



**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



ARCABOUÇO CONCEITUAL DE ADAPTAÇÃO DE RECURSOS EDUCACIONAIS

VITOR BREMGARTNER DA FROTA

**Orientador: José Francisco de Magalhães Netto
Coorientador: Crediné Silva de Menezes (Universidade Federal do
Rio Grande do Sul)**

Manaus – Amazonas
Novembro – 2017

**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

VITOR BREMGARTNER DA FROTA

**ARCABOUÇO CONCEITUAL DE ADAPTAÇÃO DE
RECURSOS EDUCACIONAIS**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Doutor em Informática, na área de concentração em Informática na Educação

Orientador: José Francisco de Magalhães Netto
Coorientador: Crediné Silva de Menezes (Universidade Federal do Rio Grande do Sul)

Manaus – Amazonas
Novembro – 2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

F941a Frota, Vitor Bremgartner da
Arcabouço Conceitual de Adaptação de Recursos Educacionais /
Vitor Bremgartner da Frota. 2017
219 f.: il. color; 31 cm.

Orientador: José Francisco de Magalhães Netto
Coorientador: Crediné Silva de Menezes
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Ambientes Virtuais de Aprendizagem. 2. adaptação de recursos. 3. agentes de software. 4. modelo aberto de aluno. 5. Smart Learning Environments. I. Netto, José Francisco de Magalhães II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO



UFAM

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

FOLHA DE APROVAÇÃO

"Arcabouço Conceitual de Adaptação de Recursos Educacionais"

VITOR BREMGARTNER DA FROTA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Jose Francisco de Magalhães Netto

Prof. José Francisco de Magalhães Netto - PRESIDENTE

Alberto Nogueira de Castro Júnior

Prof. Alberto Nogueira de Castro Júnior - MEMBRO INTERNO

Jose Luiz de Souza Pio

Prof. José Luiz de Souza Pio - MEMBRO EXTERNO

Orivaldo de Lira Tavares

Prof. Orivaldo de Lira Tavares - MEMBRO EXTERNO

João Marcos Bastos Cavalcanti

Prof. João Marcos Bastos Cavalcanti - MEMBRO EXTERNO

Manaus, 24 de Novembro de 2017

Até aqui nos ajudou o SENHOR.

I Samuel, 7:12b

O SENHOR é meu Pastor, nada me faltará.

Salmos, 23:1

A Deus, o Senhor de todas as boas obras.

AGRADECIMENTOS

Antes de tudo agradeço a Deus, porque Ele é quem me ajudou nos momentos mais difíceis, não só do Doutorado, mas de minha vida, e que me amou antes mesmo de eu ter fé nele, sempre me abençoando e sendo provedor e construtor dos meus conhecimentos.

Queria agradecer aos meus pais que sempre me deram as condições ideais para que eu me dedicasse aos estudos, além de terem me dado o exemplo de caráter a ser seguido. Obrigado dona Raquel Bremgartner e seu Jocílio Frota! Obrigado a meus irmãos por todo o carinho e confiança em mim. São eles Rebeca, Gabriel e Davi.

Queria agradecer ao meu orientador, professor Dr. José Netto, que sempre acreditou em mim, fornecendo importantes orientações para toda a execução deste projeto e pela ideia que permeia este trabalho. Também ao meu coorientador, professor Dr. Crediné Menezes, sempre fornecendo excelentes dicas de como utilizar as tecnologias educacionais numa abordagem pedagógica.

Quero agradecer também a minha princesa Jackeline Lira, que deu muita força e companhia nessa caminhada do Doutorado para que eu pudesse concluí-lo com êxito.

Obrigado ainda aos colegas que tiveram papéis importantes de alguma forma no decorrer do Doutorado. Além disso, a lista é muito maior daqueles que caminham e caminharam comigo, que tenho medo de esquecer algum na minha fraqueza de memória, mas não me interpretem mal.

Agradeço também à CAPES pela bolsa de estudos, que possibilitou uma dedicação integral a este curso e também à Universidade Federal do Amazonas (UFAM), fornecendo um ótimo ambiente para a realização deste projeto. Agradeço à escola que ministrou aulas, o Instituto Federal do Amazonas (IFAM), que tem apoiado minhas pesquisas, como passagens para congressos.

Para finalizar, agradeço a qualquer pessoa que colaborou direta ou indiretamente com a execução deste trabalho.

Obrigado a todos!

RESUMO

Normalmente, os recursos existentes em Ambientes Virtuais de Aprendizagem (AVAs), utilizados em cursos de Educação a Distância (EaD) e em semipresenciais, são apresentados da mesma forma para todos os alunos. Isto pode não ser útil para a aprendizagem efetiva de cada aluno. A abordagem adotada para solução deste problema nesta tese baseia-se em um framework chamado ArCARE (Arcabouço Conceitual de Adaptação de Recursos Educacionais), sendo uma estratégia que permite a criação de *Smart Learning Environments* (SLEs) e adaptação de recursos para alunos em AVAs no decorrer do curso, com base no Construtivismo de Piaget, utilizando a tecnologia de sistema multiagente que manipula uma ontologia de modelo aberto de aluno composto de várias características do aluno de acordo com suas interações com o AVA e seu engajamento no curso, tais como competências, habilidades, desempenho nas atividades, frequência, preferências e estilos de aprendizagem. Estes recursos adaptados no AVA podem ser desde uma recomendação de colegas para ajudar alunos com dificuldades em atividades, como também a recomendação de propostas de atividades colaborativas com o objetivo de o aluno ter uma aprendizagem mais efetiva do conteúdo de um determinado curso. A solução descrita nesta tese também se baseia na aprendizagem colaborativa pela interação entre aprendizes, na qual é um método empregado com o propósito de construir seu conhecimento de maneira mais significativa em um AVA. Além disso, o modelo de cada aluno é alterado dinamicamente durante o curso, por meio das interações do aluno com o AVA. Este modelo é apresentado para o aluno, utilizando-se Learning Analytics (LA), com o objetivo do aluno conhecer seu estado atual no curso, permitindo uma aprendizagem auto-regulada. Um dos resultados gerados no ArCARE foi o sistema AMPARA (do inglês, *Adaptive Multi-agent Pedagogical Architectures for Resources and Activities in VLEs*, que em português fica Arquiteturas Pedagógicas Multiagente Adaptativas para Recursos e Atividades em AVAs), aplicado em disciplinas com currículos tradicionais como Cálculo Numérico e Algoritmos e Programação, na modalidade semipresencial para recomendar APs. Outro resultado foi o sistema AMPARAX (eXtended AMPARA), que foi aplicado em um curso com currículo flexível, de Pensamento Computacional para Programação de Computadores no AVA Moodle, mostrando, assim, as provas de conceito do ArCARE.

Palavras-Chave: Ambientes Virtuais de Aprendizagem; adaptação de recursos; agentes de software; modelo aberto de aluno; Smart Learning Environments.

ABSTRACT

Usually, the resources available in Virtual Learning Environments (VLEs), used in distance and blended courses, are presented in the same way for all students. This may not be useful for the effective learning of each student. The approach adopted to solve this problem in this thesis is based on a framework called ArCARE (Conceptual Framework of Adaptation of Educational Resources), being a strategy that allows the development of Smart Learning Environments (SLEs) and adaptation of resources to students in VLEs during the course, based on the Piaget Constructivism, using multi-agent system technology that manipulates an open learner model ontology composed of several characteristics of the student according to their interactions with the VLE and their engagement in the course such as competences, skills, performance in activities, frequency, preferences, and learning styles. These adapted resources in the VLE can be a recommendation of colleagues to help students with difficulties in activities, as well as the recommendation of proposed collaborative activities with the aim of the student to have a more effective learning of the course contents. The solution described in this thesis is also based on collaborative learning by the interaction between learners, in which it is a method used for the purpose of construct his knowledge in a more meaningful way in VLEs. In addition, each learner model is changed dynamically during the course, through the student's interactions with the VLE. This model is presented to the student using Learning Analytics (LA), with the objective of the student to know their current state in the course, allowing a self-regulated learning. One of the results generated in ArCARE was the AMPARA system (Adaptive Multi-agent Pedagogical Architectures for Resources and Activities in VLEs), applied in disciplines with traditional curricula such as Numerical Calculus and Algorithms and Programming in the blended mode in order to recommend PAs. Another result was the AMPARAX system (eXtended AMPARA), which was applied in a course with flexible curriculum, Computational Thinking for Computer Programming using Moodle VLE, showing, therefore, ArCARE's proofs of concept.

Keywords: Virtual Learning Environments; adaptation of resources; software agents; open learner model; Smart Learning Environments.

LISTA DE SIGLAS

ACL	Agent Communication Language
AID	Agent Identifier
AMPARA	Adaptive Multi-agent Pedagogical Architectures for Resources and Activities
AMPARAX	eXtended AMPARA
AP	Arquitetura Pedagógica
ArCARE	Arcabouço Conceitual de Adaptação de Recursos Educacionais
AVA	Ambiente Virtual de Aprendizagem
BD	Banco de Dados
BDI	Beliefs-Desires-Intentions
CAI	Instrução Assistida pelo Computador
CSCL	Computer-Supported-Collaborative-Learning
DF	Directory Facilitator
DTD	Document Type Definition
EaD	Educação a Distância
FIPA	Foundation for Intelligent Physical Agents
IA	Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IFAM	Instituto Federal do Amazonas
IMS	Instructional Management Systems
ISO	International Organization for Standardization
JADE	Java Agent Development Framework
JADEX	JADE eXtension
LA	Learning Analytics
LIP	Learning Information Package
LOM	Learning Object Metadata
MAA	Modelo Aberto de Aluno
MOOCs	Massive Open Online Courses
Moodle	Modular Object-Oriented Dynamic Learning Environment
MySQL	My Structured Query Language
PAPI	Personal and Private Information
PC	Pensamento Computacional

PHP	PHP: Hypertext Preprocessor
QCL	Qualifications, Certificates and Licenses
RDCEO	Reusable Definition of Competency or Educational Objective
RMI	Remote Method Invocation
SLE	Smart Learning Environment
SMA	Sistema Multiagente
STI	Sistema Tutor Inteligente
TIC	Tecnologia da Informação e Comunicação
UE	Unidade de Ensino
XML	Extensible Markup Language

LISTA DE TABELAS

Tabela 1: As características de aprendizagem.	36
Tabela 2. Aspectos Destacados nos Trabalhos e Propostas Visitados.	68
Tabela 3. Resumo dos Cenários do Capítulo 1.	70
Tabela 4. Comparação entre a Situação Atual dos Cenários e a Situação Proposta no ArCARE.	74
Tabela 5. Exemplo de critério de incremento/decremento de níveis de habilidades em alunos.	84
Tabela 6. Elementos utilizados em cada categoria do padrão LIP.	100
Tabela 7. Propriedades utilizadas em cada tipo de agente.	101
Tabela 8. Relações entre as classes da ontologia de MAA.	101
Tabela 9. Resumo dos Aspectos Destacados no ArCARE comparados com os trabalhos e propostas visitados.	107
Tabela 10. Conceitos de PC e suas capacidades principais na Ciência da Computação.	128
Tabela 11. Expansão da Ontologia de MAA para representação dos estados mentais dos Agentes de Adaptação de Recursos no JADEX.	150
Tabela 12. Atributos dos estudantes usados no algoritmo Apriori.	162
Tabela 13. Notas dos alunos obtidas no pré-teste e no teste final.	166
Tabela 14. Respostas dos estudantes no questionário.	170
Tabela 15. Níveis de habilidades dos estudantes após o pré-teste.	170
Tabela 16. Níveis de habilidades dos estudantes após o pós-teste.	172
Tabela 17. Resultados dos testes de normalidade e teste t aplicados aos níveis de habilidades dos alunos.	174
Tabela 18: Objetivo da Revisão Sistemática.	205
Tabela 19: Resultados gerais da quantidade de publicações encontradas.	208
Tabela 20: Técnicas de adaptação encontradas nas publicações selecionadas.	209
Tabela 21: Técnicas de adaptação encontradas em todas as publicações selecionadas que usam uma abordagem construtivista.	213

LISTA DE FIGURAS

Figura 1. Arquitetura clássica de um STI.....	48
Figura 2. Categorias do IMS LIP.	52
Figura 3. Diagrama de alto nível do i-Collaboration.	55
Figura 4. Modelo de adaptação baseado em estilos de aprendizagem e agentes.	56
Figura 5. Estrutura do Oscar CITS.	57
Figura 6. a) Principais módulos do COMTEXT b) Framework usado no COMTEXT.	58
Figura 7. Arquitetura do mecanismo de adaptação integrado a um LMS.....	59
Figura 8. Exemplos de representação de MAA, com (a) visões simples e (b) estruturada.	60
Figura 9. Página inicial do framework.	61
Figura 10. Modelo de Desenvolvimento de Processo de Aprendizagem Baseado no Construtivismo Centrado no Aluno.....	62
Figura 11. Estrutura de site.....	63
Figura 12. Projeto conceitual do sistema de e-learning ubíquo.....	64
Figura 13. Arquitetura do PULP e seu processo de personalização.....	65
Figura 14. Uma representação Model-View-Controller da plataforma MorFEu..	66
Figura 15. A apresentação de recomendações no contexto da interface OSSM das Grades de Domínio; uma célula com um símbolo de estrela representa um item recomendado.	67
Figura 16. Modelo de Referência do ArCARE.....	75
Figura 17. Arquitetura de referência do ArCARE.....	77
Figura 18. Representação gráfica: cada competência C_i tem h_{im} habilidades. ..	81
Figura 19. Plataforma de Agentes do Modelo de Aluno.	82
Figura 20. Verificação dos erros e acertos dos estudantes em uma atividade e suas habilidades associadas.	86
Figura 21. Plataforma de Agentes do Modelo de Aluno.	88
Figura 22. Atualização das habilidades do estudante em função de resultados de atividades e recomendação de recursos em função das habilidades.	91
Figura 23. Recomendação de recursos ao estudante em função de seus estilos de aprendizagem.....	92
Figura 24. Algoritmo de recomendação de recursos simples.....	93
Figura 25. Obtenção do histórico e utilização do AVA pelos Agentes de Modelo do Aluno.....	95
Figura 26. Trecho de algoritmo de adaptação de recursos aprimorado.	96
Figura 27. Representação gráfica resumida da Ontologia de MAA usada pelo ArCARE.....	99
Figura 28. Modelo Conceitual do AMPARA.....	111
Figura 29. Questionário no ColabWeb mostrado a alunos na disciplina Cálculo Numérico.	113
Figura 30. Arquitetura do AMPARA.....	115
Figura 31. Representação gráfica resumida da Ontologia de MAA usada pelo AMPARA.....	117
Figura 32. Apresentando o Modelo de Aluno Aberto a estudantes.....	119

Figura 33. Troca de mensagens entre agentes de modelo de aluno desenvolvidos no JADE.	120
Figura 34. Mensagem enviada pelo <i>Help Tutor Agent</i> ao aluno.....	121
Figura 35. Estudantes recomendados.....	121
Figura 36. Exemplos de atividades recomendadas a alunos de Cálculo Numérico.	122
Figura 37. Exemplo de atividade passada como Laboratório de Avaliação.....	122
Figura 38. Exemplo da atividade debate de teses recomendada e sua avaliação por pares.....	123
Figura 39. Interface do Minicurso de Pensamento Computacional no ColabWeb.	130
Figura 40. Uma questão de pré-teste.....	131
Figura 41. Disposição das UEs optativas no minicurso de PC.....	133
Figura 42. Uma atividade no Code.org (Labirinto).....	134
Figura 43. Disposição das APs Recomendadas no minicurso de PC.	135
Figura 44. Exemplo de uma atividade baseada em APs com suas fases.	139
Figura 45. Exemplo de avaliação a um debate de teses recomendado para estudantes.....	141
Figura 46. Estrutura interna do núcleo de um Agente de Adaptação de Recursos e seus estados mentais utilizado no curso de PC.	142
Figura 47. Processamento das percepções.....	147
Figura 48. Atualização das intenções.....	148
Figura 49. Processo deliberativo.....	149
Figura 50. Representação gráfica resumida da ontologia estendida de MAA, considerando os estados mentais.	151
Figura 51. Trecho de código XML do agente <i>Profile Situation Assessment Agent</i>	155
Figura 52. Trecho do Plano (Intenção) <i>AcareManager.java</i> que diz respeito ao DES2 (parte 1 de 3).	156
Figura 53. Trecho do Plano (Intenção) <i>AcareManager.java</i> que diz respeito ao DES2 (parte 2 de 3).	157
Figura 54. Trecho do Plano (Intenção) <i>AcareManager.java</i> que diz respeito ao DES2 (parte 3 de 3).	158
Figura 55. JADEX Control Center com os Agentes de Adaptação de Recursos a serem executados.....	159
Figura 56. Requisitando uma recomendação ao <i>Recommended Resources Agent</i>	160
Figura 57. Requisição atendida.....	161
Figura 58. Troca de mensagens entre os agentes de adaptação e recomendação de recursos feitos no JADEX.....	163
Figura 59. Página de configurações do professor.....	164
Figura 60. Link para acesso ao MAA e configurações do minicurso pelo estudante.	164
Figura 61. Apresentando o MAA aos estudantes.....	165
Figura 62. Opções para o estudante escolher que adaptações de recursos gostaria de ter no ColabWeb.	166
Figura 63. Avaliando os recursos adaptados e recomendados por níveis de satisfação.	166
Figura 64. Dispersão das notas obtidas (pré-teste e teste final) pela turma....	169

Figura 65. Dispersão dos níveis de habilidades (pré-teste e teste final) da turma.	175
Figura 66. Percentuais de trabalhos que usam e não usam uma abordagem construtivista.	212

SUMÁRIO

LISTA DE SIGLAS	ix
CAPÍTULO 1 – INTRODUÇÃO.....	19
1.1 Descrição do Problema e uma Proposta de Solução	20
Discussão dos Cenários	25
Estratégia de solução	26
1.2 Motivação e Justificativa.....	27
1.3 Questão de Pesquisa e Hipóteses.....	28
1.4 Objetivos.....	28
1.5 Metodologia Aplicada	29
1.6 Organização do Trabalho	29
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA.....	30
2.1 Abordagem Construtivista para a Aprendizagem	30
2.2 Arquiteturas Pedagógicas	31
2.3 Definindo Competências e Habilidades	32
2.4 Estilos de Aprendizagem do Aluno.....	35
2.4.1 Estilos de Aprendizagem de Felder-Silverman.....	35
2.4.2 Questionário Honey-Alonso de Estilos de Aprendizagem	37
2.5 Sistemas Multiagente.....	40
2.6 Arquitetura BDI.....	41
Raciocínio Prático.....	42
Estados Mentais.....	42
Crença.....	43
Desejo.....	44
Intenção.....	44
2.6.1 Agentes BDI.....	45
Processo Deliberativo	45
Raciocínio Meio-e-Fim	45
Estratégias de Comprometimento.....	45
2.7 Ontologias.....	46
2.8 Sistemas Tutores Inteligentes.....	46
2.9 Smart Learning Environments	48
2.10 O Perfil e o Modelo de Aluno em Ambientes Educacionais.....	49
2.10.1 Modelo de Aluno IMS LIP.....	51

2.11 Conclusões do Capítulo	53
CAPÍTULO 3 – TRABALHOS RELACIONADOS.....	54
3.1 i-Collaboration	54
3.2 Modelo Adaptativo de Ambientes Educacionais Abertos Baseado em Estilos de Aprendizagem, de Fasihuddin.....	55
3.3 Sistema Tutor Inteligente Conversacional Oscar (CITS).....	56
3.4 COMTEXT.....	57
3.5 Mecanismo de Adaptação de Graf e Kinshuk.....	58
3.6 Utilização de Modelo Aberto de Aluno no trabalho de Ahmad	59
3.7 Framework de Martins e Carrapatoso.....	60
3.8 Modelo de Processo de Aprendizagem Baseado no Construtivismo Centrado no Aluno, de Schreurs e Al-Huneidi.....	61
3.9 Aprendizagem de Programação Apoiada por Computador, de Menezes.....	62
3.10 Ambiente Ubíquo de e-Learning de Shaoling e Fangfang	63
3.11 PULP.....	64
3.12 MorFEu	65
3.13 Modelo Aberto Social de Aluno (OSSM) de Hosseini, Hsiao, Guerra e Brusilovsky 66	
3.14 Conclusões do Capítulo	67
CAPÍTULO 4 – O ARCABOUÇO CONCEITUAL DE ADAPTAÇÃO DE RECURSOS EDUCACIONAIS (ArCARE)	70
4.1 Análise dos Cenários	70
4.2 Caracterização do ArCARE	74
4.3 A Definição do ArCARE	76
4.3.1 Caracterizando as Unidades de Ensino no ArCARE	79
4.4 O Módulo de Tecnologias Inteligentes no ArCARE	80
4.4.1 Papéis dos Agentes de Modelo de Aluno	81
4.4.2 Papéis dos Agentes de Adaptação de Recursos.....	88
4.4.3 Funcionamento dos Agentes para a Adaptação de Recursos	90
4.5 A Ontologia de Modelo Aberto de Aluno	97
4.6 Comparação do ArCARE com Outras Abordagens.....	107
4.7 Conclusões do Capítulo.....	108
CAPÍTULO 5 – SISTEMA AMPARA DE RECOMENDAÇÃO DE RECURSOS	110
5.1 Modelo Conceitual, Arquitetura e Implementação do AMPARA.....	110
5.2 Técnicas de Adaptação Testadas	119
5.3 Limitações do AMPARA.....	124
5.4 Conclusões do Capítulo	124
CAPÍTULO 6 – SISTEMA EXTENDED AMPARA (AMPARAX)	126

6.1	Princípios Educacionais do Curso de Pensamento Computacional	127
6.2	Estrutura do Curso	129
6.3	Execução do Curso e Resultados Obtidos	137
	6.3.1 Modelo BDI Aplicado aos Agentes de Adaptação de Recursos	141
	Estados Mentais	142
	Crenças (Beliefs)	143
	Desejos (Desires/Goals).....	143
	Intenções (Intentions/Plans)	145
	Processos	146
	Processamento da Percepção.....	146
	Atualização das Intenções.....	147
	Deliberação.....	148
	Seleção de Desejos Candidatos.....	148
	Filtragem.....	149
	6.3.2 A Ontologia de MAA Estendida com BDI	150
	6.3.3 Desejos e Execução dos Agentes de Adaptação de Recursos.....	152
	6.3.4 Algoritmos de Mineração de Dados Usados no AMPARAX.....	161
	6.3.5 Resultados Obtidos com o Uso do AMPARAX	163
6.4	Conclusões do Capítulo	175
	CAPÍTULO 7 – CONSIDERAÇÕES FINAIS.....	177
7.1	Contribuições deste Trabalho	178
7.2	Trabalhos Futuros.....	180
7.3	Publicações	181
	REFERÊNCIAS BIBLIOGRÁFICAS	183
	APÊNDICE A – ONTOLOGIA DE MODELO ABERTO DE ALUNO	192
	APÊNDICE B – QUESTIONÁRIO DE PESQUISA DE OPINIÃO I	193
	APÊNDICE C – QUESTÕES DO PRÉ- E PÓS-TESTE.....	194
	APÊNDICE D – QUESTIONÁRIO DE PESQUISA DE OPINIÃO II.....	202
	APÊNDICE E – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	203
	APÊNDICE F – REVISÃO SISTEMÁTICA DE LITERATURA.....	204
F.1	Revisão Sistemática sobre adaptação em AVAs baseada na teoria construtivista..	204
F.1.1	Planejamento da Revisão Sistemática	205
	Objetivo	205
	Formulação das Perguntas	205
	Critérios de Inclusão de Publicações.....	206
	Listagem de Fontes.....	206
	Strings de Busca.....	206

F.1.2	Condução da Revisão Sistemática	207
	Estratégias de Extração de Informação.....	207
F.1.3	Análise e Publicação dos resultados	208
APÊNDICE G	– GLOSSÁRIO	217

CAPÍTULO 1 – INTRODUÇÃO

Neste capítulo serão apresentados o contexto e a descrição do problema, o que motivou esta pesquisa e as questões de investigação. São também apresentados os objetivos e a organização deste texto.

Para apoiar a Educação a Distância (EaD) e cursos semipresenciais são utilizados os Ambientes Virtuais de Aprendizagem (AVAs). Estes ambientes suportam o processo de comunicação entre os alunos, professores, materiais didáticos e a comunidade, fazendo com que todos participem de modo interativo (Kemczinski, 2005). Alguns exemplos de Ambientes Virtuais de Aprendizagem, são WebCT/Blackboard (Blackboard, 2017), Teleduc (Teleduc, 2017), Edmodo (Edmodo, 2017) e Moodle (Moodle, 2017).

Além disso, uma tendência nos dias de hoje é o crescimento do uso de AVAs que englobam características consideradas “inteligentes”, principalmente no que se refere à possibilidade de ensino flexível. Tais ambientes possuem características de Sistemas Tutores Inteligentes (STIs) e podem ser considerados o que chamamos de *Smart Learning Environments* (SLEs), também chamados de *Smart Educational Environments* (Kinshuk, 2016). Nesses ambientes, conforme as características do aluno, por exemplo, seus estilos de aprendizagem, o ambiente é capaz de se adaptar e exibir conteúdos e atividades com a tentativa de alcançar uma aprendizagem mais efetiva (Spector, 2016). Ferramentas no campo da Inteligência Artificial (IA), como ontologias e agentes de software, podem atuar integrados ao AVA sendo responsáveis por sua camada de inteligência. Essa abordagem dos AVAs com características inteligentes vem ao encontro das modernas teorias de educação, que trocam o modelo centrado no professor por um modelo centrado no aluno, chamado de “modelo de aluno”. Outro aspecto importante nos ambientes educacionais modernos são os sistemas “sensíveis ao contexto” (*context-awareness*), sendo que tal termo significa “a habilidade do computador de sentir e agir de acordo com a informação provida pelo seu ambiente, tais como localização, tempo, temperatura e identidade do usuário” (Economides, 2008).

Portanto, é cada vez mais frequente utilizarmos AVAs com adaptações da disposição de conteúdo ou dos recursos educacionais apresentados aos usuários. Nesta tese, convenciamos que a adaptação de recursos em AVAs é entendida não apenas como a adaptação ou ajuste do que é apresentado para o aluno no AVA, como atividades passadas, objetos de

aprendizagem, mudanças na interface e auxílio personalizado de acordo com o modelo de aluno, mas como os recursos estarão dispostos no AVA (por exemplo, como ocorrerá uma recomendação de recursos educacionais) de tal forma que permitam mudanças na organização pedagógica da aprendizagem de acordo com o perfil do aluno, utilizando tecnologias (Bremgartner, Netto e Menezes, 2017). Um dos objetivos originais dos ambientes educacionais inteligentes é guiar cada estudante para o conteúdo educacional mais apropriado (Hosseini et. al., 2015). No entanto, é importante diferenciar a adaptação da personalização de AVAs. A adaptação pode ser vista como uma perspectiva do lado do AVA, enquanto que a personalização considera uma perspectiva individual do estudante, onde ambos têm como objetivo melhorar a aprendizagem dos estudantes pelo aumento da eficiência, eficácia e satisfação do estudante (Kinshuk, 2016).

Em *Smart Learning Enviroments* ou AVAs que utilizam suas características, uma tendência é considerar o modelo do aluno, que pode ser obtido através do perfil de aluno, cujo modelo é um registro de ações do estudante, além de informações sobre seus históricos. O modelo de aluno é um dos elementos principais para a personalização dos AVAs (Brusilovsky, 1999). Enquanto que o perfil de aluno constitui-se em dados brutos dos alunos no sistema (Fröschl, 2005), o modelo de aluno constitui-se das informações consideradas mais importantes do perfil do aluno para o processo de adaptação e personalização em AVAs, a fim de alcançar um aprendizado mais efetivo por parte do aluno, como por exemplo, informações pessoais, preferências e desempenho em atividades. Instituições e organizações internacionais têm trabalhado na padronização de modelos de aluno usados em AVAs ou STIs, sendo o padrão mais conhecido e mais usado atualmente, o *Instructional Management Systems Learning Information Package* – IMS LIP (LIP, 2017).

1.1 Descrição do Problema e uma Proposta de Solução

Apesar da crescente utilização de AVAs com características de SLEs, muitas vezes estes apresentam recursos de aprendizagem da mesma maneira para todos os estudantes (problema chamado de *one-size-fits-all*) (Bremgartner, Netto e Menezes, 2014), resultando que o aprendizado pode não se tornar efetivo para todos devido às diversas características cognitivas que cada aluno possui. Isso gera, portanto, dificuldades de construção de conhecimento para alguns estudantes ou até mesmo falta de interesse do aluno pela participação do curso no AVA. Existem várias técnicas de adaptação de conteúdo para os alunos vistos na literatura, como

sistemas sensíveis ao contexto (Saccol *et al.* 2009), sistemas conversacionais (Latham, Crockett e McLean, 2013), apoio a formação de grupos (Oliveira e Tedesco, 2009), personalização com agentes (de la Iglesia *et. al.*, 2015), mas nem todos contemplam as diversas características que o aluno possui. Além disso, há uma necessidade de melhorar o processo de aprendizagem colaborativa nos AVAs, onde este deve promover o engajamento e a aprendizagem efetiva do estudante de acordo com seu perfil.

Muitas vezes, nas gerações modernas de AVAs, candidatos a SLEs, já se usam várias técnicas de adaptação, onde algumas apresentam novidades tecnológicas, mas não são sustentadas por teorias pedagógicas. Os professores e tutores poderiam até adaptar atividades e trabalhos manualmente para cada aluno de acordo com o perfil deste, mas torna-se dispendioso para o professor e os tutores tal trabalho à medida que estes precisam acompanhar cada estudante no decorrer do curso.

Portanto, ao depararmos com diversas técnicas de adaptação em AVAs, para atender às necessidades do aluno, nos questionamos sobre como podemos utilizar as modernas Tecnologias da Informação e Comunicação (TICs) para prover adaptações nos AVAs que permitam a construção do conhecimento do aluno com base em seu perfil. Percebe-se assim, que temos várias entidades que fazem parte no processo de adaptação de recursos nos AVAs, que são: usuários (estudantes, professores, tutores, entre outros), software e estratégias pedagógicas.

Para elucidar melhor estes fatos que norteiam o problema que este trabalho aborda, consideremos seis cenários de onde podemos ter adaptações de recursos tendo em vista os perfis dos usuários:

Cenário 1:

Alunos de uma disciplina de Introdução a Programação de Computadores de um típico curso na área de ciências exatas, seja nas modalidades semipresencial ou à distância, utilizando um AVA, estão com dificuldades de elaborar algoritmos para resolver problemas. Há professores e tutores monitorando o curso e interagindo com os alunos neste ambiente, entretanto não há um gerenciamento de forma automática e eficaz com relação ao acompanhamento das dúvidas que os alunos possuem sobre um determinado assunto (quando os alunos possuem alguma dificuldade na realização das atividades, que poderia ser detectada pela incorretude das respostas dos alunos). As dificuldades do aluno ao executarem as tarefas propostas também podem ser evidenciadas se uma tarefa não for entregue até a data estipulada para sua entrega. Apesar de existirem serviços que permitam uma comunicação aluno-tutor ou aluno-

professor para tratar de dúvidas nos AVAs, como fóruns e chats, nem sempre todos os alunos irão postar todas suas dúvidas, ou até mesmo poderão postar mensagens fora do escopo da disciplina, dificultando o entendimento de qualquer outro colega, tutor ou professor. Além disso, com o demasiado tempo gasto em uma turma com muitos alunos, tal situação não permitiria ao professor nem ao tutor ficarem livres para executarem outras atividades pertinentes ao curso.

Um sistema multiagente poderia sugerir recursos educacionais como atividades, desafios abertos, atividades colaborativas com colegas que possam auxiliá-lo a resolver seus problemas ou até mesmo Arquiteturas Pedagógicas em um AVA, que permitam construir o conhecimento do aluno. Para isso, seria necessário o sistema multiagente conhecer o perfil do aluno em questão, sua situação no ambiente educacional e suas necessidades. O perfil do aluno iria mudando dinamicamente, à medida que o aluno fosse interagindo com o AVA, em função, por exemplo, de seus resultados de atividades. Acredita-se que assim o estudante iria obter uma aprendizagem mais significativa.

Cenário 2:

Um grupo de alunos participantes de um curso que chamaremos de não convencional (por não possuir um currículo fixado, como ocorre em uma disciplina comum de Algoritmos e Programação), por exemplo, um curso de extensão sobre Pensamento Computacional, utiliza um AVA, seja na modalidade EaD ou semipresencial. Nesse curso há alunos de diferentes idades, seja de Ensino Médio, Ensino Superior, alunos de Educação de Jovens e Adultos (EJA). Cada estudante poderia começar o curso, por sua livre escolha, a partir de qualquer **Unidade de Ensino** (UE) disponível previamente pelo professor, que contém assuntos ou tópicos da disciplina seguidos de atividades.

Dependendo do perfil de cada aluno, o AVA poderia recomendar novas UEs, objetos de aprendizagem, atividades colaborativas, entre outros recursos, para aquele aluno. Este ficaria livre para escolher os recursos que quiser acessar disponíveis dependendo do seu perfil. No entanto, ao final do curso, todos os alunos precisariam ter adquirido todas as competências envolvidas no curso. Em outras palavras, os objetivos do curso deveriam ser alcançados, com o aluno egresso tendo aprendido sobre Pensamento Computacional aplicado em sua área. Além disso, nos recursos que envolvem atividades, muitas situações problematizadoras seriam passadas a ele, a fim de construir seu conhecimento (a exemplo do que ocorre na Aprendizagem Baseada em Problemas – *Problem-Based Learning*). Um sistema inteligente seria res-

ponsável por esse processo de recomendação e adaptação de recursos, além de estimular a aprendizagem colaborativa entre alunos, a fim de proporcionar maior liberdade e aprendizado em conjunto entre os alunos. O sistema também verificaria o aprendizado dos alunos, se foi atingido ou não. A diferença deste cenário para o Cenário 1 se dá pelo fato desse curso ter um currículo mais flexível, não estando em função dos padrões curriculares tradicionais de sala de aula ou mesmo em cursos de EaD.

Cenário 3:

Consideremos uma empresa em que os funcionários mantêm um registro de suas atividades em um banco de dados no servidor da empresa por meio de um ambiente Web. De acordo com os registros dos funcionários, o ambiente vai obtendo informações do perfil de cada um e cada vez mais vai sendo formada uma base de conhecimento a respeito desses funcionários. Supomos que a cúpula da empresa queira saber, por exemplo, além de um relatório a respeito do perfil de cada funcionário no que diz respeito as suas habilidades, queira saber quais funcionários são os mais aptos para atuarem em determinados problemas ou projetos da empresa e quem seriam os colegas ideais para ajudá-lo. Para que isso se concretizasse, poderíamos dotar esse ambiente virtual da empresa de um mecanismo de inteligência, por exemplo, agentes inteligentes que, com a ajuda de mecanismos de aprendizagem de máquina, obtenham o histórico de ações de cada funcionário, verificando o que ele fez no passado, a fim de prever quais funcionários podem atuar com sucesso para resolver determinados problemas na empresa. Os recursos recomendados aqui seriam os funcionários aptos a resolverem determinados problemas, podendo ainda, como extensão, agregar itens e ferramentas que os próprios funcionários poderiam utilizar, a partir dos dados de perfil de funcionários.

Além disso, o ambiente poderia detectar se algum funcionário está tendo alguma dificuldade para resolver algum problema ou atividade designada a ele na empresa e recomendar estratégias para esse funcionário poder resolver melhor seu problema. Ainda, conforme citado no parágrafo anterior, o ambiente também iria percebendo quais funcionários seriam os indicados para resolver determinados problemas na empresa e tal informação seria passada para seus superiores e colegas via ambiente Web. Nesse ambiente, cada informação de problema detectado e funcionário recomendado para ajudar seu colega seriam armazenadas em um repositório no servidor. Assim, um funcionário poderia ser recomendado para tratar ou resolver determinado problema na empresa de forma colaborativa com seus colegas, formando uma comunidade de prática. Em todo esse processo, ocorreria também a aprendizagem situada nos usuários dessa comunidade de prática, contendo elementos para que as dúvidas dos usuários

ao resolverem um problema ou executarem alguma tarefa fossem esclarecidas. Enfim, este cenário trata do acompanhamento das ações de funcionários em uma empresa, com o ajuste dos respectivos perfis (modelos de usuários), banco de especialistas (funcionários que possam ajudar a outros funcionários ou líderes de equipes para resolver problemas).

Cenário 4:

Em um curso que utiliza um AVA, os alunos e professores podem acessá-lo pelo seu desktop ou laptop, mas alguns usuários, por questões de comodidade e por poder acessar a qualquer tempo e lugar, preferem acessar o AVA por meio de dispositivos móveis (smartphone, tablet, entre outros), realizando algumas atividades e usufruindo de seus recursos. No entanto, existem usuários de diversos níveis de experiência em acessar diversos aplicativos e ambientes nos dispositivos móveis. Podemos agregar uma camada de inteligência composta por agentes de software nos AVAs quando acessados por dispositivos móveis para a interface ser ajustada de acordo com o perfil do aluno no que diz respeito a sua experiência em acessar AVAs nos dispositivos móveis, pois podem existir alunos que são menos experientes que outros em compreender o uso desses ambientes, principalmente quando se trata de acessar em um dispositivo móvel. Nesse caso, o sistema inteligente propiciaria uma alfabetização digital para esses estudantes.

Cenário 5:

Ainda falando a respeito dos dispositivos móveis, uma outra situação possível seria, ao aproveitar as diversas características do dispositivo móvel, por exemplo, a de saber o lugar em que o aluno está quando conectado a internet, por meio do Sistema de Posicionamento Global (*Global Positioning System – GPS*), em que um sistema inteligente poderia recomendar atividades, sugestões de recursos de vídeos, leituras, entre outros, relacionados com o lugar que o aluno se encontra e ao mesmo tempo, com o conteúdo ministrado (por exemplo, um estudante de História poderia obter mais informações sobre o ciclo da borracha no estado brasileiro do Amazonas ao passar próximo ao Teatro Amazonas). Poderíamos considerar que o modelo de aluno seria suas características adicionadas às características do dispositivo móvel. Atividades colaborativas entre os alunos utilizando os dispositivos móveis poderiam ser recomendadas, a fim de explorar a modalidade de *Mobile Learning* ou m-learning, que é o emprego da tecnologia móvel para fins educacionais (Ji *et. al.*, 2011).

Cenário 6:

Imaginemos um grupo de pesquisa de Informática (por exemplo, na área de Inteligência Artificial e Educação) que é formado por diversos alunos de diferentes níveis acadêmicos (ensino médio, ensino superior, especialização, mestrado e doutorado), além de bolsistas e técnicos de laboratório que utilizam um AVA para trocar experiências e fazer atividades solicitadas por seu orientador. Nesse grupo há alunos que são mais experientes em certos tópicos que outros colegas e todos querem aprender novos assuntos. Por exemplo, um aluno do mestrado pode ser muito hábil em Learning Analytics (LA) e cálculos estatísticos, enquanto que o de doutorado pode ser mais experiente no hardware Arduino (Arduino, 2017) e um aluno de graduação ter habilidades e competências em programação para dispositivos móveis. Se, por exemplo, o aluno de doutorado registrar no AVA que gostaria de obter ajuda de algum colega que possua experiência em desenvolver aplicações móveis, o ambiente recomendaria o aluno de graduação para auxiliar o seu colega.

Um sistema inteligente saberia o perfil de todos os alunos no AVA, suas habilidades e competências, além de seus interesses, seja por meio de questionários aplicados ou pelas manifestações dos estudantes no AVA, como por exemplo, registros de atividades, interesses, publicações, que contém metadados indicando a(s) área(s) em que cada estudante tem atuado. Cada habilidade/experiência dos alunos seria registrada a partir do início da interação dos mesmos com o AVA. O ambiente seria utilizado como forma de ajudar as trocas de experiências entre os alunos e um facilitador da organização do grupo para o próprio orientador. Para que ocorresse a atualização dos perfis de cada aluno, os alunos fariam periodicamente um relatório de suas atividades no grupo de pesquisa, constituindo um portfólio de aprendizagem.

Discussão dos Cenários

Em comum, os exemplos acima citados tratam de interações complexas entre usuários, equipamentos e software, nas quais as interações são realizadas entre todas essas entidades e tratam de questões cotidianas e que já se tornam cada vez mais frequentes e necessárias no nosso mundo real. Sendo assim, em cada um desses exemplos, a recomendação de recursos que auxiliem o aprendizado dos usuários envolvidos nos ambientes virtuais de forma colaborativa por meio de uma camada de inteligência é uma solução viável para tratar todos esses problemas que os cenários trazem. É nessa integração de provisão de recursos educacionais adaptados e personalizados com base no perfil do usuário (aqui tratado como o modelo do aluno) que esta tese está direcionada.

Estratégia de solução

Como solução estratégica para o problema descrito anteriormente e que contempla todos os cenários descritos, projetamos um arcabouço chamado ArCARE (Arcabouço Conceitual de Adaptação de Recursos Educacionais), que consiste num conjunto de elementos para promover a adaptação de recursos em ambientes educacionais (AVAs, SLEs,...), contendo agentes inteligentes e uma ontologia de modelo de aluno que abrange quatro itens para descrever o aluno: (a) desempenho em atividades e frequência no ambiente educacional; (b) os estilos de aprendizagem; (c) preferências (ou interesses) e (d) competências e habilidades. Com a integração entre os agentes e a ontologia de modelo de aluno, o AVA estará apto a adaptar recursos educacionais de acordo com as necessidades dos estudantes. O ArCARE será explicado com maiores detalhes no Capítulo 4.

Sendo assim, o cerne deste trabalho está voltado para a elaboração de um arcabouço conceitual contemplando agentes e uma ontologia de modelo de aluno contendo diversas características do aluno a fim de descrever o mesmo e com isso, utilizar este modelo para prover adaptações e recomendações de recursos educacionais em AVAs sob uma ótica construtivista, por exemplo, uma recomendação de Arquiteturas Pedagógicas (APs). Como prova de conceito, desenvolvemos implementações de dois sistemas baseados no ArCARE que foram integrados em um AVA. Assim, adaptações foram realizadas e testamos a eficácia desses dois sistemas nos dois primeiros cenários descritos, sendo cada cenário um caso de teste. No primeiro cenário, foi criado e utilizado o sistema AMPARA (do inglês, *Adaptive Multi-agent Pedagogical Architectures for Resources and Activities in VLEs*, que em português fica Arquiteturas Pedagógicas Multiagente Adaptativas para Recursos e Atividades em AVAs), que é baseado no ArCARE, aplicado em cursos convencionais e que será explicado com maiores detalhes no Capítulo 5. E no segundo cenário, explicado no Capítulo 6, foi implementada uma extensão do AMPARA (AMPARAX – eXtended AMPARA) que foi utilizada num curso de currículo flexível de Pensamento Computacional. O AVA utilizado para os dois casos de testes foi o Moodle (Moodle, 2017). Com os agentes e a ontologia de modelo de aluno, o AVA em que o conceito do ArCARE estiver inserido possui características de SLEs (apresentadas na Seção 4.2) a partir da adaptação de recursos promovida pelas tecnologias mencionadas.

1.2 Motivação e Justificativa

Com o uso de AVAs, percebe-se que ainda há muitas abordagens *one-size-fits-all* nestes ambientes. A estratégia utilizada ao propor o ArCARE para solucionar o problema descrito na Seção 1.1 permite a adaptação dos AVAs por meio dos recursos educacionais mostrados ao aluno, de acordo com o modelo do aluno baseada na teoria construtivista da aprendizagem, a fim de que o aluno obtenha maior aprendizado efetivo. Esta estratégia permite uma maior flexibilidade e poder de adaptação em AVAs, de acordo com as características dos alunos descritas no seu modelo de aluno. O ponto de partida para esse trabalho foi realizado a partir da análise detalhada de competências e habilidades dos alunos apresentado nos trabalhos de (Bremgartner e Netto, 2011a) e (Bremgartner e Netto, 2011b), onde foi observado que há uma maior facilidade de adaptação e personalização de AVAs quando se detalha os dados do aluno.

Além disso, é possível notar que há uma necessidade de melhorar a prática da aprendizagem colaborativa mediada por AVA. As dinâmicas colaborativas propostas pelo ArCARE devem promover o engajamento e a aprendizagem efetiva do estudante de acordo com seu perfil. A ontologia e os agentes de software são os elementos responsáveis pela camada de inteligência do arcabouço e eles têm recursos tecnológicos para identificar o contexto do AVA e agir conforme os eventos dinâmicos que ocorrem nele, como por exemplo, a identificação de habilidades e competências de um estudante, bem como fazer avaliações das respostas dos alunos nas atividades propostas, a fim de identificar os alunos que podem estar com dúvidas e dificuldades. Além disso, estas tecnologias são úteis para trabalhar de uma forma mais proativa e ágil ao identificar as necessidades do aluno.

Além disso, quanto a utilização do modelo de aluno, este será um Modelo Aberto de Aluno (MAA). Dessa forma, acreditamos que o aluno terá um maior conhecimento sobre a sua situação no AVA. O aluno pode se julgar muito capaz numa determinada disciplina, mas o sistema pode acusar que o aluno não está tão capaz como pensava. Assim, espera-se que o aluno procure saber se há alguma deficiência de conhecimento por parte dele na disciplina e procure suprir quaisquer deficiências por processo de reflexão, autorregulação e aprendizagem independente no ambiente. Quanto a visualização do MAA, esta pode ser vista como um tipo específico de Learning Analytics (LA), onde o que é visto é o modelo do aluno. LA é a medição, coleção, análise e relatório de dados sobre estudantes em seus contextos, para propósitos de entendimento e otimização da aprendizagem e do ambiente onde ela ocorre (Siemens, 2012). Como qualquer processo de LA visual tem como objetivo tornar dados comple-

xos disponíveis para auxiliar os usuários a interpretar diversos aspectos de aprendizagem, da mesma forma isso ocorre com o MAA. No entanto, uma diferença fundamental é que nos MAAs as inferências acerca da aprendizagem providas pela LA já ocorrem como parte do processo de modelagem (Bull e Kay, 2016).

1.3 Questão de Pesquisa e Hipóteses

Este trabalho tem como foco propor um arcabouço que permita o desenvolvimento de ambientes inteligentes para a promoção da aprendizagem por meio da adaptação de recursos em função das diversas características do aluno. Neste sentido, a questão de pesquisa e as duas hipóteses consideradas foram as seguintes:

QP: Como utilizar tecnologias da IA para conceber ambientes de aprendizagem que se ajustem às necessidades e preferências do aluno?

H1: É possível utilizar modelo de aluno para prover adaptações em AVAs de modo que a aprendizagem do aluno ocorra.

H2: Pode-se utilizar tecnologias de Inteligência Artificial (especificamente, agentes e ontologias) para prover adaptações que permitam a construção do conhecimento do aluno com base em seu modelo de aluno.

No decorrer desta tese, buscaremos responder esta questão de pesquisa (QP) e comprovar essas hipóteses (H1 e H2). A QP procura responder a forma como deve ser feita a adaptação de recursos em função do MAA. É o que esta tese pretende demonstrar, com a apresentação e validação do ArCARE.

1.4 Objetivos

O objetivo geral desta tese de doutorado é “Desenvolver um arcabouço conceitual baseado em tecnologias de IA com o qual possam ser construídos ambientes de aprendizagem que se adaptem às necessidades e preferências dos usuários.”

Sendo assim, o objetivo geral decompõe-se nos seguintes objetivos específicos:

- (1) Oferecer a possibilidade de uma interação entre alunos no ambiente, construindo conhecimentos a partir dos recursos apresentados a eles;
- (2) Propiciar a reflexão e a aprendizagem independente por parte do aluno com o uso de Modelo Aberto de Aluno (MAA);

- (3) Promover a construção de sistemas (ou ambientes) educacionais a partir das definições do arcabouço que propiciem adaptação de recursos educacionais em AVAs.

1.5 Metodologia Aplicada

Os métodos utilizados para essa pesquisa foram:

- Revisão Sistemática de Literatura;
- Análise dos trabalhos correlatos;
- Caracterização do arcabouço conceitual a partir dos resultados da Revisão de Literatura;
- Definição do arcabouço conceitual, o ArCARE;
- Realização de provas de conceito por meio de dois casos de testes a partir de sistemas desenvolvidos com as diretrizes do ArCARE;
- Análises e Conclusões dos resultados obtidos.

1.6 Organização do Trabalho

Esta tese está organizada em sete Capítulos. Para o aprofundamento da compreensão, este trabalho apresenta no Capítulo 2 a fundamentação teórica sobre os tópicos principais abordados: Construtivismo, colaboração, cooperação, Aprendizagem Colaborativa Apoiada por Computador (CSCL), modelo de aluno, Arquiteturas Pedagógicas, ontologias, sistema multi-agente (incluindo arquitetura BDI) e estilos de aprendizagem de aluno. No Capítulo 3 serão apresentados os principais trabalhos correlatos ao nosso projeto. O Capítulo 4 explica em detalhes o ArCARE. O Capítulo 5 mostra um estudo de caso da aplicação do ArCARE, consistindo num sistema chamado AMPARA e mostra os resultados obtidos. O Capítulo 6 mostra outro estudo de caso com a aplicação do AMPARA estendido, o AMPARAX, em um minicurso de Pensamento Computacional. O Capítulo 7 apresenta as considerações finais, mostrando as conclusões e contribuições desta tese, suas publicações e extensões futuras deste trabalho. Por fim, temos as Referências, seguidas pelos Apêndices A até G.

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos que fundamentam este trabalho, divididos em dois grupos. Das Seções 2.1 a 2.4 tratamos do ponto de vista pedagógico, compreendendo o construtivismo de Piaget, Arquiteturas Pedagógicas, competências, habilidades e estilos de aprendizagem, enquanto que nas seções 2.5 até 2.10 tratamos do ponto de vista tecnológico, compreendendo agentes e ontologias, arquitetura BDI, Sistemas Tutores Inteligentes, Smart Learning Environments e perfil e modelo de aluno.

2.1 Abordagem Construtivista para a Aprendizagem

Este trabalho é baseado na concepção construtivista acerca da aprendizagem. Ao aplicarmos uma teoria pedagógica neste trabalho, o Construtivismo de Piaget (1977) foi escolhido. Esta teoria é aplicada em diversos trabalhos na literatura (Dominoni et al., 2010; Oliveira e Tedesco, 2009; Bastos Filho et al., 2006; Huandong et al., 2009; Liu et al., 2009; Latham et al., 2010; Martins e Carrapatoso, 2008; Schreurs e Al-Huneidi, 2012; Tai et al., 2010), que ajuda a comprovar que o estudante pode aprender por meio das suas interações com o meio em que ele se encontra.

A definição de Construtivismo é uma concepção do conhecimento e da aprendizagem, que deriva da epistemologia genética de Jean Piaget (1977) e da pesquisa sócio-histórica de Lev Vygotsky (1984) que partem da ideia de que o conhecimento se constitui pela interação do indivíduo com o meio físico e social, por força de sua ação e não por qualquer dotação prévia, nem pelas informações que o aluno recebe. Na Educação, essa teoria reúne outras tendências atuais do pensamento, que têm em comum a insatisfação com um sistema educacional que consiste em apenas repetir, recitar, ensinar o que já está pronto, ao invés de fazer agir, operar, trocar ideias, criar, construir a partir da realidade vivida por estudantes e docentes. Nesta concepção o conhecimento não se traduz em atingir a verdade absoluta, mas numa questão de adaptação do organismo a seu meio ambiente (Santana et al, 2009).

Segundo Costa e Franco (2005), tal concepção considera que o conhecimento se consolida a partir de auto-regulações que ocorrem através das relações estabelecidas entre o sujeito e o objeto. É a partir da constituição de novas relações baseadas no conhecimento prévio,

que se alcançam patamares cognitivos superiores, sempre levando em consideração o caráter da teoria piagetiana. Neste contexto, alguns elementos presentes no ato educativo precisam ser reinterpretados. O erro, que costuma ter um caráter punitivo e inibidor, passa a ser construtivo para a aprendizagem. Ele poderá revelar elementos a respeito do processo de aprendizagem do estudante, podendo assim, auxiliar o professor nas suas estratégias de mediação. Tratar o erro como um momento privilegiado de reflexão e investigação, considerando os mesmos como acontecimentos significativos e impulsionadores dessa prática, consiste estratégia fundamental para uma proposta de aprendizagem. Mais importante que acertar a resposta é dar ênfase ao processo de elaboração da mesma. O erro possibilita ao professor identificar as concepções prévias dos alunos, proporcionando a base para a concepção de futuras intervenções pedagógicas (Costa e Franco, 2005).

Segundo Zhou et. al. (2012), o Construtivismo é uma teoria que diverge da teoria Behaviorista, que esta diz que comportamentos individuais de aprendizagem são vistos como um processo de "estímulo-resposta" de se adaptar a ambientes externos, em que se nós controlarmos o estímulo, podemos controlar os comportamentos de aprendizagem e prevê-los e, assim, somos capazes de controlar e prever os resultados de aprendizagem. O Construtivismo está relacionado a teoria cognitiva da aprendizagem, que está focada no ponto de vista que o aluno deixará de receber estímulos externos passivamente. Eles são o principal corpo de processamento de informação para selecionar os estímulos externos ativamente. Sendo assim, de acordo com o Construtivismo, conhecimento não apenas vem da educação em sala de aula, mas também é construído pelos alunos com base no seu sistema de conhecimento prévio (Zhou et. al., 2012).

2.2 Arquiteturas Pedagógicas

A compreensão, na área educacional, acerca da expressão “arquiteturas pedagógicas” (AP) tem trazido múltiplas interpretações, sendo que estas estão diretamente relacionadas com uma linha epistemológica que dá embasamento para sua proposta pedagógica (Behar, Bernardi e Silva, 2009).

A AP pode ser compreendida como a construção de estratégias pedagógicas que tem como base uma determinada teoria e seus pressupostos a fim de auxiliar na efetivação da aprendizagem com suporte de recursos tecnológicos como os Ambientes Virtuais de Aprendizagem (AVAs) e/ou videoconferência. A construção da estratégia pedagógica envolve, entre-

tanto, a formação de uma equipe interdisciplinar com a participação de profissionais das áreas de educação e computação.

Partindo da concepção elaborada por (Carvalho, Nevado e Menezes, 2005), arquiteturas pedagógicas podem ser compreendidas como “estruturas de aprendizagem realizadas a partir da confluência de diferentes componentes: abordagem pedagógica, software, Internet, inteligência artificial, educação a distância, concepção de tempo e espaço”. Ou seja, são combinados os recursos tecnológicos com a visão pedagógica, sendo esta conjunção o elemento que fundamenta a AP. Para (Carvalho, Nevado e Menezes, 2005), pressupõe-se que as APs são viabilizadas pela convergência entre os paradigmas epistemológicos e as estratégias pedagógicas, acolhendo, assim, uma possibilidade de releitura dessas, demonstrando-se, mais receptivas à aprendizagem.

Assim, a construção das arquiteturas pedagógicas pode ser metaforicamente relacionada a uma atividade artesanal, onde será tecida uma rede de relações entre as experiências vivenciadas pelos sujeitos envolvidos e a reflexão sobre diferentes fatos e objetos relacionados com o meio de atuação em estudo. Este processo, entretanto, necessita de propostas pedagógicas abertas a uma abordagem didática flexível e adaptável aos mais diversos enfoques temáticos (Carvalho, Nevado e Menezes, 2005).

Ferramentas como wikis, blogs e fóruns de discussão, bem como a utilização de técnicas de pesquisa, permitem uma interatividade que supera as práticas convencionais, o que vem a agregar na construção de ambientes integradores, coletivos e colaborativos que podem ser utilizados como arquiteturas pedagógicas nos processos educativos (Marcon, Machado e Carvalho, 2012). Alguns exemplos de arquiteturas pedagógicas são as atividades colaborativas debate de teses, controvérsia acadêmica e projetos de aprendizagem, apresentados em (Castro e Menezes, 2011).

2.3 Definindo Competências e Habilidades

Esta Seção aborda os conceitos de competências e habilidades vistos na literatura pesquisada. Segundo Valente (2002), “um aspecto que se evidencia clara e angustiadamente, quando se busca compreender o significado das competências e habilidades no contexto educacional, é a escassez de produção teórica a respeito”. Devido à escassez de fontes de consultas, acrescen-

tam-se outros fatores que dificultam o entendimento dos conceitos de habilidades e competências:

- a) Omissão das Diretrizes e Parâmetros Curriculares Nacionais (MEC, 2017) quanto aos conceitos de competências e habilidades;
- b) Nos Parâmetros Curriculares Nacionais, o quadro que consta ao final de cada área e disciplina traz as respectivas competências e habilidades, como se ambas fossem idênticas;
- c) Na literatura, ora as competências abrangem as habilidades, ora são diferenciadas conceitualmente das habilidades.

Sendo assim, diversos autores explicam suas ideias de competências e habilidades. Perrenoud (1999) expressa que não existe uma ideia clara e absoluta de competências, no qual esta palavra possui muitos significados e ninguém pode se arriscar em dar uma definição única. Perrenoud afirma também que o próprio conceito de competência mereceria longas discussões, onde a noção de tal termo designa uma capacidade de mobilizar diversos recursos cognitivos para enfrentar diversos tipos de situações (Perrenoud, 2000). Deffune e Depresbiteris (2000) também buscaram definir competências: “as definições dependem dos autores que foram baseadas, das metodologias de análise das atividades do mundo do trabalho e das maneiras como as competências serão vertidas para o currículo”.

Outros conceitos relevantes encontrados relacionados a competências são:

“Competência - s.f. Atribuição, jurídica ou consuetudinária, de desempenhar certos encargos ou de apreciar ou julgar determinados assuntos: competência de um tribunal. / Capacidade decorrente de profundo conhecimento que alguém tem sobre um assunto: recorrer à competência de um especialista.”. (Dicionário Aurélio, 2017).

“São múltiplos os significados de competência. Eu a definirei aqui como sendo uma capacidade de agir eficazmente em um determinado tipo de situação, apoiada em conhecimentos, mas sem limitar-se a eles.” (Perrenoud, 1999). O autor ainda divide as competências em principal e específicas.

“Competência é a aplicação prática de conhecimentos, aptidões, habilidades, valores, interesses – no todo ou em parte – com obtenção de resultados.” (Resende, 2000)

“Competências Básicas - conhecimentos, habilidades, atitudes e apreciações, geralmente requeridas para o desempenho bem sucedido das funções que caracterizam determinado tipo de atividade profissional.” (Ramos, 1980).

Deffune e Depresbiteris (2000) tratam de uma série de conceitos sobre competências e habilidades:

“Competência é a capacidade de uma pessoa para desenvolver atividades de maneira autônoma, planejando-as, implementando-as e avaliando-as. Competência é a habilidade de alguém usar seu conhecimento para alcançar um propósito.”

“Competência é a capacidade para usar habilidades, conhecimentos, atitudes e experiência adquirida para desempenhar bem os papéis sociais.”

Outro conceito de competência é o utilizado na especificação IMS Reusable Definition of Competency or Educational Objective (RDCEO) (2017), que é uma especificação que define as características chave de competência do aluno, independente do uso desta especificação em um contexto particular. No RDCEO, competência é uma palavra usada em um sentido geral, que inclui habilidades, conhecimentos e resultados de aprendizagem.

Referindo-se ao fato de que não há sentido em diferenciar competências e habilidades, Perrenoud é enfático:

“Existe a tentação de reservar a noção de competência para as ações que exigem um funcionamento reflexivo mínimo, que são ativadas somente quando o ator pergunta a si mesmo: O que está ocorrendo? Por que estou em situação de fracasso? O que fazer? Etc.”.

Porém, o autor nos dá uma ideia de competências e habilidades que pode ser utilizada em diversas situações na Educação:

“Competência em educação é a faculdade de mobilizar um conjunto de recursos cognitivos - como saberes, habilidades e informações - para solucionar com pertinência e eficácia uma série de situações”.

Segundo Perrenoud, pode-se concluir que a competência é um conjunto de saberes necessários para realizar uma determinada atividade e a habilidade é um saber-fazer relacionado à prática do trabalho, mais do que mera ação motora. Nesta abordagem, o autor expressa a competência como um conjunto ou feixe de habilidades, embora haja uma distinção entre os

dois conceitos (Perrenoud; Thurler *et al.*, 2002). É como se as habilidades fossem microcompetências, ou como se as competências fossem macrohabilidades.

Nesta tese são utilizados os conceitos de Perrenoud e da especificação RDCEO do padrão LIP no que tange as competências e habilidades, onde os detalhamentos sobre como tais conceitos foram utilizados para o modelo do aluno são explicados no Capítulo 4, seção 4.2.

2.4 Estilos de Aprendizagem do Aluno

São muitas as teorias propostas acerca de um tema extremamente complexo que são os processos de ensino-aprendizagem. Um ponto consensual a estas teorias é que cada indivíduo possui um ritmo e forma característica de aprender, considerando ainda que diversos fatores como os ambientais, físicos, emocionais, cognitivos e sociais influenciam esta relação (Senra, Lima e Dasilva Fwo, 2008).

Existem diversas definições para estilos de aprendizagem na literatura. Para Sadler-Smith (2001), estilo de aprendizagem é o modo distintivo e habitual pelo qual o aprendiz adquire conhecimento. Desta forma os estilos se expressam consistentemente em diferentes domínios de conteúdo e áreas e podem ser observados em termos de comportamentos típicos e tendências, demonstrando o modo pelo qual uma pessoa apresenta maior facilidade para aprender. Os estilos de aprendizagem podem apontar, por exemplo, que alguns estudantes apresentam facilidade de compreensão a partir de demonstrações matemáticas, enquanto outros necessitam de visualizações gráficas para obter conhecimento tácito.

A seguir, serão explicados as duas classificações de estilos de aprendizagem utilizadas nos estudos de casos desta tese: Felder-Silverman e Honey-Alonso.

2.4.1 Estilos de Aprendizagem de Felder-Silverman

Richard Felder, a partir de sua percepção com o alto índice de repetência e evasão nos cursos superiores, iniciou suas investigações com o objetivo de contribuir para que os educadores compreendessem melhor as formas de aprendizagem de seus estudantes. Para ele, estratégias e metodologias mais adequadas poderiam ser definidas com o propósito de aumentar a eficácia das aulas e reduzir a insatisfação dos estudantes. Assim, juntamente com a psicóloga Linda Silverman, propôs o seu modelo de estilo de aprendizagem de Felder e Silverman, (1988).

Felder e Silverman conceberam suas ideias fundando-se nos estágios de percepção e processamento da informação, já apresentados por outros teóricos, de onde propuseram quatro dimensões:

- Percepção (descoberta): os estudantes podem ser sensoriais ou intuitivos;
 - Entrada (aquisição): os estudantes podem ser visuais ou verbais;
 - Processamento (análise): os estudantes podem ser ativos ou reflexivos;
 - Entendimento (aprendizagem): os estudantes podem ser sequenciais ou globais.
- Algumas características destas dimensões podem ser vistas na Tabela 1.

Tabela 1: As características de aprendizagem.

Sensorial	Apreciam fatos, dados, experimentos, métodos padrões, tem facilidade para memorização e preferem abstrair informações pelos seus sentidos (vendo, ouvindo, tocando etc.).
Intuitivo	Apreciam princípios, conceitos e teorias, não se atentam a detalhes, não gostam de repetição, se interessam por desafios, analisam possibilidades, significados e relações entre as coisas.
Visual	Assimilam mais o que veem (figuras, gravuras, diagramas, fluxogramas, filmes etc.).
Verbal	Preferem explicações escritas ou faladas à demonstração visual, extraem mais informações em uma discussão.
Ativo	Preferem experimentar ativamente que observar e refletir. Gostam de processar as informações enquanto em atividade e não aprendem de forma passiva.
Reflexivo	Preferem sozinhos e silenciosamente processar a informação. Fazem ligações teóricas com a fundamentação da matéria e não extraem muito quando não são levados a pensar.
Sequencial	Aprendem de forma linear, por etapas sequenciais, com o conteúdo se tornando progressivamente complexo.
Global	Aprendem em grandes saltos, sintetizam o conhecimento e podem não ser capazes de explicar como chegaram às soluções.

Fonte: [Vieira Junior, 2010]

Para mensurar, portanto, a presença destes atributos, Felder e Soloman desenvolveram um questionário chamado Índice de Estilos de Aprendizagem (ILS – *Index of Learning Styles*). Este questionário pode ser respondido gratuitamente na página <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>.

2.4.2 Questionário Honey-Alonso de Estilos de Aprendizagem

Embora existam diversas investigações sobre estilos de aprendizagem com diferentes abordagens e classificações, após análise e reflexão sobre o tema, além dos estilos de aprendizagem de Felder-Silverman, adotaremos para os estilos de aprendizagem as designações apresentadas por Honey e Mumford (1992), os quais consideram que as principais características das pessoas podem ser integradas em quatro estilos de aprendizagem: ativo, reflexivo, teórico e pragmático. Admitem, ainda, que em cada aluno é possível identificar características dos vários estilos de aprendizagem, embora geralmente, cada pessoa possua um estilo dominante (Miranda e Morais, 2008). Os mesmos autores, Honey e Mumford (2000), apresentam como principais características dos estilos de aprendizagem as seguintes:

- **estilo ativo:** empenham-se em novas experiências, têm uma mente aberta, entusiasmam-se com qualquer coisa nova, são sociáveis e envolvem-se constantemente com os outros, procuram ser o centro de todas as atividades, interessam-se por desafios e situações problemáticas, manifestam forte implicação na ação;
- **estilo reflexivo:** dão prioridade à observação antes da ação, gostam de observar as experiências de diversas perspectivas, centram-se na reflexão e na construção de significados, recolhem informações tanto da sua própria experiência como da experiência dos outros, preferem pensar antes de chegarem a qualquer conclusão, gostam de observar os outros em ação e de perceber o sentido geral da discussão antes de dizerem o que está na sua própria mente;
- **estilo teórico:** tendem a estabelecer relações, deduzir, integrar os fatos em teorias coerentes, tendem a ser perfeccionistas, gostam de analisar e de sintetizar. A sua abordagem aos problemas é consistente e lógica. Procuram a racionalidade e a objetividade, sentem-se desconfortáveis com conclusões subjetivas, pensamentos laterais ou qualquer aspecto superficial;
- **estilo pragmático:** gostam muito de experimentar ideias, teorias e técnicas para ver se funcionam na prática. O seu ponto forte é a aplicação das ideias. Gostam de atuar de uma forma confiante e rápida sobre as ideias e os projetos que os atraem; tendem a evitar a reflexão e ficam impacientes com discussões sem fim. Essencialmente, são pessoas práticas, que gostam de chegar a conclusões práticas e de resolver problemas.

Tendo como objetivo melhor clarificar as principais características associadas às pessoas de cada um dos estilos, são apresentados a seguir os termos que traduzem as opiniões de Alonso et. al., (1999), os quais salientam:

- **estilo ativo:** animador, improvisador, descobridor, destemido e espontâneo. Associam ainda às pessoas com este estilo dominante as características de criativo, inovador, aventureiro, renovador, inventor, protagonista, conversador, divertido, participativo, competitivo, com desejo de aprender e de resolver problemas;

- **estilo reflexivo:** ponderado, consciente, receptivo, analítico e exaustivo. Consideram ainda, associado às pessoas com este estilo predominante as características: observador, paciente, cuidadoso, construtor de argumentos, estudioso de comportamentos, investigador, questionador e prudente;

- **estilo teórico:** metódico, objetivo, lógico, crítico e estruturado. Acrescentam ainda que as pessoas com este estilo têm as características de: disciplinado, sistemático, sintético, perfeccionista, generalizador, explorador, investigador de teorias, modelos e conceitos;

- **estilo pragmático:** experimentador, prático, direto, eficaz e realista. Os mesmos autores associam, ainda, às pessoas deste estilo as características: técnico, rápido, decidido, positivo, concreto e claro.

Para evitar que cada aluno possa ser integrado num só estilo de aprendizagem, desprezando as características próprias de outros estilos, Honey e Mumford (1992) propõem para cada estilo, cinco níveis de preferência, designados por: preferência muito alta, preferência alta, preferência moderada, preferência baixa e preferência muito baixa. Desta forma é possível ter alunos com um nível de preferência muito alta num estilo e ao mesmo tempo ter um nível de preferência muito baixa nos outros.

Como salienta Labour (2002), o questionário de Honey-Mumford é uma abordagem para ajudar os alunos a estarem atentos aos seus pontos fortes e fracos dos estilos de aprendizagem. Neste sentido, e atendendo que as características associadas aos estilos são predominantemente defensáveis no contexto de ensino e de aprendizagem, é importante que cada aluno se preocupe por desenvolver competências que o permitam considerar com preferências muito altas em todos os estilos. Após a adoção das designações dos estilos a utilizar, surge a necessidade de saber com que instrumentos podem ser identificados os estilos de aprendizagem de cada pessoa.

Quando um aluno responde a um questionário o seu estilo de aprendizagem manifesta-se nos modos privilegiados de funcionamento que emergem da representação que o aluno tem

dele mesmo nas situações de aprendizagem que as questões apresentadas lhe evocam (Chévrier et al., 2000).

Para identificar as características individuais dos alunos, os investigadores têm desenvolvido modelos e instrumentos que evidenciam os estilos de aprendizagem dos alunos. Deses instrumentos temos o “*Cuestionario Honey-Alonso de Estilos de Aprendizaje (CHAEA)*”. Esta ferramenta, de acordo com os seus autores (Alonso et al., 1999), é “um questionário fruto da tradução e adaptação ao contexto académico espanhol, do Questionário de Estilos de Aprendizagem (LSQ), *Learning Styles Questionnaire*, de P. Honey elaborado para profissionais de empresas do Reino Unido”.

O questionário Honey-Alonso de Estilos de Aprendizagem, sendo fruto da tradução e adaptação do questionário de Honey-Mumford (LSQ), segue de perto as orientações e a estrutura do questionário-base (Miranda e Morais, 2008).

Tendo como quadro de referência o modelo de aprendizagem experimental de Kolb, (Honey e Mumford, 1992) desenharam um questionário que explora as preferências de estilos de aprendizagem em quatro dimensões: ativos, reflexivos, teóricos e pragmáticos. Para Alonso et. al. (1999), as pessoas parecem manifestar preferências que se concentram mais em determinadas etapas do processo de aprendizagem pela experiência. Neste sentido, sintetizam a abordagem ao processo de aprendizagem pela experiência e relacionam-na com essas preferências do seguinte modo:

- Viver a experiência: estilo ativo;
- Reflexão: estilo reflexivo;
- Generalização, elaboração de hipótese: estilo teórico;
- Aplicação: estilo pragmático.

Após uma breve contextualização de suporte do questionário CHAEA, procurou-se caracterizar a sua formatação. O questionário CHAEA, tal como o questionário de Honey-Mumford é constituído por 80 itens breves, correspondendo 20 a cada um dos quatro estilos referidos: ativo, reflexivo, teórico e pragmático.

No questionário CHAEA, além das 80 questões, sendo 20 relativas a cada estilo de aprendizagem, distribuídas aleatoriamente, salientam-se: questões acerca dos dados pessoais dos alunos e instruções acerca do seu preenchimento. Estas são indicações breves e concisas que os alunos devem cumprir com exatidão, tendo por objetivo evitar distorções na obtenção

dos dados. Entre essas instruções, destacam-se: a solicitação para o preenchimento de todos os itens do questionário e o modo de resposta ao questionário, referindo que apenas devem colocar um sinal (+) se estão mais de acordo do que em desacordo e um sinal (-), no caso contrário, isto é, se estão mais em desacordo do que de acordo com cada um dos itens em análise. O CHAEA em português pode ser acessado gratuitamente na página <http://www.estilosdeaprendizaje.es/chaea/chaegrp2.htm>.

2.5 Sistemas Multiagente

Neste trabalho adotamos o paradigma de agentes inteligentes de software. As pesquisas abordando agentes e Sistemas Multiagente (SMAs) tiveram início nos anos 80, mas começaram a se destacar em meados da década de 90 com o foco voltado para a Inteligência Artificial Distribuída (IAD). O conceito de agentes inteligentes é importante para a área da IAD e Ciência da Computação, sendo definido como sendo um *hardware* ou *software* baseado em um sistema de computador.

Agentes representam um novo paradigma para desenvolvimento de aplicações de software, de tal forma que sua utilização tem acontecido nos mais variados tipos de aplicações, como por exemplo: aplicações na área médica, indústria, entretenimento, educação, dentre muitas outras áreas do conhecimento. Segundo Russel e Norvig (2013), “agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores”. Usando outra definição, Wooldridge (2009) diz que “agentes são sistemas computacionais capazes de ações autônomas em algum ambiente, a fim de alcançarem seus objetivos de projeto”.

Por outro lado, Montufar-Chaveznava e Mendez-Polanco (2005) sustentam as ideias de Russel e Norvig de que os agentes inteligentes, quando empregados em um sistema, devem possuir as características defendidas por Wooldridge:

- **Autonomia** – quando as ações a serem executadas pelo agente dependem somente das suas atribuições e do seu comportamento projetados pelo seu desenvolvedor, o que garante a atuação dele de forma independente do sistema em que estiver atuando;
- **Habilidade Social** – está relacionada à capacidade do agente de interagir com outros agentes em um ambiente através de trocas de mensagens. O SMA depende da cooperação e colaboração entre os agentes para que ele consiga atingir o seu objetivo principal;

- **Reatividade** – ocorre quando os agentes reagem a alguma ação que foi percebida por eles no seu ambiente de atuação, sem demonstrar qualquer preocupação com outras ações passadas;
- **Pró-atividade** – os agentes atuam em um ambiente muito mais que reagindo a ações percebidas, pois eles possuem certa inteligência que os permite inferir ou sugerir ações para o usuário do sistema, baseado em um histórico de registros.

As características citadas acima são importantes para o desempenho de um sistema de informação que utiliza uma sociedade de agentes inteligentes ou um SMA. O processo de implementação de um SMA consiste no desenvolvimento de padrões, envolvendo princípios, conceitos e modelos que permitam a criação de sociedades de agentes inteligentes autônomos. Segundo Netto (2006), estes agentes devem possuir uma auto-definição de suas funcionalidades, de suas características operacionais e do escalonamento de suas tarefas. O autor afirma ainda que integrar diversos agentes inteligentes resulta em melhorias no desempenho do sistema devido a comunicação e a cooperação que ocorre entre eles.

2.6 Arquitetura BDI

A arquitetura (ou modelo) BDI representa uma arquitetura cognitiva baseada em estados mentais, e tem sua origem no modelo de raciocínio prático humano. O nome atribuído ao modelo é justificado pelos seus estados mentais: crença, desejo e intenção (*Belief, Desire, and Intention*). Uma arquitetura baseada no modelo BDI representa seus processos internos através dos estados mentais acima citados, e define um mecanismo de controle que seleciona de maneira racional o curso das ações (Fagundes, 2004).

Segundo Wooldridge (2009), o modelo BDI é particularmente interessante porque combina três componentes distintos:

Componente filosófico: O modelo BDI é fundamentado na conhecida teoria de ação racional em humanos, proposta pelo filósofo Michael Bratman;

Componente de arquitetura de software: O modelo BDI de agentes não prescreve uma implementação específica, ou seja, o modelo pode ser implementado de maneiras diferentes. O fato de que o modelo BDI tem sido implementado com sucesso, é um ponto significativo a seu favor;

Componente lógico: O terceiro componente do modelo é um grupo de lógicas. Estas lógicas capturam os aspectos chaves do modelo BDI como um conjunto de axiomas lógicos.

O conjunto de componentes acima citado torna o modelo BDI um dos mais conhecidos e estudados modelos de raciocínio prático existentes, sendo o componente de arquitetura de software utilizado neste trabalho.

Raciocínio Prático

O raciocínio prático consiste em ponderar considerações conflitantes a favor e contra alternativas competitivas, onde as considerações relevantes são determinadas pelo que o agente crê, importa-se, valoriza e acredita (Bratman, 1990). Não devemos confundi-lo com o raciocínio teórico, o qual é dirigido a crenças.

Como exemplo de raciocínio teórico podemos citar as conhecidas premissas e sua respectiva conclusão: acredito que todos homens são mortais, e acredito que Sócrates é um homem, então concluo que Sócrates é mortal. O processo de concluir que Sócrates é mortal é um raciocínio teórico, pois afetou apenas as minhas crenças acerca do mundo. Como exemplo de raciocínio prático podemos citar o processo de decidir correr ou caminhar, uma vez que tal processo está relacionado a ações.

O raciocínio prático humano consiste em pelo menos duas atividades distintas:

Deliberação: Processo que envolve a decisão de qual estado o agente quer alcançar;

Raciocínio meio-e-fim: Processo que resulta em algum tipo de plano, o qual define como o agente alcançará o estado selecionado no processo anterior.

Para um melhor entendimento das atividades acima citadas, considere o exemplo a seguir. Após um intenso semestre de aulas, um estudante tem muitas alternativas de lazer durante o seu período de férias. Dentre as alternativas, podemos citar viagens, leituras, festas, entre outras. Chamamos deliberação o processo de escolher uma das alternativas. Digamos que o estudante resolveu viajar. Chamamos de raciocínio meio-e-fim o processo que resultará em um plano, possibilitando que o estudante chegue ao local de destino. Neste caso, um plano consiste na compra das passagens, embarque no meio de transporte, e assim por diante.

Estados Mentais

Segundo a psicologia popular, o comportamento humano pode ser previsto e explicado através da atribuição de estados mentais, como por exemplo expectativa, desejo, crença, entre outros. Entidades cujo comportamento pode ser previsto pelo método de atribuição de crenças, desejos e perspicácia de raciocínio, são chamadas sistemas intencionais (Wooldridge,

2009).

A ideia principal dessa abordagem se concentra no fato de que o agente cognitivo possui estados internos que se relacionam com o estado do ambiente com o qual interage. Estes estados seriam correspondentes aos estados mentais humanos, que apresentam um vínculo com o mundo em termos de sua existência e significância.

Segundo (Wooldridge, 2009), atribuir crenças, livre arbítrio, intenções, consciência, habilidade, ou querer a uma máquina é legítimo quando uma atribuição expressa a mesma informação acerca da máquina daquele que é expressa sobre uma pessoa. É útil quando a relação nos ajuda a entender a estrutura da máquina, seu comportamento passado ou futuro, e como consertá-la e aprimorá-la.

Existem outros motivos, além do elevado nível de abstração, para acreditarmos que a modelagem de agentes como sistemas intencionais é útil para o entendimento de programas de computadores (Wooldridge, 2009). Primeiro, e talvez a mais importante, a habilidade de comunicação entre agentes heterogêneos e auto-motivados, a qual implicaria na habilidade de falar sobre seus estados mentais. Segundo, modelos mentais são excelentes candidatos para representar informações sobre usuários finais.

Os estados mentais podem ser classificados em duas categorias: estados mentais de informação e estados mentais pró-ativos. A primeira categoria está relacionada à informação que o agente possui acerca do mundo que ele ocupa, como por exemplo crença e conhecimento. Os estados mentais pró-ativos são aqueles que de alguma maneira guiam as ações do agente, como por exemplo os desejos e as intenções.

A seguir, são apresentados os três estados mentais adotados pelo modelo BDI.

Crença

A crença é um estado mental intencional fundamental para as interações dos agentes, com noção idêntica a de conhecimento, cujo conteúdo externo é uma proposição. As crenças representam o conhecimento sobre o mundo. Em termos computacionais, 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. As crenças são essenciais porque o mundo é dinâmico (os eventos passados precisam ser lembrados), e os sistemas têm apenas uma visão local do mundo (eventos fora da sua esfera de percepção devem ser lembrados) (Georgeff et. al., 1999).

Wooldridge (2009) define as crenças como informações que um agente tem acerca do mundo no qual ele se encontra. Essas crenças podem ser incompletas ou incorretas.

Desejo

Segundo Giraffa (1999), os desejos são relacionados eventualmente ao estado de mundos que o agente quer provocar. Os desejos não dirigem necessariamente o agente a agir, isto é, o fato de um agente possuir um desejo não significa agir para o satisfazer. Significa que antes de um determinado agente decidir o que fazer, ele passa por um processo de racionalização e confronta os seus desejos com as suas convicções. O agente escolherá os desejos que são possíveis de acordo com algum critério.

Os desejos (com frequência imprecisamente chamados de objetivos) formam um essencial componente sobre o estado do sistema. Os desejos representam um estado final que o agente quer verificar. Os softwares convencionais são orientados a tarefa invés de orientados a objetivos (Georgeff et al., 1999).

Intenção

As intenções correspondem aos estados de mundo que o agente quer efetivamente provocar, ou seja, existe um comprometimento em realizá-las. Podem ser consideradas um subconjunto dos desejos, mas ao contrário destes, devem ser consistentes. As intenções são formadas a partir de um processo de deliberação e a partir do refinamento de outras intenções. No entanto, um agente pode conter intenções iniciais inseridas pelo usuário.

Normalmente, o termo intenção é empregado tanto para caracterizar um estado mental quanto para caracterizar uma ação. O estado mental da intenção está direcionado para o futuro e não desencadeará obrigatoriamente uma ação. A ação intencional está direcionada para o presente e representa o ato de agir imediatamente (Wooldridge, 2009).

Wooldridge (2009) aponta como propriedades das intenções:

- **Intenções direcionam o raciocínio meio-e-fim:** Uma vez formada a intenção, um agente deve tentar realizá-la através de um plano. Caso um plano particular venha a falhar, um agente deve tentar outro;

- **Intenções persistem:** Uma intenção deve persistir até que seja realizada, e deve ser abandonada somente se for constatado que não é mais possível realizá-la, ou a razão que a formou deixou de existir;

- **Intenções restringem deliberações futuras :** O agente não selecionará novas

intenções que são inconsistentes com as atuais;

- **Intenções influenciam as crenças sobre as quais os raciocínios práticos futuros são baseados:** Um agente realiza planos futuros partindo do pressuposto que as suas intenções serão realizadas.

2.6.1 Agentes BDI

Discutiremos a seguir alguns aspectos gerais inerentes ao desenvolvimento de agentes compatíveis com o modelo BDI, tais como processos internos, estratégias de comprometimento e reconsideração de intenções.

Processo Deliberativo

Consiste em formar as novas intenções do agente com base nas crenças, desejos e intenções atuais do mesmo. Normalmente, o processo é formado por duas etapas:

- **Geração de opções:** Consiste na escolha de um conjunto de opções (desejos) levando em conta as crenças e as intenções atuais do agente;

- **Filtragem:** Tem como objetivo escolher a melhor alternativa gerada pela etapa anterior, a qual formará a nova intenção.

Raciocínio Meio-e-Fim

É um processo que está associado a ideia de planos, os quais têm como objetivo realizar as intenções do agente. A IA elaborou diversas soluções para o problema de planejamento, como métodos de busca em espaços de possíveis seqüências de ações. Porém, uma das abordagens mais comuns para o problema do planejamento consiste em uma biblioteca de planos prontos, que contém um conjunto de ações aplicáveis a uma determinada situação.

Estratégias de Comprometimento

Quando uma intenção é selecionada através de um processo deliberativo, o agente deve se comprometer com a realização da mesma. A estratégia de comprometimento implica em persistência temporal e especifica quando um agente deve abandonar uma intenção. Podemos classificar as estratégias de comprometimento em:

- **Cega (blind):** Um agente manterá uma intenção até que ela seja realizada;
- **Obsecada (single-minded):** Um agente manterá uma intenção até que ela seja realizada ou que ele chegue a conclusão que não é mais possível realizá-la;

- **Liberal (open-minded):** Uma intenção é mantida enquanto o agente crer que é possível realizá-la.

2.7 Ontologias

Aliado à tecnologia de agentes de software, outro conceito abordado é o de ontologias. Uma ontologia é um modelo de informações representando um conjunto de conceitos num domínio específico, estruturados e inter-relacionados entre si, de entendimento compartilhado em comum por uma comunidade de usuários (Campos, 2007). Conceitos são organizados em hierarquias de classes e possuem atributos e relações entre si. Uma ontologia pode ser representada em linguagem “inteligível” por agentes de software e usada por estes para fazer inferências sobre os conceitos desse domínio.

Ontologias foram propostas no contexto da Web Semântica, uma extensão da Web atual, formada por documentos compreensíveis unicamente por pessoas, para uma Web em que documentos seriam auto-descritíveis, de forma que seu conteúdo possa ser “compreendido” por programas especiais, os agentes inteligentes de software, que assim poderiam “raciocinar” e fazer inferências sobre o conteúdo de documentos, ajudando as pessoas em diferentes tarefas de recuperação de informações que exijam raciocínio, decisões, inferência de conclusões a partir de informações não explicitamente disponíveis ou de informações contextuais. Nas palavras de Berners-Lee a Web Semântica “é uma extensão da Web atual, na qual é dada à informação um significado bem definido, permitindo que computadores e pessoas trabalhem em cooperação” (Berners-Lee; Hendler; Lassila, 2001). Um dos pilares da Web Semântica, juntamente com os metadados e os agentes de software, são as ontologias (Jacob, 2003).

2.8 Sistemas Tutores Inteligentes

Esse trabalho utiliza características de Sistemas Tutores Inteligentes (STIs). Por isso, é importante explicar a respeito deste tópico. As pesquisas na construção de Ambientes Inteligentes para a Aprendizagem Assistida por Computador, também conhecidos como Sistemas Tutores Inteligentes, tiveram início aproximadamente nos anos 70. Alguns autores como Carbonell (1970) e Sleeman e Brown (1982) começaram a observar que os programas de Instrução Assistida pelo Computador (CAI) possuíam estrutura de transmissão de conhecimento de caráter sequencial, previamente determinadas e, por consequência, com incapacidade de adaptação real às necessidades e estilo individual dos alunos. Estes autores propuseram um sistema que incorporava técnicas de IA a fim de tentar criar um ambiente que levasse em consideração os

diversos estilos cognitivos de cada aluno que utilizasse o programa. A este tipo de software denominaram de Tutores Inteligentes ou Sistemas Tutores Inteligentes (Viccari e Giraffa, 2003). Os STIs constituem uma área de pesquisa que objetiva criar sistemas de ensino com habilidades e conhecimentos semelhantes aos dos humanos (Neto, 2006). Sua principal meta é oferecer ao aluno um ambiente personalizado, tanto na forma quanto no conteúdo, superando uma das grandes dificuldades do ensino, seja pelos métodos tradicionais ou através de recursos computacionais, que é o tratamento igual para pessoas com conhecimentos e dificuldades diferentes. Isto fica claro quando Self (1990) diz que “Sistemas Tutores Inteligentes são sistemas de ensino baseados em computador que tentam se adaptar às necessidades dos alunos”.

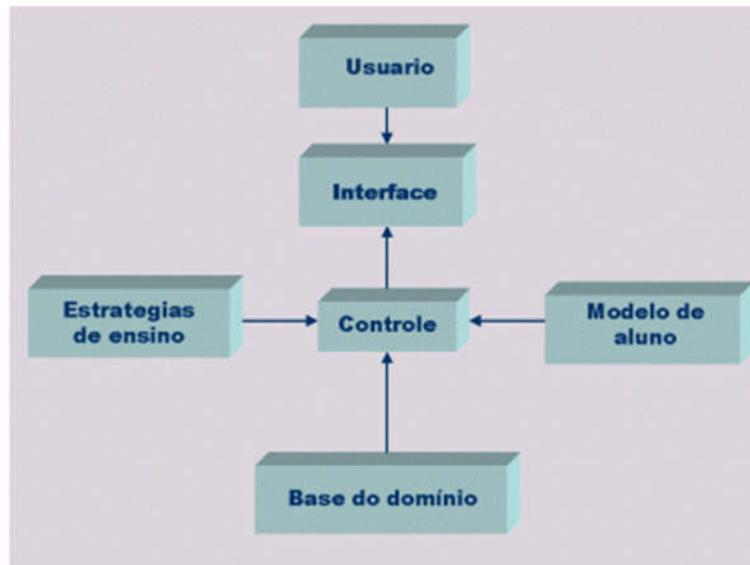
A capacidade de tratar cada um dos alunos que utilizam o sistema de uma forma diferenciada é o aspecto mais importante e que caracteriza um STI, pois, apesar das muitas dúvidas ainda existentes sobre o processo da aprendizagem humana, uma das certezas que já se tem é que as pessoas aprendem de maneiras diversas.

Os STIs são programas que, ao interagirem com o aluno, modificam as suas bases de conhecimento (aprendem), percebem as intervenções do aluno e possuem a capacidade de adaptar as estratégias de ensino de acordo com o desenrolar da interação do aluno com o ambiente. Este tipo de programa de ensino permite que o conteúdo seja gerado sob demanda, além de possibilitar que a interação seja iniciada pelo aluno, através de perguntas, dividindo, assim, com ele o controle sobre a aprendizagem.

Segundo Viccari e Giraffa (2003), os STIs foram projetados como uma tentativa de fazer com que os programas educacionais deixassem de ser meros “viradores de páginas” e se tornassem um elemento mais ativo no processo de ensino-aprendizagem. Para as autoras, a abordagem dos STIs, mesmo que timidamente, vem ao encontro das modernas teorias de educação, que estão trocando o modelo centrado no professor, por um modelo centrado no aluno. Todas essas diferenças devem-se ao fato dos STIs separarem três questões importantes: o que deve ser ensinado; para quem será ensinado e como será ensinado.

A resposta da primeira questão define o tipo de conteúdo a ser ensinado pelo sistema (módulo ou modelo de domínio); a resposta à segunda, define para que tipo de aluno este conteúdo será ensinado (módulo ou modelo do aluno) e, por fim, a resposta para a terceira, indica a maneira pela qual o conteúdo deve ser apresentado (módulo ou modelo do tutor). Essas respostas formaram a base para a pesquisa em STI já na década de 70, e são apresentados como a arquitetura clássica de um STI, conforme Figura 1.

Figura 1. Arquitetura clássica de um STI.



Fonte: Viccari e Giraffa (2003).

A Seção 2.10 define o que é o perfil e o modelo de aluno e os relaciona, mostrando suas semelhanças e diferenças.

2.9 Smart Learning Environments

Quando tratamos do termo *smart learning*, não nos referimos apenas à ideia de melhorar a aprendizagem, mas também consideramos a necessidade de adaptação e personalização em um ambiente educacional, levando em consideração os lugares onde a aprendizagem ocorre (Gros, 2016). Uma das características mais importantes dos *Smart Learning Environments* (SLEs) é que os dados utilizados servem de feedback para que o aluno participe do processo de aprendizagem personalizada (Gros, 2016). De acordo com Zhu et al. (2016, p.3), "não há uma definição clara e unificada de *smart learning* até agora. Pesquisadores multidisciplinares e profissionais educacionais estão constantemente discutindo o conceito".

No entanto, uma tendência comum em muitos ambientes educacionais que são candidatos a SLEs utilizam tecnologias inteligentes, como computação em nuvem, Learning Analytics ou Big Data, determinando a forma como os dados de aprendizagem podem ser capturados, analisados e direcionados para melhorar a aprendizagem e o ensino e apoiar o desenvolvimento de aprendizagem personalizada e adaptável.

Além disso, o desenvolvimento e a implementação de SLEs vai além da aplicação de tecnologias inteligentes. Um SLE não só permite aos alunos acessar recursos digitais e interagir com sistemas de aprendizagem em qualquer lugar e a qualquer momento, mas também

fornece ativamente a orientação de aprendizagem necessária, sugestões, ferramentas de suporte ou recomendação de recursos educacionais no lugar certo, no momento certo e da forma correta. É importante, ainda, que haja uma fusão de aspectos tecnológicos com pedagógicos, a fim de que haja maior aprendizado e engajamento por parte dos alunos (Spector, 2016).

Ainda de acordo com Spector (2016), um SLE deve possuir as seguintes características:

- Conhecimento – acesso a informações pertinentes ao AVA e a habilidade de adicionar ou modificar as informações;
- Apoio a tarefas – a habilidade de fazer uma tarefa ou prover ao usuário ferramentas e informações necessárias para resolver uma tarefa (um recurso);
- Sensitividade ao aprendiz – a habilidade de manter e fazer uso de um perfil do usuário, bem como prover apoio apropriado e conhecimento;
- Sensitividade ao contexto – a habilidade de reconhecer situações específicas, incluindo aquelas situações na qual um estudante pode estar precisando de alguma ajuda;
- Reflexão e feedback – a habilidade para criticar uma solução ou desempenho do estudante e/ou prover feedback significativo e em tempo hábil baseado no progresso do estudante e seu perfil, além de suas respostas às tarefas.

2.10 O Perfil e o Modelo de Aluno em Ambientes Educacionais

No contexto dos sistemas de software adaptativos, segundo Fröschl (2005), um perfil de usuário constitui-se em dados brutos dos usuários do sistema, e do ponto de vista de ambientes educacionais, o perfil de um aluno é uma descrição pessoal de cada aluno participante de um curso. Todos podem preencher ou alterar seus dados no ambiente.

As informações que fazem parte do perfil do aluno podem vir de diversas fontes, como o próprio aluno, a interação do aluno com o ambiente, o professor ou o administrador do sistema. Dependendo do tipo de dados que se têm em um perfil de aluno, poderão ser utilizadas as informações mais relevantes que irão constituir em um modelo de aluno para algum objetivo específico no sistema (adaptação do sistema), como preferências, conhecimento, objetivos e histórico de navegação. Nos sistemas educacionais adaptativos ou STIs, os dados dos alunos são coletados e com isso, os modelos de alunos são criados.

Sendo assim, o modelo do aluno pode ser obtido através do perfil de aluno, cujo modelo é um registro de ações do estudante, além de informações sobre seus históricos. Um modelo de aluno contém as crenças, as informações relevantes, o conhecimento que o sistema

possui sobre o aluno. Este modelo deve ser dinâmico, pois deve refletir também as mudanças no estado cognitivo que ocorrem com o aluno no decorrer das suas interações com o ambiente. Existem diversas formas de definir perfis/modelos de aluno, cada ambiente seleciona uma abordagem conforme as características e objetivos que pretende abranger.

O modelo de aluno, no contexto de STIs, é definido como a representação do conhecimento e das habilidades cognitivas do aluno num determinado momento (Ribeiro, Reategui e Boff, 2007). Segundo Casas (1998), o Modelo do Aluno é “o especialista em técnicas de ensino que seleciona os conceitos, fixa os níveis de dificuldade do ensino e controla o processo de aprendizagem. O modelo do aluno contém as informações relevantes, do ponto de vista do tutor, a respeito do aluno. É a presença deste modelo que permite ao sistema tutor adaptar-se a cada estudante, individualizando a instrução. Este módulo representa o conhecimento e as habilidades cognitivas do aluno em um dado momento. É constituído por dados estáticos e dados dinâmicos que serão de fundamental importância para o tutor poder comprovar hipóteses a respeito do aluno”. Sendo assim, o Modelo do Aluno contém uma representação do estado do conhecimento do aluno no momento que interage com o STI. A partir desse modelo e do conteúdo a ser ensinado, o sistema deve ser capaz de inferir a melhor estratégia de ensino a ser utilizada em seguida. Um modelo realista do aluno implica numa atualização dinâmica, à medida que o sistema avalia o desempenho do estudante. Todas as informações relativas ao aluno e de sua interação com o sistema que compõem o seu modelo são armazenadas em um banco de dados (Liu, 2009).

Além da personalização de conteúdo e do ambiente, os modelos de aluno também são utilizados para reconhecer planos, descobrir caminhos de solução, recomendar conteúdo, analisar desempenho e desvendar habilidades de resolução de problemas. A partir dos dados contidos neste modelo, o AVA define a melhor estratégia de ensino para aquele aluno. Ter um modelo de aluno bem definido implica em uma constante atualização do mesmo, à medida que o sistema avalia a interação do aluno com o AVA.

Definem-se os dados estáticos do estudante como informações que podem ser preenchidas em um formulário pelo próprio aluno antes de usar pela primeira vez o AVA, como: nome, áreas de interesse, preferências, curso e estilos de aprendizagem. Já os dados dinâmicos referem-se ao desempenho do estudante de acordo com as questões formuladas pelo professor e ao uso que faz interagindo com o sistema, como por exemplo, sua quantidade de acessos, o assunto relativo às perguntas que ele elabora no fórum, se o aluno costuma fazer ou não os trabalhos propostos pelo professor e suas competências e habilidades que os alunos vão ad-

quirindo ao longo do uso do AVA ou STI. Como exemplo de dados dinâmicos, para cada área de conhecimento que o aluno pertence podem ser definidos também níveis de suas habilidades e competências face à maturidade do estudante através das atividades realizadas, sendo estes níveis atualizados a cada nova experiência do aluno (indicando que ele estaria mais apto a resolver problemas ou auxiliar seus colegas nesta área), tendo por base também a quantidade de erros e acertos nas atividades propostas. Outra informação dinâmica importante do aluno é o seu grau de sociabilidade no ambiente, ou seja, se ele costuma perguntar frequentemente no fórum ou costuma postar algum arquivo interessante ou ainda, usa o AVA frequentemente. Além disso, deve ser levado em conta se o aluno possui a habilidade de ser proativo (por exemplo, se ele entrega rapidamente os exercícios propostos pelo tutor ou professor).

Existem diversos padrões de modelos de aluno usados em AVAs ou STIs, como IEEE PAPI (2017), IMS LIP (2017), EDUPERSON (2017), Universal Learning Format (ULF) (2017), entre outros. A subseção a seguir descreve o padrão mais conhecido e utilizado atualmente, o LIP.

2.10.1 Modelo de Aluno IMS LIP

Organizações internacionais como Institute of Electrical and Electronics Engineers (IEEE) e Instructional Management Systems (IMS), entre outras, trabalham na padronização de informações de EaD, definindo modelos de dados para informações sobre os alunos, ou seja, modelos de aluno. Sendo assim, neste trabalho adotamos o padrão Instructional Management Systems Learning Information Package (IMS LIP), por oferecer amplo suporte na construção de modelos de aluno e ser o mais usado atualmente (LIP, 2017).

O IMS LIP, definido em 2001 pelo Global Learning Consortium da IMS, é um padrão que possui diversos casos de uso descritos em sua especificação, entre eles da perspectiva individual de padronizar dados de histórico de um aluno em uma aplicação para fins específicos, bem como fazer um gerenciamento de carreira de alunos. O padrão LIP possui esquemas XML para implementação do modelo. As informações do aluno são divididas em 11 categorias, conforme pode ser visto na Figura 2.

As 11 categorias do IMS LIP são:

- *Identification*: contém os dados demográficos e biográficos do aluno;
- *Accessibility*: contém dados sobre a acessibilidade da informação ao usuário, idioma, preferências (físicas e tecnológicas);
- *Goal*: descreve os objetivos e desejos do aluno;

- *QCL*: nesta categoria são descritas as qualificações, certificados e diplomas do aluno;
- *Activity*: qualquer atividade relacionada com estudo, já concluída ou em andamento, está referenciada nesta categoria;
- *Competency*: descreve as habilidades do aluno, experiências e conhecimentos já adquiridos. Essas habilidades podem ser associadas com base no histórico de realizações do aluno, conforme podem ser obtidas com dados das categorias *QCL* e *Activity*;
- *Interest*: contém os interesses do aluno e seus hobbies;
- *Affiliation*: possui informação sobre membros de organizações profissionais;
- *Transcript*: contém um resumo por instituições das realizações acadêmicas do aluno;
- *Security key*: contém as informações de segurança e privacidade relacionadas ao aluno;
- *Relationship*: contém os relacionamentos definidos entre o usuário e sua identificação, acessibilidade, qualificações, competências, objetivos, atividades, interesses, chaves de segurança e afiliações.

Figura 2. Categorias do IMS LIP.

learnerinformation	
contenttype ?	
identification *	competency *
goal *	accessibility *
qcl *	transcript *
activity *	affiliation *
interest *	securitykey *
relationship *	

Fonte: LIP (2017).

Um dos maiores problemas do LIP é apresentar categorias conflitantes e que se sobrepõem, como por exemplo, as categorias *QCL* (Qualificações, Certificações e Licenças), *Activity* (Atividades), *Transcript* (Transcrições) e *Competency* (Competência). É muito comum surgir dúvidas entre os usuários do LIP sobre quais dados pertencem a que categoria, princi-

palmente quando se trata de dados relativos a atividades desenvolvidas, pois esse tipo de informação está presente nas categorias *Performance* (Desempenho) e *Portfolio* (Portfólio).

Nas categorias do modelo IMS LIP foram definidos elementos que em sua maioria são opcionais, além de estruturas de dados que podem ser estendidas. Além disso, o LIP é mais usado atualmente (Bremgartner, 2017a). No padrão LIP é descrito um esquema XML que define os elementos de dados, conteúdo, atributos e um arquivo DTD (Document Type Definition), estabelecendo os elementos e conteúdos que podem ser usados em cada categoria.

Como se pode notar, um dos principais problemas do LIP é o fato dele ser muito extenso, descrevendo muitas categorias de dados que normalmente não serão utilizadas em um único sistema. O que normalmente acontece é que cada sistema utiliza as categorias que consideram mais relevantes. Além disso, uma limitação do LIP é que ele não apresenta mecanismos técnicos para separação dos dados, e isto pode ser um problema quando combinam-se informações de diferentes fontes ou quando se tem a necessidade de separar diferentes tipos de informações por razões de segurança ou integridade. Uma das vantagens do LIP é apresentar um exemplo de implementação em XML, o que demonstra que ele pode ser facilmente utilizado na prática. Além disso, a documentação do LIP oferece guias práticos para implementação do modelo.

2.11 Conclusões do Capítulo

Vimos neste Capítulo os principais conceitos que permeiam este trabalho, utilizados no nosso arcabouço construído, explicado com mais detalhes no Capítulo 4. A seguir, no Capítulo 3 serão apresentados os principais trabalhos relacionados ao nosso tema de pesquisa.

CAPÍTULO 3 – TRABALHOS RELACIONADOS

Nesse capítulo os trabalhos relacionados a esta tese são mostrados. Tais trabalhos confirmam que o tema deste trabalho é amplamente utilizado na literatura.

Nas seções a seguir, serão apresentados os trabalhos relacionados a esta tese, que contemplam os diversos conceitos vistos no Capítulo 2 e nas técnicas de adaptação de recursos em AVAs encontradas na Revisão Sistemática de Literatura apresentada no Apêndice F. Tais conceitos, aliados às técnicas de adaptação de recursos em AVAs, têm sido utilizados para apoiar os processos de ensino e aprendizagem mediados pela tecnologia. É importante ressaltar que a Revisão Sistemática ajudou a embasar a proposta do ArCARE, assim como outras revisões (*ad hoc*) que foram realizadas, muito mais direcionadas de acordo com o tema específico desta tese (incluindo as Arquiteturas Pedagógicas), a fim de atualizar o escopo de trabalhos pesquisados.

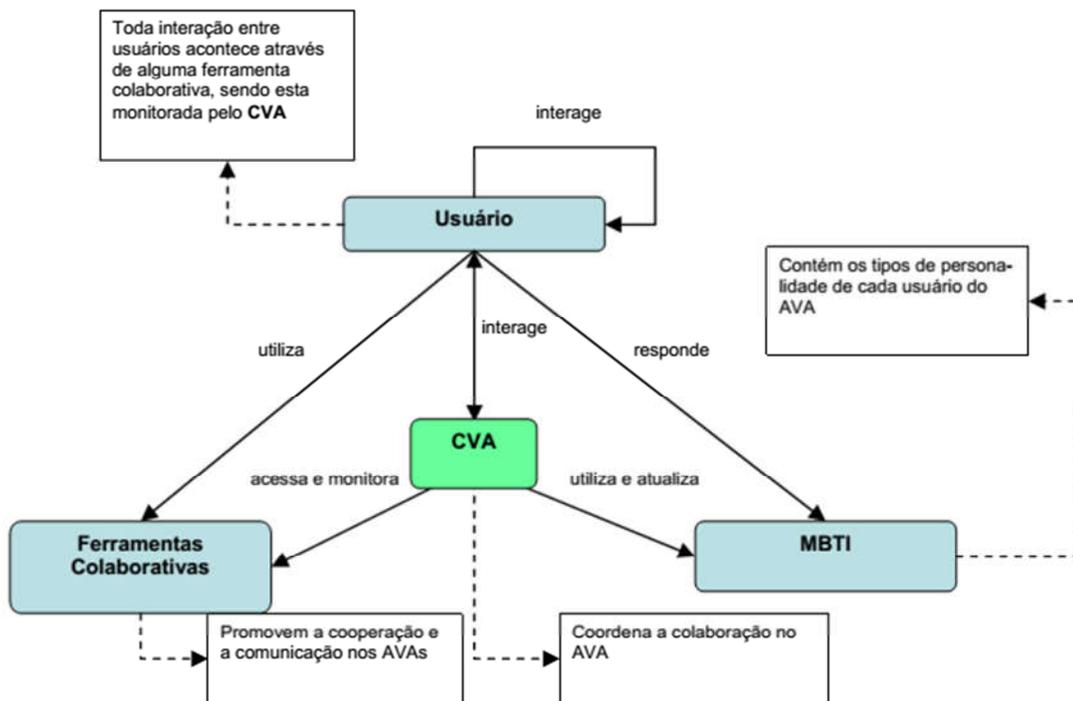
3.1 i-Collaboration

Como forma de superar o sentimento geral de isolamento e conseqüente alta evasão de alunos em AVAs, o trabalho de Oliveira e Tedesco (2009) apresenta os resultados de um experimento com o modelo chamado i-Collaboration (Oliveira e Tedesco, 2010) que promove a colaboração entre os usuários de um AVA. O i-Collaboration baseia-se na utilização de agentes companheiros virtuais de aprendizagem (CVAs).

Nesse trabalho, busca-se propor meios de promover a colaboração entre os estudantes de AVAs através do uso de CVAs integrados à ferramentas colaborativas e da adaptação de interfaces e percepções no ambiente com base nos gostos particulares de cada aluno. O design do i-Collaboration é centrado no usuário, considerando sempre o contexto computacional em que cada estudante se encontra no ambiente. O contexto é descrito neste trabalho como o que está por trás da habilidade de definir o que é ou não relevante em um dado momento, e, neste caso em particular permite que o sistema filtre e dissemine informações mais úteis e adapte seus serviços às necessidades particulares do estudante, provendo recomendações e adaptações em interfaces (para que fiquem mais flexíveis e fáceis de usar). O i-Collaboration permite ainda acesso descentralizado às informações disponíveis no AVA, em acordo com vários

conceitos da Web 2.0. Para decidir a melhor maneira de interagir com os estudantes, o CVA utiliza e atualiza o MBTI (*Myers-Briggs Type Indicator*), que é uma avaliação psicométrica que se utiliza de um questionário para medir preferências psicológicas quanto à forma como as pessoas vêem o mundo e tomam decisões. O MBTI é o teste de personalidades utilizado pelo i-Collaboration para categorizar os estudantes de acordo com suas diferentes personalidades. O CVA pode ainda atualizar os perfis dos usuários com base em seus comportamentos no ambiente. Dado que um estudante tenha inicialmente sido classificado de um tipo de personalidade, porém tenha se comportado diferentemente no AVA, o CVA pode e deve atualizar o perfil deste. A Figura 3 mostra o diagrama, em alto nível, do i-Collaboration.

Figura 3. Diagrama de alto nível do i-Collaboration.



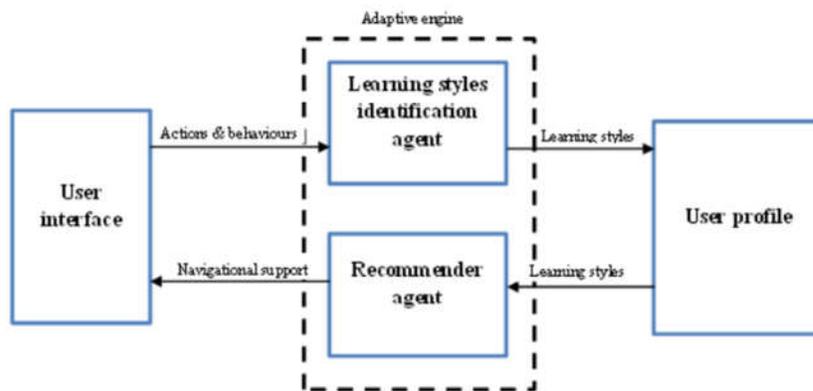
Fonte: Oliveira e Tedesco (2010).

3.2 Modelo Adaptativo de Ambientes Educacionais Abertos Baseado em Estilos de Aprendizagem, de Fasihuddin

O trabalho de Fasihuddin, Skinner e Athauda (2014) apresenta uma proposta de um modelo adaptativo para personalizar o aprendizado em ambientes educacionais abertos, especificamente os Massive Open Online Courses (MOOCs) baseado na teoria dos estilos de aprendizagem, particularmente os estilos de aprendizagem de Felder-Silverman (1988). Este modelo consiste em dois agentes principais para realizar suas funcionalidades. Primeiro, um agente de

identificação, que é responsável de identificar os estilos de aprendizagem dos alunos, monitorando certos padrões de comportamentos dos estudantes ao utilizarem objetos de aprendizagem. Em segundo lugar, o agente recomendador, que é responsável por fornecer um suporte de navegação adaptável com base nos estilos de aprendizagem e preferências identificadas. O trabalho apresenta uma descrição do modelo e suas funcionalidades, mostrados na Figura 4, incluindo os padrões que podem ser monitorados em ambientes abertos de aprendizagem para identificar os estilos de aprendizagem e também como o suporte à adaptação pode ser fornecido com base nos estilos identificados.

Figura 4. Modelo de adaptação baseado em estilos de aprendizagem e agentes.



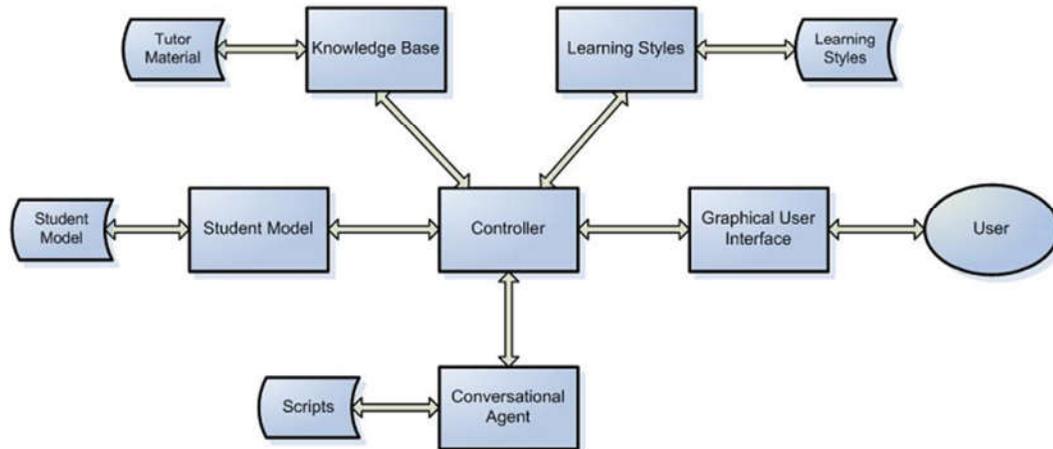
Fonte: Fasihuddin *et al.* (2014).

3.3 Sistema Tutor Inteligente Conversacional Oscar (CITS)

O Sistema Tutor Inteligente Conversacional Oscar (Oscar CITS) (Latham *et al.*, 2010) é um sistema que utiliza uma interface de linguagem natural para permitir aos alunos construir seu próprio conhecimento através de discussões. O Oscar CITS imita um tutor humano, detectando e adaptando-se de forma dinâmica em função dos estilos de aprendizagem dos alunos, durante a conversação com o aluno. Ou seja, é utilizado um agente de conversação para conduzir as aulas. O Oscar CITS pretende imitar um tutor humano por estimativa e adaptar-se dinamicamente ao estilo de aprendizagem de um aluno durante uma conversa de tutoria. O Oscar também oferece análise de solução inteligente e apoio a problemas para os alunos. Ao modelar implicitamente o estilo de aprendizagem do aluno durante a tutoria, o Oscar pode personalizar a tutoria a cada aluno individual para melhorar a eficácia dessa tutoria. O trabalho apresenta uma estrutura, mostrada na Figura 5, para a construção do CITS. O Oscar é também um CITS baseado na web com uma interface de agente conversacional que conduz as aulas e a tutoria, fazendo perguntas, mostrando figuras, filmes e oferecendo um feedback inteligente

para os estudantes. Um estudo piloto inicial foi realizada na tutoria de estudantes de Ciência e Engenharia de graduação utilizando o questionário de Índice de Estilos de Aprendizagem, nos estilos de aprendizagem de Felder-Silverman.

Figura 5. Estrutura do Oscar CITS.

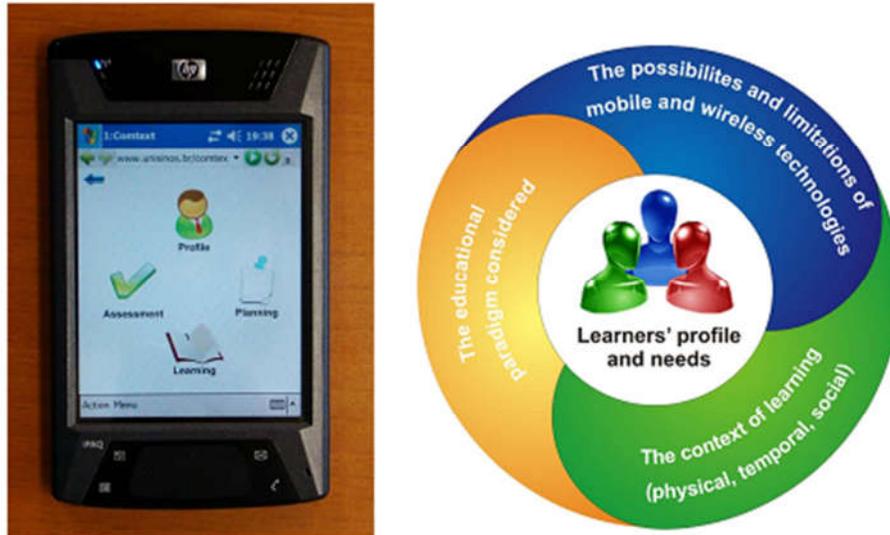


Fonte: Latham et al. (2010).

3.4 COMTEXT

Usando tecnologias móveis, o paradigma educacional considerado no AVA COMTEXT (Saccol *et al.*, 2009) mostrado na Figura 6a) é interacionista-construtivista. As ferramentas dentro do ambiente de aprendizagem COMTEXT têm como objetivo apoiar o desenvolvimento de competências, promovendo a interação entre a comunidade de alunos, com foco no compartilhamento de conhecimento e o desenvolvimento de habilidades e atitudes. Dessa forma, este trabalho apresenta e discute a respeito de um framework mostrado na Figura 6b) que tenta integrar quatro elementos e suas interdependências que são úteis para o desenvolvimento de aplicações educacionais ubíquas: a) perfis e necessidades dos alunos; b) o contexto envolvido nos alunos móveis; c) o paradigma educacional adotado; e d) as possibilidades e limitações das Tecnologias de Informação e Comunicação Sem Fio. Por fim, o framework é aplicado para analisar o processo de desenvolvimento do AVA COMTEXT, projetado para suportar o desenvolvimento de competências de usuários móveis.

Figura 6. a) Principais módulos do COMTEXT b) Framework usado no COMTEXT.



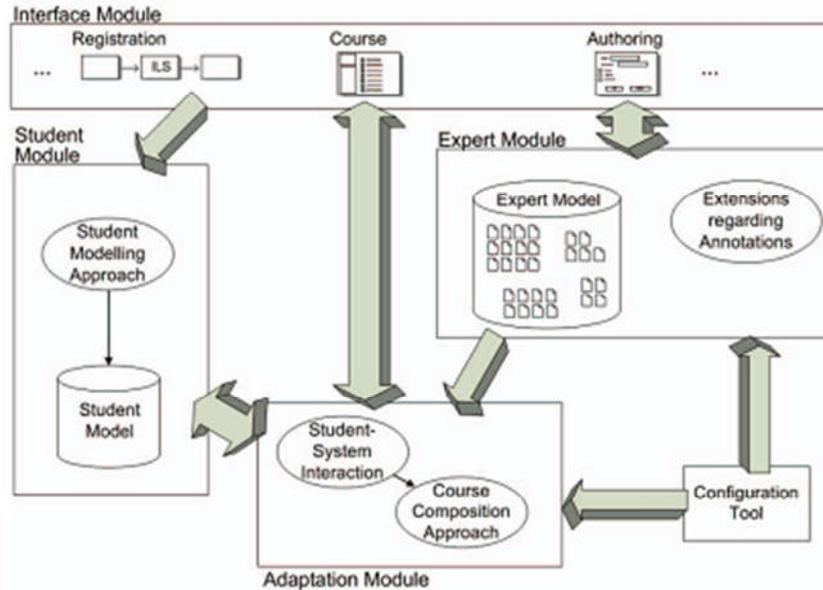
Fonte: Saccol *et al.* (2009)

3.5 Mecanismo de Adaptação de Graf e Kinshuk

Utilizando a abordagem de modelo de aluno, o trabalho de Graf e Kinshuk (2009) introduz um mecanismo de adaptação que é baseado em uma abordagem avançada de modelagem de aluno que identifica os estilos de aprendizagem. Com base nos estilos de aprendizagem identificados, o mecanismo adaptativo compõe cursos que correspondem a esses estilos, com o objetivo de tornar a aprendizagem mais fácil para os alunos. São usados os estilos de aprendizagem de Felder-Silverman (1988). Além disso, o mecanismo adaptativo visa ser fácil de usar para os professores por ser genérico e adaptável, permitindo-lhes ajustar o mecanismo para a estrutura de seus cursos e suas preferências. O mecanismo de adaptação proposto utiliza as informações obtidas a partir da modelagem do aluno para fornecer cursos adaptativos. A Figura 7 apresenta a arquitetura do mecanismo adaptativo integrado em um LMS. O módulo de adaptação é o elemento central do mecanismo adaptativo. Ele é ligado ao módulo do estudante, módulo especialista, módulo de interface e ferramentas de configuração e usa as informações destes módulos/ferramentas para compor os cursos que se encaixam aos estilos de aprendizagem dos alunos. Semelhantemente aos STIs, o módulo especialista armazena os objetos de aprendizagem disponíveis e informações sobre os seus tipos, o módulo de estudante fornece informações sobre os estilos de aprendizagem dos alunos, e o módulo de interface é res-

ponsável por interagir com os alunos. Além disso, a ferramenta de configuração fornece informações sobre os tipos incorporados de objetos de aprendizagem e as características de adaptação que indicam como os diferentes tipos de objetos de aprendizagem devem ser usados para alunos com diferentes estilos de aprendizagem.

Figura 7. Arquitetura do mecanismo de adaptação integrado a um LMS.

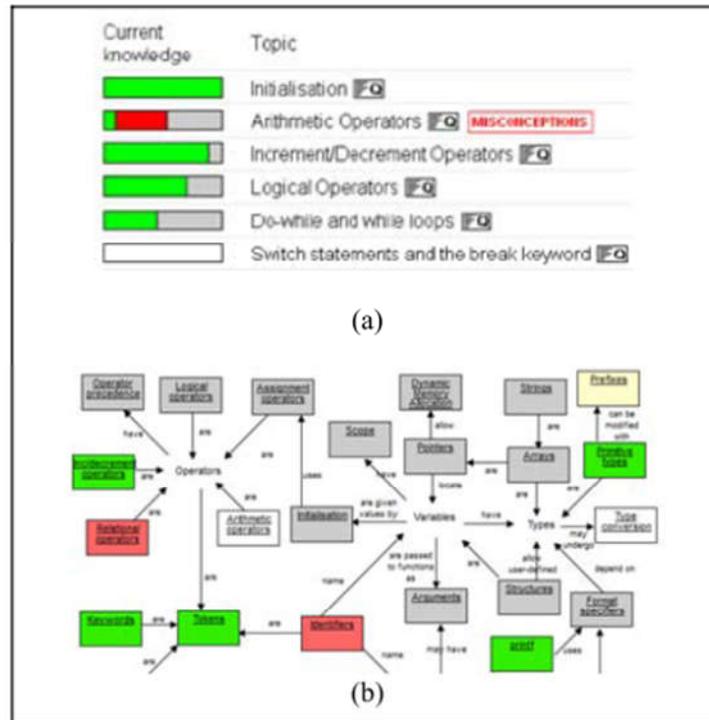


Fonte: Graf e Kinshuk (2009).

3.6 Utilização de Modelo Aberto de Aluno no trabalho de Ahmad

No que diz respeito ao uso de MAA, Ahmad (2013) usa um Modelo Aberto de Aluno simples, apresentando o interesse dos alunos em ver seu modelo de aprendiz e considera o potencial de MAAs como uma abordagem para incentivar o estudo auto-dirigido ao longo da vida entre os estudantes em uma instituição de Ensino Superior na Malásia. Sendo assim, este trabalho descreve um AVA adaptativo que usa MAA (onde a adaptação aqui constitui-se no próprio MAA) e apresenta os interesses dos estudantes em ver seus MAAs. Aqui, pretende-se dar aos alunos um certo controle no seu aprendizado, que pode encorajá-los a serem mais responsáveis e autônomos. A Figura 8 mostra um exemplo de apresentações simples e estruturada de um MAA para alunos.

Figura 8. Exemplos de representação de MAA, com (a) visões simples e (b) estruturada.



Fonte: Ahmad (2013).

3.7 Framework de Martins e Carrapatoso

No trabalho de Martins e Carrapatoso (2008), uma abordagem construtivista para uma Ferramenta Educacional de Hiperídia Adaptativa é apresentada. A plataforma de ensino avalia o conhecimento do aluno e apresenta conteúdos adaptados de acordo com esta característica do aluno, além de seus estilos de aprendizagem de Kolb. Esta plataforma permite que estudantes e professores criem e consolidem de forma autônoma o conhecimento, com feedback automático e apoio constante, por meio de metodologias de ensino e atividades educativas exploradas de uma forma construtivista. A plataforma é baseada na arquitetura do AHA! (Adaptive Hypermedia Architecture) (Wu *et al.*, 1999). O AHA! É um sistema web de hiperídia adaptativa que é apto para desempenhar adaptação que é baseada nas ações de navegação do usuário, conforme pode ser visto na Figura 9 (Wu *et al.*, 1999). A adaptação dessa proposta é baseada na progressiva autoavaliação (exercícios, tarefas, etc...) resolvidas pelo estudante que envolvem uma dificuldade e um tópico. O esquema de atividades é configurado pelo professor, mas é individualizado para cada estudante de acordo com seu nível de conhecimento,

competências, habilidades e percurso de aprendizagem. A plataforma é também conectada a tutoriais que são contextualmente acessados pelos estudantes quando eles falham um passo do percurso. A abordagem construtivista é também no sentido de sugerir algumas referências para o estudante de acordo com a resposta de cada passo (exercícios, tarefas, etc...).

Figura 9. Página inicial do framework.

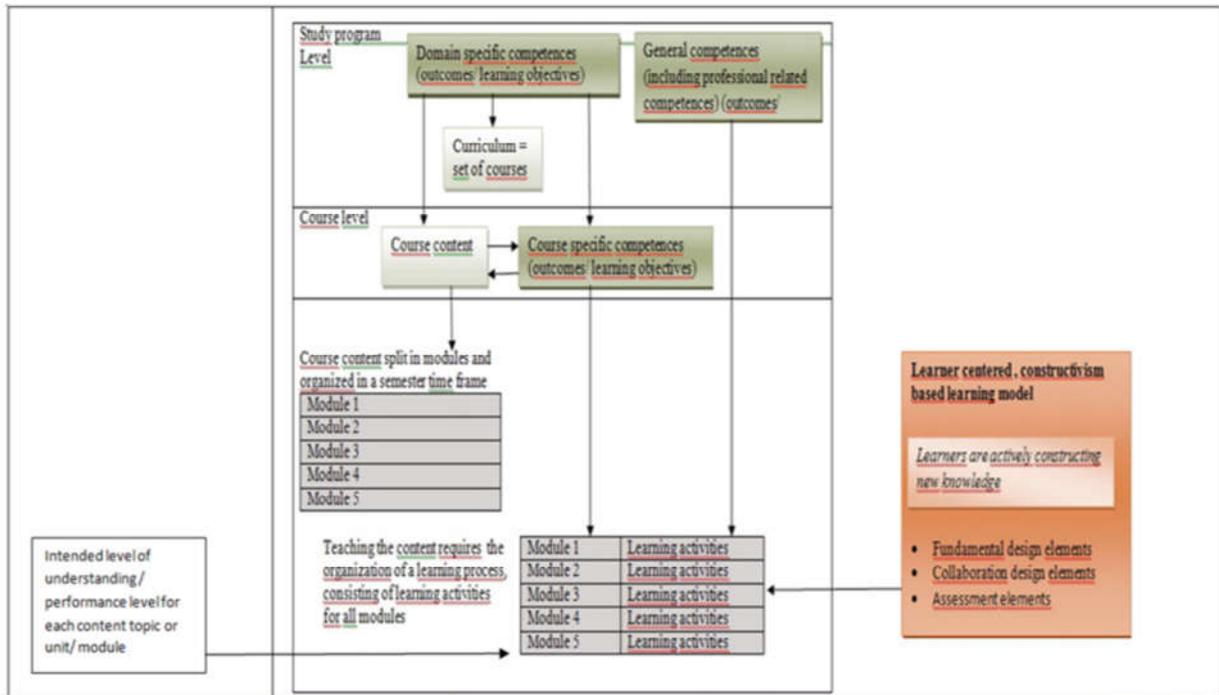


Fonte: Martins e Carrapatoso (2008).

3.8 Modelo de Processo de Aprendizagem Baseado no Construtivismo Centrado no Aluno, de Schreurs e Al-Huneidi

Por outro lado, no trabalho de Schreurs e Al-Huneidi (2012) competências gerais, de domínio e específicas para um curso são definidas com base em uma abordagem construtivista. Para cada módulo do curso o nível requerido de aprendizagem é também definido. Esta metodologia foi construída permitindo que o processo de aprendizagem fosse centrado no aluno. Sendo assim, esse processo começa com um conjunto de competências predefinidas a serem alcançadas pelo estudante no fim do processo de aprendizagem. É considerado que um aluno é competente se ele está apto a mostrar relevantes habilidades e conhecimentos a fim de resolver um problema. Os professores devem definir no curso quais as competências específicas do curso (ou objetivos de aprendizagem). No decorrer do curso, este se adapta de acordo com essas competências que devem ser alcançadas pelos estudantes. Sendo assim, nessa pesquisa foi criado o Modelo de Desenvolvimento de Processo de Aprendizagem Baseado no Construtivismo Centrado no Aluno, mostrado na Figura 10. Há uma integração entre a aprendizagem baseada em competências e a aprendizagem baseada no construtivismo nesse modelo.

Figura 10. Modelo de Desenvolvimento de Processo de Aprendizagem Baseado no Construtivismo Centrado no Aluno.



Fonte: Schreurs e Al-Huneidi (2012).

3.9 Aprendizagem de Programação Apoiada por Computador, de Menezes

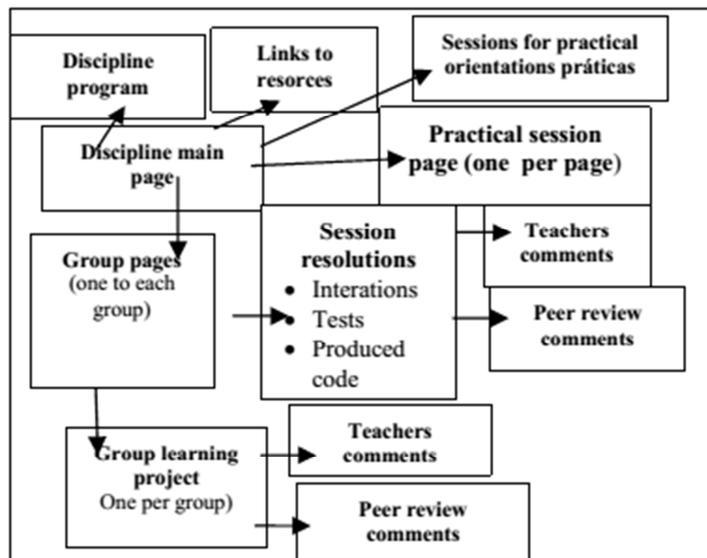
Na revisão sistemática feita, foram encontrados trabalhos que lidam com Construtivismo e aprendizagem colaborativa. Segundo Dillenbourg (Dillenbourg, 1999), a definição amplamente utilizada de "aprendizagem colaborativa" é "uma situação em que duas ou mais pessoas aprendem ou tentam aprender algo juntos." E, em algumas obras que tratam com aprendizagem colaborativa, encontramos o termo "Computer-Supported Collaborative Learning" (CSCL), ou Aprendizagem Colaborativa Apoiada por Computador. CSCL é um campo de pesquisa com foco em atividades de colaboração entre os alunos apoiados por computadores. Com o apoio de redes de computadores, os alunos podem fazer atividades cooperativas de aprendizagem como "educação acompanhada mútua, discussão em grupo, exercícios e projetos em equipe, sem a limitação de tempo e espaço".

A proposta de Menezes et al. (2008) apresenta uma abordagem pedagógica apoiada por um ambiente virtual para melhorar e facilitar a autoria cooperativa de programas de computador na Ciência da Computação. Esse trabalho é focado no ensino da resolução de problemas e utiliza CSCL. O objetivo desse trabalho é apresentar uma proposta pedagógica baseada

em ambientes digitais para melhorar e facilitar a cooperação no desenvolvimento de programas (CSPL – Computer Supported Programming Learning – Aprendizagem de Programação Apoiada por Computador), com o objetivo de formar bons programadores de computador na graduação. Usa-se as TICS para apoiar o ensino e aprendizagem em cursos de engenharia e tecnologia. O desenvolvimento de um programa de computador é, neste trabalho, considerado um processo de resolução de problemas.

Nessa proposta, o processo de resolver problemas no CSPL é composto por 5 etapas: a) entender o problema; b) planejamento; c) desenvolvimento; d) avaliação do processo e seus resultados; e e) socialização dos resultados. A Figura 11 apresenta a estrutura de um site (ambiente) produzido para prover a CSPL.

Figura 11. Estrutura de site.



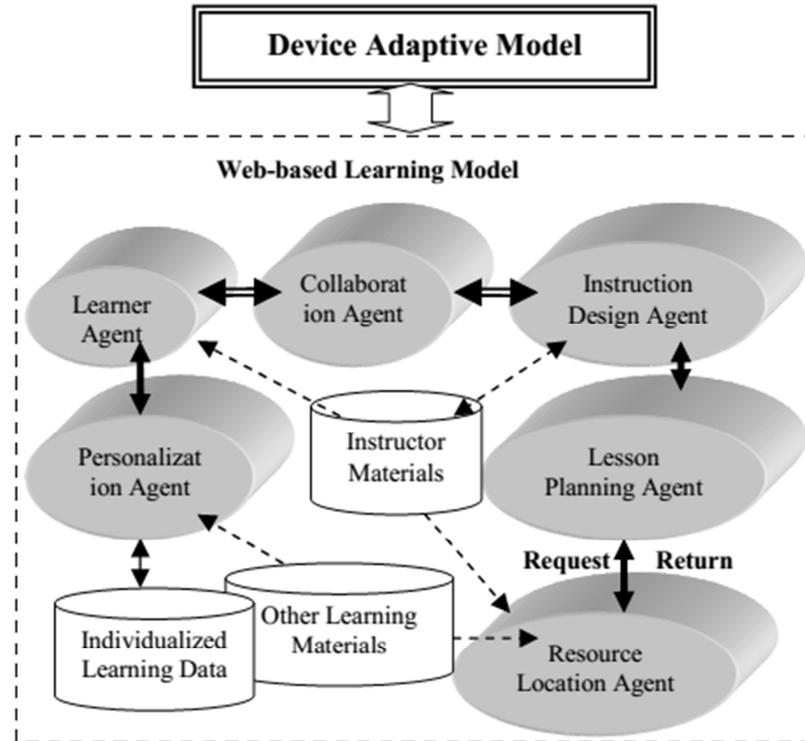
Fonte: Menezes et al. (2008).

3.10 Ambiente Ubíquo de e-Learning de Shaoling e Fangfang

No trabalho de Shaoling e Fangfang (2009) é proposto um ambiente de e-learning que pode ser usado para fornecer aprendizagem personalizada, que utiliza a interação de agentes inteligentes, para obter informações de forma automatizada, planejar aulas, personalizar materiais de aprendizagem e permitir a colaboração entre os instrutores e alunos em uma perspectiva construtivista. Sendo assim, é proposta uma arquitetura de um ambiente de e-learning ubíquo, que é baseada em agentes inteligentes. A Figura 12 ilustra essa arquitetura. Estudantes podem aprender usando computadores (PCs), notebooks, celulares PDAs no ambiente de aprendizagem ubíquo. O projeto conceitual dessa arquitetura inclui o modelo de adaptação do dispositi-

vo e o modelo de aprendizagem baseado na Web. Este determina "o que" e "quando" deve ser feito a fim de entregar entregar materiais de aprendizagem adequados para o estudante.

Figura 12. Projeto conceitual do sistema de e-learning ubíquo.



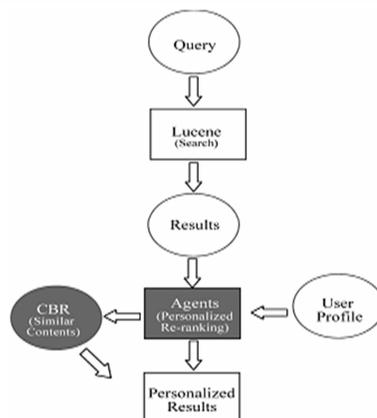
Fonte: Shaoling e Fangfang (2009).

3.11 PULP

Além disso, o trabalho de Ayoola e Phelan (2009) descreve um ambiente de m-learning personalizado, o PULP (Personalised Ubiquitous Learning Platform), para o University College Dublin visando proporcionar a disponibilidade de conteúdo para promover uma aprendizagem construtivista eficiente em qualquer lugar e a qualquer hora para os alunos. O ambiente provê conteúdo sensível ao contexto e oferece acesso tanto através de computadores desktop quanto dispositivos móveis, uma ferramenta interativa, aprendizagem colaborativa, interfaces personalizadas, reutilização de objetos de aprendizagem, recomendação de recursos educacionais (baseados nos interesses dos alunos), e colegas com características semelhantes. A arquitetura do PULP é constituída por 5 camadas, como mostrado na Figura 13, além de seu processo de personalização. O conteúdo é apresentado via interface inteligente adaptativa do usuário.

Figura 13. Arquitetura do PULP e seu processo de personalização.

Social Interactivity	Mails, Messages, Chat, Collaborative Groups
Personalised Service	Monitoring, Searching, Communication, Evaluation, Navigation Support, Content Delivery
Recommendation System	Data, Mediator, Search, Monitoring, Recommendation, Delivery
Data Storage	Profile, Content, Communication, Activities
SCORM	Editor, Importer, Exporter



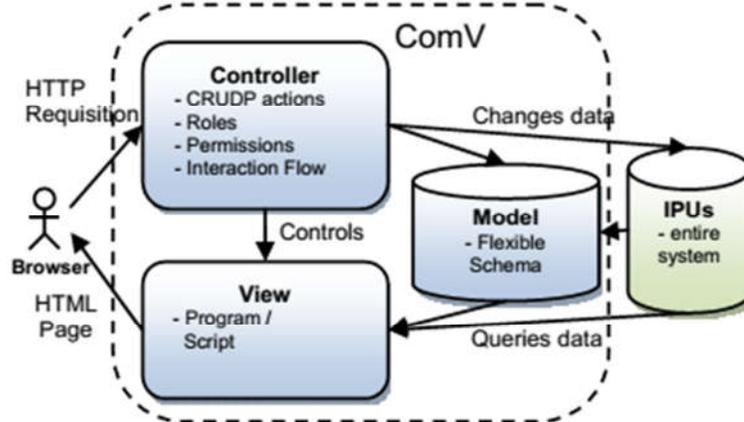
Fonte: Ayoola e Phelan (2009).

3.12 MorFEu

Utilizando arquiteturas pedagógicas, o trabalho de Santos, Castro e Menezes (2012) apresenta uma plataforma para a criação de ambientes virtuais, com flexibilidade para serem combinados e serem definidos ou mudados no decorrer do curso sem perda de dados. Uma plataforma de software foi desenvolvida para modelar e implementar um total de 11 ambientes virtuais conhecidos que vão desde os habituais, como fórum, blog e wiki a exemplos mais complexos, como o debate de teses. Esta plataforma é baseada na perspectiva do projeto MorFEu (Multi-Organizador Flexível para Espaços Virtuais).

Usando elementos simples, o MORFEu pode definir uma grande variedade de ambientes virtuais que podem ser flexíveis e aptos a mudanças em tempo de execução do uso desses ambientes. Tal abordagem se baseia na arquitetura Model-View-Controller (MVC) para os chamados Veículos de Comunicação (ComVs) e as Unidades de Produção Intelectual (IPUs), que são elementos que permitem o registro das atividades dos usuários envolvidos no banco de dados, sem perda de dados no momento de execução dos ambientes e assim, permitindo a flexibilização do conteúdo. Uma representação desses elementos podem ser vistos na Figura 14.

Figura 14. Uma representação Model-View-Controller da plataforma MorFEu.

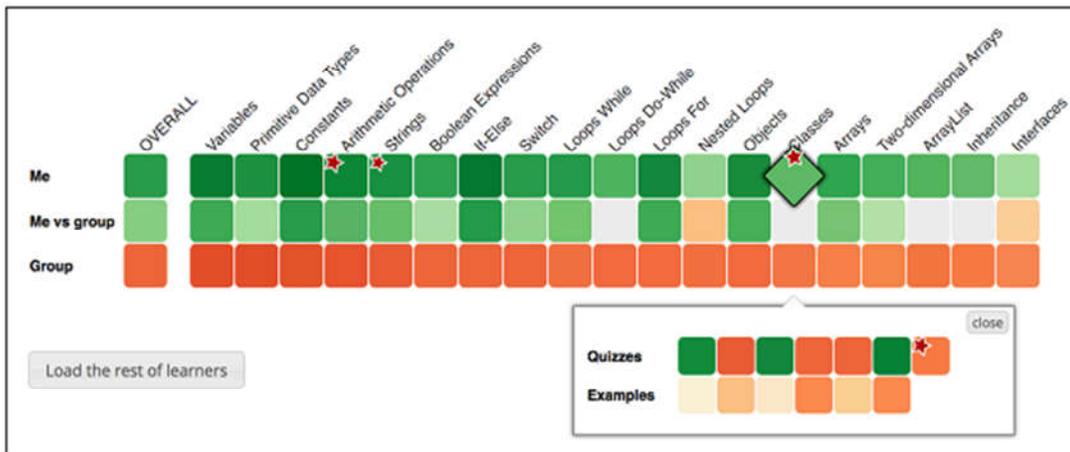


Fonte: Santos, Castro e Menezes (2012).

3.13 Modelo Aberto Social de Aluno (OSSM) de Hosseini, Hsiao, Guerra e Brusilovsky

No trabalho de Hosseini, Hsiao, Guerra e Brusilovsky (2015), os autores exploraram a ideia de combinar orientação social com sistemas de orientação mais tradicionais baseados no conhecimento na esperança de suportar uma navegação de conteúdo mais ótima. O trabalho propôs uma abordagem de sequenciação baseados em algoritmos gulosos (*greedy*) com o objetivo de maximizar o nível de conhecimento de cada aluno e implementá-lo no contexto de uma interface de Modelo Aberto Social de Aluno (OSSM – Open Social Student Model). A Figura 15 mostra uma captura de tela da interface com uma recomendação de conteúdo para um estudante, chamada de Grades de Domínio. O sistema organiza o conteúdo do curso em tópicos, que são exibidos como colunas da grade. A primeira linha mostra o progresso do conhecimento tópico por tópico do aluno usando tons de verde em diferentes densidades; Quanto mais escura a cor, maior o progresso. A terceira linha mostra o progresso agregado do resto dos alunos da classe em tons de laranja. A segunda linha apresenta uma cor diferencial que compara o progresso atual do aluno com o progresso geral da turma.

Figura 15. A apresentação de recomendações no contexto da interface OSSM das Grades de Domínio; uma célula com um símbolo de estrela representa um item recomendado.



Fonte: Hosseini, Hsiao, Guerra e Brusilovsky (2015).

3.14 Conclusões do Capítulo

Nesta Seção, faremos uma análise dos trabalhos relacionados com a nossa proposta. Os ambientes e as propostas apresentadas têm finalidades e arquiteturas distintas, sendo, em sua maioria, voltados para uso na Web, tendo em vista o caráter de personalização e adaptação de recursos apresentados aos usuários que os utilizam sob uma ótica construtivista.

Dos ambientes e trabalhos visitados, destacamos as seguintes características (embora nem todos os trabalhos necessariamente contemplem todas elas):

- Abordagem construtivista da aprendizagem como teoria educacional;
- Utilização de perfil e/ou modelo de aluno;
- Uso de sistemas multiagente para prover as flexibilidades dos sistemas;
- Registro das ações dos usuários, formando seu histórico.

Na Tabela 2, relacionamos os aspectos destacados com os respectivos trabalhos. Na tabela, os números de I a XIII referem-se aos ambientes e propostas visitados, respectivamente, das seções 3.1 a 3.13, sendo: (I) i-Collaboration; (II) Fasihuddin; (III) Oscar CITS; (IV) COMTEXT; (V) Graf e Kinshuk; (VI) Ahmad; (VII) Martins e Carrapatoso; (VIII) Schreurs e Al-Huneidi; (IX) Menezes; (X) Shaoling e Fangfang; (XI) PULP; (XII) MorFEu; e (XIII) OSSM. A marcação "X" indica que foram encontradas, dentro dos trabalhos analisados, evidências da presença do aspecto.

Tabela 2. Aspectos Destacados nos Trabalhos e Propostas Visitados.

Aspecto	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII
Adaptação de recursos	X	X	X	X	X	X	X	X		X	X	X	X
Sistema Multi-agente	X	X	X							X	X		
Perfil/Modelo de aluno	X	X	X	X	X	X		X		X	X		X
MAA						X							X
Sensibilidade ao contexto	X			X						X	X		
m-learning				X						X	X		
Estilos de aprendizagem		X	X		X		X						
Competências				X		X	X	X					X
Arquiteturas pedagógicas												X	
Aprendizagem Colaborativa	X			X					X	X	X	X	
Histórico de ações dos usuários													X
Learning Analytics													X

Fonte: o autor.

Assim, de acordo com a Tabela 2, embora não haja uma homogeneidade entre os trabalhos analisados, podemos perceber que existem vários estudos na literatura que lidam com características construtivistas, aprendizagem colaborativa, modelo de aluno, entre outras características aplicadas aos AVAs. Em casos significativos evidenciou-se a preocupação em dotar os ambientes de facilidades que estimulassem a aprendizagem dos estudantes de forma diferente da estrutura tradicional de AVAs *one-size-fits-all* (que apresenta todos os recursos de aprendizagem da mesma maneira para todos os alunos, de forma estática).

No entanto, observou-se com estes trabalhos descritos acima que, embora muitos deles tratem sobre adaptação de AVAs usando a teoria construtivista da aprendizagem, muitos des-

tes estudos utilizam características específicas dos alunos de forma separada. Em outras palavras, alguns estudos lidam apenas com estilos de aprendizagem, outros com competências dos alunos; outros, apenas dependendo do equipamento que o aluno utiliza, como o dispositivo móvel. Assim, o modelo de aluno não é capaz de informar ao AVA do perfil global do aluno e sua situação atual com maior precisão no ambiente de aprendizagem. Consequentemente, o AVA fica mais vulnerável a tomar decisões erradas na tentativa de auxiliar o aluno em suas necessidades reais. Outros trabalhos lidam com o modelo de aluno, mas falta uma abordagem que estimule a construção do conhecimento do aluno. Outros, que lidam com Arquiteturas Pedagógicas, necessitam de uma adaptação mais proativa dos recursos de aprendizagem apresentados ao aluno (ou até mesmo das próprias APs).

Entretanto, é bom salientar que os sistemas visitados contêm aspectos que combinados podem formar a base da solução do arcabouço conceitual pretendido nesta tese. O ArCARE difere ao considerar várias características dos estudantes ao mesmo tempo (habilidades, interesses, estilos de aprendizagem), variando de acordo com o histórico de interações dos alunos com o AVA e também ao apresentar o modelo de aluno para o mesmo, sendo um MAA, a fim de obter um diagnóstico mais preciso da situação dos alunos no ambiente educacional e estimulando-os a aprender cada vez mais, além de proporcionar adaptação de conteúdo e recursos em AVAs, em função de seu modelo de aluno utilizando uma abordagem construtivista. O Capítulo 4 a seguir descreve o ArCARE com mais detalhes, onde apresentamos uma estrutura que concilie os importantes aspectos aqui destacados com as necessidades específicas do projeto.

CAPÍTULO 4 – O ARCABOUÇO CONCEITUAL DE ADAPTAÇÃO DE RECURSOS EDUCACIONAIS (ArCARE)

Este capítulo tem por objetivo apresentar o Arcabouço Conceitual de Adaptação de Recursos Educacionais, o ArCARE, a partir da análise dos cenários descritos no Capítulo 1. Suas características serão explicadas em detalhes.

Após a análise dos trabalhos relacionados no Capítulo 3, nesse Capítulo é descrito um arcabouço de sistemas para adaptação de recursos educacionais em ambientes de aprendizagem com o objetivo de adaptar tais ambientes de acordo com o perfil do estudante. Nesta tese, convencionamos que os termos “arcabouço conceitual” e “framework conceitual” possuem o mesmo significado. Um framework (ou arcabouço) conceitual é um conjunto de conceitos usados para resolver um problema de um domínio específico (Luck e d'Inverno, 2001). Sendo assim, o objetivo central do arcabouço proposto, o ArCARE, é definir a estrutura e as funcionalidades necessárias a um sistema capaz de adaptar e/ou recomendar recursos apresentados a alunos em ambientes educacionais. Portanto, a partir da análise dos cenários descritos no Capítulo 1, são definidas as linhas gerais de desenvolvimento de sistemas ou ambientes computacionais que podem ser criados a partir do arcabouço, como: a arquitetura de referência do arcabouço, as operações básicas que os sistemas devem prover e como implementá-las.

4.1 Análise dos Cenários

Nesta seção faremos uma análise dos cenários apresentados no Capítulo 1. Em seguida, procuramos elementos para definir o ArCARE. A análise dos cenários visa encontrar os elementos principais no processo de recomendação de recursos educacionais e caracterizar as relações entre eles. A Tabela 3 apresenta um resumo dos cenários discutidos no Capítulo 1, contendo suas características e problemas que podem ocorrer em cada um deles.

Tabela 3. Resumo dos Cenários do Capítulo 1.

Cenário	Características	Problemas
1. Alunos com dúvidas em um AVA	- Alunos interagindo no AVA;	- Falta de um mecanismo de monitoramento automático

	<ul style="list-style-type: none"> - Perfil do aluno sendo atualizado mediante interações do aluno; - Professor elabora atividades, recursos, podendo acompanhar o desempenho do aluno; - Currículo do curso no formato padrão (formal) 	<ul style="list-style-type: none"> dos os alunos; - Falta de um auxílio personalizado automático a alunos com dúvidas e de acordo com seu perfil.
2. Alunos em um curso que não segue um currículo formal	<ul style="list-style-type: none"> - Alunos interagindo no AVA; - Perfil do aluno sendo atualizado mediante interações do aluno; - Professor elabora atividades, materiais; - Currículo do curso em um formato mais flexível (informal), uma vez que os estudantes podem escolher Unidades de Ensino para estudarem, sem uma ordem pré-definida; - Estudantes de diversos níveis de conhecimento e faixas etárias. 	<ul style="list-style-type: none"> - Falta de ajustes de recursos mostrados aos alunos dependendo do seu nível de conhecimento; - Falta de um mecanismo de monitoramento automático dos alunos; - Falta de um mecanismo de adaptação de recursos no ambiente, como sugestões do que o estudante poderia estudar para atingir certos níveis de competências e habilidades.
3. Comunidade de prática em uma empresa	<ul style="list-style-type: none"> - Funcionários em uma empresa registram suas atividades em um ambiente e trocam experiências, alimentando uma base de conhecimento a respeito de suas informações; 	<ul style="list-style-type: none"> - Necessidade de relatórios otimizados a respeito de desejáveis características dos funcionários; - Como detectar de forma automática que algum funcionário está tendo alguma

	<ul style="list-style-type: none"> - Os chefes na empresa obtêm relatórios a respeito dos trabalhos realizados e da produtividade de seus funcionários; - Ocorre uma comunidade de prática, permitindo interações entre funcionários para resolver problemas de uma empresa. 	<ul style="list-style-type: none"> dificuldade em realizar uma tarefa na empresa? - Necessidade de selecionar de forma automática qual funcionário pode realizar determinado trabalho ou resolver um problema específico na empresa; - Falta de um mecanismo de recomendação de estratégias e ferramentas que funcionários podem utilizar para resolver determinados problemas.
4. Usuários utilizando dispositivos móveis	<ul style="list-style-type: none"> - Estudantes de diferentes níveis de experiência com o uso de dispositivos móveis 	<ul style="list-style-type: none"> - Falta de um mecanismo de ajustes dos ambientes educacionais de acordo com o nível de familiaridade com o equipamento que o estudante usa;
5. Recomendação e adaptação de recursos sensíveis ao contexto	<ul style="list-style-type: none"> - Adaptação de aulas de acordo com os elementos de contexto para a aula (equipamento que alunos e professores utilizam, local do aluno, experiência do aluno em determinados assuntos) 	<ul style="list-style-type: none"> - Falta de padronização na identificação de características relevantes para a adaptação, por exemplo, o lugar onde o aluno se encontra - Falta de um mecanismo de preparação automática de aulas dependendo do contexto do aluno, do equipamento e do curso ou disciplina em questão.
6. Colaboração em um grupo de pesquisa	<ul style="list-style-type: none"> - Estudantes de diversos níveis de escolaridade, habili- 	<ul style="list-style-type: none"> - Falta de um mecanismo de recomendação automática de

	dades específicas, interesses no grupo interagindo em um AVA e fazendo registros de suas atividades, como publicações e portfólios de aprendizagem a respeito do que eles têm aprendido;	colegas nesse grupo para ajudar aqueles que solicitarem ajuda em um tópico; - Como avaliar se algum pesquisador/aluno nesse grupo detém conhecimentos em determinadas áreas? Quais parâmetros seriam necessários para tal avaliação?
--	--	---

Fonte: o autor.

Os seis cenários apresentados na Tabela 3 têm em comum as seguintes características:

- Há usuários utilizando um ambiente virtual, podendo ser um AVA, e há conteúdos e recursos sendo apresentados a esses usuários;
- Compreendem situações em que é importante obter e analisar o perfil do usuário envolvido por meio de seu histórico de ações, além de atualizá-lo;
- Existem usuários que possuem variadas habilidades específicas ao executarem atividades;
- Há uma interação entre os usuários do sistema (alunos, professores, funcionários);
- Os usuários podem acessar os ambientes com qualquer dispositivo ou equipamento que quiserem, além de poderem acessar em qualquer tempo e lugar.

No entanto, observando os cenários na Tabela 3, constatamos que:

- Há uma alta dependência de que os usuários resolvam manualmente seus problemas e suas atividades: todos os ambientes dependem de pessoas para dirimirem dúvidas, iniciarem processos e trocarem informações entre si;
- Há um problema de *one-size-fits-all* nesses ambientes: o que é mostrado para um usuário qualquer é mostrado para um usuário de habilidades e competências de maiores níveis, tornando a disponibilidade de recursos no ambiente de forma estática;
- Em situações relevantes as pessoas precisam estar no mesmo lugar e no mesmo momento para interagirem; alguns ambientes apoiados por computador favorecem interações à distância, mas ainda necessitando da presença síncrona dos participantes;

- Nos ambientes usados hoje em dia, há uma dificuldade em deduzir de maneira automática o que cada usuário está precisando para resolver um determinado problema, ou mesmo precisando para aprender o conteúdo de uma disciplina ou área de conhecimento, resultando em problemas não solucionados, ou em se ter soluções não satisfatórias e até mesmo não obtermos uma aprendizagem significativa por parte dos usuários do ambiente virtual.

4.2 Caracterização do ArCARE

Partindo dessa análise dos cenários realizada na Seção 4.1, levantamos as principais características necessárias para que seja permitida a adaptação de recursos em função das características do usuário e chegamos à Tabela 4, que forma a base do que chamamos de Arca-bouço Conceitual de Adaptação de Recursos Educacionais (ArCARE).

Tabela 4. Comparação entre a Situação Atual dos Cenários e a Situação Proposta no ArCARE.

	Situação Atual	Situação Proposta
I	Dificuldade de alunos tirarem suas dúvidas em AVAs	Recomendação de colegas para ajudarem os alunos com dúvidas; recomendação de atividades colaborativas
II	Conteúdos e recursos apresentados da mesma maneira a todos os usuários, de forma estática	Considerar o perfil do usuário para apresentar recursos ou conteúdos em ambientes virtuais, além de permitir a autorreflexão do aluno quanto a sua situação no ambiente educacional, por meio de um Modelo Aberto de Aluno
III	Dificuldades de interação entre estudantes em AVAs	Fazer o uso de ferramentas que estimulem a aprendizagem colaborativa dos estudantes
IV	Modelo tradicional de ensino nos AVAs: dispor conteúdo seguindo um currículo formal para os estudantes, mesmo que haja alguma adaptação	Permitir o uso de currículos flexíveis (informais) e cursos não convencionais, a fim de que o aluno não fique limitado a um curso previamente programado
V	Falta de maior integração entre equipamentos dos alu-	Considerar um modelo de aluno que contemple todas estas características, a fim de prover uma

	nos, habilidades dos alunos, estilos de aprendizagem para recomendar recursos adequados a eles	recomendação de recursos mais precisa considerando os diferentes contextos do estudante
VI	Falta de um mecanismo de descoberta das mudanças que podem ocorrer nos perfis dos alunos, suas interações com os ambientes	Dotar os ambientes de um mecanismo que permita obter o histórico das interações dos estudantes, a fim de obter inferências mais precisas a respeito deles, por meio, por exemplo, de aprendizagem de máquina
VII	Falta de um mecanismo de descoberta dos melhores recursos recomendados	Dotar os ambientes de um mecanismo que permita obter o histórico das utilizações dos recursos, a fim de utilizar esses dados de histórico para otimizar os processos de recomendações futuras de recursos, inclusive em outros cursos, permitindo seu reuso.

Fonte: o autor.

Além dessas soluções propostas na Tabela 4, no nosso framework conceitual adotamos as diretrizes propostas por Spector (2016), uma vez que o ArCARE deve permitir a construção de *Smart Learning Systems* (ou *Environments*) – SLEs – ou mesmo tornar um AVA com características de SLE, definidas na Seção 2.9. Portanto, tomando cada solução proposta apresentada na terceira coluna da Tabela 4, as diretrizes descritas na Seção 2.9 e as conclusões obtidas do Capítulo 3, apresentamos na Figura 16 os elementos que constituem um modelo de referência que caracterizam o ArCARE e suas dimensões nas quais eles atuam.

Figura 16. Modelo de Referência do ArCARE.

Conhecimento: uso de Modelo Aberto de Aluno (MAA), composto por estilos de aprendizagem, interesses, competências e habilidades e desempenho do estudante em atividades, manipulação de dados do MAA, informações a respeito do aluno e outros usuários do ambiente, acesso e atualização de informações de forma automática
Apoio a tarefas: informações dos Cursos, das Unidades de Ensino e dos Recursos, estímulo a aprendizagem colaborativa, ajuda aos estudantes,

como: aviso sobre o deadline de uma atividade; adaptação e recomendação de recursos para o aprendiz realizar uma certa atividade ou vencer uma certa dificuldade; recomendação de pessoas que possam sugerir alternativas de ações para o aprendiz resolver alguma dificuldade

Sensitividade ao Aprendiz: Histórico de ações dos usuários, acompanhamento do usuário em sua trajetória pelo curso, verificar interações entre usuários no ambiente, descobrir suas necessidades, informações sobre pessoas da comunidade de aprendizagem e seus interesses

Sensitividade ao Contexto: Avaliação do estado atual do usuário no ambiente em função do seu estado em diferentes cenários, informações sobre as atividades disponíveis em certo momento

Reflexão e Feedback: Apresentar os recursos recomendados, adaptados e o MAA ao usuário

Fonte: o autor.

A Seção 4.3 a seguir apresenta a definição do ArCARE e sua arquitetura de referência.

4.3 A Definição do ArCARE

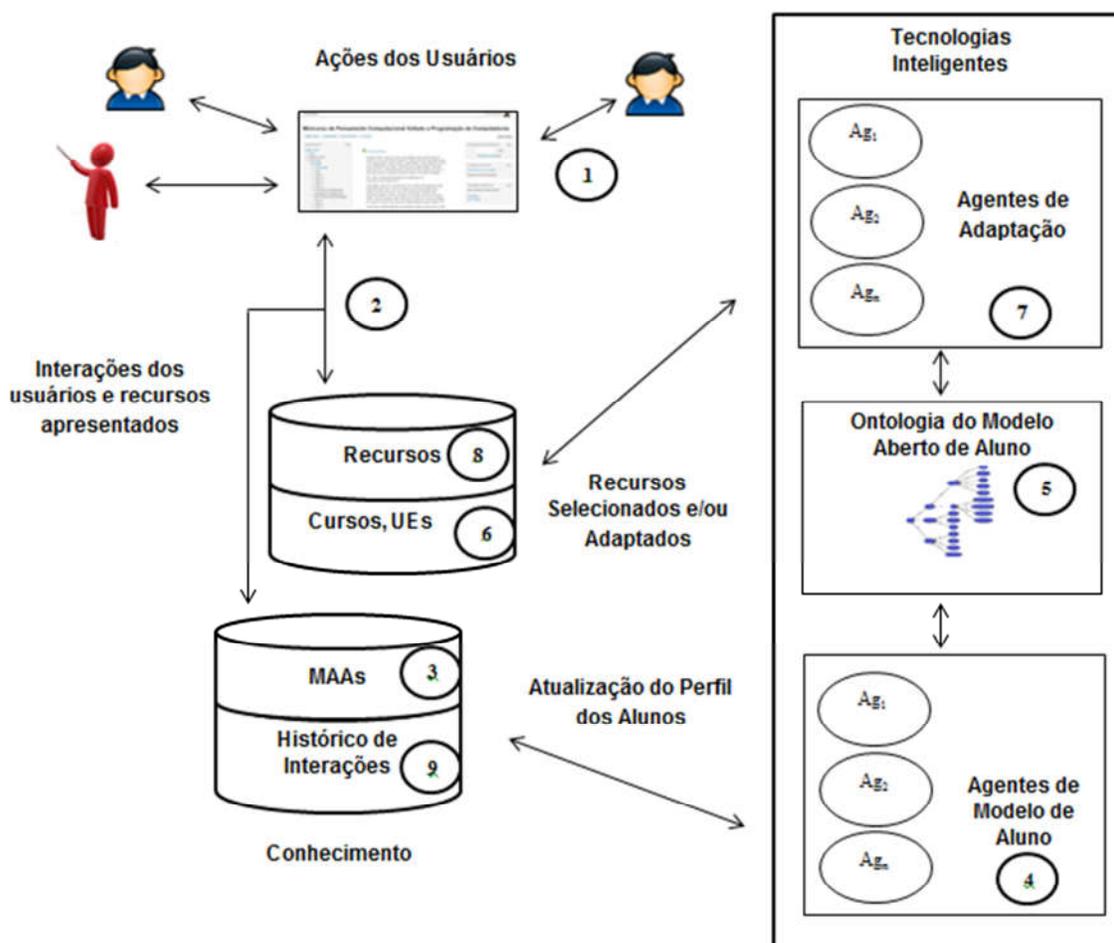
A arquitetura de referência contendo os elementos que constituem o ArCARE, com base no modelo de referência da Figura 16, é mostrada na Figura 17. Nesta arquitetura, é mostrado em maiores detalhes o arcabouço que constitui no processo de personalização do AVA por meio da adaptação e recomendação de recursos educacionais, em função das características do aluno, fazendo uso de agentes de software e uma ontologia que descreve o MAA. É composto de três componentes fundamentais: **tecnologias inteligentes, conhecimento e espaço de ações do usuário**.

O módulo de tecnologias inteligentes contém toda a camada de inteligência fornecida pelo ArCARE. É composto por dois grupos de agentes: os que controlam e manipulam o modelo do aluno e os agentes que são responsáveis pela adaptação de recursos do ambiente educacional, a partir dos dados do aluno. O módulo de tecnologias inteligentes também conta com a Ontologia de Modelo Aberto de Aluno.

O módulo de conhecimento contém todos os dados a respeito do ambiente educacional, por exemplo, um Banco de Dados (BD) do AVA. Destacamos na arquitetura da Figura 17 que este módulo possui informações a respeito do histórico de interações dos usuários, informações dos MAAs, recursos, cursos e suas Unidades de Ensino (UEs). É sobre essa área de conhecimento que o módulo de adaptação de recursos atua.

Por sua vez, o espaço de ações dos usuários refere-se às interações dos usuários com o ambiente educacional. Tais ações são registradas no módulo de conhecimento.

Figura 17. Arquitetura de referência do ArCARE.



Fonte: o autor.

Cada elemento apresentado na Figura 17 (com uma numeração para identificação) faz parte do processo de adaptação e recomendação de recursos educacionais. Nesse arcabouço definimos algumas premissas. Primeiramente, consideramos que os usuários (estudantes, professores, administradores do ambiente) estão sempre em interação com o AVA (1). O estudante acessa o AVA, onde pode utilizar os recursos apresentados a ele, atualiza seus dados cadastrais, executa atividades propostas pelo professor, acessa o seu MAA, a fim de saber seu desempenho no decorrer do curso e fazer autorreflexões. MAAs permitem aos usuários (estudantes, professores, pares e/ou outros *stakeholders* no processo educacional) verem o conteúdo do modelo do aluno de um STI, SLE ou qualquer outro ambiente educacional de forma entendível por seres humanos (Bull e Kay, 2016). Usar MAA refere-se a tornar o modelo de aluno explícito ao estudante, de modo a fornecer um recurso adicional através do autoconhe-

cimento e possível autorregulação do processo de aprendizagem com o objetivo de melhorar a aprendizagem e autonomia do aluno. De acordo com Bull, Quigley e Mabbott (2006), os objetivos do MAA são incentivar a reflexão, a aprendizagem independente e a avaliação formativa/acompanhamento dos progressos.

O estudante pode escolher também que parâmetros seus podem ser usados para o processo de adaptação (podendo escolher adaptações apenas por estilos de aprendizagem, competências ou interesses, alguma combinação entre essas variáveis ou até nenhuma adaptação). Isso evitaria o que chamamos de **adaptação forçada**. O professor pode elaborar cursos, **Unidades de Ensino (UEs)**, conforme em (6), atividades, lançar notas, inserir recursos no repositório de recursos (8) do AVA que ficam salvos no BD, além de executar outras ações conforme suas atribuições. Já os administradores são os responsáveis pela configuração do ambiente e outras atribuições que lhe são designadas no próprio AVA.

No nosso arcabouço conceitual definimos que o aluno precisa ter o seu perfil inicial, ou seja, os dados que compõem seu modelo inicial, a fim de que o ambiente possa começar a ser adaptado, ao invés deste ter que esperar por diversas manifestações do estudante. Para isso, as primeiras interações dos estudantes com o AVA são registradas (2), tais como: questionários para identificação de estilos de aprendizagem, testes iniciais para diagnóstico de competências do estudante e questões pessoais de acordo com o curso que ele estiver participando, a fim de inicializar seus níveis de competências e habilidades, mesmo que essas sejam atualizadas posteriormente, diferentemente dos estilos de aprendizagem. Além disso, seu histórico de utilização do ambiente em cursos passados é obtido (9), caso o estudante já tenha usado o AVA anteriormente.

De posse dessas informações, começa a ser formado o MAA (3), que é atualizado a cada interação do estudante com o AVA, por meio dos agentes que manipulam esse modelo, os **agentes de modelo do aluno** (4). Ao manipularem os dados dos alunos, estes agentes utilizam uma ontologia que descreve o seu modelo, a **ontologia de modelo aberto do aluno** (5), que contém regras para trocas de mensagens entre agentes, além de definições e regras que fazem parte do modelo do aluno. Além disso, no MAA, temos o **modelo do recurso**, que em linhas gerais, consiste nos dados mais relevantes a respeito dos recursos a serem utilizados no processo de adaptação/recomendação no AVA. Esse modelo do recurso, assim como a ontologia de MAA, são descritos com mais detalhes na seção 4.4.

Por sua vez, os **agentes de adaptação de recursos** (7) permitem a adaptação e seleção de recursos (8) que estão armazenados no BD do AVA e que consideram mais adequados para cada estudante no curso ou disciplina, utilizando os dados do MAA (3) e a ontologia do modelo do aluno (5). Por fim, esses recursos são apresentados aos alunos para estes utilizarem no AVA (1). Percebe-se que o processo de recomendação de recursos pode ser visto como um processo cíclico e contínuo, uma vez que novos recursos são apresentados cada vez que o modelo de aluno é atualizado.

4.3.1 Caracterizando as Unidades de Ensino no ArCARE

Aqui apresentamos, de uma forma geral, como podem ser os cursos em que o método de adaptação de recursos provido pelo ArCARE pode ser utilizado. O ArCARE permite que as adaptações de recursos sejam em cursos tradicionais, formais, mas também em cursos flexíveis, ou seja, que não seguem um currículo predeterminado ou predefinido. Sendo assim, em um curso flexível, o professor pode criar seu curso ou disciplina, definindo as **Unidades de Ensino (UEs)**, que são os blocos ou tópicos com o conteúdo da disciplina. Um recurso também pode ser uma UE.

Cada curso flexível pode ser composto por k UEs, onde cada UE possui n competências ($C_{k1}, C_{k2}, \dots, C_{kn}$) ou habilidades associadas à mesma (onde espera-se que o aluno alcance essas competências ou habilidades ao término da UE), seu conteúdo apresentado e uma ou mais avaliações. A respeito das competências associadas às UEs, cada UE de um curso só pode ser mostrada a um aluno se este possui os níveis mínimos de competências e habilidades necessárias para a apresentação de tais UEs, exceto se o professor permitir a livre apresentação de UEs a qualquer aluno. Os alunos podem começar o curso por qualquer UE disponível, segundo suas livres escolhas.

É importante lembrar que o papel do professor, em todo esse processo, consiste na montagem do curso ou disciplina no AVA, elaborar as atividades, verificar o estado do aluno no ambiente e ser o responsável por definir e executar o currículo do curso e as etapas do curso. O professor também é o responsável por permitir, caso queira, que o aluno refaça ou não as atividades já passadas, no caso de o aluno ter errado essas atividades passadas anteriormente.

Além disso, o professor, assim como o administrador do ambiente, pode inserir os recursos no ambiente com seus metadados preenchidos, que compõem o **modelo do recurso** (estilos de aprendizagem, competências, interesses de alunos, entre outros dados associados,

formando um **repositório** ou **catálogo de recursos**), podendo editá-los a qualquer momento e selecionar que competências e habilidades estão associadas a cada recurso (e seus níveis normalizados, por exemplo, numa escala de 0 a 10).

O professor também configura o curso indicando todas as competências e habilidades envolvidas neste, a fim de que o estudante saiba quais são as competências e habilidades do curso e que ele possa ver o seu alcance de competências e habilidades no decorrer do curso, pela apresentação de seu MAA.

4.4 O Módulo de Tecnologias Inteligentes no ArCARE

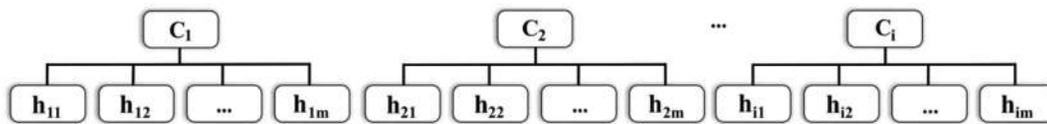
O processo de adaptação começa quando o estudante responde aos questionários iniciais e os agentes obtêm o histórico da sequência de ações passadas do estudante, para que o ambiente possa conhecer o perfil do mesmo, formando a base de conhecimento dos agentes, tanto os de modelo de aluno como os de adaptação de recursos. Os questionários podem ser sobre os estilos de aprendizagem, questões pessoais e/ou testes iniciais de desempenho do aluno a fim de obter um diagnóstico inicial do aluno no que diz respeito aos seus níveis de habilidades e competências com relação a disciplina ou curso que o estudante participa. A partir do momento em que o AVA obtém essas informações, é disponibilizado o modelo (perfil) do aluno de forma visível ao estudante, por isso, sendo chamado de Modelo Aberto de Aluno (MAA).

Ao trabalharmos com níveis de habilidades dos alunos e os resultados de atividades com o uso do MAA, um conceito relacionado que nos deparamos neste trabalho foi o de Learning Analytics (LA), muito utilizado em trabalhos na área de Informática na Educação. LA refere-se a um novo campo de pesquisa na qual são analisados os dados gerados por alunos, com o propósito de melhorar o processo de ensino-aprendizagem (Siemens, 2012). Para que isso ocorra, devem existir ferramentas que forneçam informações de forma organizada. A visualização é uma ferramenta comum usada para refletir sobre informações que seriam difíceis de compreender com descrições textuais ou números tabulados (Few, 2006). Utilizamos essas características de LA, onde o aluno pode ver o seu próprio perfil, no formato de MAA, além do professor poder ver também os perfis de seus alunos nesse mesmo formato.

Em nosso trabalho, usamos o conceito de competências e habilidades baseadas na definição de Perrenoud na Seção 2.3. Sendo assim, cada competência do aluno é composta por um conjunto de habilidades específicas, como mostrado graficamente na Figura 18. Formalmente, seja C o conjunto de competências $\{C_1, C_2, \dots, C_i\}$, onde cada elemento C_i de C é composto por diversas habilidades, ou seja, $C_i = \{h_{i1}, h_{i2}, \dots, h_{im}\}$. Dessa forma, $C = \{\{h_{11},$

$h_{12}, \dots, h_{1m_1}\}, \{h_{21}, h_{22}, \dots, h_{2m_2}\}, \dots, \{h_{i1}, h_{i2}, \dots, h_{im_i}\}\},$ indicando que cada competência C_i é composta ou formada por m_i habilidades, com i e $m_i > 0$. Além disso, podemos perceber que cada habilidade pertence a pelo menos uma competência, e que cada competência possui pelo menos uma habilidade. Por exemplo, no contexto da disciplina de Cálculo Numérico, componente curricular de cursos de Engenharia e Ciências Exatas, segundo Burden e Faire [Burden e Faire, 2010] existem diversas técnicas para encontrar raízes de funções, onde cada técnica está associada a uma ou mais competências necessárias para resolver esse tipo de questão. Como exemplo, uma abordagem para a competência “Achar as raízes de funções pelo método da Bisseção” pode ser definida como a composição das seguintes habilidades: “Manuseio da máquina de calcular”, “Compreensão de funções (continuidade, Teorema do Valor Médio, plotagem de gráficos, convergência e divergência, etc.)” e “Entendimento do funcionamento do Método da Bisseção (escolha do intervalo inicial, aplicação de fórmula, critério de seleção do novo intervalo, etc.)”.

Figura 18. Representação gráfica: cada competência C_i tem h_{im} habilidades.



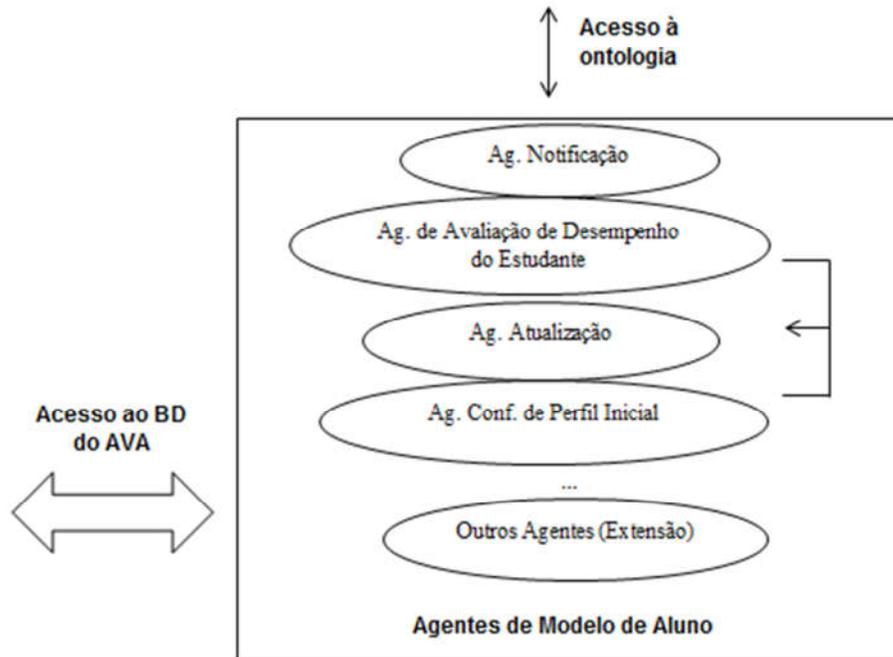
Fonte: o autor.

Quando o estudante finaliza cada atividade no ambiente, os agentes de modelo de aluno e a ontologia que descreve tal modelo, com suas regras, verificam como o estudante se saiu naquela atividade e com isso, atualizam os níveis de competências e habilidades em função dos resultados obtidos na respectiva atividade.

4.4.1 Papéis dos Agentes de Modelo de Aluno

A Figura 19 mostra com maiores detalhes a plataforma dos agentes que manipulam o modelo de aluno. Nessa plataforma, ocorre a comunicação dos agentes tanto com o BD do AVA quanto com a ontologia, de acordo com a arquitetura do ArCARE mostrada na Figura 17. Além disso, há trocas de mensagens entre os próprios agentes, sendo quatro agentes-padrão, com suas funções de agente, embora mais agentes possam ser desenvolvidos dependendo da aplicação baseada no ArCARE desenvolvida.

Figura 19. Plataforma de Agentes do Modelo de Aluno.



Fonte: o autor.

Os agentes-padrão para a manipulação do modelo de aluno mostrados na Figura 19 são:

- *Agente de Configuração de Perfil Inicial*: configura os níveis de competências, habilidades e estilos de aprendizagem iniciais dos alunos em função das manifestações iniciais do aluno no ambiente, a partir do seu primeiro acesso ao AVA, como por exemplo, respostas dos alunos aos questionários respondidos no início de um curso. Além disso, também busca os dados de histórico das ações dos estudantes no AVA, a fim de manter o MAA cada vez mais atualizado. Esse agente envia estas informações ao *Agente de Atualização*. As operações do Agente de Configuração de Perfil Inicial são realizadas por uma função de agente f_{in} ;
- *Agente de Avaliação de Desempenho do Estudante*: responsável pela avaliação das atividades respondidas pelos alunos, além de sua frequência no AVA e envia as pontuações e frequências dos alunos ao *Agente de Atualização*. Para fazer tais operações, o Agente de Avaliação de Desempenho do Estudante possui uma função, f_{av} ;
- *Agente de Atualização*: responsável por atualizar os dados dos alunos de acordo com as interações dos mesmos com o ambiente, mediante as informações oriundas de todos

os agentes que enviam informações dos alunos para ele. Possui uma função de atualização, f_{at} , para executar essa operação;

- *Agente de Notificação*: agente que envia mensagens para o aluno via AVA a fim de mantê-lo informado a respeito de modificações no AVA, como disponibilidade de recursos apresentados, alterações no MAA e novas atividades passadas pelo professor. Possui uma função, f_{not} , para fazer essas operações.

A **função de inicialização de informações**, f_{in} , é definida como: $f_{in} = \{obtem(ea_1, ea_2, \dots, ea_n, i_1, i_2, \dots, i_y)$, $calcula(h_1a_1, h_1a_2, h_1a_3, \dots, h_1a_m, h_2a_1, h_2a_2, h_2a_3, \dots, h_pa_m, k_1, k_2, \dots, k_m, d_1, d_2, \dots, d_m, t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_m)\}$,

Onde: ea_n é o estilo de aprendizagem de ordem n ; h_{pa_m} : é uma habilidade h_p associada a uma atividade a_m , em que podemos ter p habilidades para cada uma das m atividades; i_y : é o y -ésimo interesse do estudante; $obtem(ea_1, ea_2, \dots, ea_n, i_1, i_2, \dots, i_y)$ é uma subfunção que serve para o Agente de Perfil Inicial coletar os dados relativos aos estilos de aprendizagem e interesses dos alunos; $calcula(h_1a_1, h_1a_2, h_1a_3, \dots, h_1a_m, h_2a_1, h_2a_2, h_2a_3, \dots, h_pa_m, k_1, k_2, \dots, k_m, d_1, d_2, \dots, d_m, t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_m, f_1, f_2, \dots, f_m, g_1, g_2, \dots, g_m)$ é uma subfunção contida nesse agente que calcula os níveis de habilidades e competências iniciais dos alunos associadas com as atividades ou testes iniciais que eles responderem, além de outros testes no decorrer do curso ou disciplina no AVA; $k_m, d_m, t_m, c_m, g_m, f_m$: são parâmetros definidos no parágrafo a seguir, na Equação 1, onde para cada habilidade do aluno h_x , sendo $1 \leq x \leq p$, é calculado o resultado obtido nas m atividades associadas com as p habilidades, de acordo com a **função genérica de cálculo de habilidades s** , sendo

$$h_x = s(f_i, g_i, c_i, h_i a_i, d_i, t_i, k_i) \text{ (Equação 1)}$$

Em que g_i é a nota do aluno obtida na atividade i ; f_i é um fator de normalização que permite que $0 \leq h_x \leq 10$; c_i é um valor de peso para cada uma das m atividades, obedecendo a algum critério a respeito de erros e acertos. Por exemplo, c_i pode valer +1, se o aluno acertar a questão completamente ou -1, caso ele erre completamente; $h_i a_i$ é de quanto vale o nível de habilidade em uma determinada atividade i , onde temos m atividades; d_i : dificuldade da questão, especificada pelo professor ao elaborá-la; t_i : é um fator de peso de acordo com o tempo de duração que o aluno respondeu a questão, especificado pelo professor ao elaborar a questão; k_i indica o número de tentativas que o aluno teve nesta questão até entregá-la de vez, com $k_i \neq 0$. Caso o aluno não tenha feito nenhuma tentativa, o valor de h_x é diretamente igual a 0. Caso seja a primeira tentativa, $k_i = 1$. Diversas equações de cálculo de habilidades dos alunos podem ser criadas a partir da função genérica definida na Equação 1.

Para que ocorra a atualização das habilidades do aluno em função de suas repostas às atividades, conforme estão definidas as variáveis da Equação 1, é preciso estabelecer um **critério de atualização**. De quanto os níveis das habilidades do estudante são incrementados ou decrementados, depende dos critérios de atualização de cada curso, sendo estabelecidos pelo professor, coordenador ou outro responsável que administra o AVA. Ao ser estabelecido esse critério de atualização, os agentes de modelo de aluno podem executar as atualizações no perfil do aluno. Nesse trabalho, convencionamos que os níveis de competências e habilidades dos alunos, dos recursos e das atividades passadas aos alunos variam em uma escala normalizada de 0 a 10. Ou seja, caso o valor de uma habilidade h_x seja menor que 0, assume-se o valor 0 e se for maior que 10, assume-se o valor 10. A Tabela 5 mostra um exemplo de critério utilizado na variável c_i para incrementar/decrementar níveis de habilidades dos estudantes em função de suas respostas em uma atividade. Supõe-se que nessa atividade as únicas notas obtidas possíveis sejam 0, 3, 7 e 10.

Tabela 5. Exemplo de critério de incremento/decremento de níveis de habilidades em alunos.

Nota obtida na atividade	Acréscimo/Decréscimo dos níveis de habilidades (c_i)
0	-1
3	-0,5
7	+0,5
10	+1

Fonte: o autor.

A **função de avaliação de desempenho do estudante**, f_{av} , é definida como: $f_{av} = \{ob\acute{t}em(fd,fs), calcula(h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, ta_1, ta_2, ta_m, k_1, k_2, \dots, k_m, d_1, d_2, \dots, d_m, t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_m), analisa(he_1, he_2, \dots, he_x)\}$,

Onde: fd é a frequência diária do estudante no AVA; fs , a frequência semanal; h_{pa_m} : é a habilidade h_p associada a uma atividade a_m , em que podemos ter p habilidades para cada uma das m atividades; ta_m é o tempo de duração que o estudante levou para executar a atividade a_m ; $ob\acute{t}em(fd,fs)$ e $calcula(h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, ta_1, ta_2, ta_m, k_1, k_2, \dots, k_m, d_1, d_2, \dots, d_m, t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_m)$ são subfunções do Agente de Avaliação de Desempenho do Estudante que são definidas de forma análoga à função f_{in} . Ressaltando-se ainda, que para cada habilidade, é calculado o resultado obtido nos questionários e outras atividades utilizando-se a Equação 1; he_x : habilidade extra x do aluno, não associada às atividades

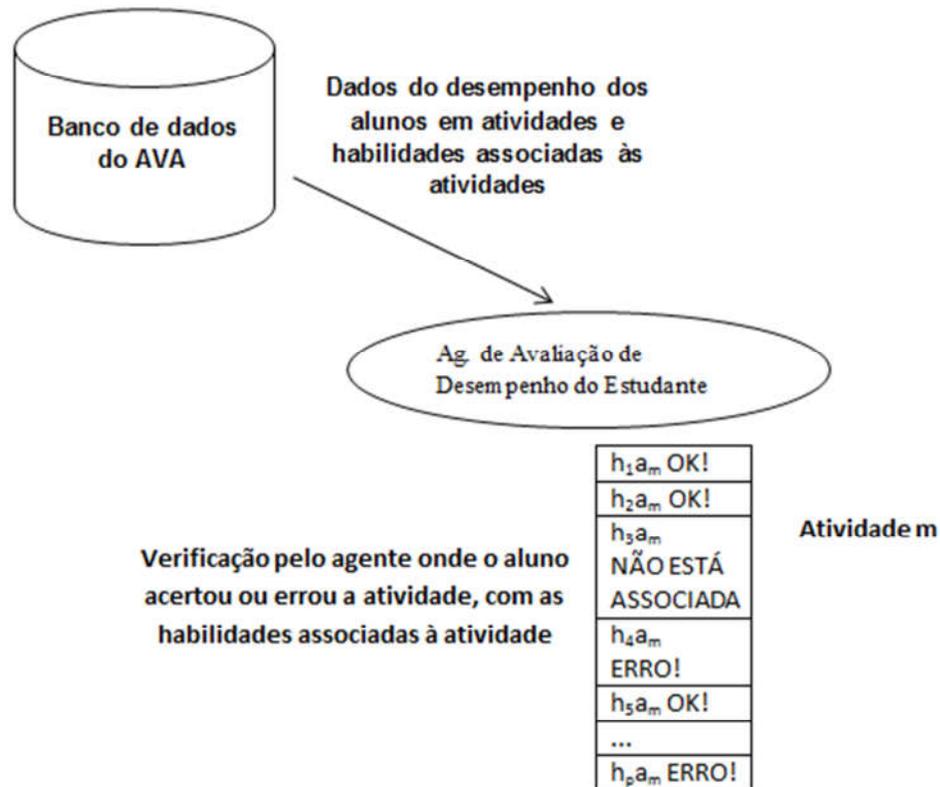
des passadas no AVA. Cada habilidade extra pode ser obtida pela função analisa(he_1, he_2, \dots, he_x), que será explicada no parágrafo a seguir.

Devemos observar que para cada atividade, além das habilidades específicas, como por exemplo, domínio de estrutura de repetição em um algoritmo, existem **habilidades genéricas**, mas que são essenciais para uma boa resolução de um problema ou uma atividade, como por exemplo, interpretação de textos, habilidades estas que podem ser estruturadas de acordo com a Taxonomia de Bloom no domínio cognitivo (Ferraz e Belhot, 2010) (Bloom, Hastings e Madaus, 1971). O Agente de Avaliação de Desempenho do Estudante precisa identificar essas habilidades nos alunos, se eles as possuem ou não. Para isso, a subfunção analisa(he_1, he_2, \dots, he_x) é utilizada, pois estas habilidades extras não são explicitamente associadas com a atividade passada. O total de habilidades extras possível é definido pelo programador dos agentes, em acordo com o professor da disciplina ou curso. Ao existirem as habilidades explícitas das atividades e habilidades extras, os agentes de adaptação de recursos podem tomar as medidas adequadas para aprimorar as habilidades do aluno, por exemplo, por meio de uma recomendação de recursos adicionais.

Um aspecto importante que merece atenção é que em uma atividade frequentemente há diversas habilidades associadas. Supomos, nesse caso, que o aluno tenha algumas dessas habilidades, mas outras, não. Por exemplo, em Cálculo Numérico, ao aluno se deparar com uma questão que envolva a competência de achar raízes de funções pelo método da bisseção, o aluno pode ter a habilidade de manusear a máquina de calcular, mas ter dificuldades em lidar com gráficos de funções. Sendo assim, acredita-se que o aluno não irá conseguir obter a nota máxima em uma questão. Logo, deve haver um elemento nos agentes de modelo de aluno, especificamente na função f_{av} do Agente de Avaliação de Desempenho do Estudante, que avalia a atividade respondida do ponto de vista de cada habilidade associada a uma atividade m de forma separada, uma a uma habilidade, a fim de saber em quais habilidades o aluno está com deficiências. A Figura 20 mostra um exemplo, onde o Agente de Avaliação de Desempenho do Estudante obtém as respostas dos alunos em atividades do banco de dados do AVA e avalia, nesse momento, uma atividade m , onde nessa atividade são verificadas as partes da questão que o aluno acertou (indicadas por um OK!) e onde o aluno errou (indicadas por um ERRO!). As habilidades não associadas à atividade m não são verificadas, por isso não vão refletir na atualização dos níveis de habilidades do estudante. Dessa forma, são percorridas todas as habilidades associadas àquela atividade e é verificado, de acordo com a resposta do aluno, se tal resposta indica ou não que ele tem tal habilidade, para depois o perfil do aluno

ser atualizado. Essa questão, na verdade, depende da atividade passada ao aluno, onde os agentes de modelo de aluno e de adaptação de recursos podem ser implementados, na prática, como convier para atender a essas necessidades.

Figura 20. Verificação dos erros e acertos dos estudantes em uma atividade e suas habilidades associadas.



Fonte: o autor.

A **função de atualização do perfil do aluno**, f_{at} , é definida como: $f_{at} = \{\text{salvar}(\text{fd}, \text{fs}, \text{ea}_1, \text{ea}_2, \dots, \text{ea}_n, i_1, i_2, \dots, i_y, h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, k_1, k_2, \dots, k_m, t_1, t_2, \dots, t_m), \text{acessou}(\text{aluno}, \text{recurso}, \text{curso}, \text{horário}, \text{requisições_de_recurso}), \text{feedback_satisfeito}(\text{aluno}, \text{recurso}, \text{curso}, \text{nível_satisfação}), \text{avaliação_pre_pos_recurso}(\text{aluno}, \text{recurso}, \text{curso}, \text{lista_de_atividades_pre}, \text{lista_de_atividades_pos})\}$, onde a subfunção $\text{salvar}(\text{fd}, \text{fs}, \text{ea}_1, \text{ea}_2, \dots, \text{ea}_n, i_1, i_2, \dots, i_y, h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, k_1, k_2, \dots, k_m, t_1, t_2, \dots, t_m)$ armazena no banco de dados do AVA os elementos oriundos das funções f_{in} e f_{av} . As demais subfunções serão explicadas no parágrafo a seguir.

A f_{at} também é utilizada após uma recomendação de recursos. Após cada recomendação feita, é verificado se o aluno acessou ou não tal recurso, pela subfunção $\text{acessou}(\text{aluno}, \text{recurso}, \text{curso}, \text{horário}, \text{requisições_de_recurso})$, onde $\text{requisições_de_recurso}$ indica quantas

vezes o aluno tentou ver os recursos disponíveis para ele no AVA, em que isto serve para indicar se ele é um aluno que está solicitando ajuda no ambiente ou não. A f_{at} também serve para saber se o aluno ficou ou não satisfeito com a utilização deste (por meio de uma simples pergunta ao aluno no AVA), pela subfunção `feedback_satisfeito(aluno, recurso, curso, nível_satisfação)`. O feedback do aluno pode ser por meio de um portfólio de aprendizagem ou mesmo uma pesquisa de satisfação para o aluno avaliar de quanto o recurso foi útil para ele. Com isso, cada recurso vai possuindo um registro de utilizações, sendo registrados seus sucessos ou fracassos ao serem recomendados para o aluno. Esses sucessos e fracassos estão de acordo com 2 parâmetros: 1) feedback do aluno, conforme já mostrado na subfunção `feedback_satisfeito(aluno, recurso, curso, nível_satisfação)` e 2) se o aluno obteve boas pontuações (acima da média) nas atividades posteriores à recomendação dos recursos (`lista_de_atividades_pos`). No entanto, para essa verificação de sucesso ou fracasso de um recurso, somente são avaliadas atividades cujas competências associadas sejam as mesmas competências associadas aos recursos apresentados anteriormente. Todas essas verificações são realizadas pela subfunção `avaliação_pre_pos_recurso(aluno, recurso, curso, lista_de_atividades_pre, lista_de_atividades_pos)`, em que `lista_de_atividades_pre` consiste em todas as atividades que o aluno fez antes de utilizar um determinado recurso. Um recurso apresentado a um aluno em que ocorreu fracasso após sua utilização não é recomendado novamente a ele. Nesse caso, outros recursos são apresentados a este aluno.

Dessa forma, é formado um **histórico de utilização dos recursos** apresentados, armazenado no banco de dados do AVA, e com isso, um metachecimento por parte dos agentes, permitindo uma classificação (ranking) dos recursos mais úteis pela função f_{at} e os mais recomendados, pois se um recurso é frequentemente apresentado para diversos estudantes, podemos perceber que para uma ou mais turmas daquela disciplina, esse recurso é importante. Essas informações tornam-se particularmente úteis, uma vez que, quando os agentes de adaptação de recursos conhecem os recursos mais úteis e/ou mais apresentados, estes podem ser futuramente recomendados nas turmas posteriores do mesmo curso ou disciplina, permitindo seu reuso de uma forma mais otimizada, sem precisar refazer todas as avaliações previamente realizadas na turma anterior.

Cada estudante também faz parte deste histórico de utilização de recursos, de forma que seja útil associar diversos tipos de perfis de alunos com recursos mais úteis, otimizando o processo de raciocínio no sistema multiagente quando os agentes de adaptação de recursos entrarem em ação. Com estes registros de ações passadas, tanto dos recursos quanto da utili-

zação dos recursos pelos estudantes, estes também são associados com outras variáveis de perfil dos estudantes, conforme já citadas, como seus estilos de aprendizagem, frequências no ambiente e interesses, a fim de descobrir padrões que alimentam a base de conhecimento dos agentes, otimizando o processo de adaptação de recursos no AVA pelos agentes de adaptação de recursos.

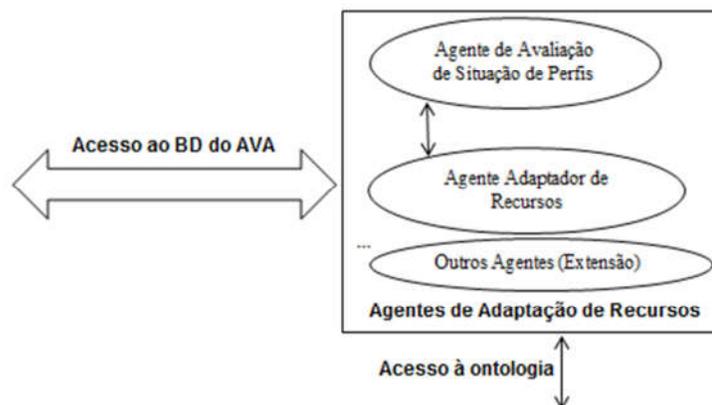
A **função de notificação**, f_{not} , é definida como: $f_{not} = \{envia(e, msg, c)\}$, onde $envia(e, msg, c)$ é uma subfunção que envia ao estudante e de um curso c alguma mensagem msg . É importante ressaltar que msg geralmente é um texto, mas podem ser utilizados outros elementos para compor a mensagem para o estudante, como por exemplo, links para as atividades ou recursos apresentados no AVA.

Além disso, em relação à frequência do estudante, são enviadas mensagens de alerta via e-mail para estudantes que estão ausentes do AVA pelo Agente de Notificação, utilizando a função f_{not} de acordo com a avaliação de frequência no AVA feita pelo Agente de Avaliação de Desempenho do Aluno, com sua função f_{av} . Porém, é importante ressaltar que o professor pode alterar os critérios de verificação de frequência do estudante no AVA como convier.

4.4.2 Papéis dos Agentes de Adaptação de Recursos

A Figura 21 mostra com maiores detalhes a plataforma de agentes que atuam na adaptação, seleção e recomendação de recursos. Nessa plataforma, ocorre a comunicação dos agentes tanto com o BD do AVA quanto com a ontologia, de acordo com a arquitetura do ArCARE mostrada na Figura 17. Além disso, há trocas de mensagens entre os próprios agentes, sendo dois agentes-padrão, embora dependendo da aplicação que o ArCARE possa lidar, mais agentes possam ser desenvolvidos.

Figura 21. Plataforma de Agentes do Modelo de Aluno.



Fonte: o autor.

Os agentes-padrão para a adaptação de recursos na Figura 21 são:

- *Agente de Avaliação de Situação de Perfis*: agente que verifica e avalia a situação corrente do estudante no ambiente, verificando que medidas podem ser tomadas para auxiliar no aprendizado do aluno. Tal agente pode, por exemplo, fazer a busca por dúvidas, erros dos alunos em atividades respondidas, avaliar os níveis de habilidades e competências dos alunos, buscar os estilos de aprendizagem dos mesmos, além das demais informações que compõem o MAA e por fim, mapear esses dados em recursos de aprendizagem necessários a serem disponibilizados no AVA. Com estas informações, este agente envia uma mensagem ao *Agente Adaptador de Recursos* os dados dos alunos com os respectivos recursos necessários e suas devidas adaptações associados a eles. Para fazer tais operações, este agente possui uma função, f_{sit} ;
- *Agente Adaptador de Recursos*: agente que obtém as informações do *Agente de Avaliação de Situação de Perfis* e executa o processo de apresentar no AVA os recursos para cada aluno em função de seu MAA. Para fazer tais operações, este agente possui uma função, f_{adapt} .

A **função de verificação de situação do aluno**, f_{sit} , é definida como: $f_{sit} = \{verifica_e_combina(c, f, ea_1, ea_2, \dots, ea_n, h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, hc_1, hc_2, \dots, hc_q, h_{1r_1}, h_{1r_2}, \dots, h_{1r_r}, h_{2r_1}, h_{2r_2}, h_{2r_3}, \dots, h_{sr_r}, ea_{1r_1}, ea_{1r_2}, \dots, ea_{1r_r}, ea_{2r_1}, ea_{2r_2}, \dots, ea_{nr_r}, i_1, i_2, \dots, i_y, i_{1r_1}, i_{1r_2}, \dots, i_{yr_r})\}$,

Onde: c é o curso que o aluno está matriculado; f é a frequência do estudante; ea_n é o estilo de aprendizagem de ordem n do aluno; h_{pa_m} : é a habilidade h_p associada a uma atividade a_m , em que podemos ter p habilidades para cada uma das m atividades; hc_q é a q -ésima habilidade pertencente ao curso ou disciplina em que o aluno está matriculado; h_{sr_r} : é a habilidade h_s associada a um recurso r_r , em que podemos ter s habilidades para cada um dos r recursos; ea_{nr_r} : é o estilo de aprendizagem ea_n associado a um recurso r_r , em que podemos ter n estilos de aprendizagem para cada um dos r recursos; i_y : é o y -ésimo interesse do estudante; i_{yr_r} : é i y -ésimo interesse associado ao recurso r .

É importante ressaltar ainda, que para cada recurso r , são comparados os níveis de habilidades e os estilos de aprendizagem dos estudantes com os respectivos níveis de habilidades e estilos de aprendizagem associados aos recursos candidatos a serem adaptados, pela subfunção $verifica_e_combina(c, f, ea_1, ea_2, \dots, ea_n, h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}, hc_1, hc_2, \dots, hc_q, h_{1r_1}, h_{1r_2}, \dots, h_{1r_r}, h_{2r_1}, h_{2r_2}, h_{2r_3}, \dots, h_{sr_r}, ea_{1r_1}, ea_{1r_2}, \dots, ea_{1r_r}, ea_{2r_1}, ea_{2r_2}, \dots, ea_{nr_r}, i_1, i_2, \dots, i_y, i_{1r_1}, i_{1r_2}, \dots, i_{yr_r})$. Além disso, esta função também leva em conta os interes-

ses dos alunos, ao sugerir recursos para os mesmos. Dependendo do critério de comparação aplicado nessas informações, o recurso será ou não adaptado.

A **função de adaptação de recursos**, f_{adapt} , é definida como: $f_{\text{adapt}} = \{\text{executa_estratégia}(c,e,ta,tr,r_1,r_2,\dots,r_r)\}$,

Onde: c é o curso que o aluno está matriculado; e é o estudante; ta é a estratégia pedagógica aplicada na adaptação do recurso, podendo ser uma recomendação ou um ajuste no recurso já apresentado ao estudante, onde isso depende de cada recurso e o propósito que ele deve executar no AVA; r é cada recurso associado ao curso ou disciplina c ; tr é o tempo que o recurso estará disponível ao aluno, definido previamente pelo professor ou programador dos agentes.

4.4.3 Funcionamento dos Agentes para a Adaptação de Recursos

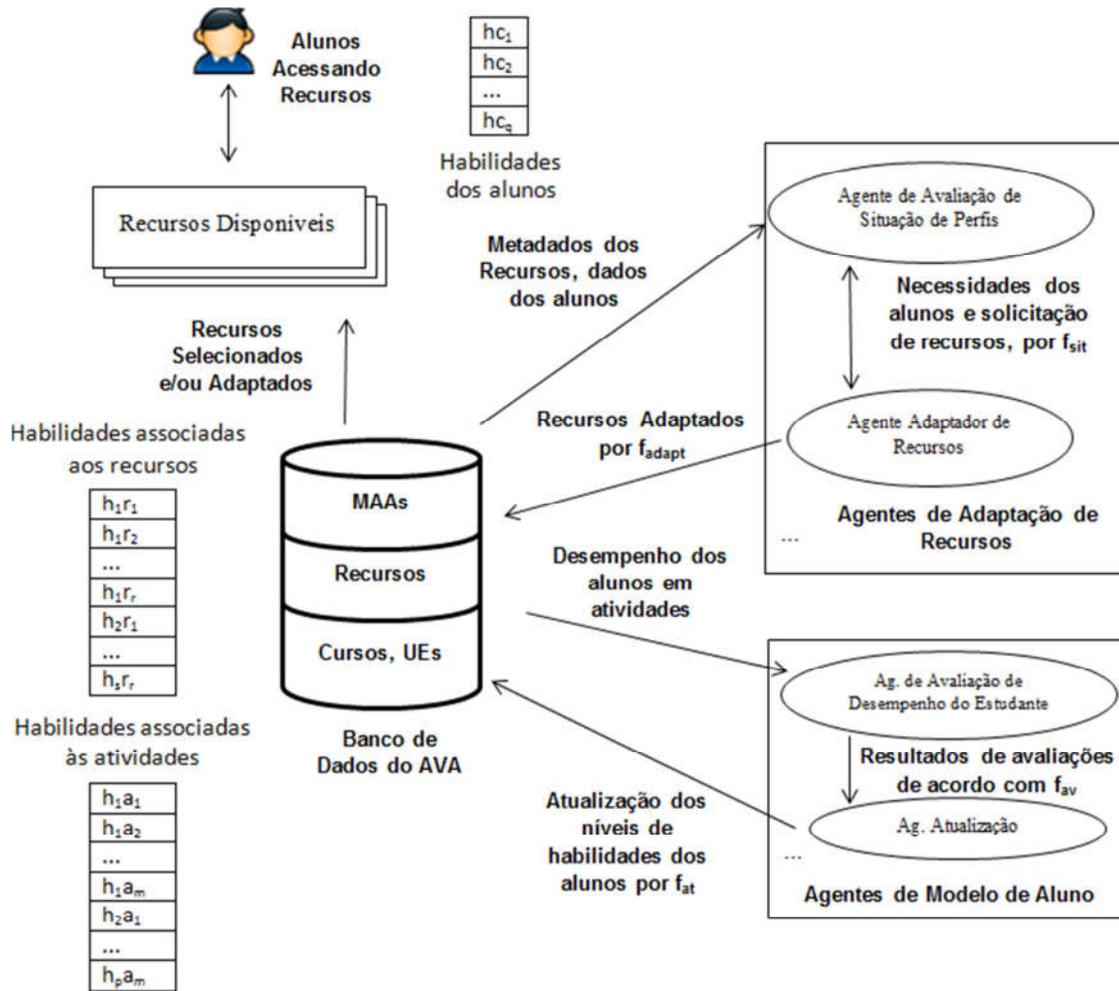
A Figura 22 mostra um esquema detalhado, a partir da arquitetura do ArCARE mostrada na Figura 17, de como ocorre a atualização do perfil do estudante em função de suas respostas às atividades, além de um exemplo de uma adaptação de recursos (recomendações) em um AVA após as habilidades terem sido atualizadas. Quanto ao período ou intervalo de tempo que os agentes entrariam em execução, isso dependeria de cada curso em que os agentes estiverem atuando e veremos exemplos de aplicações nos Capítulos 5 e 6.

Ao executar as atividades, obviamente o estudante pode acertar ou errar. No entanto, aqui consideramos que quando o estudante não faz uma atividade, é atribuída a nota zero a ele. Sendo assim, ao acertar ou errar uma atividade, respectivamente os níveis das q habilidades dos estudantes em um determinado curso (hc_1, hc_2, \dots, hc_q) que corresponderem às mesmas habilidades de cada uma das m atividades ($h_{1a_1}, h_{1a_2}, h_{1a_3}, \dots, h_{1a_m}, h_{2a_1}, h_{2a_2}, h_{2a_3}, \dots, h_{pa_m}$), ambos armazenados no BD do AVA, vão sendo incrementados ou decrementados, conforme mostrado na Figura 22, pelos agentes de modelo de aluno, especialmente pela f_{av} do Agente de Avaliação de Desempenho do Estudante.

Após as atualizações dos níveis de habilidades do estudante terem sido realizadas, os agentes de adaptação de recursos comparam os níveis de habilidades dos alunos e as habilidades associadas aos r recursos ($h_{1r_1}, h_{1r_2}, \dots, h_{1r_r}, h_{2r_1}, h_{2r_2}, h_{2r_3}, \dots, h_{sr_r}$), especificamente, o Agente de Avaliação de Situação de Perfis, pela f_{sit} , que é mostrado na Figura 22. Para cada recurso que tenha um objetivo de aumentar os níveis de habilidades dos alunos, caso as habilidades dos estudantes sejam inferiores às habilidades associadas ao recurso em questão, este é apresentado ao estudante. Caso contrário, tal recurso não será apresentado ao estudante. No

entanto, opcionalmente definido pelo professor ou pelo próprio aluno, a depender do curso, para recursos que são materiais complementares para um estudante que já alcançou uma determinada habilidade, esses tipos de recursos são apresentados.

Figura 22. Atualização das habilidades do estudante em função de resultados de atividades e recomendação de recursos em função das habilidades.

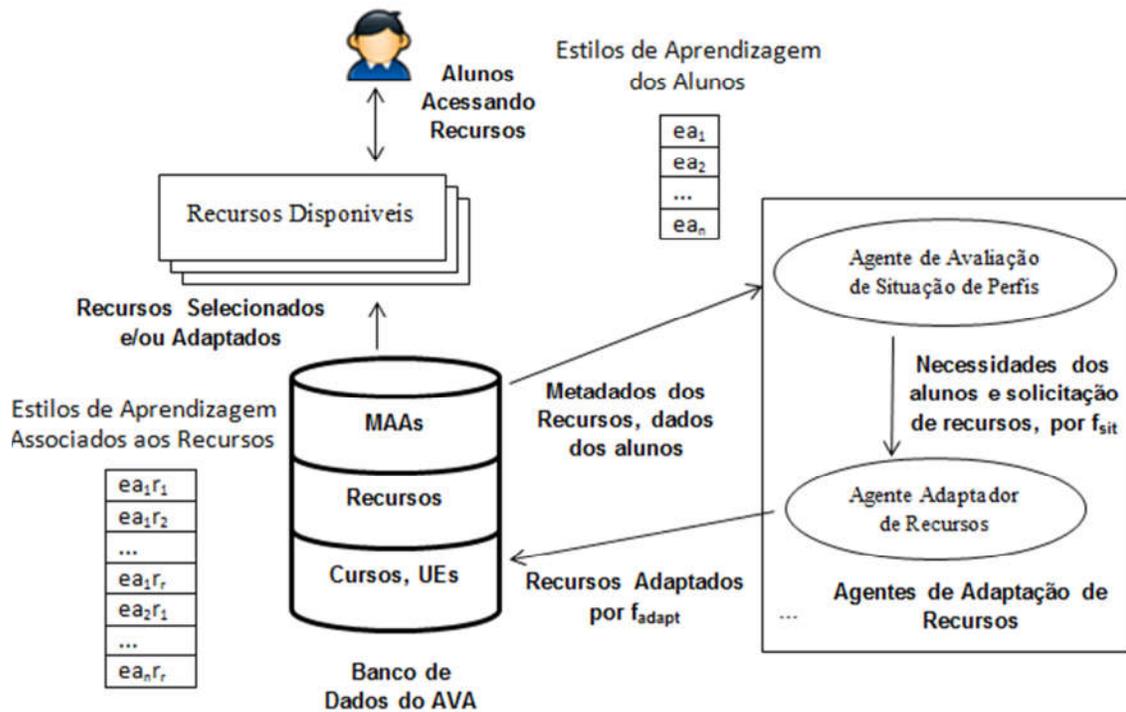


Fonte: o autor.

Em relação aos estilos de aprendizagem dos alunos, o processo de recomendação de recursos é ilustrado na Figura 23. Como se vê, é realizada uma comparação entre os níveis dos n estilos de aprendizagem mínimos requeridos dos r recursos (ea_{1r_1} , ea_{1r_2} , ... ea_{1r_r} , ea_{2r_1} , ea_{2r_2} , ... , ea_{nr_r}) e os níveis dos n estilos de aprendizagem do aluno (ea_1 , ea_2 , ..., ea_n), pela f_{sit} . Caso os níveis dos estilos de aprendizagem dos alunos sejam maiores ou iguais aos níveis mínimos dos estilos de aprendizagem associados aos recursos, então estes recursos serão apresentados aos estudantes pela f_{adapt} .

Neste trabalho, seguindo a definição de Kolb (1984) a respeito dos estilos de aprendizagem, convenciamos que os estilos de aprendizagem não mudam, ou seja, eles podem ser definidos como um estado duradouro e estável que deriva de configurações consistentes das transações entre o indivíduo e seu meio ambiente. No entanto, é importante ressaltar que há trabalhos na literatura que indicam que os estilos de aprendizagem dos estudantes podem mudar no decorrer de suas vidas acadêmicas, principalmente na transição da infância para a adolescência (Diaz e Cartnal, 1999) (Creative Learning, 2016), cuja questão não está no escopo desta tese.

Figura 23. Recomendação de recursos ao estudante em função de seus estilos de aprendizagem.



Fonte: o autor.

É importante ressaltar, ainda, que entre as atividades passadas para o estudante, pode ocorrer de uma atividade não contemplar alguns ou todos os estilos de aprendizagem do aluno. Nesse caso, tal disposição das atividades do AVA é de responsabilidade do professor, conforme já descrito no início dessa seção.

Sendo assim, na Figura 24, segue um algoritmo em pseudocódigo de recomendação de recursos, pela função `adapta_recurso_simples(curso, estudante, ação, lista_de_recursos_disponíveis)`, uma instanciação da função de adaptação f_{adapt} do Agente Adap-

tador de Recursos, com a finalidade de que os alunos alcancem níveis maiores de habilidades. No algoritmo da Figura 24, apenas os estilos de aprendizagem, as habilidades e os interesses dos estudantes foram verificados.

Figura 24. Algoritmo de recomendação de recursos simples.

```

Função adapta_recurso_simples (curso, estudante, ação, lista_de_recursos_disponíveis):
  Se (ação=RECOMENDAÇÃO) então:
    Lista_de_recursos_recomendados ← nulo;
      Para cada (recurso pertencente a lista_de_recursos_disponíveis) faça:
        Se (estudante.lista_de_habilidades ≤ recurso.lista_de_habilidades) e
        (estudante.estilos_de_aprendizagem ≥ recurso.estilos_de_aprendizagem) e (estudante.lista_de_interesses = recurso.lista_de_interesses) então:
          Adiciona (lista_de_recursos_recomendados, recurso);
        Fim se
      Fim para
    Fim se
  Retorna lista_de_recursos_recomendados;

```

Fonte: o autor.

Como podemos perceber, este algoritmo da Figura 24 leva em consideração habilidades, estilos de aprendizagem e interesses. No entanto, na prática, outras características são levadas em conta no ArCARE. Dessa forma, a fim de que o processo de adaptação de recursos leve em consideração um número maior de situações práticas, nos parágrafos a seguir colocamos outras situações que o ArCARE deve lidar.

Na recomendação de recursos, podem haver casos em que para um determinado aluno haja uma grande quantidade de recursos recomendados, por exemplo, 50. Para evitar esse possível excesso de recursos recomendados, adotamos um critério de serem recomendados, no máximo, os r recursos considerados melhores (sendo r escolhido pelo estudante, a depender de suas permissões no AVA), pela ação da f_{adapt} , embora o professor possa alterar essa quantidade máxima de recursos recomendados. Nesse trabalho, adotamos a heurística de que um recurso é considerado **melhor** (com a finalidade de que os alunos alcancem níveis maiores de habilidades) quando obedece aos seguintes critérios, nessa ordem: I) Recursos em que ocorrer $\text{Min}(h_i r_j - h_i)$, $1 \leq i \leq s$, $1 \leq j \leq r$, $h_i r_j - h_i > 0$, ou seja, menor diferença positiva entre os níveis de habilidades dos recursos ($h_i r_j$) e dos alunos (h_i), obedecendo a regra de que os níveis de habilidades dos alunos devem ser menores ou iguais aos níveis de habilidades associados aos recursos; II) Recursos em que ocorrer $\text{Min}(\text{Abs}(ea_i r_j - ea_i))$, $1 \leq i \leq n$, $1 \leq j \leq r$, ou seja, menor módulo da diferença entre os níveis de estilos de aprendizagem do estudante e os associados

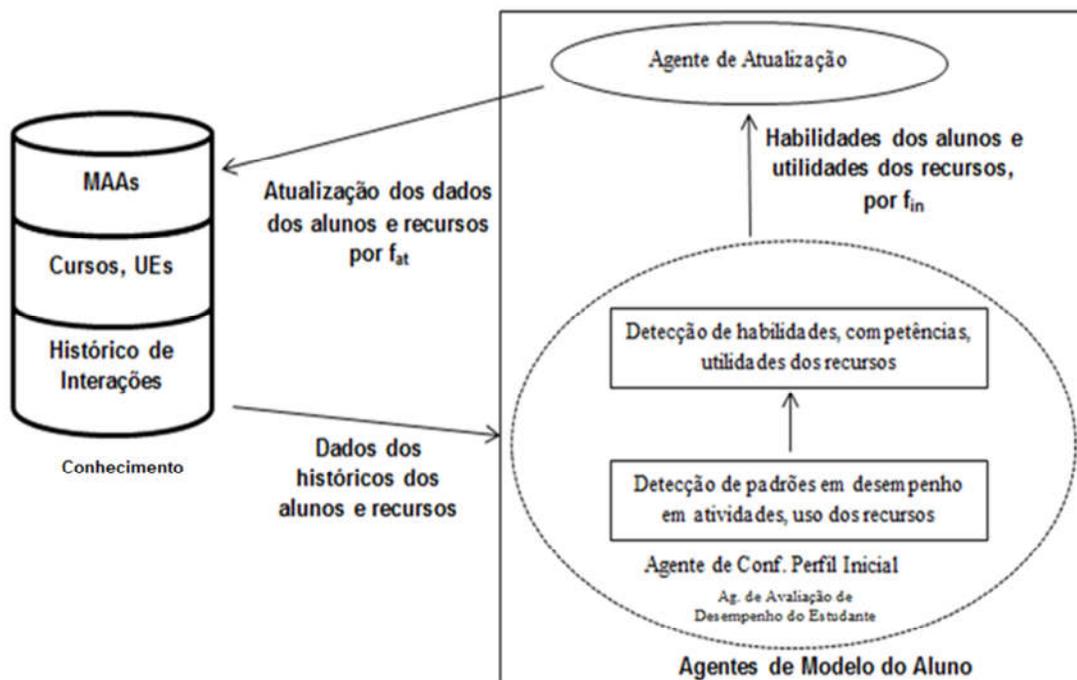
aos recursos; III) de quanto o recurso foi útil para os outros alunos (os que tiverem maiores níveis de utilidade serão recomendados, onde esses níveis de utilidade são atualizados por f_{at}), de acordo com o seu histórico de recomendações, sendo este histórico avaliado pela f_{adapt} da seguinte maneira: os agentes selecionam um recurso baseados em recomendações passadas daquele recurso; para isso, o algoritmo de mineração de dados *Apriori* (Agrawal e Srikant, 1994) é aplicado, tomando como variáveis os feedbacks dos estudantes após utilizar determinado recurso, o desempenho do estudante após o uso daquele recurso e a frequência de recomendações daquele recurso. Os critérios I) e II) se justificam pelo fato de que um recurso adaptado ou recomendado para alunos deve possuir os níveis de habilidades e estilos de aprendizagem o mais próximo possível dos níveis dos alunos (sendo que os níveis de habilidades dos recursos devem ser maiores que os dos alunos), a fim de que os alunos, ao receberem os recursos no AVA, possam aprender mais e aumentarem seus níveis de habilidades. Mesmo assim, pode haver situações em que nenhum recurso é recomendado para um ou mais alunos, de acordo com as regras de recomendação. Nesse caso, para esses estudantes sem recursos recomendados, serão exibidos recursos mais similares possíveis baseados nos critérios I), II) e III).

A fim de evitar somente o aparecimento de recursos recomendados no AVA, o que pode às vezes ser imperceptível para o estudante, no AVA deve haver uma opção para o estudante clicar em um link para ver quais recursos foram recomendados a ele. Além disso, é importante também definir a respeito da disponibilidade do recurso. Por padrão, adotou-se que os recursos ficam disponíveis para o estudante até que I) ele não queira mais aquele recurso, tendo no ambiente uma opção de não mostrar mais tal recurso ou II) o próprio professor oculte aquele recurso, podendo excluí-lo caso tenha sido ele mesmo o criador de tal recurso.

Com isso, é montado o **histórico de ações no AVA**, seja pelas interações dos alunos, pelas utilidades dos recursos, entre outras informações de uma forma mais generalizada do que apenas o histórico de utilizações dos recursos. Para que ocorra a obtenção das informações do histórico de ações passadas no AVA pelo aluno, os Agentes de Configuração de Perfil Inicial e de Avaliação de Desempenho do Estudante utilizam entram em ação, como mostrado na Figura 25 a seguir. São detectados **padrões** de desempenho em atividades e a respeito da utilização dos recursos a fim de saber o quanto um recurso foi útil e se ele ajudou o estudante a resolver ou não seus problemas na disciplina, situações essas já citadas nos parágrafos anteriores. Em seguida, esses padrões são mapeados em valores de habilidades do estudante e utilidade do recurso, cujos valores são utilizados pela f_{at} do Agente de Atualização, a fim de que

as informações a respeito dos estudantes e recursos sejam atualizadas, informações essas que serão importantes para os Agentes de Adaptação de Recursos utilizarem em processos de adaptação de recursos.

Figura 25. Obtenção do histórico e utilização do AVA pelos Agentes de Modelo do Aluno.



Fonte: o autor.

Portanto, percebemos que há vários elementos essenciais num processo de adaptação de recursos no ArCARE: agentes, funções de agentes, ontologia, banco de dados do AVA, histórico de ações no ambiente, cursos, características dos alunos (competências, habilidades, estilos de aprendizagem, frequência no ambiente e interesses), os recursos em si, equipamento que o estudante utiliza e as estratégias pedagógicas envolvidas. Consideramos aqui que todo curso é voltado para o desenvolvimento de competências dos alunos em uma ótica Construtivista.

Com base em todos os elementos descritos nesta seção, percebemos que o ArCARE não pode ser apenas um recomendador de recursos, embora essa seja uma de suas principais funções; e além de recomendar, muitos parâmetros e circunstâncias descritas devem ser consideradas no processo de adaptação de recursos educacionais. A Figura 26 apresenta o algoritmo da Figura 24 aprimorado, com a função `adapta_recurso_aprimorada` (`curso`, `estudante`, `ação`, `lista_de_recursos_disponíveis`, `qtde`) considerando a utilidade do recurso, quantidade máxima de recursos recomendados e situações em que nenhum recurso seria recomendado pela verificação simples do algoritmo da Figura 24. Essas verificações estão de acordo, inclu-

sive, com os 3 critérios citados anteriormente no que diz respeito a um recurso ser considerado melhor.

Figura 26. Trecho de algoritmo de adaptação de recursos aprimorado.

```
Função adapta_recurso_aprimorada (curso, estudante, ação, lista_de_recurso_disponíveis, qtde):  
  Enquanto (cont(lista_de_recurso_recomendados) ≤ qtde)  
    Se (ação=RECOMENDAÇÃO)  
      lista_de_recurso_recomendados ← nulo;  
      Para cada (recurso pertencente a lista_de_recurso_disponíveis) faça:  
        Se (estudante.lista_de_habilidades ≤ recurso.lista_de_habilidades) e (estudante.estilos_de_aprendizagem ≥ recurso.estilos_de_aprendizagem) e (estudante.lista_de_interesses = recurso.lista_de_interesses) então:  
          Adiciona (lista_de_recurso_recomendados, recurso);  
        Senão  
          Se recurso.utilidade = max_utilidade (lista_de_recurso_disponíveis) então:  
            Adiciona (lista_de_recurso_recomendados, recurso);  
        Fim Se  
      Fim Se  
    Fim para  
    Se (Lista_de_recurso_recomendados = nulo) então:  
      Para cada (recurso pertencente a lista_de_recurso_disponíveis) faça:  
        Se (estudante.lista_de_habilidades - recurso.lista_de_habilidades) =  
        Min(Abs(estudante.lista_de_habilidades - recurso.lista_de_habilidades)) e (estudante.estilos_de_aprendizagem ≥ recurso.estilos_de_aprendizagem) e (estudante.lista_de_interesses = recurso.lista_de_interesses) então:  
          Adiciona (lista_de_recurso_recomendados, recurso);  
        Senão  
          Se recurso.utilidade = max_utilidade (lista_de_recurso_disponíveis) então:  
            Adiciona (lista_de_recurso_recomendados, recurso);  
        Fim Se  
      Fim Se  
    Fim para  
  Fim Se  
  Fim enquanto  
Retorna lista_de_recurso_recomendados;
```

Fonte: o autor.

A seção a seguir apresenta e descreve a ontologia de MAA.

4.5A Ontologia de Modelo Aberto de Aluno

Outra importante etapa neste trabalho foi construir uma ontologia de domínio que representa o MAA. Além de permitir uma maior facilidade de inferências e interpretações semânticas do sistema e do AVA, a ontologia também é útil para a correta manipulação das trocas de mensagens entre os agentes e definição dos elementos do ArCARE, como o aluno e o recurso. Para o modelo de aluno foi utilizado o padrão IMS LIP (2017), que é um padrão largamente usado hoje em dia e que permite extensões.

Originalmente, conforme explicado no Capítulo 2, o LIP tem 11 categorias (ou classes) de dados do estudante, mas apenas cinco foram usadas (*Activity*, *Competency*, *Identification*, *Accessibility* e *Interest*). As outras 6 categorias restantes do LIP não foram usadas porque elas não fazem parte do escopo deste trabalho, ou já são substituídas pelas categorias que foram utilizadas, evitando qualquer conflito de informações com dados semelhantes. Além disso, este padrão foi estendido para as necessidades deste trabalho, acrescentando-se 3 outras categorias: *CCPP_Device*, *Learning_Styles* e *Frequency*. A categoria *CCPP_Device* descreve os dispositivos utilizados pelos alunos. Assim como há padrões para o modelo do aluno, existem normas para a caracterização de dispositivos móveis, com as suas características como tamanho da tela, desempenho da CPU, memória, entre outros. Um dos mais usados atualmente e usado neste trabalho é o *Composite Capabilities/Preference Profile* (CC/PP) (2017). Usando a categoria *Frequency*, esta descreve a frequência dos alunos no AVA e também é verificado se o aluno terminou ou não suas atividades. Finalmente, a categoria *Learning_Styles* contém informações sobre os estilos de aprendizagem. Para este trabalho, utilizamos os estilos de Felder-Silverman e Honey-Alonso.

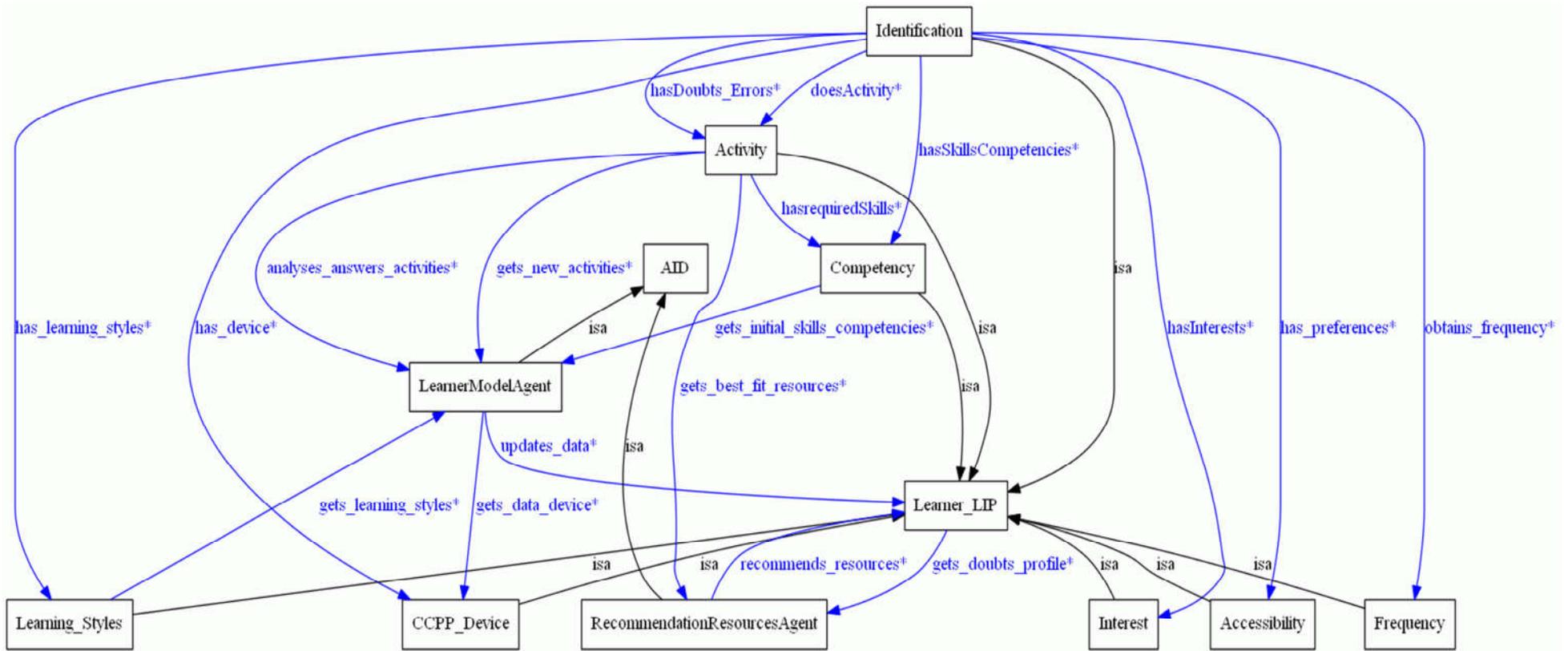
Na categoria *Activity* estão as informações acerca das atividades propostas pelo professor. É importante ressaltar que esta categoria descreve também os recursos, como por exemplo, arquiteturas pedagógicas, visto que alguns dos recursos são atividades e muitos metadados servem para caracterizar um recurso. Sendo assim, em vez de criarmos outra categoria de extensão no padrão LIP, optamos por utilizar a categoria *Activity* já existente e utilizar seus campos, acrescentando alguns necessários para uma melhor diferenciação de atividades e recursos, cujos campos são mostrados na Tabela 6. Todos os recursos possuem metadados para documentá-los utilizando o padrão LTSC/IEEE Learning Object Metadata (LOM) (2002). Esses metadados são úteis para o mapeamento feito entre os campos do LOM nos recursos e os estilos de aprendizagem do aluno, a fim de que os recursos recomendados aos alunos sejam mais precisos. Neste caso, os recursos podem ser tratados como objetos de aprendi-

zagem, embora nem todo recurso seja, de fato, um objeto de aprendizagem. Sendo assim, os campos do LOM que fazem parte das características dos recursos estão de acordo com um subconjunto dos campos de metadados selecionado a partir do trabalho de Araújo *et al.* (2014), o qual propôs um mapeamento dos campos de metadados com relação às características dos estilos de aprendizagem do aluno. Conceitualmente nesta tese, quando tratamos as informações específicas do recurso, mesmo utilizando a categoria *Activity* do padrão LIP, temos um **modelo do recurso**, que consiste nos dados do recurso utilizados para o processo de recomendação ou adaptação deste no ambiente educacional.

A ontologia de modelo de aluno na classe *Competency* descreve os níveis de competências e habilidades de cada aluno em uma disciplina ou curso. As descrições das competências do aluno são compatíveis com o padrão IMS RDCEO (2017). No ArCARE, cada competência do aluno é composta por um conjunto de habilidades específicas, conforme já citado na seção 4.4. Os agentes fazem avaliações frequentes das atividades dos alunos, sendo uma avaliação formativa, ou seja, obtendo os desempenhos destes e atualizando os níveis de habilidades dos alunos. Outras categorias utilizadas são *Identification* (para os dados pessoais do aluno), *Accessibility* (dados de acessibilidade do usuário, credenciais no AVA) e *Interest* (contendo os interesses dos estudantes).

A ontologia de modelo de aluno foi desenvolvida no editor Protégé (2017) e tem três classes que relacionam os dados do aluno com o sistema multiagente, seguindo a orientação proposta em (Bellifemine et al., 2007): *Concept*, classe abstrata cujas filhas são as subclasses *AID* (que contém os agentes que fazem parte do sistema multiagente) e *Learner_LIP* (contendo as categorias LIP utilizadas neste trabalho, além de *CCPP_Device*, *Learning_Styles* e *Frequency*); *Predicate*, responsável pelos fatos sobre o ambiente onde os agentes atuam; e *AgentAction* (contendo as ações dos agentes). Todas estas três classes permitem que a ontologia descreva os agentes, seus atributos e relações. A Figura 27 mostra uma representação gráfica resumida da ontologia de MAA, onde aparece apenas a classe *Concept* e suas filhas *Learner_LIP* e *AID* que descrevem, respectivamente, o padrão LIP estendido e os dois tipos de agentes que permeiam o conceito do ArCARE. Uma versão completa desta ontologia está no Apêndice A desta tese.

Figura 27. Representação gráfica resumida da Ontologia de MAA usada pelo ArCARE.



Fonte: o autor.

Na Figura 27, as relações e regras entre as classes descrevem os *facets* da ontologia, representados por setas azuis e pretas (estas últimas indicando apenas a relação "isa", que significa "é um", de subclasse para superclasse). Como alguns exemplos de *facets* em setas azuis na Figura 27, o estudante faz uma atividade (*doesActivity*) e tem dúvidas ou erros (*hasDoubts_Errors*). Estes dois *facets* ligam as classes *Identification* e *Activity*. Além disso, um *RecommendationResourcesAgent* (um tipo de agente adaptador de recursos) recomenda recursos (*recommends_resources*), por exemplo, APs de acordo com o perfil do estudante e as atividades que mais se ajustam a um determinado perfil de aluno (*gets_best_fit_resources*, que vem da classe *Activity*).

A Tabela 6 descreve os elementos (propriedades ou slots) de cada categoria (classes filhas de *Learner_LIP*) do padrão LIP que foram utilizados na ontologia, além das 3 categorias adicionadas ao padrão, *CCPP_Device*, *Learning_Styles* e *Frequency*. Algumas propriedades possuem explicações mais detalhadas para melhor entendimento. É importante ressaltar que os campos da categoria *Activity* foram definidos para abrigar tanto atividades como recursos. Os campos que dizem respeito somente a recursos (que começam com *res_*) possuem valores nulos quando um objeto *Activity* é, de fato, apenas uma atividade e não um recurso adaptado.

Tabela 6. Elementos utilizados em cada categoria do padrão LIP.

Categoria do padrão LIP	Elementos utilizados
<i>Identification</i>	id, name, address, contactinfo, agent, agree_disagree_OLM
<i>Accessibility</i>	preference, extension (para acesso seguro no ambiente)
<i>Interest</i>	Description
<i>Activity</i> *	id, typename, contentype, date, testimonial, keyword, evaluation, description, startdate, lom_title, lom_interactivity_type, lom_difficulty, lom_interactivity_level, lom_duration, lom_learning_resource_type, lom_format, lom_structure, lom_language, type_res (identificando se é um recurso, uma atividade ou uma atividade e recurso ao mesmo tempo), res_comp_skills ⁽¹⁾ , res_learning_styles ⁽²⁾ , res_interests ⁽³⁾ , weight_c (peso da atividade), attempts_k (número de tentativas), res_utility_level, res_level_satisfaction, extension (livres extensões futuras definidas pelos projetistas de sistemas) (1) competências subdivididas em habilidades específicas requeridas e seus níveis para execução da atividade em função de um curso específico

	<p>(2) níveis de estilos de aprendizagem necessários para a recomendação de um recurso</p> <p>(3) recursos que possuem dados de interesses de usuários a quem devem ser recomendados</p> <p>*Os elementos que começam com o prefixo lom_ são campos do padrão LOM usados para as atividades e recursos.</p>
<i>Competency</i>	contentype, description (em concordância com o padrão IMS RDCEO), extension (neste caso, as extensões são as competências subdivididas em habilidades específicas e seus níveis para cada aluno)
<i>CCPP_Device</i>	version, display_width, display_height, name_browser, name_device, uses_mobile_device, vendor
<i>Learning_Styles</i>	fs_ls_sensing_intuitive, fs_ls_active_reflective, fs_ls_visual_verbal, fs_ls_sequential_global, hm_ls_pragmatist, hm_ls_theorist, hm_ls_activist, hm_ls_reflector, extension (para outros estilos de aprendizagem)
<i>Frequency</i>	last_date_access, number_of_access, duration_execution_activity, d_freq, w_freq

Fonte: o autor.

Quanto aos agentes de modelo de aluno (*LearnerModelAgent*) e agentes de adaptação, ou recomendação de recursos (*RecommendationResourcesAgent*) mostrados na Figura 27, a Tabela 7 mostra as propriedades desses tipos de agentes.

Tabela 7. Propriedades utilizadas em cada tipo de agente.

Tipo de Agente	Propriedades (ou slots)
<i>LearnerModelAgent</i>	core_func (função núcleo ou principal do agente), comm_element (elemento de comunicação)
<i>RecommendationResourcesAgent</i>	core_func (função núcleo ou principal do agente), comm_element (elemento de comunicação)

Fonte: o autor.

A Tabela 8, a seguir, mostra as relações existentes entre as classes da ontologia de MAA da Figura 27 e as relações que fazem parte das funções de agente da Seção 4.4 (que não estão na Figura 27). A Classe 1 é de onde a seta parte e a Classe 2, onde a seta alcança.

Tabela 8. Relações entre as classes da ontologia de MAA.

Relação	Classe 1	Classe 2	Descrição
<i>hasInterests</i>	<i>Identification</i>	<i>Interest</i>	Estabelece que todo aluno pos-

			sui pelo menos um interesse
<i>has_preferences</i>	<i>Identification</i>	<i>Accessibility</i>	Estabelece que todo aluno possui dados de credenciais e preferências de acessibilidade no AVA
<i>obtains_frequency</i>	<i>Identification</i>	<i>Frequency</i>	Estabelece que os dados de frequência do aluno são sempre atualizados a cada acesso no AVA
<i>has_learning_styles</i>	<i>Identification</i>	<i>Learning_Styles</i>	Estabelece que cada aluno possui estilos de aprendizagem
<i>has_device</i>	<i>Identification</i>	<i>CCPP_Device</i>	Estabelece que um aluno usa um determinado equipamento ou dispositivo
<i>gets_data_device</i>	<i>LearnerModelAgent</i>	<i>CCPP_Device</i>	Estabelece que os agentes de modelo de aluno obtêm acesso aos metadados do equipamento do aluno
<i>updates_data</i>	<i>Learner_LIP</i>	<i>LearnerModelAgent</i>	Estabelece que os dados do modelo do aluno são atualizados por um agente de modelo de aluno
<i>doesActivity</i>	<i>Identification</i>	<i>Activity</i>	Estabelece que um aluno é responsável por fazer uma ou mais atividades
<i>hasDoubts_Errors</i>	<i>Identification</i>	<i>Activity</i>	Estabelece que um aluno possui dúvidas e pode cometer erros em uma atividade
<i>hasSkillsCompetencies</i>	<i>Identification</i>	<i>Competency</i>	Estabelece que cada aluno possui determinadas competências e habilidades
<i>gets_learning_styles</i>	<i>Learning_Styles</i>	<i>LearnerModelAgent</i>	Estabelece que os agentes de modelo de aluno têm acesso aos estilos de aprendizagem de cada aluno
<i>gets_initial_skills_comp etencies</i>	<i>Competency</i>	<i>LearnerModelAgent</i>	Estabelece que os agentes de modelo do aluno acessam os

			dados dos alunos relativos às suas competências e habilidades iniciais
<i>Hasrequiredskills</i>	<i>Activity</i>	<i>Competency</i>	Estabelece que cada atividade possui habilidades necessárias para serem executadas
<i>analyses_answers_activities</i>	<i>Activity</i>	<i>LearnerModelAgent</i>	Estabelece que os agentes de modelo de aluno corrigem as atividades e/ou analisam as notas dos alunos obtidas, a fim de atualizar os níveis de habilidades dos mesmos
<i>gets_new_activities</i>	<i>Activity</i>	<i>LearnerModelAgent</i>	Estabelece que os agentes de modelo de aluno obtêm dados a respeito de cada nova atividade lançada no AVA
<i>gets_best_fit_resources</i>	<i>Activity</i>	<i>RecommendationResourcesAgent</i>	Estabelece que os agentes de adaptação ou recomendação de recursos obtêm os melhores recursos a serem adaptados ou recomendados para os alunos no AVA
<i>gets_doubts_profile</i>	<i>Learner_LIP</i>	<i>RecommendationResourcesAgent</i>	Estabelece que os agentes de adaptação ou recomendação de recursos obtêm informações a respeito dos alunos que estão com alguma dificuldade ou dúvida no AVA
<i>recommends_resources</i>	<i>RecommendationResourcesAgent</i>	<i>Learner_LIP</i>	Estabelece que os agentes de adaptação ou recomendação de recursos devem recomendar ou disponibilizar os recursos educacionais de acordo com o perfil de cada aluno no AVA
<i>gets_better</i>	<i>RecommendationResourcesAgent</i>	<i>Activity</i>	Estabelece que um recurso só pode ser recomendado para um aluno quando ele é o melhor naquele momento, segundo os 3

			critérios da Subseção 4.4.3.
<i>fin_get</i>	<i>Learning_Styles, Interest</i>	<i>InitialProfileAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fin_calculates</i>	<i>Activity</i>	<i>InitialProfileAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fpa_get</i>	<i>Frequency</i>	<i>LearningAssessmentAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fpa_analyses</i>	<i>Competency</i>	<i>LearningAssessmentAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fpa_calculates</i>	<i>Activity</i>	<i>LearningAssessmentAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fup_accessed</i>	<i>Activity, Identification</i>	<i>UpdateProfileAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fup_save</i>	<i>UpdateProfileAgent</i>	<i>Learner_LIP</i>	Função de agente descrita na Subseção 4.4.1
<i>fup_evaluation_pre_pos_resource</i>	<i>Activity, Identification</i>	<i>UpdateProfileAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fup_feedback_satisfied</i>	<i>Activity, Identification</i>	<i>UpdateProfileAgent</i>	Função de agente descrita na Subseção 4.4.1
<i>fnot_send</i>	<i>NotifierActivityAgent</i>	<i>Identification</i>	Função de agente descrita na Subseção 4.4.1
<i>fsit_checks_and_combines</i>	<i>Learner_LIP</i>	<i>Profile_Situation_Assessment_Agent</i>	Função de agente descrita na Subseção 4.4.2
<i>fa-dapt_executes_strategy</i>	<i>Recommended_Resources_Agent</i>	<i>Identification</i>	Função de agente descrita na subseção 4.4.2

Fonte: o autor.

A seguir, seguem alguns axiomas presentes na ontologia. Foram utilizados os quantificadores existenciais, universais e outros símbolos e regras presentes na lógica de primeira ordem, de acordo com a Tabela 8 e a Figura 27.

Os axiomas que fazem parte da ontologia e que estão diretamente relacionados ao modelo do aluno são:

1. Toda competência é formada por uma ou mais habilidades:

$$(\forall x) \text{Competency}(x) \rightarrow (\exists y) \text{Skills}(y) \wedge \text{Composes}(x,y)$$

2. Uma habilidade pode estar associada a mais de uma competência:

$$(\forall x) \text{Skills}(x) \rightarrow (\exists y) \text{Competency}(y) \wedge \text{Associated}(x,y)$$

3. Se um estudante tem uma competência, logo ele tem uma habilidade:

$$(\forall x \forall y) \text{Competency}(x) \wedge \text{Learner_LIP}(y) \wedge \text{hasSkillsCompetencies}(y,x) \rightarrow (\exists z) \text{Skills}(z) \wedge \text{hasSkillsCompetencies}(y,z)$$

4. Todo aluno tem o seu modelo no padrão LIP. Sendo assim, todo aluno pode ser visto como uma instância do modelo LIP:

$$(\forall x) \text{Learner_LIP}(x) \leftrightarrow \text{Identification}(x)$$

5. Um agente que atualiza os dados do aluno no AVA (UpdateProfileAgent) é um agente do tipo Agente de Modelo de Aluno:

$$(\forall x) \text{UpdateProfileAgent}(x) \rightarrow \text{LearnerModelAgent}(x)$$

6. Todo aluno tem seus dados atualizados no ambiente pelo UpdateProfileAgent:

$$(\forall x \forall y) \text{Learner_LIP}(x) \wedge \text{UpdateProfileAgent}(y) \rightarrow \text{updates_data}(y,x)$$

7. Todo aluno possui pelo menos um interesse:

$$(\forall x) \text{Learner_LIP}(x) \rightarrow (\exists y) \text{Interest}(y) \wedge \text{hasInterests}(x,y)$$

8. Todo aluno possui dados de acessibilidade no AVA, como preferências e credenciais:

$$(\forall x) \text{Learner_LIP}(x) \rightarrow (\exists y) \text{Accessibility}(y) \wedge \text{has_preferences}(x,y)$$

9. Um aluno é muito frequente no AVA se ele faz mais de 3 acessos por dia, é normalmente frequente se acessa de 3 a 1 vezes por dia, pouco frequente se acessa pelo menos 1 vez por semana e não frequente, caso contrário:

$$(\forall x \forall y) \text{Learner_LIP}(x) \wedge \text{Frequency}(y) \wedge (\text{obtains_frequency}(\text{A_DAY}, x, y) > 3) \leftrightarrow \text{high_frequency}(x, y)$$

$$(\forall x \forall y) \text{Learner_LIP}(x) \wedge \text{Frequency}(y) \wedge (1 \leq \text{obtains_frequency}(\text{A_DAY}, x, y) \leq 3) \leftrightarrow \text{normal_frequency}(x, y)$$

$$(\forall x \forall y) \text{Learner_LIP}(x) \wedge \text{Frequency}(y) \wedge (\text{obtains_frequency}(\text{A_WEEK}, x, y) \geq 1) \wedge \neg \text{high_frequency}(x, y) \wedge \neg \text{normal_frequency}(x, y) \leftrightarrow \text{low_frequency}(x, y)$$

$$(\forall x \forall y) \text{Learner_LIP}(x) \wedge \text{Frequency}(y) \wedge \neg \text{high_frequency}(x, y) \wedge \neg \text{normal_frequency}(x, y) \wedge \neg \text{low_frequency}(x, y) \leftrightarrow \text{no_frequency}(x, y)$$

10. Cada estudante possui seus estilos de aprendizagem:

$$(\forall x) \text{Learner_LIP}(x) \rightarrow (\exists y) \text{Learning_Styles}(y) \wedge \text{has_learning_styles}(x, y)$$

11. Cada estudante possui seu próprio dispositivo (ou equipamento que utiliza para acessar o AVA)

$$(\forall x) \text{Learner_LIP}(x) \rightarrow (\exists y) \text{CCPP_Device}(y) \wedge \text{has_device}(x, y)$$

12. Um agente que acessa os dados do dispositivo ou equipamento que o estudante utiliza para acessar o AVA (DeviceIdentifierAgent) é um agente do tipo Agente de Modelo de Aluno:

$$(\forall x) \text{DeviceIdentifierAgent}(x) \rightarrow \text{LearnerModelAgent}(x)$$

$$(\forall x \forall y) \text{DeviceIdentifierAgent}(x) \wedge \text{CCPP_Device}(y) \rightarrow \text{gets_data_device}(x, y)$$

13. Todo estudante é responsável por executar uma ou mais atividades no AVA:

$$(\forall x \exists y) \text{Learner_LIP}(x) \wedge \text{Activity}(y) \rightarrow \text{doesActivity}(x, y)$$

14. Um agente que configura os níveis iniciais de habilidades para um aluno (InitialSkillsAgent) é um agente do tipo Agente de Modelo de Aluno:

$$(\forall x) \text{InitialSkillsAgent}(x) \rightarrow \text{LearnerModelAgent}(x)$$

15. Um agente que avalia as respostas dos alunos no AVA (ActivityAssessmentAgent), atribuindo-lhe notas, é um agente do tipo Agente de Modelo de Aluno:

$$(\forall x) \text{ActivityAssessmentAgent}(x) \rightarrow \text{LearnerModelAgent}(x)$$

16. O ActivityAssessmentAgent e o InitialSkillsAgent enviam dados de atualização ao UpdateProfileAgent, por meio de troca de mensagens:

$$(\forall x \forall y \forall z \forall w) \text{Learner_LIP}(x) \wedge \text{ActivityAssessmentAgent}(y) \wedge \text{InitialSkillsAgent}(z) \wedge \text{analyses_answers_activities}(x, y) \wedge \text{gets_initial_skills_competencies}(x, z) \rightarrow (\exists w) \text{UpdateProfileAgent}(w) \wedge \text{updates_data}(x, w)$$

Os axiomas definidos na ontologia que possuem relação com o processo de recomendação de recursos no AVA são:

17. Cada atividade possui uma ou mais habilidades necessárias para serem executadas:

$$(\forall x) \text{Activity}(x) \rightarrow (\exists y) \text{Skills}(y) \wedge \text{hasrequiredskills}(x,y)$$

18. Um recurso só pode ser recomendado se ele for o melhor para aquele perfil, de acordo com as regras da subsecção 4.4.3

$$(\forall w \forall x \forall y \forall z) \text{Learner_LIP}(x) \wedge \text{Activity}(y) \wedge \text{Interest}(w) \wedge \text{RecommendationResourcesAgent}(z) \wedge \text{recommends_resources}(x,y,w,z) \leftrightarrow \text{gets_better}(y,x,w).$$

19. Um agente RecommendationResourcesAgent verifica os alunos que têm dúvidas ou dificuldades no AVA e recomenda os melhores recursos a eles, em função de seus estilos de aprendizagem, competências e interesses:

$$(\forall x \forall y \forall z) \text{Learner_LIP}(x) \wedge \text{Activity}(y) \wedge \text{hasDoubts_Errors}(x,y) \wedge \text{RecommendationResourcesAgent}(z) \wedge \text{gets_doubts_profile}(z,y,x) \rightarrow (\exists w \exists t \exists u) \text{Interest}(w) \wedge \text{Competency}(t) \wedge \text{Learning_Styles}(u) \wedge \text{recommends_resources}(z,y,w,x).$$

Os agentes, juntamente com as regras, axiomas e definições oferecidas pela ontologia de MAA, compõem o módulo de adaptação de recursos no ArCARE. É importante ressaltar que a ontologia de MAA aqui descrita é extensível e pode ser modificada de acordo com as necessidades de cada projeto baseado no ArCARE.

4.6 Comparação do ArCARE com Outras Abordagens

Nas seções anteriores, vimos as definições para os elementos gerais do ArCARE. Na Tabela 9, relacionamos os aspectos que os sistemas baseados no ArCARE podem ter, em comparação com os trabalhos mostrados no Capítulo 3, repetindo a Tabela 2, mas acrescentando uma coluna que mostra os aspectos que o ArCARE lida. Na Tabela 9, os números de I a XIII referem-se aos ambientes e propostas visitados, respectivamente, das seções 3.1 a 3.13, sendo: (I) i-Collaboration; (II) Fasihuddin; (III) Oscar CITS; (IV) COMTEXT; (V) Graf e Kinshuk; (VI) Ahmad; (VII) Martins e Carrapatoso; (VIII) Schreurs e Al-Huneidi; (IX) Menezes; (X) Shaoling e Fangfang; (XI) PULP; (XII) MorFEu; (XIII) OSSM; e, por fim, colocamos como (XIV) os sistema baseados no ArCARE. A marcação "X" indica que foram encontradas, dentro dos trabalhos analisados, evidências da presença do aspecto.

Tabela 9. Resumo dos Aspectos Destacados no ArCARE comparados com os trabalhos e propostas visitados.

Aspecto	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV
Adaptação de	X	X	X	X	X	X	X	X		X	X	X	X	X

recursos														
Sistema Multi-agente	X	X	X							X	X			X
Perfil/Modelo de aluno	X	X	X	X	X	X		X		X	X		X	X
MAA						X							X	X
Sensibilidade ao contexto	X			X						X	X			X
m-learning				X						X	X			X
Estilos de aprendizagem		X	X		X		X							X
Competências				X		X	X	X					X	X
Arquiteturas pedagógicas												X		X
Aprendizagem Colaborativa	X			X					X	X	X	X		X
Histórico de ações dos usuários													X	X
Learning Analytics													X	X

Fonte: o autor.

4.7 Conclusões do Capítulo

Qualitativamente, o ArCARE amplia o escopo de aplicações que permitem adaptações de recursos que podem ser feitos em AVAs, STIs e SLEs. Com a discussão acima, com os aspectos que o ArCARE lida, apresentados na Tabela 9, além dos conceitos e técnicas de adaptação de recursos em AVAs sugeridos em Bremgartner et al. (2015) e baseados na proposição inicial elaborada por Bremgartner et al. (2014) e nas análises dos protótipos elaborados em Bremgartner et al. (2014b) e em Bremgartner et al. (2015), definimos Adaptação de Recursos Educacionais como “a ação de adaptar e/ou recomendar os recursos educacionais (e. g., atividades passadas, entrega de objetos de aprendizagem, mudanças na interface e auxílio personalizado) em AVAs de acordo com as características do estudante”. E um Arcabouço Conceitual de Adaptação de Recursos Educacionais é um framework que apoia a criação, manutenção e ampliação de sistemas ou ambientes educacionais, AVAs, STIs e SLEs que possuem técnicas de

adaptação de recursos educacionais em função do Modelo Aberto de Aluno. Esse framework é composto pela ontologia de MAA adicionado aos agentes de modelo de aluno e agentes de adaptação (ou recomendação) de recursos. Além disso, esse arcabouço é centrado nos usuários do ambiente (estudantes, aprendizes, funcionários, professores, administradores, tutores), utilizando-se como base pedagógica o Construtivismo de Piaget.

Vimos, portanto, nos capítulos anteriores, os aspectos teóricos e conceitos que fazem parte do ArCARE. E neste capítulo, o arcabouço conceitual (ou framework). E nos dois capítulos seguintes, serão apresentadas as provas de conceito (instanciações) do uso do arcabouço aqui definido.

CAPÍTULO 5 – SISTEMA AMPARA DE RECOMENDAÇÃO DE RECURSOS

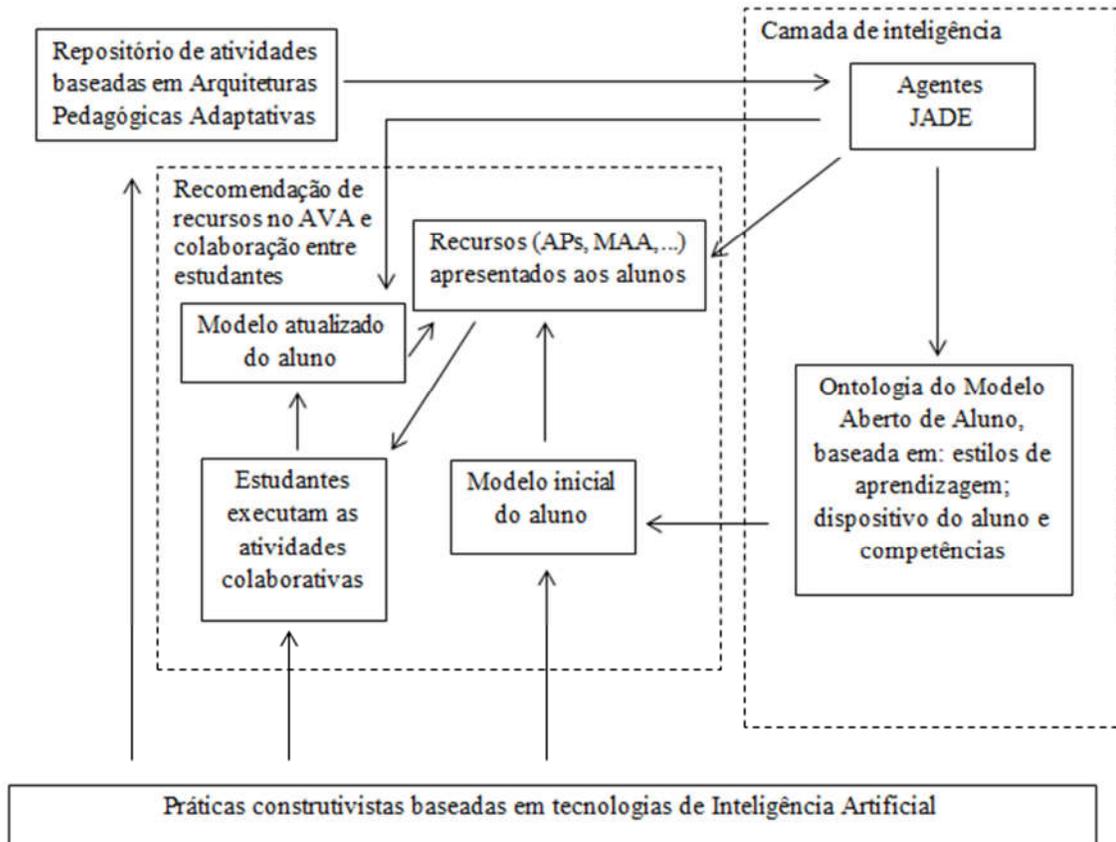
Este capítulo tem por objetivo apresentar uma instanciação do ArCARE como prova de conceito, o sistema AMPARA, em cursos que utilizam o AVA Moodle. Testes foram aplicados em turmas reais e seus resultados são mostrados.

Para validarmos as propostas descritas na definição do ArCARE no Capítulo 4, realizamos uma simulação do Cenário 1, detalhado no Capítulo 1. O cenário resumidamente trata da adaptação de recursos educacionais por recomendação de atividades, desafios abertos, colegas que possam auxiliar outros estudantes a resolver seus problemas ou até mesmo Arquiteturas Pedagógicas em um AVA, por meio de feedbacks e técnicas que permitem construir o conhecimento do aluno que esteja com dúvidas em atividades. O cenário em questão está relacionado ao problema da recomendação de recursos educacionais para estudantes com dificuldades em AVAs, que é abordado por Bremgatner, Netto e Menezes (2014a), Bremgatner, Netto e Menezes (2014b) e Bremgatner, Netto e Menezes (2015b), tendo como base as competências e habilidades dos estudantes e seus resultados nas atividades propostas, informações estas que compõem seu MAA.

5.1 Modelo Conceitual, Arquitetura e Implementação do AMPARA

Uma das instanciações do ArCARE, como prova de conceito e que lida com o problema descrito no Cenário 1, foi o desenvolvimento do sistema AMPARA (do inglês, *Adaptive Multi-agent Pedagogical Architectures for Resources and Activities in VLEs*, que no português fica Arquiteturas Pedagógicas Multiagente Adaptativas para Recursos e Atividades em AVAs). O modelo conceitual do AMPARA é mostrado na Figura 28. Neste modelo, é mostrado em detalhes o processo de personalização de atividades colaborativas baseadas em Arquiteturas Pedagógicas em função do MAA, fazendo uso do sistema multiagente. Os agentes foram desenvolvidos no framework baseado na linguagem Java para a construção de agentes de software JADE (2017) (Java Agent Development Framework).

Figura 28. Modelo Conceitual do AMPARA.



Fonte: o autor.

O AMPARA foi aplicado em duas turmas: uma de Cálculo Numérico na Universidade Federal do Amazonas (UFAM) e uma de Algoritmos e Programação no Instituto Federal do Amazonas (IFAM).

Nesse modelo conceitual da Figura 28, o processo de adaptação de recursos começa quando o professor prepara atividades, recursos e registra estudantes no AVA. O estudante, por sua vez, interage com o AVA, seja cadastrando os seus dados pessoais, postando algo no fórum ou resolvendo atividades propostas no ambiente. Para este estudo de caso, utilizamos o AVA Moodle (2017). Além disso, no AVA foram criados três questionários para os alunos responderem no início do curso, com o objetivo de obter o perfil e o modelo de aluno inicial, porque esta é a primeira informação que compõe a base de conhecimento sobre os alunos no banco de dados. Os questionários são: 1) Índice de Estilos de Aprendizagem (*Index of Learning Styles*), buscando conhecer os estilos de aprendizagem de Felder-Silverman (1988) dos estudantes; 2) Questionário de Estilos de Aprendizagem de Honey-Alonso (Alonso e Gallego, 2017). Optou-se por escolher estes dois questionários visando ter uma informação mais preci-

sa a respeito dos alunos quanto aos seus estilos de aprendizagem, obtidos pela combinação dos dois questionários; 3) Questionário específico de habilidades e competências relacionadas à disciplina que o aluno está cursando, por exemplo, sobre algoritmos e programação de computadores, onde os alunos respondem a perguntas sobre o seu nível de conhecimento em conceitos de programação referentes a disciplina de Algoritmos e Programação (estruturas condicionais; repetição; estruturas de dados; nível de experiência no uso da linguagem de programação da disciplina – a linguagem C; experiência de trabalhar com linguagens de programação no ramo profissional). Na disciplina de Cálculo Numérico, as perguntas foram sobre programação de computadores, onde os alunos responderam a perguntas sobre o seu nível de conhecimento em conceitos de programação referentes a disciplina de Cálculo Numérico (estruturas condicionais; repetição; estruturas de dados; nível de experiência no uso de uma linguagem de programação, escolhido para esta disciplina o Scilab (2017), e a experiência de usar o computador em Cálculo Numérico). Os links para acessar estes questionários e para acessar o MAA ficam disponíveis no AVA. O AMPARA é uma extensão dos trabalhos de Bremgartner, Netto e Menezes (2014a); Bremgartner, Netto e Menezes (2014b). Nessa aplicação, não trabalhamos com os dados de histórico de interações dos alunos com o AVA, pois o AMPARA foi um projeto-piloto que utilizou os conceitos e estruturas fornecidas pelo ArCARE e dados de histórico de alunos ainda não eram considerados nessa fase inicial.

Os três questionários passados aos alunos foram disponibilizados no ColabWeb (colabweb.ufam.edu.br), que é um AVA baseado no Moodle em sua versão 2.6 que contém diversas disciplinas pertencentes ao Instituto de Computação (IComp) da Universidade Federal do Amazonas que podem ser acessados online. O ColabWeb foi utilizado para a complementação do AMPARA, uma vez que os serviços de repositório de recursos, login de usuários, entre outros, já estão prontos neste AVA. Inicialmente, a disciplina escolhida para os questionários foi a de Cálculo Numérico, composta por 33 alunos da UFAM e depois os questionários foram aplicados na turma de Algoritmos e Programação, composta por 49 alunos do curso de Mecatrônica do Instituto Federal do Amazonas Campus Manaus Distrito Industrial (IFAM CMDI). A Figura 29 mostra os questionários disponíveis para os alunos acessarem no ColabWeb.

Figura 29. Questionário no ColabWeb mostrado a alunos na disciplina Cálculo Numérico.

Tópico 2

Questionário 01. Índice de Estilos de Aprendizagem (Index of Learning Styles)

Instruções:
Prezado estudante, neste questionário queremos saber mais sobre seus estilos de aprendizagem e características cognitivas com o objetivo de prover um curso adaptado às suas necessidades. As informações passadas por você neste questionário fazem parte de uma pesquisa, com finalidade experimental, e serão mantidas em sigilo. Este questionário não precisa ser preenchido completamente de uma só vez. Caso deseje, você pode interromper o questionário e continuar em outro momento acessando o ColabWeb. Este questionário é qualitativo, e não quantitativo. Apesar de valer pontuação extra na disciplina de Cálculo Numérico, esta pontuação é para quem completar o questionário. Ou seja, o questionário deve ser respondido completamente, mas não existe resposta correta. Portanto, como ele é pessoal, procure ser o mais sincero possível. Além disso, despreze a mensagem "Método de avaliação: Nota mais alta". Isto aparece porque para este questionário utilizamos o questionário padrão do ColabWeb.
Dúvidas, favor enviar e-mail para vitorbref@gmail.com
Total de Perguntas: 44

Questionário 02. Questionário de Estilos de Aprendizagem de Honey-Alonso

Instruções:
Este questionário está sendo aplicado para identificar seu estilo preferido de aprendizagem de Honey-Alonso. Não existem respostas corretas nem erradas. Será útil na medida que seja sincero(a) em suas respostas. Se está mais de acordo que em desacordo com o item coloque um (+). Se está mais de desacordo que em acordo com o item coloque um (-). As informações passadas por você neste questionário fazem parte de uma pesquisa, com finalidade experimental, e serão mantidas em sigilo. Este questionário não precisa ser preenchido completamente de uma só vez. Caso deseje, você pode interromper o questionário e continuar em outro momento acessando o ColabWeb. Este questionário é qualitativo, e não quantitativo. Apesar de valer pontuação extra na disciplina de Cálculo Numérico, esta pontuação é para quem completar o questionário. Ou seja, o questionário deve ser respondido completamente, mas não existe resposta correta. Portanto, como ele é pessoal, procure ser o mais sincero possível. Além disso, despreze a mensagem "Método de avaliação: Nota mais alta". Isto aparece porque para este questionário utilizamos o questionário padrão do ColabWeb.
Dúvidas, favor enviar e-mail para vitorbref@gmail.com
Total de Perguntas: 80

Questionário 03. Habilidades Específicas do Curso

Instruções:
Prezado estudante, neste questionário queremos saber mais sobre suas habilidades específicas relacionadas a este curso, Cálculo Numérico, especialmente a respeito de programação de computadores e outras questões relevantes. Isto será útil para adaptar o curso de acordo com suas características. As informações passadas por você neste questionário fazem parte de uma pesquisa, com finalidade experimental, e serão mantidas em sigilo. Este questionário não precisa ser preenchido completamente de uma só vez. Caso deseje, você pode interromper o questionário e continuar em outro momento acessando o ColabWeb. Este questionário é qualitativo, e não quantitativo. Apesar de valer pontuação extra na disciplina de Cálculo Numérico, esta pontuação é para quem completar o questionário. Ou seja, o questionário deve ser respondido completamente, mas não existe resposta correta. Portanto, como ele é pessoal, procure ser o mais sincero possível. Além disso, despreze a mensagem "Método de avaliação: Nota mais alta". Isto aparece porque para este questionário utilizamos o questionário padrão do ColabWeb.
Dúvidas, favor enviar e-mail para vitorbref@gmail.com
Total de Perguntas: 7

Fonte: o autor.

O questionário ficou disponível cerca de um mês para os alunos acessarem e lançarem suas respostas. Após os alunos responderem o questionário, seus perfis iniciais foram obtidos, com os resultados dos questionários calculados pelo próprio AVA.

Por sua vez, a arquitetura geral do AMPARA, desenvolvida a partir do seu modelo conceitual, é mostrada na Figura 30. Os agentes JADE manipulam os dados dos alunos e seus modelos de aluno no banco de dados do AVA, executando atualizações, neste caso, atuando como agentes de modelo do aluno. O modelo de aluno é descrito por sua ontologia, que também é útil para a manipulação correta de troca de mensagens entre os agentes. Por sua vez, os agentes de adaptação e recomendação de recursos, a partir dos dados obtidos do modelo de aluno, selecionam recursos educacionais contidos no repositório (que fica no banco de dados), como por exemplo, APs, para adaptar o AVA. Finalmente, o aluno acessa o AVA adaptado em função de seu perfil e o seu MAA. Os agentes desenvolvidos no JADE e que são responsáveis por manipular os dados do modelo do aluno, são:

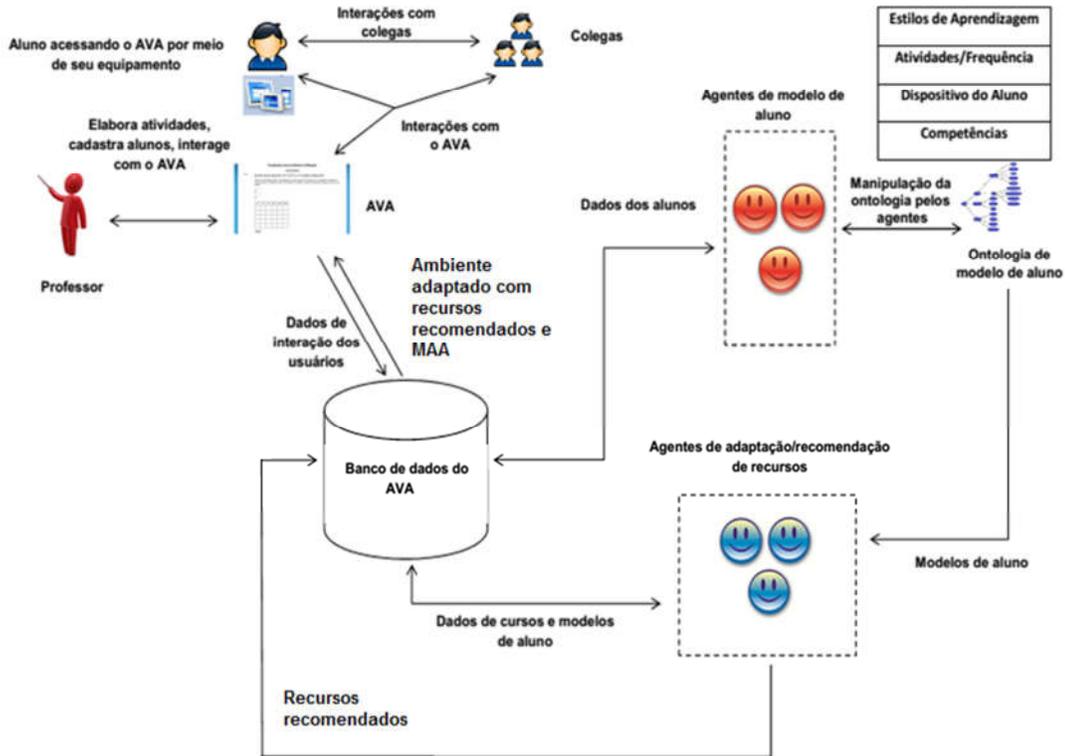
- *Initial Profile Agent*: configura os níveis de competências, habilidades e estilos de aprendizagem iniciais dos alunos em função das respostas dos alunos dos 3 questionários respondidos no início do curso e envia estas informações ao *Update Profile Agent*.

As funções f_{in} utilizadas aqui foram *obtem()* e *calcula()*, mas não considerando o histórico de ações dos usuários;

- *Learning Assessment Agent*: responsável pela avaliação das atividades respondidas pelos alunos, e envia as pontuações dos alunos ao *Update Profile Agent*. As funções f_{av} utilizadas aqui foram *calcula()* e *analisa()*;
- *Update Profile Agent*: responsável por atualizar os dados dos modelos de alunos mediante as interações dos mesmos com o ambiente, mediante as informações oriundas do *Initial Profile Agent*, *Learning Assessment Agent* e *Frequency Evaluation Agent*. A única função f_{at} utilizada aqui foi *salvar()*;
- *Help Tutor Agent*: uma instância do Agente de Notificação apresentado na subseção 4.4.1. Agente notificador de novas atividades passadas pelo professor, a fim de manter o aluno informado acerca de novas atividades, pela função f_{not} *envia()*;
- *Frequency Evaluation Agent*: agente que verifica a frequência do aluno no AVA. Convencionamos aqui que um aluno é muito frequente no AVA se ele faz mais de 3 acessos por dia, é normalmente frequente se acessa de 3 a 1 vez por dia, regularmente frequente se acessa pelo menos 1 vez por semana e não frequente caso contrário. Os dados de frequência do aluno serão enviados ao *Update Profile Agent*. A função f_{av} utilizada aqui foi *obtem()*;
- *Open Learner Model Agent*: Apresenta ao professor quem são os alunos que discordam de seu modelo de aluno, a fim de que o professor possa tomar as medidas necessárias para atualização dos dados do aluno. Utilizou a função f_{at} *salvar()* para atualização;
- *Device Identifier Agent*: responsável por detectar informações a respeito do equipamento que o estudante está utilizando, se é um computador, smartphone, tablet ou qualquer outro, a fim de ajustar o AVA de acordo com as características do dispositivo que o estudante estiver utilizando. Utilizou a função f_{in} *obtem()* para detecção do dispositivo;

Os agentes desenvolvidos no JADE, responsáveis pela adaptação e recomendação de recursos aos alunos no AVA, são:

Figura 30. Arquitetura do AMPARA.



Fonte: o autor.

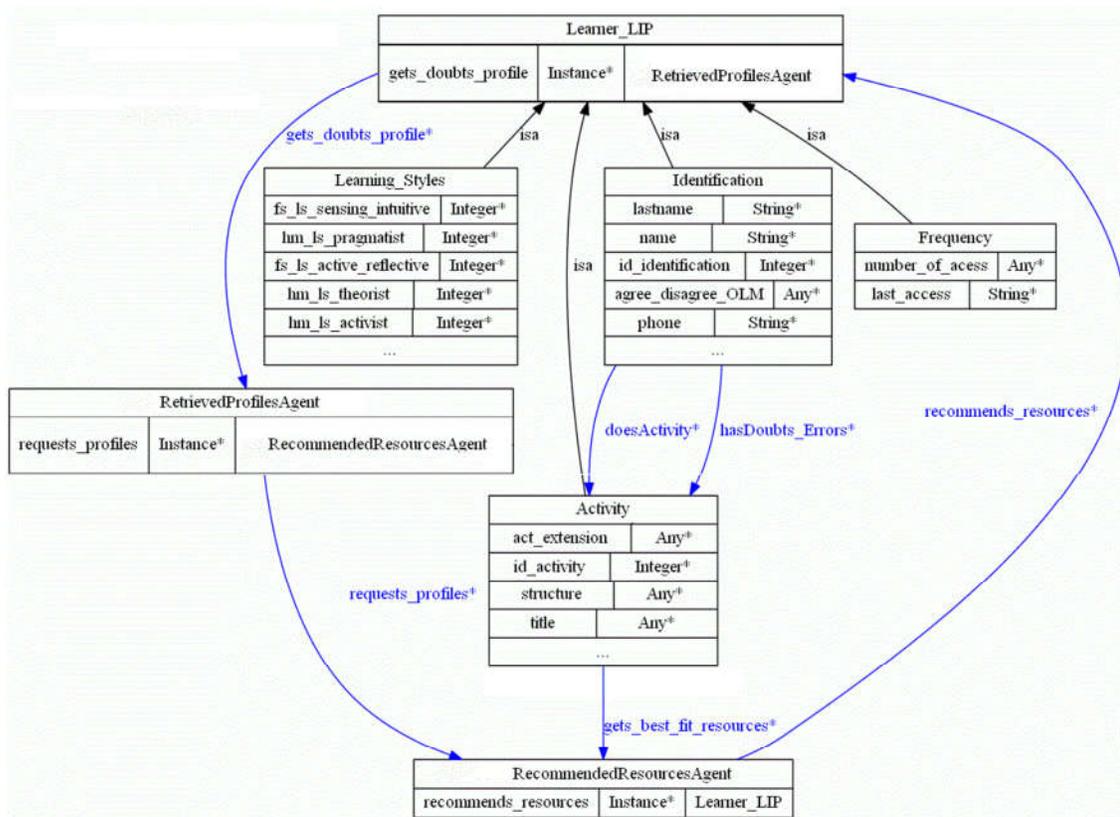
- Retrieved Profiles Agent:* agente que faz a busca por dúvidas, erros dos alunos em atividades respondidas, avalia os níveis de habilidades e competências dos alunos, busca os estilos de aprendizagem dos mesmos, além das demais informações que compõem o MAA e por fim, mapeia esses dados em recursos de aprendizagem a serem disponibilizados no AVA, adaptando-o de acordo com o MAA. Estes recursos podem ser: colegas que possam tirar dúvidas, notificações de acompanhamento do aluno, sugestão de leituras e atividades baseadas em arquiteturas pedagógicas de acordo com o perfil de estudante. Com estas informações, este agente envia uma mensagem ao *Recommended Resource Agent* os dados dos alunos com os recursos associados a eles. Utilizou a função f_{sit} *verifica_e_combina()*;
- Recommended Resources Agent:* agente que obtém as informações do *Retrieved Profiles Agent* e executa o processo de mostrar no AVA os recursos para cada aluno em função de seu MAA. Utilizou a função f_{adapt} *executa_estratégia()*.

Outra importante etapa neste estudo de caso foi construir a ontologia de domínio que representa o modelo de aluno, baseada na ontologia já existente apresentada no Capítulo 4,

com algumas modificações aqui. Para o modelo de aluno foi utilizado o padrão IMS LIP, que é um padrão largamente usado hoje em dia e que permite extensões. Originalmente, o LIP tem 11 categorias (ou classes) de dados do estudante, mas apenas cinco foram usadas (*Activity*, *Competency*, *Identification*, *Accessibility* e *Interest*). Além disso, este padrão foi estendido para as necessidades deste trabalho, em três outras categorias: *CCPP_Device*, *Learning_Styles* e *Frequency*. A categoria *CCPP_Device* descreve os dispositivos utilizados pelos alunos. Usando a categoria *Frequency*, esta descreve a frequência dos alunos no AVA e também é verificado se o aluno terminou ou não suas atividades. Finalmente, a categoria *Learning_Styles* contém informações sobre os estilos de aprendizagem de Felder-Silverman e Honey-Alonso.

Na categoria *Activity* estão as informações acerca das atividades propostas pelo professor. A ontologia de modelo de aluno na classe *Competency* descreve o nível de habilidade de cada aluno, o que é útil para identificar os alunos e seus níveis de competências e habilidades em uma disciplina ou curso. Nesse trabalho, cada competência do aluno é composta por um conjunto de habilidades específicas. Os agentes fazem avaliações frequentes das atividades dos alunos, sendo uma avaliação formativa, obtendo seu desempenho e atualizando os níveis de habilidades dos alunos. Outras categorias utilizadas são *Identification* (para os dados pessoais do aluno), *Accessibility* (dados de acessibilidade do usuário, credenciais no AVA) e *Interest* (contendo os interesses dos estudantes). A ontologia de modelo de aluno foi desenvolvida no editor Protégé e tem três classes principais: *Concept*, cujas filhas são as subclasses *AID* (descrevendo os agentes que fazem parte do sistema multiagente) e *Learner_LIP* (contendo as categorias LIP utilizadas neste trabalho, além de *CCPP_Device*, *Learning_Styles* e *Frequency*); *Predicate*, responsável pelos fatos sobre o ambiente onde os agentes atuam; e *AgentAction* (contendo as ações que os agentes realizam no AVA). Todas estas três classes permitem que a ontologia seja integrada ao sistema multiagente. A Figura 31 mostra uma representação gráfica resumida da ontologia de MAA, onde aparece apenas a subclasse *Learner_LIP* de *Concept*, que descreve o padrão LIP estendido e apresenta os *slots* (propriedades ou atributos) das demais classes. Por exemplo, na classe *Learning_Styles*, uma propriedade é *fs_ls_active_reflective*, indicando se o aluno tende a ser mais ativo ou reflexivo nos estilos de aprendizagem de Felder-Silverman.

Figura 31. Representação gráfica resumida da Ontologia de MAA usada pelo AMPARA.



Fonte: o autor.

Os 2 agentes (subclasses de *AID*) apresentados na Figura 31 são o *Retrieved Profiles Agent* e *Recommended Resources Agent*. As relações e regras entre as classes descrevem os *facets* da ontologia, representados por setas azuis e pretas (estas últimas indicando apenas a relação "isa", que significa "é um", de subclasse para superclasse). Como alguns exemplos de *facets* em setas azuis na Figura 31, o estudante faz uma atividade (*doesActivity*) e tem dúvidas ou erros (*hasDoubts_Errors*). Estes dois *facets* ligam as classes *Identification* e *Activity*. Além disso, o *Recommended Resources Agent* recomenda recursos (*recommends_resources*), por exemplo, APs de acordo com o perfil do estudante e as atividades que mais se ajustam a um determinado perfil de aluno (*gets_best_fit_resources*, que vem da classe *Activity*).

Na categoria *Activity* são informadas as atividades propostas pelo professor. A ontologia de modelo de aluno na classe *Competency* descreve o nível de habilidade de cada aluno, o que é útil para identificar os alunos e seus níveis de competências e habilidades em uma disciplina ou curso. Em nossa proposta, cada competência do aluno é composta por um conjunto de habilidades específicas (Bremgartner, Netto e Menezes, 2014b). Por exemplo, na disciplina

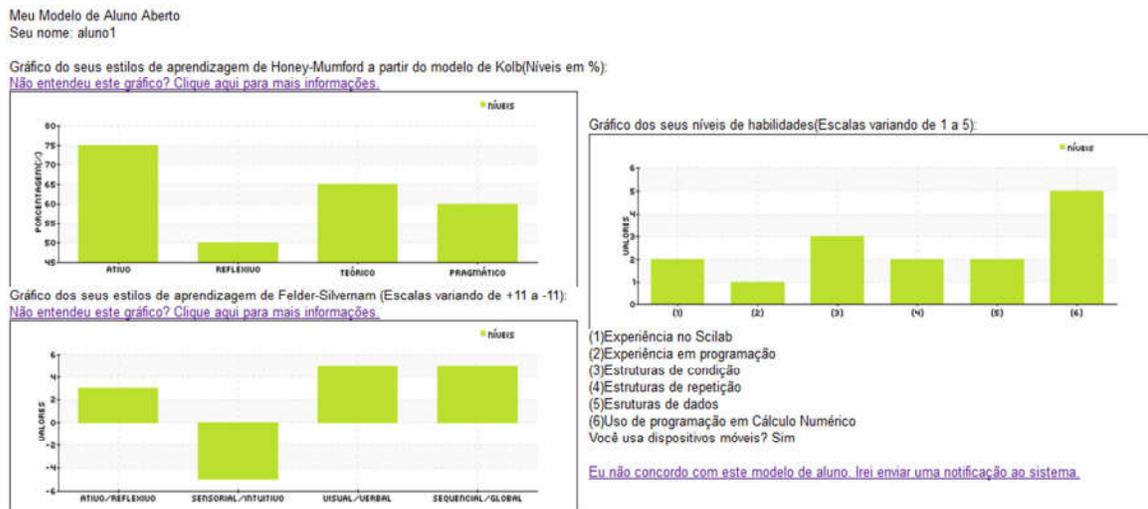
de Cálculo Numérico, a competência “Encontrar raízes de funções pelo método da bisseção” consiste nas seguintes habilidades: “Manuseio da máquina de calcular”, “Compreensão de funções (continuidade, Teorema do Valor Intermediário, gráficos de plotagem, convergência e divergência)” e “Compreender as operações do método da bisseção (escolha do intervalo inicial, a fórmula e os critérios para a escolha do novo intervalo)” (Burden e Faire, 2010). As outras seis categorias restantes do LIP não foram usadas porque elas não fazem parte do escopo deste trabalho, ou já são substituídas pelas categorias que foram utilizadas, evitando qualquer conflito de informações com dados semelhantes. Quanto às regras e axiomas-padrão definidas no Capítulo 4 na ontologia de MAA (Seção 4.5), essas foram utilizadas através da integração da ontologia com os agentes por meio do plugin *BeanGenerator* instalado no Protégé, que converte a ontologia criada em código-fonte na plataforma JADE.

Quanto aos recursos, foram utilizadas arquiteturas pedagógicas já conhecidas na literatura, como por exemplo, o debate de teses, controvérsia acadêmica e projetos de aprendizagem descritas em Castro e Menezes (2011) como resultado de recomendação de propostas de atividades pelos agentes, agregando modelagem de aluno e práticas construtivistas. Essas atividades são personalizadas de acordo com o estilo de aprendizagem do aluno. Por exemplo, para quem tem o estilo de aprendizagem mais conclusivo é melhor aplicar tarefas rápidas, com menor tempo de duração e sem muitos rodeios.

O aluno pode acessar seu modelo por meio de um link no AVA, que direciona para a página *profile.php*, integrada ao Moodle. É importante que o MAA seja facilmente compreendido pelo aluno. Assim, a Figura 32 mostra um exemplo de MAA apresentado aos alunos composto de três gráficos, que é o resultado das respostas dos alunos aos questionários. No primeiro gráfico são apresentados os estilos de aprendizagem de acordo com o questionário de Honey-Alonso, em porcentagens dos estilos *Ativo*, *Reflexivo*, *Teórico* e *Pragmático*. No segundo, os estilos de Felder-Silverman (*Ativo/Reflexivo*, *Sensorial/Intuitivo*, *Visual/Verbal* e *Sequencial/Global*) são apresentados em valores ímpares que variam de -11 a +11. Por exemplo, se um aluno tem no campo *Visual/Verbal* valor +7, isso significa que ele é mais *Visual*. Se fosse -7, ele seria mais *Verbal*. Finalmente, o terceiro gráfico mostra as habilidades do aluno em um curso específico, no nosso estudo de caso, Cálculo Numérico. Além disso, na Figura 32 é mostrado se o estudante usa ou não dispositivos móveis para acessar o AVA, e esta informação é dinamicamente alterada (pois um estudante pode estar usando ou não um dispositivo móvel em um determinado momento) e um link para as situações em que o aluno não concorda com o seu modelo de aluno mostrado. Nesse link, um aluno pode enviar uma men-

sagem para o sistema, informando que não concorda com seu modelo. Esse link é útil para detectar se há alunos que concordam ou não com seus respectivos modelos de aluno, uma vez que é possível que um estudante se julgue autossuficiente e muito capaz no curso, mas não seja considerado capaz pelas avaliações feitas pelo sistema. Com base na resposta do aluno, os agentes JADE farão uma nova atualização no modelo do aluno. Nos testes aplicados, apenas 1 aluno da disciplina Algoritmos e Programação não concordou com seu modelo de aluno.

Figura 32. Apresentando o Modelo de Aluno Aberto a estudantes.



Fonte: o autor.

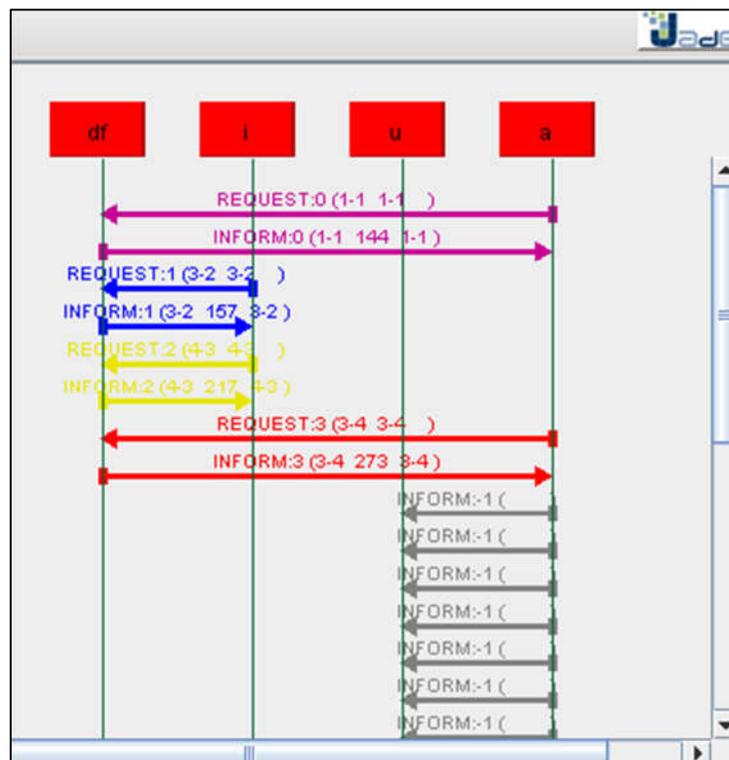
5.2 Técnicas de Adaptação Testadas

No AVA foram testadas duas técnicas de adaptação de recursos, tanto para a turma de Cálculo Numérico quanto para a turma de Algoritmos e Programação. Essas duas técnicas são a recomendação de atividades baseadas em Arquiteturas Pedagógicas e a recomendação de colegas para ajudar os alunos que tinham dúvidas em atividades.

O processo ocorre da seguinte forma: os alunos acessam o AVA, executam as atividades propostas pelo professor e dúvidas ou erros podem surgir, detectados pelos agentes. Ao mesmo tempo em que os alunos interagem com o AVA, seus modelos de aluno são atualizados no banco de dados através do sistema multiagente, que também é responsável, juntamente com a ontologia, por procurar pelos alunos com melhores níveis de habilidades específicas, capazes de ajudar o estudante que estiver com dúvidas nas atividades. Assim, os estudantes com perfis apropriados encontrados pelo sistema multiagente serão recomendados no AVA para o aluno que tem dúvidas ou errou as atividades. A Figura 33 mostra as mensagens trocadas entre agentes implementados no framework JADE no padrão FIPA. Um deles é o agente

Learning Assessment Agent (representado por “a”) e outro, o *Initial Profile Agent* (representado por “i”), que fazem requisições a um agente auxiliar chamado *Directory Facilitator* (“df”), que existe por padrão no JADE e é responsável pelo correto gerenciamento das mensagens, registrando agentes que solicitam e que fornecem serviços. Em seguida, é estabelecida a comunicação entre os agentes *Update Profile Agent* (“u”) e *Learning Assessment Agent* e *Initial Profile Agent*, a fim de que estes dois agentes enviem informações para o *Update Profile Agent*, onde este é responsável por atualizar no banco de dados do Moodle as informações relativas ao modelo do aluno.

Figura 33. Troca de mensagens entre agentes de modelo de aluno desenvolvidos no JADE.



Fonte: o autor.

Escolhemos utilizar também a correção colaborativa em pares (colegas que corrigem colegas, oferecendo feedback), a fim de simplificar o processo de correção de atividades que exigem respostas discursivas. Ao trabalharmos com Arquiteturas Pedagógicas e propostas de trabalho, utilizamos diversas perguntas, como um questionário sobre um problema, para o aluno elaborar melhor suas ideias e respostas.

A Figura 34 mostra uma notificação do *Help Tutor Agent* para um aluno que não conseguiu terminar a atividade no tempo especificado pelo professor no curso de Cálculo Numérico. Nesta mensagem de notificação, há um link para o aluno clicar para obter uma tela de

colegas recomendados pelo sistema que podem ajudá-lo. Esta tela é mostrada na Figura 35, numa tabela que indica o nome da atividade, os colegas recomendados para ajudar nessa atividade e os interesses destes colegas recomendados. Consideramos que essas informações são úteis para incentivar a interação entre os alunos. No final destes dois testes foi observada uma melhoria de 180% de acerto dos alunos. Dos 33 alunos da turma de Cálculo Numérico, apenas quatro inicialmente responderam corretamente todas as atividades propostas. Após a aplicação do recurso de recomendação, 12 alunos responderam corretamente as mesmas atividades, embora ainda seja um número menor que a metade dos alunos da turma.

Figura 34. Mensagem enviada pelo *Help Tutor Agent* ao aluno.



Fonte: o autor.

Figura 35. Estudantes recomendados.

Dúvidas em atividades? Veja aqui colegas que podem ajudar!		
Atividade	Estudantes Recomendados	Interesses
Resolver o problema do paraquedista usando Interpolação no Scilab	Aluno 2	programação
Resolver o problema do paraquedista usando Interpolação no Scilab	Aluno 3	cálculos, operar máquinas

Fonte: o autor.

A Figura 36 mostra exemplos de atividades recomendadas para alunos de Cálculo Numérico. É possível perceber que em cada atividade mostrada aos alunos, houve uma AP utilizada, neste caso, dois debates de teses e uma controvérsia acadêmica. Antes de fazerem as atividades, os alunos passaram por um certo treinamento, pois muitos não haviam experimentado no Moodle a atividade do tipo Laboratório de Avaliação (originalmente, em inglês chamada de *Workshop*). O Laboratório de Avaliação é uma atividade de avaliação entre pares (participantes) com uma vasta gama de opções. Os participantes podem avaliar os projetos de outros participantes e

exemplos de projeto em diversos modos. Este módulo também organiza o recebimento e a distribuição destas avaliações (Moodle, 2017).

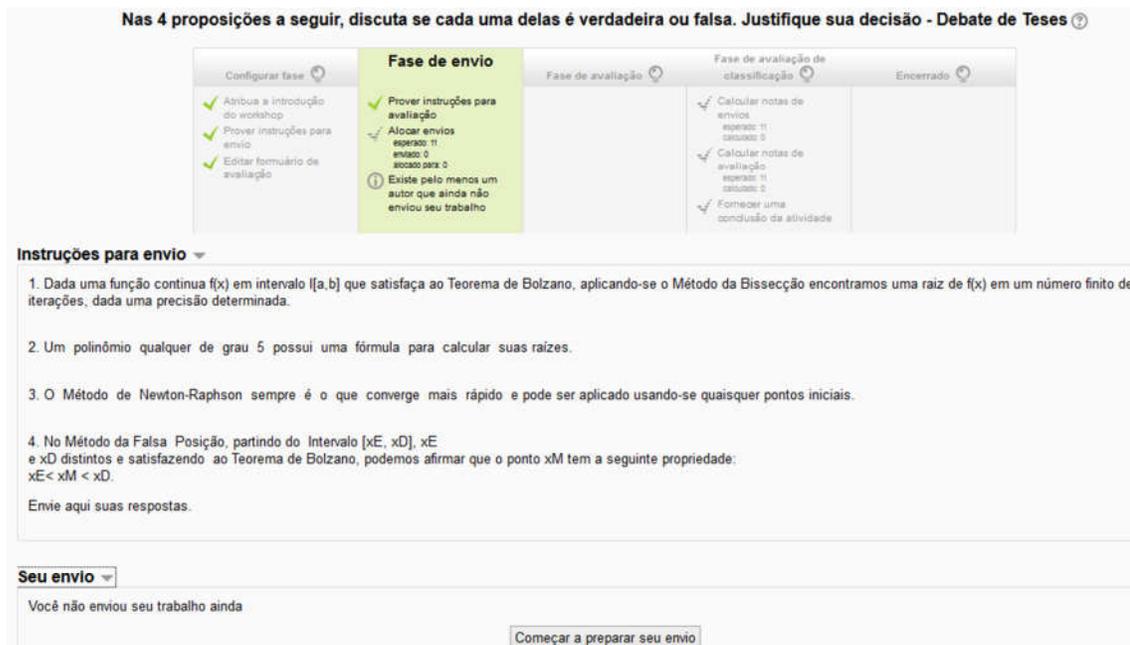
Figura 36. Exemplos de atividades recomendadas a alunos de Cálculo Numérico.



Fonte: o autor.

Em cada atividade de Laboratório de Avaliação da Figura 36, a adaptação ocorre primeiramente utilizando a heurística de que grupos ou pares de alunos para a atividade *debate de teses* podem ser formados com alunos de perfis heterogêneos, conforme indicado em Castro e Menezes (2011). Os perfis foram diferenciados por estilos de aprendizagem. A Figura 37 mostra com mais detalhes uma das atividades da Figura 36, especificamente a apresentada no “Tópico 6”.

Figura 37. Exemplo de atividade passada como Laboratório de Avaliação.



Fonte: o autor.

É importante ressaltar que estas recomendações de atividades são feitas pelo *Recommended Resources Agent*. A Figura 38 mostra um exemplo de um debate de teses recomendado para alguns alunos, bem como a avaliação da resposta de um aluno por seu colega.

Figura 38. Exemplo da atividade debate de teses recomendada e sua avaliação por pares.

The screenshot displays a web interface for a 'Debate de Teses 1' activity. At the top, there is a title 'Debate de Teses 1' with a help icon. Below it, a 'Descrição' section contains instructions: 'Determine se cada uma das sentenças seguintes é verdadeira ou falsa. Se a sentença for falsa, explique por quê.' and a specific question: 'a) A experiência demonstrou que a parte mais difícil de resolver um problema em computador é produzir um programa em C.' The main content area shows a student's submission titled 'minha tese 1' with a profile picture, name, and submission details (sent on May 22, 2015, at 21:25, modified at 21:40). Below the submission is a file named 'TESE de [redacted] 1.pdf'. A peer evaluation section, 'Sua avaliação', shows a score of 80 out of 80. The 'Formulário de avaliação' includes a question 'Você concorda com a resposta do colega?' with radio buttons for 'Não' and 'Sim', where 'Sim' is selected. A 'Feedback global' section shows the comment 'Questão respondida perfeitamente!'.

Fonte: o autor.

Embora aqui tenhamos mostrado figuras da aplicação do AMPARA na turma de Cálculo Numérico, o processo de adaptação de recursos baseados em APs foi realizado de forma análoga na turma de Algoritmos e Programação.

Por sua vez, os alunos participantes dos testes na turma de Algoritmos e Programação foram questionados sobre as atividades realizadas, cujo questionário completo está no Apêndice B desta tese. Identificamos alguns comentários de alunos que evidenciam o aprendizado e interesse pelo curso adaptado, apoiados na crença de que o anúncio mais efetivo de algo é o trazido pelo seu autor (Novak et. al., 2014). Alguns dos depoimentos foram: "*com a correção dos meus colegas aprendi mais*"; "*eu ajudei e fui ajudado com muitas correções*"; "*A avaliação das atividades dos colegas foi feita de forma bem coerente a mim testando meus conhecimentos e verificando o conhecimento dos meus colegas*"; "*primeira vez que presencio uma didática dessa forma, seria excelente se as demais matérias fossem dessa forma*"; "*Gostei, pois foi feito de maneira individual e ao mesmo tempo pudemos compartilhar conhecimento, e perceber que podemos chegar ao mesmo resultado usando métodos diferentes.*"; "*eu achei interessante, principalmente porque vim zerado de algoritmos e linguagem C e tive a capacidade de avaliar alguém que já conhecia, pelo nível que foi a resposta dele e acho que ele conhecia mais do que eu*"; "*nós que somos avaliados, é bom (...) estar na situação tipo profes-*

...podendo avaliar os colegas e poder interagir com as diferentes respostas que iremos encontrar"; "achei interessante para ver a dificuldade de cada aluno e assim, (...) poder ajudar (...) formando um único conhecimento geral". Uma crítica feita pelos alunos foi quanto à interface dos debates de teses apresentados no AVA.

Pode-se perceber nestes testes realizados que resultados interessantes de aprendizagem colaborativa podem ser obtidos com a abordagem utilizada do AMPARA, uma das provas de conceito do ArCARE, visto que a recomendação de atividades baseadas em arquiteturas pedagógicas é uma técnica construtivista útil para melhorar o aprendizado efetivo do aluno e que incentiva a prática de atividades colaborativas em AVAs.

5.3 Limitações do AMPARA

Nos testes realizados nessas duas turmas, é possível observar que há um processo de adaptação de recursos recomendados a alunos por meio de APs recomendadas no Moodle. No entanto, no AMPARA não foram exploradas todas as funcionalidades estabelecidas conceitualmente pelo ArCARE. Isso pode ser percebido observando-se a Tabela 4, apresentada na Seção 4.2. Percebemos que os itens IV, V e VII da Tabela 4 não são ou são parcialmente contemplados, pois:

- No item IV, nas duas turmas apresentadas, o currículo não é flexível, mas segue os padrões formais. Isso se deve ao fato de que para as disciplinas serem ministradas, elas devem seguir as normas estabelecidas pelas Instituições de Ensino;
- Quanto ao item V, ele é parcialmente coberto, pois são considerados habilidades dos alunos, estilos de aprendizagem, interesses, mas não especificamente os equipamentos que eles utilizam, se é um dispositivo móvel ou um desktop;
- Por fim, no item VII, não há uma análise de histórico de recursos recomendados, verificação se determinados recursos foram úteis ou não, nem análise de feedback dos estudantes.

5.4 Conclusões do Capítulo

Podemos dizer que o AMPARA foi um projeto-piloto de instanciação do ArCARE, onde não foram utilizadas todas as funcionalidades baseadas no ArCARE. No entanto, mesmo com tais limitações, podemos ver aspectos positivos na recomendação de recursos pelo AMPARA, desde o nível de rendimento da turma quanto às boas impressões que os alunos demonstraram

ter. No Capítulo 6 a seguir, será apresentado um segundo estudo de caso, utilizando um maior número de funcionalidades de adaptação de recursos em AVAs providas pelo ArCARE.

CAPÍTULO 6 – SISTEMA EXTENDED AMPARA (AMPARAX)

Este capítulo tem por objetivo apresentar mais uma aplicação como prova de conceito do ArcARE, o eXtended AMPARA (AMPARAX), em um curso não convencional, ou seja, que segue um currículo flexível, utilizando o AVA Moodle. Testes foram aplicados em turma real em um curso de Pensamento Computacional e seus resultados são mostrados.

Nesse capítulo será apresentado mais uma aplicação do ArcARE, o sistema AMPARAX, que é uma extensão do AMPARA, pois para o AMPARAX foi utilizado o JADEX (2017), além do JADE (2017). O AMPARAX foi aplicado em um curso de Pensamento Computacional (PC) semipresencial, tendo como base as situações propostas apresentadas no cenário 2 do Capítulo 1. Esse curso é considerado não-convencional por não estar ligado a um currículo fixo, padrão (Sarmiento et al., 2016), diferentemente do que ocorre em disciplinas consideradas convencionais, como Cálculo Numérico e Algoritmos e Programação, vistas no Capítulo 5. Um currículo flexível significa dizer que cada estudante tem a opção de escolher um conjunto de Unidades de Ensino (UEs) para desenvolver o seu perfil profissional (Sarmiento et al. 2016).

A escolha de aplicar o ArcARE para o Cenário 2 no curso de PC se justifica na dificuldade de aceitar, nos dias de hoje, que a melhor forma de conduzir o processo educacional é a “sala de aula” tradicional, tanto nas modalidades presencial, semipresencial e à distância, onde o professor conduz o processo em um único e constante ritmo. Afinal, cada vez mais, os estudantes estão acostumados a fazer múltiplas atividades ao mesmo tempo, e a desaceleração causada pela forma passiva de educação possibilitada pela sala de aula tradicional desestimula toda e qualquer intenção de aprender. Segundo Zanetti e Oliveira (2015), devemos repensar essa forma de ensino. E é nesse contexto que a abordagem do ArcARE nos Ambientes Virtuais de Aprendizagem surge como alternativa, não só motivacional como, principalmente, potencializadora do processo educacional. Utilizar as técnicas de adaptação de recursos educacionais providos pelo ArcARE se baseia na ideia de que alunos ingressantes em cursos da área de ciências exatas, tanto no nível técnico ou superior, quando em contato pela primeira vez com disciplinas introdutórias de programação de computadores sentem dificuldades, o que muitas vezes promove a desmotivação logo no início do curso.

Assim, nessa aplicação temos uma proposta metodológica que delinea um contexto de aprendizagem de programação, estimulando o desenvolvimento do Pensamento Computacional. Também apresenta um estudo realizado com alunos de Ensino Superior. Nesse contexto, é válido afirmar que o Pensamento Computacional é altamente requerido no repertório de habilidades dos alunos em disciplinas relacionadas à programação de computadores. O termo “computacional” não pode ser compreendido como sinônimo de “programação”, mas sim como uma maneira de se compreender os sistemas computacionais, como eles funcionam e como são projetados e programados (Zanetti e Oliveira, 2015).

6.1 Princípios Educacionais do Curso de Pensamento Computacional

De acordo com a *Computer Science Teachers Association (CSTA)* e a *International Society for Technology in Education (ISTE)* (2016), o termo Pensamento Computacional (PC, do inglês, *Computational Thinking*) (Wing, 2006), é um processo de resolução de problemas que inclui as seguintes habilidades ou capacidades (mas não limitada a elas):

- Formulação de problemas de uma forma que nos permita usar um computador e outras ferramentas que ajudam a resolvê-los;
- Organização e análise de dados de uma forma lógica;
- Representação de dados através de abstrações, como modelos e simulações;
- Automatização de soluções através do pensamento algorítmico (uma série de passos ordenados);
- Identificar, analisar e implementar soluções possíveis com o objetivo de alcançar a mais eficiente e eficaz combinação de passos e recursos;
- Generalizar e transferir este processo de resolução de problemas para uma grande variedade de problemas.

De acordo com o Google (2017), PC é um processo de resolução de problemas, que inclui um número de características e disposições. O PC é essencial para o desenvolvimento de aplicações computacionais, mas também pode ser usado para apoiar a resolução de problemas em todas as disciplinas, incluindo as de ciências humanas, matemática e ciências. Os alunos que aprendem sobre o PC em todo o currículo de um curso podem começar a ver uma relação entre as disciplinas acadêmicas, bem como entre a vida dentro e fora da sala de aula.

Sendo assim, a Tabela 10 mostra quais são os conceitos de PC principais e suas respectivas capacidades na Ciência da Computação. No contexto do ArCARE, cada capacidade aqui é considerada uma habilidade.

Tabela 10. Conceitos de PC e suas capacidades principais na Ciência da Computação.

Conceito em PC, capacidade (aptidão)	Definição	Exemplo de uso na Ciência da Computação
Coleta de dados	Processo de coleta da informação apropriada.	Encontrar uma fonte de dados para a área do problema.
Análise dos dados	Criar sentido entre os dados, encontrar padrões e tirar as conclusões.	Escrever um programa para fazer cálculos estatísticos básicos em um conjunto de dados.
Representação dos dados	Representar e organizar os dados de forma apropriada em gráficos, tabelas, palavras ou imagens.	Estruturas de uso de dados, como: matriz, lista encadeada, pilha, fila, gráfico, tabela hash, etc...
Decomposição do problema	Dividir as tarefas em partes menores e gerenciáveis.	Definir objetos e métodos; definir a função principal e outras funções.
Abstração	Reduzir a complexidade para definir a ideia principal.	Utilizar procedimentos para encapsular um conjunto de comandos muitas vezes repetidos que executam uma função; usar condicionais, repetições, recursividade, etc...
Algoritmos e procedimentos	Série de passos ordenados obtidos para resolver um problema ou atingir algum objetivo.	Estudar algoritmos clássicos; implementar um algoritmo para resolver um problema.
Automação	Possuir computadores ou	-

	máquinas que façam tarefas repetitivas e tediosas.	
Paralelização	Organizar recursos para, simultaneamente, realizar tarefas de modo a alcançar um objetivo comum.	<i>Threading, pipelining</i> , dividindo-se os dados ou a tarefa de tal forma que sejam processados em paralelo.
Simulação	Representação ou modelo de um processo. A Simulação envolve também experiências em andamento utilizando modelos.	Uma animação de um algoritmo.

Fonte: ISTE/CSTA (2016).

A seção 6.2 a seguir apresenta a estrutura do curso de PC baseado no ArCARE.

6.2 Estrutura do Curso

Assim como nos testes aplicados no Capítulo 5, o minicurso de PC foi hospedado no ColabWeb. A Figura 39 mostra a interface padrão do minicurso. Nele, existem os Questionários Obrigatórios, Unidades de Ensino (UEs) Obrigatórias, UEs Opcionais e as Arquiteturas Pedagógicas (APs) Recomendadas, que, neste minicurso, são os recursos que serão recomendados aos estudantes.

Os Questionários Obrigatórios são compostos por questionários para obter informações iniciais dos estudantes, como por exemplo, os questionários relativos aos estilos de aprendizagem dos alunos e são realizados da mesma forma que nos testes apresentados no Capítulo 5, compreendendo os estilos de aprendizagem de Felder-Silverman e Honey-Alonso. Além disso, entre os questionários iniciais está o teste inicial (também chamado de teste introdutório ou pré-teste). As UEs Obrigatórias são as unidades pré-definidas no curso, que o aluno deverá executar, consistindo na parte obrigatória do curso, embora o aluno possa escolher a ordem em que ele irá estudar cada UE obrigatória. Por sua vez, as UEs Opcionais são unidades extras, em que o aluno possui liberdade para escolher o que ele quiser aprender e

quais atividades no curso ele quer fazer. Por fim, as APs recomendadas são os recursos recomendados aos estudantes a fim de que eles interajam entre si, desenvolvendo suas habilidades.

Figura 39. Interface do Minicurso de Pensamento Computacional no ColabWeb.



Fonte: o autor.

Os objetivos educacionais do curso estiveram de acordo com Zanetti e Oliveira, (2015), onde há quatro objetivos bem definidos alinhados às dificuldades comumente encontradas em alunos iniciantes em programação: (i) avaliar o próprio conhecimento; (ii) compreender e interpretar problemas; (iii) resolver problemas aplicando o Pensamento Computacional e; (iv) depuração de erros e correção dos mesmos.

É importante ressaltar que as UEs Obrigatórias foram baseadas no curso de Pensamento Computacional do Google (2017), onde as capacidades (ou habilidades) dos alunos a serem desenvolvidas estão de acordo com a Tabela 10 mostrada anteriormente. O curso é composto por quatro unidades, onde cada uma delas foca nos seguintes aspectos:

- **Introdução ao Pensamento Computacional:** O que é PC? – O que é Pensamento Computacional, onde isso ocorre, por que você deveria se importar com isso e como isso está sendo aplicado.
- **Explorando Algoritmos:** Passeio através de exemplos de algoritmos usados na área do curso do estudante. Reconhecer por que os algoritmos são ferramentas poderosas

para melhorar aquilo que podemos fazer e que tecnologias podem ser úteis para a implementação de algoritmos automáticos.

- **Encontrando Padrões:** Explorar exemplos de padrões em várias atividades e desenvolver os próprios processos de solucionar um problema por meio de reconhecimento de padrões.
- **Desenvolvendo Algoritmos:** Aumentar a autoconfiança do aluno em aplicar o processo computacional para um dado problema e reconhecer como os algoritmos podem articular um processo de solução ou até mesmo uma regra para resolver determinados problemas.

O pré-teste realizado teve o objetivo de obter o diagnóstico das capacidades (ou habilidades) do aluno de acordo com a Tabela 10. A Figura 40 mostra uma questão de exemplo do pré-teste. As questões do pré- e dos pós-teste estão no Apêndice C desta tese. Estas questões foram retiradas do material oficial da AP CollegeBoard Computer Science Principles – Effective Fall (2016) e, por isso, estão em inglês.

Figura 40. Uma questão de pré-teste.

Questão 1
Ainda não respondida
Vale 1,00 ponto(s).
Marcar questão
Editar questão

2. A programmer completes the user manual for a video game she has developed and realizes she has reversed the roles of goats and sheep throughout the text. Consider the programmer's goal of changing all occurrences of "goats" to "sheep" and all occurrences of "sheep" to "goats." The programmer will use the fact that the word "foxes" does not appear anywhere in the original text. Which of the following algorithms can be used to accomplish the programmer's goal?

(A) First, change all occurrences of "goats" to "sheep."
Then, change all occurrences of "sheep" to "goats."

(B) First, change all occurrences of "goats" to "sheep."
Then, change all occurrences of "sheep" to "goats."
Last, change all occurrences of "foxes" to "sheep."

(C) First, change all occurrences of "goats" to "foxes."
Then, change all occurrences of "sheep" to "goats."
Last, change all occurrences of "foxes" to "sheep."

(D) First, change all occurrences of "goats" to "foxes."
Then, change all occurrences of "foxes" to "sheep."
Last, change all occurrences of "sheep" to "goats."

Escolha uma:

1. A

2. B

3. C

4. D

Fonte: AP CollegeBoard (2016).

Além do pré-teste, os alunos também responderam ao Termo de Consentimento Livre e Esclarecido (apresentado no Apêndice E desta tese) nas UEs Obrigatórias. Em seguida, foi passado o conteúdo programático do minicurso, que pode ser executado em qualquer ordem pelos estudantes, nas UEs Obrigatórias, que ficou assim definido:

1. Definição de Pensamento Computacional
 - 1.1 Diversas definições na literatura para Pensamento Computacional
 - 1.2 Exemplos de aplicação do Pensamento Computacional nas diversas áreas da ciência
 - 1.3 Pré-teste
2. Decomposição de Algoritmos
 - 2.1 Exemplo de decomposição: caixeiro-viajante
 - 2.2 Trabalho Prático
3. Noções de Reconhecimento de padrões
 - 3.1 Exemplo de Reconhecimento de Padrões: Compressão de Imagens
 - 3.2 Trabalho Prático
4. Abstração em Algoritmos
 - 4.1 Exemplo de Abstração em Algoritmos: Torres de Hanói
 - 4.2 Noções de análise de complexidade computacional
 - 4.3 Trabalho Prático
5. Projeto de Algoritmos
 - 5.1 Teste Final abordando as diversas estruturas de algoritmos.

Quanto às UEs optativas deste curso, elas foram colocadas no curso com o objetivo de o aluno poder escolher atividades complementares para desenvolver suas capacidades da forma que desejar. A Figura 41 mostra os links para as primeiras atividades optativas (no formato de atividade *Workshop* do Moodle, assim como no Capítulo 5), e para instruções no minicurso de PC no ColabWeb. As atividades foram tiradas do site Code.org (2017) e a Figura 42 mostra um exemplo de atividade, que constitui em organizar blocos de construção de programas em um labirinto para que o pássaro chegue ao seu destino, o porco.

Figura 41. Disposição das UEs optativas no minicurso de PC.

Unidades de Ensino Optativas

 Instruções Iniciais para Atividade do Labirinto - Plataforma Code.org

 Instruções para as atividades - É desejável ter uma conta no Programaê; entrar no item Ensinar (Ilíás) e selecionar Escolher Curso>>Praticando e Aprendendo

 Plataforma Code.Org

Contém Atividades Extras cujas respostas devem ser anexadas nos links abaixo

  Entregue aqui suas respostas das atividades da Fase 2: O Labirinto em formato .zip

AP Programação em Pares

  Entregue aqui suas respostas das atividades da Fase 5: O Artista em formato .zip

AP Programação em Pares

  Entregue aqui suas respostas das atividades da Fase 7: O Artista 2 em formato .zip

AP Programação em Pares

Fonte: o autor.

Além disso, o processo de execução dessas atividades foi baseado em Arquiteturas Pedagógicas. O estudante, ao resolver cada questão, anexa sua resposta no ColabWeb, onde irá posteriormente interagir com seu colega por naquela AP. Diante do exposto, este minicurso utiliza APs adaptadas com base no trabalho de Marques et al. (2016) para suportar o aprendizado de programação. A estrutura de uma AP adotada, definida por Reinoso e Tavares (2015), é composta por objetivo pedagógico (o que aprender), atividades pedagógicas (o que fazer), método pedagógico (como fazer) e recursos tecnológicos que viabilizam a execução das atividades planejadas. No entanto, nesta tese, trabalhamos com as APs adaptadas, onde suas descrições são apresentadas a seguir.

Figura 42. Uma atividade no Code.org (Labirinto).



Fonte: Code, (2017).

Segue a descrição da primeira AP, a Programação em Pares Adaptada:

- Objetivo pedagógico: explorar a aprendizagem de programação em pares, desenvolver o espírito crítico nos alunos ao explorarem várias possíveis soluções e explorar a aprendizagem em uma construção colaborativa de solução.
- Atividades: construir uma solução para um ou mais exercícios propostos pelo professor; revisar e testar a solução de seu par selecionado pelos agentes de recomendação de recursos; em dupla, os alunos constroem uma terceira solução diferente das duas soluções anteriores da dupla.
- Método: 1) o professor propõe um exercício, ou lista de exercícios, para os alunos; 2) cada aluno constrói uma solução para o exercício proposto e 3) a envia para o professor; 4) pares de alunos serão formados a partir das escolhas feitas pelos agentes de adaptação de recursos; 5) cada membro de um par revisa a solução do seu colega. Após a revisão, 6) cada par deve construir colaborativamente uma nova solução para o exercício, diferente das soluções construídas anteriormente pelos membros do par, e enviam a solução para o professor.
- Recursos tecnológicos: armazenador de exercícios ou listas de exercícios para acesso dos alunos pelo Moodle (ColabWeb); um editor de soluções por meio da atividade Workshop (ou Laboratório de Avaliação) do Moodle (ColabWeb); um visualizador que permita ao professor ver cada solução dos alunos; agente que forma as duplas; e um editor de soluções que permita a dois alunos construírem colaborativamente uma

solução, pelo ColabWeb; uma ferramenta para restringir a entrega de soluções similares às soluções anteriormente entregues pelo aluno. Todas essas ferramentas, exceto o agente, são fornecidas pelo ColabWeb.

Quanto às Arquiteturas Pedagógicas Recomendadas, a Figura 43 mostra links de algumas APs recomendadas no minicurso de PC no ColabWeb. As APs aparecem um pouco mais claras, em cinza, porque estão invisíveis no curso, aparecendo somente aos estudantes em que receberam recomendações de APs. É importante ressaltar que o processo de recomendação nesse estudo de caso onde o ArCARE é aplicado é detalhado na Seção 6.3.

Figura 43. Disposição das APs Recomendadas no minicurso de PC.

Arquiteturas Pedagógicas Recomendadas



Debate de Teses 2.2 a)

Determine se a sentença é verdadeira ou falsa. Justifique.

a) Todas as variáveis devem ser atribuídas a um tipo quando são declaradas.



Debate de Teses 2.9 a)

Determine se a sentença é verdadeira ou falsa. Justifique.

a) Os seguintes identificadores são todos nomes de variáveis válidos:

`_under_bar_`, `m928134`, `t5`, `j7`, `her_sales$`, `his_$account_total`, `a`, `b$`, `c`, `z` e `z2`.



Projeto de Aprendizagem 5.8

Compare e contraste as instruções *break* e *continue*.

Fonte: o autor.

Além das já conhecidas APs debate de teses e projeto de aprendizagem, que foram utilizadas, outras duas APs foram criadas e utilizadas, voltadas para o Pensamento Computacional baseadas no trabalho de Marques et al. (2016). Segue a descrição da primeira AP, Discussão em Pensamento Computacional Adaptada:

- Objetivo pedagógico: explorar a aprendizagem de conceitos ligados ao Pensamento Computacional em pares ou em grupos, desenvolver o espírito crítico nos alunos ao explorarem possíveis soluções de problemas e explorar a aprendizagem em uma construção colaborativa de solução.
- Atividades: discutir uma solução para um ou mais exercícios propostos pelo professor por meio de opiniões em cada atividade passada; revisar e testar a solução de seu par ou grupo selecionado pelos agentes de recomendação de recursos; em dupla ou grupo,

os alunos constroem uma terceira solução diferente das duas soluções anteriores da dupla.

- Método: 1) o professor propõe um exercício, ou lista de exercícios, para os alunos; 2) cada aluno constrói uma solução para o exercício proposto, com base em discussões e 3) a envia para o professor; 4) pares ou grupos de alunos serão formados a partir das escolhas feitas pelos agentes de adaptação de recursos; 5) cada membro de um par ou grupo revisa a solução do seu colega. Após a revisão, 6) cada par ou grupo deve construir colaborativamente uma nova solução para o exercício, diferente das soluções construídas anteriormente pelos membros do grupo, e enviam a solução para o professor.
- Recursos tecnológicos: armazenador de exercícios ou listas de exercícios para acesso dos alunos pelo Moodle (ColabWeb); um editor de soluções por meio da atividade Workshop (ou Laboratório de Avaliação) do Moodle (ColabWeb); um visualizador que permita ao professor ver cada solução dos alunos; agentes que formam as duplas ou grupos; e um editor de soluções que permita a dois ou mais alunos construírem colaborativamente uma solução, pelo ColabWeb; uma ferramenta para restringir a entrega de soluções similares às soluções anteriormente entregues pelo aluno.

Por fim, segue a descrição da AP Projeto em Pensamento Computacional Adaptado:

- Objetivo pedagógico: explorar a aprendizagem de conceitos ligados ao Pensamento Computacional em pares ou em grupos, estimular a criatividade dos alunos, desenvolver o espírito crítico nos alunos ao explorarem possíveis soluções de problemas e explorar a aprendizagem em uma construção colaborativa de solução.
- Atividades: elaborar um projeto que envolva a solução de um problema envolvendo programação de computadores e Pensamento Computacional; revisar e testar a solução de seu par ou grupo selecionado pelos agentes de recomendação de recursos; no final, cada aluno modifica o seu projeto inicial, deixando-o ajustado com base nas revisões de seus colegas.
- Método: 1) o professor propõe um problema para os alunos ou propõe que os alunos criem um problema a ser tratado; 2) cada aluno elabora uma ideia de projeto a fim de resolver o problema e 3) a envia para o professor; 4) pares ou grupos de alunos serão formados a partir das escolhas feitas pelos agentes de adaptação de recursos; 5) cada membro de um par ou grupo revisa a ideia de projeto do seu colega. Após a revisão, 6)

cada aluno deve reformular seu projeto, com base nas observações de seus colegas, e envia a solução para o professor.

- Recursos tecnológicos: armazenador de exercícios ou listas de exercícios para acesso dos alunos pelo Moodle (ColabWeb); um editor de soluções por meio da atividade Workshop (ou Laboratório de Avaliação) do Moodle (ColabWeb); um visualizador que permita ao professor ver cada solução dos alunos; agentes que formam as duplas ou grupos; e um editor de soluções que permita a dois ou mais alunos construírem colaborativamente uma solução, pelo ColabWeb; uma ferramenta para restringir a entrega de soluções similares às soluções anteriormente entregues pelo aluno.

É importante ressaltar que nesse processo, evita-se a resolução dos exercícios por parte do professor e busca-se uma postura menos diretiva por parte do docente, embora a qualquer momento que este julgar interessante, acesse o AVA para ver o desempenho dos alunos. Os problemas propostos pelo professor são resolvidos utilizando as contribuições dos alunos. Neste caso, os agentes funcionam como elementos recomendadores de recursos e mediadores para selecionarem os pares de alunos que vão interagir em uma AP. Ou seja, os agentes funcionam como facilitadores e não como um guia que apresenta uma solução passo-a-passo. Os agentes devem incentivar os alunos a colocarem suas soluções (corretas ou não) para serem compartilhadas com os colegas nas APs criadas para que ocorra a discussão entre pares e enfim, o aprendizado mútuo. As APs utilizadas nesse minicurso foram: debate de teses, programação em pares adaptada, projeto de aprendizagem, projeto em pensamento computacional adaptado e discussão em pensamento computacional adaptada

A seção 6.3 a seguir relata a execução do minicurso de PC sob a aplicação do ArCARE. Importante destacar que para o processo de adaptação de recursos nesse minicurso, os dados utilizados tanto dos alunos quanto os dados associados aos recursos foram os níveis de habilidades e de estilos de aprendizagem, além do grau de utilidade dos recursos já recomendados.

6.3 Execução do Curso e Resultados Obtidos

Nesse estudo de caso, utilizamos o AMPARAX baseado no framework ArCARE no no minicurso de Pensamento Computacional, composto inicialmente por 38 alunos. Durante o curso, alguns alunos desistiram, não comparecendo mais às aulas e não realizando as atividades. Esses alunos foram retirados do experimento. Dessa forma, foram considerados válidos para o

experimento apenas os alunos que realizaram o pré-teste, o teste final e as demais atividades previstas pelo cronograma do minicurso. Portanto, esse curso teve como público 33 estudantes de Ensino Superior do Instituto Federal do Amazonas Campus Manaus Distrito Industrial (IFAM-CMDI), do curso de Tecnologia em Mecatrônica Industrial, onde o curso teve a carga horária total de 30 horas. Convencionamos que os horários de execução dos agentes seriam durante todo o dia e no decorrer de todo o período do minicurso de PC.

Inicialmente, foi passada uma atividade em um formato de pré-teste, conforme já apresentado na seção 6.2 e na Figura 40, a fim de obter os níveis de habilidades iniciais dos estudantes. Na prática adotada para este curso, os estudantes devem resolver problemas propostos refletindo quais são as ações corretas a desempenhar e que estruturas de programação são necessárias. E quando algum erro aparece, os estudantes podem testar e avaliar uma nova possível solução através de discussão com seus colegas, buscando ter um resultado melhor em suas atividades.

Tanto no pré- quanto no pós-teste, as habilidades envolvidas foram as mesmas, mas obviamente, com perguntas diferentes. As habilidades relacionadas aos testes foram as apresentadas na Tabela 10 classificadas pelo Google Computational Thinking Course (2017) em parceria com a International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) (2016), que são: domínio em projeto de algoritmos e programação, abstração, decomposição de problemas, simulação, reconhecimento de padrões e análises de dados. Além disso, duas outras habilidades foram avaliadas a fim de que tivéssemos uma informação mais acurada a respeito do desempenho do estudante em atividades: domínio em estruturas de condição e repetição.

Os agentes de modelo aluno foram implementados no framework JADE (2017). Para este minicurso, seguindo a descrição formal do ArCARE no Capítulo 4, os agentes implementados foram:

- *Initial Profile Agent*: instanciação do *Agente de Configuração de Perfil Inicial*;
- *Learning Assessment Agent*: instanciação do *Agente de Avaliação de Desempenho do Estudante*;
- *Update Profile Agent*: instanciação do *Agente de Atualização*, que obtém informações do *Initial Profile Agent* e *Learning Assessment Agent*.

Por sua vez, os agentes de adaptação de recursos foram implementados no framework JADEX (2017), que é uma extensão do JADE e que utiliza o modelo *Beliefs-Desires*-

Intentions (BDI) (Georgeff et al., 1999), a fim de que as adaptações de recursos fossem baseadas em estados mentais dos agentes. Tais agentes implementados para o minicurso de PC foram:

- *Profile Situation Assessment Agent*: instanciação do *Agente de Avaliação de Situação de Perfis*;
- *Recommended Resources Agent*: instanciação do *Agente Adaptador de Recursos* e do *Agente de Notificação*, pois ele apresenta os recursos no AVA aos estudantes.

Logo após a obtenção do perfil inicial de cada aluno por meio dos 3 questionários, foram iniciadas as recomendações de recursos adaptados, neste caso, Arquiteturas Pedagógicas de acordo com os perfis dos estudantes. A Figura 44 mostra um exemplo de uma atividade do tipo *Workshop* no ColabWeb baseada na AP debate de teses (nesse caso, o Debate de Teses 4), com todas as suas fases: configuração, envio (etapa onde os alunos enviam suas respostas), fase de avaliação (onde os colegas interagem entre si, lançando notas nas atividades de seus colegas), avaliação de classificação (onde o professor faz suas observações e lança sua nota final com base nas notas dadas pelos colegas) e a fase de encerramento, indicando que a atividade está terminada. A fase selecionada na Figura 44 é a fase de configuração que consiste em definir a duração da atividade, escalas de notas, métodos de avaliação, entre outras características da atividade. Abaixo das fases disponíveis na AP, fica a sua descrição, que consiste no enunciado da atividade.

Figura 44. Exemplo de uma atividade baseada em APs com suas fases.

Debate de Teses 4

Configurar fase

Configurar fase	Fase de envio	Fase de avaliação	Fase de avaliação de classificação	Encerrado
<ul style="list-style-type: none"> ✓ Defina a descrição do workshop ✓ Forneça instruções para envio ✓ Edite formulário de avaliação 	<ul style="list-style-type: none"> ✓ Forneça instruções para avaliação ✓ Aloque envios <ul style="list-style-type: none"> esperado: 38 enviado: 2 alocado para: 0 ⓘ Prazo dos envios: segunda, 25 Mai 2015, 01:45 (843 dias atrás) ⓘ As restrições de tempo não se aplicam a você 	<ul style="list-style-type: none"> ⓘ Aberto para avaliação de segunda, 25 Mai 2015, 01:45 (843 dias atrás) ⓘ As restrições de tempo não se aplicam a você 	<ul style="list-style-type: none"> ✓ Calcule notas de envios <ul style="list-style-type: none"> esperado: 38 realizado: 0 ✓ Calcule notas de avaliação <ul style="list-style-type: none"> esperado: 38 calculado: 0 ✓ Forneça uma conclusão da atividade 	

Descrição

Determine se cada uma das sentenças seguintes é verdadeira ou falsa. Se a sentença for falsa, explique o porquê.

a) A experiência tem mostrado que a parte mais difícil de resolver um problema de computador é produzir um programa em C.

Fonte: o autor.

Na fase de envio, os alunos devem enviar, no ambiente, as suas respostas. Depois, ocorre a fase de avaliação, onde os estudantes irão interagir entre si no AVA, corrigindo e comentando as respostas dos seus pares. A Figura 45 apresenta um exemplo de avaliação por um aluno no mesmo debate de teses da Figura 44, após o colega ter enviado sua resposta no ambiente. Após essas avaliações, é gerada a nota que o aluno obteve na atividade. Isso será útil para o posterior cálculo do ganho ou perda de níveis de habilidades dos alunos, pelo *Learning Assessment Agent*, de acordo com a Equação 1 da subseção 4.4.1. Sendo assim, tomando a Equação 2 como instância da Equação 1, para a atividade de exemplo mostrada na Figura 48, temos:

$$h_x = \sum_{i=1}^m (f_i \cdot g_i \cdot c_i \cdot h_{i a_i} \cdot d_i \cdot t_i \cdot \frac{1}{k_i}) \text{ (Equação 2)}$$

Em que, neste debate de teses, f_i é um fator de normalização que permite que $0 \leq h_x \leq 10$, onde devido as diversas variáveis presentes nesta fórmula e pelos valores que elas podem assumir, $f_i = 1/5000$; g_i é a nota do aluno na atividade, que pode ser 0, 30, 70 ou 100, já configurada dessa forma no ColabWeb, dependendo dos erros e acertos do aluno; c_i poderia assumir os valores +1, +0,5, -0,5 ou -1; $h_{i a_i}$ pode assumir os seguintes valores de níveis de habilidades, para cada habilidade x : Simulação (2), Domínio em programação (8), Domínio em estruturas condicionais (6), Domínio em estruturas de repetição (6); Abstração (7); Domínio em projeto de algoritmos (8); Reconhecimento de padrões e análise de dados (5) e Decomposição de problemas (8) d_i : dificuldade da questão, nesse caso, valendo 6, numa escala de 0 a 10; t_i , neste caso, assume valor igual a 1, pois os alunos entregaram dentro do prazo; k_i poderia assumir os valores 1 ou 2, pois o aluno poderia fazer no máximo 2 tentativas nesta atividade. Além disso, é importante ressaltar que a seleção deste recurso para os alunos estão de acordo com os critérios de seleção de **recursos melhores** a serem recomendados, de acordo com a subseção 4.4.3.

Figura 45. Exemplo de avaliação a um debate de teses recomendado para estudantes.

Debate de Teses 4

Envio avaliado

Debate de Teses 4
por [nome oculto]
enviado em sexta, 9 Jun 2017, 20:42

Não, a experiência mais difícil é a formulação matemática do problema, ou seja, a parte lógica do problema.

Instruções para avaliação ▾
Você concorda com o texto do colega?

Sua avaliação
por [nome oculto]
Nada avaliado ainda

Formulário de avaliação ▾

Critérios	Níveis
Você concorda com a resposta do colega?	<input type="radio"/> Não <input checked="" type="radio"/> Sim

Feedback global

Retorno para o autor

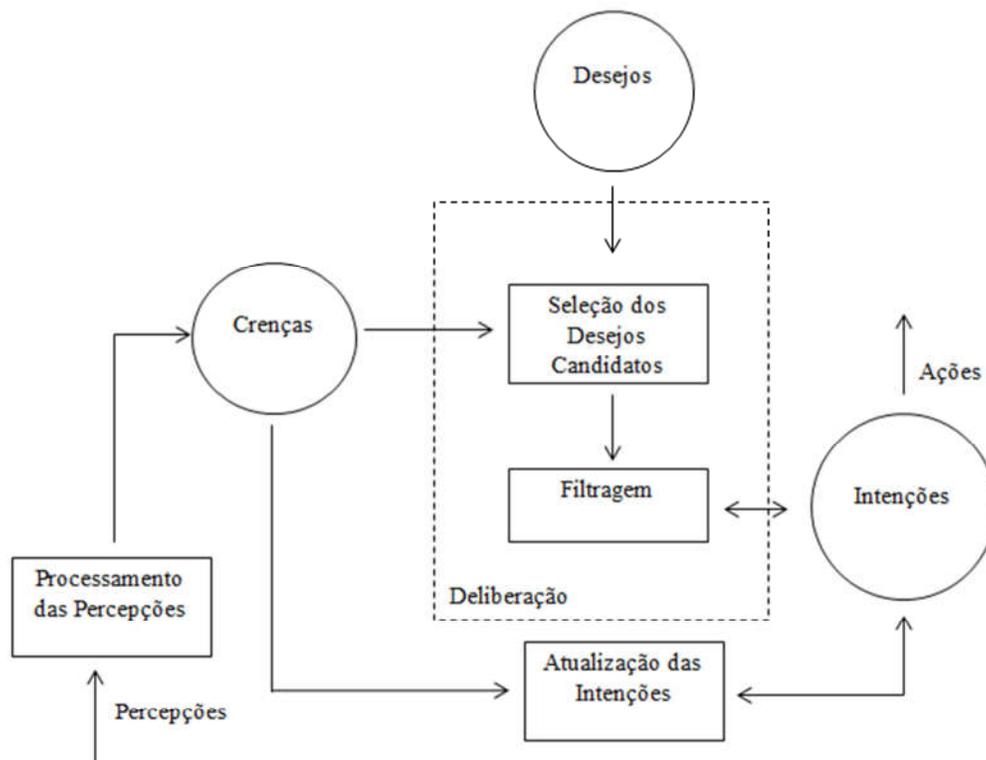
Fonte: o autor.

6.3.1 Modelo BDI Aplicado aos Agentes de Adaptação de Recursos

A adaptação do recursos no ColabWeb para o minicurso de PC foi feita pela *Recommended Resources Agent* e pelo *Profile Situation Assessment Agent*, ambos desenvolvidos no framework JADEx, utilizando os estados mentais dos agentes na arquitetura BDI. Essa adaptação no minicurso ocorreu de duas maneiras 1) seleção de colegas (pares) para corrigir as respostas de cada estudante e 2) recomendação de APs de acordo com o perfil do estudante. Sendo assim, neste cenário utilizamos APs com correção em pares.

A Figura 46 ilustra a estrutura interna do núcleo de cada um dos agentes de adaptação de recursos utilizados no AMPARAX. Essa estrutura pode ser classificada como um modelo BDI, pois representa de maneira explícita os estados mentais de crença, desejo e intenção. Tal estrutura se baseia no trabalho de Fagundes (2004) e no trabalho de Todorov et al. (2016).

Figura 46. Estrutura interna do núcleo de um Agente de Adaptação de Recursos e seus estados mentais utilizado no curso de PC.



Fonte: o autor.

As características básicas desses agentes são:

- **Autonomia:** eles operam sem a intervenção direta dos usuários ou outros agentes. Eles têm o controle sobre suas próprias ações e estados internos;
- **Reatividade:** eles percebem o ambiente a todo momento e respondem às mudanças que ocorrem nele;
- **Proatividade:** eles não apenas agem em resposta para o ambiente deles. Eles são aptos a serem proativos exibindo um comportamento orientado a objetivos;
- **Sociabilidade:** eles estão aptos a interagirem e cooperarem com outros assistentes via alguma linguagem de comunicação (e. g. Agent Communication Language - ACL).

Estados Mentais

A seguir, apresentamos detalhadamente a modelagem de cada um dos estados mentais. Os estados mentais do núcleo dos agentes de recomendação de recursos, nessa aplicação em um curso de PC, consistem em:

- **Crenças (Beliefs):** O que o agente já sabe a respeito dos alunos e dos recursos, ou seja, todas as informações necessárias provenientes do banco de dados do AVA e das regras de comunicação e adaptação de recursos fornecidas pela ontologia de MAA.
- **Desejos (Desires/Goals):** Os desejos, ou objetivos que os agentes devem alcançar. Possuem um objetivo imediato, que é de adaptar diversos recursos para cada aluno, de acordo com seu perfil, no AVA;
- **Intenções (Intentions/Plans):** Cada tarefa que o agente vai executar em um determinado momento. Por exemplo, um agente pode ter uma intenção de consultar quais alunos obtiveram notas acima de 7 em um questionário valendo de 0 a 10. Depois, outra intenção de selecionar que recursos serão indicados para este aluno. E depois, fornecer esses recursos ao aluno.

Crenças (Beliefs)

As crenças representam os conhecimentos que o agente possui acerca do ambiente que ocupa e faz parte. Uma crença descreve as características de algo, que chamaremos de objeto. Esse objeto pode ser, por exemplo, um aluno, um recurso ou uma atividade no AVA. O grupo de crenças determina o plano (intenção) que precisa ser executado.

Uma crença contém as seguintes informações:

- **Agente:** nome do agente que possui a crença;
- **Código Identificador:** código para identificar e reconhecer o objeto ao qual a crença se refere, pois em alguns casos é possível que objetos distintos possuam as mesmas características;
- **Características:** descrição do objeto ao qual a crença se refere;
- **Localização Temporal:** corresponde ao instante de aquisição da crença, indicando de maneira implícita se uma crença está desatualizada.

Desejos (Desires/Goals)

Os desejos representam estados do mundo que o agente quer alcançar. O modelo de desejo proposto pela presente arquitetura não faz referência a um objeto específico, mas às características que o descrevem. Esta abordagem apresenta como vantagem a possibilidade de caracterizar um objeto tanto quanto necessário, podendo assim abranger um conjunto de objetos

referenciando características comuns, ou apenas um objeto referenciando características particulares.

Sendo assim, um desejo é formado pelos seguintes atributos:

- **Agente:** nome do agente que possui o desejo;
- **Prioridade:** usada no processo que seleciona os desejos candidatos a gerar intenções;
- **Desejos Incompatíveis:** correspondem aos desejos que não poderão gerar intenções quando o desejo em questão já tiver gerado uma intenção, ou seja, evita que desejos gerem intenções que não podem ser executadas paralelamente a alguma intenção já existente;
- **Condições:** conjunto de condições que denotam os estados do mundo que fazem com que um desejo seja considerado candidato a gerar intenções, e estados do mundo que indicam que uma intenção gerada pelo desejo foi realizada;
- **Ações:** conjunto de ações a serem executadas por uma intenção gerada pelo desejo, de modo a satisfazê-la.

Um desejo é selecionado segundo a sua prioridade, e tem associado a si um conjunto de condições, as quais expressam as transformações no ambiente que o agente quer provocar. Cada condição possui os seguintes atributos:

- **Desejo:** uma condição pertence a um desejo;
- **Pré-condição:** denota as características que um objeto deve possuir para que a pré-condição seja satisfeita;
- **Pós-condição:** denota as características que um objeto deve assumir para que uma intenção gerada pelo desejo seja considerada realizada.

Um desejo é selecionado como candidato a gerar uma intenção quando todas as pré-condições são satisfeitas. As pós-condições são usadas apenas pelas intenções geradas pelo desejo para verificar sua realização.

Para exemplificar, o desejo de recomendar um objeto do tipo recurso possui uma condição, cuja pré-condição é a existência de um recurso disponível e um objeto aluno que possua o perfil adequado para usar tal recurso, e a pós-condição é quando o recurso já está recomendado. Neste caso, não recomendado ou recomendado são características de um objeto recurso. Da maneira como foi modelado, este desejo se refere a um recurso qualquer, e se qui-

sermos nos referir a um determinado recurso, devemos adicionar uma característica particular que o torne único, como a sua identificação (id).

É possível criarmos uma condição sem pré-condição, sem pós-condição, ou mesmo sem ambas. Quando uma pré-condição ou pós-condição não for especificada, partimos do pressuposto que ela é sempre satisfeita. Um desejo cujas condições não possuem pré-condições é sempre selecionado como desejo candidato.

Intenções (Intentions/Plans)

As intenções, assim como os desejos, correspondem a estados do mundo que o agente quer provocar, mas ao contrário dos desejos, as intenções exigem um comprometimento por parte do agente. O modelo de intenção proposto é bastante similar ao modelo de desejo, sendo que a fundamental diferença reside nas condições da intenção.

Uma intenção é composta pelos seguintes atributos:

- **Agente:** nome do agente que possui a intenção;
- **Desejo:** referência ao desejo que a gerou;
- **Localização Temporal:** corresponde ao instante em que a intenção foi gerada;
- **Estado:** uma intenção pode assumir os seguintes estados:
 - **Ativa:** estado que a intenção assume após criada;
 - **Executada:** a intenção assume este estado após ter as suas ações executadas;
 - **Realizada:** uma intenção é dita realizada quando todas as suas pós-condições são satisfeitas;
 - **Impossível:** uma intenção impossível caso uma das pré-condições deixe de existir antes de o agente realizá-la;
- **Condições:** conjunto de condições que denotam os estados do mundo antes, e depois da intenção ser realizada;
- **Ações:** conjunto de ações a serem executadas de modo a satisfazer a intenção.

Uma intenção tem associada a si um conjunto de condições, as quais expressam as transformações que o agente quer provocar no ambiente através da execução do conjunto de ações. Mas ao contrário das condições adotadas pelos desejos, as condições das intenções estão associadas a objetos. Cada condição é caracterizada pelos seguintes atributos:

- **Intenção:** uma condição pertence a uma intenção;
- **Crença:** corresponde a crença associada à condição;

- **Pré-condição:** no momento em que a intenção é gerada, a pré-condição corresponde às características da crença associada, ou seja, o agente acredita que o objeto referenciado pela crença associada preenche os requisitos para satisfação desta pré-condição;
- **Pós-condição:** no momento em que a intenção é realizada, a pós-condição corresponde às características da crença associada, ou seja, o agente acredita que o objeto referenciado pela crença associada assumiu o estado que satisfaz a pós-condição.

Quando uma intenção é gerada, ela é considerada ativa e possui todas as pré-condições satisfeitas. Uma intenção é dita realizada quando possui todas as pós-condições satisfeitas. Digamos que o desejo de recomendar um recurso gerou uma intenção, pois o agente passou a crer que existe um recurso que não está recomendado, o que corresponde à única pré-condição do desejo. Uma vez criada a intenção, o agente executará um conjunto de ações que recomendará o recurso ao(s) aluno(s) adequados. Em seguida, o agente percebe que o recurso está recomendado. Isto faz com que a intenção seja considerada realizada, pois a sua única pós-condição foi satisfeita. Uma vez que as condições dos desejos podem ocorrer sem pré-condição, sem pós-condição, ou sem ambas, as condições das intenções também podem ocorrer sem elas. Uma intenção sem pós-condições é sempre considerada realizada.

Processos

Os processos apresentados a seguir são responsáveis pela manipulação dos estados mentais.

Processamento da Percepção

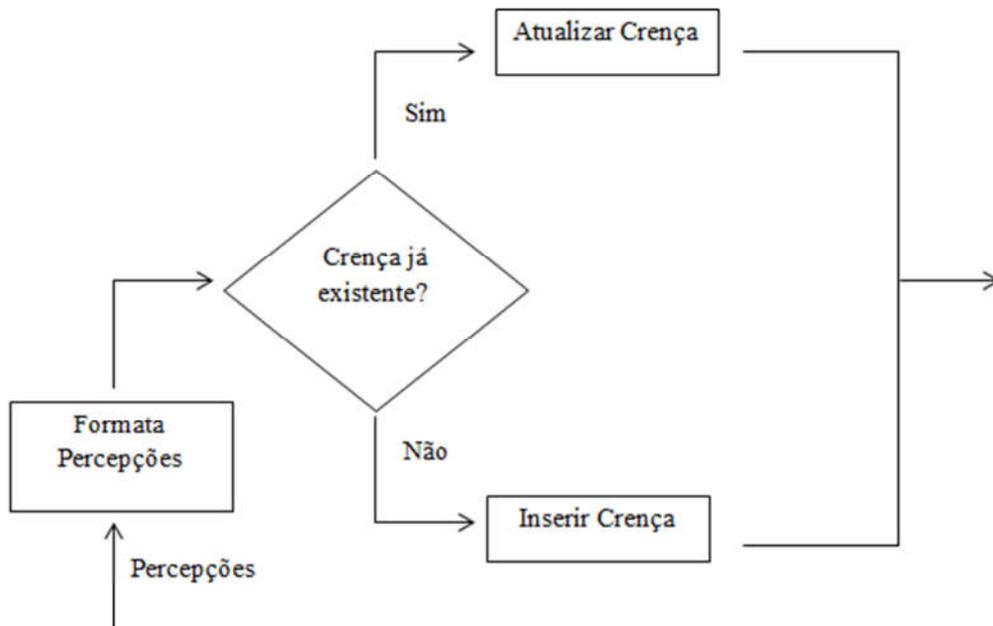
O processamento da percepção tem como objetivo registrar tudo aquilo que o agente percebe, mantendo a consistência de suas crenças. Este processo recebe como entrada uma percepção que o agente obteve do seu ambiente, a qual é composta por um código que identifica um objeto, e um conjunto de características do mesmo.

Durante o processamento da percepção, em cada interação dos estudantes com o AVA, se alguma mudança no AVA é detectada, uma verificação é executada primeiro a fim de saber e decidir quais são as condições corretas para alcançar um certo objetivo e se possível, um plano (intenção) é construído.

Antes de ser inserida, uma percepção deve ser adequada ao formato adotado pelas crenças. Caso exista uma crença que se refere ao mesmo objeto referido pela percepção, a crença é considerada obsoleta e a percepção é inserida como uma crença atualizada. Manter as crenças obsoletas permite que o agente mantenha um histórico ao longo de sua existência.

O processamento da percepção é detalhado na Figura 47.

Figura 47. Processamento das percepções.



Fonte: o autor.

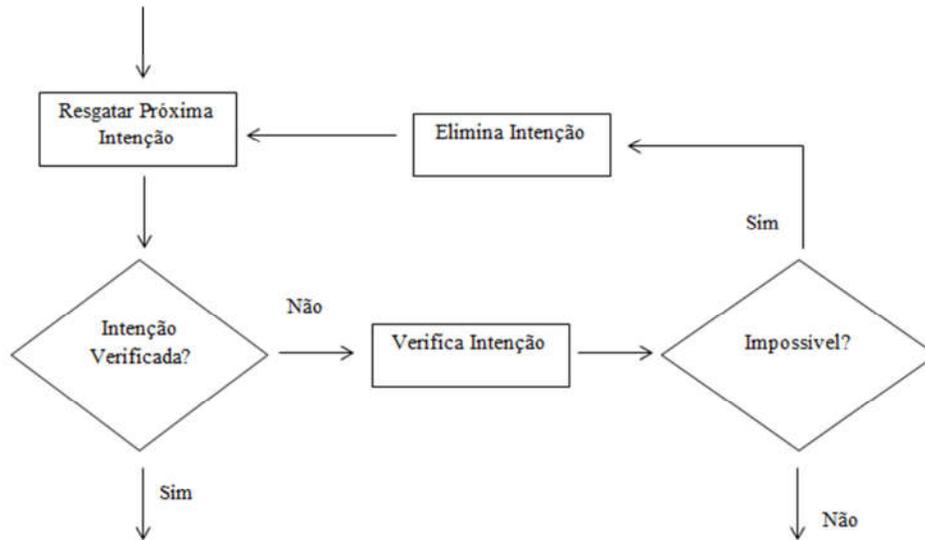
Atualização das Intenções

O processo de atualização das intenções, detalhado na Figura 48, elimina intenções impossíveis e realizadas. Uma intenção é considerada realizada quando o agente acredita que todas suas pós-condições estão satisfeitas. Uma intenção é considerada impossível quando um agente deixa de acreditar em alguma de suas pré-condições, ou seja, o motivo que a originou deixou de existir.

Como exemplo, imaginemos que um agente recomendador de recurso possui o desejo de recomendar um recurso para um estudante. As pré-condições são: um agente verificando um perfil de um estudante e um recurso disponível adequado para aquele perfil, por exemplo, uma AP contendo um vídeo. As pós-condições são: o respectivo aluno selecionado e o respectivo recurso recomendado, associado ao aluno no AVA. Digamos que em um determinado momento o agente passa a crer nas pré-condições, e o desejo de recomendar um recurso para estudante gera uma intenção. Caso uma das pré-condições deixe de existir antes do agente realizar a intenção, ela será considerada impossível. Seria impossível recomendar um recurso se não houver aluno com perfil adequado para acessá-lo. Caso o agente passe a crer nas pós-

condições, a intenção é considerada realizada, ou seja, no caso de ocorrer uma situação em que um recuso foi recomendado a um aluno.

Figura 48. Atualização das intenções.



Fonte: o autor.

Em determinados casos, certas condições devem ser mantidas para que as intenções sejam realizadas, ou seja, a pré-condição é igual a pós-condição. Por exemplo, uma pré e pós-condição poderia já ser um recurso recomendado para um estudante, para os casos que temos um histórico de recomendações de recursos, onde estes recursos recomendados foram muito úteis. Portanto, as pré e pós-condições poderiam ser as mesmas a respeito de determinados recursos recomendados a alunos, a fim de que os recursos com altos graus de utilidade sejam recomendados a alunos de forma otimizada.

Deliberação

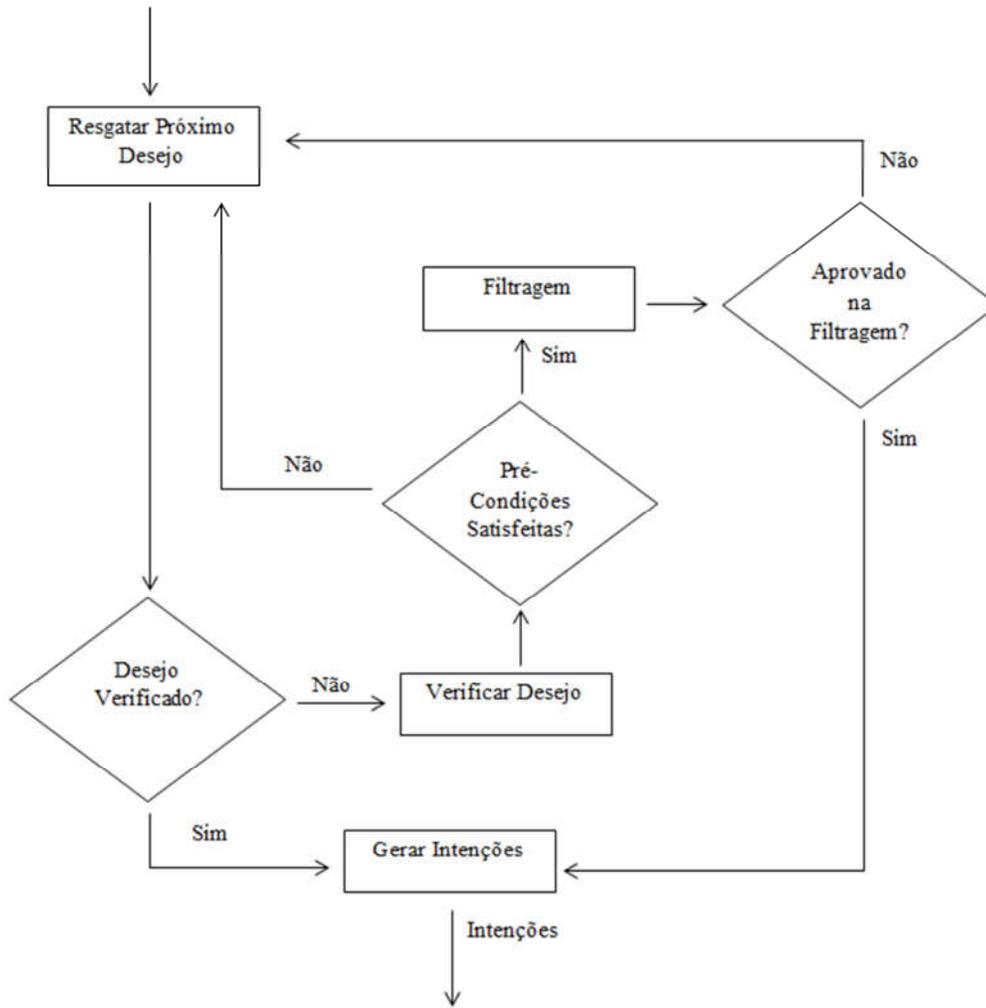
O processo deliberativo consiste em raciocinar e decidir um estado do mundo que deve ser atingido. É formado por duas etapas, seleção de desejos candidatos e filtragem, as quais são detalhadas na Figura 49.

Seleção de Desejos Candidatos

Durante esta etapa, é selecionado um conjunto de desejos candidatos a gerar intenções. Os desejos são selecionados quando o agente passa a crer que todas as suas pré-condições estão

satisfeitas, ou seja, quando existe pelo menos uma crença relacionada a cada condição. Esse processo seleciona primeiro os desejos de maior prioridade.

Figura 49. Processo deliberativo.



Fonte: o autor.

Filtragem

Os desejos candidatos selecionados na etapa anterior são encaminhados ao processo de filtragem, que elimina os desejos que são incompatíveis com as intenções do agente. O processo de filtragem elimina os desejos candidatos cujas ações não podem ser executadas de maneira paralela às ações pertencentes a alguma intenção existente. Tais desejos estão listados no componente de desejos incompatíveis do desejo que gerou a intenção. Como exemplo de incompatibilidade, citamos o fato de um agente de recomendação querer recomendar um recur-

so não adequado a um determinado perfil. É possível que tenhamos intenções cujas ações conduzam um determinado objeto a estados distintos, sendo que neste caso sabemos com certeza que apenas uma das intenções será realizada. As restantes serão consideradas impossíveis e removidas pelo processo de atualização de intenções, uma vez que o objeto assumiu um estado distinto da pré-condição e da pós-condição. Os desejos candidatos que sobrevivem a este processo geram uma intenção, a qual contém um conjunto de ações que ao serem executadas, devem satisfazê-la.

6.3.2 A Ontologia de MAA Estendida com BDI

Os agentes desenvolvidos no JADEx utilizaram a ontologia criada no Protégé (2017). A ontologia de MAA existente e apresentada no Capítulo 4 foi estendida para representar os estados mentais dos agentes de adaptação de recursos, sendo uma ontologia de aplicação, onde no Apêndice A é apresentada a ontologia em sua versão completa. A extensão da ontologia criada consiste nas seguintes classes, filhas de *Concept*:

- *MentalState*: é uma classe abstrata, que tem como classes herdeiras as classes *Belief*, *Desire* e *Intention*;
- *Belief*: representa uma crença;
- *Desire*: representa um desejo;
- *Intention*: representa uma intenção;
- *Conditions*: é uma classe abstrata, que representa condições associadas a estados mentais pró-ativos, e tem como classes herdeiras as classes *DesireConditions* e *IntentionConditions*;
- *DesireConditions*: representa uma condição associada a um desejo;
- *IntentionConditions*: representa uma condição associada a uma intenção.

A Tabela 11 apresenta as classes e propriedades da ontologia de MAA estendida. Por sua vez, a Figura 50 mostra uma versão resumida da ontologia estendida com as classes e propriedades criadas, junto com as relações que a classe *MentalState* possui com os agentes *Recommended Resources Agent* e *Profile_Situation_Assessment_Agent*.

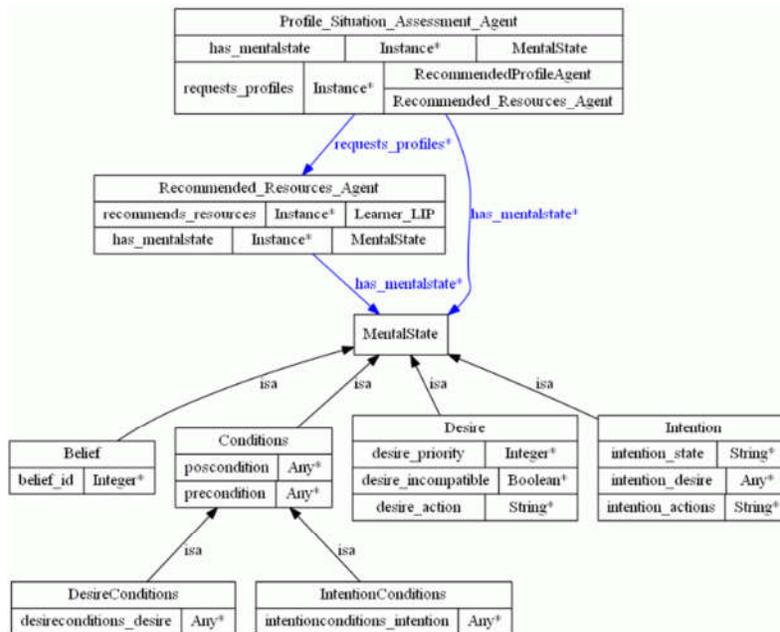
Tabela 11. Expansão da Ontologia de MAA para representação dos estados mentais dos Agentes de Adaptação de Recursos no JADEx.

Classe	Propriedades	Descrição
<i>MentalState</i>	<i>has_mentalstate</i> , relacionada	Classe pai das classes <i>Belief</i> , <i>De-</i>

	com <i>Recommended Resources Agent</i> e <i>Profile Situation Assessment Agent</i>	sire, Intentions. Descreve os estados mentais dos Agentes de Adaptação de Recursos.
<i>Belief</i>	<i>belief_id</i>	Crenças
<i>Desire</i>	<i>desire_priority, desire_incompatible, desire_action</i>	Desejos
<i>Intentions</i>	<i>intention_state, intention_desire, intention_actions</i>	Intenções
<i>Conditions</i>	<i>preconditions, postconditions</i>	Classe pai das classe DesireConditions e IntentionConditions
<i>DesireConditions</i>	<i>desireconditions_desire</i>	Condições associadas a um desejo
<i>IntentionConditions</i>	<i>intentionconditions_intention</i>	Condições associadas a uma intenção

Fonte: o autor.

Figura 50. Representação gráfica resumida da ontologia estendida de MAA, considerando os estados mentais.



Fonte: o autor.

6.3.3 Desejos e Execução dos Agentes de Adaptação de Recursos

Para desenvolver os agentes não foi necessário criar intenções iniciais, nem crenças iniciais. As intenções são obtidas somente quando os agentes passam a crer nas pré-condições de um desejo, e as crenças são obtidas através das percepções no AVA pelo acesso ao banco de dados. Neste caso, foi necessário apenas criar os desejos dos agentes, que são descritos a seguir.

DES1 – Obter atividades respondidas pelos alunos:

- *Agente: Profile Situation Assessment Agent*
- *Pré-condições: Estudantes fizeram as tarefas*
- *Pós-condições: Identificação das atividades respondidas*
- *Desejos incompatíveis: 1, 2, 3, 4, 5*
- *Prioridade: 1*
- *Ações:*
 - *Acessar o BD do AVA*
 - *Verificar quais atividades foram respondidas*
 - *Verificar quem fez as atividades respondidas*

DES2 – Obter os perfis dos alunos que responderam as atividades:

- *Agente: Profile Situation Assessment Agent*
- *Pré-condições: Identificação das atividades respondidas*
- *Pós-condições: Dados dos alunos que fizeram as atividades*
- *Desejos incompatíveis: 2, 3, 4, 5*
- *Prioridade: 2*
- *Ações:*
 - *Acessar o BD do AVA*
 - *Obter os níveis de habilidades e estilos de aprendizagem de todos os estudantes que fizeram as atividades. Executar uma instância da função verifica_e_combina(...)*
 - *Enviar mensagem ao Recommended Resources Agent contendo os dados de estudantes e suas atividades realizadas – Mensagem 1*

DES3 – Selecionar colegas para corrigirem as atividades dos alunos nas APs:

- *Agente: Recommended Resources Agent*
- *Pré-condições: - Obtenção da Mensagem 1 do Profile Situation Assessment Agent*
- *Obtenção dos níveis de habilidades e estilos de aprendizagem de todos os estudantes que fizeram as atividades*

- *Pós-condições: Seleção de pares de estudantes para interagirem em cada AP respondida*
- *Desejosincompatíveis:3*
- *Prioridade: 3*
- *Ações:*
 - *Selecionar pares adequados de estudantes para interagirem de acordo com os perfis dos alunos utilizando o algoritmo K-Means*
 - *Registrar no Banco de Dados do AVA os pares de alunos para cada atividade (AP)*

DES4 – Identificar recursos a serem recomendados para estudantes:

- *Agente: Profile Situation Assessment Agent*
- *Pré-condições: Obtenção dos níveis de habilidades e estilos de aprendizagem de todos os estudantes, tendo ou não realizado as atividades*
- *Pós-condições: Seleção de recursos a serem recomendados para estudantes*
- *Desejosincompatíveis:4, 5*
- *Prioridade: 3*
- *Ações:*
 - *Mapear recursos a serem recomendados de acordo com o perfil de cada aluno, segundo os critérios de um recurso ser considerado melhor, da subseção 5.3.3 (podendo, também, utilizar o algoritmo Apriori)*
 - *Enviar mensagem ao Recommended Resources Agent contendo os dados dos alunos e os respectivos recursos recomendados a eles – Mensagem 2*

DES5 – Recomendar recursos para estudantes no AVA:

- *Agente: Recommended Resources Agent*
- *Pré-condições: Obtenção da Mensagem 2 do Profile Situation Assessment Agent*
- *Pós-condições: Recurso recomendado ao estudante*
- *Desejosincompatíveis:5*
- *Prioridade: 4*
- *Ações:*
 - *Registrar no Banco de Dados do AVA os recursos recomendados para cada estudante. Executar uma instância da função `executa_estratégia(...)`*

Tanto o *Profile Situation Assessment Agent* quanto o *Recommended Resources Agent* foram desenvolvidos em XML no JADDEX. Porém, os dois agentes tiveram um código Java associado a eles, o *ArcareManager.java*, que consiste na implementação das intenções (planos) dos agentes. A Figura 51 a seguir mostra um trecho do código do *Profile Situation As-*

sessment Agent em XML, especialmente se referindo ao DES2, indicado pela tag <goals>. Importante destacar que para os desejos serem executados, precisa-se de um desejo auxiliar, o inicializador (*rp_initiate*). A instância da função f_{sit} *verifica_e_combina(...)* aqui é *df_search*, onde DF significa Directory Facilitator, que tem a mesma utilidade do DF dos agentes feitos em JADE. A mensagem de requisitar uma recomendação para o *Recommended Resources Agent* é executada pelo evento *request_recommendation*, onde a performativa é a SFi-pa.REQUEST, em que o padrão de mensagens é iniciado por "start". Por fim, é verificado se a mensagem foi ou não enviada com sucesso. Por sua vez, as Figuras 52 a 54 mostram trechos do *ArcareManager.java* que são responsáveis pela execução dessa parte. Embora a função (nesse caso, método) *verifica_e_combina(...)* não esteja recebendo nenhum parâmetro nesta implementação, as informações de que ela precisa para executar suas ações são provenientes do banco de dados do AVA.

Figura 51. Trecho de código XML do agente *Profile Situation Assessment Agent*.

```

<goals>
    <!-- Include request goal type from dfcap.
Inicialização do desejo. -->
    <achievegoalref name="rp_initiate">
        <concrete ref="procap.rp_initiate"/>
    </achievegoalref>

    <!--DES2-desejo de obter os alunos que fizeram as atividades -->
    <achievegoalref name="df_search">
        <concrete ref="dfcap.df_search"/>
    </achievegoalref>
</goals>

<plans>
    <!-- Plano de enviar uma mensagem (MSG1) de requisição ao RecommendedRe-
sources Agent -->
    <plan name="msg1">
        <body class="ArcareManager"/>
        <waitqueue>
            <messageevent ref="request_recommendation"/>
        </waitqueue>
    </plan>
</plans>

<events>
    <messageevent name="request_recommendation" direction="receive"
type="fipa">

        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.REQUEST</value>
        </parameter>
        <match>$content instanceof String &&&
((String)$content).startsWith("start")</match>

    </messageevent>
    <!-- The answer message after success. -->
    <messageevent name="inform" direction="send" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.INFORM</value>
        </parameter>
    </messageevent>
    <!-- The answer message after some error occurred. -->
    <messageevent name="failure" direction="send" type="fipa">
        <parameter name="performative" class="String" direction="fixed">
            <value>SFipa.FAILURE</value>
        </parameter>

```

Fonte: o autor.

Figura 52. Trecho do Plano (Intenção) *AcareManager.java* que diz respeito ao DES2 (parte 1 de 3).

```
public class ArcareManager extends Plan
{
    //----- attributes -----
    //DB
    private Connection conn;
    private Statement stmt, stmt2, stmt3, stmt4, stmt5, stmt6, stmt7, stmt8;
        // JDBC driver name and database URL
        static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";

/**
 * Create a new plan.
 */
public ArcareManager()
{
    getLogger().info("Created:"+this);
}

//----- methods -----

/**
 * The plan body.
 */
public void body()
{
    boolean rslt = true;
        Integer maxIdx = null;

        try {

                                //STEP 2: Register JDBC driver
//STEP 3: Open a Connection
.
.
.
        //STEP 4: Execute a query
                System.out.println("Creating statement...");
                stmt = conn.createStatement();
                stmt2 = conn.createStatement();
                stmt3 = conn.createStatement();
                stmt4 = conn.createStatement();
                stmt5 = conn.createStatement();
                stmt6 = conn.createStatement();
                stmt7 = conn.createStatement();
                stmt8 = conn.createStatement();
```

Fonte: o autor.

Figura 53. Trecho do Plano (Intenção) *AcareManager.java* que diz respeito ao DES2 (parte 2 de 3).

```
public void verifica_e_combina()
{

    String sql,sql2,sql8=" ";

    sql = "select * from vitor_workshop where workshopid in ( 61, 62, 63, 64, 65, 66, 92,
93, 94, 95, 96 ,97,82 )";

    ResultSet rs = stmt.executeQuery(sql);

    //STEP 5: Extract data from result set
    while(rs.next()){
        //Retrieve by column name
        int id = rs.getInt("workshopid");

//Obter as respostas dos alunos

        String sql4 = "select * from mdl_workshop_submissions
where workshopid= "+id +
                    " and authorid in ( SELECT mdl_user.id as ID
FROM mdl_role_assignments " +
                    " JOIN mdl_user ON
mdl_user.id = mdl_role_assignments.userid JOIN mdl_context ON
mdl_role_assignments.contextid = " +
                    " mdl_context.id WHERE
mdl_role_assignments.roleid = 5 AND mdl_context.contextlevel = 50 AND
mdl_context.instanceid = 253 ) ";

        ResultSet rs4= stmt4.executeQuery(sql4);
        if(rs4.first()){

            int author= rs4.getInt("authorid");
            int sid= rs4.getInt("id");
            //obter os alunos que fizeram as atividades
            String sql5 = "select * from vitor_user where
userid="+author;

            ResultSet rs5= stmt5.executeQuery(sql5);
            if(rs5.first()){
                //selecionar um revisor que nao seja o proprio autor,
                nem que esteja revisando algum trabalho (exceto se não tiver revisores restantes)
                //estilos de aprendizagem
                int kolb_pragmatico=
rs5.getInt("init_kolb_pragmatico");
                int kolb_activo= rs5.getInt("init_kolb_activo");
                int kolb_reflexivo=
rs5.getInt("init_kolb_reflexivo");
                int kolb_teorico=
rs5.getInt("init_kolb_teorico");
```

Fonte: o autor.

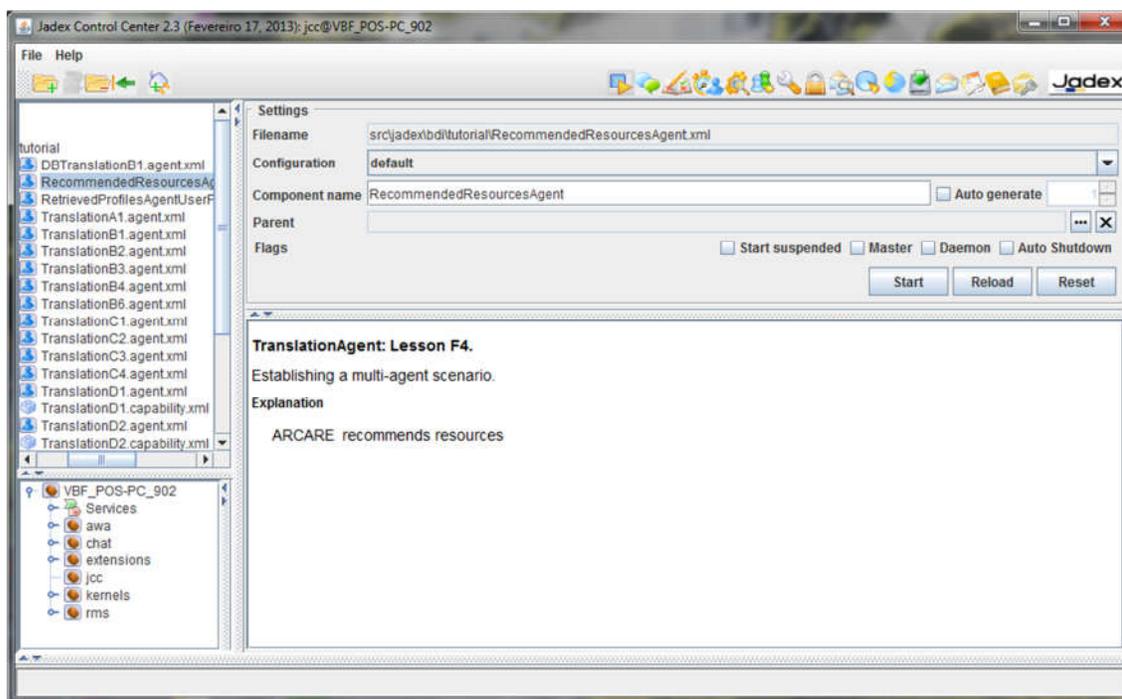
Figura 54. Trecho do Plano (Intenção) *AcareManager.java* que diz respeito ao DES2 (parte 3 de 3).

```
int fs_a_r= rs5.getInt("init_fs_a_r");
int fs_s_i= rs5.getInt("init_fs_s_i");
int fs_v_v= rs5.getInt("init_fs_v_v");
int fs_s_g= rs5.getInt("init_fs_s_g");
//skills
int simulation= rs5.getInt("simulation");
int dom_prog= rs5.getInt("dom_prog");
int cond= rs5.getInt("cond");
int rep= rs5.getInt("rep");
int google_abstraction=
rs5.getInt("google_abstraction");
int google_alg_project=
rs5.getInt("google_alg_project");
int google_patt_recogn_data_an=
rs5.getInt("google_patt_recogn_data_an");
int google_decomposition=
rs5.getInt("google_decomposition");
//cluster, para saber a que grupo o alu-
no pertence: 0 ou 1?
int cluster = rs5.getInt("cluster");
}
```

Fonte: o autor.

Os agentes são executados na plataforma do JADEX em sua versão 2.3 pelo *Control Center*, conforme mostrado na Figura 55. Cada agente é inicializado quando é selecionado (no menu a esquerda) e o botão *Start* é clicado. É importante ressaltar que os agentes JADEX criados para o uso no minicurso de PC utilizaram a estrutura já montada de um exemplo que está incluído nos pacotes do JADEX, cujo exemplo consiste em executar traduções de palavras em inglês para o alemão. Isso explica o porquê de existirem outros agentes na Figura 55, embora não utilizados.

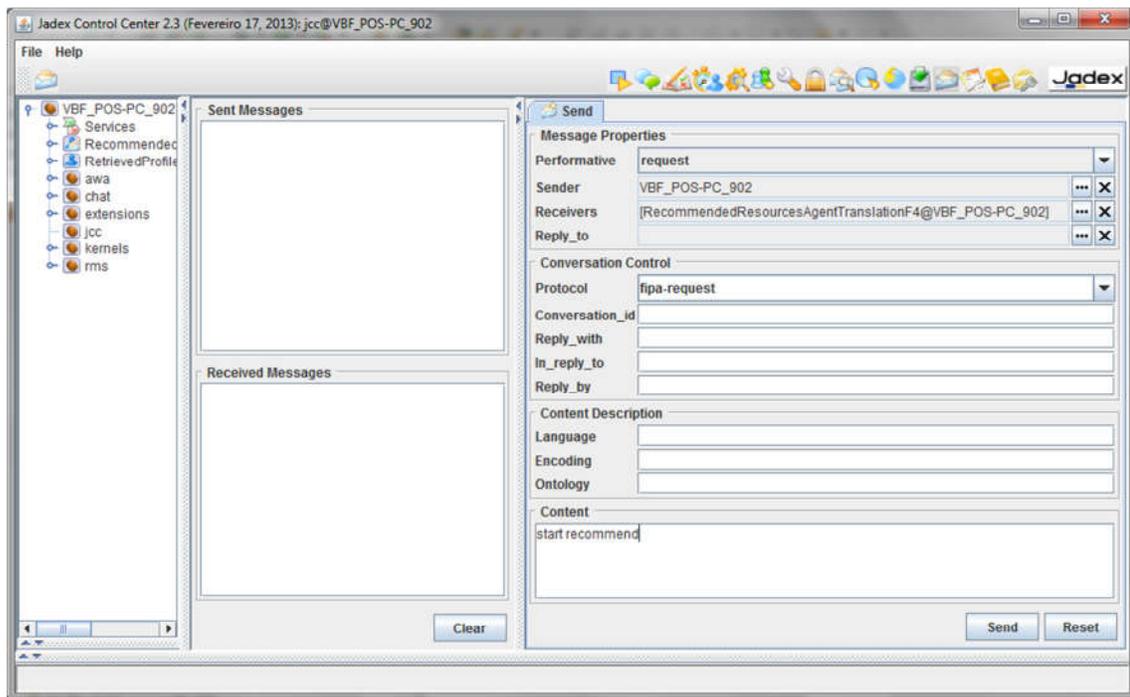
Figura 55. JADEX Control Center com os Agentes de Adaptação de Recursos a serem executados.



Fonte: o autor.

Por sua vez, as Figura 56 e 57 mostram, respectivamente, uma requisição (manual, embora possa haver requisições automáticas pelo *Profile Situation Assessment Agent* de acordo com o propósito da aplicação) sendo preparada ao *Recommended Resources Agent* e depois enviada, com a instrução "*start recommend*", onde "*recommend*" foi uma palavra-chave criada para requisitar uma recomendação ou adaptação de recursos. Todas essas conversações foram permitidas pela ferramenta *Conversation Center* do JADEX. O protocolo utilizado foi o *FIPA.request*.

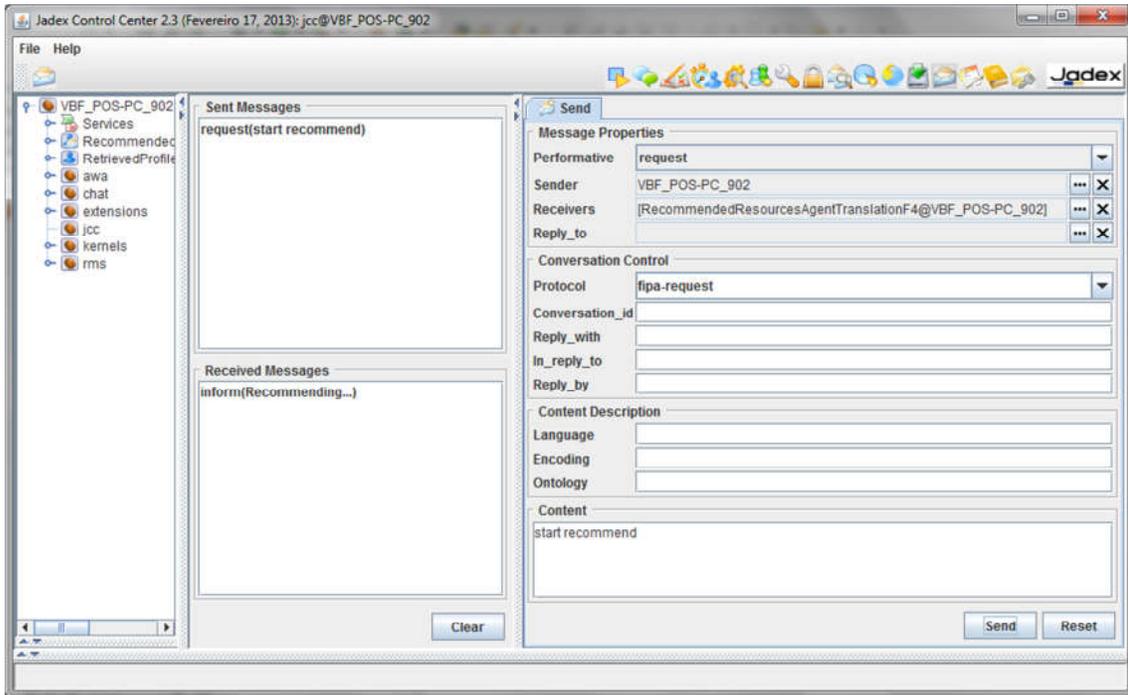
Figura 56. Requisitando uma recomendação ao *Recommended Resources Agent*.



Fonte: o autor.

O *Recommended Resources Agent* fez a escolha de estudantes para corrigir os trabalhos de seus colegas baseados em seus perfis, na fase de avaliação das APs. Estudantes de diferentes perfis foram escolhidos para formar seus pares de executor-avaliador dentro das atividades. Em cada registro de estudante no AVA, o ambiente percebe continuamente, com a ajuda dos agentes e da ontologia, quem são os estudantes mais adequados para corrigirem as atividades de seus colegas.

Figura 57. Requisição atendida.



Fonte: o autor.

6.3.4 Algoritmos de Mineração de Dados Usados no AMPARAX

Para a escolha dos pares que irão avaliar seus colegas, usamos o algoritmo K-Means na ferramenta Weka (2017), considerando os níveis de habilidades dos alunos, independentemente dos estilos de aprendizagem e interesses. Partimos da premissa que estudantes com diferentes habilidades, estilos de aprendizagem e interesses poderiam interagir em atividades colaborativas usando diferentes pontos de vista, com base em Castro e Menezes (2011). A mineração de dados aqui pode ser muito útil em descobrir informações valiosas que podem ser usadas para caracterizar estudantes baseados em seus registros de ações no AVA. A clusterização tem como objetivo particionar n ($n = 33$ perfis) observações em K clusters, nos quais cada observação pertence ao cluster com a média mais próxima. Utilizamos o algoritmo de clusterização K-Means e definimos 2 clusters ($K = 2$), dividindo os 2 grupos (avaliadores e avaliados), onde obtivemos os seguintes resultados: Cluster 0 (níveis de habilidades menores): 20 estudantes (60,61%); Cluster 1 (níveis de habilidades maiores): 13 estudantes (39,39%). Cada estudante do Cluster 0 interagiu com no mínimo um estudante do Cluster 1 e vice-versa.

Para a recomendação de recursos baseados em APs, usamos a f_{sit} *verifica_e_combina(...)*, na qual é usada para a recomendação de recursos úteis (no critério 3, ex-

plicado na subseção 4.4.3) utilizamos o algoritmo Apriori (Agrawal e Srikant, 1994), que é um dos algoritmos mais utilizados de regras de associação, com um suporte mínimo de 10% e confiança mínima 90%. Usamos como atributo de classe o nível de utilidade do recurso recomendado (ur_r) e outros atributos listados na Tabela 12, que são justamente as características usadas para a adaptação de recursos nesse curso de PC, ou seja, o modelo do recurso utilizado.

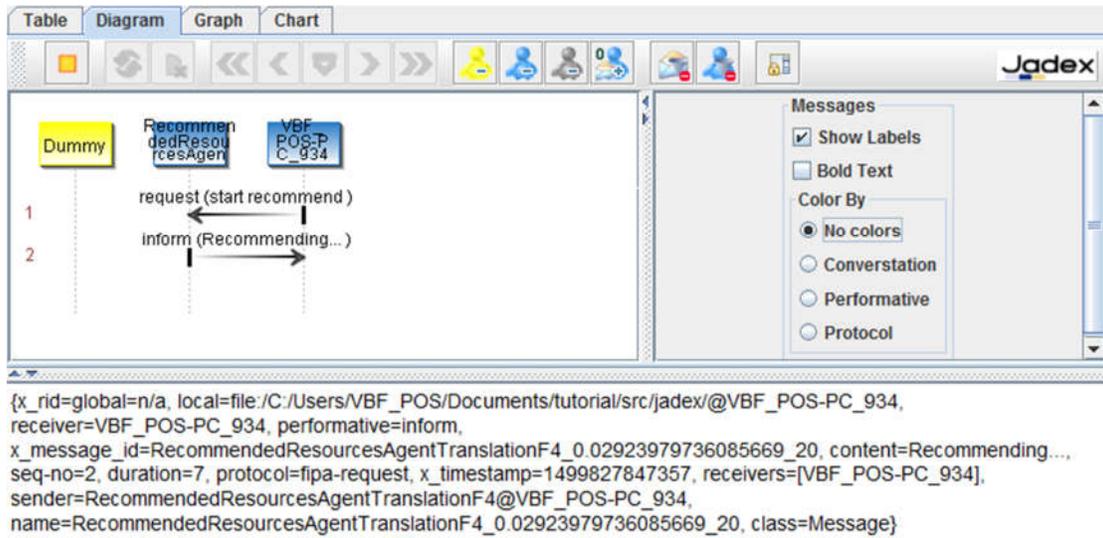
A respeito da execução dos agentes de recomendação de recursos, a Figura 58 mostra as trocas de mensagens entre o *Profile Situation Assessment Agent* (uma instância do *Agente de Avaliação de Situação de Perfis*, na Figura 58 chamado de *VBF_POS-PC_934*) e *Recommended Resources Agent* (uma instância do *Agente Adaptador de Recursos*), através do uso da ferramenta do *JADEX Communication Analyzer*, que permite a visualização da troca de mensagens. Há um terceiro agente, o *Dummy*, responsável por auxiliar os diálogos entre os agentes.

Tabela 12. Atributos dos estudantes usados no algoritmo Apriori.

Atributo	Definição	Valores Possíveis
ativo/reflectivo, sensorial/intuitivo, visual/verbal e sequencial/global	Estilos de Aprendizagem de Felder-Silverman	-11 to +11, valores ímpares
Ativo, reflexivo, pragmático, teórico	Estilos de Aprendizagem de Honey-Alonso (Kolb)	0% a 100%
Domínio em projeto de algoritmos e programação, abstração, decomposição de problemas, simulação, reconhecimento de padrões e análises de dados, domínio em estruturas de condição e repetição.	Habilidades	0,0 a 10,0
Utility_level (ur_r), em inglês	Nível de Utilidade de um Recurso	0,0 a 1,0

Fonte: o autor.

Figura 58. Troca de mensagens entre os agentes de adaptação e recomendação de recursos feitos no JADEX.



Fonte: o autor.

6.3.5 Resultados Obtidos com o Uso do AMPARAX

A seguir, serão apresentadas as opções de configurações a respeito do ColabWeb, tanto por parte do professor quanto por parte dos alunos. Em seguida, resultados obtidos com nossa abordagem do ArCARE no ColabWeb serão mostrados.

No minicurso, para o professor, houve uma página extra onde o mesmo poderia configurar as opções de adaptação e também os níveis de habilidades que ele julgasse adequados para as atividades que os alunos fossem executar. A Figura 59 mostra a página de configurações do ponto de vista do professor.

Figura 59. Página de configurações do professor.

ArCARE

Configurações do Professor

Permissões de Adaptação no ColabWeb:

- Estilos de Aprendizagem
- Habilidades
- Interesses

Níveis de habilidades (Escala variando de 1 a 10):

Configure os níveis de habilidades da seguinte Unidade de Ensino Obrigatória:

[Atividade sobre Explorando Algoritmos...](#)

1. Domínio em programação:
2. Projetar Algoritmos:
3. Abstração:
4. Decomposição:
5. Reconhecimento de padrões e/ou Análise de Dados:

Fonte: o autor.

Quanto ao MAA, por sua vez, este deveria ser facilmente entendido pelo estudante. Com isso, procuramos desenvolver uma interface fácil de utilizar e compreender por parte do aluno a respeito de seu MAA. A Figura 60 mostra links no ColabWeb para o estudante acessar o seu MAA, em uma página *profile.php*, e realizar configurações como lhe convier no minicurso, além de poder acessar e avaliar os recursos recomendados e adaptados a ele.

Figura 60. Link para acesso ao MAA e configurações do minicurso pelo estudante.



Fonte: o autor.

Já a Figura 61 mostra um exemplo de MAA apresentado ao estudante, na página *profile.php*, no formato de gráfico de barras. O gráfico apresentado mostra os níveis (que variam de 0 a 10) das 8 habilidades utilizadas nesse minicurso de um estudante em um determinado momento, além dos seus estilos de aprendizagem de Honey-Alonso (em escalas percentuais) e Felder-Silverman (em escalas de -11 a +11).

Figura 61. Apresentando o MAA aos estudantes.



Fonte: o autor.

Por sua vez, a Figura 62 mostra opções para os estudantes selecionarem quais características eles preferem utilizar para que o processo de adaptação de recursos seja executado no minicurso de Pensamento Computacional. E por fim, a Figura 63 apresenta todos os recursos recomendados a um estudante para que ele selecione o seu nível de satisfação para cada recurso recomendado segundo a escala Likert (1 – Muito Inútil a 5 – Muito Útil). Além disso, o estudante pode acessar o recurso clicando no link em que aparece o próprio nome do recurso, a fim de que possa acessá-lo mais rápido.

Figura 62. Opções para o estudante escolher que adaptações de recursos gostaria de ter no ColabWeb.

Fonte: o autor.

Figura 63. Avaliando os recursos adaptados e recomendados por níveis de satisfação.

Avalie e/ou acesse o seguinte recurso adaptado e recomendado para você:

[AP Scratch 21 Discussão em Pensamento Computacional Adaptada](#)

1 - Muito Inútil 2 - Pouco Inútil 3 - Neutro 4 - Pouco Útil 5 - Muito Útil

Avalie e/ou acesse o seguinte recurso adaptado e recomendado para você:

[AP Scratch 22 Discussão em Pensamento Computacional Adaptada](#)

1 - Muito Inútil 2 - Pouco Inútil 3 - Neutro 4 - Pouco Útil 5 - Muito Útil

Fonte: o autor.

Após a execução do pós-teste (ou teste final), comparamos as notas dos alunos obtidas no pré-teste com as notas do pós-teste. A Tabela 13 apresenta as notas obtidas pelos estudantes durante o minicurso avaliado pela pesquisa. Comparando as médias da turma no pré- e no pós-teste, em uma escala de 0 a 10, a turma obteve uma média de 4,93 com um desvio-padrão de 3,86, com 39,39% de estudantes aprovados (quando a nota obtida é maior ou igual a 6), enquanto que no pós-teste a média da turma foi 8,35 com um desvio-padrão de 3,57, um aumento na média geral de 169,37%, com 84,84% de estudantes aprovados.

Tabela 13. Notas dos alunos obtidas no pré-teste e no teste final.

Aluno	Pré-Teste	Teste Final
Aluno 1	8,57	7,14
Aluno 2	5,71	0
Aluno 3	1,43	10
Aluno 4	8,57	10
Aluno 5	10	10

Aluno 6	4,29	8,57
Aluno 7	1,43	10
Aluno 8	0	10
Aluno 9	3,57	10
Aluno 10	0	10
Aluno 11	0	10
Aluno 12	0	10
Aluno 13	0	10
Aluno 14	10	10
Aluno 15	2,86	10
Aluno 16	8,57	10
Aluno 17	4,29	10
Aluno 18	10	10
Aluno 19	2,86	10
Aluno 20	1,43	10
Aluno 21	2,86	10
Aluno 22	2,86	10
Aluno 23	10	10
Aluno 24	4,29	10
Aluno 25	7,14	10
Aluno 26	3,57	10
Aluno 27	10	10
Aluno 28	8,57	10
Aluno 29	10	0
Aluno 30	10	10

Aluno 31	0	0
Aluno 32	0	0
Aluno 33	10	0
Média das Notas	4,93	8,35
Desvio-padrão	3,86	3,57

Fonte: o autor.

Aplicamos o teste de normalidade Shapiro-Wilk nos dados da Tabela 13, e obtivemos $W = 0,73483$ e $p\text{-valor} = 0,000000001349$, comprovando que esses dados seguem uma distribuição normal. Com isso, fizemos um teste t estatístico pareado (pois é sobre a mesma turma) para testar as seguintes hipóteses com base nos resultados mostrados na Tabela 13:

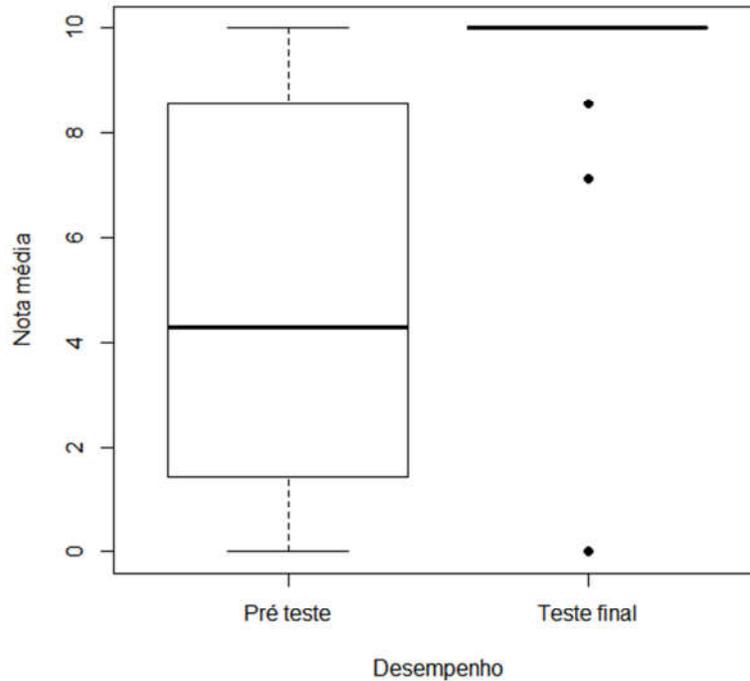
- H_0 : não há diferença entre as notas dos alunos nos testes aplicados e
- H_1 : há diferença entre as notas dos alunos nos testes aplicados.

Aplicando o teste t, os seguintes resultados foram produzidos:

$t = -3,5983$, $p\text{-valor} = 0,001066$.

Dessa forma, aplicando um intervalo de confiança de 95%, **rejeita-se H_0 e aceita-se H_1** , pois $p\text{-valor} < 0,05$. Ou seja, pode-se afirmar que existem diferenças significativas entre as notas obtidas pelos alunos no pré-teste e no pós-teste. A Figura 64 exhibe graficamente, em *box plots*, a dispersão observada pelos dados obtidos em ambos os testes. Todos esses testes estatísticos foram realizados no software RStudio (2017).

Figura 64. Dispersão das notas obtidas (pré-teste e teste final) pela turma.



Fonte: o autor.

A fim de sabermos as opiniões dos estudantes sobre o minicurso adaptado, perguntamos aos estudantes sobre as atividades realizadas, onde o questionário completo está no Apêndice D desta tese. As perguntas foram: Q1) Você avaliou algum colega? Q2) Você recebeu alguma recomendação de recurso educacional? Q3) Você concorda com seu MAA apresentado a você? Q4) Os recursos apresentados a você foram úteis? A Tabela 14 apresenta as porcentagens das respostas dos estudantes às questões. Conforme a Tabela 14, a maioria dos estudantes na turma avaliaram pelo menos um colega (83,87%), receberam pelo menos uma recomendação de um recurso educacional (87,09%), concordaram com o MAA apresentado (83,87%) e consideraram os recursos recomendados úteis (96,42%). Porém, na pergunta Q3, 3,58% da turma (1 estudante) respondeu “Depende”, visto que ele não concordou no início do curso com seu MAA ou discordou das recomendações de recursos feitas a ele.

Pode ser visto nesses testes que adaptações estão sendo realizadas com sucesso com a abordagem do ArCARE, uma vez que a adaptação e recomendação de atividades baseadas em APs por agentes e uma ontologia de MAA é uma técnica para melhorar o aprendizado do es-

tudante e seu engajamento no curso, estimulando a prática de atividades colaborativas em AVAs.

Tabela 14. Respostas dos estudantes no questionário.

Respostas	Perguntas			
	Q1	Q2	Q3	Q4
Sim	83,87%	87,09%	83,87%	96,42%
Não	16,13%	12,91%	12,91%	3,58%
Depende	0%	0%	3,22%	0%

Fonte: o autor.

Ao verificarmos a evolução dos níveis de habilidades dos alunos no decorrer do mini-curso, as Tabelas 15 e 16 apresentam, respectivamente, os demais níveis de habilidades dos alunos após o pré-teste e após o pós-teste. Observou-se pelos dados da Tabela 16 que houve um aumento desses níveis. As habilidades, nas duas tabelas, foram enumeradas de (1) a (8), sendo: (1) Simulação; (2) Domínio em programação; (3) Domínio em estruturas condicionais; (4) Domínio em estruturas de repetição; (5) Abstração; (6) Domínio em projeto de algoritmos; (7) Reconhecimento de padrões e análise de dados e (8) Decomposição de problemas.

Tabela 15. Níveis de habilidades dos estudantes após o pré-teste.

Aluno	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Aluno 1	2	2	2	2	3	2	2	3
Aluno 2	2	2	2	2	3	2	2	3
Aluno 3	0	0	0	0	0	0	0	0
Aluno 4	2	0	2	2	0	0	2	0
Aluno 5	0	2	0	0	3	2	0	3
Aluno 6	2	2	2	2	3	2	2	3
Aluno 7	2	2	2	2	3	2	2	3
Aluno 8	0	2	0	0	3	2	0	3
Aluno 9	4	4	4	4	4	4	4	4

Aluno 10	2	2	2	2	3	2	2	3
Aluno 11	0	0	0	0	4	0	0	4
Aluno 12	0	0	0	0	1	0	0	1
Aluno 13	2	2	2	2	3	2	2	3
Aluno 14	2	2	2	2	3	2	2	3
Aluno 15	0	4	0	0	3	4	0	3
Aluno 16	0	0	0	0	0	0	0	0
Aluno 17	2	0	2	2	2	0	2	2
Aluno 18	0	0	0	0	0	0	0	0
Aluno 19	0	0	0	0	0	0	0	0
Aluno 20	0	0	0	0	0	0	0	0
Aluno 21	2	0	2	2	3	0	2	3
Aluno 22	0	0	0	0	2	0	0	2
Aluno 23	0	0	0	0	0	0	0	0
Aluno 24	0	0	0	0	0	0	0	0
Aluno 25	2	0	2	2	0	0	2	0
Aluno 26	2	2	2	2	3	2	2	3
Aluno 27	0	0	0	0	0	0	0	0
Aluno 28	0	0	0	0	0	0	0	0
Aluno 29	0	0	0	0	2	0	0	2
Aluno 30	0	0	0	0	0	0	0	0
Aluno 31	0	0	0	0	0	0	0	0
Aluno 32	4	0	4	4	1	0	4	1
Aluno 33	0	0	0	0	2	0	0	2
Média das Notas	0,97	0,85	0,97	0,97	1,64	0,85	0,97	1,64

Desvio-padrão	1,24	1,23	1,24	1,24	1,48	1,23	1,23	1,48
----------------------	------	------	------	------	------	------	------	------

Fonte: o autor.

Tabela 16. Níveis de habilidades dos estudantes após o pós-teste.

Aluno	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Aluno 1	1	0	0	0	1	1	0	2
Aluno 2	9	7	8	7	10	9	7	10
Aluno 3	4	3	3	3	4	4	3	5
Aluno 4	10	10	10	10	10	10	10	10
Aluno 5	5	6	4	4	10	8	3	10
Aluno 6	0	0	0	0	1	0	0	1
Aluno 7	9	8	8	8	10	9	8	10
Aluno 8	2	0	1	0	2	1	0	2
Aluno 9	1	0	0	0	1	0	0	1
Aluno 10	0	0	0	0	0	0	0	0
Aluno 11	8	6	6	6	10	7	7	10
Aluno 12	10	10	8	8	10	10	8	10
Aluno 13	3	0	0	0	2	2	1	3
Aluno 14	0	0	0	0	0	0	0	0
Aluno 15	7	10	6	6	10	10	6	10
Aluno 16	0	0	0	0	0	0	0	0
Aluno 17	9	6	9	9	10	6	9	10
Aluno 18	1	0	0	0	1	1	0	2
Aluno 19	2	0	1	0	2	2	0	3
Aluno 20	1	0	1	1	2	1	0	2

Aluno 21	10	6	9	9	10	8	9	10
Aluno 22	6	6	6	6	10	6	6	10
Aluno 23	5	3	3	3	5	4	3	5
Aluno 24	1	0	1	1	2	1	0	2
Aluno 25	10	6	10	9	10	7	9	10
Aluno 26	8	6	7	7	10	8	6	10
Aluno 27	1	0	0	0	1	0	0	1
Aluno 28	1	0	0	0	8	1	0	8
Aluno 29	1	0	1	1	2	1	0	2
Aluno 30	1	0	0	0	1	1	0	2
Aluno 31	1	0	1	0	2	1	0	2
Aluno 32	1	0	0	0	1	1	0	1
Aluno 33	4	6	3	3	10	7	3	10
Média das Notas	4	3	3,22	3,07	5,1	3,85	2,97	5,28
Desvio-padrão	3,7	3,64	3,63	3,6	4,28	3,69	3,64	4,15

Fonte: o autor.

Aplicamos o teste de normalidade Shapiro-Wilk nos dados das Tabelas 15 e 16, para cada uma das 8 habilidades, e obtivemos os valores de W e p-valor apresentados na Tabela 17, comprovando que esses dados seguem uma distribuição normal. Com isso, fizemos um teste t estatístico pareado (pois é sobre a mesma turma), em cada uma das 8 habilidades, para testar as seguintes hipóteses com base nos resultados mostrados nas Tabelas 15 e 16:

- H0: não há diferença entre os níveis de habilidades dos alunos nos testes aplicados e
- H1: há diferença entre os níveis de habilidades dos alunos nos testes aplicados.

Aplicando o teste t, os resultados de t e p-valor estão apresentados na Tabela 17, para cada habilidade. Aplicamos um intervalo de confiança de 95%. Podemos perceber, com os resultados da Tabela 16, que em todos os casos, **rejeita-se H0** e **aceita-se H1**, pois p-valor < 0,05. Ou seja, pode-se afirmar que existem diferenças significativas entre os níveis de habili-

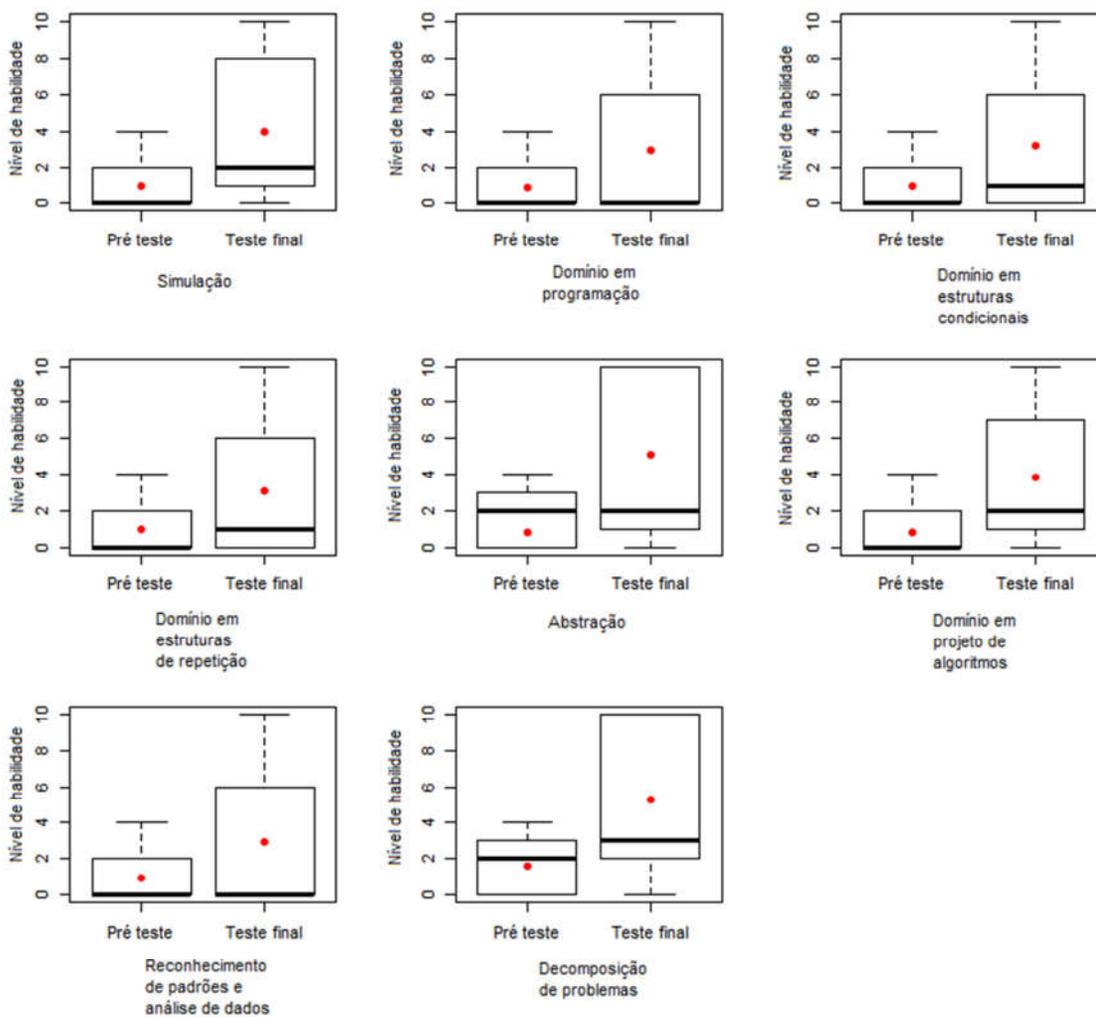
dades dos alunos no pré-teste e no pós-teste. A Figura 65 exibe graficamente, em *box plots*, a dispersão observada em cada nível de habilidade. Os pontos vermelhos são as médias, de acordo com a Tabela 16.

Tabela 17. Resultados dos testes de normalidade e teste t aplicados aos níveis de habilidades dos alunos.

Habilidade	Shapiro-Wilk		Teste t	
	W	p-valor	t	p-valor
Simulação	0,75816	0,000000004479	4,6371	0,00005688
Domínio em programação	0,70249	0,000000002853	3,295	0,002411
Domínio em estruturas condicionais	0,73582	0,000000001417	3,5139	0,001341
Domínio em estruturas de repetição	0,7288	0,000000001001	3,3047	0,00235
Abstração	0,76821	0,000000007682	4,7013	0,00004725
Domínio em projeto de algoritmos	0,74784	0,000000002611	4,5161	0,00008061
Reconhecimento de padrões e análise de dados	0,716	0,000000000538	3,1489	0,003539
Decomposição de problemas	0,77829	0,00000001341	5,0681	0,0000163

Fonte: o autor.

Figura 65. Dispersão dos níveis de habilidades (pré-teste e teste final) da turma.



Fonte: o autor.

6.4 Conclusões do Capítulo

Nos testes realizados nesse minicurso de Pensamento Computacional, é possível observar que há um processo de adaptação de recursos recomendados a alunos por meio de APs recomendadas no Moodle. Além disso, ocorre a adaptação das APs pela escolha de pares de alunos a interagirem entre si. Com os resultados do pré- e pós testes e das evoluções dos níveis de habilidades de alunos, percebemos que houve aprendizado por parte do aluno, utilizando o processo de adaptação de recursos educacionais fornecidos pelo sistema criado a partir do framework ArCARE no AVA Moodle. Além disso, em sua maioria, os alunos gostaram dessa prática de ensino-aprendizagem oferecida pela ArCARE, de acordo com suas respostas dos questionários de pesquisa de opinião.

Nos testes aplicados, se verificarmos em todas as características do ArCARE mostradas na Tabela 9 da Seção 4.5, percebemos que somente não foi utilizada a adaptação baseada no equipamento que o aluno utiliza (dispositivo móvel ou um desktop), visto que tal equipamento utilizado não tinha relevância para o minicurso e o cenário de testes para o qual ele foi aplicado, embora tal característica possa ser utilizada em projetos futuros.

Percebemos que as práticas pedagógicas aplicadas, neste Capítulo, apontam para uma concepção de conhecimento construtivista e buscam promover a transformação e a aprendizagem cooperativa/colaborativa e por autorreflexão, mediante a troca de saberes entre os alunos, constituindo-se em uma comunidade de aprendizagem (Novak et al., 2014) adaptada pelos mecanismos de adaptação fornecidos pelo ArCARE.

O Capítulo 7, a seguir, apresenta as considerações finais dessa tese, com as suas contribuições, publicações e propostas de trabalhos futuros.

CAPÍTULO 7 – CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as conclusões sobre o trabalho realizado, as suas contribuições, trabalhos futuros e publicações produzidas como resultado desta pesquisa.

Com o abrangente uso de AVAs, cada vez mais é notória a necessidade de que eles forneçam serviços adaptados de acordo com as necessidades dos alunos. Sendo assim, nesta tese, uma Revisão Sistemática de literatura foi feita para encontrar trabalhos relacionados com adaptação de recursos em AVAs e com os resultados desta revisão, surgiu a nossa proposta descrita nesta tese. Dessa forma, este trabalho mostrou um arcabouço, o ArCARE, que permite adaptação de recursos em AVAs baseada na teoria construtivista da aprendizagem, propiciando o desenvolvimento de *Smart Learning Environments*.

Dois cenários foram utilizados para realizar a prova de conceito do modelo apresentado. No primeiro, onde criamos um sistema chamado AMPARA, foram desenvolvidos um sistema multiagente e uma ontologia de Modelo Aberto de Aluno no padrão IMS LIP capazes de selecionar técnicas de adaptação no AVA Moodle em cursos com currículos formais. Esta estratégia permite uma maior personalização dos recursos apresentados em função das características dos alunos. O processo de adaptação descrito na arquitetura do AMPARA tinha como objetivo aumentar a interação entre os alunos de um mesmo curso ou disciplina e a interação deles com o AVA, aumentando assim o aprendizado efetivo e maiores acertos nas tarefas executadas. No entanto, o AMPARA não contemplava algumas possibilidades que o ArCARE poderia prover no processo de adaptação de recursos em AVAs.

A fim de utilizar de uma forma mais completa as funcionalidades providas pelo ArCARE, executamos um outro teste, consistindo na extensão do AMPARA chamada AMPARAX, que também se constitui em um sistema multiagente associado a uma ontologia de MAA que atuou no Moodle, mas agora trabalhando com a arquitetura BDI de agentes, learning analytics, algoritmos de mineração de dados, históricos das ações dos estudantes no AVA e cursos com currículos flexíveis.

Os resultados obtidos nos testes realizados, nos dois estudos de casos, mostraram que as adaptações fornecidas pelo ArCARE, seja a recomendação de alunos para ajudar os seus colegas com dúvidas, a formação de pares para interação e correção de atividades, além da

recomendação de atividades, baseadas em Arquiteturas Pedagógicas e através da aprendizagem colaborativa, são soluções baseadas no ArCARE que podem ajudar a construir o conhecimento do aluno de forma útil e eficaz e que é uma possível solução para o problema da falta de disposição personalizada de recursos para os alunos que acessam e utilizam AVAs. A abordagem do ArCARE pode ser aplicada em qualquer AVA, uma vez que os agentes e o ambiente educacional podem compartilhar a mesma base de dados.

7.1 Contribuições deste Trabalho

Algumas contribuições para a comunidade científica e para a sociedade em geral foram alcançadas através do trabalho executado durante o desenvolvimento desta tese. Essa Seção, em consonância com os objetivos levantados no Capítulo 1, reúne e descreve as mais relevantes delas.

Um importante resultado obtido é a discussão levantada sobre as técnicas de adaptação em AVAs de um ponto de vista construtivista. Procurou-se identificar, com base em uma fundamentação sólida, quais as principais técnicas de adaptação de recursos em AVAs. O produto dessa investigação, descrito ao longo do Capítulo 3 e do Apêndice F, é um conjunto de ideias que poderiam ser aplicadas para adaptação de recursos em AVAs, utilizando as técnicas mais comumente utilizadas, como aprendizagem colaborativa e adaptação por agentes.

Com base em toda a fundamentação teórica e análise dos trabalhos relacionados, esse trabalho apresentou no Capítulo 4 uma estratégia de solução para o problema do *one-size-fits-all* nos AVAs, por meio de um arcabouço conceitual para adaptação de recursos educacionais em AVAs, o ArCARE. A partir do arcabouço proposto, diversos *Smart Learning Systems* podem ser desenvolvidos, de acordo com os objetivos educacionais de cada disciplina ou curso, como por exemplo, o sistema AMPARA do Capítulo 5 e o AMPARAX do Capítulo 6. Dessa maneira, por meio dessa contribuição, procurou-se ir além da simples constatação de um problema teórico e sua descrição, para se alcançar mudanças qualitativas reais na execução de cursos (ou disciplinas) que utilizam AVAs, sejam presenciais, semipresenciais ou a distância. Foram criados um MAA baseado no padrão IMS LIP, que permitisse a autorreflexão do aluno a respeito de sua própria aprendizagem, e uma estrutura conceitual multiagente que trocasse mensagens entre si. Além disso, ao mesmo tempo, o ArCARE provê adaptação e recomendação de recursos colaborativos em AVAs para estudantes, e permite um currículo flexível no curso em que ele estiver atuando.

Portanto, a principal contribuição do ArCARE é permitir mudanças na organização pedagógica da aprendizagem de acordo com as características dos estudantes utilizando as tecnologias de sistemas multiagente e ontologias baseadas no Modelo Aberto de Aluno. Assim, através da apresentação dessa nova abordagem pedagógica no processo de ensino-aprendizagem, abrem-se caminhos para diferentes visões e avanços de pesquisas educacionais na área de adaptação de AVAs e desenvolvimento de *Smart Learning Systems* (ou *Environments*).

Voltando à questão de pesquisa e às hipóteses apresentadas na Seção 1.3:

QP: Como utilizar tecnologias da IA para conceber ambientes de aprendizagem que se ajustem às necessidades e preferências do aluno?

H1: É possível utilizar modelo de aluno para prover adaptações em AVAs de modo que a aprendizagem do aluno ocorra.

H2: Pode-se utilizar tecnologias de Inteligência Artificial (especificamente, agentes e ontologias) para prover adaptações que permitam a construção do conhecimento do aluno com base em seu modelo de aluno.

A primeira hipótese (H1) buscou comprovar na literatura o uso de modelos de aluno para provisão de adaptações em AVAs. De fato, isso existe e foi discutido nos Capítulos 2, 3 e é apresentado no Apêndice F.

A segunda hipótese (H2), com base na confirmação da primeira hipótese, busca prover adaptação de recursos em função do modelo de aluno, mas com foco na construção do conhecimento do aluno para seu aprendizado efetivo. É visto que já existem trabalhos na literatura que tratam destas questões, como apresentado no Capítulo 3 e no Apêndice F.

Por fim, a QP procurou responder a forma como deve ser feita a adaptação de recursos em AVAs em função do perfil do aluno. É o que esta tese pretendeu demonstrar, com a apresentação do ArCARE. Utilizamos a tecnologia de sistemas multiagente para trabalhar com a ontologia de modelo de aluno. Os resultados dos testes realizados em turmas reais, juntamente com comparações de outros trabalhos na literatura (em relação aos aspectos de adaptação de recursos que eles utilizam), puderam nos dizer que esta abordagem é útil para estimular o aprendizado dos alunos, contribuindo com as pesquisas nessa área. Além disso, foi observado que mostrar o modelo de aluno para o mesmo, caracterizando um MAA, contribuiu com a aprendizagem do estudante.

Com o arcabouço conceitual apresentado nesta tese e suas características descritas no Capítulo 4, além de seus estudos de caso e seus resultados de experimentos como provas de conceito nos Capítulos 5 e 6, pudemos notar que a nossa estratégia pedagógica com o ArCARE foi bem recebida e que pode provocar impactos positivos para o aprendizado dos alunos. Com isso, concluímos que as hipóteses e a questão de pesquisa na Seção 1.3 foram respondidas e os objetivos descritos na Seção 1.4 foram alcançados.

7.2 Trabalhos Futuros

Diferentes trabalhos futuros são vislumbrados para esta pesquisa. Em primeiro lugar, mais experimentos para validação do arcabouço apresentado, o ArCARE, devem ser conduzidos. Dessa forma, através de investigações mais longas e completas, será possível identificar com clareza as reais contribuições alcançadas no processo de adaptação de recursos em AVAs, utilizando aprendizagem colaborativa.

Os próximos passos desta pesquisa, como sugestão de trabalhos futuros para aprimorar o ArCARE, estão apresentados a seguir:

- a) Criar diversos ambientes inteligentes baseados no ArCARE, sendo necessário para isso ajustar o MAA com os dados necessários a respeito do perfil do aluno ou usuário nos ambientes virtuais e das peculiaridades de cada curso ou disciplina, como por exemplo, que habilidades estarão sendo desenvolvidas nos alunos em um curso específico;
- b) Criação de novas arquiteturas pedagógicas flexíveis, permitindo a exploração de novas estratégias de adaptação com base nessas arquiteturas, de acordo com os objetivos e necessidades dos cursos em que elas estiverem inseridas;
- c) Utilizar as estruturas de adaptação de recursos do ArCARE para uso em dispositivos móveis, além de utilizar arquiteturas pedagógicas adaptativas para estes dispositivos, o que poderiam ser chamadas de **arquiteturas pedagógicas móveis**, explorando, por exemplo, os Cenários 4 e 5 do Capítulo 1;
- d) Utilizar as ideias propostas pelo ArCARE em sistemas que trabalhem com MOOCs, fazendo análise em bases de dados com um grande volume de informações. Com isso, aprofundar a utilização do ArCARE em sistemas multiagente que utilizem Learning Analytics, MAA e mineração de dados;
- e) Explorar os cenários ainda não explorados do Capítulo 1, como por exemplo, o Cenário 3. Importante ressaltar que nesse cenário não tratamos de um AVA expli-

citamente, mas sim de uma comunidade de prática, onde a utilização do ArCARE seria uma instância de suas possibilidades de aplicações;

- f) Ao considerar a generalização do framework, como possível extensão, espera-se que ele possa ser utilizado em diferentes casos, especialmente em cursos inclusivos (por exemplo, para alunos portadores de deficiências) e que possa abrir espaço para o trabalho interdisciplinar envolvendo professores e pesquisadores das áreas de Educação e Computação.

7.3 Publicações

O trabalho descrito na presente tese originou artigos aceitos em conferências científicas nacionais e internacionais. São eles:

- **BREMGARTNER, Vitor Frota;** NETTO, J.M. ; MENEZES, C. S. Agent-Based Conceptual Framework for Collaborative Educational Resources Adaptation in Virtual Learning Environments. In: XXVIII Congresso Brasileiro de Informática na Educação 2017, Recife, PE. Anais do SBIE 2017, 2017.
- **BREMGARTNER, Vitor Frota;** NETTO, J.M. ; MENEZES, C. S. Conceptual Framework of Educational Resources Adaptation for Improve Collaborative Learning in Virtual Learning Environments. In: 47th Frontiers in Education Conference, Indianapolis, USA. Proceedings of 47th Frontiers in Education Conference, 2017.
- **BREMGARTNER, Vitor Frota.;** DE MAGALHAES NETTO, J. F. ; MENEZES, C. S. Conceptual Framework for Collaborative Educational Resources Adaptation in Virtual Learning Environments. Proceedings of the 18th International Conference on Artificial Intelligence in Education (AIED 2017), Wuhan, China, 2017.
- **BREMGARTNER, Vitor Frota.;** DE MAGALHAES NETTO, J. F.; MENEZES, C. S. . Arcabouço Conceitual de Adaptação de Recursos Educacionais para Recomendação de Arquiteturas Pedagógicas. In: V Congresso Brasileiro de Informática na Educação (CBIE 2016), Uberlândia, MG. Anais do V Congresso Brasileiro de Informática na Educação, VIII WAPSEDI, 2016.
- **BREMGARTNER, Vitor Frota;** NETTO, J.M. ; MENEZES, C. S. Adaptation Resources in Virtual Learning Environments under Constructivist Approach: A

- Systematic Review. In: 45th Frontiers in Education Conference, 2015, El Paso, USA. Proceedings of 45th Frontiers in Education Conference, 2015.
- **BREMGARTNER, Vitor Frota**; NETTO, J.M. ; MENEZES, C. S. Explorando arquiteturas pedagógicas recomendadas por meio de agentes e ontologia de modelo do aluno em ambientes virtuais de aprendizagem. In: XXVI Congresso Brasileiro de Informática na Educação 2015, Maceió, AL. Anais do SBIE 2015.
 - **BREMGARTNER, Vitor Frota.** ; NETTO, J.M. ; MENEZES, C. S. Using Agents and Open Learner Model Ontology for Providing Constructive Adaptive Techniques in Virtual Learning Environments. In: 14th Ibero-American Conference on Artificial Intelligence - IBERAMIA 2014, Santiago. Proceedings of 14th Ibero-American Conference on Artificial Intelligence - IBERAMIA 2014, 2014.
 - **BREMGARTNER, Vitor Frota.** ; NETTO, J.M. ; MENEZES, C. S. Utilizando Agentes e Ontologia de Modelo de Aluno Aberto para Prover Adaptação de Conteúdos Construtivistas em Ambientes Virtuais de Aprendizagem. In: Simpósio Brasileiro de Informática na Educação - 25º SBIE, Dourados, MS. Anais do 25º Simpósio Brasileiro de Informática na Educação, 2014.
 - **BREMGARTNER, V.** ; DE MAGALHAES NETTO, J. F M. Improving collaborative learning by personalization in Virtual Learning Environments using agents and competency-based ontology. In: 2012 IEEE Frontiers in Education Conference (FIE) Proceedings. Seattle, USA, 2012.
 - **BREMGARTNER, Vitor Frota.** ; NETTO, J.M. Relato de uma Experiência de Auxílio Personalizado a Alunos em Ambiente Virtual de Aprendizagem Utilizando Agentes. In: Congresso Brasileiro de Informática na Educação 2012 (CBIE 2012), Rio de Janeiro, RJ. Anais do 18º Workshop de Informática na Escola (18º WIE), 2012.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ahmad, N. Self-Directed Learning: Student's Interest in Viewing the Learner Model. Proceeding of 3rd International Conference on Research and Innovation in Information Systems (ICRIIS), pp. 493-498, Malásia, 2013.
- Agrawal, R. e Srikant, R. Fast algorithms for mining association rules in large databases. Proceedings of the International Conference on VeryLarge Databases, Santiago, Chile, 1994.
- Alonso, C. e Gallego, D. Questionário Honey-Alonso de Estilos de Aprendizagem. URL: <http://www.estilosdeaprendizaje.es/chaea/chaegrp2.htm>, 2017.
- Alonso, C.M., Gallego, D.J. e Honey, P. Los estilos de aprendizaje: Procedimientos de diagnóstico y mejora. Bilbao: Ediciones Mensajero, 1999.
- AP CollegeBoard Computer Science Principles – Effective Fall 2016, <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>, 2016.
- Araújo, R. D. *et al.* Autoria automática de objetos de aprendizagem a partir de captura multimídia e associação a estilos de aprendizagem. Anais do XXV Simpósio Brasileiro de Informática na Educação, Dourados, MS, 2014.
- Arduino – Home. URL: <https://www.arduino.cc/>, 2017.
- Argote, L. and Ingram, P. Knowledge transfer: A Basis for Competitive Advantage in Firms. *Organizational Behavior and Human Decision Processes*, pp. 150–169, 2000.
- Ausubel, D.P. *The acquisition and retention of knowledge: a cognitive view.* Kluwer Academic Publishers, 2000.
- Ayoola, O. L.; Phelan, E. M. Accumulating a Personalised Ubiquitous Learning Platform for Enhancing the Third Level Educational Experience. *International Conference on Computational Science and Engineering*, 2009.
- Basili, V. R.; Caldiera, G.; Rombach, H. D. *The Experience Factory.* Marciniak, J. J. (Ed). In: *Encyclopedia of Software Engineering*, New York, John Wiley & Sons, 1994.
- Bastos Filho, O. C.; Axt, M.; Labidi, S.; Silveira, P. D.; Thomaz, A. R. Sistema Inteligente de Desafios Abertos – IOCS: Uma proposta de adaptação dos padrões do Método Clínico Piagetiano em Plataforma Multiagentes. *CINTED-UFRGS - Novas Tecnologias na Educação*, V. 4 N° 1, Julho, 2006.
- Behar, P.; Bernardi, M; Silva, K. Arquiteturas Pedagógicas para a Educação a Distância: a construção e validação de um objeto de aprendizagem. *RENOTE - Revista Novas Tecnologias na Educação*, ISSN 1679-1916, CINTED-UFRGS, V.7. N.1, 2009.
- Bellifemine, F.; Caire, G. *Developing Multi-Agent Systems with JADE.* Ed. Wiley, England, 2007.
- Beni, G. and Wang, J. *Swarm Intelligence in Cellular Robotic Systems*, Proceeding of NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 1989.
- Berners-Lee, T.; Hendler, J.; Lassila, O. *The Semantic Web*, Scientific American, 2001.

- Biolchini, J. et al. Systematic Review in Software Engineering. Relatório Técnico, COPPE/UFRRJ, 2005. Disponível em: www.cin.ufpe.br/~in1037/leitura/systematicReviewSE-COPPE.pdf.
- Blackboard. Educate, Innovate, Everywhere. URL: <http://www.blackboard.com/>, 2017.
- Bloom, B. S.; Hastings, J. T.; Madaus, G. F. Handbook on formative and sommativa evaluation of student learning. New York: McGrawHill, 1971.
- Bratman, M. E. What is intention? MIT Press: Cambridge, 1990.
- Bremgartner, V.; Netto, J.F.M.; Menezes, C.S. . Using Agents and Open Learner Model Ontology for Providing Constructive Adaptive Techniques in Virtual Learning Environments. In: 14th Ibero-American Conference on Artificial Intelligence - IBERAMIA 2014, Santiago, 2014.
- Bremgartner, V.; Netto, J.F.M.; Menezes, C.S. Adaptation Resources in Virtual Learning Environments under Constructivist Approach: A Systematic Review. In: 2015 IEEE Frontiers in Education Conference (FIE) Proceedings, El Paso, USA. 2015.
- Bremgartner, V.; Netto, J.F.M.; Menezes, C.S. Conceptual Framework of Educational Resources Adaptation for Improve Collaborative Learning in Virtual Learning Environments. In: 2017 IEEE Frontiers in Education Conference (FIE) Proceedings, Indianápolis, USA, 2017.
- Bremgartner, V.; Netto, J.F.M.; Menezes, C.S. Explorando Arquiteturas Pedagógicas Recomendadas por meio de Agentes e Ontologia de Modelo do Aluno em Ambientes Virtuais de Aprendizagem. In: Simpósio Brasileiro de Informática na Educação - 26º SBIE, 2015, Maceió, AL. Anais do 26º Simpósio Brasileiro de Informática na Educação, 2015.
- Bremgartner, V. e Netto, J.F.M. An Adaptive Strategy to Help Students in e-Learning Systems Using Competency-Based Ontology and Agents. Proc. IEEE International Conference on Intelligent Systems Design and Applications – ISDA 2011, 2011.
- Bremgartner, V.; Netto, J.F.M. Auxílio Personalizado a Estudantes em Ambientes Virtuais de Aprendizagem Utilizando Agentes e Competências. Anais do XXII Simpósio Brasileiro de Informática na Educação (SBIE), Aracaju, SE, 2011.
- Bremgartner, V.; Netto, J.M. Sistema Inteligente de Recomendação Baseado no Modelo do Aluno. Dissertação de Mestrado. Universidade Federal do Amazonas – UFAM. Manaus, AM, 2012.
- Brusilovsky, P. Adaptive and Intelligent Technologies for Web- based Education. Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, pp 19-25, 1999.
- Bull, S. e Kay, J. SMILI©: a Framework for Interfaces to Learning Data in Open Learner Models, Learning Analytics and Related Fields. International Journal of Artificial Intelligence in Education, v26 n1 p293-331, 2016.
- Bull, S., Quigley, S., Mabbott, A.: Computer-based Formative Assessment to Promote Reflection and Learner Autonomy, Engineering Education, vol. 1, no. 1, pp. 8–18, 2006.
- Burden, R. L. e Faire, J. D. Numerical Analysis. Brooks/Cole, Cengage Learning, 9th Ed., Boston, USA, 2010.
- Campos, M. L. A. *et al.* Ontologias: Representando a Pesquisa na Área Através de Mapa Conceitual. Encontro Nacional de Pesquisa em Ciência da Informação, UFBA, Salvador, 2007.

- Carbonell, J. R. AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems*, 11, 190-202, 1970.
- Carvalho, M. J. S.; Nevado, R. A.; Menezes, C. S. Arquiteturas Pedagógicas para Educação a Distância: Concepções e Suporte Telemático. *Anais - XVI Simpósio Brasileiro de Informática na Educação*, 1, 362-372, 2005.
- Casas, L. A. Contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual. Tese de Doutorado em Engenharia de Produção. Florianópolis, UFSC, 1998.
- Castro, A. N. e Menezes, C. S. Aprendizagem Colaborativa com Suporte Computacional. In: Pimentel, M. e Fuks, H. *Sistemas Colaborativos*. Rio de Janeiro: Campus. ISBN: 978-85-352-4669-8, 2011.
- CC/PP. Composite Capabilities/Preference Profiles: Structure and Vocabularies 2.0. URL: <http://www.w3.org/Mobile/CCPP/>, 2014.
- Chévrier, J., Fortin, G., Leblanc, R. e Théberge, M. La construction du style d'apprentissage. *Éducation et francophonie*, XXVIII, (1), 2000.
- Code. Code.org: Anybody can learn. URL: <https://code.org/>, 2017.
- Costa, L. A. C.; Franco, S. R. K. Ambientes Virtuais de Aprendizagem e suas Possibilidades Construtivistas. *CINTED-UFRGS - Novas Tecnologias na Educação*, V. 3 Nº 1, Maio, 2005.
- Creative Learning Centre. About Learning Styles - Prashnig Style Solutions. URL: <http://www.creativelearningcentre.com/About-Learning-Styles.html>, 2016.
- Crooks, T. The Validity of Formative Assessments. *British Educational Research Association Annual Conference*, University of Leeds, 2001.
- Deffune, D.; Depresbiteris, L. Competências, habilidades e currículos da educação profissional: crônicas e reflexões. São Paulo: SENAC, São Paulo, 2000.
- Díaz, D. P., e Carnal, R. B. Students' learning styles in two classes: Online distance learning and equivalent on-campus. *College Teaching* 47(4), 130-135, 1999.
- Dicionário Aurélio on-line. URL: <http://www.dicionariodoaurelio.com/>, 2017.
- Dillenbourg, P. What do you mean by collaborative learning? In P. Dillenbourg (Ed) *Collaborative-learning: Cognitive and Computational Approaches*. (pp.I-19). Oxford, Elsevier, 1999.
- de la Iglesia, D. G., Calderón, J. F., Weyns, D., Milrad, M., & Nussbaum. M. A Self-Adaptive Multi-Agent System Approach for Collaborative Mobile Learning. In *IEEE Transactions on Learning Technologies*, vol. 8, no. 2, pp. 158-172, 2015.
- Dominoni, M.; Riva, G.; Pinardi, S. Omega Network: an Adaptive Approach to Social Learning. *Intelligent Systems Design and Applications (ISDA)*, 10th International Conference on, 2010.
- Economides, A. A. Context-Aware Mobile Learning. *The Open Knowledge Society. A Computer Science and Information Systems Manifesto. First World Summit. Proceedings Communications in Computer and Information Science*, Springer, Athens, Greece, 2008.
- Edmodo. Conectar-se com alunos e professores na sua sala de aula digital. URL: <https://www.edmodo.com/?language=pt-br>, 2017.

- EDUPERSON. MPDLMediaWiki. URL: <http://colab.mpdl.mpg.de/mediawiki/EduPerson>, 2017.
- Fagundes, M. Um Ambiente para Desenvolvimento de Agentes BDI. Trabalho de Conclusão de Curso. Universidade Federal de Pelotas, RS, 2004.
- Fasihuddin, H.; Skinner, G. e Athauda, R. Towards an Adaptive Model to Personalise Open Learning Environments using Learning Styles. Proceedings of 2014 International Conference on Information, Communication Technology and System (ICTS 2014), Surabaya, Indonesia, 2014.
- Felder, R. M.; Silverman, L. K. Learning and teaching styles in engineering education. *Engineering Education*, 78, pp. 674–681, 1988.
- Ferraz, A. P. C. M. e Belhot R. V. Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gest. Prod.*, São Carlos, v. 17, n. 2, p. 421-431, 2010.
- Few, S. Information dashboard design: The effective visual communication of data. Sebastopol, CA, USA, O'Reilly Media, 2006.
- Fröschl, C. User Modeling and User Profiling in Adaptive E-learning Systems. Master's Thesis, Institute for Information Systems and Computer Media, Graz University of Technology, Graz, Austria, 2005.
- Georgeff, M., Pell, B., Pollack, M., Tambe, M. e Wooldridge, M. The Belief-Desire-Intention Model of Agency. In: *Intelligent Agents V*, LNAI 1555, Springer, pp. 1–10, 1999.
- Giraffa, L.M. M. Uma arquitetura de tutor utilizando estados mentais. Tese de Doutorado. Porto Alegre: CPGCC/UFRGS, 1999.
- Google Computational Thinking Course for Educators. URL: <https://computationalthinkingcourse.withgoogle.com/>, 2017.
- Graf, S.; Kinshuk. Advanced Adaptivity in Learning Management Systems by Considering Learning Styles. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2009.
- Gros, B. The Design of Smart Educational Environments. *Smart Learning Environments*, Springer, Open Access, 3:15, 2016.
- Honey, P. e Mumford, A. *The manual of learning styles*. Maidenhead: Peter Honey, 1992.
- Honey, P. e Mumford, A. *The Learning styles helper's guide*. Maidenhead Berks: Peter Honey Publications, 2000.
- Hosseini, R., Hsiao, I-H., Guerra, J., e Brusilovsky, P. What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling. Proceedings of 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, pp 155-168, 2015.
- Huandong, C.; Shulei, W.; Chunhui, S.; Jinmei, Z.; Juntao, C.; Dong, K. E-Learning System Model Construction Based Constructivism. *Fifth International Joint Conference on INC, IMS and IDC*, 2009.
- IEEE, LTSC. Draft Standard for Learning Technology - Learning Object Metadata. Technical report, IEEE Standards Department, New York, 2002.

- ISTE – International Society for Technology in Education; CSTA Computer Science Teachers Association. Computational Thinking Leadership Toolkit First Edition. URL: <https://www.iste.org/explore/articleDetail?articleid=152>, 2016.
- Jacob, E. K. Ontologies and the Semantic Web. *Bulletin of the American Society for Information Science and Technology*, p. 19-22, 2003.
- JADE. Java Agent DEvelopment Framework, www.jade.tilab.com/, 2017.
- JADEX. JADEX Active Components, www.activecomponents.org/bin/view/About/Features, 2017
- Ji, Z.; Meere, D.; Ganchev, I.; O'Droma, M. Implementation of an Intelligent Framework for Utilization within an InfoStation-based mLearning Environment. 11th IEEE International Conference on Advanced Learning Technologies (ICALT 2011), 2011.
- Kemczinski, A. Método de Avaliação para Ambientes E-Learning. Tese Doutorado em Engenharia da Produção da UFSC – Universidade Federal de Santa Catarina, Florianópolis-SC, 205 p., 2005.
- Khan Academy. Cursos, aulas e prática on-line gratuitos. URL: <https://pt.khanacademy.org/>, 2017.
- Kinshuk. *Designing Adaptive and Personalized Learning Environments (Interdisciplinary Approaches to Educational Technology)*. Routledge, 1st edition, UK, 2016.
- Kitchenham, B. Procedures for Performing Systematic Reviews. Joint Technical Report TR/SE-0401. Software Engineering Group, Department of Computer Science, Keele University, Australia. Disponível em: http://www.idi.ntnu.no/emner/empse/papers/kitchenham_2004.pdf, 2004.
- Labour, M. Learner empowerment via raising awareness of learning styles in foreign language teacher training, In S. J. Armstrong et al.(Eds.), *Learning Styles: Realibility & Validity*, Proceedings of the 7th Annual ELSIN Conference, 227-234. Ghent: Ghent University, Belgium & ELSIN, 2002.
- Latham, A., Crockett, K., McLean, D., Edmonds, B. e O'Shea, K. Oscar: An Intelligent Conversational Agent Tutor to Estimate Learning Styles. Proceedings of 2010 IEEE International Conference on Fuzzy Systems (FUZZ), Barcelona, Spain, 2010.
- LIP. Learner Information Package Specification 1.0.1. URL: <http://www.imsglobal.org/profiles/lipinfo01.html#5.1/>, 2017.
- Liu, W. Using data mining to dynamically build up just in time learner models. Master's Thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, Canada, 2009.
- Liu, D.; Ma, S.; Ru, Q.; Guo, Z.; Ma, S. Design of Multi-strategic Learning Environment based on Constructivism. First International Workshop on Education Technology and Computer Science Design, 2009.
- Luck, M.; d'Inverno, M. A conceptual framework for agent definition and development. *The Computer Journal*, 44(1):1--20, 2001.
- Mafra, S. N.; Travassos, G. H. Estudos Primários e Secundários Apoiando a Busca por Evidência em Engenharia de Software. Relatório Técnico, PESC - COPPP/UFRJ, 2006. Disponível em: <www.cos.ufrj.br/uploadfiles/1149103120.pdf>.

- Marcon, K.; Machado, J. B.; Carvalho, M. J. S. Arquiteturas pedagógicas e redes sociais: uma experiência no Facebook. In: Anais do I Congresso Brasileiro de Informática na Educação: Rio de Janeiro, RJ, 2012.
- Marques, G. R.; Tavares, O. L.; Almeida, R. F. e Morati Junior, R. G. Ambiente para aprendizagem de programação fundamentado em arquiteturas pedagógicas. Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE), Uberlândia, MG, 2016.
- Martins, C. e Carrapatoso, E. Constructivist Approach for an Educational Adaptive Hypermedia Tool. Eighth IEEE International Conference on Advanced Learning Technologies, 2008.
- MEC. Portal do Ministério da Educação. Parâmetros Curriculares Nacionais. URL: portal.mec.gov.br/seb/arquivos/pdf/livro01.pdf, 2017.
- Mendeley. Free reference manager and PDF organizer. www.mendeley.com, 2017.
- Menezes, C. S.; Tavares, O. L.; Nevado, R. A.; Cury, D. Computer Supported Co-Operative Systems to Support the Problem Solving - A Case Study of Learning Computer Programming. 38th ASEE/IEEE Frontiers in Education Conference, 2008.
- Miranda, L e Morais, C. Estilos de Aprendizagem: O Questionário CHAEA Adaptado para Língua Portuguesa. Revista de Estilos de Aprendizagem, nº1, vol 1, abril de 2008.
- Montufar-Chaveznava, R.; Mendez-Polanco, J. A. Multiagent Architecture for Telerobotics. Proceedings of the 15th International Conference on Electronics, Communications and Computers (CONIELECOMP). p. 149 - 153 . Computer Society, IEEE. Puebla, México, 2005.
- Moodle. A Free, Open Source Course Management System for Online Learning. URL: <http://www.moodle.org/>, 2017.
- Musa, D. L.; Oliveira, J. P. M.. OntoLearner: Uma Ontologia para Perfis de Alunos Baseada em Padrões, Anais Do XVIII SBIE – Simpósio Brasileiro de Informática na Educação. São Paulo – SP, 2007.
- Neto, W. C. Web Semântica na Construção de Sistemas de Aprendizagem Adaptativos. Tese de Doutorado, Universidade Federal de Santa Catarina, 2006.
- Netto, J. F. M. Uma Arquitetura para Ambientes Virtuais de Convivência: uma Proposta Baseada em Sistemas Multiagente. Tese de Doutorado, Universidade Federal do Espírito Santo, Vitória, 2006.
- Novak, S.; Aragón, R.; Ziede, M. L. e Menezes, C. S. Aprendizagem em rede na educação a distância: Práticas e reflexões. Evangraf, Porto Alegre, RS, 2014.
- Oliveira, E. A. e Tedesco, P. i-Collaboration in Practice: Results from our Investigation within the Cleverpal Environment. Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), pp. 347-351, 2009.
- Oliveira, E. A. e Tedesco, P. i-collaboration: Um modelo de colaboração inteligente personalizada para ambientes de EAD. Revista Brasileira de Informática na Educação (RBIE), Volume 18, N.1, pp. 17-31, 2010.
- PAPI. IEEE, Draft 8 standard for learning technology. Public and Private Information for Learners. URL: www.edutool.com/papi, 2017.
- Perrenoud, P. Construir as competências desde a escola. Artmed, Porto Alegre, RS, 1999.

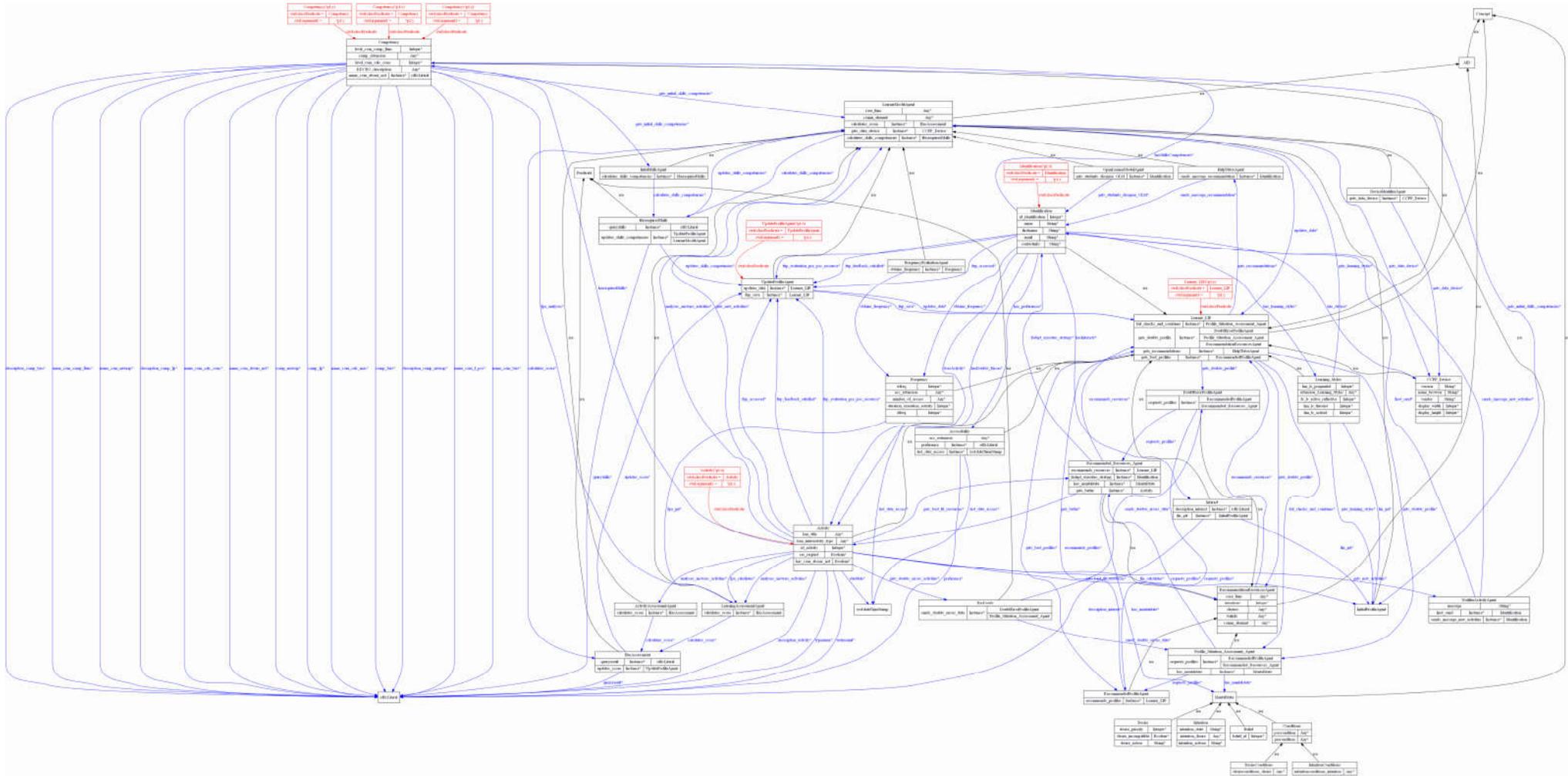
- Perrenoud, P. 10 Novas Competências para Ensinar. Artmed, Porto Alegre, RS, 2000.
- Perrenoud, P; Thurler, M. G. *et al.* As Competências para Ensinar no Século XXI – A Formação dos Professores e o Desafio da Avaliação. Artmed, Porto Alegre, RS, 2002.
- Piaget, J. O desenvolvimento do pensamento: equilíbrio das estruturas cognitivas. Dom Quixote, Lisboa, 1977.
- Protégé. The Ontology Editor, www.protege.stanford.edu/, 2017.
- Ramos, C. L. C. Supervisor educacional de recursos humanos: competências básicas para sua formação e desempenho. Dissertação de Mestrado. Faculdade de Educação da Universidade Federal do Paraná, Curitiba, 1980.
- Ramozzi-Chiarotino, Z. Psicologia e Epistemologia Genética de Jean Piaget. São Paulo, EPU, 1988.
- RDCEO. Reusable Definition of Competency or Educational Objective 1.0. URL: http://www.imslobal.org/competencies/rdceov1p0/imsrdceo_infov1p0.html, 2017.
- Reinoso, L. F. e Tavares, O. L. MVLIBRAS: ambiente digital para comunidades de aprendizagem com recursos inclusivos para surdos. In Anais do XXVI Simpósio Brasileiro de Informática na Educação, Maceió, AL, 2015.
- Resende, E. O livro das competências: desenvolvimento das competências: a melhor autoajuda para pessoas, organizações e sociedade. Qualitymark, Rio de Janeiro, 2000.
- Ribeiro, J. P. A. Reategui, E. Boff, E. Integrando um Agente Pedagógico para Recomendação de Tutores a um Sistema de Gerência de Cursos. In: Revista Novas Tecnologias na Educação, vol. 5, nº 1, 2007.
- RStudio. Open source and enterprise-ready professional software for R. URL: <https://www.rstudio.com/>, 2017
- Russel, S.; Norvig, P. Inteligência Artificial. Editora Campus, São Paulo, 2013.
- Saccol, A. Z.; Kich, M.; Schlemmer, E.; Reinhard, N.; Barbosa, J. L. V.; Hahn, R. A Framework for the Design of Ubiquitous Learning Applications. 42nd Hawaii International Conference on System Sciences, 2009.
- Sadler-Smith, E. The relationship between learning style and cognitive style. Personality and individual differences, vol. 30, n. 4, pp. 609-616, 2001.
- Santana, E. F. Z. et. al. Um Ambiente Baseado na WEB 2.0 para Atividades de Simulação na Educação Médica Construtivista. Anais do XX Simpósio Brasileiro de Informática na Educação, (SBIE 2009) Florianópolis, SC, Brasil, 2009.
- Santos, L. N.; Castro, A. N.; Menezes, C. S. Flexible Virtual Environments for Teaching and Learning. 42nd ASEE/IEEE Frontiers in Education Conference Proceedings (FIE), Seattle, WA, USA, 1388-1393, 2012.
- Sarmiento, C., Duarte, O., Barrera, M., Soto, R. Semi-Automated Academic Tutor for the Selection of Learning Paths in a Curriculum: An Ontology-Based Approach. Proceeding of the 8th International Conference on Engineering Education (ICEED), Selangor, Malaysia, 2016.
- Schreurs, J. e Al-Huneidi, A. Design of Learner-Centered constructivism based Learning Process. Proceedings of the Federated Conference on Computer Science and Information Systems, 2012.

- Scilab. Open source software for numerical computation, www.scilab.org/, 2017.
- Self, J. Theoretical Foundations for Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education*. v. 1 n. 4. p. 3 – 14, 1990.
- Senra, C. M. S.; Lima, G. F. C. A.; Dasilva Fwo. A Relação entre os Estilos de Aprendizagem de Richard Felder e os Tipos Psicológicos de Carl Jung. In: *Anais do Seminário Nacional de Educação Profissional E Tecnológica*, 1, Belo Horizonte, 2008.
- Shaoling, D.; Fangfang, Z. Intelligent Agents Improving the Efficiency of Ubiquitous e-Learning. *Management and Service Science*. MASS '09. International Conference on, 2009.
- Siemens, G., Learning Analytics: Envisioning a Research Discipline and a Domain of Practice, *ACM 2nd International Conference on Learning Analytics and Knowledge*, pp. 4-8, 2012.
- Skinner, B. F. *The Technology of Teaching*. New York: Appleton-Century-Crofts, 1968.
- Sleeman, D. & Brown, J. S. *Intelligent Tutoring Systems*. New York: Academic Press, 1982.
- Spector, J. M.. *Smart Learning Environments: Concepts and Issues*. Proceedings of 27th SITE 2016 - Savannah, USA, 2016.
- Tai, D.; Lai, L.; Hu, Y.; Wang, Y.; Chen, J. The Study of Constructivism Oriented Web-Based Learning on the Performance of Technological College Students. *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on, 2010.
- TelEduc. Ensino à Distância. Núcleo de Informática Aplicada à Educação, Universidade Estadual de Campinas. URL: <http://www.teleduc.org.br/>, 2017.
- TLL. Teaching & Learning Laboratory. Massachusetts Institute of Technology. URL: <https://tll.mit.edu/help/intended-learning-outcomes>, 2014.
- Todorov, J.; Daskalov, B.; Stoyanov, S.; Popchev, I. e Valkanov, V. Learning Intelligent System for Student Assistance – LISSA. Proceedings of the IEEE 8th International Conference on Intelligent Systems, 2016.
- Travassos, G. H. et al. An Environment to Support Large Scale Experimentation in Software Engineering. In: *13th International Conference on Engineering of Complex Computer Systems*, Belfast, 2008.
- ULF. Universal Learning Format URL: xml.coverpages.org/ulfSpecification20001204.pdf>, 2017.
- Valente, S. M. P. Parâmetros Curriculares e Avaliação nas Perspectivas do Estado e da Escola. Tese de Doutorado em Educação. UNESP, São Paulo, 2002.
- Vicari, R. M. e Giraffa, L. M. M. Fundamentos dos Sistemas Tutores Inteligentes. In: Barone, D. (Org). *Sociedades artificiais: a nova fronteira da inteligência das máquinas*. Porto Alegre: Bookman. ISBN: 85-363-0124-4, 2003.
- Vieira Junior, N. Estilos de aprendizagem: é seguro medi-los e utilizá-los?. P@rtes.V.00 p.eletrônica. URL: www.partes.com.br/educacao/estilosdeaprendizagem.asp, 2010.
- Vygotsky, L.S. *Formação social da mente*. Martins Fontes, São Paulo, 1984.
- Weka: Data Mining Software, <http://www.cs.waikato.ac.nz/ml/weka/>, 2017.
- Wenger, E.; McDermott, R. A.; Snyder, W.. *Cultivating communities of practice: A guide to managing knowledge*. Harvard Business Press, 2002.

- Wing, J. Computational Thinking. *Communications of the ACM*, 3 ed.:33-35, 2006.
- Wooldridge, M. *An Introduction to Multiagent Systems*. Ed. Wiley, England, Second Edition, 2009.
- Wu, J.; Houben, G.J.; De Bra, P. User Modeling in Adaptive Hypermedia Applications. *Proceedings of the Interdisciplinary Conferentie Informatiewetenschap*, pp. 10-21, Amsterdam, 1999.
- Zanetti, H. A. P.; Oliveira, C. L. V.; *Prática de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional*. Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE), 2015.
- Zhou, W.; Cui, B.; Wang B.; Shi, Q.; Yokoi, S. An Exploration of Ubiquitous Learning in Computer Fundamental Learning Scenario. *The 7th International Conference on Computer Science & Education (ICCSE 2012)*, 2012.
- Zhu, Z.-T.; Yu, M.-H.; Riezebos, P. A Research Framework of Smart Education. *Smart Learning Environments*. 3(1), 1–17, 2016.

APÊNDICE A – ONTOLOGIA DE MODELO ABERTO DE ALUNO

A figura a seguir mostra a representação gráfica da ontologia de aplicação que representa o MAA em sua forma completa utilizada no AMPARAX.



APÊNDICE B – QUESTIONÁRIO DE PESQUISA DE OPINIÃO I

A seguir, o questionário aplicado para os alunos do curso de Algoritmos e Programação no teste do Sistema AMPARA do Capítulo 5.



Universidade Federal do Amazonas
Instituto de Computação
Pesquisa de Opinião: Debate de Teses no ColabWeb
Turma: TMI11, 2015/01, IFAM CMDI



1. Você conseguiu fazer o exercício postado no Ambiente ColabWeb de Debate de Teses? Quais as maiores dificuldades e facilidades que você teve ao fazer tal exercício?

2. Você avaliou algum trabalho de colega? Se sim, descreva como foi o processo de interação que você teve com o aluno avaliado.

3. O que você achou desta forma de recomendação de atividades em função do seu perfil?

4. Você concorda com seu modelo de aluno (perfil) mostrado no ColabWeb? Por quê?

5. Espaço para comentários, elogios, críticas, vantagens e dificuldades a atividade passada no ColabWeb.

Nome (Opcional):

APÊNDICE C – QUESTÕES DO PRÉ- E PÓS-TESTE

A seguir, as questões que foram utilizadas para o pré- e pós-teste dos alunos no estudo de caso do Minicurso de Pensamento Computacional no Capítulo 6, com o uso do AMPARAX. Estas questões foram retiradas do material oficial da AP CollegeBoard Computer Science Principles – Effective Fall 2016 (2016) e, por isso, estão em inglês.

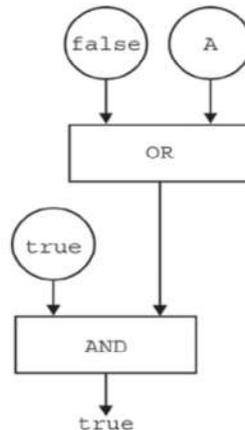
Questões utilizadas no pré-teste:

2. A programmer completes the user manual for a video game she has developed and realizes she has reversed the roles of goats and sheep throughout the text. Consider the programmer's goal of changing all occurrences of "goats" to "sheep" and all occurrences of "sheep" to "goats." The programmer will use the fact that the word "foxes" does not appear anywhere in the original text.

Which of the following algorithms can be used to accomplish the programmer's goal?

- (A) First, change all occurrences of "goats" to "sheep."
Then, change all occurrences of "sheep" to "goats."
- (B) First, change all occurrences of "goats" to "sheep."
Then, change all occurrences of "sheep" to "goats."
Last, change all occurrences of "foxes" to "sheep."
- (C) First, change all occurrences of "goats" to "foxes."
Then, change all occurrences of "sheep" to "goats."
Last, change all occurrences of "foxes" to "sheep."
- (D) First, change all occurrences of "goats" to "foxes."
Then, change all occurrences of "foxes" to "sheep."
Last, change all occurrences of "sheep" to "goats."

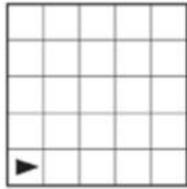
4. The figure below shows a circuit composed of two logic gates. The output of the circuit is `true`.



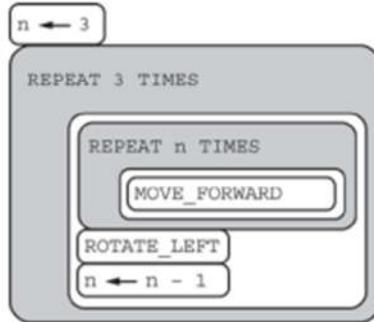
Which of the following is a true statement about input A?

- (A) Input A must be `true`.
- (B) Input A must be `false`.
- (C) Input A can be either `true` or `false`.
- (D) There is no possible value of input A that will cause the circuit to have the output `true`.

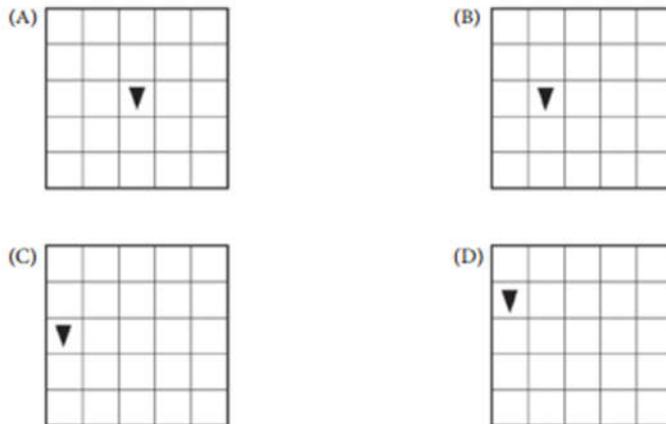
5. The following question uses a robot in a grid of squares. The robot is represented as a triangle, which is initially in the bottom left square of the grid and facing right.



Consider the following code segment, which moves the robot in the grid.



Which of the following shows the location of the robot after running the code segment?



7. A certain social media Web site allows users to post messages and to comment on other messages that have been posted. When a user posts a message, the message itself is considered data. In addition to the data, the site stores the following metadata.
- The time the message was posted
 - The name of the user who posted the message
 - The names of any users who comment on the message and the times the comments were made

For which of the following goals would it be more useful to analyze the data instead of the metadata?

- (A) To determine the users who post messages most frequently
- (B) To determine the time of day that the site is most active
- (C) To determine the topics that many users are posting about
- (D) To determine which posts from a particular user have received the greatest number of comments

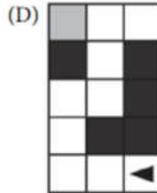
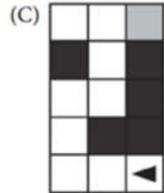
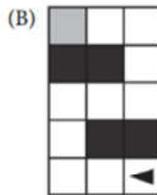
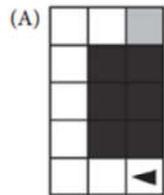
8. The program segment below is intended to move a robot in a grid to a gray square. The program segment uses the procedure `GoalReached`, which evaluates to `true` if the robot is in the gray square and evaluates to `false` otherwise. The robot in each grid is represented as a triangle and is initially facing left. The robot can move into a white or gray square but cannot move into a black region.

```

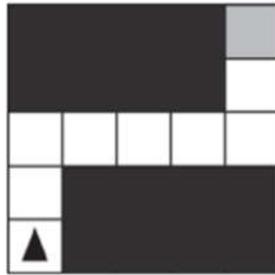
REPEAT UNTIL (GoalReached ())
{
  IF (CAN_MOVE (forward))
  {
    MOVE_FORWARD ()
  }
  IF (CAN_MOVE (right))
  {
    ROTATE_RIGHT ()
  }
  IF (CAN_MOVE (left))
  {
    ROTATE_LEFT ()
  }
}

```

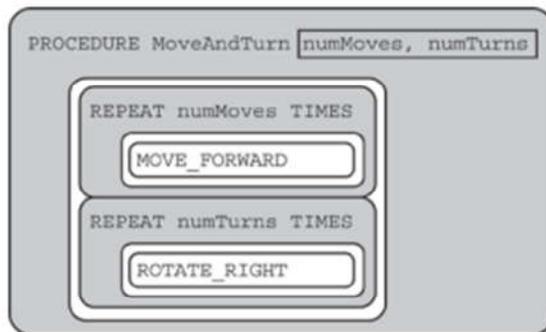
For which of the following grids does the program NOT correctly move the robot to the gray square?



14. The figure below shows a robot in a grid of squares. The robot is represented as a triangle, which is initially facing upward. The robot can move into a white or gray square but cannot move into a black region.



Consider the procedure `MoveAndTurn` below.



Which of the following code segments will move the robot to the gray square?

- (A) `MoveAndTurn 1, 2`
`MoveAndTurn 3, 4`
`MoveAndTurn 0, 2`
- (B) `MoveAndTurn 2, 1`
`MoveAndTurn 4, 1`
`MoveAndTurn 2, 0`
- (C) `MoveAndTurn 2, 1`
`MoveAndTurn 4, 3`
`MoveAndTurn 2, 0`
- (D) `MoveAndTurn 3, 1`
`MoveAndTurn 5, 3`
`MoveAndTurn 3, 0`

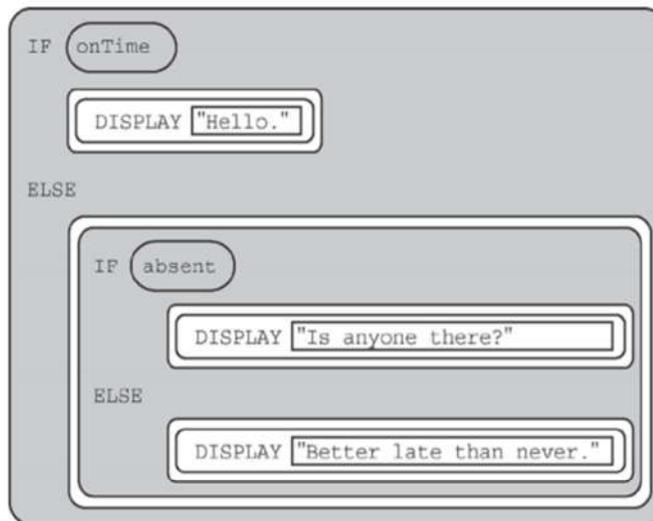
21. Which of the following algorithms require both selection and iteration?

Select two answers.

- (A) An algorithm that, given two integers, displays the greater of the two integers
- (B) An algorithm that, given a list of integers, displays the number of even integers in the list
- (C) An algorithm that, given a list of integers, displays only the negative integers in the list
- (D) An algorithm that, given a list of integers, displays the sum of the integers in the list

Questões utilizadas no pós-teste:

10. Consider the code segment below.



If the variables `onTime` and `absent` both have the value `false`, what is displayed as a result of running the code segment?

- (A) Is anyone there?
- (B) Better late than never.
- (C) Hello. Is anyone there?
- (D) Hello. Better late than never.

13. There are 32 students standing in a classroom. Two different algorithms are given for finding the average height of the students.

Algorithm A

- Step 1: All students stand.
- Step 2: A randomly selected student writes his or her height on a card and is seated.
- Step 3: A randomly selected standing student adds his or her height to the value on the card, records the new value on the card, and is seated. The previous value on the card is erased.
- Step 4: Repeat step 3 until no students remain standing.
- Step 5: The sum on the card is divided by 32. The result is given to the teacher.

Algorithm B

- Step 1: All students stand.
- Step 2: Each student is given a card. Each student writes his or her height on the card.
- Step 3: Standing students form random pairs at the same time. Each pair adds the numbers written on their cards and writes the result on one student's card; the other student is seated. The previous value on the card is erased.
- Step 4: Repeat step 3 until one student remains standing.
- Step 5: The sum on the last student's card is divided by 32. The result is given to the teacher.

Which of the following statements is true?

- (A) Algorithm A always calculates the correct average, but Algorithm B does not.
- (B) Algorithm B always calculates the correct average, but Algorithm A does not.
- (C) Both Algorithm A and Algorithm B always calculate the correct average.
- (D) Neither Algorithm A nor Algorithm B calculates the correct average.

15. Biologists often attach tracking collars to wild animals. For each animal, the following geolocation data is collected at frequent intervals.
- The time
 - The date
 - The location of the animal

Which of the following questions about a particular animal could NOT be answered using only the data collected from the tracking collars?

- (A) Approximately how many miles did the animal travel in one week?
 - (B) Does the animal travel in groups with other tracked animals?
 - (C) Do the movement patterns of the animal vary according to the weather?
 - (D) In what geographic locations does the animal typically travel?
16. A summer camp offers a morning session and an afternoon session. The list `morningList` contains the names of all children attending the morning session, and the list `afternoonList` contains the names of all children attending the afternoon session.

Only children who attend both sessions eat lunch at the camp. The camp director wants to create `lunchList`, which will contain the names of children attending both sessions.

The following code segment is intended to create `lunchList`, which is initially empty. It uses the procedure `IsFound (list, name)`, which returns `true` if `name` is found in `list` and returns `false` otherwise.

```
FOR EACH child IN morningList
{
  <MISSING CODE>
}
```

Which of the following could replace `<MISSING CODE>` so that the code segment works as intended?

- (A)

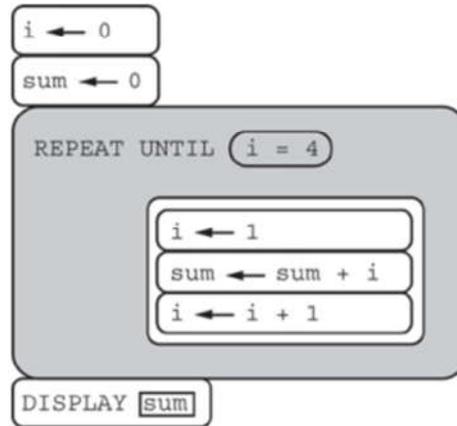
```
IF (IsFound (afternoonList, child))
{
  APPEND (lunchList, child)
}
```
- (B)

```
IF (IsFound (lunchList, child))
{
  APPEND (afternoonList, child)
}
```
- (C)

```
IF (IsFound (morningList, child))
{
  APPEND (lunchList, child)
}
```
- (D)

```
IF ((IsFound (morningList, child)) OR
(IsFound (afternoonList, child)))
{
  APPEND (lunchList, child)
}
```

17. Consider the following program code.



Which of the following best describes the result of running the program code?

- (A) The number 0 is displayed.
 - (B) The number 6 is displayed.
 - (C) The number 10 is displayed.
 - (D) Nothing is displayed; the program results in an infinite loop.
19. An office building has two floors. A computer program is used to control an elevator that travels between the two floors. Physical sensors are used to set the following Boolean variables.

Variable	Description
onFloor1	set to true if the elevator is stopped on floor 1; otherwise set to false
onFloor2	set to true if the elevator is stopped on floor 2; otherwise set to false
callTo1	set to true if the elevator is called to floor 1; otherwise set to false
callTo2	set to true if the elevator is called to floor 2; otherwise set to false

The elevator moves when the door is closed and the elevator is called to the floor that it is not currently on. Which of the following Boolean expressions can be used in a selection statement to cause the elevator to move?

- (A) (onFloor1 AND callTo2) AND (onFloor2 AND callTo1)
- (B) (onFloor1 AND callTo2) OR (onFloor2 AND callTo1)
- (C) (onFloor1 OR callTo2) AND (onFloor2 OR callTo1)
- (D) (onFloor1 OR callTo2) OR (onFloor2 OR callTo1)

22. A teacher uses the following program to adjust student grades on an assignment by adding 5 points to each student's original grade. However, if adding 5 points to a student's original grade causes the grade to exceed 100 points, the student will receive the maximum possible score of 100 points. The students' original grades are stored in the list `gradeList`, which is indexed from 1 to `n`.

```

i ← 1
REPEAT n TIMES
{
  <MISSING CODE>
  i ← i + 1
}

```

The teacher has the following procedures available.

Procedure	Explanation
<code>min (a, b)</code>	Returns the lesser of the two values <code>a</code> and <code>b</code>
<code>max (a, b)</code>	Returns the greater of the two values <code>a</code> and <code>b</code>

Which of the following code segments can replace `<MISSING CODE>` so that the program works as intended?

Select two answers.

- (A) `gradeList[i] ← min (gradeList[i] + 5, 100)`
- (B) `gradeList[i] ← max (gradeList[i] + 5, 100)`
- (C) `gradeList[i] ← gradeList[i] + 5`
`IF (gradeList[i] > 100)`
`{`
`gradeList[i] ← gradeList[i] - 5`
`}`
- (D) `gradeList[i] ← gradeList[i] + 5`
`IF (gradeList[i] > 100)`
`{`
`gradeList[i] ← 100`
`}`

Respostas:

1 - C	12 - A
2 - C	13 - C
3 - C	14 - C
4 - A	15 - C
5 - A	16 - A
6 - C	17 - D
7 - C	18 - D
8 - D	19 - B
9 - D	20 - A
10 - B	21 - B, C
11 - C	22 - A, D

APÊNDICE E – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Prezado(a).

Somos uma equipe de pesquisa do Instituto de Computação (IComp) da Universidade Federal do Amazonas (UFAM), desenvolvendo técnicas de ensino (com adaptações de acordo com o perfil do usuário) que têm por objetivo motivar o aprendizado dos alunos em pensamento computacional com ênfase em programação de computadores e minimizar os índices de reprovação em disciplinas introdutórias de programação (para alunos). Nesta etapa do projeto, queremos fazer um balanço do conhecimento dos estudantes para verificar relações com as técnicas de ensino.

Por isso, solicitamos seu consentimento para a realização de testes e avaliações. Para decidir sobre o seu consentimento, é importante que você conheça as seguintes informações sobre a pesquisa:

- Os dados coletados durante a atividade destinam-se estritamente às atividades de análise e desenvolvimento de técnicas de ensino de programação.
- Temos o compromisso de divulgar os resultados de nossas pesquisas para os participantes do projeto. A divulgação desses resultados pauta-se no respeito à sua privacidade. O anonimato dos participantes será preservado em quaisquer documentos que elaborarmos.
- O consentimento para participar da atividade é uma escolha livre, feita mediante prestação de todos os conhecimentos necessários para a realização desta pesquisa.
- A atividade pode ser interrompida a qualquer momento, segundo a sua disponibilidade e vontade.
- Estamos disponíveis para contato através do e-mail: vitorbref@gmail.com

De posse dessas informações, gostaríamos que você se pronunciasse acerca da atividade:

- Dou meu consentimento para a sua realização
- Não consinto com a sua realização

APÊNDICE F – REVISÃO SISTEMÁTICA DE LITERATURA

Após a revisão inicial da literatura, uma Revisão Sistemática foi planejada e conduzida para identificar as principais técnicas de adaptação de conteúdo em AVAs e as informações mais relevantes obtidas estão descritas na seção F.1. Estas informações foram úteis para embasar o projeto do ArCARE, uma vez que o mesmo é baseado em adaptação de recursos educacionais.

F.1 Revisão Sistemática sobre adaptação em AVAs baseada na teoria construtivista

A Revisão Sistemática requer um esforço considerável quando comparado com uma revisão de literatura informal. Enquanto que a revisão de literatura informal é conduzida de forma ad-hoc, sem planejamento e critérios de seleção estabelecidos a priori, a revisão sistemática segue um protocolo formal para conduzir uma pesquisa sobre um determinado tema, com uma sequência bem definida de passos metodológicos (Mafra e Travassos, 2006). O processo para a condução de revisões sistemáticas envolve três etapas (Mafra e Travassos, 2006):

1. **Planejamento da Revisão:** os objetivos da pesquisa são listados e o protocolo da revisão é definido;
2. **Condução da Revisão:** nesta atividade, as fontes para a revisão sistemática são selecionadas, os estudos primários são identificados, selecionados e avaliados de acordo com os critérios de inclusão, exclusão, e de qualidade estabelecidos durante o protocolo da revisão;
3. **Análise e Publicação dos Resultados:** os dados dos estudos são extraídos e sintetizados para serem publicados.

Conduzimos a revisão sistemática deste trabalho, baseado nas três etapas citadas anteriormente, e de acordo com as diretrizes definidas por Biolchini et al. (2005), Mafra e Travassos (2006) e Kitchenham (2004). Entretanto, como o objetivo deste trabalho é realizar um estudo exploratório de caracterização de área e, portanto, não envolve um estudo de comparação, podemos dizer que esta revisão sistemática se caracteriza como uma *quasi*-sistemática (Travassos et al., 2008), pois segue o mesmo processo da revisão sistemática e preserva o mesmo formalismo.

A revisão sistemática teve por finalidade identificar e conhecer as principais técnicas de adaptação de recursos de aprendizagem em AVAs presentes na literatura, focando na teoria pedagógica do Construtivismo.

F.1.1 Planejamento da Revisão Sistemática

Objetivo

O objetivo deste estudo foi esquematizado a partir da estrutura proposta pelo paradigma GQM (*goal, question, and metric*) (Basili et al., 1994), mostrado na Tabela 18.

Tabela 18: Objetivo da Revisão Sistemática.

Analisar, identificar, avaliar	Trabalhos que apresentam adaptação de recursos em AVAs
Com o propósito de	Obter evidências (estado da arte dos trabalhos desta área de pesquisa)
Com relação ao	Uso dessas técnicas de adaptação de recursos em AVAs
Do ponto de vista de	Sistemas Inteligentes e/ou Adaptativos Aplicados à Educação, utilizando uma abordagem construtivista
No contexto de	Ambientes Virtuais de Aprendizagem, Construtivismo, Personalização, Inteligência Artificial e Educação.

Fonte: o autor.

Formulação das Perguntas

Na Revisão Sistemática realizada, buscamos respostas para as três perguntas a seguir:

- **Q1:** Quais técnicas de adaptação apresentadas a estudantes em AVAs estão sendo correntemente usadas?
- **Q2:** A partir da resposta de Q1, quais técnicas de adaptação apresentadas a estudantes usam uma abordagem construtivista?
- **Q3:** Que novas ideias poderiam ser criadas pelos desenvolvedores de sistemas educacionais objetivando usar adaptação de recursos em AVAs de uma forma construtivista?

É importante ressaltar que estas questões dizem respeito a Revisão Sistemática e são mais genéricas do que as questões de pesquisa próprias desse trabalho descritas no Capítulo 1.

Crítérios de Inclusão de Publicações

- As publicações devem estar disponíveis na *web*;
- As publicações devem estar descritas em inglês ou português;
- As publicações devem contemplar temas que tenham relação com adaptação de recursos e/ou práticas construtivistas em AVAs.

Listagem de Fontes

- 1) As bibliotecas digitais que foram consultadas estão listadas abaixo:
 - Compendex: <<http://www.engineeringvillage2.org>>
 - IEEE Computer Science Digital Library: <<http://ieeexplore.ieee.org>>
 - Science Direct - Elsevier: <<http://www.sciencedirect.com/>>
 - Scopus: <<http://www.scopus.com>>
 - ACM Digital Library: <<http://portal.acm.org/dl.cfm>>
 - Anais do Simpósio Brasileiro de Informática na Educação (SBIE).
- 2) Livros e páginas Web que permitam buscas manuais relacionadas a adaptação de recursos e práticas construtivistas em AVAs.
- 3) Livro que aborda o assunto sobre Construtivismo de Piaget que foi utilizado no período de realização desta revisão sistemática. Foi utilizado o seguinte livro:
 - (Ramoszi-Chiarotino, 1988) Ramoszi-Chiarotino, Zélia. Psicologia e Epistemologia Genética de Jean Piaget. São Paulo, EPU, 1988.

Strings de Busca

As strings de busca foram utilizadas nas bibliotecas digitais, exceto para o SBIE, por esta fonte não permitir pesquisas utilizando strings de busca. Neste caso, foi feita uma busca manual nos anais deste evento. Para as demais bibliotecas digitais, os principais termos utilizados nas strings de busca foram: “profile”, “profiling”, “profilling”, “personalization”, “personalisa-

tion”, “adaptation”, “adaptive”, “adaptivity”, “context-awareness”, “VLE”, “virtual learning environments”, “constructivism”, “constructivist”, “CSCL”, “computer-supported-collaborative-learning”, “collaborative learning”.

F.1.2 Condução da Revisão Sistemática

As publicações recuperadas pelas máquinas de busca foram organizadas pelo gerenciador de referências bibliográficas Mendeley (2017). O Mendeley permitiu a indexação dos itens, ou seja, criou uma lista com os nomes e outras informações para pesquisas instantâneas. Ele ainda possui um rastreador automático de referências internas nos documentos, campos de pesquisa e filtros detalhados, e possibilita a anotação participativa e identificação de repetições.

No período da execução da Revisão Sistemática houve ajustes no protocolo de revisão, pois antes não era considerada a teoria construtivista da aprendizagem, que depois foi acrescentada. Além disso, termos relacionados a práticas construtivistas, como CSCL, e perfil do aluno foram adicionados. As publicações foram selecionadas de acordo com os critérios de inclusão estabelecidos no protocolo da revisão. As referências das publicações que estavam inacessíveis pela Web foram excluídas, assim como as publicações repetidas que foram acessadas por máquinas de busca diferentes. Além disso, foram excluídas as publicações que nitidamente tratavam de outros assuntos não pertinentes à pesquisa.

Estratégias de Extração de Informação

Em uma primeira avaliação, ainda na página da máquina de busca, foi realizado o primeiro estudo, onde foram considerados os itens:

- Título
- Resumo
- Palavras-Chave

Os artigos selecionados para a segunda análise tiveram os seguintes dados extraídos:

- [E1] Título: o título do trabalho;
- [E2] Autor(es): o(s) autor(es) do trabalho;
- [E3] Palavras-Chave: palavras-chaves do trabalho;
- [E4] Base/Fonte: em que base o artigo foi encontrado e onde o trabalho foi publicado (journal, proceedings, etc...);

- [E5] Ano de publicação: ano de publicação do trabalho;
- [E6] Tipo de estudo: por exemplo, estudo de caso, experimento, relatório de experiência, pesquisa de opinião, dentre outros;
- [E7] Resumo: uma breve descrição do estudo;
- [E8] Resultados: o grau de eficiência/eficácia da solução ou técnica de adaptação apresentada em cada trabalho, e ainda, se a técnica obteve sucesso ou não, se estiverem estas informações presentes no artigo;
- [E9] Técnica de adaptação utilizada: Como se deu o processo de adaptação de recurso no ambiente utilizado;
- [E10] Perfis: como é descrito o perfil do aluno no trabalho;
- [E11] Tecnologias: Tecnologias empregadas (ontologia, agentes, etc.) e os softwares utilizados para cada tecnologia empregada;
- [E12] Local/AVA: Onde foram realizados os experimentos, sob quais circunstâncias e que AVA foi utilizado, e o equipamento, se explícito;
- [E13] Recomendação para melhoria: sugestões propostas para melhoria da técnica de adaptação, se houver alguma;
- [E14] Citação de outros trabalhos relacionados ao tema: relação de autores que realizaram trabalhos relacionados com o tema desta pesquisa;
- [E15] Perspectivas futuras: recomendações para trabalhos futuros, se houver.
- [E16] Comentários: observações gerais sobre o estudo, se necessário.

F.1.3 Análise e Publicação dos resultados

A Tabela 19 apresenta a consolidação da quantidade de artigos encontrados pelas máquinas de buscas.

Tabela 19: Resultados gerais da quantidade de publicações encontradas.

Base	Nº Total de Publicações	Publicações Selecionadas Após a Primeira Análise ^(a)	Publicações Selecionadas Após a Segunda Análise
Compendex	3	2	1
IEEEExplore	459	142	65
Science Direct (Else-	74	23	5

vier)			
Scopus	141	13	2
ACM	58	2	2
Anais SBIE	183	8	4
Buscas manuais	33	29	29
TOTAL	951	219	108

(a) Considerar que nas bases Scopus, ACM, Compendex haviam artigos repetidos. Por isso, houve uma larga diferença entre IEEE e estas outras bases quanto ao número de artigos encontrados.

Fonte: o autor.

De acordo com a Tabela 19, nas bibliotecas digitais procuradas, 951 publicações foram encontradas no total. Dessas, apenas 108 foram selecionadas, após a segunda análise, correspondendo a 11, 35% do total de publicações encontradas. Para chegar a esse número de publicações selecionadas, foram usados os critérios estabelecidos na Subseção F.1.1.

A Tabela 20 resume todas as técnicas de adaptação encontradas nas publicações selecionadas, ordenadas pelo número de ocorrências de técnicas de adaptação encontradas, independente de usar ou não práticas construtivistas. Assim, a técnica de sistemas sensíveis ao contexto foi a com maior número de ocorrências (51), seguida de adaptação por agentes (35), aprendizagem colaborativa (33), e assim por diante. Na Tabela 20 é também possível ver uma descrição mais detalhada de algumas técnicas de adaptação para uma melhor compreensão. É importante ressaltar que em uma publicação pode ter ocorrido mais de uma técnica de adaptação. Portanto, a soma de todas as ocorrências de técnicas de adaptação é maior do que o número de todas as publicações selecionadas.

Tabela 20: Técnicas de adaptação encontradas nas publicações selecionadas.

Posição	Técnica de Adaptação	Número de Ocorrências	Descrição Adicional
1	Sensibilidade ao contexto (Context-awareness)	51	Definida na Seção 1.1
2	Adaptação por agentes	35	Usando agentes de software. Agentes são definidos na Seção 2.2
3	Aprendizagem colaborativa	33	Definida no Capítulo 2
4	Estilos de aprendizagem	19	Definidos na Seção 2.7

5	Lógica difusa (fuzzy)	10	
6	Interesses dos estudantes	10	
7	Planejamento	8	O planejamento é "a tarefa de começar uma sequência de ações que irá alcançar um objetivo" (Russel e Norvig, 2013)
8	Adaptação baseada em competências	8	Relacionada às competências que um estudante possui
9	Modelo do aluno	7	Definido na Subseção 2.6.1
10	Ontologias	7	
11	Recomendação de conteúdo	7	
12	Padrões	7	Padrões foram usados permitindo adaptação de recursos em AVAs. Os mais citados foram o IMS LIP para modelo de aluno e o CC/PP para dispositivos móveis
13	Instrução programada	3	Consiste de um processo de auto-aprendizagem que apresenta material estruturado em uma seqüência lógica e empiricamente desenvolvida (Skinner, 1968)
14	Tecnologia de enxames (swarm intelligence)	2	Consiste no "comportamento coletivo de sistemas auto-organizados descentralizados, naturais ou artificiais" (Beni e Wang, 1989)

15	Aprendizagem significativa	2	Refere-se a "uma forma de aprendizagem, onde o novo conhecimento para ser obtido está relacionado com o conhecimento prévio" (Ausubel, 2000)
16	Métodos estatísticos	1	
17	Objetivos de aprendizagem	1	Objetivos de aprendizagem "devem descrever o que os alunos devem saber ou serem capazes de fazer no final do curso tais que eles não podiam fazer antes" (TLL, 2014)
18	Adaptação por templates	1	Usando templates predefinidos para adaptação
19	Cadeias de Markov	1	
20	Modelo de algoritmo usando fórmulas matemáticas	1	
21	Avaliação formativa	1	Este processo é "uma série de procedimentos formais e informais de avaliação utilizados pelos professores durante o processo de aprendizagem, a fim de modificar as atividades de ensino e aprendizagem para melhorar a desempenho do estudante" (Crooks, 2001)
22	Algoritmos genéticos	1	
23	Transferência de conhecimento	1	A transferência de conhecimento é "o processo através do qual uma unidade (por

			exemplo, grupo, departamento ou divisão) é afetada pela experiência de outra" (Argote e Ingram, 2000)
24	Mensagens de feedback	1	Mensagens de feedback fornecidas a estudantes de uma forma construtiva, estimulando os estudantes alcançarem a resposta correta

Fonte: o autor.

A Figura 66 mostra um gráfico com os percentuais de trabalhos utilizando abordagem construtivista dos 108 trabalhos selecionados. 59 (55%) trabalhos selecionados não usam práticas construtivistas, enquanto que 49 (45%) usam.

Figura 66. Percentuais de trabalhos que usam e não usam uma abordagem construtivista.



Fonte: o autor.

Um fato observado neste estudo foi encontrar 58 publicações (53,7% do total) que utilizam m-learning das 108 publicações selecionadas. Além disso, para a técnica de adaptação que utiliza padrões, que tem 7 ocorrências na Tabela 19, os padrões mais citados foram o IMS LIP para as características dos alunos e Composite Capabilities/Preference Profile (CC/PP) (CC/PP, 2017) para caracterizar o dispositivo móvel que o aluno usa. É importante salientar, em relação à técnica de adaptação por estilos de aprendizagem (19 ocorrências na Tabela 19), os estilos que apareceram com maior frequência foram os de Felder-Silverman (1988). Para adaptação por meio de agentes, a ferramenta mais utilizada foi o Java Agent Development Framework (JADE) (JADE, 2017).

Além disso, a Tabela 21 mostra apenas as técnicas de adaptação em que a abordagem construtivista foi usada de todas as 108 publicações selecionadas ordenadas por número de ocorrências. Nesta situação, "aprendizagem colaborativa" foi a técnica com o maior número de ocorrências encontradas (25), seguida de adaptação por agentes (17), sensibilidade ao contexto, estilos de aprendizagem e a lógica difusa (9 ocorrências cada), e assim por diante.

Tabela 21: Técnicas de adaptação encontradas em todas as publicações selecionadas que usam uma abordagem construtivista.

Posição	Técnica de Adaptação	Número de Ocorrências
1	Aprendizagem colaborativa	25
2	Adaptação por agentes	17
3	Sensibilidade ao contexto (Context-awareness)	9
4	Estilos de aprendizagem	9
5	Lógica difusa (fuzzy)	9
6	Interesses dos estudantes	8
7	Ontologias	4
8	Modelo do aluno	3
9	Recomendação de conteúdo	2
10	Planejamento	2
11	Adaptação baseada em competências	2
12	Métodos estatísticos	1
13	Objetivos de aprendizagem	1
14	Cadeias de Markov	1
15	Avaliação formativa	1
16	Mensagens de feedback	1

Fonte: o autor.

Voltando às perguntas da Subseção F.1.1, baseado nas informações obtidas a partir dos resultados da Revisão Sistemática, as respostas encontradas serão descritas a seguir.

- **Q1:** Quais técnicas de adaptação apresentadas a estudantes em AVAs estão sendo correntemente usadas?
- **R1:** A Tabela 20 mostra todas as técnicas de adaptação em AVAs encontradas nesta pesquisa e o número de suas ocorrências. De acordo com a Tabela 20, as 3 técnicas de

adaptação encontradas com o maior número de ocorrências foram: sensibilidade ao contexto (context-awareness), a adaptação por agentes, e a aprendizagem colaborativa, respectivamente com 51, 35 e 33 ocorrências cada. Assim, podemos notar que a sensibilidade ao contexto é a técnica de adaptação atualmente mais comum na literatura pesquisada. Nos trabalhos lidos, a sensibilidade do contexto poderia ser alcançada por meio de recursos nos equipamentos dos alunos (por exemplo, GPS, RFID, redes de sensores sem fio) (Ji et al., 2011) e as características dos alunos (preferências de navegação, respostas das atividades, a frequência dos alunos em AVAs e seu modelo de aluno) (Graf e Kinshuk, 2009). É importante ressaltar que em alguns trabalhos a técnica de sensibilidade ao contexto foi utilizada junto com outras técnicas vistas na Tabela 20, como a adaptação por agentes, modelo de aluno e padrões.

- **Q2:** A partir da resposta de Q1, quais técnicas de adaptação apresentadas a estudantes usam uma abordagem construtivista?
- **R2:** De todas as técnicas de adaptação encontradas na Tabela 20, a Tabela 21 mostra apenas as técnicas de adaptação em AVAs usando uma abordagem construtivista e o número de suas ocorrências. E de todas as 24 técnicas de adaptação descritas na Tabela 20, 16 delas têm sido utilizadas em ambientes construtivistas de acordo com esta pesquisa. Além disso, de todas as 218 ocorrências pelas soma delas na Tabela 20, 95 (43,5%) utilizaram técnicas de adaptação de AVAs com uma abordagem construtivista. Neste caso, a aprendizagem colaborativa teve o maior número de ocorrências (25), seguida de adaptação por agentes (17), e a sensibilidade ao contexto (9). É importante ressaltar que embora a aprendizagem colaborativa muitas vezes é presente em abordagens construtivistas, nem todos os estudos que utilizam o construtivismo tratam com a aprendizagem colaborativa e vice-versa. Isso pode ser visto nas Tabelas 20 e 21. Por exemplo, apesar de haver 33 ocorrências encontradas para aprendizagem colaborativa, 8 delas não utilizam práticas construtivistas.
- **Q3:** Que novas ideias poderiam ser criadas pelos desenvolvedores de sistemas educacionais objetivando usar adaptação de recursos em AVAs de uma forma construtivista?
- **R3:** Com base em alguns estudos com uma maior ênfase nas tecnologias em vez dos aspectos pedagógicos, a primeira sugestão é verificar a efetividade da abordagem pe-

dagógica, não só as tecnologias utilizadas. Outra sugestão para os desenvolvedores de sistemas educacionais é considerar a utilização das 8 técnicas de adaptação encontradas neste estudo que não usam práticas construtivistas, exceto a instrução programada (padrões, tecnologias de inteligência de enxame; aprendizagem significativa, adaptação por templates; modelo de algoritmo utilizando fórmulas matemáticas; algoritmos genéticos; e transferência de conhecimento). Ou, pode-se utilizar algumas técnicas de adaptação com base em teoria construtivista e também as mesmas técnicas baseadas na instrução programada, para comparar qual das duas abordagens dariam resultados mais satisfatórios. Outra sugestão é integrar várias técnicas que já utilizam práticas construtivistas, permitindo AVAs mais adaptáveis às necessidades dos alunos. Por exemplo, o AVA pode proporcionar atividades que promovam o raciocínio do aluno e feedbacks construtivos para os alunos de acordo com os estilos de aprendizagem, níveis de habilidade dos estudantes, respostas de atividades e interesses. Assim, através de feedback, que permite o diálogo com o aluno, ele não é apenas um receptor de informações. Este diálogo também pode ser mediado por agentes de software. Outra sugestão é o desenvolvimento de aplicações e estratégias construtivistas para ambientes de m-learning, uma vez que muitos trabalhos (53,7% das 108 publicações selecionadas) utilizando m-learning foram detectados nesta Revisão Sistemática. Por exemplo, as técnicas de adaptação poderiam estimular a aprendizagem colaborativa entre os alunos e adaptar o conteúdo apresentado aos alunos dependendo das características dos dispositivos móveis, tais como o tamanho da tela, capacidade de processamento, memória, entre outros, mas de uma forma construtivista. Além disso, os desenvolvedores podem projetar estratégias adaptativas construtivistas, tais como passar questionários para conhecer o estilo de aprendizagem do aluno e em função do resultado deste questionário, recomendar atividades que estimulem a construção do conhecimento do aluno, dependendo também dos resultados de avaliações de atividades do aluno, e promover a cooperação e colaboração entre os alunos em AVAs através de agentes e tecnologias móveis. Outra ideia é usar o padrão de dispositivos móveis CC/PP para ser estendido com componentes do padrão IMS LIP, integrando o perfil do dispositivo e o modelo de aluno.

Com base nestes resultados obtidos pela Revisão Sistemática e nas respostas das questões de pesquisa acima, optamos por desenvolver o ArCARE, que é baseado em agentes e ontologias que promovam a aprendizagem colaborativa entre estudantes.

Além disso, selecionamos os trabalhos que consideramos os melhores relacionados com esta pesquisa, apresentados no Capítulo 3.

APÊNDICE G – GLOSSÁRIO

Adaptação de recursos: Ajustes de recursos em um AVA ou adaptações em um AVA por meio da recomendação de recursos educacionais.

Adaptação forçada: Quando ocorre uma adaptação em um AVA sem que o aluno queira.

Aprendizagem Baseada em Problemas: É um método de aprendizagem em que os estudantes se deparam inicialmente com um problema, o qual é sucedido por uma investigação em um processo de aprendizagem centrada no estudante.

Arquiteturas Pedagógicas Móveis: Arquiteturas Pedagógicas adaptadas ou ajustadas exclusivamente para uso em dispositivos móveis.

Catálogo de Recursos: Repositório de recursos criados e disponíveis a serem recomendados ou adaptados em AVAs.

Comunidade de prática: Definido por Wenger et. al. (2002) como um grupo de pessoas que compartilham uma preocupação, um conjunto de problemas ou uma paixão sobre um tema e que aprofundam seus conhecimentos e experiências nesta área, interagindo em uma base contínua.

Histórico de Utilização de Recursos: Informações relativas a todo o processo de recomendação, adaptação e utilização dos recursos educacionais em AVAs pelos alunos.

Mobile learning (m-learning): emprego da tecnologia móvel para fins educacionais.

Modelo do recurso: os dados mais relevantes para o processo de adaptação e recomendação de recursos em AVAs, como por exemplo, os níveis de habilidades e competências dos estudantes e os níveis relacionados aos recursos.

Recomendação de recursos: dispor recursos de forma personalizada para cada usuário de acordo com seu perfil.

Unidades de Ensino: cada item de um curso ou disciplina disposto em um AVA, como tópicos, conteúdos, atividades ou outros recursos educacionais.