras de Dedução Natural	26
3.3	Ambientes Virtuais de Aprendizagem.....	28
3.4	Objetos de Aprendizagem	30
3.5	Agentes de Software e Sistemas Multiagente.....	32
3.6	Arquitetura BDI.....	34
3.7	A linguagem AgentSpeak(L) e a Plataforma Jason	35
3.8	O Padrão FIPA e a Plataforma JADE.....	38
3.9	Agentes Pedagógicos	41
3.10	A Infraestrutura MILOS	42
3.11	Ontologias no Sistema Heráclito	43
3.12	Estratégias e Táticas Pedagógicas	46
3.13	Estratégias e Táticas para Ensino de Lógica	49
3.14	Planos Gerenciais de Ensino Didático aplicados no Ensino de Lógica.....	50
4.	TRABALHOS RELACIONADOS	54
4.1.	O Contexto Geral da Pesquisa.....	54
4.2.	Ferramentas de Apoio ao Ensino de Lógica.....	60
5	O SISTEMA HERÁCLITO	66
5.1	Diagrama de Caso de Uso do Sistema Heráclito.....	66
5.2	Arquitetura do Sistema Heráclito	66
5.3	Editor de provas do Sistema Heráclito – Versão Desktop.....	69
5.3.1	Interface com o Usuário	70
5.3.2	O Login do usuário	71
5.3.3	A Tutoria no Heráclito.....	72

5.4	Uso do Editor de Provas – Versão Desktop	72
5.5	Editor de Provas em Modo Local	75
5.6	Editor de Provas do Sistema Heráclito – Versão Móvel	76
5.6.1	Interface Móvel.....	76
5.6.2	Testes	79
6	AGENTE MEDIADOR.....	80
6.1	Apoio Pedagógico do Agente Mediador	80
6.2	Estratégias de Mediação	81
6.3	Comunicação entre os Agentes	88
6.4	Desenvolvimento do Protótipo do Agente Mediador	91
7	EXPERIMENTOS E VALIDAÇÃO	95
7.1.	Objetivos do Experimento	95
7.2.	Avaliação do Experimento	96
7.3.	Condução do Experimento	97
7.4.	Resultados e Análises do Experimento	98
7.5	Resultados e Análise da Avaliação do Experimento	101
8	CONCLUSÕES	107
	REFERENCIAS	111
	ANEXO I.....	116
	ANEXO II	121
	ANEXO III.....	128
	ANEXO IV	129

1 INTRODUÇÃO

A presente dissertação é uma dissertação de Computação Aplicada e, como tal, tem seus esforços de pesquisa centrados tanto em tecnologias computacionais, quanto na utilização destas tecnologias em uma área de aplicação em particular.

Assim, esta dissertação foi organizada em dois grandes eixos de pesquisa: Pedagógico e Tecnológico. O eixo Pedagógico está relacionado à área de aplicação, onde busca identificar áreas de conhecimentos ainda pouco exploradas, buscando métodos alternativos de ensino para essas áreas, oferecendo um desafio importante para o trabalho de pesquisa. O segundo eixo tem relação com as tecnologias computacionais utilizadas para tratar das questões de pesquisa derivadas da área de aplicação.

Neste trabalho o foco de aplicação está voltado ao estudo da disciplina de Lógica. Esta disciplina é considerada uma disciplina básica e obrigatória para todos os cursos que abrangem computação e informática, de acordo com o currículo oficial do MEC. Nesse currículo, a disciplina de lógica é uma disciplina que deve ser oferecida nos semestres iniciais, para incentivar o pensamento lógico-formal e as habilidades de resolução de problemas simbólicos.

Com relação ao eixo Tecnológico, a presente pesquisa se concentra nas Tecnologias de Informação e Comunicação (TIC) aplicadas na educação, particularmente àquelas focadas no suporte à relação professor-tutor-aluno e que exploram as possibilidades tecnológicas que o uso de Objetos de Aprendizagem (OA), Ambientes Virtuais de Aprendizagem (AVA), Agentes e Ontologias, podem oferecer ao processo de ensino-aprendizagem¹.

Neste contexto, os OA se apresentam como possibilidades de potencializar o processo de ensino-aprendizagem de forma significativa com relação aos conteúdos propostos. Os OA despontam na educação presencial e na EAD como uma tecnologia que apesar de não ser recente, está em constante evolução, beneficiando assim professores e alunos, nos diversos contextos da aprendizagem. Já AVA por sua vez, são sistemas integrados e abrangentes capazes de promover o engajamento do aluno, onde sua interação com o ambiente e o conteúdo proposto é fundamental para que possam organizar suas ideias, compartilhar seus conhecimentos tornando-se sujeitos autônomos de sua aprendizagem.

O trabalho está organizado no que segue: o capítulo 2 contextualiza o problema de pesquisa, os objetivos e a metodologia empregado no projeto. O capítulo 3 apresenta a fundamentação teórica, juntamente com as bases tecnológicas que foram utilizadas no desenvolvimento do sistema computacional (denominado de Sistema Heráclito), que será o alvo da pesquisa tecnológica da dissertação. O capítulo 4 apresenta um estudo dos trabalhos relacionados com o tema de pesquisa. O capítulo 5 contextualiza o sistema Heráclito e seu editor de provas. O capítulo 6 aborda detalhes das estratégias de mediação desenvolvidas para o agente Mediador do Sistema Heráclito, bem como a interação, comunicação e arquitetura do

¹ Para Vygotsky, justamente por sua ênfase nos processos sócios históricos, a ideia de aprendizado inclui a interdependência dos indivíduos envolvidos no processo. O termo ensino-aprendizagem é a tradução da palavra russa *obuchenie*, e significa algo como “processo de ensino-aprendizagem”, incluindo sempre aquele que aprende aquele que ensina e a relação entre estas pessoas. Pela falta de um termo equivalente em inglês, a palavra *obuchenie* tem sido traduzida ora como ensino, ora como aprendizagem e assim re-traduzida para o português. O conceito para Vygotsky tem um significado mais abrangente sempre envolvendo interação social [OLIVEIRA, 1999, p. 57].

agente tutor mediador. O capítulo 7 apresenta o planejamento e os resultados do experimento realizado com o sistema Heráclito e o capítulo 8 apresentam as conclusões e trabalhos futuros deste trabalho.

2 QUESTÃO DE PESQUISA

Este capítulo apresenta uma visão geral sobre as questões que motivaram esta pesquisa, mostrando os objetivos a serem alcançados, bem como a metodologia a ser utilizada para a solução do problema.

2.1 Justificativa

Atualmente, o uso das Tecnologias de Informação e Comunicação (TICs) vem provocando alterações nas relações entre as competências intelectuais e o conhecimento, oportunizando mudanças de paradigmas em situações de aprendizagem. Isso afeta diretamente as relações de ensino-aprendizagem e as modalidades de educação. Neste contexto, os Objetos de Aprendizagem (OAs) surgem como possibilidades de potencializar o processo de ensino-aprendizagem de forma mais significativa com relação aos conteúdos propostos. Os OAs despontam na educação presencial e na educação a distância como uma tecnologia que apesar de não ser recente, está em constante evolução, beneficiando assim professores e alunos, nos diversos ambientes de aprendizagem e seus contextos.

Nesse sentido o presente trabalho traz como foco de estudo a disciplina de Lógica. Esta disciplina é considerada uma disciplina básica e compreende todos os cursos que abrangem computação e informática. Esta disciplina é fundamental para a formação dos alunos, ajudando-os no desenvolvimento de suas habilidades de análise lógica, formalização e resolução de problemas. Estas habilidades, por sua vez, são necessárias para as atividades diversas como: programação, análise e especificação de requisitos, projeto de bancos de dados, dentre outras áreas da computação e da informática.

Dados estatísticos de várias turmas de Lógica, além de dados empíricos obtidos através de oito anos de experiência docente ministrando Lógica em duas universidades distintas, apontam para índices muito altos de reprovação e desistência nesta disciplina. Dados compilados em uma universidade brasileira, durante o período de 2007 ao primeiro semestre de 2012, apontam para uma média de aprovação de apenas 56% na disciplina de Lógica. Dos alunos restantes, 24% foram considerados desistentes, por terem cancelado a disciplina ou terem excesso de faltas, enquanto 20% assistiram a disciplina até o final, mas foram reprovados. Essa questão se torna ainda mais complexa nos cursos ministrados na modalidade à distância (modalidade EAD). Dados preliminares compilados durante o ano de 2011 apontam para índices de desistência e reprovação superiores: apenas 37% de aprovação na disciplina de Lógica, com 11% desistentes e 52% reprovados.

As desistências, em particular, tendem a ocorrer no início da disciplina, principalmente quando os conteúdos de Dedução Natural começam a ser ensinados no contexto da Lógica Proposicional. Na prática, as dificuldades começam quando conceitos como fórmula, regra de dedução e prova formal começam a ser ensinados.

A questão de pesquisa que o Projeto Heráclito aborda é um problema de Ensino Superior relacionado à disciplina de Lógica que atualmente forma parte do currículo definido pela SBC e MEC para os cursos de Bacharelado e Licenciatura em

Computação, sendo ministrada tipicamente nos primeiro ou segundo semestre destes cursos.

O problema está relacionado aos altos índices de reprovação e desistência que ocorrem nesta disciplina, tanto no caso de ensino presencial, quanto em ensino à distância (EAD). Estes índices foram detectados em um levantamento de dados feito durante o período que compreende o primeiro semestre de 2007 ao segundo semestre de 2012, com base nas turmas de Lógica de dois professores distintos de uma universidade brasileira. O estudo considerou dados obtidos nas modalidades presencial e a distância. A Tabela 1 mostra os dados combinados de ambos os professores desta universidade no período de 2007/1 até 2012/1, trazendo o número de alunos aprovados, cancelados, reprovados e sem frequência para cada semestre analisado.

Tabela 1 – Dados combinados das turmas de Lógica – Modalidade presencial.

Semestre	Turmas	Aprovados	Cancelados	Reprovados	Sem Frequência	TOTAL DE ALUNOS
2007/1	5	69	23	60	28	180
2007/2	5	93	15	43	26	177
2008/1	6	121	27	62	51	261
2008/2	6	123	24	24	29	200
2009/1	6	172	30	25	50	277
2009/2	5	99	16	24	28	167
2010/1	9	161	31	73	31	296
2010/2	4	98	20	27	27	172
2011/1	5	135	28	29	26	218
2011/2	7	102	18	24	41	185
2012/1	5	144	10	49	0	203
TOTAL	63	1317	242	440	337	2336

Com base nessa Tabela, foram extraídas as médias de alunos Aprovados e Cancelados / Reprovados / Sem Frequência, as quais são mostradas em um gráfico a fim de analisar a variação entre os semestres. A Figura 1 mostra a evolução das médias supracitadas para os semestres de 2007/1 a 2012/1.

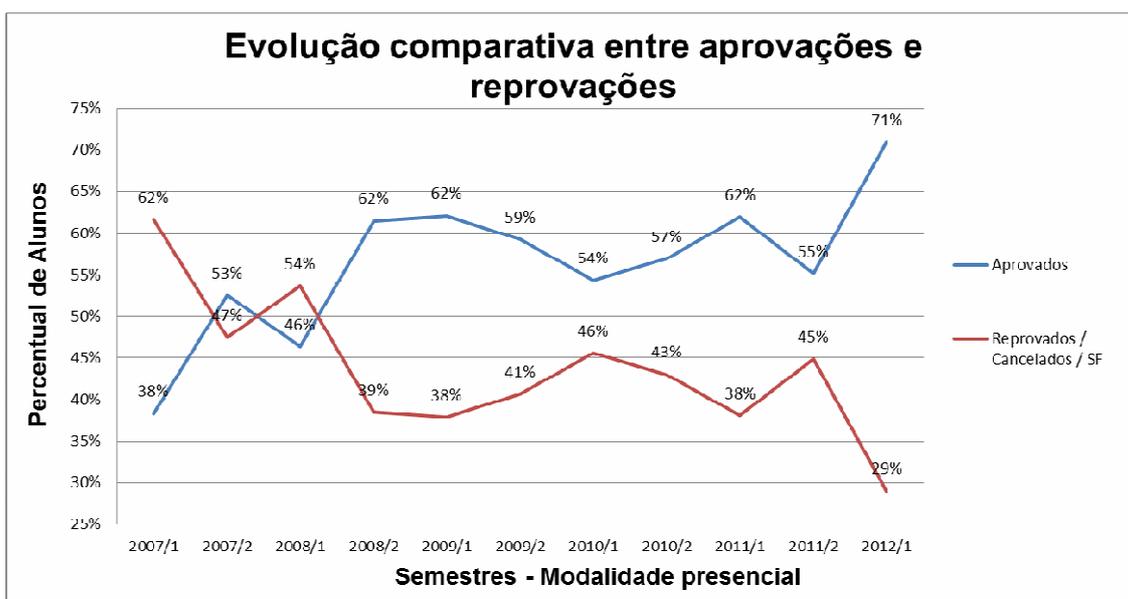


Figura 1 – Evolução das taxas de aprovação e reprovação, modalidade presencial.

Apesar de 2012/1 ter sido um semestre atípico, onde 71% dos alunos foram aprovados, a média de aprovação dos outros semestres se mostra muito inferiores, como, por exemplo, 55% em 2011/2.

Embora os dados coletados estejam restritos a uma universidade, existe uma expectativa que este caso seja típico e possa ser generalizado para outras instituições de ensino superior. A experiência prévia de um dos professores ministrando a mesma disciplina em outra universidade, durante um período de quatro anos, indica um quadro similar, apesar destes dados estatísticos não terem sido coletados.

Apesar de flutuações anuais, pode-se claramente perceber os altos índices de reprovação e ainda mais altos índices de desistências por parte dos alunos. Esta tendência não é um artifício provocado apenas por um dos professores.

Na modalidade EAD esta situação se torna ainda mais crítica. Dados nesta modalidade, compilados de 17 turmas ministradas pelo professor X no período de 2010/2 a 2011/2 com um total de 276 alunos apontam para índices de desistência e reprovação mais altos do que os vistos na modalidade presencial: apenas 37% de aprovação, com 11% reprovados e 52% desistentes. A análise acima indica que, mesmo no caso presencial, já existe uma necessidade de melhoria do quadro de desistências e reprovações. No caso da modalidade de EAD a situação é ainda mais complicada. Olhando para estes resultados acredita-se que sem um apoio mais concreto para o ensino mediado por computador para esta disciplina, a tendência da situação é manter-se ou até mesmo vir a apresentar um quadro pior.

Na Tabela 2 é possível visualizar o número de alunos aprovados, cancelados, reprovados e sem frequência nos semestre de 2010/2, 2011/1 e 2011/2.

Tabela 2 – Dados das turmas de Lógica do professor X – Modalidade a distância.

Semestre	Turmas	Aprovados	Cancelados	Reprovados	Sem Frequência	TOTAL DE ALUNOS
2010/2	4	16	2	31	0	49
2011/1	9	30	22	77	0	129
2011/2	7	57	5	36	0	98
TOTAL	20	103	29	144	0	276

Com base nessa Tabela, foram extraídas as médias de alunos Aprovados e Cancelados / Reprovados / Sem Frequência, as quais são mostradas em um gráfico a fim de analisar a variação entre os semestres. A Figura 2 mostra a evolução das médias supracitadas para os semestres de 2007/1 a 2012/1.

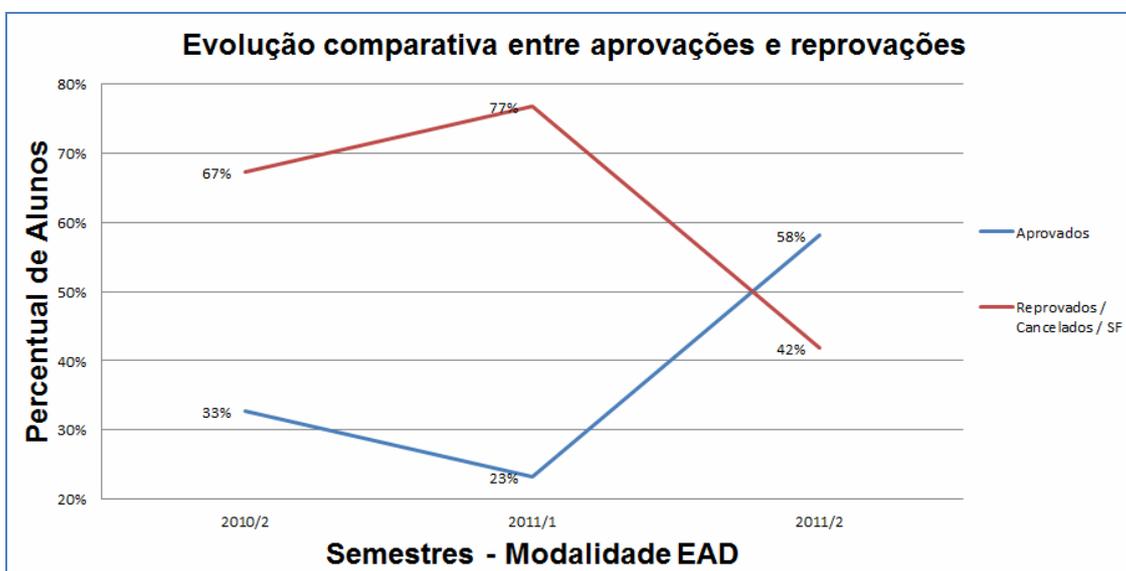


Figura 2 – Evolução das taxas de aprovação e reprovação, modalidade EAD.

Com base nestes dados, a expectativa é que o Sistema Heráclito possa contribuir com a eventual melhoria deste quadro de desistências e reprovações, estando particularmente focado no suporte à relação professor/tutor/aluno no contexto das modalidades presencial e EAD, justamente no ensino da Dedução Natural na Lógica Proposicional, ao menos em relação às desistências.

2.2 Apresentação

O contexto que esta pesquisa aborda é um problema de Ensino Superior relacionado à disciplina de Lógica que atualmente forma parte do currículo definido pela SBC e MEC para os cursos de Bacharelado e Licenciatura em Computação [MEC - DIRETRIZES CURRICULARES NACIONAIS, 2012], sendo ministrada tipicamente no primeiro ou segundo semestre destes cursos.

O problema mencionado está relacionado aos altos índices de reprovação e desistência que ocorrem nesta disciplina, tanto no caso de ensino presencial, quanto em ensino à distância.

Em termos pedagógicos, pretende-se avaliar as possibilidades que um método dialético² de ensino, embasado em uma pedagogia sócio histórica, pode oferecer no contexto de ensino da Dedução Natural de Lógica Proposicional.

Em termos técnicos, pretende-se verificar se a mediação por computador, usando as tecnologias combinadas de OA e AVA, com suporte de agentes pedagógicos e ontologias, pode contribuir para diminuir esses índices.

Assim o problema central da pesquisa pode ser resumido como:

Verificar se é possível combinar efetivamente um método dialético e sócio histórico de ensino e um modelo de mediação por computador em um agente pedagógico de mediação capaz de auxiliar o processo de ensino-aprendizagem da Dedução Natural na Lógica Proposicional.

2.3 Objetivos

A questão de pesquisa abordada pelo presente trabalho está inserida no contexto de um projeto maior de pesquisa, o projeto Heráclito, que visa criar um sistema de tutoria de ensino de Lógica que possa apoiar tanto os alunos quanto os professores nas atividades didático-pedagógicas dessa disciplina.

Assim, os objetivos de pesquisa específicos da presente dissertação são os seguintes:

- Definir um modelo de mediação dialético e sócio histórico para o processo de ensino-aprendizagem da Dedução Natural na Lógica Proposicional.
- Especificar detalhadamente as propriedades do agente pedagógico que será responsável pelos processos dialéticos de mediação.
- Modelar o comportamento deste agente e implementar um protótipo do mesmo.
- Especificar e implementar a comunicação entre os agentes.
- Projetar, desenvolver e testar uma facilidade de interface para o usuário, baseada em OAs, que seja apropriada para o ensino de Dedução Natural na Lógica Proposicional.

² A perspectiva dialética baseia-se numa concepção diferente da abordada tradicionalmente sobre homem e conhecimento. Esta perspectiva entende o homem como um ser ativo e de relações. Assim, por conseguinte entende que o conhecimento não é transferido ou depositado pelo outro (conforme a concepção tradicional), nem é inventado pelo sujeito (tratado na concepção espontaneísta), mas sim que o conhecimento é construído pelo sujeito na sua relação com os outros e com o mundo [VASCONCELLOS, 1992].

- Criar um OA capaz de oferecer apoio na elaboração de demonstrações formais sob as regras da Dedução Natural na Lógica Proposicional;
- Planejar e conduzir testes empíricos em sala de aula.
- Empreender uma análise crítica destes resultados.

2.4 Metodologia de desenvolvimento do Sistema Heráclito

Em termos gerais a metodologia utilizada neste trabalho segue as diretrizes técnicas e metodológicas adotadas na Infraestrutura MILOS (detalhada na seção 3.7 deste trabalho) que prevê o uso de camadas de ontologias, agentes pedagógicos e facilidades de interface para o usuário.

Em termos pedagógicos, a metodologia a ser adotada na didática e pedagogia dos OAs criada para o sistema de suporte a objetos OBAA será baseada no método dialético de ensino, embasado em uma pedagogia sócia histórica, no contexto do aprendizado de Lógica. Os conceitos sócio históricos utilizados neste trabalho são embasados nos referidos trabalhos publicados nos SBIE do ano de 2007 [GLUZ; PASSERINO; VICARI, 2007] e 2008 [GLUZ; PASSERINO; VICARI, 2008].

O sistema computacional de ensino de Lógica projetado e desenvolvido no decorrer desta dissertação, foi denominado de Heráclito. O projeto do Sistema Heráclito está fundamentado em um ciclo de projeto baseado em protótipos.

O primeiro protótipo (OA) teve como objetivo definir e testar as propriedades da interface de usuário do Heráclito, o qual se constituiu de um editor de provas no contexto de Lógica, com características interativas, capaz de oferecer apoio na elaboração de demonstrações formais sob as regras da Dedução Natural na Lógica Proposicional. Isso também incluiu não apenas conteúdos didáticos de Lógica, mas também todo o ambiente de execução necessário para que este editor de provas possa ser utilizado pelos alunos em seus computadores.

Posteriormente a isso foi desenvolvido um sistema multiagente composto por um conjunto de agentes pedagógicos (perfil do usuário, mediador e especialista), responsáveis pela interação do aluno com o sistema, além da definição da forma de comunicação entre esses agentes e o editor de provas do Heráclito.

Estes agentes foram desenvolvidos na linguagem Java e na linguagem AgentSpeak(L), com suporte da plataforma JADE para comunicação entre eles. Estas linguagens e plataforma foram escolhidas por sua versatilidade, eficiência, portabilidade e segurança. Mais informações sobre as linguagens de programação e a plataforma JADE são apresentadas nas seções 3.5 e 3.6.

3 FUNDAMENTAÇÃO DA PESQUISA

Este capítulo apresenta as bases tecnológicas e o embasamento metodológico da pedagogia sócio histórica como método de ensino para a criação do Sistema Heráclito. Este sistema depende principalmente das tecnologias relacionadas aos objetos de aprendizagem, incluindo a definição de padrões de metadados destes objetos. Também são tecnologias importantes para o projeto, os agentes pedagógicos e a infraestrutura de agentes MILOS, sendo construída para suportar os requisitos e funcionalidades dos OAs compatíveis com o padrão OBAA.

3.1 Processo sócio histórico: A teoria da Mediação de Vygotsky

A teoria sócio-histórico-cultural de Vygotsky tem sua abordagem sustentada nas seguintes dimensões:

- Social: é fundamental para marcar a questão das relações humanas;
- Histórica: trata dos diversos momentos e épocas que marcaram a humanidade e em como o tempo influenciou ser humano;
- Cultural: útil para vermos e valorizarmos toda a produção humana em diferentes contextos.

O objetivo de uma abordagem sociocultural é explicar as relações entre a ação humana, de um lado, e as situações culturais, institucionais e históricas em que essa ocorre, de outro [BERNI, 2013].

Vygotsky dedicou-se, principalmente, ao estudo daquilo que chamamos de funções psicológicas superiores (desenvolvimento cognitivo) ou processos mentais (pensamento, linguagem, comportamento volitivo, atenção consciente, memória), ou seja, aqueles mecanismos psicológicos mais sofisticados, mais complexos. Tais funções envolvem o controle consciente do comportamento, a ação intencional e a liberdade do indivíduo em relação às características do momento e do espaço presentes (fazendo referencia ao meio social) [OLIVEIRA, 1999].

Um conceito central para a compreensão das concepções vygotskianas sobre o funcionamento psicológico é o conceito de mediação. Mediação, em termos genéricos, é o processo de intervenção de um elemento intermediário numa relação, onde esta, deixa, então, de ser direta e passa a ser mediada por este elemento e pode ser vista na Figura 3 abaixo:

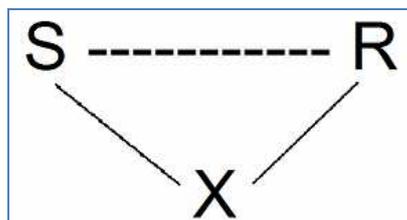


Figura 3 – Processo de estímulo-resposta mediado.
Fonte: [OLIVEIRA, 1999].

A relação deste processo simples estímulo-resposta é substituída por um ato complexo e pode ser representado da seguinte forma: S= Estímulo/R= resposta e X= elo intermediário ou elemento mediador (como mostra a Figura 3). Um exemplo de estímulo-resposta mediado pode ser observado quando um indivíduo aproxima sua mão da chama de uma vela e a retira rapidamente ao sentir dor. Nesse processo está estabelecida uma relação direta entre o calor da chama e a retirada da mão. Se, no entanto, o indivíduo retirar a mão quando apenas sentir o calor e lembrar-se da dor sentida em outra ocasião, a relação entre a chama da vela e a retirada da mão estará mediada pela lembrança da experiência anterior. Nesse novo processo o impulso direto para reagir é inibido, e é incorporado um estímulo auxiliar que facilita a complementação da operação por meios indiretos. No exemplo da vela, o estímulo (S) seria o calor da chama e a resposta (R) seria a retirada da mão. A lembrança da dor ou o aviso de outra pessoa sobre o risco da queimadura seriam elementos mediadores, intermediários entre o estímulo e a resposta. A presença de elementos mediadores introduz um elo a mais nas relações organismo/meio, tornando-as mais complexas e ao longo do desenvolvimento do indivíduo as relações mediadas passam a predominar sobre as relações diretas [OLIVEIRA, 1999]. Desta forma, entende-se que é pela mediação que se dá a internalização (reconstrução interna de uma operação externa) de atividades e comportamentos sócio históricos e culturais, típicos do domínio humano [MOREIRA, 1999].

Nesse sentido, Vygotsky trabalha com a noção de que a relação do homem com o mundo não é uma relação direta, mas, fundamentalmente, uma relação mediada [OLIVEIRA, 1999]. Ao destacar a importância das interações sociais, traz a ideia da mediação e da internalização como aspectos fundamentais para a aprendizagem, defendendo que a construção do conhecimento ocorre a partir de um intenso processo de interação entre os sujeitos e o meio social [TASSONI, 2013].

Segundo Driscoll, 1995, p. 229 *apud* Moreira [1999] “Não é por meio do desenvolvimento cognitivo que o indivíduo se torna capaz de socializar, é na socialização que se dá o desenvolvimento dos processos mentais superiores.

Vygotsky distinguiu dois tipos de elementos mediadores: os instrumentos e os signos.

3.1.1 O uso de instrumentos e signos

Vygotsky busca compreender as características do homem através do estudo da origem e desenvolvimento da espécie humana, tornando o surgimento do trabalho e a formação da sociedade humana, com base no trabalho, como sendo o processo básico que vai marcar o homem como espécie diferenciada. É o trabalho que, pela ação transformadora do homem sobre a natureza, que acaba unindo homem e natureza, criando a cultura e a história humanas. É no trabalho que se desenvolve a atividade coletiva e, portanto, ações sociais e também a criação e utilização de instrumentos. O instrumento é um elemento interposto entre o trabalhador e o objeto de trabalho, ampliando as possibilidades de transformação da natureza [MOREIRA, 1999].

A invenção e o uso de signos como meios auxiliares para solucionar um dado problema psicológico (lembrar, comparar, relatar, escolher, etc), é análoga a invenção de instrumentos. O signo age como um instrumento da atividade psicológica de maneira análoga ao papel de um instrumento de trabalho. Os instrumentos, porém, são elementos externos ao indivíduo, voltados para fora dele. Sua função é provocar mudanças nos objetos, controlar processos da natureza. Os signos por sua vez são orientados para o próprio sujeito, para dentro do indivíduo. São ferramentas que auxiliam nos processos psicológicos e não nas ações concretas, como os instrumentos [OLIVEIRA, 1999]. Para Vygotsky, instrumentos e signos são construções sócio-históricas e culturais, e é com a interiorização de instrumentos e sistemas de signos (internalização) produzidos culturalmente, via interação social que o sujeito se desenvolve cognitivamente [MOREIRA, 1999].

3.1.2 Zona de Desenvolvimento Proximal (ZDP)

Mais formalmente, a zona de desenvolvimento proximal³ é definida por Vygotsky como a distância entre o nível de desenvolvimento cognitivo real do indivíduo, tal como medido por sua capacidade de resolver problemas independentemente, e o seu nível de desenvolvimento potencial, tal como medido através da solução de problemas sob orientação (de um adulto, no caso de uma criança) ou em colaboração com companheiros mais capazes [Vygotsky, 1988, p. 97 *apud* Moreira, 1999], ou seja, é a partir da existência desses dois níveis de desenvolvimento – real e potencial – que Vygotsky define a ZDP.

A ZDP refere-se assim, ao caminho que o indivíduo vai percorrer para desenvolver funções consolidadas, estabelecidas no seu nível de desenvolvimento real. Definindo funções que ainda não amadureceram, mas que estão no processo de maturação. É uma medida do potencial de aprendizagem e representa a região na qual o desenvolvimento cognitivo ocorre. É dinâmica e está em constantemente mudando, ou seja, é um domínio psicológico em constante transformação: aquilo que uma criança é capaz de fazer com a ajuda de alguém hoje, ela conseguirá fazer sozinha amanhã.

A interação social que provoca a aprendizagem deve ocorrer dentro da ZDP, mas ao mesmo tempo, tem um papel importante na determinação dos limites dessa zona, ou seja, o limite inferior é fixado pelo nível real de desenvolvimento do aprendiz, já o superior é determinado por processos instrucionais que podem ocorrer no brincar, no ensino formal ou informal ou no trabalho. Independentemente do contexto, o importante é a interação social [Driscoll, 1995, p. 233 *apud* Moreira, 1999].

3.1.3 Aprendizado, desenvolvimento e ensino

Desde o momento em que o desenvolvimento das funções mentais superiores exige a internalização de instrumentos e signos em contextos de interação, a aprendizagem se converte em condição para o desenvolvimento dessas funções, desde que se situe precisamente na ZPD. Na perspectiva teórica de Vygotsky, a aprendizagem é que é necessária para o desenvolvimento.

³ A expressão “zona de desenvolvimento proximal” aparece, às vezes, nas traduções para a língua portuguesa, como “zona de desenvolvimento potencial” [OLIVEIRA, 1999, p. 58].

Na concepção de Vygotsky sobre as relações entre desenvolvimento e aprendizado, e particularmente sobre a ZDP, estabelece-se forte ligação entre o processo de desenvolvimento e a relação do indivíduo com seu ambiente sócio cultural e com sua situação de organismo que não se desenvolve plenamente sem o suporte de outros indivíduos de sua espécie. É na ZDP que a interferência de outros indivíduos é a mais transformadora.

Para Vygotsky, o único bom ensino é aquele que está à frente do desenvolvimento cognitivo e o dirige. A única boa aprendizagem é aquela que está avançada em relação ao desenvolvimento. A implicação dessa concepção de Vygotsky para, como por exemplo, o ensino escolar torna-se imediata. Se o aprendizado impulsiona o desenvolvimento, então a escola tem um papel essencial na construção do ser psicológico adulto dos indivíduos que vivem em sociedades escolarizadas. Mas o desempenho deste papel só se dará adequadamente quando, conhecendo o nível de desenvolvimento dos alunos, a escola dirigir o ensino não para etapas intelectuais já alcançadas, mas sim para estágios de desenvolvimento ainda não incorporados pelos alunos, funcionando realmente como um motor de novas conquistas psicológicas. Para a criança que frequenta a escola, o aprendizado escolar é o elemento central no seu desenvolvimento.

Como na escola o aprendizado é um resultado desejável e o professor é o participante que já internalizou significados socialmente compartilhados para materiais educativos do currículo, é na interação social que deve caracterizar-se o ensino. O professor tem o papel explícito de interferir na ZDP dos alunos, provocando avanços que não ocorreriam espontaneamente. Os procedimentos regulares que ocorrem na escola – demonstração, assistência, fornecimento de pistas, instruções, - são fundamentais na promoção do bom ensino ao qual Vygotsky se referiu. A criança não tem condições de percorrer sozinha o caminho do aprendizado, mas com a intervenção de outras pessoas – que neste caso específico aplica-se a escola, ao professor – torna-se fundamental para a promoção do desenvolvimento do indivíduo.

Em um episódio de ensino, o professor, de alguma maneira, apresenta ao aluno significados socialmente aceitos, no contexto de matéria de ensino, para determinado signo – da Física, da Matemática, da Língua Portuguesa. O aluno deve, então, de alguma maneira, devolver ao professor o significado que captou. O professor, nesse processo, é responsável por verificar se o significado que o aluno captou é aceito, compartilhando socialmente. A responsabilidade do aluno é verificar se os significados que captou são aqueles que o professor pretendia que ele captasse e se são aqueles compartilhados no contexto da área de conhecimento em questão. O ensino se consuma quando um aluno e o professor compartilham significados.

Esse intercambio é fundamental para a aprendizagem e conseqüentemente, na ótica de Vygotsky, para o desenvolvimento cognitivo. Sem interação social, ou sem intercambio de significados, dentro da ZDP do aprendiz, não há ensino, não há aprendizagem e não há desenvolvimento cognitivo. Interação e intercambio implicam, necessariamente, que todos os envolvidos no processo ensino-aprendizagem devam falar e tenham a oportunidade de falar.

3.2 Dedução Natural na Lógica Proposicional (DNLP)

Dedução natural é um dos sistemas dedutivos utilizados para construir demonstrações formais na Lógica e uma proposição lógica (ou apenas uma proposição) é uma frase ou sentença da língua portuguesa (ou de qualquer outro idioma) que pode assumir apenas um de dois valores verdade: ou a frase é verdadeira (ela diz uma verdade) ou ela é falsa (diz uma falsidade) [GLUZ e PY, 2010].

Para que ocorra uma derivação formal, é necessário formalizar a expressão que queremos demonstrar. Formalizar significa traduzir da forma linguística usual para uma notação lógica, ou seja, uma forma que é entendível para qualquer um, independente da língua que fala, e que também reduz o espaço ocupado pela frase escrita, tendo em vista que podemos utilizar uma notação mais econômica, a lógica.

Dentre estas notações, as proposições são usualmente simbolizadas (representadas) por letras maiúsculas do início do alfabeto: A, B, C, ... Os valores lógicos das proposições são representados de forma resumida usando V para verdadeiro e F para falso.

As proposições podem ser simples ou compostas. As proposições compostas são formadas de proposições simples conectadas através de operadores (ou conetivos) lógicos. Estes operadores ou conetivos representam as seguintes operações lógicas:

- Conjunção
- Disjunção
- Negação
- Implicação (ou condicional)
- Bi-implicação (ou bicondicional)

Devido à riqueza da língua portuguesa, palavras com significados semelhantes são representadas pelo mesmo operador lógico. A Tabela 3 mostra expressões comuns em português associadas a diversos operadores lógicos.

Tabela 3 – Expressões em português e operadores lógicos

Expressão em português	Operador Lógico	Expressão Lógica
E; mas; também	Conjunção	$A \wedge B$
Ou	Disjunção	$A \vee B$
Se A, então B A implica B A, logo B A somente se B A é condição suficiente para B	Condicional	$A \rightarrow B$
A se e somente se B	Bicondicional	$A \leftrightarrow B$
Não A É falso que A... Não é verdade que A... Não é o caso que A...	Negação	$\neg A$

Fonte: [GLUZ e PY, 2010].

A lógica formal lida com um tipo particular de argumento, denominado de argumento dedutivo, que nos permite deduzir uma conclusão Q , com base num conjunto de proposições P_1 a P_n , onde Q e P_1 a P_n representam fórmulas inteiras bem-formadas da lógica proposicional.

Um argumento dedutivo pode ser representado de forma simbólica da seguinte forma:

$$P_1 \wedge P_2 \wedge P_3 \wedge \dots \wedge P_n \rightarrow Q$$

As proposições P_1 a P_n são denominadas de **hipóteses** ou **premissas** do argumento. A proposição é denominada de **conclusão** do argumento. Em termos de língua natural este tipo de simbolismo pode ser lido como:

“ P_1, P_2, \dots, P_n **acarretam** Q ” ou

“ Q **decorre** de P_1, P_2, \dots, P_n ” ou

“ Q se **deduz** de P_1, P_2, \dots, P_n ” ou ainda

“ Q se **infere** de P_1, P_2, \dots, P_n ”

Este método é baseado na aplicação de **regras de dedução** (ou **regras de inferência**) que modificam fórmulas de modo a preservar seu valor lógico.

3.2.1 Regras de Dedução Natural

Existem dois tipos básicos de regras de dedução:

- Regras de inferência que se baseiam em implicações tautológicas⁴, ou seja, implicações materiais provadamente tautológicas.
- Regras de equivalência que se baseiam nas equivalências tautológicas permitem substituir uma fórmula pela outra, já que ambas são equivalentes.

As regras que são baseadas em implicações que já se tenha demonstrado serem tautológicas serão denominadas de Regras de Inferência. A Tabela 4 apresenta as regras básicas de inferência da lógica proposicional.

Tabela 4 – Regras Básicas de Inferência

Inclusão de Operadores	Exclusão de Operadores
Redução ao absurdo (raa) - $\neg\mathbf{I}$ $\begin{array}{l} P \\ \dots \\ Q \wedge \neg Q \\ \hline \neg P \end{array}$	Dupla negação (dn) - $\neg\mathbf{E}$ $\frac{\neg\neg P}{P}$
Prova condicional (pc) - $\rightarrow\mathbf{I}$ $\begin{array}{l} P \\ \dots \end{array}$	Modus Ponens (mp) - $\rightarrow\mathbf{E}$ $\frac{P \quad P \rightarrow Q}{Q}$

⁴ Onde um argumento é considerado verdadeiro, válido.

$\frac{Q}{P \rightarrow Q}$	Q
Conjunção(cj) - $\wedge I$ $\frac{P \quad Q}{P \wedge Q}$	Simplificação(sp) - $\wedge E$ $\frac{P \wedge Q}{P} \quad \frac{P \wedge Q}{Q}$
Adição(ad) - $\vee I$ $\frac{P}{P \vee Q} \quad \frac{P}{Q \vee P}$	Eliminação da disjunção - $\vee E$ $\frac{P \vee Q \quad P \rightarrow R \quad Q \rightarrow R}{R}$
Introdução da equivalência - $\leftrightarrow I$ $\frac{P \rightarrow Q \quad Q \rightarrow P}{P \leftrightarrow Q}$	Eliminação da equivalência - $\leftrightarrow E$ $\frac{P \leftrightarrow Q}{P \rightarrow Q} \quad \frac{P \leftrightarrow Q}{Q \rightarrow P}$

Fonte: [GLUZ e PY, 2010].

Estas 10 regras são completas no sentido que permitem a manipulação de todos os operadores da lógica proposicional. Do lado esquerdo estão listadas as regras que permitem a inclusão de um dado operador em uma nova fórmula a ser adicionada no decorrer da prova. As regras do lado direito permitem adicionar uma nova prova à demonstração, onde um determinado operador foi eliminado.

Além destas regras básicas, existe também um conjunto de regras derivadas que também podem ser usadas em demonstrações. Embora todas as regras derivadas apresentadas na Tabela 5 possam ser substituídas por demonstrações sem o uso delas, elas podem ser bastante úteis em determinados casos, facilitando o processo de demonstração.

Tabela 5 – Regras de Inferência Derivadas

Modus Tollens (mt) $\frac{P \rightarrow Q \quad \neg Q}{\neg P}$	Silogismo Hipotético (sh) $\frac{P \rightarrow Q \quad Q \rightarrow R}{P \rightarrow R}$
Silogismo Disjuntivo (sd) $\frac{P \vee Q \quad \neg P}{Q}$	Dilema Construtivo (dc) $\frac{P \vee Q \quad P \rightarrow R \quad Q \rightarrow S}{R \vee S}$
Exportação (exp) $\frac{(P \wedge Q) \rightarrow R}{P \rightarrow (Q \rightarrow R)}$	Inconsistência (inc) $\frac{P \quad \neg P}{Q}$

Fonte: [GLUZ e PY, 2010].

A ideia básica deste método é começar com as premissas P_1, P_2, \dots, P_n (supostamente verdadeiras) e tentar aplicar regras de dedução até terminar com a conclusão Q . Esta conclusão teria que ser, então, verdadeira uma vez que os valores lógicos são sempre preservados sob as regras de inferência [GLUZ e PY, 2010].

Dessa forma uma demonstração formal da lógica proposicional seria formada por uma série de linhas numeradas com a seguinte estrutura:

1.	P_1	(hipótese 1)
2.	P_2	(hipótese 2)
...		
n .	P_n	(hipótese n)
$n+1$.	F_1	(fórmula obtida aplicando-se uma regra de dedução sobre as fórmulas anteriores)
$n+2$.	F_2	(fórmula obtida aplicando-se uma regra de dedução sobre as fórmulas anteriores)
...		
$n+m$.	F_m	(fórmula obtida aplicando-se uma regra de dedução sobre as fórmulas anteriores)
	Q	(fórmula obtida aplicando-se uma regra de dedução sobre as fórmulas anteriores)

Neste tipo de argumento a conclusão Q simplesmente é a fórmula contida na última linha obtida através da aplicação de uma regra de dedução.

A sequência de fórmulas obtidas por este processo é denominada de **sequência de demonstração** ou apenas de **demonstração formal** da conclusão em função de suas premissas.

3.3 Ambientes Virtuais de Aprendizagem

Com o avanço e o desenvolvimento de novas tecnologias de informação e comunicação nos últimos anos, inúmeras são as possibilidades de expansão, que impulsionam e transformam a maneira de ensinar e aprender. O uso combinado de recursos tecnológicos e humanos traz uma nova proposta, ampliando horizontes e indo além das localizações geográficas.

Nesse sentido, os Ambientes Virtuais de Aprendizagem (AVAs) vêm sendo cada vez mais utilizados no âmbito acadêmico como uma opção tecnológica que visa atender a demanda educacional. Em termos conceituais, os AVAs consistem em mídias que utilizam o ciberespaço para veicular conteúdo e permitir interação entre os atores do processo educativo (PEREIRA, 2007). Outros fatores que influenciam a qualidade do processo educativo são: o envolvimento do aluno, da proposta pedagógica, dos materiais utilizados, de professores qualificados, tutores e monitores (que atuam na equipe técnico-pedagógico), assim como a infraestrutura, demais ferramentas e recursos tecnológicos utilizados no ambiente.

Assim, ao utilizarmos AVAs, papéis tradicionais como o de professores e alunos, podem ser redefinidos, de forma que o aluno se torne um agente mais ativo no

processo de aprendizagem, produzindo conhecimento, estabelecendo relações, colaborando com os colegas e socializando ideias.

O ferramental disponível nos AVAS pode ser considerado como um conjunto de recursos empregados para a aprendizagem, fornecendo um ensino contextualizado. Dos diversos Sistemas de Gerenciamento da Aprendizagem (SGAs), do inglês *Learning Management Systems* (LMS), dois padrões suportados pelos OAs se destacam nas atuais implementações de AVAs, que são os padrões SCORM e IMS Learning Design, por possuírem uma maior conformidade.

A Tabela 6 traz um comparativo entre as especificações destes padrões.

Tabela 6 – Comparativo de AVAs nos padrões SCORM e IMS

Funcionalidade	ADL SCORM 2004	IMS Learning Design
LMS com suporte	Alumni Gestum, Atutor, Amadeus, Aulanet, Blackboard, Claroline, Eureka, Ilias, iTutor, LearningSpace, Moodle, Sakai, WebAula, WebCT, MPLS 2, etc.	Atutor, Moodle (a partir da versão 2.0) e outros com base na Engine CooperCore. Reload Player (permite visualizar, mas não é LMS)
LMS sem suporte	Solar (em implementação)	Solar, Teleduc
Flexibilidade para diferentes abordagens pedagógicas	Projetado principalmente para auto-aprendizagem com conteúdos mais sequenciais	Suporta as mais variadas abordagens, bastando combinar os objetos, os serviços, os atores e seus papéis nas atividades de aprendizagem
Interatividade entre o aluno e o conteúdo	Permite, dependendo mais da forma como o conteúdo foi projetado	Permite, dependendo mais da forma como o conteúdo foi projetado
Interação entre aluno com professor e outros alunos	Não suportado	Prevê os papéis dos participantes e o nível de interação
Relação entre conteúdo e outras ferramentas do LMS	Não suportado	Prevê os papéis dos participantes e o nível de interação
Registro das atividades do aluno (tracking)	Permite	Não suporta rastreamento, ficando a cargo do LMS

Fonte: DUTRA e TAROUCO [2011].

Nesta Tabela é possível observar que ambas as especificações contêm características semelhantes ou complementares, fazem parte de consórcios que buscam a normalização e compatibilidade de soluções, utilizam-se da linguagem XML para diversos fins. O IMS Learning Design diferencia-se do SCORM por ser mais voltado para o conteúdo, pois sua utilização está mais voltada para o processo de ensino/aprendizagem, do que somente a relação de um único aluno diretamente com o conteúdo.

Em uma pesquisa feita nas Universidades do Brasil⁵ (dentre elas: Federais, Estaduais, Municipais e Privadas) foi possível obter como resultado o de que pelo menos 98% destas instituições de ensino superior fazem o uso do ambiente virtual Moodle e mais algum outro ambiente de aprendizagem (normalmente desenvolvido pela própria instituição).

Os serviços disponíveis pelos AVAs, o que inclui não só o Moodle, mas também alguns dos outros AVAs são utilizados pelas diversas instituições de ensino e também por corporações empresariais.

3.4 Objetos de Aprendizagem

Estudos sobre os OAs são relativamente recentes, não existindo ainda um consenso universalmente aceito sobre sua definição. Assim encontramos na literatura muitos autores que possuem conceitos fundamentados na ideia dos OAs serem uma porção reutilizável de conteúdo educacional, tais como:

“Qualquer entidade digital ou não digital, que possa ser usada, reusada e referenciada com tecnologia no suporte ao aprendizado” [IEEE-LTSC 2002].

“Qualquer recurso digital que possa ser reutilizado no suporte ao ensino” [WILLEY, 2001].

“Um pedaço digital de material de aprendizagem que leva a uma identificação clara de tópico ou resultados de aprendizagem e tem potencial para ser reutilizado em diferentes contextos” [MASON e PEGUER, 2003].

Os OAs podem ser criados em qualquer mídia e formato, podendo ser tão simples como uma animação ou uma apresentação de slides ou tão complexos como uma simulação. A partir daí percebe-se que os mesmos podem ser classificados como OAs simples e OAs complexos. Os OAs catalogados como simples são considerados arquivos únicos, tais como: arquivo texto, imagem, animação, ou seja, qualquer arquivo que não sofre granularidade. Já os OAs catalogados como complexos são compostos por uma série de arquivos que podem estar incluídos em uma pasta ou mesmo em um arquivo compactado, onde é possível de trabalhar sua granularidade e, principalmente, onde o conceito de reusabilidade é mais fortemente empregado [BRASIL, 2007].

Diversos são os fatores que favorecem o uso de OAs na área educacional. Para que uma entidade digital seja considerada um OA, a mesma deve apresentar características que a identifique como tal. Desta forma, os OAs podem ser analisados em duas perspectivas: pedagógica e técnica.

As características pedagógicas estão relacionadas com a concepção de objetos que facilitem o trabalho de professores e alunos, visando à aquisição do conhecimento tais como [DIAS, 2009]:

⁵ Esta pesquisa foi realizada fazendo uma seleção contendo todas as Universidades do Brasil, dentre elas: Federais, Estaduais, Municipais e Privadas, acessando seus sites e portais EAD constatando o uso de um ou mais AVAs.

- (1) Interatividade, sistema que oferece suporte às concretizações e ações mentais;
- (2) Autonomia, recursos de aprendizagem que proporcionem a iniciativa e tomada de decisão;
- (3) Cooperação, onde usuários trocam ideias e trabalham coletivamente sobre o conceito apresentado;
- (4) Cognição, onde se refere às sobrecargas cognitivas colocadas na memória do aprendiz durante a instrução;
- (5) Afetividade, que está relacionado ao componente biológico do comportamento humano e motivações por parte do aluno com sua aprendizagem e colegas.

As características técnicas referem-se, por exemplo, as questões de padronização, classificação, armazenamento, recuperação, transmissão e reutilização dos OAs. São características técnicas específicas dos OAs [DIAS, 2009]:

- (1) Acesso, que indica se um OA pode ser utilizado remotamente em muitos outros locais;
- (2) Agregação, que indica quais recursos podem ser agrupados em conjuntos maiores de conteúdos, incluindo estruturas tradicionais de cursos;
- (3) Autonomia, que verifica se o objeto pode ser usado individualmente;
- (4) Classificação, que permite a catalogação dos objetos auxiliando na identificação dos mesmos, facilitando o trabalho dos mecanismos de busca,
- (5) Formatos digitais dos conteúdos;
- (6) Durabilidade, que indica se a contínua usabilidade de recursos educacionais se mantém quando a base tecnológica muda, sem reprojeção ou recodificação;
- (7) Interoperabilidade, que implica em utilizar os OAs em diferentes locais, independente de ferramentas ou plataformas;
- (8) Reusabilidade, onde a reusabilidade varia de acordo com a granularidade do OA.

Os aspectos técnicos dos OAs são profundamente influenciados por questões de padronização. Funcionalidades como acesso, agregação, reutilização, interoperabilidade, além do intercâmbio de informações entre os diversos tipos de formatos digitais, entre outras características dos OAs, são diretamente dependentes da existência de padrões para se tornarem possíveis. Assim, por exemplo, a reutilização, que consiste em uma forma eficiente de readaptar o conteúdo dos OAs para diferentes tipos de contextos e usuários, somente se torna possível com o auxílio dos padrões. Nesse sentido surgiram algumas iniciativas visando padronizar a especificação, construção e a identificação dos OAs, através da adoção de modelos e padrões para o desenvolvimento destes [DIAS, 2009].

Nesse contexto, podem-se citar organizações e grupos de pesquisas que já vêm trabalhando para construir e aprimorar a eficiência e eficácia dos objetos de aprendizagem, concentrando-se principalmente na definição de padrões. São eles:

Learning Technology Standard Comitee (LTSC) do Institute of Electrical and Electronics Engineers (IEEE), a Alliance of Remote Institute of Electrical and Distribution Networks for Europe (ARIADNE), o IMS Global Learning Consortium, a Canadiam Core e a Advanced Distributed Learning Initiative que têm contribuído significativamente na definição de padrões de indexação (metadados) [KRATZ; CRESPO; SCOPEL; BARBOSA, 2007].

No caso das padronizações dos OAs foram definidas várias recomendações a respeito dos metadados que catalogam e que caracterizam os OAs, bem como também, recomendações de como estes OAs podem ser encapsulados e como seus conteúdos podem ser navegados [W3C, 2010]. Nesta lista temos como principais: O LOM (Learning Object Metadata) do IEEE Learning Technology Standards Committee (LTSC), DCMI (Dublin Core Metadata Initiative), SCORM (Sharable Content Object Reference Model) ADL [ADL, 2010] e IMS do Global Learning Design [DUTRA e TAROUCO, 2010].

No projeto do sistema Heráclito foi adotado a proposta de metadados definida pelo projeto OBAA, de forma a aplicar e estender os conceitos e tecnologias previstas nessa proposta para o domínio de ensino de Lógica Formal [GLUZ e VICARI, 2010] incluindo suporte para:

- a) Adaptabilidade e interoperabilidade de OAs em plataformas digitais como Web, TV Digital e A proposta de padrão de metadados OBAA foi definida com base no padrão IEEELOM dispositivos móveis;
- b) Compatibilidade do padrão com o panorama de padrões internacionais;
- c) Acessibilidade aos OAs por todos os cidadãos, inclusive aqueles com necessidades especiais, e;
- d) Independência e flexibilidade tecnológica do padrão, que não necessita de tecnologias proprietárias, e que permite que inovações tecnológicas sejam incorporadas, sem perder a compatibilidade com o material já desenvolvido.

Os metadados OBAA resultantes dessa análise foram definidos como uma extensão dos metadados IEEELOM, formada pela adição de novos grupos de metadados: metadados técnicos multiplataforma, metadados pedagógicos, metadados de acessibilidade e metadados de segmentação de conteúdos multimídia.

3.5 Agentes de Software e Sistemas Multiagente

Um agente pode ser descrito como qualquer entidade que seja capaz de perceber o seu ambiente através dos seus sensores e agir sobre ele por intermédio de atuadores [Russell e Norvig, 2004], conforme pode ser visto na Figura 4.

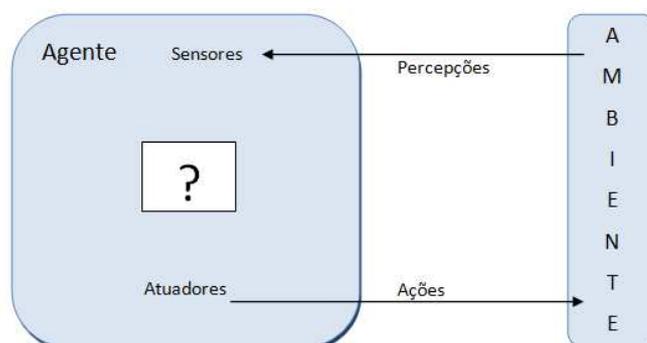


Figura 4 – Agentes interagem com ambientes por meio de sensores e atuadores.
Fonte: RUSSELL e NORVIG [2004].

Mais especificamente, um agente de software recebe uma entrada de dados, que pode ser desde conteúdos de arquivos, pacotes de dados até entradas sensoriais e atua no ambiente exibindo em monitores, escrevendo arquivos e enviando pacotes pela rede, por exemplo, [RUSSEL e NORVIG, 2004].

Ao se projetar um agente, o primeiro passo sempre deve ser a especificação do ambiente de maneira tão completa quanto possível. Os ambientes podem ser dos tipos: completas ou parcialmente observáveis, determinísticos ou estocásticos, episódicos ou sequenciais, estáticos ou dinâmicos, discretos ou contínuos e de agente único ou multiagentes. Todos os agentes podem melhorar seu desempenho por meio do aprendizado [RUSSEL e NORVIG, 2004].

Os sistemas multiagentes são compostos por múltiplos agentes trabalhando em conjunto. Estes componentes podem estar em uma mesma máquina ou espalhados em uma rede. Este tipo de sistema, normalmente, é indicado para resolver problemas que são difíceis ou impossíveis de serem resolvidos por um único agente. Estes agentes, quando inseridos em uma comunidade interagem com os demais para formar suas crenças a partir do conhecimento já adquirido pelos outros componentes do sistema [WOOLDRIDGE, 2002].

Um dos principais aspectos do desenvolvimento de sistemas multiagentes diz respeito à forma sob a qual os agentes irão se comunicar. Sem mecanismos de comunicação é praticamente impossível a definição de agentes.

Os métodos de comunicação podem ser divididos em dois grupos: quadro de avisos e trocas de mensagens. Os sistemas baseados em quadro de avisos mantêm uma área de trabalho comum na qual os agentes podem trocar informações, dados e conhecimentos. Uma ação de comunicação é iniciada quando um agente escreve alguma coisa no quadro negro. Os agentes podem acessar este quadro a qualquer momento. Na prática, um agente não lê todas as novas informações escritas, mas somente aquelas que lhe interessam. Um sistema multiagente pode ter vários quadros negros. Se um sistema possui um grande número de agentes, a quantidade de informação escrita cresce exponencialmente e, portanto, a busca por informações relevantes se torna cada vez mais pesada. Para resolver esse problema, quadros negros mais avançados tem sua área dividida em subáreas, cada qual privativo a um agente ou a um grupo deles. Nesse caso, um agente só precisa monitorar a região a ele associada. Esse tipo de sistema prove um conceito muito flexível para a cooperação e comunicação entre agentes, pois são independentes da estratégia de cooperação selecionada. Entretanto, a leitura das informações do quadro negro pode causar uma sobrecarga do sistema. Por isso, a importância dos sistemas baseados em troca de mensagens vem crescendo continuamente [BRENNER e RÜDIGER, 1998].

A comunicação usando mensagens forma uma base flexível para a implementação de estratégias de coordenação complexas. Neste trabalho o método adotado é o da troca de mensagens. A Figura 5 ilustra o conceito básico do sistema de troca de mensagens.

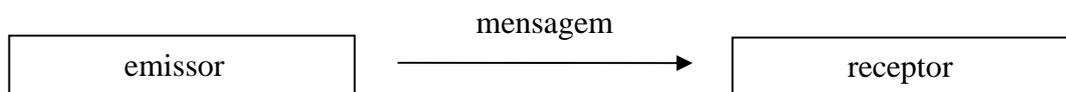


Figura 5 – Conceito básico do sistema de troca de mensagens.
Fonte: BRENNER e RÜDIGER [1998]

Um agente, denominado emissor, envia uma mensagem a outro agente, o receptor. Comparando com os sistemas de quadro negro, as informações são trocadas diretamente entre os agentes.

Estas trocas de mensagens podem ser feitas de três formas:

- Ponto-a-ponto: As mensagens são enviadas para um agente específico que deve ser conhecido pelo emissor;
- *Broadcast*: A mensagem é enviada para todos os agentes da sociedade;
- *Multicast*: A sociedade de agentes é dividida em grupos. Desta forma as mensagens podem ser enviadas somente aos agentes de determinados grupos.

Para que o mecanismo de troca de mensagens possa ser utilizado na implementação de estratégias de cooperação é preciso que seja definido um protocolo de comunicação que especifique o exato processo de comunicação, um formato para as mensagens e a linguagem de comunicação escolhida. O mecanismo de troca de mensagens necessita da definição de um protocolo de comunicação que especifique o exato processo de comunicação, um formato para as mensagens e uma linguagem de comunicação. Esses requisitos nos levam às Linguagens de Comunicação de Agentes (ACL) e aos serviços de transportes de mensagens [FIPA, 2011].

Neste trabalho todo o processo de comunicação entre os agentes se dará por troca de mensagens na linguagem FIPA-ACL [FIPA, 2011], suportadas pela plataforma de agentes JADE [BELLIFEMINE; CAIRE; GREENWOOD, 2007], que é compatível com os padrões da FIPA [FIPA, 2011].

3.6 Arquitetura BDI

Os agentes individuais de um dado sistema multiagente podem ser projetados e desenvolvidos de várias formas distintas. Uma das arquiteturas de agentes individuais bastante difundidas é a baseada no modelo BDI (do inglês *Beliefs – Desires – Intentions*). O modelo BDI representa uma arquitetura cognitiva para agentes inteligentes, baseada em estados mentais, e tem sua origem no modelo de raciocínio prático humano, a qual consiste em explicar o comportamento humano baseado em três estados mentais fundamentais: crenças, desejos e intenções [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

As crenças (*Beliefs*) representam o conhecimento sobre o mundo no qual o agente está inserido (representação do ambiente, que pode ou não corresponder à realidade). Do ponto de vista computacional, crenças são apenas uma maneira de representar o estado do mundo, seja através de variáveis, uma base de dados relacional, ou expressões simbólicas em um cálculo de predicados.

Os desejos (*Desires*) são representados por diferentes estados que o agente pode perseguir ou coisas que ele gostaria de realizar (são relacionados ao estado de mundos que o agente quer provocar). Os desejos não dirigem necessariamente o agente a agir. Antes de um determinado agente decidir o que fazer, ele passa por um processo deliberativo, confrontando os seus desejos com as suas crenças e escolhendo os desejos que são possíveis de acordo com um critério. Os desejos resultantes desse processo passam a serem intenções do agente [Oliveira, 2012].

As intenções (*Intentions*) têm como objetivo alcançar o estado que o agente decidiu perseguir e para o qual ele comprometeu recursos (correspondem a estados de mundo que o agente quer atingir). As Intenções podem ser consideradas um subconjunto dos desejos, mas ao contrário destes, devem ser consistentes. Quando uma intenção é selecionada, o agente deve se comprometer com a realização da mesma. Para tanto é necessário executar um raciocínio prático, buscando identificar que ações devem ser executadas para satisfazer a intenção. Este tipo de raciocínio está diretamente associado a um processo de planejamento com objetivo de realizar as intenções do agente.

O modelo BDI personifica estes estados mentais em agentes de software. Dessa forma, a ação dos agentes é direcionada por uma consideração desses estados, como subsídio para atingir seus objetivos. A escolha do modelo BDI se justifica por ser bem aceito na modelagem de agentes que precisam de algum nível de autonomia, adaptabilidade, personalidade e inferência, por se revelar o mais completo em termos cognitivos.

Neste trabalho o Agente Mediador (foco deste trabalho) foi projetado e desenvolvido com base no modelo BDI. Maiores informações sobre o Agente Mediador podem ser obtidas no capítulo 6.

3.7 A linguagem AgentSpeak(L) e a Plataforma Jason

A linguagem de programação AgentSpeak(L) foi projetada com o objetivo de possibilitar a construção de agentes na arquitetura BDI. Ela é uma extensão natural de programação em lógica para a arquitetura de agentes BDI, que representa um modelo abstrato para a programação de agentes e tem sido a abordagem predominante na implementação de agentes inteligentes [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

Um agente AgentSpeak(L) corresponde à especificação de um conjunto de crenças que formarão a base de crenças inicial e um conjunto de planos e reagem a eventos, gerados a partir de mudanças no ambiente ou na sua estrutura interna, utilizando a biblioteca de planos.

Estes planos (ou conjunto de planos) fazem referência a ações básicas que um

agente é capaz de executar em seu ambiente e é composto por três elementos: evento ativador (ou evento de disparo – *triggering event*) denotando o propósito do plano, a representação do contexto (*context*) uma conjunção de literais de crenças e corpo (*body*). Estas três partes do plano são sintaticamente separadas por ':' e '<-' como segue no exemplo abaixo:

```
triggering_event : context <- body.
```

Um evento ativador (ou fato gerador) de um plano existe precisamente para que o agente possa pesquisar em sua biblioteca, quais planos específicos podem ser utilizados em algum dado evento. Um agente pode ter dois tipos de atitudes mentais que são importantes em um programa de agente. São elas: mudanças nas crenças, onde se referem às informações que os agentes têm sobre seu ambiente ou sobre outros agentes, e mudanças nos objetivos do agente.

O contexto de um plano é usado para verificar a situação atual do ambiente, de forma a determinar se o plano pode ser executado ou não, dado as últimas informações que o agente tem sobre o seu meio ambiente. Nesse sentido, um plano só é escolhido para a execução se o seu contexto for uma consequência lógica de crenças do agente. Um plano que tem um contexto que é avaliado como *verdadeiro* dado às crenças atuais do agente, é dito como *aplicável* nesse momento, e é um candidato para a execução do mesmo. Tipicamente, um agente terá vários planos diferentes para alcançar a mesma meta. Um contexto é simplesmente um conjunto de expressões relacionais, e tem a sua sintaxe muito semelhante ao Prolog.

O corpo de um plano, em termos gerais, é simplesmente uma sequência de fórmulas, separadas por ponto e vírgula (“;”), que determinam a sequência de ações a serem executadas para que o agente atinja o objetivo proposto contido no plano. Existem seis tipos diferentes de fórmulas que podem aparecer em um corpo de plano. São elas: ações (*Actions* – funções nativas do Jason⁶), metas de realização (*Achievement Goals*), metas de teste (*Test Goals*), notas mentais (*Mental Notes*), ações internas (*Internal Actions*) definidas pelo usuário e expressões (*Expressions* – como, por exemplo: $X \geq Y * 2$). Outros objetivos podem aparecer dentro do corpo, sendo chamados de subobjetivos.

Como o agente está constantemente monitorando (observando) seu ambiente, a cada ciclo de raciocínio um novo plano será escolhido para ser executado, com o propósito de atingir seus objetivos, podendo ou não alterar seu ambiente. Os planos executados pelo agente se encontram armazenados em uma biblioteca de planos, inicialmente formada pelos planos que o próprio programador escreveu em sintaxe AgentSpeak.

Nesse sentido, todas as informações dos desejos, assim como as alternativas disponíveis do agente para ativar as intenções estão implicitamente representadas nos planos. Sensíveis ao contexto, os planos necessitam que certas condições sejam

⁶ A ferramenta Jason (do inglês Java-based AgentSpeak Interpreter Used with Saci for Multi-Agent Distribution Over the Net) é uma plataforma de desenvolvimento de sistemas multiagentes que proporciona a interpretação e execução de programas escritos na linguagem AgentSpeak(L).

satisfeitas para serem executados. Nesse sentido o contexto deve ser uma consequência lógica da base de crenças do agente. O corpo do plano é uma sequência de ações básicas que o agente deve realizar (atingir) quando uma instancia do plano é selecionada para execução [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

Já a ferramenta Jason (do inglês *Java-based AgentSpeak Interpreter Used with Saci for Multi-Agent Distribution Over the Net*) é uma plataforma de desenvolvimento de sistemas multiagentes que proporciona a interpretação e execução de programas escritos na linguagem AgentSpeak(L).

Na Figura 6 tem-se a ilustração da arquitetura do Jason. Sua representação do conjunto abaixo é composta por:

- Retângulos: representam os principais componentes arquitetônicos que determinam o estado do agente (ou seja, a base de crença, o conjunto de eventos, a biblioteca plano, e o conjunto de intenções);

- Caixas arredondadas e círculos: representam as funções utilizadas no ciclo de raciocínio.

- Losangos: representam funções de seleção de um elemento a partir de um determinado conjunto e círculos para representar alguns dos processos envolvidos na interpretação de agentes AgentSpeak (L).

Ambas as caixas arredondadas e losangos representam as funções que podem ser personalizados pelos programadores, enquanto que os círculos são partes essenciais do intérprete que não podem ser modificados. A diferença entre caixas arredondadas e losangos é que este último denota funções de seleção.

Na Figura abaixo também é possível visualizar a descrição de como o intérprete Jason executa um programa de agente (passo-a-passo):

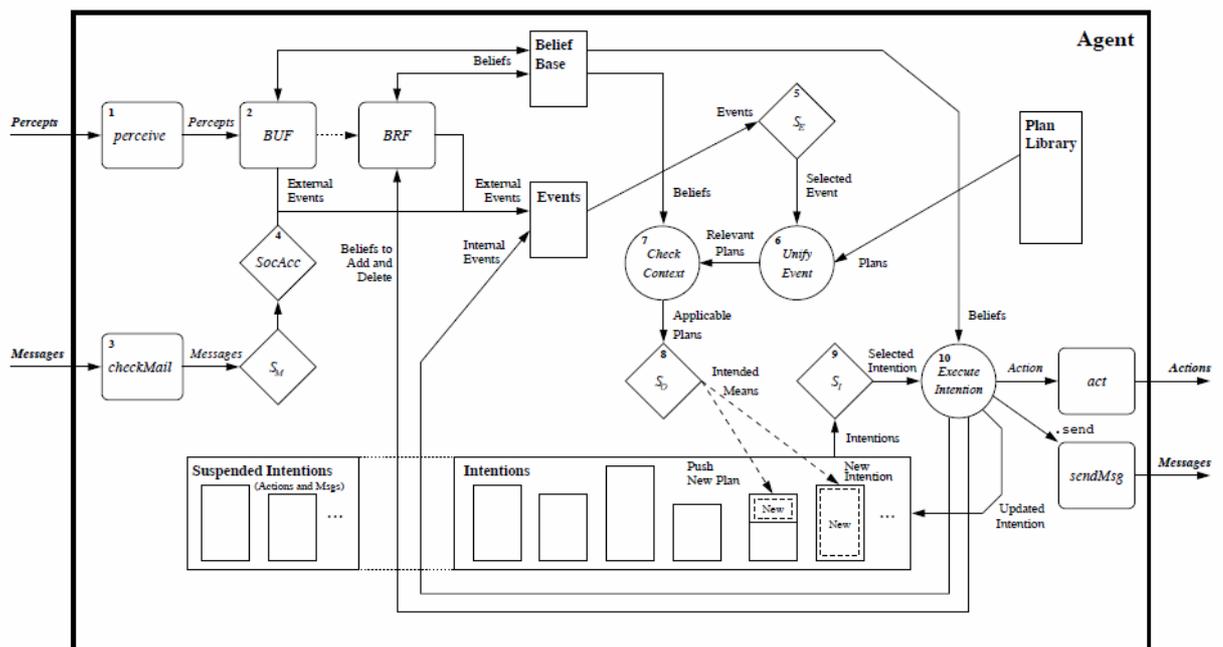


Figura 6 – Ciclo de Raciocínio do Jason.

Fonte: BORDINI; HÜBNER; WOOLDRIDGE [2007, p. 68].

Um agente opera por meio de um ciclo de raciocínio e este pode ser dividido em 10 principais passos, como pode ser visto:

1. Ambiente perceber o ambiente;
2. Atualiza a base de crenças;
3. Recebe comunicação de outros agentes;
4. Seleciona “socialmente aceitável” as mensagens;
5. Seleciona um evento;
6. Recupera todos os planos relevantes;
7. Determina se os planos são aplicáveis;
8. Seleciona um plano aplicável;
9. Seleciona uma intenção para posterior execução;
10. Executa um passo de uma intenção;

Cada ciclo de interpretação atualiza a lista de eventos de acordo com a percepção proveniente do ambiente e das informações advindas da própria execução do agente.

A configuração do interpretador é feita em arquivos com extensão .mas2j e a programação dos agentes AgentSpeak(L) em arquivos com extensão .asl.

Jason é implementado na linguagem Java (executada em múltiplas plataformas), sendo Open Source e distribuído sob a licença GNU LGPL.

3.8 O Padrão FIPA e a Plataforma JADE

A FIPA (*Foundation for Intelligent Physical Agents*) é uma associação internacional composta por companhias e organizações que têm como objetivo compartilhar esforços com o intuito de produzir especificações para tecnologias de agentes genéricas. Foi criada em 1996 sem fins lucrativos para desenvolver uma coleção de normas relativas à tecnologia de agentes de software [FIPA, 2011].

Os documentos criados definem regras que permitem a uma sociedade de agente existir, operar e ser gerenciada. A partir desses princípios, a FIPA define um modelo de referência para uma plataforma de agentes e se baseia em três agentes especiais, responsáveis pela gerencia da plataforma e pela descrição da linguagem de conteúdo para gerenciamento de agentes e ontologias: o MAS (*Agent Management System*), o DF (*Directory Facilitator*) e o ACC (*Agent Communication Channel*), como mostra a Figura 7 abaixo:

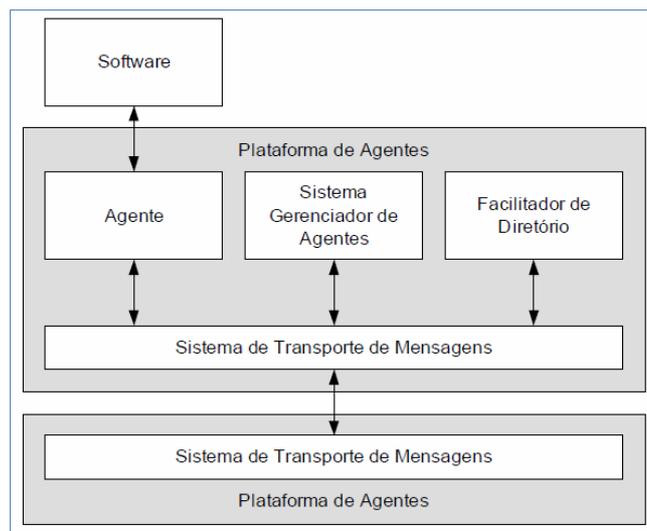


Figura 7 – Modelo de referência para uma plataforma de agentes FIPA.
Fonte: SILVEIRA [2012]

O modelo de plataforma padrão especificado pela FIPA é composto pelos seguintes componentes: O agente (*Agent*) tem suas tarefas definidas de acordo com o objetivo da aplicação e realiza toda sua comunicação com agentes através de troca de mensagens e relacionando-se com aplicação externa (software). O sistema gerenciador de agentes ou *Agent Management System* (MAS) é o agente que supervisiona o acesso e o uso da plataforma de agentes. Ele provê guia de endereços e controle de ciclo-de-vida, mantendo um diretório de identificadores de agentes e estados de agentes. Ele é o responsável pela autenticação de agentes e pelo controle de registro. Cada agente tem que se registrar no AMS (Agent Management System) para obter um ID válido. O diretório facilitador (*Directory Facilitator - DF*) é o agente que provê o serviço de páginas amarelas (*yellow-pages*) dentro da plataforma. O sistema de transporte de mensagens ou *Message Transport System* é o agente responsável por prover toda a comunicação entre agentes dentro e fora da plataforma [BORDINI; HÜBNER; WOOLDRIDGE, 2007).

As mensagens FIPA têm a seguinte estrutura como mostra a Figura 8 abaixo:

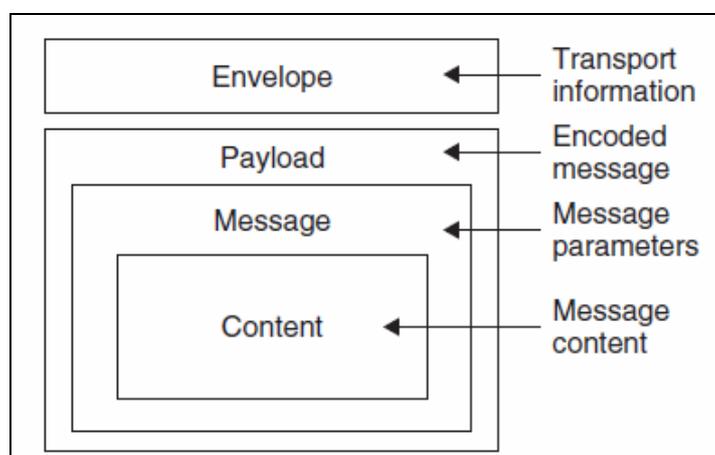


Figura 8 – Estrutura de mensagens FIPA.
Fonte: BORDINI; HÜBNER; WOOLDRIDGE [2007]

As especificações FIPA definem um modelo de referência para uma plataforma de agentes e também um conjunto de serviços fornecidos ao se conceber sistemas multiagentes interoperáveis, e o JADE (*Java Agent Development Framework*) é uma plataforma (em conformidade com as especificações da FIPA) para o desenvolvimento e execução de sistemas baseados em agentes de *software* que segue as recomendações da FIPA. Está escrito em Java, por questões de portabilidade, e distribuído sob a licença de *software* livre LGPL (*Lesser General Public License*).

A estrutura das mensagens trocadas na comunicação entre agentes é baseada na linguagem ACL (*Agent Communication Language*) definida pela FIPA e contém campos tais como, variáveis que indicam o contexto ao qual a mensagem se refere e, o tempo limite que pode ser aguardado até a resposta ser recebida (*timeout*), visando suporte a interações complexas e conversas paralelas múltiplas [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

Uma mensagem FIPA-ACL contém um conjunto de um ou mais parâmetros de mensagem. Estes parâmetros FIPA-ACL são divididos em cinco categorias: tipo do ato comunicativo, participantes da comunicação, conteúdo de mensagem, descrição do conteúdo e controle de conversação. A lista completa dos parâmetros e suas respectivas categorias são mostradas na Tabela 7.

Tabela 7 – Parâmetros das mensagens FIPA-ACL

Tipo do ato comunicativo	
Performative	Denota o tipo do ato comunicativo da mensagem.
Participantes da Comunicação	
Sender	É o emissor da mensagem.
Receiver	É o receptor da mensagem.
Reply-to	Indica que as próximas mensagens dessa conversação deverão ser enviadas para o agente indicado pelo parâmetro reply-to.
Conteúdo de mensagem	
Content	É o conteúdo ou conhecimento transportado pela mensagem.
Descrição do conteúdo	
Language	É a linguagem no qual o conhecimento está expresso.
Encoding	Aponta a codificação utilizada na expressão da linguagem de conteúdo.
Ontology	É a ontologia utilizada para dar significado à expressão de conteúdo.
Controle de conversação	
Protocol	É o protocolo de interação utilizado para essa mensagem pelo agente emissor.
Conversation-id	Introduz uma expressão que será usada para identificar a conversação em andamento.
Reply-with	Introduz uma expressão que será usada pelo agente que responderá a essa mensagem para identificá-la.
In-reply-to	É a expressão que referencia a mensagem à qual se está respondendo.
Reply-by	Explicita um tempo máximo durante o qual o agente emissor estará esperando por uma resposta a esta mensagem.

Fonte: FIPA [2011]

A comunicação entre agentes no JADE é a baseada em mensagens assíncronas, ou seja, um agente que deseja se comunicar deve apenas transmitir uma mensagem para

um destino identificado (ou conjunto de destinos), não existindo nenhum tipo de dependência temporal entre o transmissor e o receptor. Neste tipo de comunicação, a segurança da plataforma é preservada, pois ela provê mecanismos próprios de autenticação e verifica os direitos e prioridades atribuídos a cada agente, ou seja, quando é necessário, a aplicação pode verificar a identidade do emissor da mensagem e prevenir ações relacionadas a ele que são limitadas ou não permitidas [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

A plataforma JADE fornece suporte a ontologias e a linguagens de conteúdo. Por padrão, a verificação de ontologia e a codificação de conteúdo são executadas automaticamente pela plataforma, cabendo ao programador apenas selecionar a ontologia e/ou a linguagem de sua preferência.

As ontologias desenvolvidas para atuarem em conjunto com o Heráclito estão em formato OWL⁷ (conforme recomendações do W3C). Mais informações são encontradas na seção 3.8 e seção 5.3.

3.9 Agentes Pedagógicos

É possível empregar agentes inteligentes no âmbito da educação, como por exemplo, por meio de sistemas tutores inteligentes, os quais são responsáveis por aprender/coletar informações relevantes sobre o aluno. Com estes dados, é possível auxiliar o aluno de forma individualizada no aprendizado, bem como tomar decisões pedagógicas sobre a melhor maneira de como apresentar o conteúdo de forma didática. Para isso, estes agentes devem ter a capacidade de identificar o aluno pelo seu perfil e preferências.

A aplicação de conceitos de agentes em Sistemas Multiagentes (SMA), na criação de Sistemas Tutores Inteligentes (*Intelligent Tutoring Systems* - ITS) ou Ambientes Inteligentes é um ponto chave no âmbito da Educação, visto que estes ambientes têm papéis pedagógicos ou educacionais destinados a facilitar ou apoiar o processo de ensino.

Estes agentes, comumente denominados de Agentes Pedagógicos, podem ser agentes cooperativos que trabalham com o suporte de um sistema educacional, ou então podem ser agentes animados ou personalizados que interagem diretamente com os usuários, para auxiliá-los em processos de ensino/aprendizagem [VICARI e GLUZ, 2010].

Segundo [VICARI e GLUZ, 2010] as arquiteturas baseadas neste tipo de abordagem são variações da arquitetura funcional tradicional dos sistemas tutores inteligentes, que são usualmente divididos nos seguintes módulos:

- Módulo do modelo de aluno: é responsável pelas funções tipicamente atribuídas ao módulo que representa o aluno dentro de um sistema tutor.

⁷ Web Ontology Language é uma linguagem para definir e instanciar ontologias na Web. Uma ontologia OWL pode incluir descrições de classes e suas respectivas propriedades e seus relacionamentos. Também fornece um vocabulário adicional com uma semântica formal e isso faz com que as máquinas interpretem o conteúdo da Web mais facilmente que com XML, RDF e RDFS (RDF Schema).

Isto inclui, dentre outras, o registro das informações a respeito do aluno e o acompanhamento do processo de ensino-aprendizagem deste, pelo menos no contexto das interações do aluno com o ambiente;

- Módulo de estratégias de ensino: responsável por definir as formas de apresentação do material instrucional ao aluno.
- Módulo do domínio de ensino: responsável pela interface com o domínio de ensino correspondente. Este agente incorpora as funções usualmente atribuídas ao módulo especialista de um sistema tutor;

No caso de uma arquitetura baseada em agentes, os agentes pedagógicos usualmente implementam as funções destes três tipos de módulos (e possivelmente de outros tipos de módulos). O controle é distribuído entre os agentes, mas o usuário vê o sistema como um só. Internamente, entretanto o sistema está organizado como um SMA. Na concepção de sistemas educacionais, as maiores vantagens na abordagem baseada em agentes e SMA são a modularidade que este tipo de abordagem oferece, em conjunto com as possibilidades de interoperabilidade e reuso de conhecimentos de ensino em outros sistemas [VICARI e GLUZ, 2010].

3.10 A Infraestrutura MILOS

A infraestrutura MILOS (*Multiagent Infrastructure for Learning Object Support*) tem como objetivo criar as bases tecnológicas que permitam efetivar a adoção da proposta metadados OBAA [GLUZ e VICARI, 2010], estando centrada no projeto e implementação de ferramentas tecnológicas que permitam o suporte integral ao ciclo de vida OAs compatíveis com a proposta OBAA.

Esta infraestrutura é extensivamente baseada no uso de agentes pedagógicos, sistemas multiagentes e ontologias. Em termos arquiteturais a infraestrutura está dividida em três grandes níveis de abstração (ver Figura 9):

- Camada de ontologias.
- Camada de agentes.
- Camada de serviços de interface.

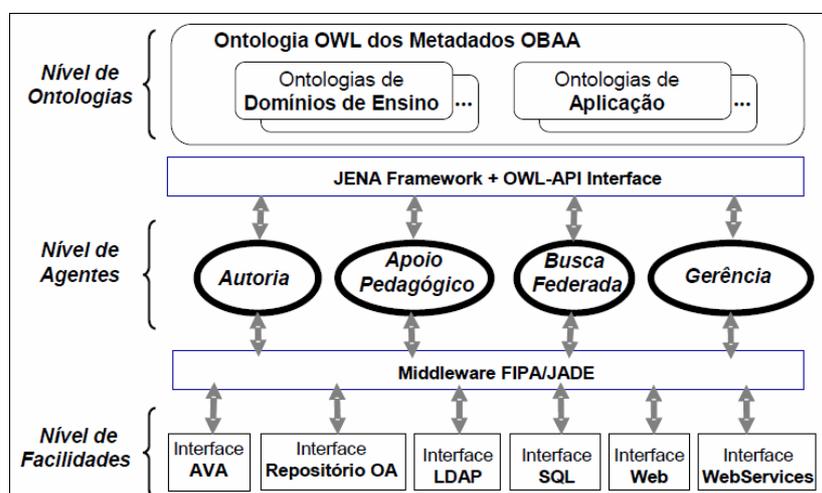


Figura 9 – Organização Geral da Infraestrutura MILOS.
Fonte: VICARI e GLUZ [2011].

A camada de ontologias contém além da ontologia de metadados OBAA, as ontologias sobre domínios de ensino e aplicações educacionais ou multimídia, todas construídas usando a ontologia OBAA como o vocabulário de termos básico [VICARI e GLUZ, 2011]. As ontologias contidas no nível de Ontologias deverão ser obrigatoriamente definidas em OWL.

A camada de agentes implementa as principais funcionalidades da MILOS, onde os seguintes sistemas multiagente implementados sobre JADE são previstos [VICARI e GLUZ, 2011]:

- Sistema de Busca Federada: cria e gerencia federações de catálogo de OAs, permite a busca dos metadados em toda a federação de forma transparente ao usuário;
- Sistema de Apoio Pedagógico: suporte para busca dependente do contexto e uso de estratégias de ensino;
- Sistema de Autoria: assistência para autoria de OAs multimídia e multiplataforma;
- Sistema de Gerência de OAs: gerenciamento e publicação de OAs nas distintas plataformas.

O papel dos agentes de *software* (ou agentes artificiais) na infraestrutura MILOS é implementar as funcionalidades e serviços que a infraestrutura deve disponibilizar aos seus diversos usuários. É importante diferenciar a atuação dos agentes do papel das ontologias: embora as ontologias possam especificar as propriedades de um domínio de aplicação, definindo inclusive as ações, percepções, algoritmos e heurísticas que são possíveis neste domínio, elas são essencialmente especificações estáticas deste conhecimento. Assim são os agentes de *software* que realmente transformam este conhecimento estático em comportamento dinâmico capaz de trazer benefícios aos seus usuários.

A camada de serviços de interface oferece facilidades de interface para Web permitindo que os agentes e sistemas multiagente da MILOS possam ser acessados por ambientes e aplicações educacionais, além de permitir também que estes agentes tenham acesso aos repositórios de OA, serviços de diretórios, plataformas de EAD, bancos de dados e servidores Web [VICARI e GLUZ, 2011].

As bases tecnológicas da infraestrutura MILOS são formadas pela combinação das tecnologias de agentes inteligentes e da engenharia de conhecimentos. Ambas as tecnologias têm raízes em pesquisas na área de Inteligência Artificial (IA), originadas nos anos setenta e oitenta, mas que tiveram um amplo desenvolvimento nos anos noventa. A Engenharia de *Software* Orientada a Agentes e a Engenharia de Ontologias, que são as versões mais recentes destas tecnologias, já consolidadas como técnicas de engenharia serão empregadas no projeto da MILOS.

3.11 Ontologias no Sistema Heráclito

Uma ontologia fornece um vocabulário para um domínio de interesse e uma especificação do significado dos termos em que o vocabulário está inserido. Dependendo da precisão desta especificação, a noção de ontologia engloba vários dados ou modelos conceituais, como por exemplo, as classificações, os esquemas de banco de dados ou teorias totalmente axiomáticas [EUZENAT e SHVAIKO, 2007].

Os vocabulários definem os conceitos e relacionamentos (também conhecido como "termos"), usado para descrever e representar uma área de preocupação. Estes vocabulários são usados para classificar os termos que podem ser usados em: uma aplicação particular, para caracterizar as relações possíveis, e definir possíveis restrições sobre o uso desses termos. Na prática, os vocabulários podem ser muito complexos (com vários milhares de termos) ou muito simples (descrevendo um ou dois conceitos apenas) [W3C, 2011].

Embora existam muitas definições sobre o termo ontologia na literatura e alguns autores apresentem pontos de vista diferentes da mesma realidade, o conceito sobre Ontologia é definido pela filosofia como uma ciência compartilhada de algum universo de discurso ou domínio de interesse, onde o conjunto de conceitos, suas definições e inter-relacionamentos, têm uma visão unificada para resolver problemas envolvendo significados [EUZENAT e SHVAIKO, 2007].

Na Computação a Ontologia abrange dois campos que são a conceitualização compartilhada de um domínio e a forma explícita e formal para permitir o processamento por máquinas. Tais campos podem ter diferentes formas:

- Thesaurus com relações semânticas entre termos (sinonímia);
- Taxonomia;
- Diagrama de classes;
- Base de conhecimento;
- Classes, propriedades e suas relações;
- Instâncias de classes.

Na forma explícita uma ontologia é uma representação concreta e manipulável de (ou parte de) uma conceitualização. Esta conceitualização poderia ser implícita, existente somente na cabeça de alguém, ou embutida em um módulo de software. Uma ontologia explícita pode ter diversas formas, mas sempre inclui um vocabulário de termos e alguma especificação de seu significado [FILETO, 2011].

Segundo [FILETO, 2011] já a compartilhada é a representação de um grupo de pessoas, em certo domínio, que devem concordar sobre o significado de entidades e relações expressas em uma ontologia. O grau de formalização do vocabulário e dos significados especificados pode variar consideravelmente sendo:

- Altamente informal: expressa em linguagem natural;
- Semi-informal: expressa em uma forma restrita e estruturada de linguagem natural;
- Semi-formal: expressa em uma linguagem artificial e formalmente definida;

- Rigorosamente formal: expressa em termos meticulosamente definidos com semântica formal, teoremas e provas de propriedades como consistência e completude.

Em um contexto geral estes campos podem ser encontrados em Ontologias classificadas em quatro tipos:

- *Ontologia de domínio*: esta modela um domínio de conhecimento específico (exemplo: medicina, biologia, engenharia, direito);
- *Ontologia de aplicação*: esta modela objetos de uma aplicação (exemplo: um modelo de classes);
- *Ontologia de alto nível*: esta modela objetos comuns que são geralmente aplicáveis a uma grande variedade de situações (exemplo: padrão de metadados Dublin Core);
- *Meta-ontologia*: esta define meta-classes que determinam características de outras ontologias ou modelos em nível de abstração mais baixo.

Podendo ser aplicadas em diversos contextos, tais como:

- Buscas semânticas: Ex.: Estado (São Paulo), Cidade (São Paulo);
- Interoperabilidade;
- Agentes móveis;
- Reuso e composição de recursos;
- Buscas semânticas para Objetos de Aprendizagem;
- Buscas semânticas para serviços Web;

A aplicação destas ontologias pode se dar entre pessoas e organização (comunicação), entre sistemas (interoperabilidade) e em componentes reutilizáveis que necessitem confiabilidade e especificação (engenharia de sistemas). Outras distinções podem ser importantes dentro de cada categoria citada, tais como a natureza do software, tipo de usuários e domínio de aplicação.

Com relação a sua estrutura, dentre os principais paradigmas de modelagem de ontologia, temos os baseados nas noções de Frames e na semântica das Lógicas Descritivas (DL – Description Logics). Estes dois paradigmas apresentam semelhanças em seu modelo de construção: ambos são construídos sobre a noção de classes, representando conceitos do discurso do domínio; classes possuem instâncias; propriedades (slots) descrevem atributos dessas classes e relacionamentos entre elas; restrições e facetas expressam restrições sobre os valores das propriedades/slots [XAVIER, 2010].

Quanto à formalização dos domínios de ensino e das aplicações educacionais, espera-se que sejam representados através de ontologias OWL (Ontology Web Language).

Tanto o projeto OBAA-MILOS, quanto o sistema Heráclito, têm em sua modelagem o uso de ontologias na linguagem OWL [W3C, 2010]. Esta linguagem é uma linguagem ontológica, composta por três versões com níveis de expressividade diferentes, desenvolvida pelo W3C (*World Wide Web Consortium*). A construção da

OWL foi motivada pelo importante papel previsto para as ontologias na Web Semântica (na disponibilização de vocabulários precisamente definidos e processáveis por máquinas⁸ para utilização em anotações semânticas) e também pelo reconhecimento das linguagens existentes na época, RDF (*Resource Definition Format*) e RDF-S não serem expressivas o suficiente.

A linguagem OWL busca lidar com a heterogeneidade das representações de conhecimento através de uma linguagem padronizada para desenvolvê-la. Assim a escolha de se justifica pela flexibilidade de utilização e ampla adoção pela comunidade científica. Além disso, a linguagem OWL é uma recomendação da W3C para a especificação de ontologias de acordo com a iniciativa da Web Semântica.

Para a construção das ontologias utilizadas neste trabalho adotou-se a ferramenta Protégé. O Protégé é uma ferramenta gratuita, de código aberto e por disponibilizar um conjunto de ferramentas para a construção de modelos de domínios e aplicações baseadas em conhecimento com ontologias. Seu editor Protégé-owl é fortemente integrado a API Jena (API Java para RDF e OWL), mas oferece uma API própria, também desenvolvida em Java, que permite a criação e/ou adaptação de funcionalidades. A plataforma Protégé foi desenvolvida como uma coleção de plug-ins, que podem ser alterados para a adaptação do comportamento da plataforma [PROTÉGÉ, 2011].

As ontologias de domínio de ensino e estratégias de aprendizagem foram construídas através desta ferramenta e são apresentadas na seção 5.3.

3.12 Estratégias e Táticas Pedagógicas

Aprender é tomar conhecimento de si. É tornar-se capaz de (algo), graças ao estudo, a observação, a experiência. É tomar conhecimento de algo e retê-lo na memória [FERREIRA, 2010, p. 57]. É uma atividade que acontece com a pessoa e que é realizada pela pessoa. Em outras palavras, aprender é um processo onde o aprendiz aprende um ofício ou arte e este aprendizado torna-se o ato, ou processo ou efeito de aprender.

De acordo com Bordenave e Pereira [1999, p.25] a aprendizagem é um processo integrado no qual toda a pessoa (intelecto, afetividade, sistema muscular) se mobiliza de maneira orgânica.

Assim como aprender, ensinar também é um processo, mas não necessariamente o mesmo processo. Ensinar é transmitir conhecimentos, fazer uso de métodos para chegar aos objetivos propostos [FERREIRA, 2010, p. 291]. Ensinar, por sua vez, é um processo de facilitar que uma ou mais pessoas aprendam e cresçam intelectual e moralmente, fornecendo-lhes situações planejadas de tal modo que quem aprende vive as experiências necessárias para que se produzam neles as modificações desejadas, de uma maneira mais ou menos estável [BORDENAVE e PEREIRA, 1999].

⁸ A ideia de “processável por máquinas” significa que a informação em questão explicitamente preparada para inferência computacional e reutilização por várias aplicações.

No propósito de ensinar o aluno a aprender, vários fatores, como a motivação e o conhecimento prévio do aluno, a relação do professor com o aluno, sua atitude em relação ao assunto a ser ensinado, além da estrutura e organização do conteúdo de ensino, intervêm na operação de ensino.

Sejam quais forem os objetivos estabelecidos para alcançar o efetivo ensino, o próximo passo é definir o caminho ou estratégia. Isso facilita a passagem dos alunos da situação em que se encontram até alcançarem os objetivos fixados, tanto de natureza técnico-profissional como os de desenvolvimento individual como pessoa humana e como agente transformador de sua sociedade.

Definir o caminho ou estratégia para ensinar é um ponto chave que em conjunto com os fatores que afetam o processo de ensino pode ser denominado de estratégia pedagógica, estratégia de ensino ou estratégia de ensino-aprendizagem. Entretanto, dado a abordagem sócio histórica do presente trabalho [PASSERINO, GLUZ e VICARI, 2007] esse tipo de estratégia é visto como uma estratégia de mediação para o processo de ensino-aprendizagem, que é composta de ações pedagógicas ou didáticas que possam ajudar na evolução desse processo.

O termo estratégia pode ser entendido, de maneira geral, como um plano, uma construção ou elaboração, enquanto que as ações que compõem esse plano propriamente ditas são denominadas de táticas. A estratégia é um processo cognitivo que tem como objetivo alcançar objetivos específicos, aplicando meios disponíveis ou explorando condições favoráveis com vistas a alcançar estes objetivos [FERREIRA, 2010, p. 321].

Na aprendizagem estas estratégias são técnicas ou métodos que os alunos usam para adquirir informação com o propósito de facilitar a aquisição, o armazenamento e/ou a utilização da informação, podendo ser vistas como esquemas de planos que definem formas de tratar o conteúdo. Em outras palavras, as estratégias de aprendizagem podem ser consideradas como qualquer procedimento adotado para a realização de uma determinada tarefa [SEIXAS, 2005].

Bordenave e Pereira [1999] apresentam uma variedade de possíveis técnicas, resumidas na Tabela 8.

Tabela 8 – Técnicas de ensino

OBJETIVO EDUCATIVO	TÉCNICAS ADEQUADAS
Dar aos alunos numa classe numerosa ocasião de participar, quer formulando perguntas, quer formulando respostas e perguntas, ou expressando opiniões e posições.	Phillips 66 Díade Grupos de cochicho Times de observação
Aprofundar a discussão de um tema ou problema, chegando a conclusões (consenso).	Grupos pequenos Grupos de integração vertical/horizontal
Desenvolver capacidade de observações e crítica do desempenho em grupo.	Grupos de verbalização e observação
Produzir grande quantidade de ideias em prazo curto, com alto grau de originalidade e desinibição.	Tempestade cerebral
Conseguir que todos os participantes expressem suas opiniões.	Pergunta circular
Estudar e analisar um tema por um pequeno grupo de especialistas ou pessoas interessadas, para ilustração dos	Painel

demais.	
Apresentar diversos aspectos de um mesmo tema ou problema, para fornecer informações e esclarecer conceitos.	Simpósio
Meditar coletivamente sobre um tema importante, com ajuda de obras e pessoas para consulta, a fim de chegar a uma tomada de posição.	Reflexão ou círculo de estudos
Enfrentar pessoas com ideias opostas para que de sua confrontação surjam subsídios para orientar as opiniões do público presente.	Debate Painel de oposição
Desenvolver a capacidade analítica e preparar-se para saber enfrentar situações complexas, mediante o estudo coletivo de situações reais ou fictícias.	Estudo de casos
Desenvolver a empatia ou a capacidade de desempenhar os papéis de outros e de analisar situações de conflito.	Dramatização (Sociodrama, Psicodrama)
Investigar diversos aspectos de um problema e colocar os resultados em comum.	Seminário
Desenvolver a capacidade de estudar um problema em equipe, de forma sistemática.	Estudo orientado em equipes
Debilitar o dogmatismo e aumentar a flexibilidade mental mediante o reconhecimento da diversidade de interpretações sobre um mesmo assunto.	Diálogos sucessivos
OBJETIVO EDUCATIVO	TÉCNICAS ADEQUADAS
Aprender a trabalhar em equipe na solução de problemas.	Método de projetos
Aprender fazendo e resolvendo problemas com a intervenção de recursos humanos competentes e o benefício da discussão em grupo.	Oficina ou laboratório (Workshop)

Fonte: Bordenave e Pereira [1999].

O uso e aplicações das técnicas de ensino listadas acima dependem basicamente da função educativa que se quer realizar, podendo ser usadas em combinação umas com as outras, tais como o Simpósio e o Seminário. Outra forma de combinação é o do professor organizar uma estratégia de ensino que combine atividades individuais dos alunos com atividades em grupo.

Com relação a estratégias adequadas em ambientes informatizados, onde um procedimento (atividade, exercício, etc.) está em desenvolvimento suportado por computador, os autores Ohlsson [1987] e Breuker [1988] propõem algumas táticas a serem utilizadas por um agente tutor inteligente⁹ no ensino de conteúdos.

Essas táticas consolidam e operacionalizam as estratégias, gerando o comportamento do agente tutor frente ao estudante. A Tabela 9 apresenta estas táticas.

Tabela 9 – Táticas de agentes tutores

APRESENTAÇÃO DO OBJETIVO
Tática que consiste na definição de termos necessários para a descrição do processo computacional necessário

⁹ Um agente tutor tem como principal objetivo o de facilitar ou intermediar junto ao aluno, ferramentas, pontos de vista, reflexões, que o levem a refletir sobre suas ações e conseqüentemente reelaborar suas hipóteses e procedimentos.

para inspirar o aluno à recordação;
APRESENTAÇÃO DE PRECURSORES
Um precursor (aquele que vai adiante) é uma habilidade que um aluno deve ter antes de tentar aprender um processo. O tutor relembra como uma situação semelhante já foi resolvida. O processo de lembrança dos passos utilizados na solução de problemas similares pode ser feito através de revisão e da marcação dos passos que já são familiares ao aluno, reavivando seus conhecimentos prévios.
REPRESENTAÇÃO DE PROPÓSITOS
Tática que consiste em deixar claro para o aluno um determinado procedimento, ou seja, o que é possível alcançar através de um procedimento. O tutor pode descrever o propósito de um novo procedimento, levando em conta o seu objetivo e o resultado esperado com sua aplicação. Os procedimentos já utilizados podem ser exibidos como contra exemplos em face ao novo procedimento oferecido.
APRESENTAÇÃO DE JUSTIFICATIVAS
O tutor pode justificar um procedimento através de anotações, relacionando cada passo executado na busca da solução de um problema. Pode fazer uso de procedimentos equivalentes, desde que já tenham sido descritos e justificados.
RELACIONAMENTO DE ERROS (BUGS)
Essa tática é aplicada de forma intencional de modo a marcar o passo errado gerado durante a execução de um processo, ou seja, o conhecimento é todo estruturado e associado a uma coleção de erros que permitam identificar os pontos onde o aluno apresenta dificuldades ou falsas concepções.
RELACIONAMENTO DE SOLUÇÃO DO ALUNO
O tutor pode induzir o aluno a fazer uma auto avaliação, ou seja, fazer uma verificação dos passos seguidos, descobrir as possíveis causas do erro, buscar explicações e identificar o erro ocorrido.

Fonte: Ohlsson [1987].

Nessas táticas é possível observar concepções pedagógicas mais voltadas para o aluno, que é levado a pensar sobre suas ações, na forma de um pensamento reflexivo, com uma retroalimentação de seu trabalho, através do mapeamento de seus passos.

Com isso, a aplicação do uso das estratégias pedagógicas em ambientes computacionais deve perpassar a produção e orientação de atividades didáticas que despertem o entusiasmo pela aprendizagem, que orientem a produção de conhecimento, que problematize situações e instaure o diálogo. Essa concepção de estratégias pedagógicas pode ser aplicada em ambas as modalidades de ensino favorecendo o diálogo e a participação ativa dos alunos.

3.13 Estratégias e Táticas para Ensino de Lógica

Cabe ressaltar aqui que esta disciplina é o objeto de estudo deste trabalho e a pesquisa realizada teve como principal foco a aplicação das estratégias de ensino-aprendizagem voltadas ao ensino de Lógica por meio de uma ferramenta tecnológica de apoio ao professor e ao aluno.

Nesse sentido a Lógica enquanto disciplina acadêmica tem suas particularidades. Apesar de estar tradicionalmente relacionada com a Filosofia e o Direito, ela hoje desempenha um papel extremamente importante em outras áreas como, por exemplo, as que compreendem as Ciências Exatas e Tecnológicas.

No que se refere ao ensino da Lógica, Silvestre [2010] traz como estratégias de aprendizagens, apenas estratégias básicas compostas pela exposição do conteúdo, com exemplos específicos e a resolução de exercícios propostos.

Já em Baronett [2009] as estratégias de aprendizagem trazem temas expositivos de conteúdos, seguidos de exemplos e exercícios dirigidos a cada capítulo. A resolução destes problemas combina três tipos de conteúdo: apresentação, estrutura das fórmulas e todo um mecanismo de táticas e estratégias para inferências de regras, seguidas de exercícios para fixação. Ao optarmos por uma estratégia do tipo mais trivial em um ambiente computacional é possível inferir uma dialética mais facilitada. Assim ao invocar uma explicação do agente tutor mediador, este pode propor sugestões, apresentar help ou ajuda, o aluno pode pedir o próximo passo, por onde seguir, quais opções seguir e/ou qual regra é mais adequada na resolução de um problema ou procedimento a ser tomado.

O livro didático de Lógica para Computação de Gluz e Py [2010] abrange os tópicos vistos em uma disciplina de um semestre, para cursos de Ciência da Computação. Com relação às estratégias de aprendizagem, o livro é baseado em um método dialético aplicado em aula presencial, complementado pela execução e discussão de exemplos e exercícios. Para reforçar os conceitos apresentados em cada capítulo, o livro contém no final uma seção com diversos exercícios em diferentes níveis de aprendizado.

3.14 Planos Gerenciais de Ensino Didático aplicados no Ensino de Lógica

Segundo Ferreira [2010], um plano pode ser um conjunto de métodos e medidas para a execução de um empreendimento ou projeto. Trata-se de um roteiro, um instrumento de referência.

O ato de planejar envolve um exercício da razão e da sensibilidade, que engloba atividades de maior ou menor complexidade no cotidiano de trabalho e, sob essa ótica, propicia a construção de planos para enfrentar situações atuais ou futuras.

O planejamento pode ser definido como um importante instrumento que favorece fazer escolhas e a elaboração de planos que ajudam a enfrentar os processos de mudança. Assim, compreende um conjunto de conhecimentos teóricos e práticos ordenados de modo a possibilitar a interação com a realidade, programar as estratégias e as ações necessárias, para alcançar os objetivos e as metas pré-estabelecidas. Portanto, é um processo contínuo que permite uma postura ativa por parte de quem os organiza com relação àqueles que fazem parte da ação e o meio em que se atua.

Quando o foco é o ensino Menegolla e Sant'Anna [2010] afirmam que alguns pontos importantes em um planejamento são necessários. São eles:

- ✓ O professor precisa prever as ações e procedimentos que serão realizados com seus alunos;
- ✓ O professor precisa organizar as atividades discentes visando atingir os objetivos educacionais estabelecidos, tais como:
 - Selecionar o material e organizar os conteúdos mais significativos para seus alunos;
 - Organizar os conteúdos de forma lógica, obedecendo à estrutura

da disciplina;

- Selecionar os melhores procedimentos e recursos, de forma a desencadear um ensino mais eficiente, orientando o professor em como deve agir;
- ✓ O professor precisa agir com maior segurança na sala de aula, de forma a evitar a improvisação, a repetição e a rotina no ensino;
- ✓ O professor precisa facilitar uma maior integração com as mais diversas experiências de aprendizagem;
- ✓ O professor precisa ter uma visão global de toda a ação docente e discente para que possam tomar decisões de forma cooperativa e participativa.

Ao fazer uso do planejamento o professor antecipa, tradicionalmente, de forma organizada todas as etapas do seu trabalho e com isso identifica os objetivos que pretende atingir, indicando conteúdos que serão desenvolvidos, selecionando procedimentos a serem utilizados como estratégia de ação, prevendo quais os instrumentos que fará uso para avaliar o processo de ensino dos alunos.

O Sistema Heráclito é composto por planos gerenciais de ensino que foram desenvolvidos para gerenciar determinadas ações dos alunos, previstas com base nos passos a serem realizados no contexto do desenvolvimento das atividades propostas. Em cada planejamento destas ações, estratégias de aprendizagem são selecionadas e aplicadas de acordo com a necessidade da ação realizada pelo aluno, ou seja, com base nas ações por este tomadas, o sistema entende que para uma melhor compreensão de resolução do exercício, algumas medidas estratégicas podem auxiliar neste processo.

4. TRABALHOS RELACIONADOS

Neste capítulo são apresentados e analisados trabalhos relacionados ao tema da dissertação. O contexto mais geral de pesquisa da dissertação está relacionado com as pesquisas sobre aplicações das tecnologias de objetos de aprendizagem (OA) e de ambientes digitais ou virtuais de ensino (AVAs). Nesse sentido, realizou-se um estudo sobre esses temas nas produções científicas recentes dos últimos quatro anos (2008, 2009, 2010 e 2011).

Em termos específicos a presente pesquisa está diretamente relacionada com a criação e aplicação de tutores inteligentes no ensino de Lógica, particularmente restrito ao tópico de ensino de dedução na Lógica Proposicional.

Na próxima seção são apresentados os resultados da análise dos trabalhos relacionados ao tema geral da dissertação. A seção seguinte mostra a análise dos trabalhos sobre tutores para Lógica.

4.1. O Contexto Geral da Pesquisa

Como parte da análise do contexto geral da pesquisa, em primeiro lugar foi realizado um amplo estudo identificando os trabalhos recentes sobre o uso das tecnologias de OAs e de AVAs, tanto em casos separados, onde apenas constavam dados e informações sobre o uso de um tipo de tecnologia, quanto em casos combinados, onde aparecem ambas tecnologias, sendo descritos como foram utilizadas.

Posteriormente foi selecionado um conjunto de trabalhos significativos sobre a aplicação dessas tecnologias e feita uma análise aprofundada dos temas tratados nesses trabalhos.

Para tanto foram analisados congressos, seminários, revistas, *workshops* e demais eventos na área de Informática e Educação dos últimos quatro anos (2008, 2009, 2010 e 2011) na busca destas informações. Foram incluídas nessas análises trabalhos dos seguintes eventos e revistas:

- CSBC (Congresso da Sociedade Brasileira de Computação)
- LACLO (Comunidade Latinoamericana de Objetos de Aprendizagem)
- MoodleMoot Brasil (Conferência dedicada aos usuários, desenvolvedores e administradores do Moodle).
- RBIE (Revista Brasileira de Informática na Educação)
- RENOTE (Revista Novas Tecnologias na Educação)
- SBIE (Simpósio Brasileiro de Informática na Educação)
- TISE (Congresso Internacional de Informática Educativa)
- WIE (Workshop de Informática na Educação)

Também foram consultados materiais disponíveis na biblioteca digital da SBC (bibliotecadigital.sbc.org.br) que possui em seu acervo: periódicos, livros, *workshops*, entre outros.

A Tabela 10 mostra os últimos quatro anos de publicações com os números de trabalhos relacionados à OAs e AVAs encontrados por eventos (quantitativamente).

Tabela 10 – Trabalhos sobre OAs e AVAs encontrados nos principais eventos de Computação e Informática na Educação

Publicação	2008	2009	2010	2011
CSBC	7	4	6	13
LACLO	-	23	14	35
MoodleMoot Brasil	27	26	35	51
RBIE	5	3	6	9
SBIE	19	16	25	58
TISE	6	2	12	12
WIE	16	16	12	11
RENOTE	32	23	17	48
TOTALIZANDO	112	113	127	237

Observa-se, na Tabela 10, um aumento gradativo de trabalhos publicados nesta área. Estes artigos foram classificados em seis categorias:

- Aplicações técnicas de OAs: Os artigos com aplicações técnicas foram classificados nesta categoria por estarem diretamente ligados às questões de padronização, classificação, armazenamento, recuperação, transmissão e/ou reutilização dos OAs. Nesse sentido seu foco é voltado ao desenvolvimento e criação de novas ferramentas, fazendo uso de tecnologias diversas em seu contexto.
- Aplicações pedagógicas de OAs: Os artigos com aplicações pedagógicas foram classificados nesta categoria por estarem diretamente ligados às questões de interatividade, autonomia, cooperação, cognição e/ou afetividade. Nesse sentido tem seu foco voltado para os métodos e propostas de usabilidade, sempre com vistas a facilitar o trabalho dos professores e alunos no processo de ensino e aprendizagem.
- Aplicações técnicas de AVAs: Esta categoria faz uso do mesmo conceito utilizado nas aplicações técnicas de OAs.
- Aplicações pedagógicas de AVAs: Esta categoria faz uso do mesmo conceito utilizado nas aplicações pedagógicas de OAs.
- Aplicações técnicas de OAs+AVAs combinados: Essa categoria faz uso de ambas às ferramentas OAs e AVAs em seu contexto e utiliza o conceito das aplicações técnicas de OAs e AVAs.
- Aplicações pedagógicas de OAs+AVAs combinados: Essa categoria faz uso de ambas às ferramentas OAs e AVAs em seu contexto e utiliza o conceito das aplicações pedagógicas de OAs e AVAs.

A Figura 10 mostra o número de artigos encontrados para cada uma das categorias citadas nos últimos quatro anos.

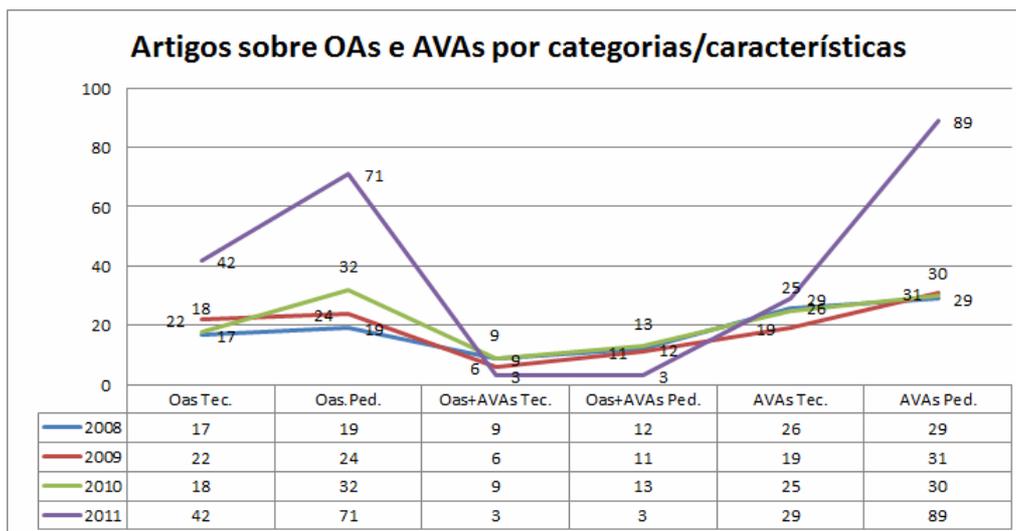


Figura 10 – Trabalhos relacionados ao uso de OAs e AVAs divididos por categoria.

Observa-se na Figura 10 uma maior preponderância de trabalhos exclusivos sobre AVA (AVAs Tec. + AVAs Ped. = 278 trabalhos nos quatro anos) em comparação aos trabalhos cujo foco está restrito aos OA (OAs Tec. + OAs Ped. = 245 trabalhos nos quatro anos). Menos trabalhos ainda foram encontrados nas áreas conjuntas de OA e AVA (OAs e AVAs Tec. + OAs e AVAs Ped = apenas 66 trabalhos nos quatro anos), indicando um campo ainda inicial de pesquisa.

Uma tendência que pode ser percebida é que após uma distribuição igualitária dos trabalhos que pesquisam o uso conjunto de OAs+AVAs até 2010 (21 trabalhos sobre este tema em 2008, 17 trabalhos em 2009 e 21 trabalhos em 2010), houve uma diminuição significativa nos trabalhos que integram ambas as tecnologias em 2011, com apenas 6 trabalhos.

Outra tendência que pode ser observada é um aumento nos trabalhos que pesquisam aplicações técnicas de AVAs (99 trabalhos) sobre aplicações pedagógicas de AVA (179 trabalhos) durante o período analisado. Após um pico de 31 trabalhos em 2009, combinada com uma tendência de incremento de trabalhos de caráter técnico sobre AVA, encontrou-se uma distribuição quase idêntica em 2010 (30 trabalhos) e um aumento em 2011 mais que o dobro do ano de 2010, chegando a 89 trabalhos. Por outro lado no caso dos OA (técnicos e pedagógicos) e de AVAs Ped. pode-se perceber um elevado aumento de produção (quase o dobro de trabalhos se comparado com o ano anterior de 2010).

Dos trabalhos analisados, 15 artigos foram escolhidos para uma análise comparativa mais detalhada. Os critérios usados nesta comparação foram elaborados com base em características elementares que um OA deve abordar (também descrito no capítulo 3, seção 3.2). São eles:

(1) Domínio de Ensino: indica se os OA utilizados no trabalho estão restritos a algum domínio específico de ensino ou não;

(2) Uso do OA: Técnico: o OA é essencialmente um veículo para pesquisa de aspectos técnicos resultantes do de OA e tecnologias educacionais similares; Pedagógico: o OA está efetivamente orientado a ser usado em atividades usuais de ensino;

(3) AVA: AVA onde os OA estão sendo disponibilizados;

(4) Objetivo Pedagógico: define o objetivo pedagógico da aplicação dos OA no contexto de um AVA;

(5) Abordagem Pedagógica: define qual tipo de escola pedagógica-psicológica de pensamento (Construtivista, Sócio Interacionista, Behaviorista, entre outras) está servindo de base teórica para o estudo;

(6) Ênfase Tecnológica: qual a ênfase do estudo, em termos das diversas tecnologias computacionais e educacionais existentes;

(7) Foco da Pesquisa: declara os temas específicos da pesquisa.

Dos trabalhos analisados 15 foram selecionados dos respectivos eventos: SBIE (4 trabalhos), MoodleMoot Brasil (4 trabalho), RBIE (2 trabalhos), TISE (1 trabalho), WIE (2 trabalho) e LACLO (2 trabalhos) no período de 2008 a 2011, com base nas características elementares que um OA deve abordar e principalmente por combinarem de forma clara as tecnologias de OA e AVA. A Tabela 11 mostra o resultado desta comparação.

Tabela 11 – Análise comparativa sobre OAs e AVAs

	Domínio de Ensino	Uso do OA	AVA	Objetivo Pedagógico	Abordagem Pedagógica	Ênfase Tecnológica	Foco da Pesquisa
1 ¹⁰	Não restrito	Técnico	Moodle	Análise de estratégias pedagógicas em cursos online	Não definida	Agentes, ontologias, OAs, AVAs, padrões, modelo computacional pedagógico	Framework para OAs, múltiplas plataformas e tomada de decisão por agentes
2 ¹¹	Educação com suporte as TICs	Pedagógico	Moodle	Capacitação técnica e pedagógica de docentes	Não definida	Designer, AVAs, OAs, ferramentas para edição de imagem, vídeo e som	Docentes, OAs, cursos e aulas
3 ¹²	Não restrito	Não definido	Moodle	Compartilhamento de OAs entre instituições e/ou professores	Não definida	Rede P2P, de compartilhamento/plugins, AVA, OAs	OAs, AVA, docentes e instituições de ensino
4 ¹³	Não restrito	Não definido	Moodle	Auxílio à busca de OAs em ambiente local	Não definida	OAs, AVA, técnicas de filtragem colaborativa e baseada em conteúdo	Alunos da instituição
5 ¹⁴	Informática na Educação Especial	Pedagógico	Edukito	Impulsionar uma visão pedagógica que coloca o sujeito aprendiz no centro do processo educativo	Concepção epistemológica sócio Interacionista	Oficina multimídia acessível, simulação de interface	Inclusão digital, social e de convivência entre PNEs

¹⁰ O artigo original pode ser encontrado nos anais do MoodleMoot Brasil do ano de 2008.

¹¹ O artigo original pode ser encontrado nos anais do LACLO do ano de 2010.

¹² O artigo original pode ser encontrado nos anais do MoodleMoot Brasil do ano de 2009.

¹³ O artigo original pode ser encontrado nos anais do LACLO do ano de 2009.

¹⁴ O artigo original pode ser encontrado nos anais do TISE do ano de 2008.

	Domínio de Ensino	Uso do OA	AVA	Objetivo Pedagógico	Abordagem Pedagógica	Ênfase Tecnológica	Foco da Pesquisa
6 ¹⁵	Matemática	Pedagógico	Moodle	Computador como meio intelectual criativo	Não definida	OAs, AVAs, <i>flash</i>	Alunos e uso de TICs para apoio aos conteúdos
7 ¹⁶	Física	Pedagógico	Moodle	Melhorar o ensino de instituições do ensino médio da rede estadual/municipal	Não definida	OAs, AVAs, <i>Weblab</i>	Alunos, novas TICs, maior interação do aprendiz.
8 ¹⁷	Física	Pedagógico	Moodle	Como aprender a utilizar as atividades do Moodle	Não definida	OAs, AVA, <i>Hot Potatoes</i>	Recursos do <i>Moodle</i> associado aos OAs
9 ¹⁸	Não restrito	Não definido	LOCAL	Autoaprendizagens com repositório GOAL em ambiente ubíquo	Não definida	OAs, Mobilidade, Ambientes de educação Ubíqua	Interoperabilidade de repositórios e acesso a OAs
10 ¹⁹	Restrito ao EJA	Não definido	Moodle	Reconstrução de materiais instrucionais em papel na forma de OAs	Metodologia SESIEduca – Interacionista-constructivista-sistêmica	OAs, AVA, ferramentas de texto, áudio, vídeo e som	Construção de OAs de formatos diversos pelos alunos
11 ²⁰	Não restrito	Pedagógico	Moodle	Capacitação técnica e pedagógica de docentes em EAD	Não definida	OAs e AVA	Aspectos relativos à gestão da EAD e ao conceito de arquiteturas pedagógicas na EAD
12 ²¹	Não restrito	Técnico	Moodle	Reutilização de OAs em repositórios	Não definida	OAs, AVAs, Metadados, Padrão de Metadados e Ferramenta Web	Ferramenta Web para o preenchimento e conversão de metadados dos OAs para o padrão OBAA
13 ²²	Algoritmos Computacionais	Técnico	Rooda	Avaliar o modelo afetivo do aluno com base no modelo de appraisal de Scherer	Os pressupostos de Scherer e Piaget	AVA, OA e Redes Bayesinas	Inventariar aspectos afetivos à medida que se desenvolve uma atividade de aprendizagem por meio de OAs.
14 ²³	Lógica de Programação	Técnico	Moodle	Apoiar à metodologia de ensino de programação	Não definida	AVA, OA e PHP	Implementar uma metodologia capaz de tratar cada aluno com seu ritmo próprio de aprendizagem
15 ²⁴	Não restrito	Técnico	Tidia e Moodle	Uso de jogos no processo de aprendizagem	Não definida	Tecnologias Web, AVA, OA e técnica de folksonomia.	Criação de estruturas taxonômicas e folksonômicas aplicada ao contexto dos jogos, provendo categorizações e classificações

¹⁵ O artigo original pode ser encontrado nos anais do RBIE 2009.

¹⁶ O artigo original pode ser encontrado nos anais do MoodleMoot Brasil do ano de 2009.

¹⁷ O artigo original pode ser encontrado nos anais do MoodleMoot Brasil do ano de 2010.

¹⁸ O artigo original pode ser encontrado nos anais do SBIE do ano de 2009.

¹⁹ O artigo original pode ser encontrado nos anais do WIE do ano de 2010.

²⁰ O artigo original pode ser encontrado nos anais do RBIE 2010.

²¹ O artigo original pode ser encontrado nos anais do SBIE do ano de 2010.

²² O artigo original pode ser encontrado nos anais do SBIE do ano de 2011.

²³ O artigo original pode ser encontrado nos anais do SBIE do ano de 2011.

²⁴ O artigo original pode ser encontrado nos anais do WIE do ano de 2011.

Pode-se verificar na Tabela 11 que os trabalhos analisados são bastante variados. Estes trabalhos apresentam diversas características pedagógicas e técnicas, compreendendo variados domínios de ensino, com objetivos pedagógicos e abordagens pedagógicas distintas. Os trabalhos fazem uso de diversas tecnologias (que vão desde um simples editor de texto a ambientes complexos de software como programação e sistemas multiagentes), sempre focando meios e formas de utilização conjunta de OAs e AVAs associados às ferramentas tecnológicas.

Uma constatação inicial é que ainda não se firmou o importante hábito de explicitar o referencial pedagógico ou psicológico que fundamenta o processo pedagógico destes trabalhos, que foi explicitamente definido em apenas três trabalhos. Também se confirma a preponderância do *Moodle* como AVA de suporte e a preponderância dos estudos envolvendo aspectos pedagógicos sobre os técnicos.

Em todos os trabalhos analisados o principal papel dos AVAs foi intermediar como repositório e ferramenta de apoio ao ensino, à utilização dos OAs por parte dos professores e alunos. Porém, analisando-se com mais detalhe a ênfase tecnológica e o foco da pesquisa dos trabalhos foi possível observar que no caso dos sistemas com suporte de agentes pedagógicos (casos 1, 6 e 13), proveram o acompanhamento de atividades. Os AVAs também forneceram algum suporte para trabalho colaborativo, mas sem ênfase específica em nenhum dos casos, incluindo em alguns casos o suporte para trabalho individual (principalmente casos 5, 9 e 14).

Isso indica que continua sendo fundamental desenvolver para os AVAs atuais um suporte integrado, universal e interoperável para as várias funcionalidades técnicas e pedagógicas relacionadas aos OAs:

(1) Disponibilização de conteúdos (repositório): essa é uma funcionalidade básica do AVA, provendo materiais e conteúdos educacionais representados na forma de OAs;

(2) Acompanhamento/monitoração de atividades didático-pedagógico: somente disponibilizar o material na forma de OA não é suficiente, também seria importante incorporar o uso pedagógico-didático deste OA no contexto dos mecanismos de acompanhamento do AVA;

(3) Avaliação/teste (*feedback* para o professor/aluno): mesma situação do caso anterior, não faz sentido deixar o OA apenas como um “pacote” isolado que apenas é disponibilizado ao aluno;

(4) Trabalho colaborativo (aprendizagem e estudo em grupo): os AVAs geralmente oferecem ferramentas para interação e trabalho colaborativos, mas seria importante integrar as facilidades oferecidas por estas ferramentas com os OAs;

(5) Autoaprendizagem (estudo individual): uma AVA oferece um ambiente para o ensino que pode incentivar a autoaprendizagem, porém, apenas disponibilizar OAs como “pacotes” fechados, sem uma maior interação e integração com o resto do ambiente, pode, na verdade, desestimular o auto aprendizado.

Os resultados desse estudo, em particular os itens (2), (3) e (5) acima, vão claramente ao encontro dos objetivos do Sistema Heráclito, sendo proposto neste trabalho. O objetivo principal do Heráclito está totalmente alinhado ao item (2), de fornecer o acompanhamento didático-pedagógico, no domínio da Lógica Proposicional, através de um sistema de ensino que integre de forma transparente ao usuário, as capacidades dos OAs aos AVAs. O Heráclito também pode ser utilizado em atividades de avaliação (item (3)), contribuindo para processos de autoaprendizagem (item (5)) e auto avaliação empreendidos de forma autônoma pelo aluno.

Assim, espera-se que o Sistema Heráclito possa contribuir no atendimento de necessidades muito importantes e atuais das pesquisas sobre OAs e AVA, ainda não cobertas pelas Tecnologias Educacionais disponíveis atualmente.

4.2. Ferramentas de Apoio ao Ensino de Lógica

Esta seção apresenta um estudo do estado da arte em termos de ferramentas de apoio ao ensino de Lógica, incluindo provadores de teoremas, verificadores/editores de provas formais e sistemas que fazem o uso de tutores inteligente no contexto de Lógica.

O presente estudo teve sua ênfase na identificação de sistemas similares ao Sistema Heráclito em termos de tecnologias empregadas e também aos objetivos técnicos e científicos.

Em termos educacionais, os sistemas mais similares ao Heráclito são as ferramentas de apoio ao ensino de Lógica e que fazem uso de sistemas tutores inteligentes. As ferramentas encontradas na literatura foram as seguintes:

- ✓ Proofs Tutorial – O Proofs Tutorial é considerado uma ferramenta de aprendizagem auxiliada por computador e tem sido usado para a prática de provas em cursos de nível superior que contemplam a disciplina de matemática discreta. No Proofs Tutorial, os alunos recebem um conjunto de 10 problemas que vão desde simples aplicações lógicas de equivalência a provas de inferência mais complexas. A resolução destes problemas é feita em linhas consecutivas de uma prova, que consistem em quatro partes: a instrução, linhas de referência, o axioma usado, e as substituições que permitem que o axioma possa ser aplicado. Após a inferência, os quatro passos são verificados e se um erro é encontrado, uma mensagem de aviso é mostrada. A linha com erro é automaticamente excluída (mas o sistema guarda o log para posterior análise) e o aluno volta a resolver o problema [BARNES; STAMPER, 2012].
- ✓ Logic-ITA – O Logic-ITA é um assistente de ensino/aprendizagem de lógica proposicional com sistema inteligente. Esse assistente de ensino fornece auxílio tanto a alunos quanto a professores. O sistema incorpora três ferramentas: o Tutor Logic, o LT-Configurator e o LT Analyser. O Tutor Logic é um sistema de tutoria inteligente destinado aos alunos que permite a prática de provas formais, com exercícios prévios configurados e personalizados enquanto um feedback de retorno lhes é dado. As outras duas ferramentas são dedicadas para o professor. O LT-Configurator faz as

configurações e definições de materiais. O LT-Analyser é para monitorar o progresso da classe e coletar dados [MERCERON; YACEF, 2012].

- ✓ Sistema de tutoria inteligente baseado na Web – Este artigo relata os esforços para desenvolver um sistema de tutoria inteligente para a construção de prova em lógica proposicional. Os relatórios com os dados foram obtidos a partir de uma disciplina de ensino de graduação que compreende o conteúdo de lógica dedutiva na Universidade da Carolina do Norte em Charlotte. Nesta disciplina os alunos usam vários Applets Java para a prática de: instrução de transição de estado, resolução de problemas, análise funcional verdade, a construção de prova, e outros aspectos da lógica proposicional. Este artigo em questão teve dois objetivos: o primeiro deles foi identificar estudantes considerados de risco no estágio no início do semestre, e o segundo é o de gerar uma representação visual dos esforços do aluno na realização das atividades (estes Applets Java podem ser acessados no endereço: <http://itsxserve.uncc.edu/philosophy/tallc/applets/applets.html>) [CROY; BARNES; STAMPER, 2012].
- ✓ KRRT – A junção de suas siglas tem como significado: KR Lógica de Primeira Ordem (LPO) e KRRT (Representação do Conhecimento e Raciocínio Tutor). KRRT é um sistema baseado na Web cujo objetivo principal é ajudar o aluno a aprender linguagem natural em fórmulas (FOL) e como construir provas formais. Formalização é o processo de traduzir os argumentos da linguagem natural à lógica. KRRT foi desenvolvido para ser utilizado na Web. Para acessar o sistema, os alunos devem acessar o endereço: <http://www.cs.us.es/clg/krrt>. No formulário de inscrição o aluno é questionado sobre seu nível inicial de Lógica. Para analisar o progresso do estudante, suas informações sobre os erros frequentemente repetidos bem como sua evolução de aprendizagem são armazenadas no sistema. Esse processo é obtido por meio de um log com registro de todas as atividades realizadas. [ALONSO; ARANDA; MARTN-MATEOS, 2012].
- ✓ Mathematical Logic Tutor – Propositional Calculus (MLT-PC) – Tutor de Lógica Matemática - Cálculo Proposicional (MLT-PC). Este programa se destina a ajudar os alunos de Introdução à Lógica (OIT) e tem como principal objetivo o de proporcionar aos estudantes uma ferramenta de software capaz de auxiliá-los no estudo da primeira parte do curso, dedicada à lógica proposicional. Este sistema foi escrito em Visual Basic. Pode ser baixado gratuitamente a partir da página Web do curso no endereço: <http://www.etse.urv.es/recerca/banzai/toni/ILO>. Em sua versão atual, todas as mensagens são exibidas em catalão [MORENO, 2012].
- ✓ P-Logic Tutor – O objetivo principal do P-Logic Tutor é ensinar aos alunos conceitos fundamentais da lógica proposicional e técnicas de provação de teoremas. Neste artigo o P-Logic Tutor desempenha um duplo papel como um instrumento educativo e um ambiente de pesquisa. Ele apresenta aos alunos os conceitos fundamentais da lógica proposicional e também disponibiliza a prática na resolução de teoremas. O programa também fornece um ambiente no qual é possível acompanhar o aprendizado dos alunos, explorar as questões cognitivas de resolução de problema humano, e investigar as possibilidades de aprendizado. P-Logic Tutor é implementado

em Java e está acessível a partir do endereço: <http://www.cs.wfu.edu/-burg/JavaPackages/> [LUKINS; LEVICKI; BURG, 2012].

A seguir é realizada uma comparação entre os sistemas tutores de Lógica encontrados e o sistema Heráclito. A Tabela 12 demonstra esta comparação.

Tabela 12 – Comparação entre os sistemas tutores de Lógica pesquisados e o sistema Heráclito

Características avaliadas		Ferramentas de apoio ao ensino de Lógica - Sistemas Tutores						
		Proofs Tutorial	Logic-ITA	Sistema de tutoria inteligente baseado na Web (Applets Java)	KRRT	Mathematical Logic Tutor	P-Logic Tutor	Sistema Heráclito
1	Suporte ao conceito de OA	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
2	Uso de demonstração como modelo de aluno	SIM	SIM	SIM	SIM	SIM	SIM	SIM
3	Gera resultado da prova automaticamente	NÃO	NÃO	NÃO	NÃO	SIM	NÃO	SIM
4	Completa provas parciais	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
5	Possibilidade de gerar o próximo passo	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
6	Permite que o aluno reflita sobre o erro	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
7	Suporte para AVA	NÃO	NÃO	SIM	NÃO	NÃO	NÃO	SIM

Os critérios usados nesta comparação foram elaborados com base nos estudos realizados²⁵ (estudo do estado da arte) em termos de ferramentas de apoio ao ensino de Lógica. São eles:

- (1) Suporte ao conceito de OA: Todas as ferramentas trabalham com listas de exercícios e exemplos de exercícios. Nesse sentido este item trata do uso da abordagem de OA (descritos na seção 3.2) de forma mais direcionada a elaboração de materiais educacionais, principalmente no que diz respeito às possibilidades de criações de listas por parte do professor e do reuso destas, podendo ser reconfiguradas ou reorganizadas conforme o contexto.
- (2) Uso de demonstração como modelo de aluno: faz uso de uma demonstração formal que oferece uma estrutura simbólica apropriada para acompanhar o processo de ensino-aprendizagem de dedução na Lógica.

²⁵ Das ferramentas encontradas e utilizadas na Tabela 12 para comparação entre os sistemas tutores de Lógica pesquisados e o sistema Heráclito, os mais recentes sistemas datavam do ano de 2005, o que nos leva a crer que este assunto parou de ser estudado e abordado nas pesquisas do meio acadêmico (pelo menos temporariamente).

- (3) Geração automática da demonstração: é importante que o sistema possa gerar demonstrações de forma automática, para evitar sobrecarregar o professor. Essas demonstrações devem ser similares a demonstrações que possam ser produzidas por professores ou alunos, no mínimo deve utilizar exatamente as mesmas regras de dedução disponíveis aos alunos.
- (4) Completa provas parciais: o sistema deve ser capaz de completar uma prova parcial exatamente no ponto em que o aluno parou, mesmo quando este tenha uma ou mais regras hipotéticas em aberto.
- (5) Possibilidade de gerar o próximo passo: o sistema deve se capaz de gerar automaticamente o próximo passo no caminho da finalização da prova.
- (6) Permite que o aluno reflita sobre o erro: é importante que seja permitido ao aluno refletir sobre um passo possivelmente incorreto, sem oferecer a explicação e solução de forma imediata. Tal solução e explicação serão eventualmente fornecidas, mas somente após um processo dialético (com uso de agentes pedagógicos e de estratégias de aprendizagem utilizadas e selecionadas conforme o contexto trazendo um feedback e informando o aluno sobre a sua situação).
- (7) Suporte para AVA: idealmente o sistema deve poder se integrar a algum tipo de AVA.

Além de tutores de Lógica, no ensino dessa disciplina também podem ser empregadas ferramentas como provadores automáticos, verificadores de provas e editores de prova. As ferramentas encontradas na literatura são descritas abaixo:

- ✓ Assistente de provas Coq: É um sistema formal de geração de provas. Coq implementa uma especificação de programa e linguagem de alto nível matemático chamado Gallina que é baseado em uma linguagem expressiva formal chamada de cálculo de construções indutivas, onde uma lógica de ordem superior e uma linguagem de programação funcional são combinadas. Como um sistema de desenvolvimento de prova, Coq fornece métodos interativos, prova de decisão e algoritmos de semi-decisão, e uma tática de idiomas para deixar o usuário definir seus métodos próprios de prova. Como uma plataforma para a formalização da matemática ou o desenvolvimento de programas, Coq fornece suporte para notações de alto nível, conteúdos implícitos e vários outros tipos úteis de macros [COQ, 2012].
- ✓ Assistente de provas HOL: HOL4 é a mais recente versão do assistente de prova interativo para lógica de ordem superior (HOL). É um ambiente de programação em que teoremas podem ser provados. HOL4 é particularmente apropriado como uma plataforma para a implementação de combinações de execução dedução, e verificação de propriedade. É um projeto de código aberto com uma licença BSD que permite a sua livre utilização em produtos comerciais [HOL4, 2012].
- ✓ Proveedor de teorema Otter: Otter (Organized Techniques for Theorem-proving and Effective Research) é um proveedor automático que trabalha

com teoremas para lógica de primeira ordem. O provador utiliza-se de mecanismos para provar teoremas aplicando regras de resolução binária, hiperresolução, resolução-UR e para modulação binária. Algumas de suas outras habilidades e características são as conversões de fórmulas de primeira ordem. Otter é codificado em ANSI C, é gratuito, e é portátil para muitos tipos diferentes de computadores [OTTER, 2012].

- ✓ Provador de teorema Prover9/Mace4: Prover9 é um provador de teoremas automatizado para a lógica de primeira ordem e equacional, e Mace4 procura por modelos finitos e contra. Prover9 é o sucessor do provador Otter [PROVER9, 2012].
- ✓ Provador de teorema EProver: EProver é um provador de teoremas para a lógica de primeira ordem com igualdade plena. O provador aceita uma especificação de problema, geralmente constituídos por uma série de condições ou fórmulas, e uma conjectura, quer em forma parcial ou completa. Seu desenvolvimento começou como parte do projeto E-SETHEO na TUM. O primeiro lançamento público foi em 1998, e o sistema tem sido continuamente melhorado desde então [The E Theorem Prover, 2012].
- ✓ Provador de teorema SPASS: É um provador de teoremas automatizado para Lógica de Primeira Ordem com Igualdade. SPASS oferece algumas funcionalidades, tais como: análise formal de software, sistemas, protocolos, abordagens formais, procedimentos de decisão, prova de teoremas modal lógica. Possui uma interface Web funcional [SPASS, 2012].
- ✓ JAPE: O JAPE é um provador de teoremas com uma interface gráfica que permite a um usuário editar facilmente uma prova em uma das lógicas suportadas por ele, entre elas a dedução natural para as lógicas proposicional e predicativa. Possui versões Unix (Sparc, Solaris e Linux) e Machintosh e é possível mostrar a estrutura da prova como uma árvore [JAPE, 2012].
- ✓ Editor de provas Pandora: Pandora (assistente de prova para Dedução Natural) é uma ferramenta de apoio à aprendizagem projetado para orientar a construção de provas de dedução natural. Desenvolvido em Java [PANDORA, 2012].

Como mostrado anteriormente, análises preliminares indicam a existência de uma grande quantidade destas ferramentas, incluindo sistemas como: Coq e HOL, que fornecem linguagens de especificação baseados em Lógicas avançadas (Lógicas de Alta Ordem), capazes de oferecer um sofisticado apoio para a construção de provas formais nestas Lógicas; Otter, Prover9/Mace4, EProver e SPASS são provadores automáticos de Lógica de Primeira Ordem; além dos editores/verificadores de provas JAPE e Pandora.

Apesar da existência de vários sistemas, todos apresentaram deficiências importantes que inviabilizam sua utilização direta no projeto do Heráclito. O Coq e o HOL devem ser utilizados através de linguagens de programação bastante complexa que torna inviável sua aplicação em turmas dos semestres iniciais de um curso de

Computação. Os provadores automáticos citados (além de vários outros) utilizam sistemas avançados de prova (tipicamente baseados em Hiperresolução) que são completamente inapropriados para atividades de ensino, além de serem praticamente intraduzíveis (pelo menos de uma forma que seja didaticamente aceitável) para demonstrações em Dedução Natural. Os editores JAPE e Pandora são particularmente complexos de serem customizados e reusados em formas não previstas por seus autores, ao menos se se considerar que essa customização deva ser feita por um professor de Lógica e não por um especialista em programação.

Esses editores também não têm capacidades de prova automática que são necessárias para um sistema automático de apoio pedagógico, que é o objetivo do Heráclito.

5 O SISTEMA HERÁCLITO

Este capítulo apresenta uma visão geral do Sistema Heráclito e do seu editor de provas, trazendo informações sobre usabilidade e funcionalidades que o editor possui e a arquitetura desenvolvida. Para uma melhor compreensão do processo de execução do sistema, um cenário de aplicação com a principal tela de edição do editor será apresentado.

5.1 Diagrama de Caso de Uso do Sistema Heráclito

O Sistema Heráclito possui atualmente um ator que é o aluno. Este interage com o sistema por meio do editor de provas basicamente para a resolução de exercícios de dedução natural de lógica proposicional (DNLP). Ao resolver um exercício o aluno pode editá-lo, manipulando-o através do uso das regras, recebendo auxílio e ainda pode ou não solicitar ajuda ao sistema tutor da ferramenta. O diagrama de caso de uso pode ser visto na Figura 11 abaixo:

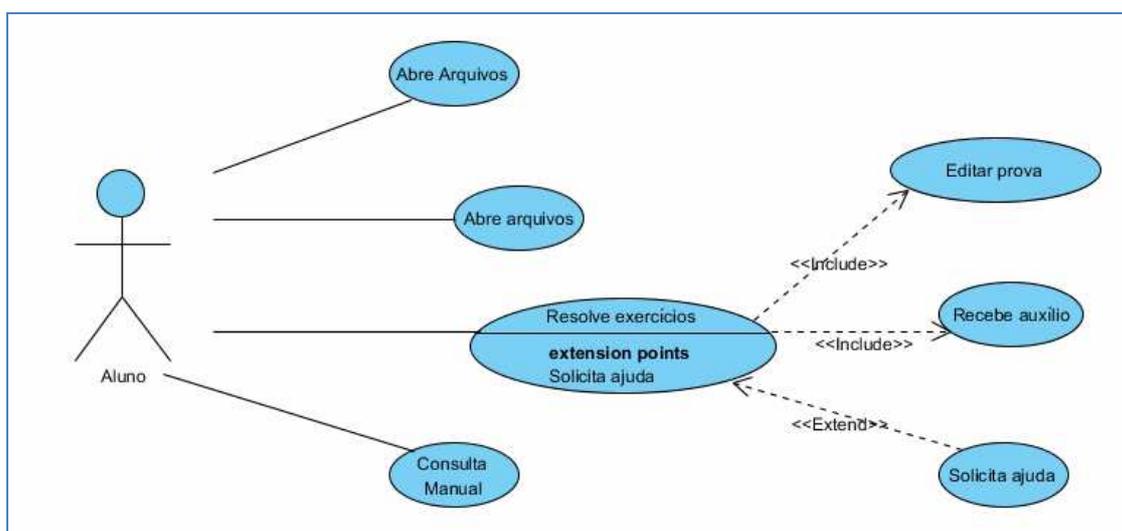


Figura 11– Caso de uso do Sistema Heráclito

5.2 Arquitetura do Sistema Heráclito

A arquitetura de *software* definida para o Sistema Heráclito é compatível com a infraestrutura de agentes MILOS (visto na seção 3.7). Nesse contexto o Heráclito se enquadra como um protótipo no Sistema de Apoio Pedagógico para o domínio de ensino de Lógica.

A arquitetura do Sistema Heráclito é apresentada na Figura 12. Nesta figura não

está indicado o armazenamento do OA em um AVA em particular, ou a transferência do OA em um AVA ao computador do aluno. Tais processos são de responsabilidade das funcionalidades já instaladas no AVA (o *Moodle*, por exemplo, já suporta integralmente este tipo de operação e foi esta a alternativa empregada nos testes do Sistema Heráclito).

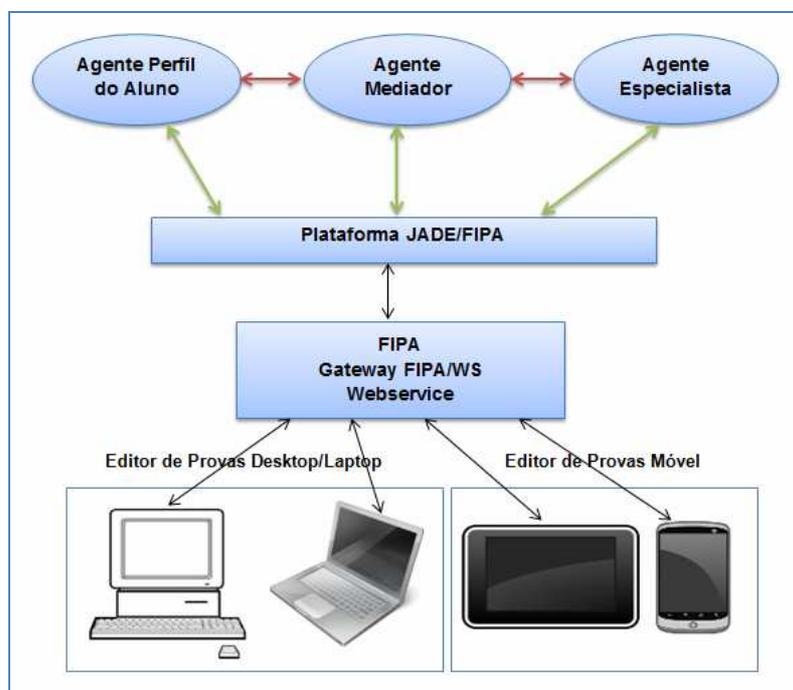


Figura 12 – Arquitetura do Sistema Heráclito.

Fonte: Elaborado pela autora e demais integrantes do projeto ObaaMilos.

Os agentes definidos são responsáveis por um ou mais cenários identificados no diagrama de especificação do sistema (descritos no capítulo 6). A organização destes agentes foi estruturada da seguinte forma:

- Agente Perfil do Aluno: Este agente representa o modelo de aluno²⁶, que no Sistema Heráclito é baseado no processo de resolução de exercícios de dedução. Conforme [DILLENBOURG e SELF, 1992] um modelo de aluno pode ser analisado em três níveis de abstração: operacional, comportamental e conceitual. Aplicando-se essa análise ao ensino de dedução na Lógica obtêm-se modelos com a seguinte organização:
 - (1) Nível Operacional: neste nível o modelo é responsável por verificar se o aluno consegue ou não aplicar as regras de dedução de forma correta.
 - (2) Nível Comportamental: o modelo avalia se o aluno consegue ser proficiente em seu processo de raciocínio para resolver um dado problema de dedução na Lógica.

²⁶ O perfil deste agente representa um modelo de aluno que no sistema Heráclito foi definido com base no processo de resolução de exercícios de dedução, ou seja, ele representa o comportamento do aluno dentro do processo de resolução dos exercícios, juntamente com as possíveis ações que o aluno poderá realizar, visto que estas ações podem ser previamente previstas.

- (3) Nível Conceitual: o modelo indica a assimilação dos conceitos envolvidos no processo de dedução na Lógica. É atingido quando o aluno compreende as implicações da dedução para outras áreas da Lógica e para áreas relacionadas como a Matemática e a Ciência da Computação.

O agente Perfil do Aluno do Heráclito trabalha nos níveis operacional e comportamental, verificando se o aluno sabe aplicar corretamente as regras de dedução e se ele é capaz de avançar no processo de resolução de um exercício de dedução até o final. Nesta última tarefa o agente Perfil do Aluno conta com a ajuda do agente Especialista do Heráclito.

O agente Perfil do Aluno ainda possui outras tarefas adicionais, tais como: guardar o histórico por aluno, gerando um registro de ações (*Log*) para que o professor possa consultar posteriormente, e para que o agente Mediador também possa consultar, para fins de decisão em seu processo pedagógico de mediação. Em outras palavras, este agente deve manter as informações relacionadas ao aluno em questão.

- Agente Mediador: Este agente representa o papel do professor em sua função didático-pedagógica, fazendo uso de diversas estratégias de ensino-aprendizagem com o intuito de auxiliar o aluno em seu processo de raciocínio na resolução de um exercício.
- Agente Especialista: Este agente também representa o papel do professor, mas em sua função de especialista no domínio de ensino de Lógica. Este agente segue o processo de resolução do aluno acompanhando a prova que o aluno está resolvendo. Este agente tem a capacidade de resolver o problema em questão do ponto em que o aluno está trabalhando até seu final. Com essa capacidade o agente Especialista pode detectar se a aplicação dos passos de dedução natural segue corretamente ou não. Este agente possui uma interface em Java e trabalha em conjunto com o provador de teoremas. Este provador foi desenvolvido em Prolog e é responsável por realizar as provas dos argumentos inseridas no sistema Heráclito.

Todas as mensagens trocadas entre esses três agentes passam obrigatoriamente pelo agente mediador, que é quem media esta comunicação entre o agente perfil do aluno e agente especialista. Mais informações sobre o Agente Mediador podem ser obtidas no capítulo 6.

O padrão OBAA [VICARI e GLUZ, 2010] prevê a possibilidade que os OAs possam utilizar serviços externos. Esses serviços tanto podem ser instalados localmente no computador onde o OA está rodando, quanto estarem disponíveis remotamente através de um protocolo de acesso. Os metadados OBAA permitem que sejam definidas todas as características necessárias para que o OA utilize um dado serviço, especificando desde os endereços dos pontos de acesso ao serviço, passando pelas mensagens e protocolos de interação utilizado pelo serviço, chegando até a linguagem e ontologia empregada nas mensagens.

No caso dos OAs Heráclito, o acesso aos agentes de apoio pedagógico será disponibilizado por meio de uma interface *Web Service*. Os OAs Heráclito implementam um cliente de serviços, que acessa o provedor de serviços implementado

pelo componente Facilidade de Acesso aos Agentes MILOS. Este provedor, por sua vez, implementa mecanismos de comunicação que convertem as requisições de serviços dos OAs Heráclito, em mensagens FIPA que podem então ser enviadas aos agentes MILOS por meio da plataforma de comunicação JADE.

A Figura 13 mostra os detalhes internos da integração do Heráclito com o Sistema de Apoio Pedagógico da MILOS.

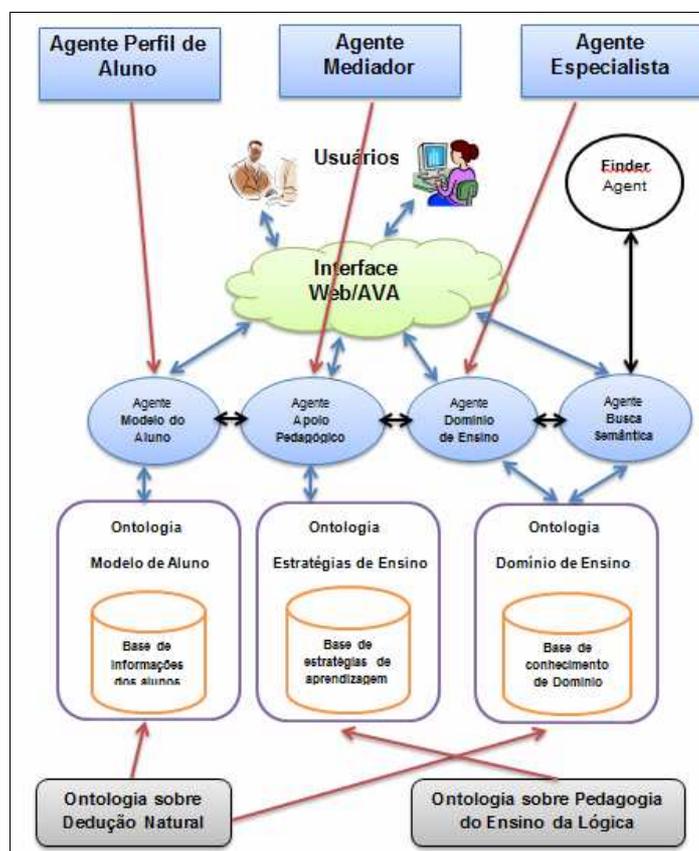


Figura 13 – Integração do Heráclito na MILOS.
Fonte: VICARI e GLUZ [2011].

A arquitetura é composta por duas ontologias: a ontologia de conteúdo de Lógica (domínio de ensino) e a ontologia de estratégias pedagógicas sobre o ensino de Lógica, descritas no capítulo 5.4.

5.3 Editor de provas do Sistema Heráclito – Versão Desktop

O Heráclito se apresenta aos seus usuários como um objeto de aprendizagem que têm por objetivo auxiliar os alunos da disciplina de Lógica em como elaborar provas de argumentos formais por meio das regras da Dedução Natural.

Um OA Heráclito é formado por um arquivo em formato Jar que pode ser obtido diretamente do AVA sendo utilizado na disciplina ou através do site do Portal

MILOS²⁷. Este arquivo Jar contém além do aplicativo Editor de Provas e de seu manual de usuário, listas de exercícios preparadas pelo professor, exemplos de demonstrações corretas e materiais (tanto em forma gráfica quanto textual) explicativos sobre a lista e sobre os conteúdos de Lógica tratados pelo objeto.

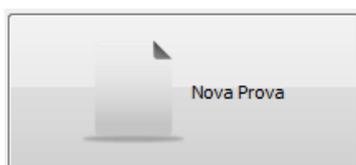
O Heráclito tem em sua tela inicial de boas vindas, 4 botões com opções distintas e um espaço para o usuário realizar o *Login* de usuário como pode ser visto na Figura 14 abaixo:



Figura 14 – Tela inicial do Sistema Heráclito.

5.3.1 Interface com o Usuário

Um OA Heráclito é formado por uma *applet* Java que pode ser obtida diretamente do AVA sendo utilizado na disciplina ou através do site do Portal MILOS²⁸. A tela inicial deste OA é composta por 4 botões. Abaixo uma breve descrição de cada botão e suas funcionalidades.



Botão Nova Prova: Esta opção permite ao usuário iniciar uma nova prova a partir de um conjunto de opções de exercícios de demonstração selecionados pelo professor. É onde as edições das provas de lógica são realizadas.

²⁷ O Portal MILOS está disponível em: <http://obaa.unisinos.br/>.

²⁸ O Portal MILOS está disponível em: <http://obaa.unisinos.br/>.



Botão Exemplos de Provas: Esta opção dá acesso a argumentos já provados, elaborados pelo professor e é composta por uma série de exercícios que foram divididos em 3 níveis de complexidade: inicial, intermediária e avançada.

Nível 1 (básico) Exercícios:

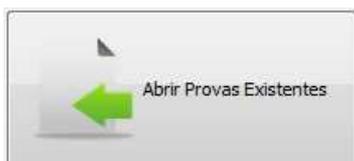
Este conjunto de exercícios é projetado para que o aluno possa experimentar as regras básicas.

Nível 2 (Intermediário) Exercícios:

Este conjunto de exercícios é projetado para que o aluno experimente uma combinação de regras.

Nível 3 (Avançado) Exercícios:

Este conjunto de exercícios consiste em alguns argumentos interessantes e mais difíceis (de maior complexidade).



Botão Abrir Provas Existentes: Esta opção dá acesso às provas feitas pelo aluno que foram testadas e salvas no computador (podendo ser abertas ou retomadas a qualquer momento).



Botão Manual do Heráclito: manual de instruções e de funcionamento da ferramenta.

Uma vez instalado no computador do aluno, o Heráclito se comportará como um editor de provas que permite ao aluno resolver os problemas de demonstração. Quando o editor opera em modo local não é oferecido serviço de tutoria. Este tipo de serviço somente estará disponível depois do *login* do usuário. Depois do *login* realizado o suporte pedagógico entrará em operação através do acesso aos agentes de apoio pedagógico para o ensino de Lógica, disponibilizados pelo servidor MILOS.

5.3.2 O Login do usuário

Para usar os serviços de tutoria do Heráclito é necessário realizar *login*. Em sua atual fase, o usuário é cadastrado previamente e recebe um usuário e uma senha do sistema. Ao realizar o *login* com este usuário e senha, ele estará logado com o servidor, podendo usufruir dos serviços oferecidos pela ferramenta. A Figura 15 mostra onde realizar este procedimento:

**Para utilizar os serviços de tutoria dos agentes do heráclito,
é necessário estar conectado com a Internet e realizar o login abaixo:**

Digite o seu username:
 Digite a sua senha:

Figura 15 – Tela de Login do Heráclito

Nesse processo, é possível contar com agentes de *software*, em especial com o agente mediador, que tem um papel estratégico (no desenvolvimento pedagógico), mediando e participando ativamente no desenvolvimento do exercício.

Esta opção não é obrigatória, podendo o Heráclito ser utilizado apenas como editor de provas. Porém, a não realização do *login*, como descrito anteriormente, implica na não habilitação dos agentes (sem suporte).

5.3.3 A Tutoria no Heráclito

O serviço de tutoria do Heráclito visa ajudar o aluno na resolução dos exercícios, fazendo o papel do professor. À medida que um aluno for usando o Heráclito na resolução dos exercícios, os agentes pedagógicos do serviço de tutoria estarão monitorando suas ações, prontos para ajudá-lo em caso de necessidade. Se o aluno não conseguir avançar na realização de uma prova, este poderá pedir dicas ou sugestões ao tutor por meio do botão Ajuda. O botão Ajuda também poderá ser acionado a qualquer momento da prova.

O serviço de tutoria usa um conjunto de estratégias de aprendizagem, específicas para Lógica, ao ajudar o aluno a realizar uma prova de Dedução Natural. Essas Estratégias e Táticas Pedagógicas podem ser vistas no capítulo 6.

5.4 Uso do Editor de Provas – Versão Desktop

A principal interface da ferramenta Heráclito é a tela de edição de provas e argumentos (acionada por meio da interface inicial do sistema no botão nova prova).

Uma vez selecionado o botão Nova Prova, uma segunda tela com edição de provas será aberta (em guias). Esta tela é a principal interface do Heráclito com o usuário. E pode ser visualizada na Figura 16 abaixo:

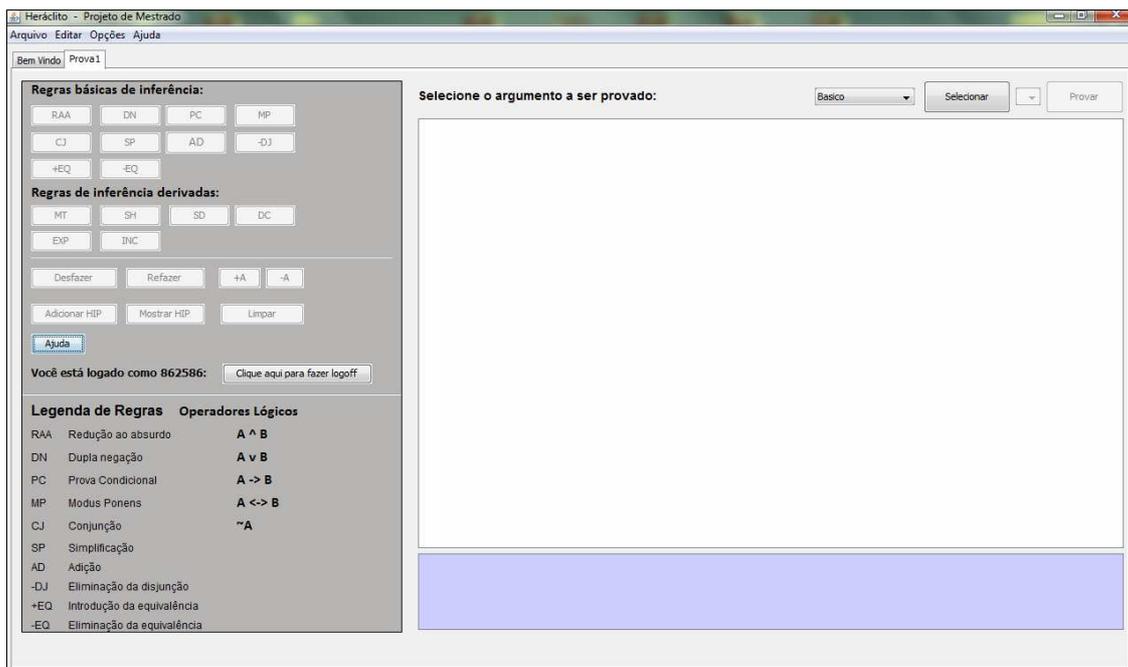


Figura 16 – Tela de edição do Sistema Heráclito

É nesta tela que os argumentos formais são provados.

Esta tela é composta por botões laterais os quais representam as regras de inferências básicas e derivadas.

A demonstração é elaborada passo-a-passo com base na aplicação das regras de inferência (básica e derivada) e para uma melhor compreensão deste processo, um exemplo para este cenário de aplicação será mostrado.

Para dar inicio a provação do argumento é necessário primeiramente:

1. Escolher o nível do argumento (Básico, Intermediário ou Avançado).
2. Após escolher o nível, uma lista com exercícios é mostrada. Seleciona-se o exercício e;
3. Em seguida o botão provar é acionado (clicando sobre ele).

A Figura 17 abaixo mostra esta etapa descrita anteriormente.

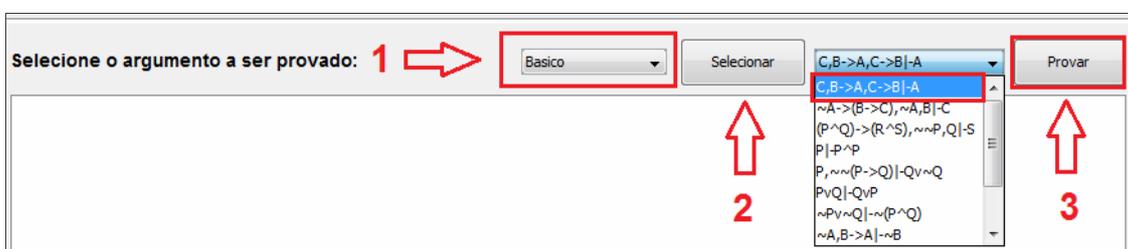


Figura 17 – Tela de edição do Sistema Heráclito selecionando um argumento

Ao selecionar um exemplo de argumento o processo de edição é iniciado.

O argumento vai para a tela de edição e com a ajuda dos botões laterais, a demonstração é elaborada passo-a-passo.

Antes de aplicar qualquer regra se faz necessário que o aluno primeiramente insira todas as hipóteses do argumento a ser provado. Em nosso exemplo (que segue a figura mostrada anteriormente) temos o seguinte argumento: $C, B \rightarrow A, C \rightarrow B \vdash A$.

As hipóteses deste argumento são as seguintes:

C ,

$B \rightarrow A$,

$C \rightarrow B$, e a Figura 18 abaixo ilustra a sequência destas inserções:

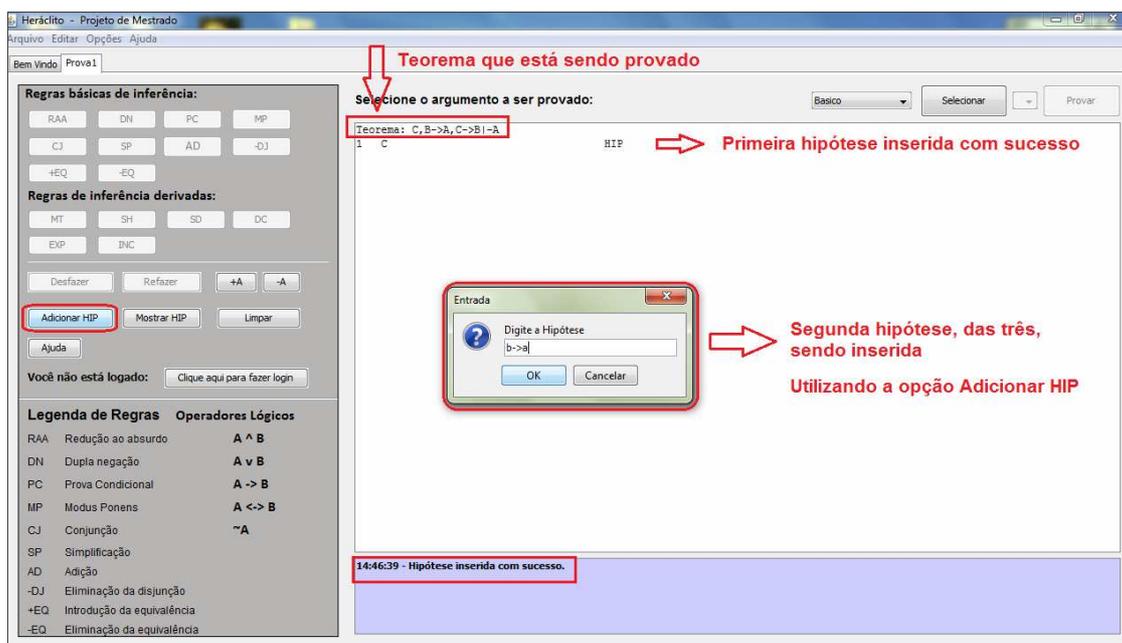


Figura 18 – Tela de edição do Sistema Heráclito adicionando hipóteses.

A situação de aprendizagem mostrada na figura acima está categorizada no nível Básico de dificuldade (sendo, portanto, na teoria, um argumento de fácil resolução).

O aluno inicia a demonstração do argumento formal: $C, B \rightarrow A, C \rightarrow B \vdash A$.

Que é representado no Heráclito pelo texto “ $C, B \rightarrow A, C \rightarrow B \vdash A$ ” e cujo raciocínio lógico elementar para resolvê-lo (após a inserção de todas as hipóteses) é aplicar sucessivamente a regra de Modus Ponens.

1. Primeiro “ C ” e “ $C \rightarrow B$ ”, resultando “ B ”.
2. Posteriormente, usando “ B ” e “ $B \rightarrow A$ ”, resultando “ A ” e;
3. Finalizando a demonstração. Caso o aluno execute essas regras terá sucesso e conseguirá provar o argumento, sem necessidade de intervenção do Heráclito.

A Figura 19 abaixo mostra esse processo:

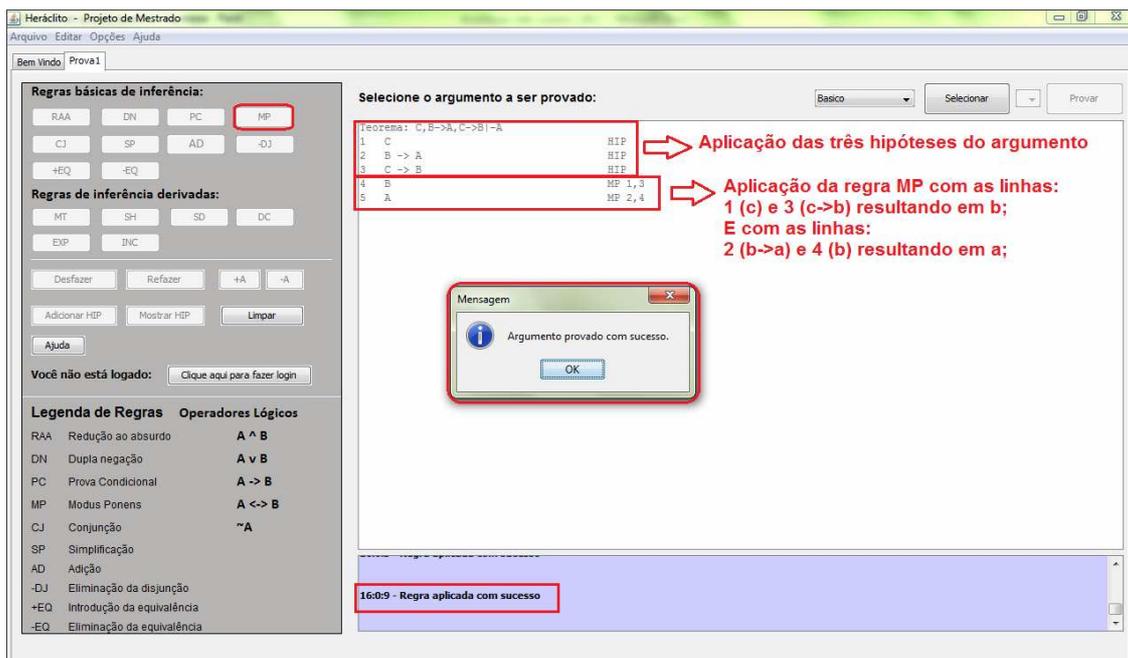


Figura 19 – Tela de edição do Sistema Heráclito aplicando a regra Modus Ponens.

5.5 Editor de Provas em Modo Local

A operação do Editor de Provas do Heráclito em modo com suporte do serviço de tutoria é tratada no capítulo 6, onde são apresentadas as várias estratégias e táticas pedagógicas que podem ser aplicadas pelo tutor (agente Mediador).

Entretanto, em modo local sem acesso ao serviço de tutoria, o Editor de Provas ainda é capaz de tratar de várias situações que ocorrem durante a resolução de um problema de dedução:

- Verificação da inserção de hipóteses: o Editor é responsável pela verificação se a inserção é correta ou não. Ao inserir uma hipótese, o Editor verifica se a hipótese é correta ou não. Uma hipótese é correta quando faz parte da lista de hipóteses do argumento.
- Verificação de aplicação de regras de dedução: o Editor simplesmente não permite que o aluno insira um novo passo através da aplicação incorreta de uma regra de dedução.
- Em modo local o botão Ajuda provê acesso ao manual do Editor e a exemplos de provas previamente resolvidas. O exemplo é escolhido aleatoriamente porque o Editor não possui suporte do serviço de tutoria para saber qual exemplo de uso de regra de dedução sugerir para o próximo passo.

5.6 Editor de Provas do Sistema Heráclito – Versão Móvel

Esta versão está em fase de desenvolvimento e de testes. Assim como a versão Desktop, a versão móvel também está sendo desenvolvido em Java. A versão para computador pessoal (desktop ou laptop) foi desenvolvida no ambiente NetBeans (ORACLE, 2012) e sua interface gráfica foi desenvolvida com a biblioteca Swing. A versão móvel para o sistema Android [GOOGLE, 2012] é desenvolvida no ambiente Eclipse (ECLIPSE, 2012), usando a interface XML e o kit de desenvolvimento Android (SDK). A versão móvel foi criada através de uma reengenharia do código da versão Desktop porque a biblioteca Swing não está disponível no Android, sendo necessária a reescrita do código visual [MOSSMAN; GOMES; GLUZ, 2012].

5.6.1 Interface Móvel

A interface do editor de provas (que pode ser vista na Figura 20) também é baseada em: botões, uma lista contendo o desenvolvimento da prova, um seletor de provas e a legenda esclarecendo a descrição das regras.

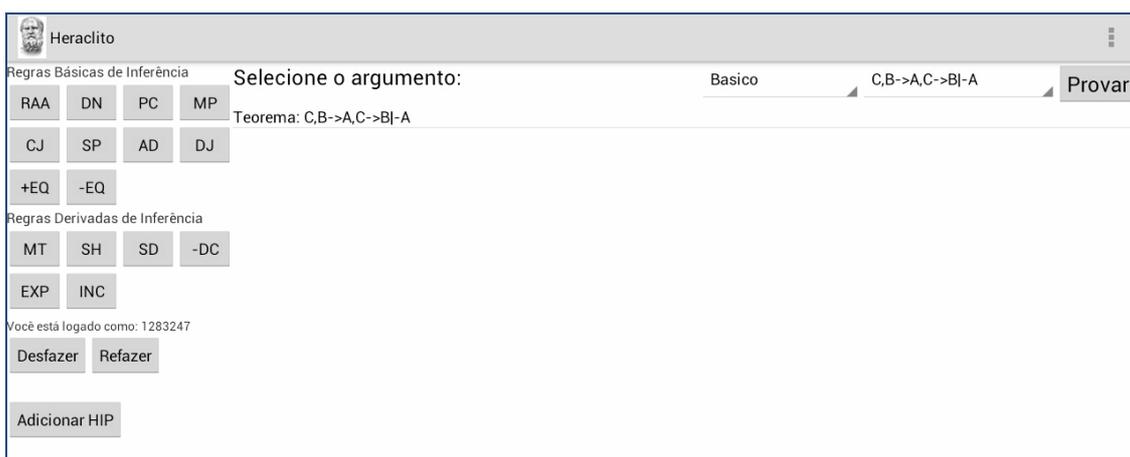


Figura 20 – Interface da versão móvel do sistema Heráclito

O quadro inferior de cor azul representa as mensagens tanto do sistema quanto dos agentes tutores existentes no servidor Heráclito. Se a mensagem for crítica, será exibida em um pop-up como pode ser vista na Figura 21.

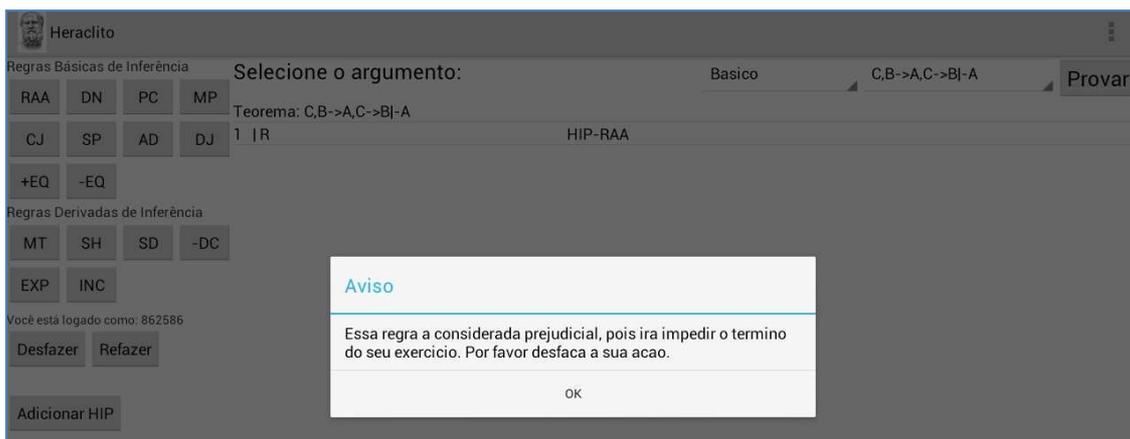


Figura 21 – Interface com mensagem crítica

Assim como o editor de provas da versão Desktop, a versão móvel do Sistema Heráclito suporta oito regras básicas de inferência não hipotética (Nolt e Rohatyn, 1991). São elas:

- MP (modus ponens);
- DN (dupla negação ou elim. da negação);
- CJ (introd. da conjunção);
- SP (separação ou elim. da conjunção);
- AD (adição ou introd. da disjunção);
- DJ (elim. da disjunção);
- +EQ (introd. da equivalência ou do bicondicional) e;
- -EQ (elim. da equivalência ou do bicondicional).

O editor suporta duas regras hipotéticas:

- RAA (redução ao absurdo) e PC (prova condicional), além das regras derivadas:
 - MT (modus tolens);
 - SH (silogismo hipotético);
 - SD (silogismo disjuntivo);
 - DC (dilema construtivo);
 - INC (inconsistência) e
 - EXP (exportação).

Todo o processo de interação é feito através de toques com os dedos diretamente nos botões do editor. O procedimento para uso é o mesmo realizado pelo mouse, porém aqui o aluno toca no botão correspondente a regras de dedução que pretende usar. Isso é indicado em verde (que traz o exemplo da regra MP - Modus Ponens). Posteriormente o aluno toca nas linhas de demonstração que serão usadas através das regras. As linhas que estão sendo utilizadas pela regra selecionada são indicadas na cor azul (Figura 22).

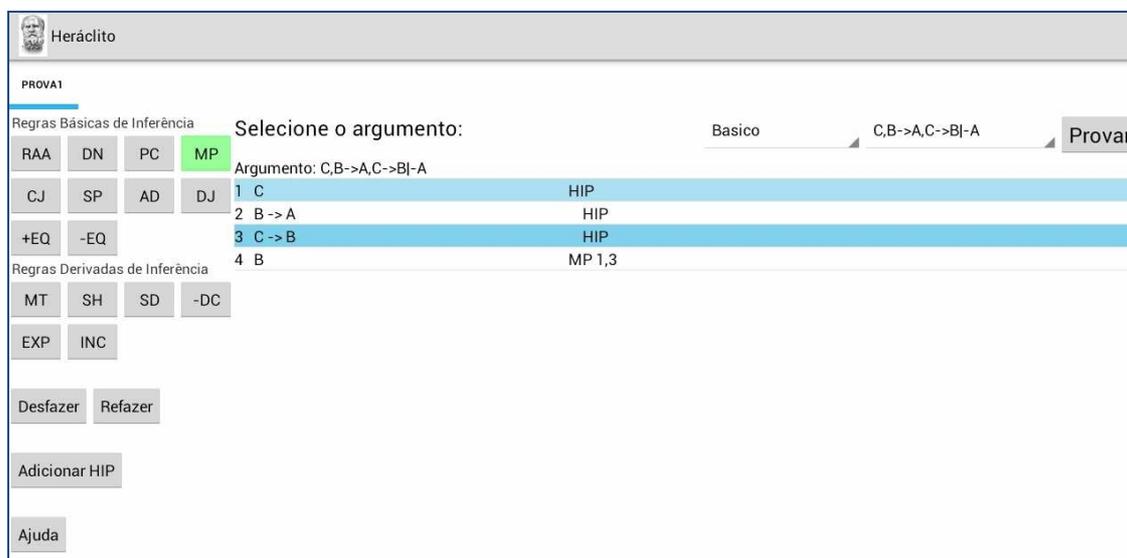


Figura 22 – Exemplo de uso de regra na versão móvel – Seleção da regra

Se as linhas selecionadas estiverem incorretas uma mensagem aparecerá. A Figura 23 demonstra esta aplicação.

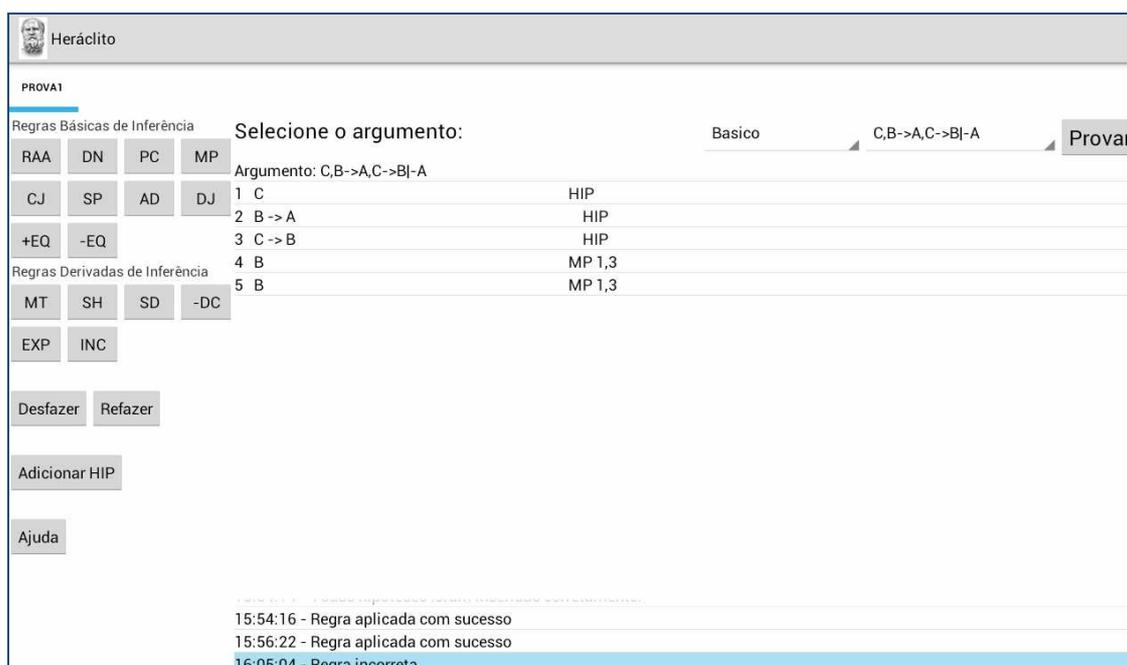


Figura 23– Exemplo de uso de regra na versão móvel – Aplicação da regra

Caso as linhas selecionadas estiverem corretas uma nova linha é adicionada ao fim da prova. No caso da Figura 24, se o aluno, após selecionar a linha 2 e selecionar a linha 4, por meio da regra MP, aparecerá à linha 5 contendo a fórmula A e a referência MP 2,4. Este processo iniciado nas Figuras 22 e 23 sobre o uso e aplicação das regras tem sua finalização na Figura 24.

The screenshot shows the Heraclito interface for a logic proof. The user is logged in as 'Heraclito'. The main area displays a theorem: $C, B \rightarrow A, C \rightarrow B \vdash A$. Below the theorem is a table of steps:

Step	Formula	Justification
1	C	HIP
2	$B \rightarrow A$	HIP
3	$C \rightarrow B$	HIP
4	B	MP 1,3
5	A	MP 2,4

At the bottom, a log shows the following actions:

- 00:37:17 - Hipótese inserida com sucesso.
- 00:37:29 - Hipótese inserida com sucesso.
- 00:37:56 - Hipótese inserida com sucesso.
- 00:37:56 - Todas as hipóteses foram inseridas corretamente.
- 00:38:04 - Regra aplicada com sucesso
- 00:38:05 - Parabans, estas no caminho certo.
- 00:39:56 - Regra aplicada com sucesso

A 'Prova completada!' button is visible at the bottom center of the interface.

Figura 24 – Finalização do processo do uso e aplicação das regras

5.6.2 Testes

Inicialmente foram realizados testes funcionais em laboratório tanto da aplicação cliente para computadores pessoais quanto para dispositivos móveis. Estes testes validaram a todas as operações da interface de usuário, sendo testada a aplicação de todas as regras de inferência da dedução natural. Assim, testes em laboratório comprovam a funcionalidade do Editor de Provas, auxiliando o aluno na correta resolução do argumento a ser provado.

Para verificar se a comunicação entre a aplicação cliente e o servidor Heráclito, primeiro foram realizados testes de envio e recebimento de mensagens entre dois agentes no mesmo computador. Logo a seguir foram realizados testes similares porém em diferentes computadores. Para verificar a interoperabilidade do Sistema Heráclito em distintos Sistemas Operacionais, esses testes de comunicação foram executados tanto no sistema Windows quanto no sistema Linux/Ubuntu. Atualmente se está procedendo aos testes de integração dos agentes tutores e da comunicação destes agentes com os editores de prova. O sistema como um todo será testado no segundo semestre de 2012. Espera-se que ele possa entrar em operação em aulas presenciais ou de EAD, incluindo o suporte à tablets, já no início de 2013.

Espera-se que a versão móvel seja mais difundida devido ao crescente uso de tablets/smartphones, criando assim, uma maneira simples de aprender disponível em sua mão, sem a necessidade de equipamentos maiores. Atualmente uma pequena parcela de estudantes possuem smartphones Android, mas, essa é uma parcela de crescimento muito alto, permitindo um maior acesso dos OAs Heráclito pelos alunos.

6 AGENTE MEDIADOR

Este capítulo apresenta detalhes das estratégias de mediação desenvolvidas para o agente Mediador do Sistema Heráclito, bem como a interação dos agentes neste sistema. Para uma melhor compreensão do processo de execução do agente Mediador, exemplos detalhados são apresentados demonstrando a forma de mediação, de comunicação. Também são apresentados detalhamentos da arquitetura vista no capítulo 5 e o desenvolvimento dos planos gerenciais de ensino didáticos.

6.1 Apoio Pedagógico do Agente Mediador

O agente Mediador é responsável por incorporar o papel do professor ou tutor dentro do Sistema Heráclito, provendo um apoio pedagógico ao aluno em seu processo de aprendizagem da dedução natural na Lógica Proposicional. Este apoio pode-se dar em várias situações deste processo.

Para todas essas situações o agente Mediador fica responsável pela efetiva intervenção de auxílio neste processo. Além destas situações, caso o aluno realize um erro ou venha repetir o mesmo erro, o agente Mediador seleciona estratégias pedagógicas apropriadas para tratar do erro.

A seguir são apresentadas as principais situações em que o agente Mediador irá interferir através da aplicação de alguma estratégia de mediação:

1. Quando ocorre uma inserção incorreta de hipóteses;
2. Quando ocorre uma aplicação incorreta de regra de dedução;
3. Quando ocorre uma aplicação correta (no nível operacional) de uma regra: neste caso, mesmo a aplicação sendo correta no nível operacional esta regra ainda pode ser problemática no nível comportamental, podendo ser categorizada em útil, redundante e prejudicial. A aplicação de uma regra útil supõe que a regra está correta (operacionalmente) e contribui para a resolução do exercício. A aplicação de uma regra redundante pode estar correta no nível operacional, mas não contribui com a resolução do exercício, e o aluno apenas aumenta o número de linhas e regras aplicadas

sem obter o resultado esperado. E a regra prejudicial quando aplicada acaba por levar o aluno a um caminho sem volta, ou seja, com a sua aplicação o exercício não poderá ser resolvido;

4. Quando o aluno fica muito tempo ocioso ou sem avançar na resolução do exercício;
5. Quando se atinge um percentual significativo da resolução da prova o aluno é informado e recebe um incentivo para prosseguir;
6. Quando faltam poucos passos (tipicamente 1 ou 2 passos) para a resolução da prova o aluno é informado e recebe um incentivo para finalizar;
7. Quando o aluno solicita ajuda. As respostas para as solicitações de ajuda podem ser de dois tipos: um exemplo de demonstração apropriado para a situação atual do aluno, ou, caso a situação do aluno seja mais crítica, uma dica, onde o sistema fornece o próximo passo a ser feito para continuar no processo de resolução.

6.2 Estratégias de Mediação

Esta seção apresenta as principais estratégias de mediação usadas para tratar das situações previstas na seção 6.1. Essas estratégias foram especificadas em UML 2.1 [UML, 2012], através de diagramas de atividade que podem ser usados para especificar os comportamentos dinâmicos de agentes [BAUER e ODELL, 2005].

As estratégias de mediação foram desenvolvidas com base no modelo de aluno, que neste trabalho representa o modelo de resolução de um exercício de dedução natural na Lógica Proposicional, ou seja, a metodologia utilizada em sala de aula foi transposta para o Sistema Heráclito.

Com isso, o primeiro processo a ser realizado em qualquer prova ocorre quando o aluno seleciona um argumento para ser provado. O agente PerfilAluno armazena esta informação ao mesmo tempo que a repassa ao agente Mediador por meio de uma mensagem. O Mediador apenas encaminha esta mensagem para o agente Especialista, de modo que este somente irá provar o argumento (passado em mensagem) e armazenar a informação para consultas futuras. A Figura 25 demonstra esse processo.

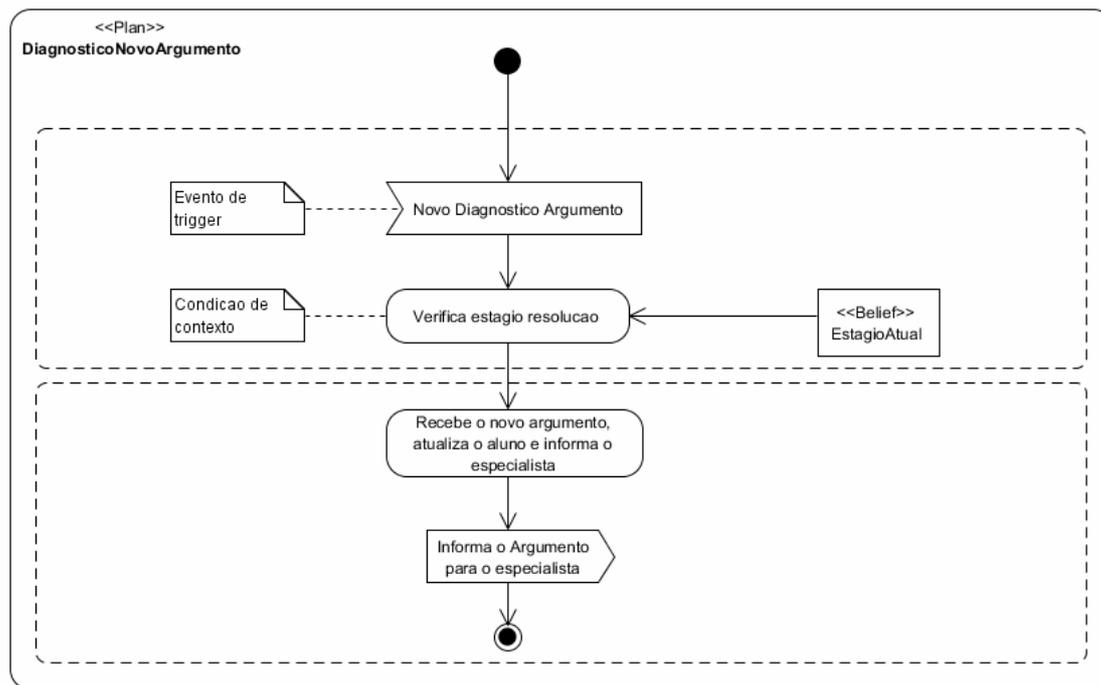


Figura 25 – Diagrama de atividades para o diagnóstico de um novo argumento.

Outra estratégia de ensino desenvolvida é a que está relacionada com a aplicação das regras básicas de inferência ou de inferência derivadas por parte do aluno.

Como visto no capítulo 5, a aplicação das regras pode ser de dois tipos: correta ou incorreta. O tratamento da aplicação de regras incorretas é feito pelo editor de provas, que controla o número de tentativas incorretas consecutivas ou não. Neste caso, além da mensagem informada ao aluno de que a regra aplicada está incorreta lhe é perguntado, como sugestão do editor, se ele necessita de ajuda. Essa estratégia pode ser vista na Figura 26.

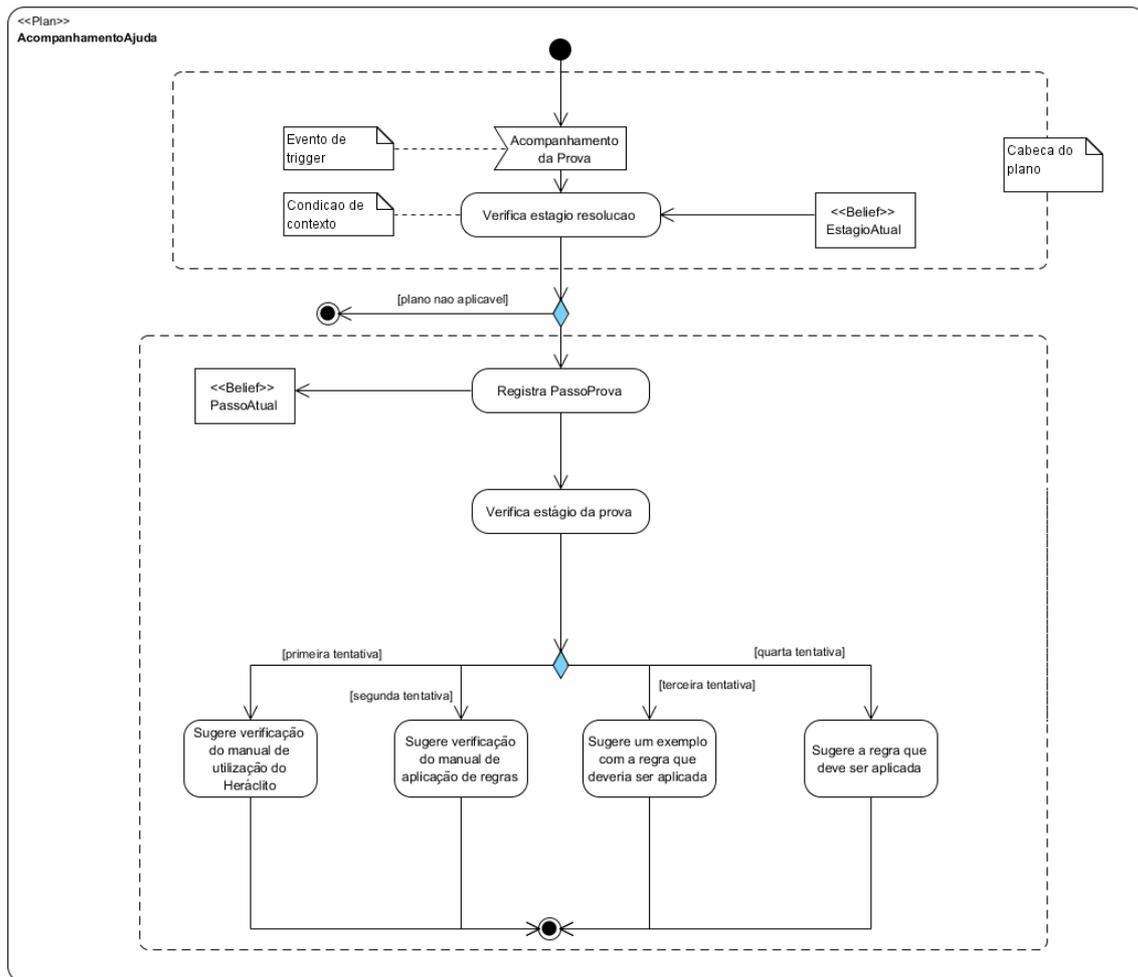


Figura 26 – Diagrama de atividades de acompanhamento de ajuda

A estratégia de mediação definida na Figura 26 busca fazer com que o aluno descubra qual regra deve aplicar, sem que ela seja informada diretamente a ele. A opção Ajuda oferece quatro frentes de suporte, onde embora o aluno tenha a liberdade de escolher qual delas deseja visualizar, o sistema faz um controle interno antes de dar a resposta direta ao aluno. A primeira delas (listando na ordem que aparecem no menu) é a opção “ver manual do Heráclito” que trata basicamente do manual de usabilidade da ferramenta. A segunda opção é o “exemplo de regras”, onde inicialmente sugere-se ao aluno que ele analise o conceito de aplicação das regras, onde o Mediador apresenta um exemplo de aplicação de regras. A terceira opção “exemplo de argumento” sugere-se ao aluno que ele visualize um exemplo de argumento já provado, onde o Mediador traz um exemplo onde já está descrita a aplicação da regra que ele deveria aplicar em sua resolução. E por fim, caso o aluno ainda necessite de ajuda e não tenha conseguido identificar qual regra deve ser aplicada, a opção “Dica” aciona o agente Mediador, que solicita ao agente Especialista que lhe envie a próxima regra que o aluno deve aplicar.

O tratamento das regras aplicadas corretamente é categorizado de três formas distintas: útil, redundante e prejudicial. Se a regra aplicada contribui para a prova do argumento, a regra é dita como útil. Nesse caso, uma mensagem de reforço positivo é enviada ao aluno lhe informando que a regra foi aplicada corretamente. Caso a regra aplicada não contribua e nem impeça a continuidade de provar o argumento, ela é classificada como redundante. Nesse caso, é feito um controle por número de tentativas

aplicadas nesta categoria e o agente Mediador executa um plano de ação com estratégias de acordo com cada situação apresentada (ver plano de ação de Acompanhamento de Ação Pedagógica de Prova – Figura 28).

Contudo, se a regra aplicada impede a continuidade da prova do argumento, ela é classificada como prejudicial. Neste caso, o aluno é informado que o passo dado impossibilita a conclusão do exercício e o sistema pede ao aluno que ele desfça sua última ação realizada. A Figura 27 mostra as estratégias definidas para lidar com a aplicação de regras corretas.

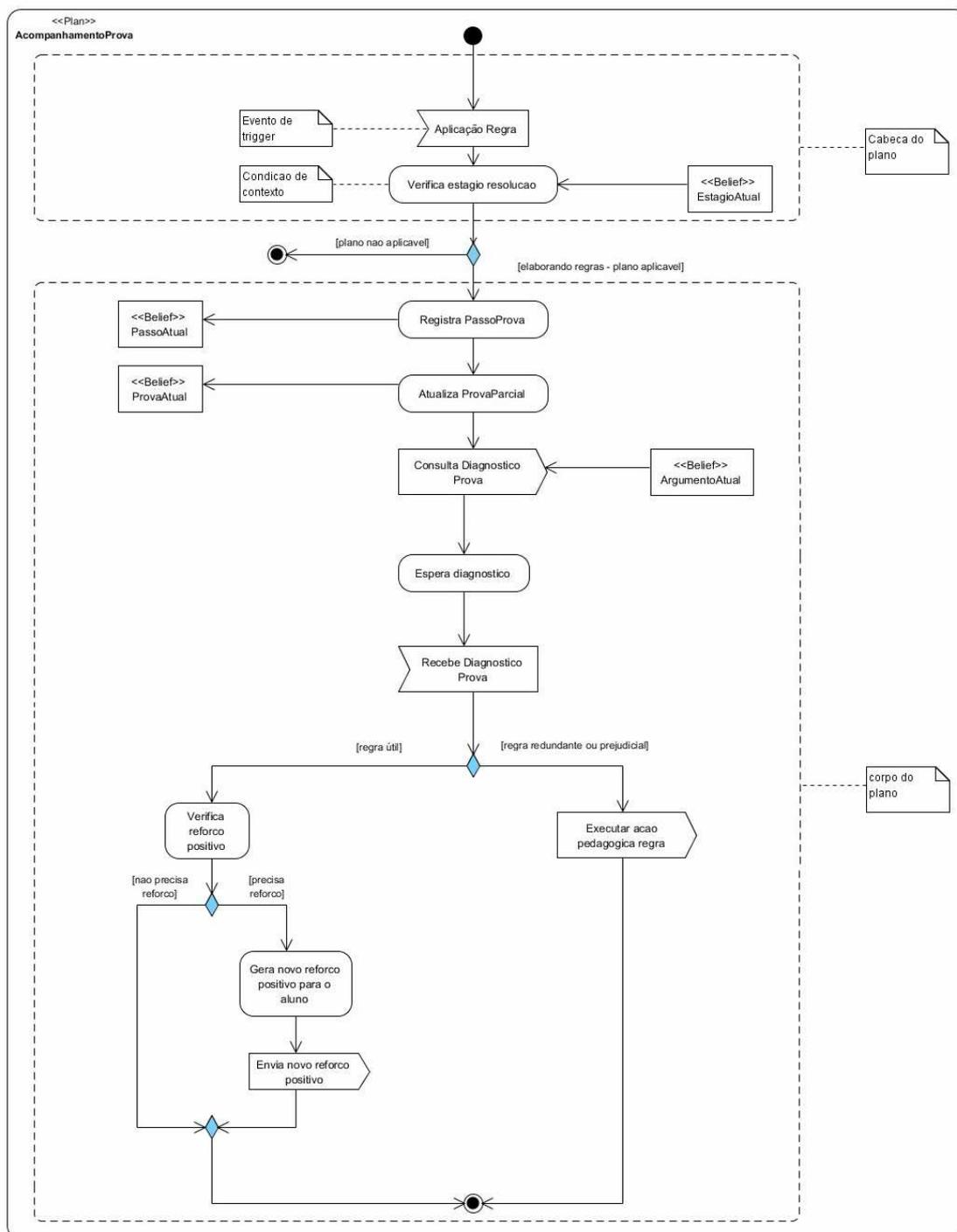


Figura 27 – Diagrama de atividades de Acompanhamento de Prova

Ainda com relação à aplicação de regras, cabe ressaltar que o status das regras corretas categorizadas anteriormente como útil, redundante e prejudicial é totalmente dependente do contexto em que está inserida. Sendo assim, faz-se necessário avaliar o que o aluno já realizou cada vez que uma nova regra é aplicada. Dessa forma, o agente PerfilAluno mantém o agente Mediador sempre informado quando uma nova regra é aplicada.

Para descobrir qual a categoria (útil, redundante ou prejudicial) da última aplicação de regra do aluno, o Mediador encaminha toda a prova parcial que o aluno desenvolveu até o momento, incluindo o último passo resultante da aplicação dessa regra, para o agente Especialista. Este agente utiliza um provador parcial de dedução natural que é capaz de completar a prova parcial do aluno, utilizando a sua prova parcial como ponto de partida e provando o argumento daquele ponto em diante. Caso esta prova não seja possível, então o último passo inviabilizou a prova do argumento, sendo considerado prejudicial. Caso contrário, são excluídos todos os passos não utilizados para a obtenção da conclusão do argumento. Se o último passo feito pelo aluno for eliminado neste processo, então esse passo é considerado redundante, senão é considerado útil. Após identificar a categoria do último passo, o Especialista passa essa informação para que o agente Mediador possa fazer uso das estratégias de aprendizagem apropriada para cada situação apresentada.

Quando a regra aplicada é útil, o aluno só é informado do status (útil), se ele aplicou, imediatamente antes, uma regra redundante ou prejudicial, de modo que essa mensagem sirva como reforço positivo servindo como incentivo para a conclusão da prova em andamento. Caso contrário, o aluno não é informado, evitando guiá-lo explicitamente para o resultado (neste caso o editor apenas informa o aluno de que a regra foi aplicada corretamente). Quando uma regra aplicada é considerada redundante ou prejudicial, a estratégia de mediação pode ser vista na Figura 28.

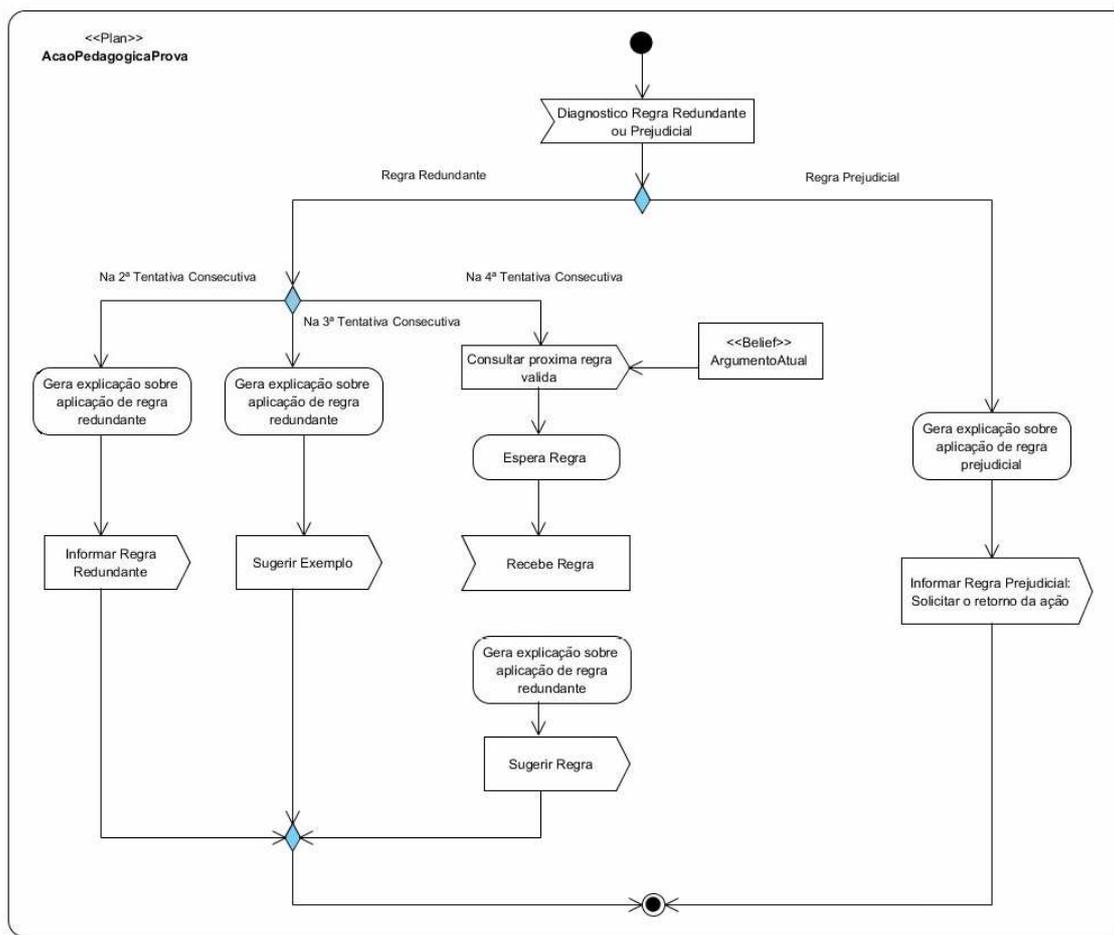


Figura 28 – Diagrama de atividades de Ação Pedagógica de uma Prova

Como pode ser visto na Figura 28, quando uma regra é identificada como prejudicial, informa-se o aluno da ocorrência, deixando claro que não será possível continuar a prova se ele não desfizer a última ação. Quando a regra é redundante, opta-se por deixar o aluno buscar a resolução, evitando interferir e guiá-lo diretamente para a resposta. Sendo assim, o aluno somente é informado quando ele aplicou duas regras redundantes em sequência ou não. Na terceira vez que o aluno aplicar uma regra redundante, sugere-se um exemplo da aplicação de regra por meio de uma prova já concluída. Esse exemplo sugerido contém a próxima regra que o aluno deveria aplicar para dar andamento na prova. Por fim, quando o aluno aplica a quarta regra redundante consecutiva ou não, ele é informado de qual regra ele deveria aplicar, uma vez que ele demonstra não entender claramente o que precisa ser feito para provar o argumento.

Outra estratégia desenvolvida para o Mediador é a estratégia que trata do caso de um aluno ocioso: quando o aluno estiver ocioso por um determinado período de tempo, envia-se uma mensagem para este aluno solicitando se ele necessita de ajuda, como pode ser visto na Figura 29.

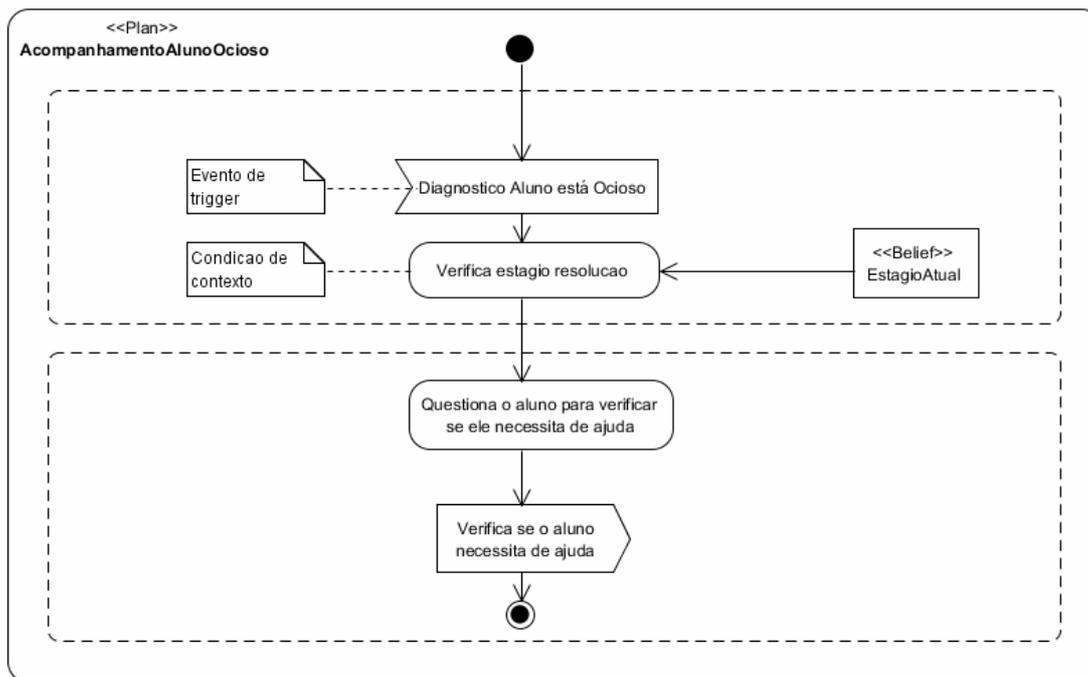


Figura 29 – Diagrama de atividades de Acompanhamento de aluno ocioso

Caso o aluno aceite a ajuda que o sistema lhe oferece este será encaminhado para o plano de ação visto na Figura 26 do diagrama de atividades de acompanhamento de ajuda.

Por fim, foi desenvolvida uma estratégia para informar o aluno sobre o status de desenvolvimento de sua prova. Quando o aluno atinge um número de passos igual ou maior do que 50% do total de passos da prova, o agente Mediador o informa (por meio de um reforço positivo) de que ele chegou à metade do desenvolvimento da prova. Essa estratégia também é aplicada quando o aluno atinge 80% da prova, quando faltam 2 passos e quando falta apenas 1 para concluir a prova. A Figura 30 mostra essa estratégia.

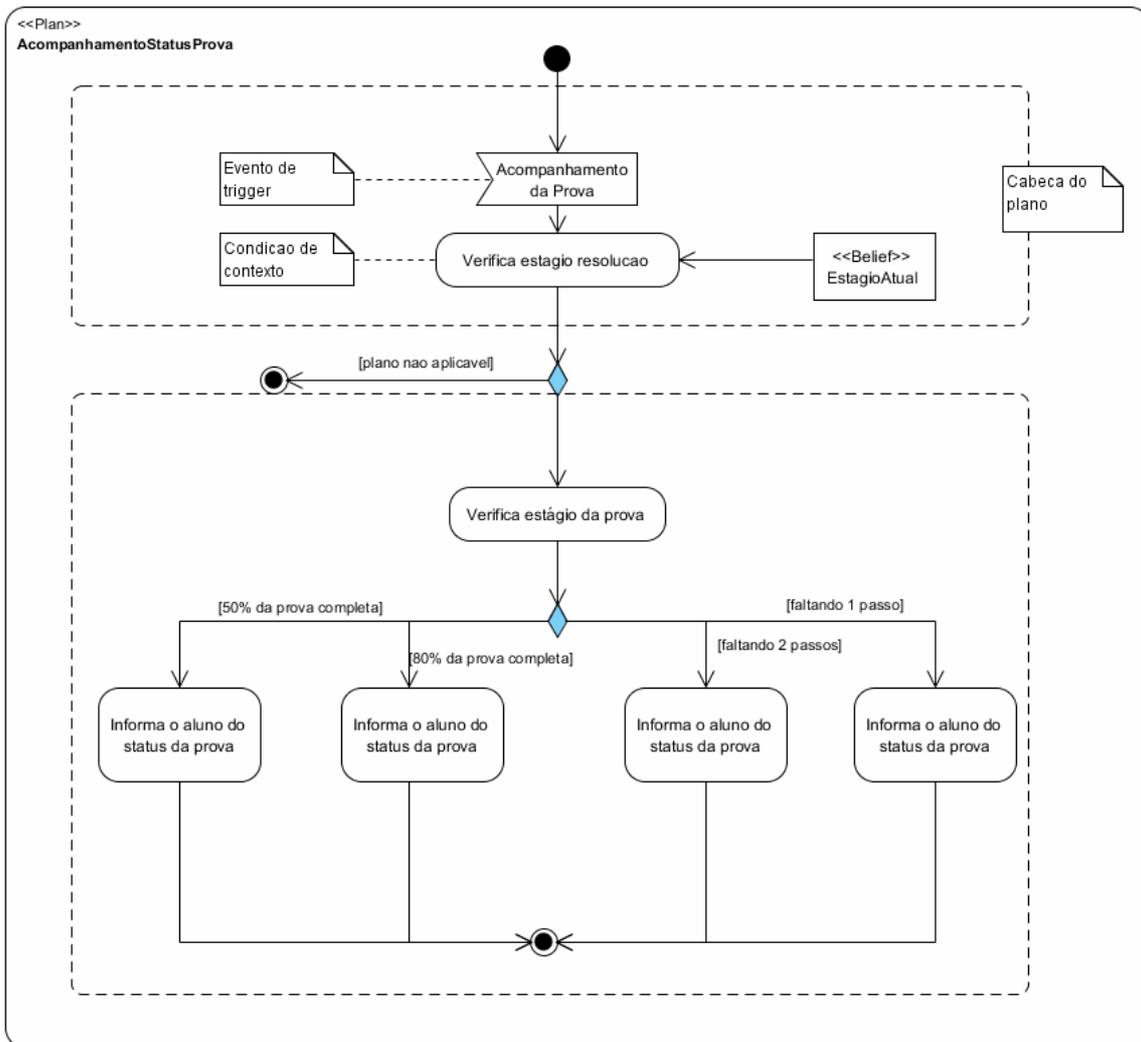


Figura 30 – Diagrama de atividades de Acompanhamento de Status da Prova

6.3 Comunicação entre os Agentes

Esta seção apresenta as interações que ocorrem entre o agente Mediador e os demais agentes do Sistema Heráclito, em função das estratégias de mediação definidas na seção 6.2. Essas interações também foram especificadas em UML 2.1 [UML, 2012], através de diagramas de sequência [BAUER e ODELL, 2005].

Todas as estratégias foram desenvolvidas com base no perfil do aluno e estão centradas no agente Mediador, o qual toma uma decisão baseado no contexto em que o aluno está inserido, fazendo uso das estratégias de aprendizagem e assim gerenciando as trocas de mensagens entre os agentes PerfilAluno e Especialista.

A Tabela 13 mostra todas as mensagens trocadas entre os agentes PerfilAluno, Mediador e Especialista, onde *Sender* é o agente que está enviando a mensagem, *Mensagem* é a denominação da mensagem, *Parâmetros* são os detalhes contidos nas mensagens e *Receiver* é o agente que recebe a mensagem. Pode-se verificar que nenhuma mensagem enviada pelo PerfilAluno vai diretamente ao Especialista e nem do

Especialista para o PerfilAluno (como mostra a Figura 12 sobre a arquitetura do sistema Heráclito no capítulo 5).

Tabela 13 – Mensagens trocadas entre os agentes.

Sender	Mensagem	Parâmetros	Receiver
PerfilAluno	novo_argumento	Argumento, Aluno	Mediador
PerfilAluno	diag_prova	Argumento, ProvaParcial, Aluno	Mediador
PerfilAluno	aluno_ocioso	Argumento, ProvaParcial, Aluno	Mediador
PerfilAluno	solic_ajuda	Argumento, ProvaParcial, Aluno	Mediador
Mediador	novo_argumento	Argumento, Aluno	Especialista
Mediador	diag_prova	Argumento, ProvaParcial, Aluno	Especialista
Mediador	obter_exemplo_regra	Argumento, ProvaParcial, Status, Aluno	Especialista
Mediador	obter_regra	Argumento, ProvaParcial, Aluno	Especialista
Mediador	validar_hipoteses	Argumento, ProvaParcial, Aluno	Especialista
Mediador	informar	Argumento, ProvaParcial, Aluno	Especialista
Mediador	diag_prova	Argumento, ProvaParcial, Status, Aluno	PerfilAluno
Mediador	informar	Mensagem, Aluno	PerfilAluno
Mediador	informar_regra	Regra, Aluno	PerfilAluno
Mediador	sugerir_exemplo	ExemploRegra, Aluno	PerfilAluno
Mediador	verifica_ajuda	Mensagem, Aluno	PerfilAluno
Mediador	solic_ajuda	Mensagem, Aluno	PerfilAluno
Mediador	novo_argumento	Argumento, Prova, Aluno	PerfilAluno
Especialista	diag_prova	Argumento, Status, ProvaParcial, Aluno	Mediador
Especialista	obter_exemplo_regra	Argumento, ProvaParcial, Status, RegraExemplo, Aluno	Mediador
Especialista	validar_hipoteses	Status, Aluno	Mediador
Especialista	informar	Mensagem, Aluno	Mediador
Especialista	obter_regra	Regra, Aluno	Mediador
Especialista	novo_argumento	Argumento, Prova, Aluno	Mediador

A seguir, são apresentados os diagramas de sequência, mostrando as trocas de mensagens entre os agentes para cada estratégia citada na seção 6.1. A Figura 31 mostra as mensagens trocadas quando o aluno inicia um novo argumento.

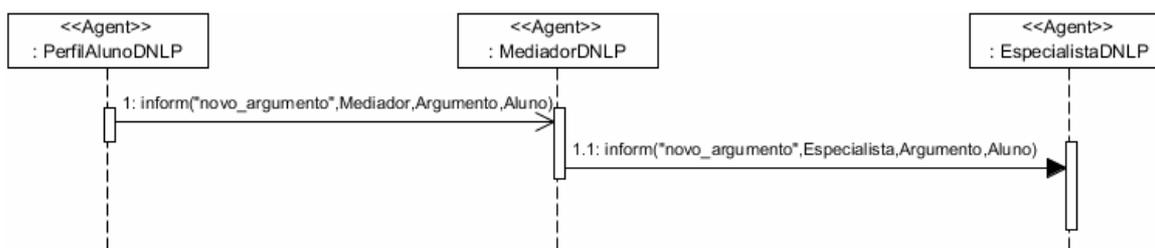


Figura 31 – Troca de mensagens entre os agentes quando o aluno inicia um novo argumento

Pode-se verificar, na Figura 31, que o agente Especialista informa a prova do

argumento para o agente Mediador que, por sua vez, informa o agente PerfilAluno. Isso ocorre, pois quando o aluno conclui a prova de seu argumento, o editor de provas pode mostrar a prova realizada pelo sistema Tutor, de modo que ele possa comparar e analisar as diferenças.

A Figura 32 mostra as mensagens trocadas entre os agentes quando o aluno aplica uma regra. Pode-se verificar que em alguns casos o Mediador decide solicitar ao agente Especialista qual deve ser a próxima regra que o aluno deve aplicar, de modo a sugerir um exemplo ou então informar diretamente a regra.

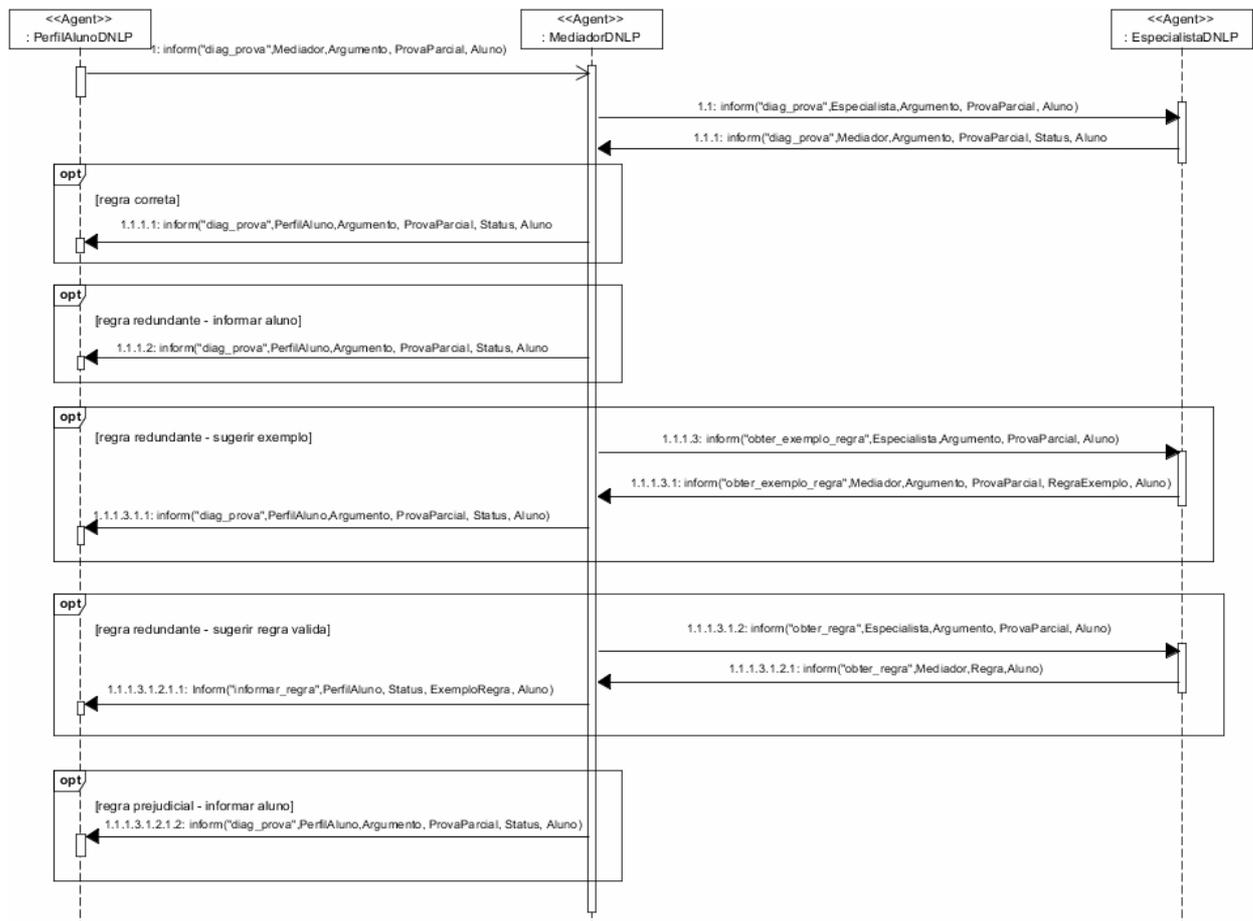


Figura 32 – Troca de mensagens entre os agentes em aplicação de uma regra

A Figura 33 mostra as mensagens trocadas quando o aluno está ocioso ou quando ele solicita uma ajuda.

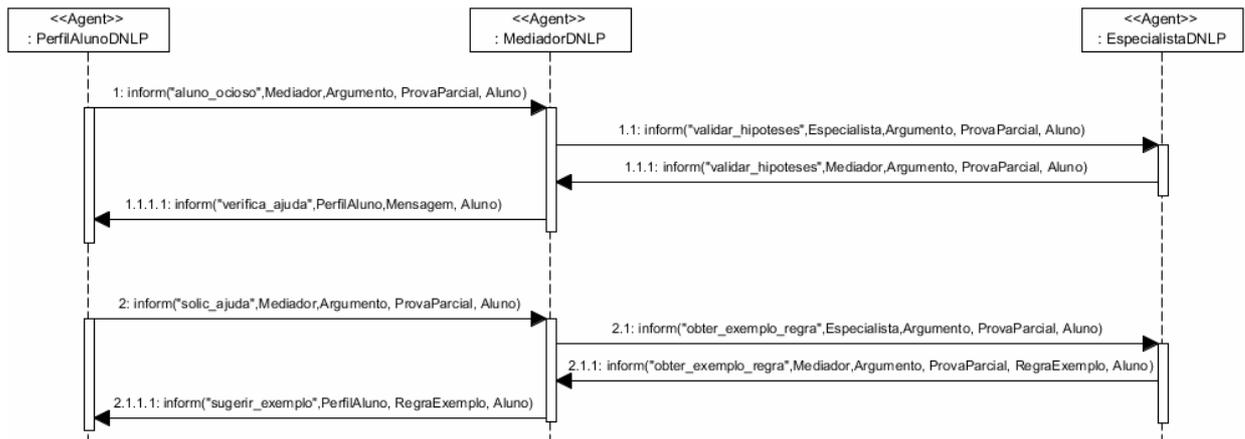


Figura 33 – Troca de mensagens entre os agentes quando o aluno está ocioso

Por fim, a Figura 34 mostra as mensagens trocadas entre os agentes quando o aluno deve ser informado do status da prova. Essa avaliação é realizada sempre que o aluno aplicar uma regra, desse modo, é possível que o aluno receba duas mensagens, uma informando o diagnóstico da regra aplicada e outra informando o status da prova.

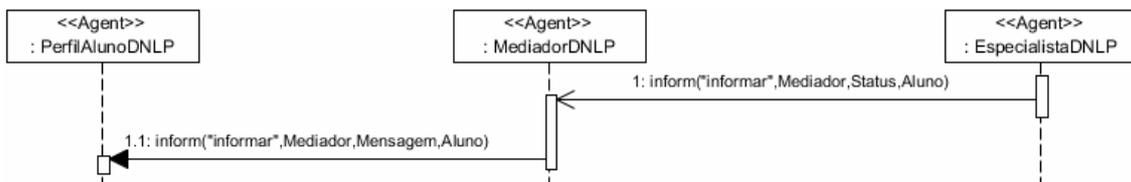


Figura 34 – Troca de mensagens entre os agentes quando o aluno é informado do status de sua prova

6.4 Desenvolvimento do Protótipo do Agente Mediador

O protótipo do agente Mediador foi desenvolvido através do mapeamento das estratégias de mediação e das interações especificadas nas seções 6.1 e 6.2 em planos de ação usando a linguagem de programação de agentes AgentSpeak(L).

A seguir, são mostrados dois exemplos de código aplicados ao agente Mediador. O primeiro exemplo, visualizado na Figura 35, mostra o plano de ação para quando o aluno inicia a prova de um novo argumento.

```

4 +novo_argumento(Argumento, Aluno): true <-
5   -novo_argumento(Argumento, Aluno);
6   .send(especialista, tell, novo_argumento(Argumento, Aluno)).

```

Figura 35 – Exemplo de plano para um novo argumento.

Este plano segue a estratégia delineada originalmente no diagrama da Figura 25. Na primeira linha, mostra-se o plano chamado `novo_argumento`. Para a execução desse plano, são necessários: o `Argumento` que o aluno tentará provar e o identificador do `Aluno` (número de matrícula, por exemplo). A seguir, a crença de que o novo argumento

foi iniciado é removido da base de conhecimento do agente, de modo que, se o aluno tentar provar esse argumento novamente em outro momento, o agente Mediador retome o processo de mediação. Por fim, a última linha envia uma mensagem ao agente Especialista chamada `novo_argumento`, com o Argumento que o aluno irá provar e o identificador do aluno.

No segundo exemplo, Figura 36, mostra-se o plano para quando o aluno aplicar uma regra no editor de provas. Esse plano implementa o início da estratégia apresentada no diagrama da Figura 27, começando a interação prevista no diagrama apresentado na Figura 32.

```
63 +diag_prova(Argumento, ProvaParcial, Aluno) [source(A)]: A == perfilaluno <-
64   -diag_prova(Argumento, ProvaParcial, Aluno) [source(A)];
65   .send(especialista, tell, diag_prova(Argumento, ProvaParcial, Aluno)).
66
```

Figura 36 – Exemplo de plano para uma aplicação de regra.

A primeira linha mostra o identificador do plano, `diag_prova`, e seus parâmetros: `Argumento`, `ProvaParcial` e `Aluno`, sendo que o `Argumento` é o que o aluno está tentando provar, a prova parcial é a prova parcial realizada até o momento e o identificador do aluno é sua identificação dentro do sistema respectivamente. Ainda assim, ocorre uma validação para verificar se o agente que enviou a mensagem foi o `perfilaluno`, uma vez que existem planos de ação distintos para cada agente. Em seguida, remove-se a mensagem da base de conhecimento, para caso o aluno envie a mesma mensagem ao agente Mediador. Por fim, o agente Mediador envia uma mensagem ao agente Especialista chamada `diag_prova` com o argumento que o aluno deseja provar, a prova parcial realizada até o momento e o identificador do aluno. Após receber essa mensagem, o agente Especialista irá validar se a regra aplicada pelo aluno é útil, redundante ou prejudicial. No caso de a regra ser prejudicial, por exemplo, o plano a ser executado por ser visualizado na Figura 37.

```
94 +diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)]: A == especialista
95   & Status == redundante & .count(primeiratentativaredundante(Aluno)) == 0 <-
96   -diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)];
97   +primeiratentativaredundante(Aluno);
98   .send(perfilaluno, tell, diag_hipotese("ok", Aluno)).
99
```

Figura 37 – Exemplo de plano para aplicação de uma regra prejudicial

O plano apresentado na Figura 37 implementa as ações previstas pela estratégia definida na Figura 28, para o caso da ação do aluno ser prejudicial. As duas primeiras linhas mostram o identificador do plano para diagnóstico de prova. Pode-se verificar que o nome do plano é o mesmo da Figura 36, porém, seus parâmetros e validações são diferentes. Esse plano receberá do agente especialista o argumento e a prova parcial que foram validados com o identificador do aluno e um status, definindo se a regra aplicada pelo aluno foi útil, redundante ou prejudicial. Pode-se verificar que esse plano será executado somente se o agente que enviou a mensagem foi o agente especialista. O status recebido foi redundante e não existe na base de conhecimento do agente a primeira tentativa de aplicação de regra que resultou como redundante. Na terceira linha é removida a crença de que o agente já recebeu a mensagem `diag_prova` e na terceira linha é adicionada uma crença, na base de conhecimento do agente, definindo que determinado aluno já aplicou a primeira regra redundante. Em seguida, envia-se a

mensagem `diag_prova` ao agente `perfilaluno` informando que a regra aplicada está “ok”. Apesar de ter sido diagnosticado que a regra aplicada é redundante, como foi a primeira vez que isso ocorreu nada deve ser feito por parte do editor de provas.

Na Figura 38, pode-se verificar que a identificação do plano é muito parecida com o da Figura 37, diferenciando unicamente na validação, onde se verifica, se o aluno já aplicou uma regra redundante (ver estratégia definida na Figura 28).

```

101 +diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)]: A == especialista
102   & Status == redundante & .count(primeiratentativaredundante(Aluno)) == 1
103   & .count(segundatentativaredundante(Aluno)) == 0 <-
104   -diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)];
105   +segundatentativaredundante(Aluno);
106   .send(perfilaluno, tell, diag_prova("Esta regra é considerada redundante, pois não contribui para a resolução.",
107       Argumento, ProvaParcial, Status, Aluno)).
108

```

Figura 38 – Exemplo de plano para a reaplicação de uma regra redundante.

A quarta linha remove a crença de que o agente já recebeu a mensagem `diag_prova`. A quinta linha adiciona uma crença na base de conhecimento do aluno, representando que pela segunda vez o aluno aplicou uma regra redundante. Ao final, o agente Mediador envia uma mensagem ao agente `perfilaluno` chamada `diag_prova` com um texto informando o aluno de que a regra é redundante, o argumento que o aluno está tentando provar, a prova parcial realizada até o momento, o status da aplicação da regra e o identificador do aluno. Essas informações serão armazenadas no agente `perfilaluno`.

Na Figura 39 pode-se verificar que a estrutura do plano é parecida com o plano apresentado nas Figuras 36, 37 e 38, contudo, passa-se a validar quantas regras redundantes, em sequência, o aluno já aplicou e com base nisso, define-se as ações a serem tomadas. Esse plano, juntamente com o plano da Figura 40 implementam a estratégia de sugestão de exemplo que é descrita na Figura 28.

```

109 +diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)]: A == especialista
110   & Status == redundante & .count(primeiratentativaredundante(Aluno)) == 1
111   & .count(segundatentativaredundante(Aluno)) = 1
112   & .count(terceiratentativaredundante(Aluno)) = 0 <-
113   -diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)];
114   +terceiratentativaredundante(Aluno);
115   .send(especialista, tell, obter_exemplo_regra(Argumento, ProvaParcial, Status, Aluno)).
...

```

Figura 39 – Exemplo de plano para aplicação de regras redundantes em sequência.

Pode-se verificar na Figura 39, que o agente Mediador envia uma mensagem para o agente Especialista chamada `obter_exemplo_regra`. Essa mensagem sugere ao agente Especialista que ele deve buscar a próxima regra a aplicada pelo aluno. Essa regra não será diretamente informada ao aluno, ela será utilizada na busca de um exemplo que mostre sua aplicação, esperando que o aluno identifique uma forma similar de aplicação em seu argumento. Sendo assim, o agente Especialista envia uma mensagem informando o agente Mediador qual regra deve ser aplicada através da mensagem `obter_exemplo_regra`, mostrada na Figura 40.

```

117 +obter_exemplo_regra(Argumento, ProvaParcial, Status, RegraExemplo, Aluno) [source(A)]: true <-
118   -obter_exemplo_regra(Argumento, ProvaParcial, Status, RegraExemplo, Aluno) [source(A)];
119   .send(perfilaluno, tell, sugerir_exemplo(Argumento, ProvaParcial, Status, RegraExemplo, Aluno)).

```

Figura 40 – Exemplo de plano para obtenção de uma regra a ser aplicada.

No plano apresentado acima, verifica-se que o agente Mediador, ao receber a regra que deve ser aplicada a seguir, envia a mensagem `sugerir_exemplo` ao agente `perfilaluno` para que ele se comunique com o editor de provas, possibilitando que o editor saiba que exemplo deve ser buscado.

Finalizando os planos de ação para regras redundantes, mostrados na Figura 28, as Figura 41 e 42 mostram o plano de ação para quando o aluno aplicar a quarta regra redundante consecutiva.

```

121 +diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)]: A == especialista
122     & Status == redundante & .count(primeiratentativaredundante(Aluno)) == 1
123     & .count(segundatentativaredundante(Aluno)) = 1
124     & .count(terceiratentativaredundante(Aluno)) = 1 <-
125     -diag_prova(Argumento, Status, ProvaParcial, Aluno) [source(A)];
126     -primeiratentativahipotese(Aluno);
127     -segundatentativahipotese(Aluno);
128     -terceiratentativahipotese(Aluno);
129     .send(especialista, tell, obter_regra(Argumento, ProvaParcial, Aluno)).

```

Figura 41 – Exemplo de plano para obtenção de uma regra a ser aplicada.

Na sexta, sétima e oitava linhas pode-se verificar que agente Mediador remove as crenças de que o aluno aplicou a sequencia de regras redundantes. Isso é feito para que quando o aluno aplique uma nova regra redundante à estratégia de mediação seja reiniciada, evitando que o aluno seja sempre informado de qual regra deve aplicar. Em seguida o agente Mediador solicita uma regra ao agente Especialista através da mensagem `obter_regra`. O qual irá devolver uma mensagem chamada `obter_regra` para o agente Mediador, com a regra que o aluno deve aplicar. O plano executado ao receber essa mensagem do agente Especialista pode ser vista na Figura 42.

```

132 +obter_regra(Regra, Aluno) [source(A)]: true <-
133     -obter_regra(Regra, Aluno) [source(A)];
134     .send(perfilaluno, tell, informar_regra(Regra, Aluno)).
135

```

Figura 42 – Exemplo de plano para informar o aluno qual regra deve ser aplicada.

Verifica-se, na Figura 42, que após o agente Mediador receber a regra do agente Especialista, é enviada uma mensagem ao agente PerfilAluno, chamada `informar_regra`. Ao receber essa mensagem, o agente PerfilAluno sabe que deve informar o aluno de qual regra ele deve aplicar.

7 EXPERIMENTOS E VALIDAÇÃO

Este capítulo apresenta o planejamento e os resultados do experimento realizado com o sistema Heráclito. O sistema Heráclito (nas versões Desktop e Móvel) está disponibilizado na página do Portal MILOS (Disponível em: <<http://obaa.unisinos.br/heraclito.htm>>) inicialmente na versão para sistemas operacionais Windows, juntamente com um questionário de avaliação inicial do Sistema de edição de provas e demonstrações do sistema Heráclito.

7.1. Objetivos do Experimento

O objetivo do experimento foi o de realizar uma avaliação qualitativa e quantitativa dos resultados alcançados pela presente dissertação em relação ao projeto do Sistema Heráclito.

São duas as principais contribuições dessa dissertação para o projeto do Sistema Heráclito:

1. O Editor de Provas, descrito no Capítulo 5, que forma o núcleo dos objetos de aprendizagem Heráclito;
2. O agente Mediador, descrito no Capítulo 6, que provê os serviços de apoio (mediação) pedagógico aos alunos do Sistema Heráclito.

Para avaliar essas contribuições experimentalmente foi criado de forma manual um objeto de aprendizagem Heráclito contendo o Editor de Provas e seu guia de utilização, uma lista com 10 de exercícios de dedução natural e um conjunto de exemplos de dedução, relacionado aos exercícios a serem resolvidos pelos alunos.

A Tabela 14 a seguir mostra os 10 exercícios de dedução submetidos aos alunos, que foram incluídos no objeto de aprendizagem de teste do Heráclito. Cada exercício é constituído de um argumento na forma:

$$P_1, P_2, \dots, P_n \vdash Q$$

As fórmulas P_1, P_2, \dots, P_n são as premissas (ou hipóteses) do argumento e a fórmula Q é a conclusão do argumento. O objetivo do exercício é elaborar uma prova formal, que demonstre que a conclusão Q se segue das premissas P_1, P_2, \dots, P_n . Para elaborar essa demonstração é usado o Editor de Prova, com suporte do serviço de tutoria.

Tabela 14 – Lista de Exercícios usada no Experimento do Heráclito

Nro.	Nível de dificuldade	Fórmula	Formato Textual usado no Heráclito
1	baixo	$P \wedge Q, P \rightarrow R, Q \rightarrow S \vdash R \wedge S$	$p \wedge q, p \rightarrow r, q \rightarrow s :-- r \wedge s$
2	baixo	$A \rightarrow B, B \rightarrow (C \rightarrow D), A \rightarrow (B \rightarrow C) \vdash A \rightarrow D$	$a \rightarrow b, b \rightarrow (c \rightarrow d), a \rightarrow (b \rightarrow c) :-- a \rightarrow d$
3	baixo	$P \rightarrow Q \vdash (P \wedge R) \rightarrow (Q \wedge R)$	$p \rightarrow q :-- (p \wedge r) \rightarrow (q \wedge r)$
4	baixo	$\neg P \rightarrow P \vdash P$	$\sim p \rightarrow p :-- p$
5	médio	$P \rightarrow (Q \vee R), \neg Q, \neg R \vdash \neg P$	$p \rightarrow (q \vee r), \sim q, \sim r :-- \sim p$
6	médio	$P \rightarrow Q \vdash \neg Q \rightarrow \neg P$	$p \rightarrow q :-- \sim q \rightarrow \sim p$
7	médio	$A \leftrightarrow Q, S \leftrightarrow R, A \wedge R \vdash S \wedge Q$	$a \leftrightarrow q, s \leftrightarrow r, a \wedge r :-- s \wedge q$
8	difícil	$P \leftrightarrow Q \vdash \neg P \leftrightarrow \neg Q$	$p \leftrightarrow q :-- \sim p \leftrightarrow \sim q$
9	difícil	$P \rightarrow Q \vdash (P \vee R) \rightarrow (Q \vee R)$	$p \rightarrow q :-- (p \vee r) \rightarrow (q \vee r)$
10	difícil	$(A \wedge B) \vee E, \neg G, C \rightarrow \neg A, \neg D \wedge (E \rightarrow (D \vee G)) \vdash \neg C$	$(a \wedge b) \vee e, \sim g, c \rightarrow \sim a, \sim d \wedge (e \rightarrow (d \vee g)) :-- \sim c$

Os exercícios foram divididos em três níveis de dificuldade em relação ao seu processo de resolução: 4 exercícios de fácil resolução (nível baixo de dificuldade) que exigem apenas o uso das 10 regras básicas de dedução e cujas provas são pouco extensas (menos de 10 passos), 3 exercícios mais complexos (nível médio) que com o auxílio das regras derivadas podem ainda ser resolvidos em menos de 10 passos. Os 3 exercícios finais (nível difícil) são mais complexos, mesmo com o uso de todas as regras básicas e derivadas ainda assim a demonstração exige mais de 10 passos.

Foi solicitado aos alunos que eles executassem o experimento em duas etapas: primeiro resolver os exercícios da lista apresentada na Tabela 14, com o uso do objeto de aprendizagem Heráclito e posteriormente preencher um questionário de avaliação sobre o objeto de aprendizagem. Foi dado um período de 10 dias para a condução do experimento (resolução da lista de exercícios e preenchimento do questionário).

7.2. Avaliação do Experimento

No experimento também se buscou avaliar a percepção dos alunos a respeito do uso do Heráclito. Para tanto foi criado um questionário de avaliação do experimento, que é apresentado no Anexo II dessa dissertação. Nesse questionário buscou-se avaliar a:

(1) Efetividade Pedagógica da Mediação: neste quesito é avaliada a percepção dos alunos em relação às estratégias de mediação empregadas pelo Sistema Heráclito, buscando identificar, também na percepção dos alunos, se Sistema Heráclito trouxe uma contribuição positiva capaz de atingir os objetivos de ensino de dedução natural.

(2) Qualidade do Conteúdo: neste quesito é avaliada a percepção dos alunos participantes com relação à seleção dos exercícios escolhidos e a qualidade do conteúdo apresentado.

(3) *Layout Visual*: neste quesito é avaliado se o Editor de Prova possui um *design* adequado ao aprendizado ao informar o conteúdo dos textos, ao posicionar imagens/figuras e ao apresentar botões e menus se estes formam um posicionamento intuitivo, proporcionando o fácil manuseio da ferramenta.

(4) *Usabilidade*: Neste quesito é avaliado o nível de usabilidade do Editor de Prova, onde é medida a percepção dos alunos sobre a facilidade de uso desta ferramenta.

O quesito de Efetividade Pedagógica da Mediação busca avaliar as características do processo de mediação no Sistema Heráclito, servindo de validação do modelo de mediação proposto para o agente Mediador no Capítulo 6.

Os quesitos de Layout Visual e Usabilidade buscam avaliar as propriedades da interface de usuário do Editor de Provas. A funcionalidade deste editor foi testada previamente em laboratório e em experimentos isolados em sala de aula.

O quesito de Qualidade de Conteúdo é apenas uma verificação cruzada para identificar a percepção dos alunos sobre a aplicabilidade do teste, em termos do conteúdo da disciplina de Lógica.

O questionário de avaliação foi composto por um total de 25 perguntas, sendo: 10 delas avaliando a contribuição do sistema tutor, 5 avaliando a qualidade do conteúdo apresentado, 3 avaliando o Layout e 7 avaliando a usabilidade do sistema e um registro opcional com informações adicionais.

7.3. Condução do Experimento

O experimento para validação do Sistema Heráclito foi feito em uma instituição de ensino superior com uma turma de graduação da disciplina de Lógica no semestre de 2012/2. O experimento foi conduzido após terem sido cobertos os conteúdos de operadores e fórmulas, tabelas-verdade e processo de dedução natural na Lógica Proposicional, logo após o grau A. Os alunos já haviam trabalhado em listas de exercício similares, mas com resolução manual e acompanhamento do professor.

Esse experimento foi iniciado em um laboratório de informática com a apresentação do objeto de aprendizagem de teste Heráclito descrito na seção 7.1. Nesta ocasião os alunos foram primeiramente orientados sobre a ferramenta, recursos, funcionalidades e formas de acesso. Os testes tiveram início no laboratório e posteriormente foram terminados pelos alunos em casa. Foi dado um período de 10 dias para a condução do experimento.

O processo de uso e resolução dos exercícios foi registrado por meio dos registros de ações (*logs*) do sistema. Após o efetivo uso, foi solicitado aos alunos que respondessem o formulário de avaliação do Sistema de edição de provas e demonstrações do sistema Heráclito (ver Anexo I).

7.4. Resultados e Análises do Experimento

Os resultados do experimento foram obtidos através de uma análise dos *logs* do sistema Heráclito, que indicam o uso efetivo da ferramenta pelos alunos. Este experimento foi realizado com uma turma teste com um total de 44 alunos matriculados. Destes, até a realização do experimento a turma contava com 35 alunos, pois 4 haviam cancelado a disciplina e 5 alunos desistiram da disciplina antes do experimento (alunos sem frequência suficiente). Dos 35 alunos que participaram do experimento, 22 efetivamente utilizaram o Sistema Heráclito e destes 22 alunos, 13 foram aprovados e 9 foram reprovados. As Tabelas 15 e 16 mostram esses dados de forma mais detalhada.

Tabela 15 – Total de alunos aprovados/reprovados com e sem o uso do Heráclito

	Total de alunos	Alunos que usaram o Sistema Heráclito	Alunos que não usaram o Sistema Heráclito
Aprovados na disciplina	16	13 (81%)	3 (19%)
Reprovados na disciplina	19	9 (47%)	10 (53%)
Desistentes da disciplina	9	-	-
Total de alunos	44	22 (50%)	13 (30%)

Tabela 16 – Total de alunos que usaram/não usaram e aprovaram/reprovaram na disciplina

	Total de alunos	Aprovados na disciplina	Reprovados na disciplina
Usaram o Sistema Heráclito	22	13 (59%)	9 (41%)
Não usaram o Sistema Heráclito	13	3 (23%)	10 (77%)
Desistentes da disciplina	9	-	-
Total de alunos	44	16 (36%)	19 (43%)

Para uma melhor constatação do efetivo uso do Sistema Heráclito por parte dos alunos que foram aprovados utilizou-se o Teste de Hipóteses e Significância juntamente com a técnica para diferenças de proporções.

O Teste de Hipótese e Significância permite aceitar e rejeitar uma hipótese determinando se as amostras observadas em determinado experimento diferem significativamente dos resultados esperados. A técnica de diferença de proporções permite que este teste seja realizado em através de proporções amostrais [SPIEGEL, 1978].

A hipótese básica que se quer testar é se o Heráclito influencia ou não no processo de aprovação dos alunos. Para tanto se podem utilizar os dados da Tabela 16, considerando os grupos de alunos que usaram e não usaram o Heráclito como populações amostrais usadas para testar a hipótese. No teste de diferença de proporções

são usadas duas proporções: a proporção P_1 corresponde ao percentual de alunos aprovados na disciplina na amostra de alunos que usaram o sistema Heráclito e a proporção P_2 corresponde ao percentual de alunos aprovados na amostra de alunos que não utilizaram o Heráclito. Os valores de P_1 e P_2 e a diferença dessas proporções são:

$$P_1 = 13 / 22 = 0,590$$

$$P_2 = 3 / 13 = 0,230$$

$$P_1 - P_2 = 0,360$$

A hipótese nula em relação a essas amostras diz que a priori não há diferença entre as proporções nas diversas populações possíveis de alunos, ou seja, que $p_1 = p_2$. Portanto a hipótese nula equivale a afirmar que o Sistema Heráclito não influenciou nas aprovações. Além disso, a hipótese nula também implica que a proporção geral p também deve ser igual a p_1 e p_2 : $p_1 = p_2 = p$. A proporção geral p pode ser estimada pela proporção P , calculada da seguinte forma (SPIEGEL, 1978, p. 304):

$$\begin{aligned} P &= (n_1P_1 + n_2P_2) / n_1 + n_2 \\ &= (13 \times 0,590 + 3 \times 0,230) / 35 \\ &= (7,67 + 0,69) / 35 \\ &= 8,36 / 35 \\ &= 0,239 \end{aligned}$$

Onde n_1 e n_2 é o tamanho de cada amostra (número de alunos em cada amostra).

Com esses valores é possível estimar o desvio padrão $\sigma_{P_1-P_2}$ através do seguinte cálculo (SPIEGEL, 1978, p. 305):

$$\begin{aligned} \sigma_{P_1-P_2} &= \sqrt{P(1-P)(1/n_1 + 1/n_2)} \\ &= \sqrt{0,239 \times 0,761 \times (1/22 + 1/13)} \\ &= \sqrt{0,239 \times 0,761 \times (0,045 + 0,077)} \\ &= \sqrt{0,239 \times 0,761 \times 0,122} \\ &= \sqrt{0,022} \\ &= 0,148 \end{aligned}$$

Com os valores de P_1 , P_2 e do desvio padrão é possível estimar o valor para a variável padronizada Z usada para testar a hipótese nula:

$$\begin{aligned} Z &= (P_1 - P_2) / \sigma_{P_1-P_2} \\ &= 0,360 / 0,148 = 2,432 \end{aligned}$$

Como estamos buscando verificar se o uso do Heráclito causa um impacto no processo de aprovação, pode-se trabalhar com um teste unilateral da hipótese nula (SPIEGEL, 1978, p. 302).

Nesse caso, valores de Z maiores que 2,33 indicam, com um nível de significância de 0,01, que a hipótese nula *não pode ser aceita*, ou seja, com um grau muito baixo de probabilidade de erro, o Sistema Heráclito causa um impacto significativo no processo de aprovação.

Ainda mais, pelas proporções P_1 e P_2 , o impacto é claramente positivo: o uso do Sistema Heráclito muito provavelmente aumenta a chance do aluno ser aprovado.

As principais limitações do experimento estão relacionadas ao pequeno número das amostras e ao processo de seleção das populações de alunos. Por questões técnicas foi possível apenas realizar este teste com uma turma, limitando então o tamanho das amostras. Independentemente disso, os alunos dessas populações se selecionaram de forma autônoma, sem a influência do professor ou dos monitores que acompanharam o experimento. Cada aluno decidiu por conta própria se queria ou não participar do experimento. Isso implica que não há uma garantia de aleatoriedade no processo de seleção.

Certamente novos experimentos precisarão ser conduzidos para tratarem destas limitações, porém, o grau de significância obtido de 0,01 é bastante alto, fornecendo um indício forte do impacto do Heráclito.

Com relação à resolução dos exercícios pelos alunos, obteve-se um total de 135 exercícios realizados (dados extraídos dos Logs) com a ajuda do Sistema Heráclito. Destes 135, 69 deles eram considerados de nível baixo, 43 deles eram de nível médio e 23 considerados de nível difícil. A Figura 43 abaixo mostra estes valores em percentuais.

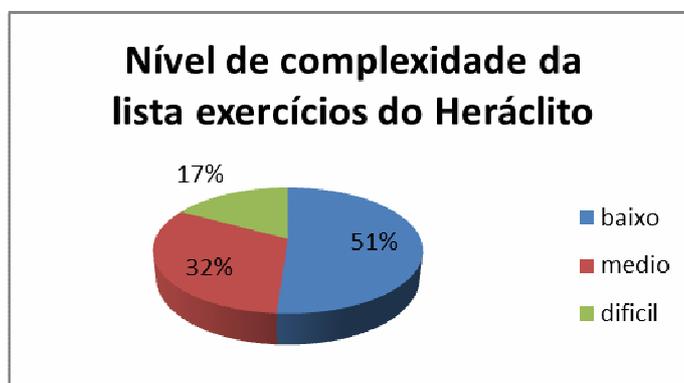


Figura 43 – Nível de complexidade dos exercícios em %

Os 22 alunos que resolveram os 135 exercícios foram distribuídos por quantidades (exemplo: 2 alunos resolveram os 10 exercícios propostos, 1 aluno resolveu 9 exercícios propostos, etc...). Os percentuais podem ser vistos na Tabela 17 abaixo:

Tabela 17 – Distribuição de exercícios realizados por alunos no experimento

Nº. Exercícios Realizados	Nº. Alunos	Percentual de Exercícios Realizados
10	2	9%
9	1	4%
8	5	23%
7	6	27%
6	2	9%
5	1	4%
4	1	5%
2	1	5%
1	1	5%
0	2	9%
135	23	100%

7.5 Resultados e Análise da Avaliação do Experimento

Os resultados da avaliação do experimento obtidos através do questionário de avaliação estão sumarizados nas tabelas a seguir, sendo apresentados separadamente por quesito (Anexo III).

A Tabela 18 mostra os resultados obtidos para 8 das 10 perguntas feitas aos alunos com relação à contribuição do Tutor no aprendizado do aluno.

Tabela 18 – Sumário dos resultados do quesito efetividade pedagógica da mediação

Efetividade Pedagógica da Mediação						
Questões		Resultados em percentual				
		Concordo	Concordo Parcial.	Indiferente	Discordo Parcial.	Discordo
1	Como aluno você acredita que o Heráclito de uma forma geral tenha facilitado o aprendizado?	70%	30%	0%	0%	0%
2	Os serviços de tutoria oferecido pelo Heráclito contribuíram para uma melhor compreensão do desenvolvimento dos exercícios?	61%	30%	0%	9%	0%
3	Você acredita que a estratégia de sugestão de exemplos na inserção de hipóteses incorretas ajuda no	57%	35%	0%	4%	4%

	processo de aprendizado?					
4	Você acredita que a estratégia de sugestão de regras na inserção de regras incorretas ajuda no processo de aprendizado?	57%	30%	9%	4%	0%
5	Você acredita que a estratégia de sugestão de ajuda ao aluno quando este se encontra “ocioso” ajuda no processo de aprendizado?	70%	13%	17%	0%	0%
6	Você acredita que a estratégia que avisa ao aluno quando este aplica uma regra prejudicial (sem volta) ajuda no processo de aprendizado?	83%	13%	4%	0%	0%
7	Você acredita que a estratégia que o tutor utiliza quando identifica que a metade da prova já foi realizada, enviando uma mensagem de reforço ao aluno pode ajudar no processo de aprendizado?	82%	9%	9%	0%	0%
8	Você acredita que a estratégia que o tutor utiliza quando identifica que faltam 1 ou 2 passos para finalizar a prova, enviando uma mensagem de reforço ao aluno, pode ajudar no processo de aprendizado?	70%	22%	4%	4%	0%
	Avaliação Média Final da Efetividade Pedagógica:	69%	23%	5%	2%	1%

Pode-se verificar que a grande maioria dos alunos apresentou respostas favoráveis a utilização do sistema, sendo destacadas as perguntas 6 e 7. Na pergunta 6, o Tutor intercede durante a prova informando o aluno de que ele não poderá concluir a prova, isso em função da última regra aplicada. Essa estratégia auxilia no esclarecimento de que certos tipos de regras não podem e não devem ser aplicadas, permitindo que o aluno repense seu raciocínio lógico de resolução e faça uma reflexão acerca dos conceitos das regras para uma nova escolha e possível conclusão da prova.

Na pergunta 7, os alunos concordam que o status do andamento da prova auxilia no processo de aprendizado, uma vez que situa o aluno dentro de seu próprio contexto indicando que está no caminho certo e têm grandes possibilidades de concluir a prova quem questão. Por esta pergunta fica claro que ao deixar o aluno ciente de sua situação existe uma motivação que impulsiona no processo de aprendizagem favorecendo o término do exercício proposto.

Além disso, pode ser observado que este quesito obteve, em média, 69% de aprovação sem nenhum tipo de restrição. Ainda assim, se considerarmos essa concordância com as parciais, 23%, tem-se um resultado de 92% de aprovação, demonstrando claramente que a mediação exerce um papel fundamental no processo de ensino-aprendizagem.

As perguntas 9 e 10 dizem respeito à quantidade de argumentos que foram provados com e sem a ajuda do Tutor. Os alunos afirmaram ter provado um total de 234 argumentos sendo que em 70 casos o uso do Tutor foi essencial para o término da prova, o que representa aproximadamente 30% dos casos. Isso indica uma contribuição

efetiva do processo de mediação.

A Tabela 19 mostra os resultados percentuais obtidos para as 5 perguntas relacionadas com o conteúdo apresentado pelo sistema Heráclito.

Tabela 19 – Sumário dos resultados do quesito qualidade do conteúdo apresentado

Qualidade do conteúdo						
Questões		Resultados em percentual				
		Concordo	Concordo Parcial.	Indiferente	Discordo Parcial.	Discordo
1	O conteúdo da lista de exercícios foi facilmente entendido e realizado no Heráclito?	62%	30%	4%	4%	0%
2	O nível de dificuldade da lista de exercícios foi distribuída de forma proporcional com a relação ao conteúdo?	48%	39%	4%	9%	0%
3	A lista de exercícios cobre boa parte do conteúdo de dedução natural?	70%	30%	0%	0%	0%
4	Os exemplos ajudaram SIM no processo de resolução do exercício.	48%	25%	9%	9%	9%
Avaliação Média Final da Qualidade do Conteúdo:		57%	31%	4%	6%	2%

Aqui, de forma similar ao caso da avaliação da mediação, também houve uma concordância significativa dos alunos que o conteúdo abordado tinha qualidade apropriada para o ensino de dedução natural na Lógica Proposicional, totalizando 88% de concordância em geral sobre a boa qualidade (57% de concordância sem restrições e 31% de concordância qualificada). Isso é importante porque demonstra a aplicabilidade do OA para fins pedagógicos.

Na Tabela 19, podem-se destacar as perguntas 1 e 3 que abordam a dificuldade de entendimento dos exercícios propostos bem como a preocupação por parte do professor em atender o conteúdo da matéria em específico em 10 questões. Como resultado, mais da metade dos alunos não tiveram dificuldade em entender o conteúdo da lista de exercícios proposta para resolução no sistema Heráclito. Na pergunta 4 observou-se que os exemplos selecionados para auxiliar os alunos foram aceitos por boa parte deles. Nesta pergunta, se somados seus percentuais de concordância obtém-se 73% de alunos que acreditaram que os exemplos auxiliaram, de alguma forma, no processo de resolução. Com relação aos 18% que discordaram parcial ou completamente, verificou-se que em alguns casos não ficou claro qual a relação entre o exemplo sugerido e a próxima regra que o aluno deveria aplicar. Essa questão ficou clara ao analisar as sugestões que os alunos fizeram acerca da ferramenta.

A Tabela 20 mostra os resultados percentuais obtidos para as 3 perguntas relacionadas com o Layout do editor de provas.

Tabela 20 – Sumário dos resultados do quesito *Layout Visual*

Qualidade do <i>Layout Visual</i>						
Questões		Resultados em percentual				
		Concordo	Concordo Parcial.	Indiferente	Discordo Parcial.	Discordo
1	O Heráclito possui um layout adequado ao aprendizado?	42%	38%	10%	10%	0%
2	Na tela de Edição do objeto, o espaço para demonstração da prova é suficiente?	66%	24%	5%	5%	0%
3	Na tela de Edição do objeto, os botões de regras de inferência são intuitivos e as informações claras?	71%	24%	0%	5%	0%
Avaliação Média Final da <i>Layout Visual</i>		60%	29%	5%	7%	0%

Pode-se verificar que a grande maioria dos alunos acredita que a ferramenta se mostra intuitiva, como pode ser visto na pergunta 2, além de que, de algum modo, auxilia no processo de aprendizagem, conforme 80% dos alunos que responderam a questão 1 da Tabela 19.

Além disso, pode ser observado que este quesito obteve, em média, 60% de aprovação sem nenhum tipo de restrição. Ainda assim, se considerarmos essa concordância com as parciais, 29%, tem-se um resultado de 89% de aprovação, demonstrando que o *Layout* é de certa forma intuitiva e auxilia no processo de aceitação da ferramenta. Com relação aos 7% dos alunos que discordaram parcialmente acerca do *Layout*, verificou-se que grande parte desses tiveram problemas relacionados à resolução dos monitores.

Por fim, a Tabela 21 mostra os resultados percentuais obtidos para as 5 perguntas relacionadas com a usabilidade do editor de provas.

Tabela 21 – Sumário dos resultados do quesito usabilidade do editor de provas

Usabilidade						
Questões		Resultados em percentual				
		Concordo	Concordo Parcial.	Indiferente	Discordo Parcial.	Discordo
1	NÃO consegui usar o objeto de aprendizagem Heráclito	0%	15%	5%	0%	80%
2	Quando estou utilizando o conteúdo para aprendizado não consigo entender o propósito da ferramenta	0%	5%	5%	20%	70%
Avaliação Média das questões gerais sobre Usabilidade		0%	10%	5%	10%	75%

3	Os menus da página principal do Heráclito funcionam?	75%	20%	5%	0%	0%
4	Os botões da página principal do Heráclito funcionam?	80%	15%	0%	5%	0%
5	Na página de Edição do objeto, as janelas de dialogo que são abertas em algumas das regras são fáceis de serem compreendidas e usadas?	55%	30%	10%	0%	5%
6	Na página de Edição do objeto, a sistemática de clicar na regra e depois nas linhas de demonstração é de fácil compreensão e utilização?	68%	11%	21%	0%	0%
7	O manual do Heráclito contempla as possíveis dúvidas do usuário?	43%	26%	26%	5%	0%
Avaliação Média sobre pontos específicos da Usabilidade		64%	20%	13%	2%	1%

Nesta etapa foi possível avaliar as questões de âmbito gerais e questões específicas sobre a usabilidade. Em ambos os casos o sistema Heráclito teve boa aceitação, obtendo nas perguntas 1 e 2 uma média de 85% (75% de concordância sem restrições e 10% de concordância qualificada) e nas demais perguntas uma média de 84% (64% de concordância sem restrições e 20% de concordância qualificada) de aceitação na interface do sistema Heráclito.

Pode-se verificar na pergunta 1 que poucos alunos tiveram problemas em utilizar o sistema Heráclito e suas funcionalidades apresentadas. Contudo 15% dos alunos tiveram problemas relacionados à incompatibilidade aos sistemas operacionais (em sua atual versão o sistema Heráclito apenas é compatível com o sistema operacional Windows).

Ainda assim, a maioria dos alunos reporta que não tiveram grandes problemas em entender o que o sistema Tutor estava tentando mostrar ao longo do processo de resolução do exercício, como pode ser visto na pergunta 2. Com relação às demais perguntas verifica-se que o editor de provas se mostra funcional, atendendo as expectativas dos alunos.

A combinação dos resultados sumarizados da usabilidade e do layout visual do Editor de Provas indica 86% de aprovação geral, com 66% de aprovação sem restrições e 20% de aprovação parcial. Isso é um forte indicador que os requisitos de usabilidade deste editor estão sendo atingidos do ponto de vista das pessoas para quem isso mais importa que, são os alunos, justamente os usuários finais desse sistema.

Os problemas detectados pelos usuários e reportados no questionário estão sendo tratados para melhorar a qualidade do Editor.

8 CONCLUSÕES

O processo de pesquisa dessa dissertação requereu uma ampla análise do quadro atual de pesquisas sobre as tecnologias de objetos de aprendizagem e ambientes virtuais de ensino. A partir desse quadro geral a dissertação se centrou nas questões educacionais relativas ao ensino de Lógica, ou mais especificamente, no ensino de dedução natural na Lógica. Foram exploradas as possibilidades de aplicação de tecnologias computacionais baseadas na combinação de agentes pedagógicos e objetos de aprendizagem para apoiar esse processo de ensino.

Em termos específicos, o presente trabalho objetivou o desenvolvimento de um modelo computacional de mediação apropriado para o ensino da Dedução Natural para a Lógica Proposicional, incorporado na forma de um agente pedagógico que auxilie o aluno em seu processo de aprendizagem, servindo como ferramenta de apoio para esse processo.

As estratégias de ensino desenvolvidas neste trabalho se mostraram capazes de fornecer uma importante ajuda no processo de aprendizado da disciplina de Lógica, como aponta a média de 69% de aprovação sem nenhum tipo de restrição e o somatório com as parciais que resultou em 92% de aprovação, demonstrando claramente que a mediação exerce um papel fundamental no processo de ensino-aprendizagem.

É importante notar que esse índice resulta da percepção dos alunos que efetivamente interagiram com o sistema. Isso fornece indícios sólidos que o Sistema Heráclito consegue transmitir aos alunos que é realmente um tutor capaz de ajudar na aprendizagem de dedução na Lógica Proposicional.

Essa é uma tarefa importante (e difícil) nos processos de ensino-aprendizagem, qualquer professor ou tutor humano sabe como é importante ganhar a confiança dos alunos, mostrando que é capaz de ajudá-los a aprender determinado tópico de ensino. Os índices de aprovação vistos acima mostram que o processo de mediação (tutoria) do Sistema Heráclito está no caminho certo.

Ainda com relação à contribuição das estratégias de aprendizagem, verificou-se que a ajuda do Tutor foi imprescindível na prova de aproximadamente 30% de todos os argumentos. Nesse contexto, acredita-se que 30% não seja somente um número, mas é também parte de todo um processo de construção de conhecimento sobre a lógica que os alunos desenvolveram durante a prova desses argumentos. Com a boa aceitação das estratégias do Tutor, ressalta-se a importância dessas estratégias de ensino, que por esta afirmação fica claro que o auxílio possibilitado pela mediação na resolução das questões está dentro de um quadro motivacional que impulsiona o processo de aprendizagem, favorecendo o término das questões propostas.

No que diz respeito à utilização da ferramenta, os alunos se mostraram bastantes favoráveis como demonstrado nas questões onde se pergunta “Como aluno você acredita que o Heráclito de uma forma geral tenha facilitado o aprendizado?” e “Os serviços de tutoria oferecida pelo Heráclito contribuíram para uma melhor compreensão do desenvolvimento dos exercícios?”. As respostas a essas perguntas mostram que os alunos acreditam que o sistema tutor facilitou o processo de aprendizado e que também

contribuiu para uma melhor compreensão do desenvolvimento dos exercícios.

Essas avaliações dos alunos fornecem indícios importantes que a utilização do sistema Heráclito poderá contribuir para a melhora dos níveis de aprovação. A verificação se essa melhora se confirma ou não é um importante trabalho futuro de pesquisa sobre o Sistema Heráclito.

Por fim, o Layout e Usabilidade se mostraram favoráveis à adesão da ferramenta por parte dos alunos, uma vez que eles não tiveram maiores problemas em entender o sistema Tutor e a interação realizada por meio de suas mensagens.

Como sugestão para melhoramento da ferramenta tem-se algumas funcionalidades vindas por parte dos alunos tais como: a opção de mostrar em tela a correta utilização da regra. Isso poderia ser feito por meio do posicionamento do mouse em cima da regra, de modo que o sistema mostraria a utilização da regra, eliminando a necessidade de acessar o manual (ou outro meio) para identificar o uso de determinada regra.

Outras sugestões de melhorias acolhidas foram com relação à gravação dos exercícios e aos tipos de regras a serem utilizadas. Quanto aos exercícios realizados, busca-se uma forma onde os dados estejam disponíveis para o usuário acessá-los e analisa-los posteriormente. Quanto aos tipos de regras a serem aplicadas, os alunos acreditam que o processo de aprendizagem pode ser melhorado se eles forem informados quando uma regra básica de dedução ou uma regra derivada deve ser aplicada, no lugar da sugestão explícita da regra. Acredita-se que com o sistema auxiliando neste ponto os alunos passariam a desenvolver melhor a sua percepção e visão quanto ao uso das regras.

As últimas sugestões estão relacionadas à usabilidade, sendo estas de ordem mais técnica, como a compatibilidade entre sistemas operacionais, o auto ajuste do sistema em resoluções distintas, bem como as funcionalidades citadas acima que já estão em fase de desenvolvimento.

Quanto a trabalhos futuros, conforme comentado antes, mostra-se importante fazer uma avaliação para confirmar a expectativa acerca de melhoras no processo de ensino-aprendizagem fazendo uma análise experimental sobre a eficácia do apoio ao ensino, preferencialmente com a utilização do grupo (ou possivelmente turma) de controle. Note-se que, como qualquer processo de avaliação de um método de ensino, tais experimentos para a medida de eficácia são sempre muito subjetivos e relativos ao contexto em que são aplicados, não fornecendo comprovações ou certificações sobre a qualidade do método em geral. O processo de mediação no Sistema Heráclito é um método de ensino implementado computacionalmente, sofrendo das mesmas dificuldades de avaliação de sua eficácia. Apesar disso, tais experimentos podem contribuir para uma melhor compreensão sobre as capacidades do processo de mediação e sobre como ele pode ser usado de forma mais proficiente em contextos de ensino.

Das pesquisas empreendidas durante a dissertação foram produzidos três artigos científicos publicados em anais de eventos. Dois artigos apresentam análises sobre a evolução das pesquisas sobre as tecnologias de OA e AVA. O artigo “Papel dos Objetos de Aprendizagem como Recurso Pedagógico em AVAs” de PENTEADO e GLUZ foi publicado nos Anais do LACLO 2011, enquanto o artigo “Suporte Técnico/Pedagógico aos OAs pelos AVAs: uma Análise Comparativa das Pesquisas Recentes” de

PENTEADO, GLUZ e GALAFASSI, foi publicado nos anais do SBIE 2011. O artigo foi reimpresso no Anexo III.

O terceiro artigo “Sistema Heráclito: Suporte a Objetos de Aprendizagem Interativos e Dialéticos Voltados ao Ensino de Dedução Natural na Lógica Proposicional” de PENTEADO e GLUZ, também foi publicado nos anais do SBIE 2011. Este artigo, que foi reimpresso no Anexo IV, apresenta uma versão inicial da arquitetura do Sistema Heráclito, que é a ferramenta computacional de apoio ao ensino de Lógica empregada nessa dissertação.

REFERENCIAS

ADL – Advanced Distributed Learning. SCORM 2004 4th Edition Version 1.1 Documentation. Disponível em: <<http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Documentation.aspx>>. Acesso em: nov. 2010.

ALONSO, J. A.; ARANDA, Gonzalo A.; MARTN-MATEOS, F. J. KRRT: Knowledge Representation and Reasoning Tutor System. Disponível em: <<http://150.214.140.135/~jalonso/publicaciones/2007-EUROCAST.pdf>>. Acesso em: Ago 2012.

BARNES, T.; STAMPER, J. Automatic Hint Generation for Logic Proof Tutoring Using Historical Data. In Journal of Educational Technology & Society, Special issue on Intelligent Tutoring Systems. Vol. 13 Edição, 1. 2010. Disponível em: <http://www.academia.edu/442299/Automatic_Hint_Generation_for_Logic_Proof_Tutoring_Using_Historical_Data>. Acesso em: Ago 2012.

BAUER, B; ODELL, J. UML 2.0 and agents: how to build agent-based systems with the new UML standard. Eng. Appl. Artif. Intell. N. 18, Vo. 2, March, 2005. p.141-157.

BELLIFEMINE, Fábio; CAIRE, Giovanni; GREENWOOD, Dominic. Developing Multi-Agent Systems with JADE. Wiley Series in Agent Technology, Series Editor: Michael Wooldridge, Liverpool University, UK. 2007.

BERNI, Regiane I G. MEDIAÇÃO: O CONCEITO VYGOTSKYANO E SUAS IMPLICAÇÕES NA PRÁTICA PEDAGÓGICA. Disponível em: <http://www.filologia.org.br/ileel/artigos/artigo_334.pdf>. Acesso em: jan. 2013.

BORDENAVE, JUAN D., PEREIRA, ADAIR M. Estratégias de Ensino-Aprendizagem. Editora Vozes, 20ª edição, 1999.

BORDINI, Rafael H, HÜBNER, Jomi Fred, WOOLDRIDGE, Michael. Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley Series in Agent Technology, 2007.

BRASIL. Ministério da Educação. Secretaria de Educação a Distância. Objetos de aprendizagem: uma proposta de recurso pedagógico. Organização: PRATA, C., NASCIMENTO, A. C. – Brasília: MEC, SEED, 2007. 154 p.

BRENNER, W.; RÜDIGER, Z.; WITTIG, H. Intelligent Software Agents. Foundations and applications. Berlin: Springer-Verlag, 1998.

COQ Proof Assistant. Disponível em: <<http://coq.inria.fr/>>. Acesso em: Ago 2012.

CROY, Marvin; BARNES, Tiffany; STAMPER, John. Towards an Intelligent Tutoring System for Propositional Proof Construction. University of North Carolina at Charlotte. Disponível em: <http://www.philosophy.uncc.edu/mjcroy/croy_pubs/Croy_Barnes_Stamper_ECAP.pdf>. Acesso em: Ago 2012.

DIAS, C. L., KEMCZINSKI, A. L., SÁ, S. V., FERLIN, J., HOUNSELL, M. S. Padrões abertos: aplicabilidade em Objetos de Aprendizagem (OAs). XX Simp. Bras. de Inf. na educação. SBIE 2009.

DIRETRIZES CURRICULARES NACIONAIS para os cursos de graduação em Computação. Disponível em: <http://portal.mec.gov.br/index.php?option=com_content&view=article&id=17616&Itemid=866#agosto>. Acessado em 05/10/2012.

DIRETRIZES CURRICULARES NACIONAIS. Cursos de Bacharelado e Licenciatura em Computação. Disponível em: <http://www.sbc.org.br/index.php?option=com_jdownloads&Itemid=195&task=finish&cid=186&catid=36>. Acesso em: nov. 2011.

DUTRA, R. L., TAROUCO, L. Objetos de Aprendizagem: Uma comparação entre SCORM e IMS Learning Design. Disponível em: <http://www.cinted.ufrgs.br/renote/jul2006/artigosrenote/a1_20138.pdf>. Acesso em: out. 2011.

ECLIPSE, Foundation. Eclipse - The Eclipse Foundation open source community website. Disponível em: <<http://www.eclipse.org>>. Acesso em Set: 2012.

EUZENAT, Jérôme; SHVAIKO, Pavel. Ontology Matching. Editora Springer, 2007.

FERREIRA, AURELIO BUARQUE DE HOLANDA. Mini Dicionário Aurélio – O dicionário da Língua Portuguesa. 8ª Edição. Editora Positivo, 2010.

FILETO, Renato. Construindo Aplicações na Web Semântica. Disponível em: <http://www.inf.ufsc.br/~fileto/Disciplinas/INE6616-2010-3/Aulas/01-SW_6pp.pdf>. Acesso em: jan. 2011.

FIPA. FIPA Specifications - Repository. Disponível em: <<http://www.fipa.org/specs/fipa00023/SC00023K.pdf>>. Acesso em: jan. 2011.

FLORES, C.; SEIXAS, L.; GLUZ, J. C.; VICARI, R. A Model of Pedagogical Negotiation. Lecture Notes in Computer Science, Heidelberg, v. 3808, n. 2005, p. 488499, 2005.

GLUZ, Carlos J; PASSERINO, Liliana M.; VICARI, Rosa Maria. Um Modelo Formal para Processos de Mediação em AVAs. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/757/743>>. Anais do Simpósio Brasileiro de Informática na Educação (SBIE), 2008.

GLUZ, Carlos J; PASSERINO, Liliana M.; VICARI, Rosa Maria. Uma proposta para Mediação Tecnológica em Espaços Virtuais de Aprendizagem. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/573>>. Anais do Simpósio Brasileiro de Informática na Educação (SBIE), 2007.

GLUZ, J. C.; VICARI, R. M. MILOS: Infraestrutura de Agentes para Suporte a Objetos de Aprendizagem OBAA. Anais do XXI SBIE, 2010.

GLUZ, J.C.; PY, M. Lógica para Computação. Coleção EAD. Editora Unisinos, 2010.

GLUZ. J. C. INTRODUÇÃO A INFRAESTRUTURA MILOS. Pós-Graduação em

Computação Aplicada (PIPCA), Universidade do Vale do Rio dos Sinos (UNISINOS). Maio, 2010.

GOOGLE Inc. Android. Disponível em: <<http://developer.android.com/>>. Acesso em Set: 2012.

HOL4. Disponível em: <<http://hol.sourceforge.net/>>. Acesso em: Ago 2012.

IEEE LTSC. Std1484.12.1 IEEE Learning Technology Standard Committee (LTSC) Standard for Learning Object Metadata (LOM). IEEE, 2002.

JAPE. Disponível em: <<http://www.cs.ox.ac.uk/people/bernard.sufrin/jape.html>>. Acesso em: Ago 2012.

KRATZ, R. A., CRESPO, S. SCOPEL, M., BARBOSA, J. Fábrica de Adequação de Objetos de Aprendizagem. Revista Brasileira de Informática na Educação. Volume 15 – Número 3 – Setembro a Dezembro de 2007. Disponível em: <<http://br-ie.org/pub/index.php/rbie/article/view/23>>. Acesso em: out. 2011.

LÓGICA. Ementa da disciplina. Universidade do Vale do Rio dos Sinos – UNISINOS, 2010.

LUKINS, Stacy; LEVICKI, Alan; BURG, Jennifer. A Tutorial Program for Propositional Logic with Human/Computer Interactive Learning. Department of Computer Science Wake Forest University. Winston-Salem, NC. Disponível em: <<http://csweb.cs.wfu.edu/~burg/BurgMain/public/media/PropLogic.pdf>>. Acesso em: Ago 2012.

MASON, R., WELLER, M. e PEGLER., C. Learning in the Connected Economy. Londres: Open University, 2003.

MENEGOLLA, Maximiliano; SANT'ANNA, Ilza Martins. Porque planejar? Como planejar? Petrópolis, RJ: Vozes, 2010.

MERCERON, Agathe; YACEF, Kalina. An Intelligent Teaching Assistant System for Logic. 12th International Conference on Artificial Intelligence in Education. Disponível em: <http://sydney.edu.au/engineering/it/~kalina/publis/lesta_yacef_its02.pdf>. Acesso em: Ago de 2012.

MOREIRA, Marco Antônio. Teorias de Aprendizagem. São Paulo: EPU, 1999.

MORENO, Antônio. Mathematical Logic Tutor - Propositional Calculus. Disponível em: <<http://aracne.usal.es/congress/PDF/AntonioMoreno.pdf>>. Acesso em: Ago 2012.

MOSSMAN, Marcel; GOMES, Lucas dos S; GLUZ, J.C. Objetos de Aprendizagem Móveis para Ensino de Dedução Natural na Lógica Proposicional. SBIE. Simpósio Brasileiro de Informática na Educação. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/1752/1513>. Acesso em dez: 2012.

OHLSSON, S. Some Principles of Intelligent Tutoring. In: LAWLER, R.; & YASDANI, M. Intelligence Tutoring. Systems. USA: Ablex, 1987.

OLIVEIRA, Marta Kohl de. VYGOTSKY – Aprendizado e desenvolvimento: Um processo sócio histórico. Editora Scipione. São Paulo, 1999.

OLIVEIRA, Stefan. HRCSystem: Sistema Multiagente BDI como Auxílio na Gestão de Profissionais por Competências. Universidade do Vale do Rio dos Sinos – UNISINOS. Programa de Pós-Graduação em Computação Aplicada. Dissertação de Mestrado. Disponível em: http://bdt.unisinos.br/tde_arquivos/1/TDE-2009-08-18T112732Z-860/Publico/StefanRosaComputacaoP.pdf. Acesso em: set de 2012.

ORACLE Inc. Welcome to NetBeans. Disponível em: <http://netbeans.org>. Acesso em set: 2012.

OTTER. Disponível em: <http://www.cs.unm.edu/~mccune/otter/>. Acesso em: Ago 2012.

PANDORA. Disponível em: <http://www.doc.ic.ac.uk/pandora/newpandora/>. Acesso em: Ago 2012.

PASSERINO, L.; GLUZ, J.; VICARI, R. MEDIATEC Mediação Tecnológica em Espaços Virtuais para Apoio ao Professor Online. RENOTE. Revista Novas Tecnologias na Educação, v. 5, p. 1, 2007.

PEREIRA, Alice T. Ambientes Virtuais de Aprendizagem – Em diferentes Contextos. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

PROTÉGÉ. Disponível em: <http://Protégé.stanford.edu/>. Acesso em out. 2011.

PROVER9. Disponível em: <http://www.cs.unm.edu/~mccune/prover9/>. Acesso em: Ago 2012.

RUSSEL, S.; NORVIG, P. Inteligência Artificial. 7ª Ed. Rio de Janeiro - Centro. Editora Campus, 2004.

SEIXAS, Louise. Estratégias Pedagógicas para um Ambiente Multiagente Probabilístico Inteligente de Aprendizagem – AMPLIA. Dissertação (Doutorado em Informática na Educação) – Centro Interdisciplinar de Tecnologias na Educação – CINTED – Universidade Federal do Rio Grande do Sul – UFRGS, Porto Alegre, RS, 2005.

SILVEIRA, Ricardo. Material de aula. Introdução a Sistemas Multiagente - Modelo de Referência FIPA. Disponível em: <http://www.inf.ufsc.br/~silveira/INE602200/Laminas/602200Aula3.pdf>. Acessado em: out de 2012.

SISTEMA E-MEC. Disponível em: <http://emec.mec.gov.br/>. Acesso em: out. 2011.

SPASS: An Automated Theorem Prover for First-Order Logic with Equality. Disponível em: <http://www.spass-prover.org/>. Acesso em: Ago 2012.

SPIEGEL, Murray R. Probabilidade e Estatística. Coleção Schaum. Editora McGRAW – Hill, 1978.

TASSONI, Elvira C M. AFETIVIDADE E APRENDIZAGEM: A RELAÇÃO PROFESSOR-ALUNO. Disponível em: <http://www.anped.org.br/reunioes/23/textos/2019t.PDF>. Acesso em jan. 2013.

The E Theorem Prover. Disponível em: <http://www4.informatik.tu->

muenchen.de/~schulz/E/E.html>. Acesso em: Ago 2012.

UML. Documents Associated With UML Version 2.1.2. Disponível em: <<http://www.omg.org/spec/UML/2.1.2/>>. Acesso em: Maio 2012.

VASCONCELLOS, Celso dos S. Metodologia Dialética em Sala de Aula. In: Revista de Educação AEC. Brasília: abril de 1992 (n. 83).

VICCARI, R.; GLUZ, J. Infraestrutura OBAA-MILOS: Infraestrutura Multiagente para Suporte a Objetos de Aprendizagem OBAA – Rel. Técnico – Especificação Detalhada da Arquitetura da Infraestrutura MILOS. UFRGS/CINTED, 2011.

VICCARI, R.; GLUZ, J. PENTEADO, F. Infraestrutura OBAA-MILOS: Infraestrutura Multiagente para Suporte a Objetos de Aprendizagem OBAA – Rel. Técnico – Especificação Preliminar da Arquitetura do Sistema de Apoio Pedagógico. UFRGS/CINTED, 2011.

VICCARI, R.; GLUZ, J.; PASSERINO, L.; et al. The OBAA Proposal for Learning Objects Supported by Agents. Procs. Of MASEIE Workshop – AAMAS 2010, Toronto, Canada, 2010.

VICCARI, R.; GLUZ, J.; SANTOS, E.; et al. Projeto OBAA – Rel. Técnico RTOBAA01 – Proposta de Padrão para Metadados de Objetos de Aprendizagem Multiplataforma. UFRGS/CINTED, 2009. Disponível em: <<http://www.portalobaa.org/obaac/padraoobaa/relatoriostecnicos/RTOBAA01.pdf/view>>.

W3C. RIF RDF e OWL Compatibilidade. Disponível em: <<http://www.w3.org/TR/2010/REC-rif-rdf-owl-20100622/>>. Acesso em dez. 2010.

W3C. Vocabularies. Disponível em: <<http://www.w3.org/standards/semanticweb/ontology>>. Acesso em jan. 2011.

W3C. World Wide Web Consortium (W3C). Disponível em: <<http://www.w3.org/>>. Acesso em: out. 2010.

WILEY, D. Connecting Learning Objects to Instructional Design Theory: A Definition, A Metaphor, and A Taxonomy. 2001. In Wiley, D. The Instructional Use of Learning Objects: Online Version. Disponível em: <<http://reusability.org/read/chapters/wiley.doc>>. Acesso em: set. 2010.

WOOLDRIDGE, M. An Introduction to Multiagent Systems. John Wiley & Sons, 2002.

XAVIER, A.C. AutoEduMat: Ferramenta de Apoio a Autoria de Metadados de Objetos de Aprendizagem para o Domínio de Ensino de Matemática. Dissertação de Mestrado. Universidade do Vale do Rio dos Sinos, 2010.

ANEXO I

- Formulário para avaliação do Sistema Heráclito:



Avaliação inicial do Sistema de edição de provas e demonstrações - Sistema HERÁCLITO

Esta avaliação inicial tem como finalidade verificar se os objetivos propostos ao desenvolver a ferramenta, foram parcialmente atingidos e fornecer dados para eventual replanejamento.

OBRIGADA!

Equipe do projeto. OBAAMILOS
*Obrigatório

Em qual instituição de ensino você estuda? *

Unisinos
 UFRGS
 Outra instituição

Qual instituição?

Por favor digite o seu número de matrícula: *

1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO *

O sistema Heráclito faz uso de algumas estratégias de aprendizagem. Você acredita que estas estratégias contribuíram no aprendizado?

	Concordo	Concordo parcialmente	Sou Indiferente	Discordo parcialmente	Discordo
1.1 - Como aluno você acredita que o Heráclito de uma forma geral tenha facilitado o aprendizado?	<input type="radio"/>				
1.2 - Os serviços de tutoria oferecidos pelo Heráclito contribuíram para uma melhor compreensão do desenvolvimento dos exercícios?	<input type="radio"/>				
1.3 - Você acredita que a estratégia de sugestão de exemplos na inserção de hipóteses incorretas ajuda no processo de aprendizado?	<input type="radio"/>				
1.4 - Você acredita que a estratégia de sugestão de regras na inserção de regras incorretas ajuda no processo de aprendizado?	<input type="radio"/>				
1.5 - Você acredita que a estratégia de sugestão de ajuda ao aluno quando este se encontra "ocioso" ajuda no processo de aprendizado?	<input type="radio"/>				
1.6 - Você acredita que a estratégia que avisa ao aluno quando este aplica uma regra prejudicial (sem volta) ajuda no processo de aprendizado?	<input type="radio"/>				
1.7 - Você acredita que a estratégia que o tutor utiliza quando identifica que a metade da prova já foi realizada, enviando uma mensagem de reforço ao aluno pode ajudar no processo de aprendizado?	<input type="radio"/>				
1.8 - Você acredita que a estratégia que o tutor utiliza quando identifica que faltam 1 ou 2 passos para finalizar a prova, enviando uma mensagem de reforço ao aluno, pode ajudar no processo de aprendizado?	<input type="radio"/>				

2.1 - USO DO TUTOR NO APRENDIZADO *

Quantas questões você respondeu COM a ajuda do tutor?

2.2 - USO DO TUTOR NO APRENDIZADO *

Quantas questões você respondeu SEM a ajuda do tutor?

3 - QUALIDADE DO CONTEÚDO *

	Concordo	Concordo parcialmente	Sou Indiferente	Discordo parcialmente	Discordo
3.1.- O conteúdo da lista de exercícios foi facilmente entendido e realizado no Heráclito?	<input type="radio"/>				
3.2.- O nível de dificuldade da lista de exercícios foi distribuída de forma proporcional com a relação ao conteúdo?	<input type="radio"/>				
3.3.- A lista de exercícios cobre boa parte do conteúdo de dedução natural?	<input type="radio"/>				
3.4.- Os exemplos ajudaram SIM no processo de resolução do exercício.	<input type="radio"/>				
3.5.- Os exemplos NÃO ajudaram no processo de resolução do exercício.	<input type="radio"/>				

QUESTÕES OPCIONAIS

QUESTÕES 4 E 5 SÃO OPCIONAIS

4 - DESIGN/LAYOUT VISUAL

	Concordo	Concordo parcialmente	Sou Indiferente	Discordo parcialmente	Discordo
4.1.- O Heráclito possui um layout adequado ao aprendizado?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.2.- Na tela de Edição do objeto, os botões de regras de inferência são intuitivos e as informações claras?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.3.- Na tela de Edição do objeto, o espaço para demonstração da prova é suficiente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5 - USABILIDADE

	Concordo	Concordo parcialmente	Sou Indiferente	Discordo parcialmente	Discordo
5.1.- NÃO consegui usar o objeto de aprendizagem Heráclito	<input type="radio"/>				
5.2.- Quando estou utilizando o conteúdo para aprendizado não consigo entender o propósito da ferramenta	<input type="radio"/>				
5.3.- Os menus da página principal do Heráclito funcionam?	<input type="radio"/>				
5.4.- Os botões da página principal do Heráclito funcionam?	<input type="radio"/>				
5.5.- Na página de Edição do objeto, as janelas de dialogo que são abertas em algumas das regras são fáceis de serem compreendidas/usadas?	<input type="radio"/>				
5.6.- Na página de Edição do objeto, a sistematica de clicar na regra e depois nas linhas de demonstração é de fácil compreensão/utilização?	<input type="radio"/>				
5.7.- O manual do Heráclito contempla as possíveis dúvidas do usuário?	<input type="radio"/>				

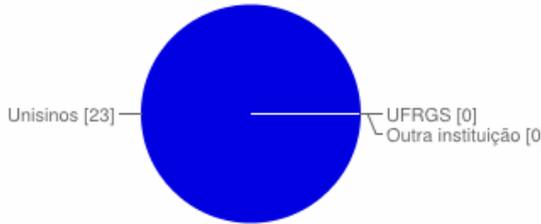
INFORMAÇÕES ADICIONAIS *

Deixe registrado aqui as suas dúvidas, sugestões, elogios, críticas, etc, etc... com relação a utilização do HERÁCLITO. Sua opinião é muito importante para o desenvolvimento e constante melhoramento desta ferramenta (esta questão é obrigatória para envio do formulário!)

ANEXO II

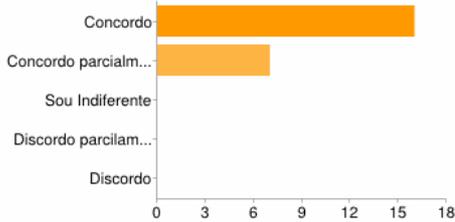
- Resultados do questionário de avaliação do Sistema Heráclito, contêm todas as respostas enviadas pelos alunos:

Em qual instituição de ensino você estuda?



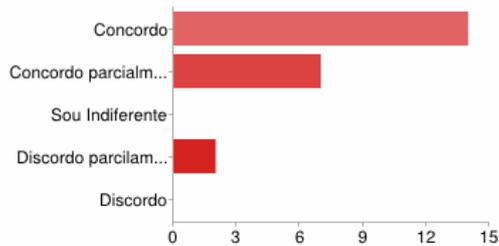
Unisinos	23	100%
UFRGS	0	0%
Outra instituição	0	0%

1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.1 - Como aluno você acredita que o Heráclito de uma forma geral tenha facilitado o aprendizado?



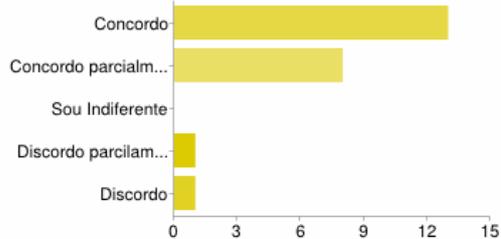
Concordo	16	70%
Concordo parcialmente	7	30%
Sou Indiferente	0	0%
Discordo parcialmente	0	0%
Discordo	0	0%

1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.2 - Os serviços de tutoria oferecido pelo Heráclito contribuíram para uma melhor compreensão do desenvolvimento dos exercícios?



Concordo	14	61%
Concordo parcialmente	7	30%
Sou Indiferente	0	0%
Discordo parcialmente	2	9%
Discordo	0	0%

1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.3 - Você acredita que a estratégia de sugestão de exemplos na inserção de hipóteses incorretas ajuda no processo de aprendizado?

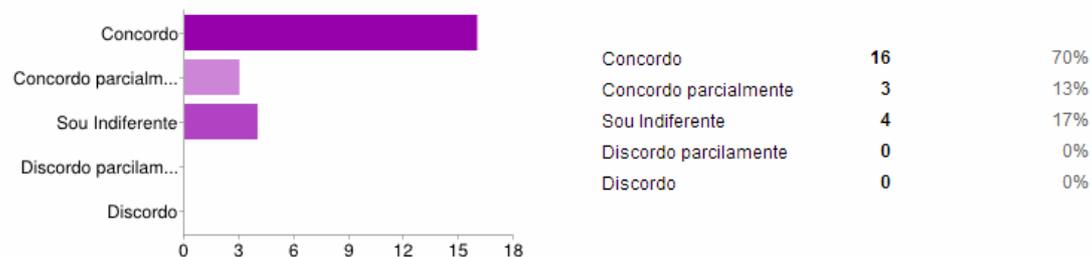


Concordo	13	57%
Concordo parcialmente	8	35%
Sou Indiferente	0	0%
Discordo parcialmente	1	4%
Discordo	1	4%

1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.4 - Você acredita que a estratégia de sugestão de regras na inserção de regras incorretas ajuda no processo de aprendizado?



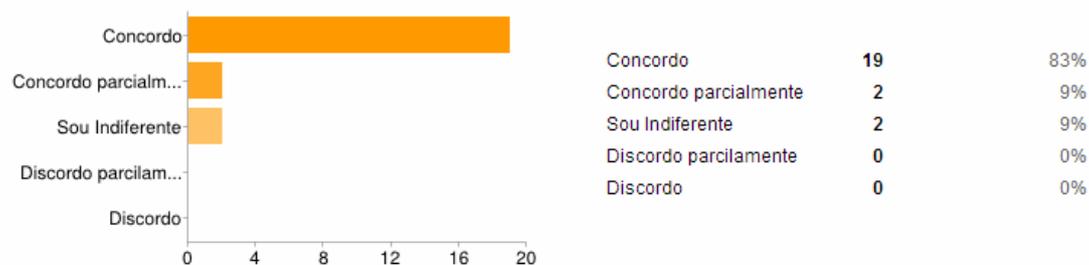
1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.5 - Você acredita que a estratégia de sugestão de ajuda ao aluno quando este se encontra "ocioso" ajuda no processo de aprendizado?



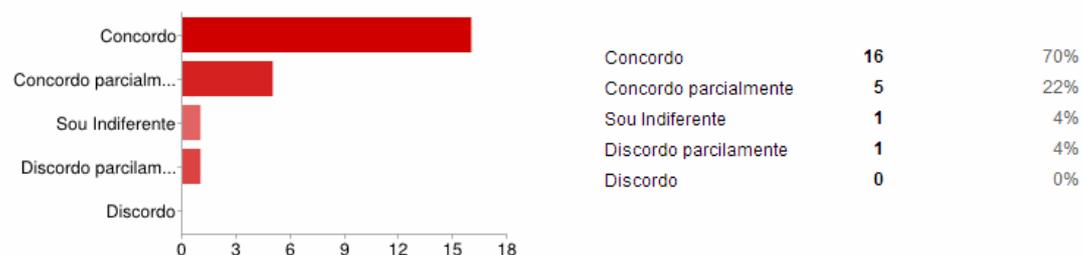
1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.6 - Você acredita que a estratégia que avisa ao aluno quando este aplica uma regra prejudicial (sem volta) ajuda no processo de aprendizado?



1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.7 - Você acredita que a estratégia que o tutor utiliza quando identifica que a metade da prova já foi realizada, enviando uma mensagem de reforço ao aluno pode ajudar no processo de aprendizado?



1 - CONTRIBUIÇÃO DO TUTOR NO APRENDIZADO - 1.8 - Você acredita que a estratégia que o tutor utiliza quando identifica que faltam 1 ou 2 passos para finalizar a prova, enviando uma mensagem de reforço ao aluno, pode ajudar no processo de aprendizado?

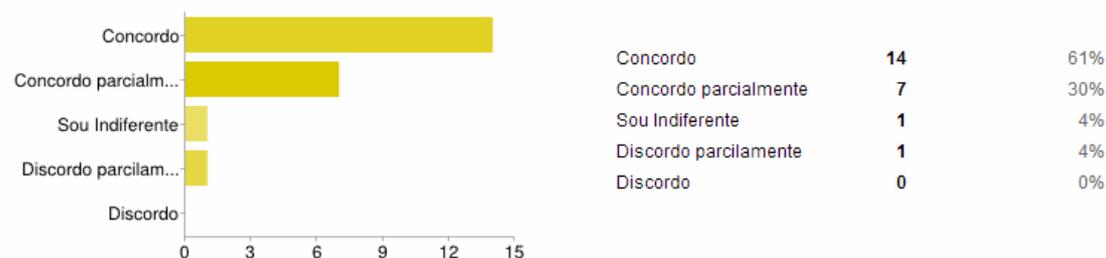


USO DO TUTOR NO APRENDIZADO

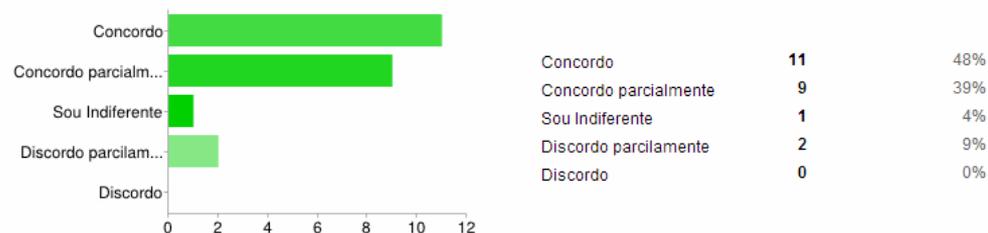
COM AJUDA DO TUTOR: 70

SEM AJUDA DO TUTOR: 164

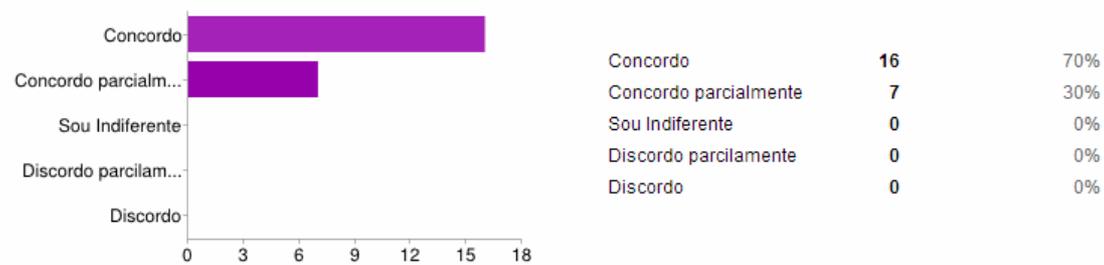
3 - QUALIDADE DO CONTEÚDO - 3.1.- O conteúdo da lista de exercícios foi facilmente entendido e realizado no Heráclito?



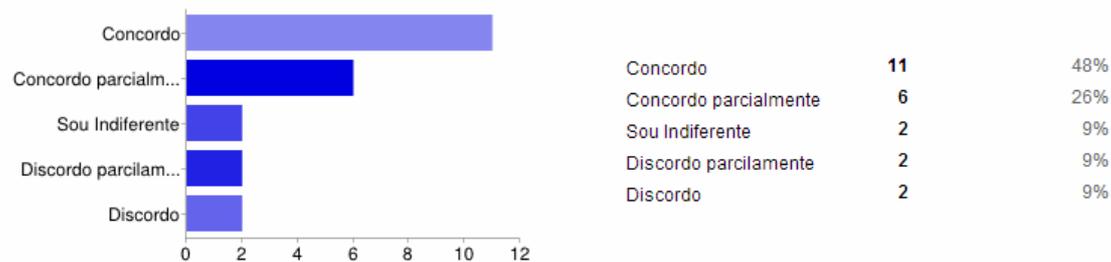
3 - QUALIDADE DO CONTEÚDO - 3.2.- O nível de dificuldade da lista de exercícios foi distribuída de forma proporcional com a relação ao conteúdo?



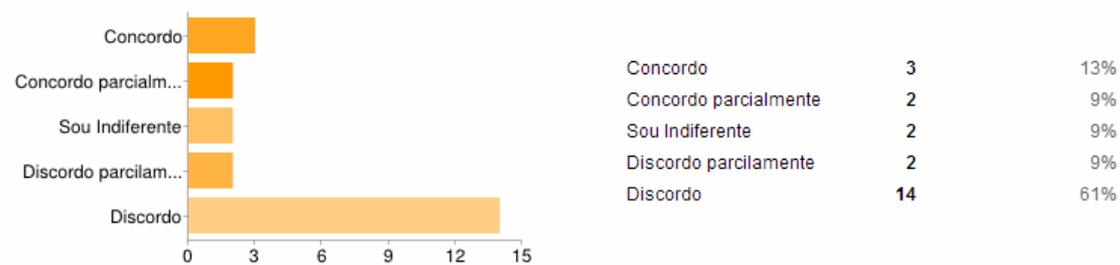
3 - QUALIDADE DO CONTEÚDO - 3.3.- A lista de exercícios cobre boa parte do conteúdo de dedução natural?



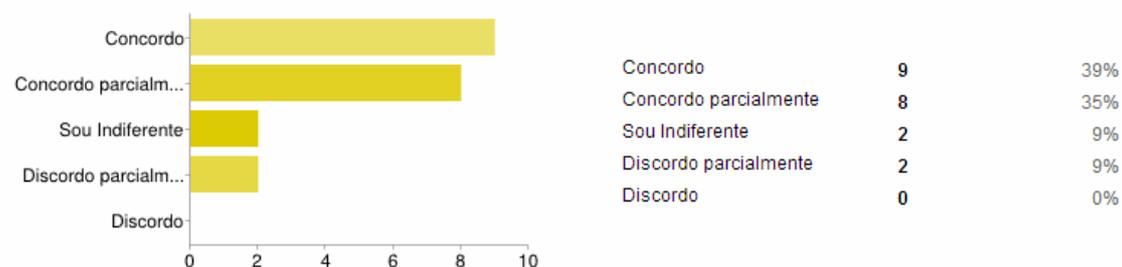
3 - QUALIDADE DO CONTEÚDO - 3.4.- Os exemplos ajudaram SIM no processo de resolução do exercício.



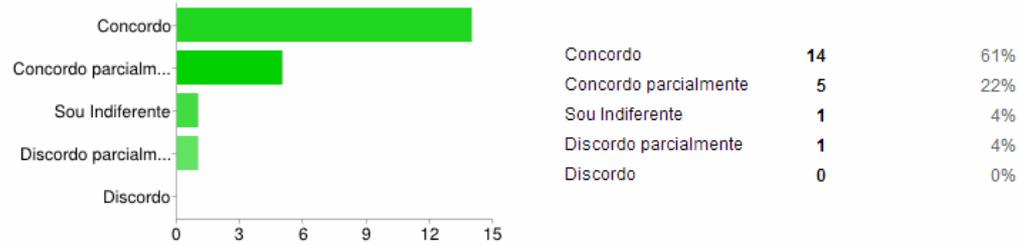
3 - QUALIDADE DO CONTEÚDO - 3.5.- Os exemplos NÃO ajudaram no processo de resolução do exercício.



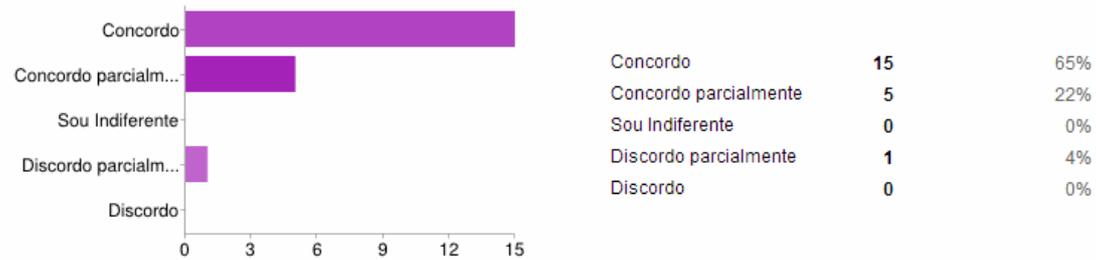
4 - DESIGN/LAYOUT VISUAL - 4.1.- O Heráclito possui um layout adequado ao aprendizado?



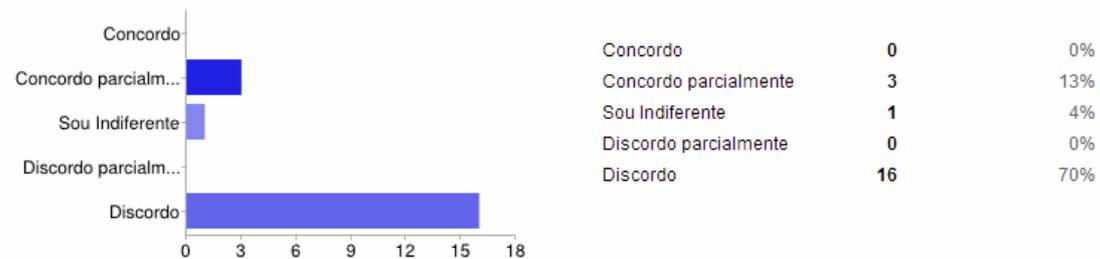
4 - DESIGN/LAYOUT VISUAL - 4.2.- Na tela de Edição do objeto, os botões de regras de inferência são intuitivos e as informações claras?



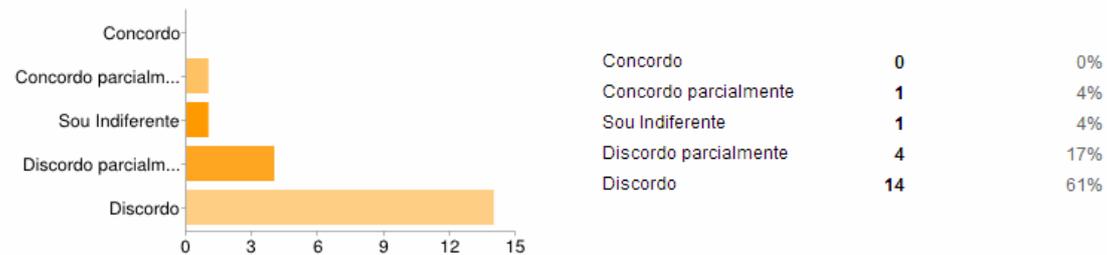
4 - DESIGN/LAYOUT VISUAL - 4.3.- Na tela de Edição do objeto, o espaço para demonstração da prova é suficiente?



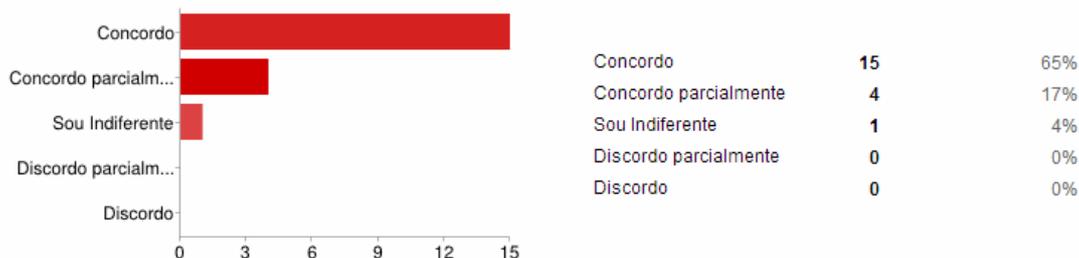
5 - USABILIDADE - 5.1.- NÃO consegui usar o objeto de aprendizagem Heráclito



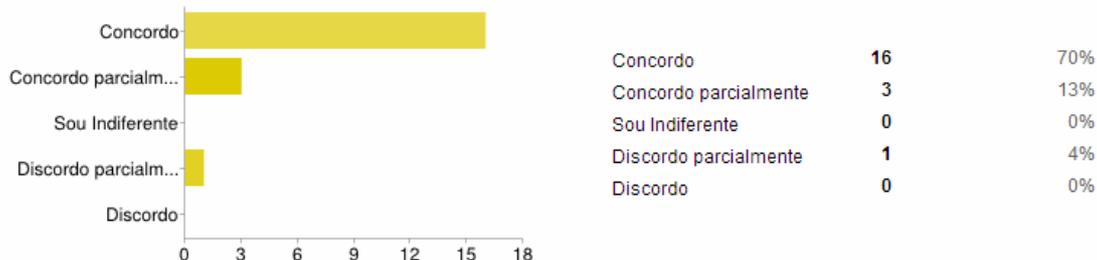
5 - USABILIDADE - 5.2.- Quando estou utilizando o conteúdo para aprendizado não consigo entender o propósito da ferramenta



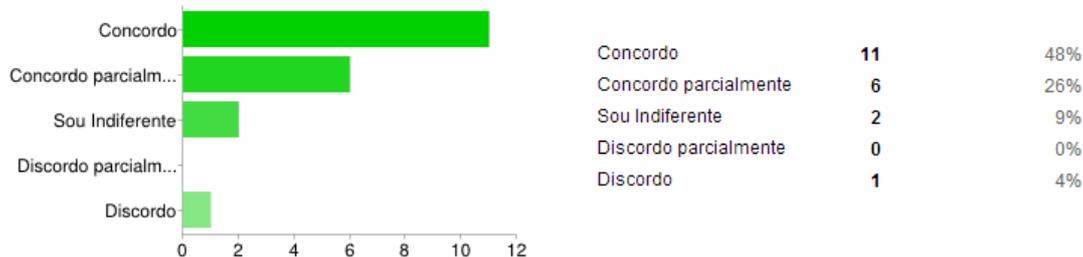
5 - USABILIDADE - 5.3.- Os menus da página principal do Heráclito funcionam?



5 - USABILIDADE - 5.4.- Os botões da página principal do Heráclito funcionam?



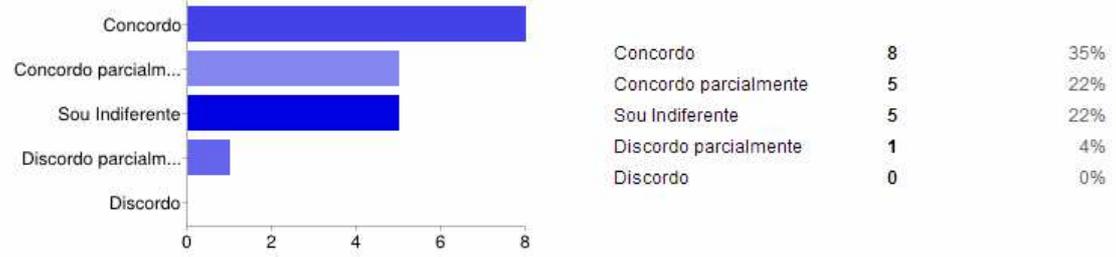
5 - USABILIDADE - 5.5.- Na página de Edição do objeto, as janelas de diálogo que são abertas em algumas das regras são fáceis de serem compreendidas/usadas?



5 - USABILIDADE - 5.6.- Na página de Edição do objeto, a sistemática de clicar na regra e depois nas linhas de demonstração é de fácil compreensão/utilização?



5 - USABILIDADE - 5.7.- O manual do Heráclito contempla as possíveis dúvidas do usuário?



ANEXO III

- Anais do XXII SBIE - XVII WIE Aracaju, 21 a 25 de novembro de 2011. Suporte Técnico/Pedagógico aos OAs pelos AVAs: uma Análise Comparativa das Pesquisas Recentes.

ANEXO IV

- Anais do XXII SBIE - XVII WIE Aracaju, 21 a 25 de novembro de 2011. Sistema Heráclito: Suporte a Objetos de Aprendizagem Interativos e Dialéticos Voltados ao Ensino de Dedução Natural na Lógica Proposicional.