

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



RODRIGO VALENÇA CAVALCANTE FRADE

UNIVIRTUAL - AMBIENTE VIRTUAL 3D MULTIAGENTE COM RECOMENDAÇÃO PERSONALIZADA DE OBJETOS DE APRENDIZAGEM

RODRIGO VALENÇA CAVALCANTE FRADE

UNIVIRTUAL - AMBIENTE VIRTUAL 3D MULTIAGENTE COM RECOMENDAÇÃO PERSONALIZADA DE OBJETOS DE APRENDIZAGEM

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof^o Francisco Milton Mendes Neto, D.Sc. Coorientador: Prof^o Rommel Wladimir de Lima, D.Sc.

	Catalogação na Fonte		
Catalogação de Publicação na Fonte LIFE		ORI ANDO TEIXEIRA .	CAMPUS MOSSORÓ

Frade, Rodrigo Valença Cavalcante.

Univirtual - ambiente virtual 3d multiagente com recomendação personalizada de objetos de aprendizagem / Rodrigo Valença Cavalcante Frade. - Mossoró, 2015. 90f: il.

1. Educação a distância. 2. Ambiente Virtual 3D. 3. Objetos de Aprendizagem. I. Título

RN/UFERSA/BCOT/388 F799u CDD 004.678

RODRIGO VALENÇA CAVALCANTE FRADE

UNIVIRTUAL – AMBIENTE VIRTUAL 3D MULTIAGENTE COM RECOMENDAÇÃO PERSONALIZADA DE OBJETOS DE APRENDIZAGEM

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: 09/02/2015.

BANCA EXAMINADORA

Prof. Dr. Francisco Milton Mendes Neto – UFERSA

Presidente

Prof. Dr. Rommel Wadimir de Lima – UERN

Primeiro Membro

Profa. Dra. Cicília Raquel Maia Leite – UERN

Segundo Membro

Prof. Dr. Eduardo Henrique da Silva Aranha – LIFRN

Terceiro Membro

Dedico este trabalho especialmente à minha avó, Maria Ercy Valença, ao meu filho, Gabriel Valença, e à minha esposa, Paola Ryane. Minhas razões!

AGRADECIMENTOS

Agradeço a Deus por proporcionar-me a vida e também pela sua companhia ao longo dela. Sei que sempre estive protegido por ele em todos os momentos dessa minha jornada.

Devo expressar minha gratidão também aos familiares que sempre me deram apoio nas decisões em que necessitei de conselhos. Especialmente, agradeço aos meus avós que se dedicaram durante um bom tempo de suas vidas me educando, ensinando a como caminhar sozinho, como lutar pelos meus ideais, enfim, me ensinando a viver. Sou muito grato ao meu avô Cláudio e à minha avó Maria Ercy pelo incentivo aos estudos, inclusive o meu ingresso na área de computação foi por motivação deles.

Na caminhada do mestrado, muitas pessoas foram essenciais em minha vida. Dentre estas, os professores do programa que com responsabilidade nos proporcionaram momentos únicos em suas aulas. Os companheiros do Laboratório de Engenharia de Software da UFERSA, dos quais além das amizades concretizadas, foi possível a troca de conhecimentos.

Ao meu orientador Francisco Milton Mendes Neto, que teve muita responsabilidade em me orientar mesmo a distância no início do curso, e agora presencialmente com paciência e muita dedicação. Me proporcionou uma grande experiência com o projeto de pesquisa, experiência essa que serei sempre grato. Além de agradecer pela sua competência, devo agradecer também pela pessoa que se mostrou ser, me compreendendo e me ajudando no que foi preciso ao longo dos dois anos de mestrado.

Não posso deixar de citar o apoio financeiro da CAPES, pois sem ele, talvez eu não tivesse concluído o curso.

Agradeço a todos que fizeram parte dessa minha jornada e que, direta ou indiretamente, tiveram influência na obtenção dessa conquista que tanto almejei.



RESUMO

O avanço dos recursos tecnológicos tem proporcionado novas possibilidades nos processos de ensino e aprendizagem. Tal avanço resultou em uma melhoria para a Educação a Distância (EaD). No entanto, esta modalidade de ensino ainda enfrenta o sério problema da evasão tendo como uma das principais causas, à falta de motivação por parte dos estudantes. Com isso, a utilização de novas tendências tecnológicas tem sido cada vez mais frequente, tendo como propósito oferecer uma maior atratividade para participação do estudante em cursos a distância. Deste modo, este trabalho tem por finalidade propor um ambiente virtual multiagente em três dimensões para apoiar a recomendação de Objetos de Aprendizagem, a fim de melhorar os processos de ensino e aprendizagem na EaD.

Palavras-chave: Ambiente Virtual 3D. Educação a Distância. Objetos de Aprendizagem. Aprendizagem Ubíqua. Agentes. Sistema Multiagente.

ABSTRACT

The advancement of technological resources has provided new possibilities in teaching and learning processes. This progress resulted in an improvement to the Distance Education (DE). However, this type of education still faces the serious problem of circumvention having as a major cause, lack of motivation among students. Thus, the use of new technological trends has been increasingly common with the purpose to provide greater attractiveness for student participation in distance learning courses. Thus, this study aims to propose a multiagent virtual environment in three dimensions to support the recommendation of learning objects in order to improve the teaching and learning processes in DE.

Key-words: 3D Virtual Environment. Distance Education. Learning Objects. Ubiquitous Learning. Agents. Multiagent System.

LISTA DE FIGURAS

Figura 1 - Os 3Is - Imersão, Imaginação e Interação. Fonte: Adaptado de (LI; YUE;	
JÁUREGUI, 2009)	21
Figura 2 – Ilustração da percepção e atuação de um agente. Fonte: Autoria própria	23
Figura 3 – Estrutura básica de um AG. Fonte: Adaptado de (ZINI, 2009)	29
Figura 4 – Aprendizagem ubíqua como uma intersecção entre a aprendizagem móvel e	
a sensibilidade ao contexto. Fonte: Adaptado de (SILVA, 2012)	33
Figura 5 – Estrutura das classes do AG. Fonte: Autoria própria	44
Figura 6 – Código-fonte da classe <i>Cromossomo</i> . Fonte: Autoria própria	46
Figura 7 – Diagrama com algumas das principais classes do AG. Fonte: Autoria própria.	47
Figura 8 – Jogo de canhão criado na plataforma OSX. Fonte: Autoria própria	51
Figura 9 – Telas do segundo jogo com dois cenários diferentes. Fonte: Autoria própria.	52
Figura 10 – Vista aérea do ambiente 3D. Fonte: Autoria própria	54
Figura 11 – Interior de um dos laboratórios. Fonte: Autoria própria	55
Figura 12 – Etapa de criação do personagem 3D. Fonte: Autoria própria	56
Figura 13 – O antes e o depois da aplicação do esqueleto no Mixamo <i>auto-rigger</i> . Fonte:	
(MIXAMO, 2014)	57
Figura 14 – Código-fonte referente à parada do NPC para interagir com o personagem	
estudante. Fonte: Autoria própria	58
Figura 15 — Trecho do código que armazena dados sobre localização geográfica	59
Figura 16 – Telas de cadastro de departamento via GPS. Fonte: Autoria própria	60
Figura 17 — Comunicação inicial entre os componentes. Fonte: Autoria própria	61
Figura 18 – Tela de cadastro de estudantes no Moodle. Fonte: Autoria própria	62
Figura 19 – Tela de autenticação [1] e cenário inicial da UniVirtual [2]. Fonte: Autoria	
própria.	63
Figura 20 – UniVirtual no Android. Fonte: Autoria própria	63
Figura 21 – Arquitetura da UniVirtual. Fonte: Autoria própria	64
Figura 22 – Modelo de tarefas do SMA. Fonte: Autoria própria	66
Figura 23 – Modelo de Papéis do SMA. Fonte: Autoria própria	66
Figura 24 – Modelo de Organização do SMA. Fonte: Autoria própria	67
Figura 25 – Modelo de Interação dos agentes. Fonte: Autoria própria	69
Figura 26 – Cenário inicial[1] e sugestão de OAs[2]. Fonte: Autoria própria	73
Figura 27 – Equação da métrica <i>Precision</i> . Fonte: Autoria própria	80
Figura 28 – Equação da métrica <i>Recall</i> . Fonte: Autoria própria	80
Figura 29 – Equação da métrica <i>F-meansure</i> Fonte: Autoria própria	81

LISTA DE TABELAS

Tabela 1 –	Propriedades dos OAs	36
Tabela 2 -	Categorias do padrão LOM	38
Tabela 3 -	Template do Agente Estudante	70
Tabela 4 -	Template do Agente Recomendador	71
Tabela 5 -	Template do Agente de Interface	72
Tabela 6 -	Resultados da recomendação por nível de dificuldade	78
Tabela 7 –	Resultados da recomendação por curso	79
Tabela 8 -	Resultados da recomendação por curso e nível de afinidade	80
Tabela 9 –	Resultados das métricas <i>Precision</i> , <i>Recall</i> e <i>F-meansure</i> da recomendação	
	por curso e nível de afinidade	81

LISTA DE ABREVIATURAS E SIGLAS

U-learning Ubiquitous Learning, página 31

ADL Advanced Distributed Learning, página 37

AG Algoritmos Genéticos, página 25

AVA Ambiente Virtual de Aprendizagem, página 16

EaD Educação a Distância, página 15

FIPA Foundation for Intelligent Physical Agents, página 51

GPL GNU Public License, página 51

IA Inteligência Artificial, página 24

IOS Iphone Operating System, página 48

JADE Java Agent Development Framework, página 51

LES Laboratório de Engenharia de Software, página 49

LMS Learning Management System, página 18

LOM Learning Object Metadata, página 36

LR Localização Real, página 41

LTSC Learning Technology Standard Committee, página 36

LV Localização Virtual, página 40

Moodle Modular Object-Oriented Dynamic Learning Environment, página 51

NPC Non-Playable Character, página 16

OA Objeto de Aprendizagem, página 15

OSX Operating System Unix, página 48

RV Realidade Virtual, página 20

SCORM Sharable Content Object Reference Model, página 37

SGA Sistema de Gestão de Aprendizagem, página 51

SMA Sistema Multiagente, página 24

SR Sistema de Recomendação, página 32

SRBC Sistema de Recomendação Baseada em Conteúdo, página 33

SRC Sistema de Recomendação Colaborativa, página 33

SRH Sistema de Recomendação Híbrida, página 33

TDV Tecnologia Digital Virtual, página 19

TIC Tecnologias da Informação e Comunicação, página 15

SUMÁRIO

1	INT	RODU	$ ilde{Q} ilde{\mathbf{A}}\mathbf{O}$ 10
	1.1	CONT	EXTUALIZAÇÃO
	1.2	PROB	LEMÁTICA 1
	1.3	OBJET	ΓΙVO GERAL
	1.4	ORGA	NIZAÇÃO DA DISSERTAÇÃO1
2	ASP	ECTO	S CONCEITUAIS
	2.1	AMBI	ENTES VIRTUAIS DE APRENDIZAGEM
		2.1.1	Definição e Evolução dos AVAs
		2.1.2	Paradigma 3D e seu Potencial em AVAs
	2.2	AGEN	TES INTELIGENTES E SISTEMAS MULTIAGENTE
		2.2.1	Definição de Agentes
		2.2.2	Tipologia dos Agentes
		2.2.3	Sistemas Multiagente
	2.3	ALGO	PRITMOS GENÉTICOS
		2.3.1	Definição de Algoritmos Genéticos
		2.3.2	Terminologia dos Algoritmos Genéticos
			2.3.2.1 Genes
			2.3.2.2 Cromossomos
			2.3.2.3 População
			2.3.2.4 Geração
			2.3.2.5 Função de Aptidão
		2.3.3	Passos de um Algoritmo Genético
	2.4	APRE	NDIZAGEM UBÍQUA
		2.4.1	Aprendizagem Móvel
		2.4.2	Ambientes Sensíveis ao Contexto
		2.4.3	Aprendizagem Ubíqua 3
	2.5	SISTE	MAS DE RECOMENDAÇÃO
		2.5.1	Definição de Sistemas de Recomendação
		2.5.2	Sistema de Recomendação Baseada em Conteúdo
		2.5.3	Sistema de Recomendação Colaborativa
		2.5.4	Sistema de Recomendação Híbrida
	2.6	OBJE	TOS DE APRENDIZAGEM
		2.6.1	Definição de Objetos de Aprendizagem
		2.6.2	Padrões de Objetos de Aprendizagem
			2.6.2.1 Padrões de Metadados
			2.6.2.2 Padrões de Integração

3	PRO	DBLEM	IA DE RECOMENDAÇÃO DE OBJETOS DE APRENDIZAGEM .	40
	3.1	ASPE	CTOS CONSIDERADOS	40
		3.1.1	Afinidade com o curso do Moodle	40
		3.1.2	Localização virtual	41
		3.1.3	Localização real/corrente	41
		3.1.4	Palavras-chave	42
		3.1.5	Dificuldade vs Nível do aluno	42
		3.1.6	Tipo de arquivo (preferência)	43
		3.1.7	Idioma	43
	3.2	DESC	RIÇÃO DO PROJETO ALGORITMO GENÉTICO	43
		3.2.1	Classe AG	44
		3.2.2	Classe Cromossomo	45
		3.2.3	Classe OAdata	46
		3.2.4	Passos do Algoritmo Genético da UniVirtual	48
4	UNI	VIRTU	JAL - AMBIENTE TRIDIMENSIONAL DE RECOMENDAÇÃO DE	
	OBJ	JETOS	DE APRENDIZAGEM PARA APOIAR A APRENDIZAGEM UBÍ-	
	QU A	A		49
	4.1	TRAB	BALHOS RELACIONADOS	49
	4.2	FERR	AMENTAS UTILIZADAS	50
		4.2.1	Game Engine Unity 3D	50
		4.2.2	Character Creator Fuse	52
		4.2.3	Plataforma JADE	53
		4.2.4	LMS Moodle	53
	4.3	IMPL	EMENTAÇÕES DO AMBIENTE UNIVIRTUAL	54
		4.3.1	Implementação do ambiente 3D	54
		4.3.2	Implementação dos personagens 3D	55
		4.3.3	Implementação do projeto de captura da localização real	58
	4.4	DESC	RIÇÃO DA UNIVIRTUAL	60
	4.5	ARQU	JITETURA DO AMBIENTE	64
	4.6	SISTE	EMA MULTIAGENTE	65
		4.6.1	Modelagem do SMA	65
		4.6.2	Modelo do Agente Estudante - SAg	70
		4.6.3	Modelo do Agente Recomendador - RAg	70
		4.6.4	Modelo do Agente de Interface - IAg	71
			4.6.4.1 Oferta dos OAs efetuada pelo NPC	72
5	VAI	JIDAÇ Â	ÃO E RESULTADOS	74
	5.1	CENÁ	ARIO DE TESTE	74
		5.1.1	Definição dos Cursos e o Repositório	74
		512	Estruturação do Repositório	74

5.1.3 Estruturação dos Perfis dos Estudantes	 75
5.1.4 Execução e Resultados do Experimento	 75
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	 82
REFERÊNCIAS	 84
APÊNDICE A CLASSE VERIFICANDODISTANCIA	 89

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A tecnologia tem sido uma grande aliada da Educação a Distância (EaD), tendo em vista que muitos recursos didáticos utilizados atualmente por professores e alunos são ferramentas digitais de alto nível de interação que auxiliam, de maneira produtiva e qualitativa, nos processos de ensino e aprendizagem. Atualmente, existem vários desses recursos atrativos, destacando-se os chamados Objetos de Aprendizagem (OA). De acordo com Wiley (2003), um OA consiste em uma entidade que possibilita seu uso, reuso ou referência durante a aprendizagem e utiliza tecnologia.

Ainda é possível citar, como referências em avanços tecnológicos, os dispositivos móveis e a comunicação sem fio, pois estes contribuem também de forma considerável para os processos de ensino e aprendizagem. Dentro deste contexto surge outro conceito, a Aprendizagem Móvel (*Mobile Learning*), que consiste na integração dessas tecnologias para melhorar o processo de aprendizagem (SACCOL et al., 2010).

Os problemas típicos da EaD, tais como a falta de motivação dos estudantes, resultando, muitas vezes, no abandono dos cursos, estimulam atualmente estudos sobre o uso de novas tecnologias (YESSAD; LABAT; KERMORVANT, 2010). A grande vantagem de integrar diferentes tecnologias, de modo que as mesmas interajam entre si para auxiliar nos processos de ensino e aprendizagem, é o fato de todos os usuários poderem usufruir de vários aspectos vantajosos dessa nova abordagem, onde destacam-se a comodidade de prover materiais, acessá-los e visualizá-los, em qualquer horário e lugar (YAU; JOY, 2011).

Atualmente, existe uma grande dificuldade em obter-se a atenção dos jovens para atividades voltadas à aprendizagem. Especificamente na modalidade EaD, este é considerado um grande desafio na presente década. Isso acontece porque os jovens possuem um grande leque de recursos informacionais, bastante interativos, que estão disponíveis ao seu alcance a todo momento. Isso tem atrapalhado o rendimento em atividades de estudo a distância, devido à utilização excessiva desses recursos. Um exemplo são as tecnologias da informação e comunicação (TIC), que dispõem de muito entretenimento, retendo, assim, a atenção dos estudantes.

De acordo com Oliveira e Tedesco (2010), um outro problema está relacionado aos ambi entes virtuais de ensino disponíveis no mercado, que, apesar de estarem em grande quantidade, ainda continuam a tratar todos os seus usuários de maneira semelhante, ou seja, sem oferecer-lhes conteúdos específicos conforme os seus interesses. Consequentemente, muitas vezes não há incentivo para uma construção e compartilhamento efetivos do conhecimento, o que também gera problemas de desmotivação e baixos rendimentos. Portanto, se faz necessária a criação de mecanismos que proporcionem aos ambientes a autonomia de se relacionarem com os

conteúdos de aprendizagem segundo o contexto dos alunos e, em seguida, de recomendarem esses, baseando-se nas características de cada um dos alunos.

Na tentativa de aprimorar estes ambientes, surge então o conceito de ambientes sensíveis ao contexto (*context-aware environments*). Consistem em ambientes que se adequam ao perfil do usuário, considerando informações fornecidas pelo próprio usuário, bem como as capturadas de forma dinâmica a partir de um determinado cenário que pode contar com a interação entre este usuário e dispositivos computacionais (MOORE; JACKSON; HU, 2010).

Tendo em vista essa concepção, o presente trabalho propõe o UniVirtual, um sistema integrado ao ambiente virtual de aprendizagem Moodle, que permite simular uma universidade virtual 3D em navegadores web ou nativamente em sistemas Android (2014). Neste ambiente é possível percorrer, com um personagem estudante, uma universidade virtual que contém alguns NPCs (*Non-Playable Characters*) ¹ controlados por agentes, no intuito de auxiliar nos processos de ensino e aprendizagem através da recomendação de OAs. No dispositivo móvel ainda é possível a captura da localização real do usuário que poderá influenciar na precisão da recomendação.

1.2 PROBLEMÁTICA

Nos processos de ensino e aprendizagem em EaD ainda é possível encontrar algumas falhas. Muitos estudantes abandonam os estudos por falta de motivação nos cursos nesta modalidade. Um fator de influência está relacionado ao uso de algumas tecnologias complexas existentes para auxiliar em atividades voltadas à educação. Segundo Behar (2013), muitos alunos ainda estão em fase de alfabetização e letramento digital, o que prejudica a relação com os recursos digitais atuais. Portanto, se faz necessária a promoção de novas tecnologias caracterizadas pela simplicidade, objetividade e precisão.

Este alto índice de abandono também é resultado da falta de motivação dos estudantes em virtude, muitas vezes, da forma de disponibilização dos conteúdos on-line nos Ambientes Virtuais de Aprendizagem (AVAs), e da cobrança excessiva sobre o estudo destes materiais para as futuras avaliações. Isto gera uma frustração no estudante, de modo que na maioria das vezes o mesmo venha a desistir do curso (YESSAD; LABAT; KERMORVANT, 2010). Outro agravante, já citado anteriormente, é o fato da maioria dos AVAs tratarem todos os seus usuários de modo similar (OLIVEIRA; TEDESCO, 2010).

Ainda existe uma grande dificuldade em obter-se a atenção dos jovens para atividades voltadas à aprendizagem. Isto é considerado um dos desafios da EaD na presente década. Acontece que os jovens possuem um grande leque de recursos informacionais, bastante interativos,

São personagens de jogos eletrônicos que não são manipulados por usuários, porém exercem um papel específico de interatividade com o jogador (SANDES; SILVA, 2012).

que estão disponíveis ao seu alcance a todo momento. Isso tem atrapalhado o rendimento em atividades de estudo na modalidade em questão, devido o uso desses recursos em excesso.

Deste modo, com base nestas afirmações é possível identificar uma barreira na modalidade EaD relacionada à falta de motivação dos estudantes nesta modalidade. Portanto, acredita-se que uma das formas de contribuição para a reversão deste quadro, pode ser proporcionada por meio do investimento da interatividade aplicada em ambientes virtuais de aprendizagem, que contemplem a realidade virtual por meio da tridimensionalidade e, que permitam a aprendizagem móvel com o uso de dispositivos móveis.

Não foram encontrados trabalhos na literatura que abrangessem, em um único ambiente, o (i) paradigma tridimensional, (i) apoio à aprendizagem móvel e (iii) recomendação personalizada de objetos de aprendizagem. Entretanto, na Seção 4.1 do Capítulo 4 são apontados alguns trabalhos relacionados com propostas que influenciaram o desenvolvimento do trabalho aqui apresentado.

1.3 OBJETIVO GERAL

Com base na problemática exposta, o presente trabalho tem como objetivo geral prover um ambiente tridimensional de aprendizagem, executável em computadores e dispositivos móveis com o sistema operacional Android, que auxilie os estudantes nos processos de ensino e aprendizagem tratando os mesmos de forma personalizada, com base nos contextos que estes possam se encontrar em determinados momentos e em seus respectivos perfis de usuários relacionados a conteúdos educacionais disponíveis em forma de OAs.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação encontra-se estruturada da seguinte forma: o Capítulo 2 trata de apresentar os aspectos conceituais, que consistem em toda fundamentação teórica relacionada à pesquisa. O Capítulo 3 define o problema de recomendação de objetos de aprendizagem. No Capítulo 4 encontra-se alguns trabalhos relacionados, em seguida, é descrita a arquitetura do ambiente virtual 3D de aprendizagem, detalhando as principais etapas do seu processo de desenvolvimento e os recursos utilizados. O Capítulo 5 trata da validação e dos resultados obtidos com o trabalho. E, por fim, o Capítulo 6 apresenta as considerações finais e os trabalhos futuros.

2 ASPECTOS CONCEITUAIS

As seções a seguir apresentam o referencial teórico abordado no presente trabalho. A Seção 2.1 trata sobre assuntos relacionados aos ambientes virtuais de aprendizagem. Na Seção 2.2 encontram-se informações sobre agentes inteligentes e sistemas multiagente. Seguidamente, a Seção 2.3 traz uma discussão sobre os algoritmos genéticos. Enquanto a Seção 2.4 contempla informações que definem a aprendizagem ubíqua. Por fim, a Seção 2.6 fala sobre os objetos de aprendizagem e suas características. Este capítulo é de extrema importância para o leitor, pois auxiliará na absorção de definições a cerca das técnicas e recursos utilizados na abordagem aqui apresentada.

2.1 AMBIENTES VIRTUAIS DE APRENDIZAGEM

Esta seção apresenta alguns assuntos relacionados aos Ambientes Virtuais de Aprendizagem (AVAs). A Subseção 2.1.1 define os AVAs e expõe a sua crescente evolução. Já a Subseção 2.1.2 ressalta a potencialização em aplicar o paradigma tridimensional em AVAs.

2.1.1 Definição e Evolução dos AVAs

O progresso constante da EaD tem demarcado um novo horizonte no âmbito educacional. A desconfiança que existia no passado quanto à qualidade nos processos de ensino e aprendizagem dessa modalidade tem se extinguido com o auxílio de novas técnicas que envolvem muitos processos e ferramentas. Como destaque a esta contribuição é possível citar os Ambientes Virtuais de Aprendizagem.

Para Silva (2006), um AVA é um sistema que reúne uma série de recursos e ferramentas, permitindo e potencializando sua utilização em atividades de ensino e aprendizagem através da Internet em um curso a distância. Em concordância, Fialho (2006) reforça que esse ambiente colaborativo de aprendizagem virtual constitui um verdadeiro espaço para o conhecimento.

Cunha e Mainente (2011), por sua vez, descrevem os AVAs como facilitadores dos processos de ensino e aprendizagem na modalidade de educação a distância, em geral uma adaptação do inglês *Learning Management System* (LMS), sendo usados como ferramentas de mediação entre professores e alunos, permitindo o esclarecimento de dúvidas, aplicação de exercícios de fixação, reforço e o acompanhamento do desempenho individual.

Na opinião de Litto e Formiga (2012), os AVAs permitem novas e potencialmente diferentes experiências de aprendizagem que não devem ser desprezadas pelo professor na busca por estratégias para que os estudantes atinjam os seus objetivos de aprendizagem.

Estes ambientes encontram-se em constante evolução devido à influência da potencialização das TICs sobre eles, que permitem uma experiência de ensino e aprendizagem diferenciada por conta dos recursos que estas tecnologias oferecem. Portanto, é notória a gama de soluções integradas aos atuais AVAs, e a cada momento novidades surgem com a promessa de proporcionar maior eficiência nas atividades de ensino e aprendizagem, mais atratividade e, consequentemente, maior estímulo aos estudantes e professores.

Um exemplo do progresso que tem sofrido os AVAs é o uso de mundos virtuais como ambientes de aprendizagem. Essa junção pode proporcionar uma mudança do estilo tradicional de educação para um sistema mais interativo, onde o aluno constrói seu conhecimento e o professor atua verdadeiramente como mediador. Com os mundos virtuais os alunos podem visitar lugares onde jamais poderiam ir e utilizar experimentos remotos que nunca teriam acesso, muitas vezes devido ao elevado custo (MARCELINO, 2010).

Existem diversos propósitos que alicerçam o uso dos mundos virtuais na educação (??MARCELINO, 2010):

- Maior motivação dos estudantes (usuários);
- O poder de ilustração da realidade virtual para alguns processos e objetos é muito maior do que outras mídias;
- Permite uma análise de muito perto;
- Permite uma análise de muito longe;
- Permite que as pessoas deficientes realizem tarefas que de outra forma não são possíveis;
- Dá oportunidades para experiências;
- Permite que o aprendiz desenvolva o trabalho no seu próprio ritmo;
- Não restringe o prosseguimento de experiências ao período da aula regular;
- Permite que haja interação, e desta forma estimula a participação ativa do estudante.

Em meio a toda esta evolução tecnológica proporcionada aos AVAs, Marcelino (2010) também destaca a interface gráfica tridimensional:

Estamos presenciando o surgimento das Tecnologias Digitas Virtuais (TDV), essas tecnologias promovem a criação de ambientes tridimensionais gráficos, onde existe um ser, também 3D, que é capaz de interagir nesse mundo, paralelo ao seu (MARCELINO, 2010).

Enfim, pode-se concluir que os AVAs estão sendo aprimorados com recursos diversos, tomando uma proporção crescente de integrações e tendo sempre como alvo a educação de forma atrativa e estimulante, tentando minimizar o déficit da EaD relacionado à evasão de estudantes ocasionada por falta de motivação.

2.1.2 Paradigma 3D e seu Potencial em AVAs

Um exemplo que se destaca em apoio a este avanço dos AVAs é a Realidade Virtual (RV), que consiste em um conjunto de metodologias e tecnologias para a criação de ambientes interativos totalmente ou parcialmente imersivos (LITTO; FORMIGA, 2012). Técnicas de imersão e interação proporcionadas pela RV fornecem um ambiente rico e interativo que excita a curiosidade, aumentando o entusiasmo para seu uso. Essas técnicas revelam um cenário extremamente motivador e desafiador para ambientes virtuais de ensino e aprendizagem (KIRNER, 2011).

Os pesquisadores Li, Yue e Jáuregui (2009) alegam que o mais importante de uma ferramenta EaD é a forma como a mesma irá atrair a atenção do seu público, em especial, para o processo de aprendizagem. Então, é de fundamental importância levar em consideração os 3Is, ilustrados na Figura 1.

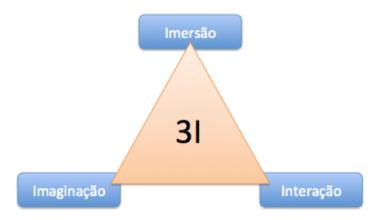


Figura 1 – Os 3Is - Imersão, Imaginação e Interação. Fonte: Adaptado de (LI; YUE; JÁUREGUI, 2009).

Incluem-se, como exemplos, utilizando essas características providas pelos 3Is:

- Entretenimento: jogos, viagens virtuais, etc;
- Saúde: cirurgias virtuais, tratamento de pacientes em UTI, reabilitação, etc;
- Negócios: maquetes virtuais, edificações, projeto de interiores, etc;
- Educação e Treinamento: simuladores de vôo, jogos educativos, etc.

Para prover a RV atualmente é comum o uso de interfaces 3D nos AVAs. Osório et al. (2004) alegam que o paradigma 3D permite a representação da informação de um modo realístico, organizando a mesma de uma forma espacial e tornando sua visualização mais intuitiva por ser mais natural aos olhos humanos.

Segundo Azevedo e Elia (2011), a integração dessas ferramentas computacionais podem proporcionar, aos usuários, novas formas de aprendizagem com a possibilidade da construção de cenários e outros recursos que representam com maior fidelidade o mundo real. Baseado nestas afirmações, Bos et al. (2013) concluem que ambientes virtuais 3D contribuem significativamente

nos processos de ensino e aprendizagem mediados por tecnologias, uma vez que estes podem trazer uma gama de diferentes recursos.

Reforçando, Litto e Formiga (2012) asseguram a intensa interação que um ambiente virtual 3D pode proporcionar:

A possibilidade de criar locais de aprendizagem mais lúdicos e ricos, em várias dimensões, provoca nos alunos uma interação mais intensa e prazerosa com os colegas, o professor, o conteúdo e, principalmente, os objetos e o próprio ambiente, em seu percurso de aprendizagem (LITTO; FORMIGA, 2012).

Entretanto, Litto e Formiga (2012) também colocam em questão que o grau de envolvimento e imersão dos alunos com o conteúdo dos cursos, os colegas e o próprio professor, em um ambiente de realidade virtual, como o SecondLife (2014), não parece ser facilmente reproduzível nos ambientes de aprendizagem tradicionais. Porém, no desenvolvimento do UniVirtual foi possível quebrar essa restrição, ou seja, foi efetuada uma integração com o ambiente virtual de aprendizagem tradicional Moodle (2014), deste modo, enriquecendo mais ainda os processos de ensino e aprendizagem da educação a distância, devido a possibilidade de acesso ao repositório de objetos de aprendizagem do moodle e aos registros de usuários.

2.2 AGENTES INTELIGENTES E SISTEMAS MULTIAGENTE

Nesta seção estão descritos alguns aspectos referentes a agentes e sistemas multiagente. Na Subseção 2.2.1 encontram-se algumas definições sobre agentes de software por vários autores. A Subseção 2.2.2 apresenta um resumo dos tipos de agentes existentes. Já a Subseção 2.2.3 contempla alguns conceitos sobre Sistemas Multiagente.

2.2.1 Definição de Agentes

Conforme Norvig e Russell (2014), agentes são softwares autônomos que percebem o ambiente por meio de sensores e atuam através de atuadores. A Figura 2 ilustra este conceito.

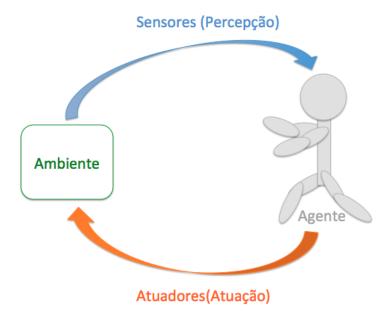


Figura 2 – Ilustração da percepção e atuação de um agente. Fonte: Autoria própria.

Outros autores como, Wooldridge (2009) por exemplo, consideram os agentes como sistemas computacionais situados em um determinado ambiente e que são capazes de efetuar ações autonomamente neste ambiente com a missão de cumprir os objetivos definidos.

Enquanto para Artero (2009) agentes são programas que realizam diálogos para negociação e coordenação de transferências de informação, ou ainda, são programas que executam várias operações para um usuário, com base na representação do conhecimento contido nos objetivos deste usuário.

Henderson-Sellers e Giorgini (2005) descrevem os agentes como sendo entidades de software ou não, porém, caracterizando-os em concordância com as ideias dos autores anteriores, contemplando: autonomia, proatividade e direcionados a objetivos.

2.2.2 Tipologia dos Agentes

Ferber (1999) alega que os agentes podem ser classificados como de software ou hardware, estáticos ou móveis, persistentes ou não-persistentes, reativos ou cognitivos, conforme a sua construção. Sendo a classificação mais comum de agentes a que diferencia agentes reativos de agentes cognitivos, devido a questões de aplicabilidade.

Os agentes reativos são aqueles que selecionam ações para serem executadas com base exclusivamente na percepção atual, sem levar em consideração algum histórico de percepções. Deste modo, são incapazes de prever ações futuras, uma vez que não possuem memória. Enquanto os agentes cognitivos são mais complexos, pois os mesmos dispõem de uma representação explícita do ambiente e dos outros agentes. Possuem memória interna, possibilitando o planejamento de ações futuras baseadas em situações específicas que ocorreram anteriormente (NORVIG;

RUSSELL, 2014).

Na tentativa de melhor especificar essa classificação, são citados a seguir, conforme Norvig e Russell (2014) e Artero (2009), alguns dos tipos básicos de agentes:

- Agentes tabela: nesta estrutura o nível de complexidade é bastante baixo, tendo em vista a sua logística, pois este tipo requer apenas uma tabela contendo todas as percepções e ações possíveis para uso. O grande problema dessa abordagem reside na necessidade de incluir todas as percepções e ações possíveis na tabela, o que demanda tempo.
- Agentes reativos simples: esses agentes selecionam ações com base na percepção atual, ignorando o restante do histórico de percepções. Os agentes reativos simples podem atuar mesmo em ambientes mais complexos. Estes agentes possuem um conjunto de regras do tipo condição-ação que substitui de forma satisfatória a estrutura do agente tabela visto anteriormente, que necessitava conter em sua tabela todas as possíveis percepções e ações. Por não possuírem memória, são incapazes de planejar ações futuras.
- Agentes baseados em modelos: também conhecidos como agentes reativos com estado interno, esse tipo de agente controla o estado atual do mundo usando um modelo interno do ambiente que depende do seu histórico de percepções e, dessa forma, reflete, no mínimo, alguns aspectos não observados no estado atual. Este agente combina as informações da percepção atual com as provenientes do modelo para gerar a descrição atualizada do estado atual. De posse dessas informações, ele escolhe uma ação da mesma forma que o agente reativo simples.
- Agentes baseados em objetivos: são conhecidos como agentes cognitivos, eles ponderam as suas ações considerando a descrição do estado atual e os objetivos almejados. Portanto, é efetuada uma combinação entre essas ações ponderadas e os objetivos desejados e, em seguida, escolhida uma ação que (no final) levará à realização de seus objetivos. É possível identificar que a tomada de decisões desse tipo de agente é distinta das regras condição-ação mencionadas anteriormente (em agentes reativos simples), pelo fato de envolver considerações do futuro.
- Agentes baseados em utilidade: com a insuficiência dos objetivos para gerar um comportamento de alta qualidade na maioria dos ambientes, este tipo de agente escolhe suas ações tentando sempre maximizar uma função de utilidade. Essa função mapeia um estado ou uma sequência de estados em um número real, que descreve o grau de satisfação do agente caso aquele estado seja alcançado.

Norvig e Russell (2014) ainda apresentam a possibilidade de serem construídos mecanismos de aprendizado para otimizar as ações desses tipos de agentes. Para isto, são utilizados algoritmos de aprendizagem. Com essa metodologia, consequentemente, o desempenho dos agentes aumenta de forma significativa, o que tem tornado esse método o preferencial em muitas

áreas de Inteligência Artificial (IA), pois a maior vantagem na utilização do aprendizado é permitir que os agentes operarem ambientes inicialmente desconhecidos, porém, que eles irão se adaptar conforme as mudanças destes ambientes.

2.2.3 Sistemas Multiagente

Com as definições e tipologia de agentes apresentadas anteriormente, se torna possível introduzir o conceito de Sistemas Multiagente (SMA). Portanto, em um sistema computacional é possível a utilização de vários agentes de software que interagem entre si, caracterizando os SMAs.

De acordo com Luger (2014), um SMA consiste em um programa de computador com "resolvedores" de problemas localizados em ambientes interativos, onde os mesmos são capazes de executar ações flexíveis, autônomas e, ainda, socialmente organizadas que podem ser dirigidas para metas ou objetivos predeterminados.

Em outras palavras ainda é possível afirmar que um SMA consiste em um conjunto de agentes colaborativos a fim de resolver um problema em comum, e que pode resolver um problema maior dividindo o mesmo em vários subproblemas, a fim de otimizar o tempo de solução.

Luger (2014) também apresenta quatro critérios para um SMA. Ele alega que o mesmo deve incluir agentes que são:

- Situados;
- Autônomos:
- Flexíveis;
- Sociais.

Artero (2009) acrescenta que é bastante comum a necessidade de se usar as habilidades de tipos distintos de agentes para solucionar determinados problemas, o que resulta na criação de um SMA. Além disso, os sistemas multiagente podem apresentar maior eficiência na solução de problemas, devido o paralelismo que pode ser obtido e a flexibilidade que esses sistemas possuem combinando as diferentes habilidades dos agentes.

A utilização de um SMA pode auxiliar em tarefas mais complexas dentro de um ambiente de aprendizagem, como, por exemplo, monitoramento de atividades de estudantes, captura de informações sobre o seu contexto dinâmico, recomendação de conteúdo baseada em seu perfil de usuário, dentre outras.

O SMA do UniVirtual dispõe de um conjunto de agentes que monitoram as atividades do estudante quanto à sua localidade física e seu acesso ao sistema. Eles capturam de forma

automática as informações dos estudantes e, em seguida, recomendam objetos de aprendizagem que possam ajudar o estudante em determinado assunto ou simplesmente recomendam algo de seu interesse. Na Seção 4.6 encontram-se explicações em detalhes sobre o SMA do ambiente UniVirtual.

2.3 ALGORITMOS GENÉTICOS

Esta seção trás alguns conceitos básicos sobre os Algoritmos Genéticos (AG) necessários para auxiliar no entendimento do AG desenvolvido para o UniVirtual. A Subseção 2.3.1 apresenta algumas definições sobre AGs. A Subseção 2.3.2 descreve a terminologia dos AGs com conceitos específicos. Por fim, a Subseção 2.3.3 mostra a estrutura básica de um AG.

2.3.1 Definição de Algoritmos Genéticos

Para se chegar a uma definição clara sobre os algoritmos genéticos, se faz necessário falar sobre a sua relação com a teoria da evolução. Segundo Linden (2012), os AGs estão inteiramente ligados aos princípios da teoria da evolução proposta por Charles Rivert Darwin (1809-1882), em especial no que diz respeito à "seleção natural", que consiste na concorrência entre indivíduos da mesma espécie, onde os que não obtiverem êxito em determinadas tarefas tendem a ter uma prole menor e esta descendência reduzida faz com que a probabilidade de ter seus genes propagados ao longo de sucessivas gerações seja menor.

Linden (2012) ainda afirma que a combinação das características dos indivíduos sobreviventes pode gerar novos indivíduos com uma capacidade de adaptação ao seu meio ambiente bem maior que a dos seus genitores. Entretanto, esses novos indivíduos são descendentes diferentes dos seus pais, podendo estes terem diferenças positivas, bem como negativas.

AGs são algoritmos evolucionários (SILVA, 2012). Estes últimos são algoritmos que usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas (LINDEN, 2012). Ainda que exista uma vasta quantidade de modelos computacionais propostos, todos eles possuem o mesmo princípio básico, ou seja, todos simulam a evolução das espécies utilizando os denominados operadores genéticos, sendo assim, baseando-se na teoria da evolução (LINDEN, 2012; NETO; FAJ, 2011).

Em acordo com essa influência da teoria da evolução sobre os AGs, Koza (1992) descreve este tipo de algoritmo da seguinte forma:

O AG é um algoritmo matemático altamente paralelo que transforma um conjunto (população) dos objetos matemáticos individuais (cadeias de caracteres de comprimento fixo típicos padronizados após sequências cromossômicas), cada um com um valor adaptativo associado, em uma nova população (ou seja, a próxima geração), usando operações padronizadas após o princípio Darwiniano de reprodução e sobrevivência do mais forte e depois ocorre naturalmente operações genéticas (recombinação) (KOZA, 1992).

Com outras palavras, Artero (2009) afirma que os AG tentam resolver problemas dos quais não existe um algoritmo conhecido. Primeiramente ele gera uma população inicial e, conforme critérios de avaliação, seleciona os melhores indivíduos dessa população, que irão servir como solução para o problema ou, do contrário, serão combinados para obter uma nova geração até que seja encontrada uma solução ou até que se perceba que não serão alcançadas melhores soluções nas novas gerações (SILVA, 2012).

Em meio a essas definições é possível considerar que, para o âmbito computacional, essas informações ainda são insuficientes. Portanto, para entender melhor o funcionamento de um AG se faz necessário ter em mente conceitos mais específicos sobre o assunto.

2.3.2 Terminologia dos Algoritmos Genéticos

Foi possível entender anteriormente, que os AGs são altamente inspirados na genética e na teoria da evolução das espécies. Entretanto, é importante a familiarização com a terminologia, pois há uma analogia muito forte entre os termos da biologia e os termos usados no campo dos AGs (LINDEN, 2012).

2.3.2.1 Genes

Segundo Neto e FAJ (2011), na genética, os genes consistem no material que poderá ser trocado entre os cromossomos durante a reprodução. Enquanto na abordagem dos AGs, estes representam os parâmetros de interesse, seguindo algum alfabeto pré-estabelecido, podendo ser representados comumente nesta área por valores 0 e 1 de um alfabeto binário que representa cada gene, consistindo apenas em uma divisão cromossomial (ARTERO, 2009; NETO; FAJ, 2011).

2.3.2.2 Cromossomos

Os cromossomos são compostos por uma cadeia de genes e representam os indivíduos da população, os quais, no caso dos AGs, representam as soluções encontradas em um problema

de otimização (ARTERO, 2009). É importante apontar que, nos sistemas naturais, um ou mais cromossomos se combinam para formar um indivíduo, porém, no caso dos AGs, os termos cromossomo e indivíduo são intercambiáveis, sendo utilizados como sinônimos (LINDEN, 2012; FONTES, 2013).

2.3.2.3 População

A população é formada por um conjunto de indivíduos que irão competir pela sobrevivência e pela reprodução, objetivando perpetuar suas características (ARTERO, 2009; FONTES, 2013).

2.3.2.4 Geração

A geração corresponde a uma população em um certo período. No caso dos AGs correspondem aos indivíduos obtidos em uma dada iteração (número de execuções) (ARTERO, 2009; FONTES, 2013).

2.3.2.5 Função de Aptidão

A função de aptidão é utilizada para medir a habilidade do indivíduo para sobreviver e reproduzir. Nos algoritmos genéticos são representadas por funções matemáticas, que devem ser maximizadas ou minimizadas conforme a relevância da habilidade do indivíduo (ARTERO, 2009). Também conhecida como Função Objetivo ou Função de Fitness (ZINI, 2009) ou mesmo Função de Avaliação (LINDEN, 2012), ela é responsável pelo funcionamento correto de um AG, pois ela faz a ligação do algoritmo com o problema real (NETO; FAJ, 2011).

2.3.3 Passos de um Algoritmo Genético

Após os conceitos vistos anteriormente, é possível definir uma estrutura básica de AG. Para isto, Linden (2012) e Neto e FAJ (2011) citam alguns passos que o mesmo deve exercer de forma algorítmica:

- 1. Inicializa-se a população de cromossomos inicial;
- Avalia-se cada um dos cromossomos na população através de uma função de aptidão, de modo a encontrar uma classificação dos indivíduos mais adaptados ao problema;

- 3. Seleciona-se os indivíduos que servirão como pais para criação de uma nova população de acordo com uma estratégia de seleção previamente definida;
- 4. Aplica-se os operadores genéticos (cruzamento e mutação) aos indivíduos selecionados no passo anterior para gerar os indivíduos da nova população;
- 5. Elimina-se os cromossomos da população antiga, de forma que os novos cromossomos gerados possam ser inseridos sem alterar o tamanho da população inicial;
- 6. Aplica-se a função de aptidão a todos os novos cromossomos e insere-se os melhores selecionados na população anterior, gerando uma nova população;
- 7. Se a população de cromossomos atual representar o resultado esperado ou se a quantidade máxima de gerações foi atingida ou ainda se o algoritmo não conseguir mais mostrar evolução, a execução deve parar, do contrário, o passo 3 deve ser reiniciado.

Para representar a estrutura básica de um AG, Zini (2009) utiliza o diagrama ilustrado na Figura 3.

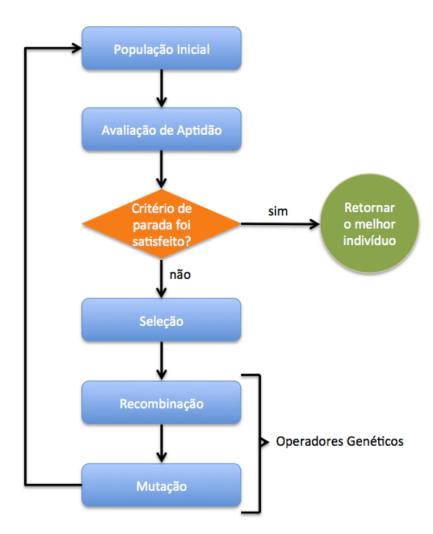


Figura 3 – Estrutura básica de um AG. Fonte: Adaptado de (ZINI, 2009).

Com o término da execução do algoritmo apresentado na figura anterior, espera-se que a população de cromossomos gerada seja a melhor adaptada à função e, por conseguinte, a que melhor represente o resultado do problema. Porém, apesar do algoritmo fornecer uma representação geral e esclarecedora a respeito do funcionamento de um AG, é apenas uma visão superficial do problema, ou seja, ele esconde aspectos mais complexos que devem ser tratados, tais como (NETO; FAJ, 2011; LINDEN, 2012; FONTES, 2013):

- 1. Escolha de uma codificação dos cromossomos adequada ao problema;
- 2. Definição do tamanho da população necessária;
- 3. Definição da forma como será realizada a mutação;
- 4. Seleção de uma função de aptidão que avalie satisfatoriamente o grau de adequação de cada indivíduo como solução do problema em questão.

Estes itens são os principais pontos a serem tratados quanto aos AGs. Mais a frente, na seção que trata da implementação do AG utilizado na UniVirtual, encontram-se mais detalhes sobre estes aspectos, porém, específicos da implementação para o ambiente virtual 3D aqui apresentado.

2.4 APRENDIZAGEM UBÍQUA

Nesta seção estão expostos dois temas que auxiliarão na compreensão do conceito de aprendizagem ubíqua. Portanto, como primeiro tema a Subseção 2.4.1 apresenta conceitos sobre aprendizagem móvel. Seguidamente, na Subseção 2.4.2 encontram-se os conceitos sobre ambientes sensíveis ao contexto. Finalmente, na Subseção 2.4.3 é apresentada a definição de aprendizagem ubíqua.

2.4.1 Aprendizagem Móvel

A aprendizagem móvel, também conhecida pelo termo em inglês *mobile learning*, está atrelada ao uso de dispositivos móveis, tais como telefones celulares, *tablets*, notebooks, em atividades de aprendizagem realizadas a partir de qualquer lugar e hora, dispondo informação e conhecimento para uma situação e lugar onde a atividade de aprendizagem acontece (CASTILLO; AYALA, 2008).

Yau e Joy (2011) afirmam que a aprendizagem móvel não se concretiza com apenas o uso de dispositivos móveis, e sim, quando o estudante tira vantagens de oportunidades de aprendizagem oferecidas por tecnologias móveis. Em outras palavras, os autores consideram a mobilidade

do estudante independente de dispositivos. Portanto, pode-se dizer que a aprendizagem móvel consiste no ato de participar de um processo de aprendizagem de maneira flexível, em qualquer lugar/horário, com qualquer dispositivo, considerando apenas as tecnologias móveis utilizadas.

A modalidade EaD, quando provida através do uso de dispositivos móveis, se enquadra no contexto de aprendizagem móvel conceituado anteriormente. Este método de ensino e aprendizagem oferece a estudantes e professores a possibilidade de usufruir das vantagens dos recursos ofertados pelas tecnologias móveis, incluindo a independência de hora e local, concretizando assim as definições apresentadas pelos autores Castillo e Ayala (2008) e Yau e Joy (2011) sobre este tema.

Entretanto, isso ainda é insuficiente para proporcionar a absorção, com eficácia, de conhecimento pelos estudantes. Isto se deve pelo fato da necessidade de oferecer um conteúdo de forma adequada, considerando-se as características cognitivas e particulares de cada estudante. Baseado nisto, observa-se a grande relevância do conceito de ambientes sensíveis ao contexto.

2.4.2 Ambientes Sensíveis ao Contexto

Os ambientes sensíveis ao contexto (*context-aware environments*) são considerados como aqueles que se adequam ao perfil do usuário, levando em consideração informações particulares fornecidas pelo próprio usuário, além daquelas captadas dinamicamente a partir de sua interação com os dispositivos computacionais (CABALLE et al., 2010).

Moore et al. (2009) também definem o contexto como sendo algo muito específico a aplicações e objetivos almejados, onde o mesmo pode implicar em inúmeras situações, ou seja, não existe um consenso em relação ao que seria de fato o contexto. Portanto, é possível considerar situações diante de comportamentos, dentre outros atributos.

Deste modo, quase todas as informações disponíveis no momento de uma interação podem ser consideradas como atributos contextuais, dentre os quais, segundo Moore et al. (2009), destacam-se:

- 1. As diversas tarefas realizadas pelos usuários no sistema;
- 2. A variada gama de dispositivos que se combinam para criar sistemas móveis, com a infraestrutura de serviços associada;
- 3. Os recursos disponíveis (ex.: condição da bateria, tamanho de tela etc.);
- 4. A situação física (ex.: nível de ruído, temperatura, nível de luminosidade etc.);
- 5. A informação espacial (ex. localização, velocidade, orientação etc.);
- 6. A informação temporal (ex. hora do dia, data etc.).

Os itens apresentados anteriormente servem para demonstrar a complexidade inerente ao contexto, sua natureza de domínio específico e a dificuldade em defini-lo e medi-lo. Na tentativa de diminuir essa complexidade, Moore et al. (2009) definem dois tipos gerais de contexto:

- 1. Contexto estático (denominado customização): diz respeito à situação na qual um perfil do usuário é criado manualmente, estando o usuário ativamente envolvido no processo e tendo um elemento de controle.
- 2. Contexto dinâmico (denominado personalização): refere-se à condição na qual o usuário é visto como sendo passivo, ou pelo menos com um pouco menos de controle. Nesse caso, o sistema monitora, analisa e reage dinamicamente ao comportamento do usuário e ao papel identificado. Muitas aplicações de aprendizagem móvel sensíveis ao contexto utilizam contextos de aprendizagem a fim de adaptar ou sugerir apropriadamente atividades e conteúdos para os estudantes (YAU; JOY, 2011).

Um cenário de sensibilidade ao contexto, como exemplo no UniVirtual, pode ser considerado o momento em que o usuário encontra-se autenticado ao sistema e, com o avatar, se desloca ao encontro de um NPC num determinado departamento/área de estudo da universidade virtual e uma recomendação de um OA é efetuada por este mesmo NPC. Sendo assim, essa detecção do avatar é consideravelmente sensível ao contexto, e o contexto seria a área de estudo relacionada a este ambiente em que o estudante se encontra presente no dado momento e as informações quanto ao seu perfil de usuário, ou seja, características particulares, dentre várias outras informações.

Ao considerar-se as características da aprendizagem móvel e a capacidade de detectar as informações de contexto do estudante, a fim de fornecer conteúdo educacional de forma personalizada, surge o conceito de aprendizagem ubíqua (SACCOL et al., 2010).

2.4.3 Aprendizagem Ubíqua

É possível definir a aprendizagem ubíqua, também conhecida como u-*learning*, de *ubiquitous learning*, como a junção da aprendizagem móvel e a sensibilidade ao contexto. Ou seja, a aprendizagem móvel pode proporcionar o uso de tecnologias móveis, enquanto que a sensibilidade ao contexto permite, através dos dispositivos ou tecnologias móveis, por exemplo, a captura do contexto de forma automática (MANDULA et al., 2011).

A aprendizagem ubíqua pode ser considerada como sendo a aprendizagem móvel que é realizada levando-se em consideração as características do contexto dos estudantes, provendo a estes conteúdos adaptados às suas necessidades (MANDULA et al., 2011). Assim, a aprendizagem ubíqua provê um novo paradigma, através do uso de dispositivos móveis, que oferece um serviço de forma transparente aos estudantes (LOUREIRO et al., 2009).

A Figura 4 representa a ideia de considerar a aprendizagem ubíqua como uma intersecção entre a aprendizagem móvel e a sensibilidade ao contexto.

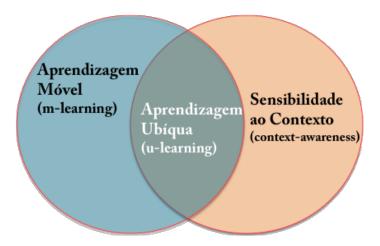


Figura 4 – Aprendizagem ubíqua como uma intersecção entre a aprendizagem móvel e a sensibilidade ao contexto. Fonte: Adaptado de (SILVA, 2012).

Vale ressaltar que a mobilidade proporcionada ao usuário, graças ao uso de dispositivos móveis em aplicações ubíquas, torna ainda mais importante a consideração do contexto de um estudante, visto que as características do estudante podem se modificar a qualquer momento. Essas modificações podem ocorrer em diversos aspectos (ex. condições físicas, recursos físicos disponíveis, recursos computacionais etc.) (LOUREIRO et al., 2009).

Também é relevante citar que, na literatura, a computação ubíqua ainda não possui um conceito bem definido. Para Mandula et al. (2011), na aprendizagem ubíqua envolvemse a aprendizagem móvel e a sensibilidade ao contexto, já para Zhao e Wang (2011), outras características estão envolvidas, como, por exemplo, a computação pervasiva. Porém, no presente trabalho é considerada a definição provida em Mandula et al. (2011), vista anteriormente.

2.5 SISTEMAS DE RECOMENDAÇÃO

Nesta seção são apresentados alguns conceitos sobre os Sistemas de Recomendação (SRs), que podem ser encontrados na Subseção 2.5.1, e sua tipologia que está dividida nas subseções 2.5.2, 2.5.3 e 2.5.4.

2.5.1 Definição de Sistemas de Recomendação

Silva, Neto e Júnior (2011) descrevem que um método de auxílio à escolha de um determinado conteúdo diante de um grande leque de opções é o uso de Sistemas de Recomendação. Os SRs auxiliam o usuário no processo de indicação ou recepção de materiais relevantes. Em uma

visão geral de um sistema típico de filtragem de informação, as pessoas fornecem recomendações como entradas e o sistema registra e direciona para os indivíduos específicos, ou seja, que são considerados potenciais interessados (CAZELLA et al., 2012).

Júnior et al. (2012) alegam que no âmbito dos AVAs, os estudantes são expostos a uma grande quantidade de OAs, o que resulta em uma demanda de tempo elevada para realizar escolhas difíceis, tendo em vista que estão ainda em processo de formação e, portanto, sem aptidão suficiente para realizar tais escolhas. Diante desse cenário, os SRs podem ser uma boa opção para a indicação de OAs apropriados para estudantes específicos.

Em Vieira e Nunes (2012) são apresentados três tipos de SRs: Sistema de Recomendação Baseada em Conteúdo (SRBC), Sistema de Recomendação Colaborativa (SRC) e Sistema de Recomendação Híbrida (SRH). As subseções a seguir descrevem cada um desses tipos.

2.5.2 Sistema de Recomendação Baseada em Conteúdo

A recomendação baseada em conteúdo é realizada através de informações recuperadas do usuário (perfil, comportamento) e/ou do seu histórico de escolhas (VIEIRA; NUNES, 2012). Essa abordagem tem suas raízes na área de recuperação de informação. Devido aos significativos avanços feitos pelas comunidades de filtragem de informação e filtragem de conteúdo, muitos sistemas baseados em filtragem de conteúdo focam na recomendação de itens com informações textuais, como documentos e sites (CAZELLA; NUNES; REATEGUI, 2010; SALES, 2014).

Desta forma, a filtragem baseada em conteúdo parte do princípio de que os usuários tendem a interessar-se por itens similares aos que demonstraram interesse no passado, definindo então a similaridade entre os itens (HERLOCKER, 2000; SALES, 2014).

Segundo Herlocker (2000), podem ser citados como exemplos de filtragem baseada em conteúdo sistemas de recuperação booleanos, onde a consulta é um conjunto de palavras-chave combinadas com operadores booleanos; sistemas de filtragem probabilística, onde o raciocínio probabilístico é utilizado para determinar a probabilidade que um documento possui de atender as necessidades de informação de um usuário; e processamento de linguagem natural.

2.5.3 Sistema de Recomendação Colaborativa

Nos sistemas colaborativos, a essência está na troca de experiências entre as pessoas que possuem interesses comuns. Nestes sistemas, os itens são filtrados com base nas avaliações feitas pelos usuários (CAZELLA; NUNES; REATEGUI, 2010). Dessa forma, os sistemas colaborativos partem do princípio que usuários semelhantes irão gostar de itens semelhantes.

Esta pontuação permite ao usuário descobrir itens que são considerados de interesse pelo grupo e evitar os itens que são considerados de pouco interesse. Sistemas mais avançados

descobrem de maneira automática relações entre usuários (vizinhos mais próximos), com base na descoberta de padrões comuns de comportamento (CAZELLA; NUNES; REATEGUI, 2010; SALES, 2014).

2.5.4 Sistema de Recomendação Híbrida

O SRH trata da recomendação através de técnicas utilizadas pelos SRBC e SRC, aumentando as possibilidades e sugerindo conteúdos que não aparentam se relacionar (MARTINS et al., 2011).

Herlocker (2000) afirma que ao integrar ambas as técnicas, a Filtragem Híbrida busca incorporar as vantagens tanto da Filtragem Colaborativa quanto da Filtragem baseada em Conteúdo, ao mesmo tempo em que minimiza possíveis limitações de uma ou outra abordagem.

2.6 OBJETOS DE APRENDIZAGEM

A presente seção trata sobre os Objetos de Aprendizagem (OAs). A mesma encontra-se organizada com duas subseções. Na Subseção 2.6.1 encontram-se algumas das definições de OAs. Já a Subseção 2.6.2 fala sobre os principais tipos de padrões de OAs, especialmente os utilizados neste trabalho.

2.6.1 Definição de Objetos de Aprendizagem

De acordo com o Comitê de Padronização de Tecnologias de Aprendizagem (LTSC – *Learning Technology Standard Comitee*) do *Institute of Electrical and Electronics Engineers* (IEEE), um OA consiste em uma entidade material educacional, digital ou não, que pode ser usada para aprendizagem, educação ou treinamento (COMMITTEE et al., 2002).

Pode-se dizer que a ideia central do conceito de OAs é permitir que os colaboradores educacionais construam objetos educativos relativamente pequenos e que possam ser usados em diferentes contextos de aprendizagem. São conteúdos digitais que permitem alcançar um objetivo educacional promovendo a reusabilidade dos conteúdos utilizados no processo de ensino e aprendizagem (CASTILLO; AYALA, 2008; SILVA, 2012).

Contudo, se faz necessária a garantia dos benefícios propostos na utilização de OAs, deste modo, algumas características são fundamentais e devem ser consideradas na criação de OAs digitais, como, por exemplo: (a) definição da estrutura de navegação; (b) adequação do conteúdo de uma mídia escrita para uma mídia eletrônica; (c) atendimento aos aspectos pedagógicos de ensino; e (d) integração do OA com diferentes tipos de ambientes de EaD (RODOLPHO, 2009).

A ADL (2008) descreve algumas propriedades para OAs. Na Tabela 1 encontram-se essas propriedades. É possível notar algumas semelhanças ou relações dessas propriedades para com as quatro características mencionadas no parágrafo anterior.

Tabela 1 – Propriedades dos OAs.

Propriedade dos	Explanação	Exemplo
OAs		
Disponibilidade	Um OA pode ser facil- mente acessado via In- ternet para ser usado em diversos locais.	Um autor de OA pode procurar um determinado registro de OA, outros registros, e repositórios, e identificar OAs relevantes que já foram desenvolvidos, possivelmente, até mesmo por outra organização, e implantar esses OAs em um ambiente para os alunos em qualquer lugar do mundo.
Interoperabilidade	Um OA pode ser utilizado e gerenciado através de diferentes plataformas de hardware, sistemas operacionais e navegadores Web, possibilitando intercâmbio efetivo entre diferentes sistemas.	Os OAs carregados em um ambiente de aprendizagem compatível com um determinado padrão podem ser carregados em um outro ambiente compatível com este mesmo padrão, e serem disponibilizados aos alunos.
Durabilidade	Um OA pode ser utilizado e reutilizado independentemente da tecnologia.	Atualizações para computadores locais, tais como a mudança para um novo sistema operacional do computador, ou alterações do lado do servidor, tais como a atualização para novas versões dos ambientes virtuais de aprendizagem, não devem ter nenhum impacto sobre a entrega de OAs aos alunos.
Reusabilidade	Um OA pode ser reutilizado diversas vezes, em diversos ambientes de aprendizagem.	OAs projetados para uma determinada organização podem ser facilmente reaproveitáveis, reorganizados, ou reescritos para/por outras organizações que tenham necessidades de aprendizagem semelhantes.

A garantia dessas propriedades se dá graças à utilização de padrões de desenvolvimento

de OAs.

Além de seu conteúdo propriamente dito, os OAs possuem uma estrutura contemplada por metadados que permitem, através dos seus elementos e atributos, descrever o seu conteúdo, sua formatação, como esse conteúdo é apresentado, além de outras informações (RODOLPHO, 2009; SILVA, 2012). Esses metadados são criados baseando-se também na utilização dos padrões de desenvolvimento de OAs.

2.6.2 Padrões de Objetos de Aprendizagem

Para Kemczinski et al. (2011), os padrões de OAs representam uma forma de organização dos dados para prover comunicação entre diferentes ambientes, bem como o seu acesso e usabilidade, além de garantir sua interoperabilidade. Esses padrões são divididos, de acordo com suas funcionalidades, em: padrões de metadados, de empacotamento, de interface e comunicação e de integração.

Dentre estes padrões, os mais adequados para a realização deste trabalho foram os de metadados, visto na Subseção 2.6.2.1, e de integração, visto na Subseção 2.6.2.2, pois ambos possuem características que contribuem para a recomendação de OAs na UniVirtual. Isso se deve pela forma de como esses dois padrões discriminam os OAs no repositório do moodle.

2.6.2.1 Padrões de Metadados

Os padrões de metadados, de acordo com Dias et al. (2009), são utilizados na identificação de recursos, auxiliando na filtragem de uma busca e na recuperação de um registro ou de um OA. Li, Yang e Liu (2008) definem os metadados de forma simplificada e objetiva, alegando que os mesmos consistem em dados sobre dados. Em outras, palavras significa dizer que os metadados são dados de objetos e descrevem informações sobre o conteúdo destes.

O LOM (*Learning Object Metadata*) (COMMITTEE et al., 2002) é um exemplo de padrão de metadado que foi desenvolvido pelo LTSC, com o objetivo de facilitar a busca, a avaliação, a aquisição e o uso de OAs, por estudantes, instrutores, ou até mesmo por processos de softwares automatizados (SILVA, 2012).

A grande vantagem na sua utilização é que, através da especificação de um esquema de dados comum, este padrão garante que as linguagens de OAs tenham um alto grau de interoperabilidade semântica (COMMITTEE et al., 2002). Neste padrão há um conjunto total de 59 metadados separados em nove categorias. Cada conjunto com seus metadados específicos (ROSSI et al., 2013). Na Tabela 2 encontram-se as nove categorias.

Tabela 2 – Categorias do padrão LOM.

Categoria	Descrição
Geral (General)	Conjunto com nove metadados para descrição
	do objeto como um todo, com títulos, palavras-
	chaves, entre outros.
Ciclo de vida (<i>Lifecycle</i>)	Conjunto de cinco metadados relacionados à his-
	tória do objeto, desde sua data de criação até
	todas suas modificações e o estado atual.
Meta-Metadado (Meta-Metadata)	Conjunto de sete metadados para a informação
	sobre os próprios metadados, como a pessoa que
	fez os registros dos OAs.
Técnico (Technical)	Conjunto de dez metadados sobre os requisitos
	e características técnicas do OA, como o tipo
	(vídeo, imagem).
Educacional (Educational)	Conjunto de onze metadados para uso educaci-
	onal e pedagógico, com dados como a idade do
	público, se o OA é para professores (como plano
	de aula) ou direto para os alunos.
Direitos (Rights)	Conjunto com três metadados para declarações
	sobre os direitos autorais e de uso.
Relação (Relation)	Conjunto com quatro metadados para criar rela-
	ções entre os OAs. Por exemplo, OA-1 "faz parte
	de" OA-2 ou OA-1 "é uma versão de" OA-2.
Anotação (Annotation)	Conjunto de três metadados para que se ponham
	comentários sobre o uso educacional do OA;
	essa categoria permite ao educador compartilhar
	suas opiniões e sugestões de uso.
Classificação (Classification)	Conjunto de sete metadados para uma especifica-
	ção do objeto em cada sistema de classificação,
	aí se põe uma relação à uma taxonomia para
	classificação específica do objeto.

2.6.2.2 Padrões de Integração

Se faz necessário descrever as características do padrão de integração SCORM (*Shareable Content Object Reference Model*) utilizado neste trabalho. O SCORM, desenvolvido pela ADL (*Advanced Distributed Learning*) (ADL, 2008), tem como forte característica do padrão de integração a unificação em um modelo de referência de diferentes tipos de padrões.

Em outras palavras, conforme Silva et al. (2012), esse padrão é composto por uma coleção de especificações, adaptadas de múltiplas fontes, que define um conjunto de procedimentos para a construção e o empacotamento de conteúdo instrucional, bem como protocolos para comunicação entre conteúdo e ambientes virtuais de aprendizagem. Portanto, a utilização do SCORM possibilita o uso das melhores características de cada um dos padrões disponíveis, o que motivou a sua escolha na abordagem aqui apresentada.

A escolha pelo SCORM se deu pelo fato de o mesmo ser considerado um padrão amplamente difundido na literatura, o que facilita inclusive em consultas sobre ele, e também devido à sua composição por outros padrões. Segundo Silva (2012), é possível utilizar, inclusive, os recursos providos pelo padrão LOM, possibilitando o benefício de desfrutar-se dos melhores recursos oferecidos por cada um dos padrões.

Desta forma é possível afirmar que o SCORM consiste em uma coleção de padrões e especificações para e-*learning* baseado na web. A utilização do mesmo foi de grande relevância, tendo em vista que, em sua coleção, encontra-se o padrão LOM do qual permitiu a manipulação de metadados dos OAs disponíveis no repositório da base de dados do Moodle.

3 PROBLEMA DE RECOMENDAÇÃO DE OBJETOS DE APRENDIZAGEM

Neste capítulo encontra-se uma descrição sobre as características que foram consideradas para a realização da recomendação de OAs implementada no ambiente multiagente da UniVirtual. Na Seção 3.1 são descritos textualmente os aspectos considerados para a recomendação no sistema. Enquanto a Seção 3.2 apresenta especificações de como foi desenvolvido o AG.

3.1 ASPECTOS CONSIDERADOS

Todas as informações que contribuem para a recomendação feita dentro da UniVirtual, se deve a partir da base de dados do Moodle, portanto, são utilizados os dados contidos nos cadastros de: cursos, usuários, locais virtuais e reais que se relacionam com os cursos, etc.

Para realizar uma recomendação de OAs adequada ao contexto do estudante é essencial levar em consideração características que podem influenciar durante a aprendizagem do mesmo. No Moodle encontra-se um repositório contendo uma certa quantidade de OAs. Estes OAs foram cadastrados de forma detalhada com uma série de características relacionadas a ele, auxiliando no casamento entre perfil de estudante e descrição do OA para as futuras recomendações. Na prática o professor é responsável por esse registro de OAs. Essa tarefa deve ser muito bem realizada, tendo em vista a grande dependência da descrição dos OAs para facilitar na sua seleção.

Na modelagem do presente problema de recomendação, são consideradas as características contidas nas próximas subseções.

3.1.1 Afinidade com o curso do Moodle

Dentre todas as características dos OAs contidos no repositório, comumente é possível não identificar o mínimo de relacionamento com alguns cursos ofertados pelo Moodle. Portanto, é de fundamental importância determinar este grau de afinidade para concretizar o quão relevante esse OA pode ser para um certo contexto. Desta forma, um peso é atribuído de acordo com uma escala que varia de 0, que implica na ausência de afinidade, a 7, que implica no grau máximo de afinidade.

No ato do cadastro dos OAs, o administrador do curso que está sendo ofertado é o responsável por definir, manualmente, o grau de afinidade desses OAs. Eles podem ser relacionados a um ou vários cursos.

3.1.2 Localização virtual

Uma outra informação levada em consideração para a definição do contexto é a Localização Virtual (LV) do estudante. Ou seja, no ambiente UniVirtual isso é determinado através do departamento relacionado a determinado curso do Moodle onde o estudante possa estar em um dado momento com o seu avatar. Isso implica em seu interesse, pois entende-se que se o mesmo entrar em um prédio que está atrelado ao curso de Ciência da Computação quer dizer que ele pretende encontrar OAs dessa área de estudo. Com a posse dessa informação, é possível verificar, nos metadados do padrão de OAs LOM, a incidência de palavras relacionadas à essa localização virtual do estudante.

Para a UniVirtual, essa é a informação mais relevante para a recomendação, assim como a afinidade com o curso, pois o princípio de pesos é o mesmo para ambas. No momento da recomendação, os metadados dos OAs são analisados em busca de informações que se relacionem com a localização virtual do estudante e os cursos que ele se encontra matriculado. Desta forma, quanto mais palavras relacionadas forem encontradas, maior será o valor atribuído à variável com escala de 0 a 7 (máximo). Caso não seja encontrada nenhuma informação relacionada, será atribuída a pontuação mínima (valor 0).

3.1.3 Localização real/corrente

A Localização Real (LR) do estudante, assim como a virtual, é levada em consideração para a construção do contexto. Desta vez trata-se da identificação geográfica em que o estudante se encontra fisicamente no corrente momento em que está sob o uso da UniVirtual no sistema Android, ou ativo no aplicativo móvel de GPS (4.3.3) responsável pela captura desta informação também no Android. Essa localização só é de fato armazenada na base de dados quando o estudante é identificado em uma área já cadastrada e devidamente relacionada a cursos do Moodle. Com a posse dessa informação, é possível verificar, nos metadados do padrão de OAs LOM, a incidência de palavras relacionadas a essa localização real do estudante.

Se comparado o grau de relevância entre a LV e a LR para recomendação de OAs, identifica-se que a LV tem maior prioridade devido à atribuição de peso à sua variável que vai até o valor 7, enquanto que a escala da LR varia de 0 a 5. A LV foi definida como prioritária com o propósito de assegurar, por exemplo, uma possível situação onde o estudante encontra-se fisicamente em um laboratório associado a um determinado curso, mas na UniVirtual se desloca com o avatar em busca de um departamento relacionado a outros cursos de áreas distintas as do laboratório que ele se encontra geograficamente falando. Então, neste exemplo, a LR tende a ficar em segundo plano.

3.1.4 Palayras-chave

Os OAs disponíveis no repositório do Moodle foram cadastrados seguindo o padrão SCORM. Eles possuem metadados que informam as palavras-chave relacionadas ao assunto daquele OA (*tag keywords*), além do título. Como já foi visto na seção sobre padrões de OAs, o padrão LOM é uma parte integrada ao SCORM, então, mais especificamente essas informações (metadados) estão localizadas no LOM.

Com a obtenção destas informações, é efetuada uma análise da incidência dessas palavraschave na descrição do curso do qual o estudante está participando. Quanto maior a quantidade de palavras-chave relacionadas, a probabilidade de que o OA possua assuntos relacionados ao curso será maior, sendo calculado, portanto, com o valor 4 multiplicado pelo número de palavras encontradas. Entretanto, quanto menor a quantidade de palavras relacionadas, o peso atribuído ao OA nessa variável torna-se proporcionalmente menor.

3.1.5 Dificuldade vs Nível do aluno

Este aspecto corresponde à dificuldade do aluno em relação ao assunto do OA. Para conseguir definir esse grau de dificuldade é levado em consideração o rendimento do aluno no curso. É importante citar que foi tomada como base a forma de avaliação da UFERSA onde, alunos com média abaixo de 3.6 são reprovados, entre 3.6 e 7.0 vão para a quarta prova, e caso a média seja igual a 7.0 ou superior o aluno é considerado aprovado na disciplina. Portanto, existem três status de dificuldade: *easy*, *medium* e *hard*; e três pesos que podem ser atribuídos a esses valores: 5, 3 e 1.

Existem três cenários para esta medição utilizando as notas dos alunos no Moodle, que variam de 0.00 a 100.00:

- 1. Caso a nota do aluno seja igual ou inferior ao valor 36.00 (equivalente a 3.6):
 - easy = 5
 - medium = 3
 - hard = 1
- 2. Caso a nota do aluno seja maior que o valor 36.00 (equivalente a 3.6) e menor ou igual a 70.00 (equivalente a 7.0):
 - easy = 3
 - medium = 5
 - hard = 3

- 3. Caso a nota do aluno seja maior que o valor 70.00 (equivalente a 7.0):
 - \bullet easy = 1
 - medium = 3
 - hard = 5

Conforme o primeiro status a maior probabilidade de oferta é de OAs considerados *easy*, já no segundo, tende mais para os OAs considerados *medium*, e por fim, no terceiro a maior chance é de recomendação de OAs definidos como *hard*. Notoriamente as maiores chances de recomendação vão para os OAs com variáveis das quais o maior peso lhes foram atribuídos.

3.1.6 Tipo de arquivo (preferência)

No cadastramento do estudante no Moodle é possível optar por preferências de tipos de arquivos dos OAs. Deste modo, são atribuídos pesos fixos (valor 5) as variáveis correspondentes aos tipos escolhidos. Na versão atual, são trabalhados três tipos de arquivos: pdf, mpeg e ppt. Então, logicamente ao escolher o tipo pdf, a variável receberá o valor 5. No ato da recomendação a probabilidade de seleção de OAs em pdf será maior que as demais.

3.1.7 Idioma

O idioma também é considerado no ato da recomendação. No cadastramento do estudante também é possível escolher o idioma de preferência. Com isso, a variável correspondente recebe um peso 5, em seguida, será considerado com maiores chances de ofertas, aqueles OAs que possuem o mesmo idioma da preferência do estudante.

3.2 DESCRIÇÃO DO PROJETO ALGORITMO GENÉTICO

O projeto do AG desenvolvido para a UniVirtual é composto por classes utilitárias capazes de capturar informações, e das classes que efetivamente realizam a abordagem genética. Este núcleo é composto basicamente por duas classes que asseguram os procedimentos característicos da metodologia de AGs. A Figura 5 ilustra um pequeno diagrama contendo essas classes.

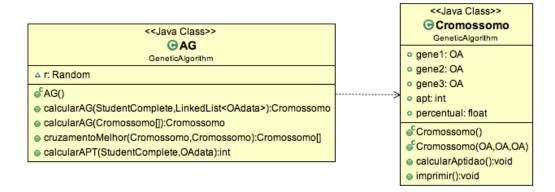


Figura 5 – Estrutura das classes do AG. Fonte: Autoria própria.

- 1. A classe *AG* representa o algoritmo genético propriamente dito, contemplando todos os parâmetros exigidos para assegurar as características dessa abordagem.
- 2. A classe *Cromossomo* representa as informações de cada um dos três OAs do repositório. Na UniVirtual sempre são recomendados três OAs, devido a isso encontra-se nesta classe a estrutura básica contendo três genes de um cromossomo.

A partir da classe AG são criadas várias instâncias da classe Cromossomo, inclusive, para inicializar a população. Mais detalhes nas Subseções 3.2.1 e 3.2.2.

3.2.1 Classe *AG*

Como já mencionado anteriormente, a classe AG representa os parâmetros necessários para a execução do AG. Abaixo os principais parâmetros:

- Gerações do AG: foi estabelecido um total de 50 gerações a cada execução do AG. Com essa quantidade de gerações foi possível obter-se resultados satisfatórios na recomendação, justificando a sua utilização.
- Dados do Estudante: este atributo é uma entidade denominada StudentComplete responsável por armazenar os dados do estudante contidos na base do Moodle. Sempre que necessário esse objeto é instanciado para possibilitar a sua manipulação (um dos agentes do SMA o manipula detalhes na Seção 4.6).
- Tamanho do cromossomo: o tamanho de cada cromossomo está estaticamente definido por três genes que representam os OAs. Entretanto o tamanho total do vetor que representa os cromossomos se dá por meio da divisão do tamanho da lista de OAs por 3, ou seja: OAlist.size() / 3.

Esta classe também dispõe de alguns métodos que auxiliam na execução do AG. Em destaque:

- calcularAG passando StudentComplete e LinkedList<OAdata> por parâmetros: é responsável por inicializar a população de cromossomos do AG obtendo a lista dos OAs e atribuir os pesos aos OAs de acordo com o perfil do estudante.
- calcularAPT: com base no cruzamento dos dados do perfil de estudante e dados dos OAs,
 esse método calcula a aptidão dos OAs e armazena o resultado de todas as avaliações.
- calcularAG passando Cromossomo por parâmetro: realiza a implementação da Roleta Viciada (LINDEN, 2012), também conhecida como técnica do giro da roleta, utilizando os dados das avaliações do calculo de aptidão.
- *cruzamentoMelhor*: este método efetua o cruzamento com um ponto de corte resultando em uma nova geração (LINDEN, 2012).

Vale salientar que a Roleta Viciada é um método de seleção de pais que consiste na criação de uma roleta virtual na qual cada indivíduo recebe um pedaço proporcional à sua avaliação de aptidão. Ou seja, para aqueles com alta aptidão é dada uma porção maior da roleta, enquanto que aos indivíduos com aptidão mais baixa é dada uma porção proporcionalmente menor. Em seguida, a roleta é "girada" e o selecionado será o indivíduo sobre o qual ela parar. Logo, os mais aptos possuem mais chances de serem escolhidos (LINDEN, 2012).

3.2.2 Classe Cromossomo

Na classe *Cromossomo* encontra-se a estrutura básica de um cromossomo que corresponde a três OAs (genes). A Figura 6 contém o código-fonte da classe *Cromossomo* para uma melhor visualização.

```
13
      public class Cromossomo {
14
           public OA gene1;
15
           public OA gene2;
           public OA gene3;
16
17
18
           public int apt;
19
           public float percentual;
20
21 🗆
           public Cromossomo(){}
22
   口
23
           public Cromossomo(OA g1, OA g2, OA g3){
24
               this.gene1 = g1;
25
               this.gene2 = g2;
26
               this.gene3 = g3;
27
28
29
   口
           public void calcularAptidao(){
           this.apt = this.gene1.aptidao+this.gene2.aptidao+this.gene3.aptidao;
30
31
32
33
   口
           public void imprimir(){
           System.out.println("Gene 1: "+this.gene1.nome +" Gene 2: "+this.gene2.nome
34
                  +" Gene 3: "+this.gene3.nome +" Aptidao: "+this.apt);
35
36
37
      3
```

Figura 6 – Código-fonte da classe *Cromossomo*. Fonte: Autoria própria.

As instâncias da classe *Cromossomo* são manipuladas pela classe *AG*. Como visto na Figura 6, trata-se de uma estrutura sem complexidade contendo apenas dois métodos:

- *calcularAptidao* (linha 29): trata-se de um método que calcula a aptidão do cromossomo através do somatório do resultado de aptidões dos OAs (genes).
- *imprimir* (linha 33): método com o propósito de auxiliar nas análises e testes. Irrelevante para o usuário, porém, de muita valia ao programador por exibir no console o valor da aptidão.

Apesar das duas últimas subseções apresentarem as classes fundamentais para o funcionamento do AG, é imprescindível que a classe utilitária *OAdata* seja descrita, pois ela é de extrema importância para a execução da classe *AG*. Portanto, a próxima subseção trata sobre este assunto.

3.2.3 Classe OAdata

Esta classe é responsável pela captura dos dados dos OAs contidos no repositório do Moodle. Na execução do algoritmo ela é instanciada pela classe *AG* para ser manipulada.

Com a explanação das classes *AG* e *Cromossomo* anteriormente, se torna possível a apresentação de um diagrama simples que contempla, desta vez além do núcleo do AG, classes extras como *OAdata*. A Figura 7 trata-se deste diagrama, onde a classe *OAdata* está destacada.

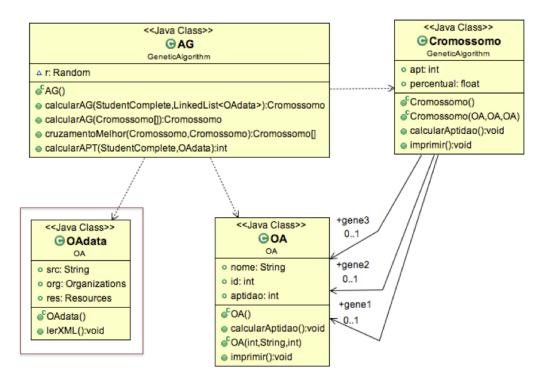


Figura 7 – Diagrama com algumas das principais classes do AG. Fonte: Autoria própria.

Este diagrama representa a comunicação entre as classes e apresenta as estruturas das mesmas. Fica clara, com sua visualização, a centralização na classe AG, pois nela todas as instâncias apontadas são manipuladas. O que a torna dependente das demais classes como, por exemplo, da classe OAdata. Assim, concretizando a fundamental importância dessas classes extras.

A classe *OAdata* ilustrada no diagrama da Figura 7 contempla três atributos, um construtor e um método. Em destaque o atributo *res* (instância de uma classe extra chamada *Resources*) e o método *lerXML*. Ambos descritos a seguir na respectiva ordem:

- *res*: este atributo recebe os dados dos OAs contidos no arquivo xml conforme as tags correspondentes por meio do método *lerXML*.
- lerXML: trata-se de um método responsável por uma tarefa indispensável que é a de coletar informações sobre toda a descrição dos OAs. Esta coleta é efetuada através da leitura ao arquivo xml relacionado a determinado OA. Cada dado é armazenado na variável res descrita anteriormente.

Assim como a classe Cromossomo, as instâncias da OAdata são manipuladas pela classe AG.

3.2.4 Passos do Algoritmo Genético da UniVirtual

As subseções anteriores trataram de descrever como encontram-se estruturadas as principais classes do projeto do AG e como elas se relacionam, assim como explicou-se os detalhes quanto aos objetivos específicos dos seus métodos.

Nesta subseção pretende-se listar os passos correntes no ato da execução do projeto de AG, com o objetivo de esclarecer, de forma superficial, as etapas efetuadas por meio das técnicas providas pelas classes já apresentadas nas últimas subseções.

Na UniVirtual o funcionamento do AG desenvolvido consiste em sete passos:

- 1. Carregar perfil do estudante;
- 2. Obter lista de OAs;
- 3. Atribuir os pesos dos OAs de acordo com o perfil carregado;
- 4. Inicializar população;
- 5. Selecionar OAs;
- 6. Substituir os piores pelos melhores enquanto a geração for menor que 50 (método da roleta viciada);
- 7. Selecionar o melhor cromossomo (que consiste em três OAs).

4 UNIVIRTUAL - AMBIENTE TRIDIMENSIONAL DE RECOMENDAÇÃO DE OBJETOS DE APRENDIZAGEM PARA APOIAR A APRENDIZAGEM UBÍQUA

Este capítulo descreve a UniVirtual, um ambiente 3D de aprendizagem ubíqua. Entretanto, na Seção 4.1 são citados, primeiramente, alguns trabalhos relacionados. Seguidamente, a Seção 4.2 apresenta as ferramentas utilizadas para o desenvolvimento da UniVirtual. A Seção 4.3 explica detalhes sobre as implementações. A Seção 4.4 traz uma descrição do ambiente virtual de apoio à aprendizagem ubíqua. A Seção 4.5 detalha a arquitetura completa do ambiente e cita os principais componentes do mesmo. A Seção 4.6 explica as funções de cada um dos agentes que compõem o sistema multiagente.

4.1 TRABALHOS RELACIONADOS

Os sistemas de recomendação têm sido muito utilizados nos últimos anos em propostas de soluções computacionais. Como exemplo, pode ser citado o trabalho de Silva (2012), que propõe um ambiente de suporte à aprendizagem móvel capaz de realizar a recomendação sensível ao contexto de OAs, definidos em um formato padrão, através do uso de ontologias e de um sistema multiagente. Este ambiente considera perfis de estudantes, localização dos mesmos, e escolhas similares de OAs.

No trabalho de Yau e Joy (2011), é apresentada uma arquitetura de aprendizagem móvel sensível ao contexto composta por três componentes principais: o perfil do estudante, um mecanismo de personalização e um repositório de OAs. O perfil do estudante serve para armazenar suas preferências de aprendizagem móvel, capturadas através de um questionário respondido pelo estudante. O mecanismo recebe esse perfil e combina as informações dele com aquelas obtidas dinamicamente através de sua interação com o ambiente. Seguidamente, são comparadas todas as informações obtidas com os metadados dos OAs disponíveis no repositório. O sistema então recomenda OAs adequados ao estudante com base nas características do seu contexto.

Wiedemann, Barbosa e Rigo (2013) oferecem um modelo para recomendação de OAs baseado na análise de similaridade entre históricos de consultas realizadas em repositórios. O modelo recebe a sequência de OAs consultados durante a sessão atual do usuário e localiza sessões cujas sequências de OAs consultados sejam similares à sequência da sessão corrente. Em seguida, os OAs encontrados nas sessões similares são então recomendados.

Para o desenvolvimento do presente trabalho foram reunidas as principais características dos trabalhos citados anteriormente. Entretanto, o que difere o presente trabalho dos demais é o desenvolvimento de um ambiente virtual tridimensional integrado a todas essas técnicas, possibilitando a recomendação autônoma e personalizada de OAs baseada no contexto que os estudantes se encontram em um dado momento através do auxílio de agentes de software.

4.2 FERRAMENTAS UTILIZADAS

Para o desenvolvimento da UniVirtual e para a sua composição arquitetural, foi necessária a utilização de algumas ferramentas, tais como: bibliotecas, frameworks, um software proprietário, bem como, um software proveniente de projeto *Open Source*. As principais ferramentas estão descritas nas próximas subseções contendo suas respectivas definições e relatos sobre a sua seleção para uso no projeto UniVirtual.

4.2.1 Game Engine Unity 3D

Para desenvolver o ambiente virtual 3D, foi de suma importância a escolha da ferramenta. Contudo, não foi uma tarefa simples, pois era necessário primeiramente analisar e testar a ferramenta para então tomar a decisão baseada na necessidade do projeto. Através de pesquisas, foi possível identificar uma ferramenta para o desenvolvimento de jogos 2D e 3D que apresenta uma popularidade muito grande atualmente, o Unity 3D (UNITY, 2014). Segundo Creighton (2010), o Unity consiste em um motor de jogo (*game engine*) ou em uma ferramenta para criação de jogos que permite pessoas criativas desenvolverem projetos ricos, e a curto prazo, devido o montante de recursos que a mesma dispõe.

Com o Unity 3D é possível criar ambientes virtuais a curto prazo, pois existem alguns componentes pré-prontos que facilitam esta atividade, e em seguida, exportá-los para diversas plataformas, como: Web (necessário o plugin do Unity), Android, IOS, OSX, Windows, etc. Os terrenos, por exemplo, são construídos e modelados através de recursos que a própria ferramenta dispõe. Outro ponto muito interessante é a possibilidade de importação de objetos 3D desenvolvidos por programas tradicionais deste seguimento, como é o caso do Blender (BLENDER, 2014). Estes atrativos do Unity despertaram o interesse em utilizar este motor para o desenvolvimento do ambiente UniVirtual.

Logo, foi decidido que seria necessário criar um projeto para testar a ferramenta. Então, inicialmente, foi desenvolvido um simples jogo de canhão para atingir alvos, onde o jogador necessitava inserir a massa da esfera que seria lançada pelo canhão e a sua força, para ajudar a alcançar estes alvos. A Figura 8 consiste na tela principal deste jogo.

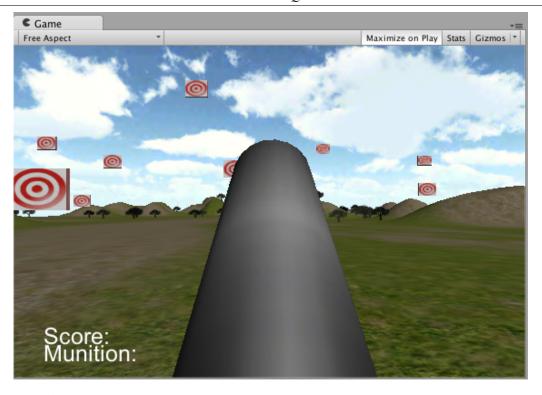


Figura 8 – Jogo de canhão criado na plataforma OSX. Fonte: Autoria própria.

Foram criadas novas versões deste jogo para Web, Android e Windows, com o intuito de testar o recurso do Unity de exportação para plataformas distintas. O resultado foi positivo, inclusive para dispositivos Android, pois a performance foi semelhante a das versões para *desktop*. Porém, ainda não havia sido o suficiente, tendo em vista que a intenção era percorrer um campus virtual universitário. Então foi criado um novo jogo e adicionado um personagem para se locomover neste. Nesta etapa foi testada a mudança de cenário e a performance do ambiente virtual instalado no servidor do Laboratório de Engenharia de Software (LES) da UFERSA. Portanto, para executar este ambiente foi necessário acessar o navegador da web com o *plug-in* do motor de jogo instalado, o "*Unity Web Player*".

Na Figura 9 encontra-se o jogo representando um ambiente virtual, e os seus dois cenários (A e B).



Figura 9 – Telas do segundo jogo com dois cenários diferentes. Fonte: Autoria própria.

Após este último experimento com a obtenção dos resultados satisfatórios, foi concretizada a utilização do Unity 3D para auxiliar no desenvolvimento da UniVirtual. Com esta ferramenta foi possível desenvolver o ambiente virtual acadêmico totalmente em 3D.

4.2.2 Character Creator Fuse

O Fuse é um software proprietário para criação de personagens 3D orientado a dados que vem com partes do corpo, roupas, malhas e texturas dinâmicas alimentados pela tecnologia *Substance Designer* (criação de texturas em alto nível) da empresa Allegorithmic (FUSE, 2014; ALLEGORITHMIC, 2014).

É possível utilizar o Fuse como um aplicativo independente para criar personagens únicos e exportá-los em T-pose, ou seja, ainda sem animações, ou usá-los com o Mixamo Autorigger para adicionar as animações. A Mixamo é a empresa proprietária do software Fuse, e ela dispõe deste recurso online de autorigger que consiste na aplicação automática de animações a personagens 3D carregados neste sistema web da Mixamo (MIXAMO, 2014).

A escolha do Fuse se deu pela necessidade da criação, em curto tempo, de personagens 3D, tendo em vista que a UniVirtual deveria conter NPCs espalhados pelos prédios do seu campus virtual, além do avatar principal que representaria o estudante. Como se tratava de uma proposta que envolvia muitas tecnologias, seria um risco efetuar essa tarefa de criação de personagens manualmente dentro do período de mestrado, por ser uma atividade que requer muitos processos além da modelagem 3D, como exemplo, a aplicação das animações o que demanda bastante tempo.

Após muitas pesquisas sobre softwares deste segmento, foi chegada a conclusão que o Fuse seria mais viável devido à sua ampla divulgação dentro da loja de *assets* (itens para projetos unity) da própria Unity, o que proporciona ao software uma certa credibilidade, de exemplos

convincentes expostos em vídeos mostrando o software em ação, além do custo acessível para o nível da ferramenta.

4.2.3 Plataforma JADE

Para o desenvolvimento dos agentes foi utilizado o JADE (*Java Agent Development Framework*), uma plataforma específica para o desenvolvimento e execução de sistemas multiagente (JADE, 2014). Este framework dá suporte ao desenvolvimento de sistemas multiagente e aplicações de acordo com os padrões FIPA para agentes inteligentes.

Ele inclui dois componentes principais: uma plataforma de agentes compatível com os padrões FIPA e um pacote para desenvolvimento de agentes. Além disso, ele também possui uma interface gráfica que pode ser utilizada durante as fases de desenvolvimento e de teste dos agentes (BELLIFEMINE; CAIRE; GREENWOOD, 2007; FONTES, 2013).

Em termos da cobertura dos padrões FIPA, o JADE implementa a especificação completa de gerenciamento de agentes, incluindo os serviços chave de AMS, DF, MTS e ACC. O JADE estende esses serviços com características adicionais, mas a conformidade com o padrão FIPA é mantida. O JADE também implementa, completamente, a pilha de comunicação de agentes FIPA, englobando desde a FIPA-ACL, para estrutura de mensagens, até o suporte para muitos dos protocolos de transporte e de interação FIPA (BELLIFEMINE; CAIRE; GREENWOOD, 2007; FONTES, 2013).

4.2.4 LMS Moodle

O *Modular Object-Oriented Dynamic Learning Environment* - Moodle, é um LMS (*Learning Management Systems*) ¹ desenvolvido com a linguagem PHP (PHP, 2014) e que utiliza um banco de dados MySQL (MYSQL, 2014). É distribuído livremente sobre a licença GNU GPL (*General Public License*) (GPL, 2007) e pode ser utilizado por aqueles que desejam disponibilizar cursos a distância através da Web (MOODLE, 2014).

Esta ferramenta faz parte da arquitetura da UniVirtual descrita em detalhes na Seção 4.5. O Moodle é de suma importância para o funcionamento do ambiente virtual 3D, pois as requisições e consultas a dados de estudantes são feitas exclusivamente na sua base de dados, inclusive, o repositório dos OAs também se encontram neste mesmo banco.

No português "Sistema de Gestão da Aprendizagem" (SGA), são softwares desenvolvidos sobre uma metodologia pedagógica para auxiliar a promoção de ensino e aprendizagem virtual ou semi-presencial (CARVALHO, 2010).

4.3 IMPLEMENTAÇÕES DO AMBIENTE UNIVIRTUAL

Esta seção relata sobre as implementações dos principais componentes que completam o ambiente UniVirtual. A Subseção 4.3.1 fala sobre o desenvolvimento do ambiente 3D. A Subseção 4.3.2 apresenta a implementação dos personagens 3D. No fim, a Subseção 4.3.3 descreve os principais detalhes do projeto de captura de localização real do estudante.

4.3.1 Implementação do ambiente 3D

Após a decisão de usar o Unity foi dado início à implementação do projeto em 3D. Inicialmente foram esboçados alguns protótipos de campus universitários e definida a quantidade de laboratórios que este teria. Com alguns dias foi possível desenvolver uma planta simples contendo: pavimentação asfáltica, calçadas, sete prédios, um estacionamento e os espaços arborizados. Baseado nesse protótipo foi possível chegar ao cenário da Figura 10.



Figura 10 – Vista aérea do ambiente 3D. Fonte: Autoria própria.

Esta imagem foi capturada a partir do Unity, trata-se da vista aérea do cenário principal da UniVirtual. Alguns objetos 3D, tais como: os prédios da parte externa, cadeiras e computadores que ficam nas cenas correspondentes à parte interna dos laboratórios, foram fornecidos através de uma parceria com os membros do projeto UFERSA Virtual (SOUZA D. F. L.; CASTRO, 2013).

Posteriormente foram criadas as partes internas dos laboratórios. Esta etapa demandou mais tempo devido os detalhes do cenário. Por exemplo, alguns objetos que foram criados

manualmente, o posicionamento destes nas cenas, bem como a aplicação de suas texturas, etc. Para seguir o padrão, os laboratórios (parte interna) foram desenvolvidos com base nos prédios do cenário principal (externo). A Figura 11 exibe o interior de um dos laboratórios.



Figura 11 – Interior de um dos laboratórios. Fonte: Autoria própria.

A ferramenta Unity colaborou bastante para a realização desta atividade, pois o método de criação dos mapas, ou seja, a forma de construção dos terrenos e a aplicação das texturas, é bastante simples e eficiente. Outro ponto positivo da ferramenta está relacionado à receptividade de objetos 3D desenvolvidos por terceiros, pois foram importados alguns componentes criados no *Blender*, por exemplo, e a compatibilidade em termos visual foi muito boa.

Ainda foi possível, com o Unity, fazer algumas modelagens simples, tais como: os arcondicionados e as lousas dos laboratórios, vistas na Figura 11 mostrada a pouco; algumas placas na entrada dos laboratórios e nos interiores, indicando o departamento, e a placa de saudações que pode ser vista na Figura 19 da Seção 4.4.

4.3.2 Implementação dos personagens 3D

Com a conclusão da criação de todas as cenas do projeto com o Unity, foi iniciada a criação dos personagem 3D. Para o desenvolvimento dessa etapa foram utilizados o Fuse, o sistema online da *Mixamo* e o Unity novamente. Primeiramente foi criado o avatar que consiste em um estudante, depois os NPCs que representam os professores em cada laboratório. Esta tarefa exige três passos básicos:

- 1. Criação do personagem em 3D (Fuse);
- 2. Aplicação da animação ao personagem (Sistema Auto-Rigger da Mixamo);

3. Aplicação dos *scripts* para utilizar as animações (Unity).

O Fuse proporciona um meio bastante prático de criação de personagens. Esta ferramenta permite uma montagem do objeto 3D por partes do corpo, através de um banco que disponibiliza essas partes pré-prontas. Primeiramente o usuário escolhe a cabeça, em seguida o corpo, depois os braços, pernas e roupas. No final é permitido efetuar algumas customizações em texturas, nas expressões faciais etc. Uma das etapas na montagem de um dos NPCs utilizados na UniVirtual encontra-se exemplificado na Figura 12.

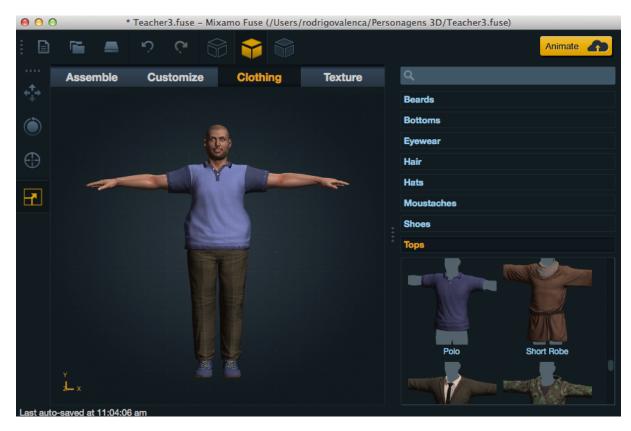


Figura 12 – Etapa de criação do personagem 3D. Fonte: Autoria própria.

Com a concretização do personagem 3D, se faz necessária a aplicação de animações para proporcionar realidade virtual. Neste processo entra a utilização do sistema online da empresa Mixamo proprietária do Fuse.

Um dado interessante é que na modelagem de um personagem 3D é imprescindível a criação do seu esqueleto para auxiliar nos movimentos de todas as partes do corpo, após isso, aplicam-se as animações. Geralmente esse processo é feito em paralelo com o desenho de todo o corpo. Porém, apesar deste processo ser considerado minucioso e trabalhoso, com o *auto-rigger*, o usuário poderá apenas solicitar a aplicação do esqueleto e marcar alguns pontos do corpo conforme a Figura 13.

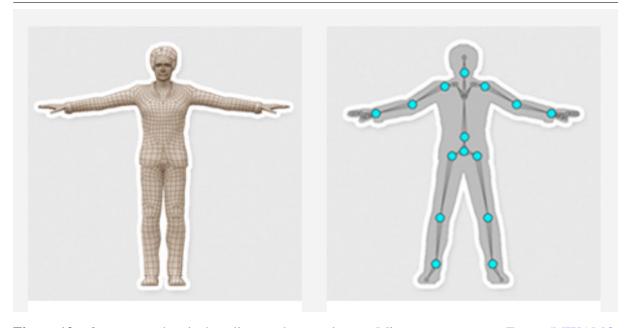


Figura 13 – O antes e o depois da aplicação do esqueleto no Mixamo *auto-rigger*. Fonte: (MIXAMO, 2014)

Agora o personagem está apto a ser equipado com animações. No site da Mixamo contém um número significativo de animações gratuitas para serem aplicadas. Comumente utilizam-se as seguintes expressões relacionadas as animações básicas: *idle*, *walking*, *right turn* e *left turn*; respectivamente: parado, andando, vira a direita e vira a esquerda. Todos os personagens da UniVirtual receberam essas animações básicas. Por fim, foram exportados em formato fbx reconhecido pelo Unity.

Uma vez concluída a criação das partes fundamentais dos personagens, chega-se a necessidade da implementação da lógica de funcionamento destes avatares agora dentro da UniVirtual. A ideia é empregar comandos capazes de proporcionar ações conforme o ambiente. Como principais ações podem ser citadas: mover o personagem estudante conforme determinados botões do teclado, mover aleatoriamente os NPCs, fazer com que os NPCs parem ao detectar o personagem estudante dentro de um determinado raio de distância e a partir de uma aproximação siga o avatar do estudante até que ele ultrapasse essa aproximação, etc.

Então foram importados todos os personagens para o Unity, em seguida, desenvolvidos *scripts* específicos para prover as funcionalidades destacadas no parágrafo anterior conforme cada perfil de personagem. Estes *scripts* foram desenvolvidos na linguagem CSharp. Como exemplo, a Figura 14 exibe um trecho do código referente à ação de parada do NPC ao identificar a aproximação do personagem estudante.

```
Vector3 PosiRela = Player.position - transform.position;
164
165
166
167 🗀
               if(PosiRela.magnitude <= 2)</pre>
168
                    anim.SetFloat("Speed", 0f, 0f, 0f);
169
                    anim.SetFloat("Turn", 0f, 0f, 0f);
anim.SetFloat("Direction", 0f, 0f, 0f);
170
171
172
                    windowRect = GUI.Window(0, windowRect, DoMyWindow, "Teacher / Student Dialogs:", myBoxStyle);
173
174
               }
175
```

Figura 14 – Código-fonte referente à parada do NPC para interagir com o personagem estudante. Fonte: Autoria própria.

A variável *PosiRela* do tipo *Vector3* (API do Unity) (linha 164) recebe a posição corrente do estudante (personagem 3D) dentro do ambiente virtual, em seguida (linha 167) é testado se ele se encontra a uma determinada distância (valor 2) entre a origem do vetor e o seu ponto de extremidade, caso essa condição seja satisfeita serão setados zeros nas seguintes animações do NPC: *Speed, Turn* e *Direction*. Isso fará com que o NPC pare diante do personagem estudante e apresente uma tela de diálogo (linha 173).

A descrição deste trecho de código-fonte da figura anterior é apenas uma amostra. Seria inviável tentar inserir todos os *scripts* desenvolvidos aqui neste trabalho, tendo em vista que são muitos. Portanto, a intenção é repassar para o leitor uma ideia do processo de como tudo foi desenvolvido.

4.3.3 Implementação do projeto de captura da localização real

Este projeto pode ser classificado como um componente para o ambiente UniVirtual, pois ele foi desenvolvido com o propósito de capturar a localização real do estudante através do GPS, no ato da autenticação no ambiente. Logicamente este componente é utilizado apenas na versão da UniVirtual para dispositivos móveis com sistemas Android, devido à necessidade do GPS.

Para o desenvolvimento deste projeto foi utilizada a plataforma Android SDK que apresenta características importantes, como, por exemplo, ser um software livre, aceitar uso de várias APIs, uso da linguagem de programação Java, etc (FRANCISCATO; MEDINA, 2008; ANDROID, 2014). E para se ter um ambiente de desenvolvimento foi necessária a utilização da Ide Eclipse (2014) com o plugin *Android Development Tools* (ADT).

A localização real do estudante é efetuada com o auxílio das classes *LocationManager*, responsável por fornecer acesso aos serviços de localização geográfica, e *Location*, que pode representar essa localização através das seguintes informações: latitude, longitude, data e hora, entre outras. É importante lembrar que o sistema necessita do cadastramento dos laboratórios ou departamentos para poder ser feita a comparação com os dados do local do estudante. Portanto, para este cadastramento de lugar é necessário utilizar as mesmas classes.

No trecho de código-fonte da Figura 15 é possível ver instâncias das classes *Locati-onManager* e *Location*, que foram criadas para apanhar seus respectivos dados referentes à localização.

```
59
           public void onClick(View v) {
60
                IniciarGPS();
61
                Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
62
                Double Latitude = location.getLatitude();
63
                Double Longitude = location.getLongitude();
               Double Altitude = location.getAltitude();
64
65
                edtCadLocalLatitude.setText(Latitude.toString());
                edtCadLocalAltitude.setText(Altitude.toString());
67
                edtCadLocalLongitude.setText(Longitude.toString());
68
```

Figura 15 – Trecho do código que armazena dados sobre localização geográfica.

No início do código é invocado o método *IniciarGPS* (linha 60) que ativa o GPS do dispositivo caso o mesmo esteja desligado, seguidamente uma variável denominada *location* (tipo *Location*) recebe parâmetros para dar início ao serviço de localização geográfica, dando assim autonomia para que seja possível a captura dos dados: latitude, longitude e altitude, através dos métodos *gets* (linhas 62, 63, 64) da classe *Location*.

Com a intenção de evitar que qualquer pessoa cadastrasse lugares, o que poderia gerar conflitos de localização, foram definidas duas restrições: é necessário ser professor e estar cadastrado na UniVirtual, para poder executar esta tarefa. Portanto, baseando-se nestas restrições existem três cenários de uso do aplicativo de localização real descritos a seguir:

- Professor abre o aplicativo independente da UniVirtual: mesmo assim ele deve fazer a
 autenticação com os mesmos dados de usuário que ele utilizar na UniVirtual, após isto
 será verificado se o local que o professor se encontra no momento já está cadastrado, caso
 não esteja é aberta uma tela para cadastramento de local, do contrário o aplicativo ficará
 rodando em segundo plano;
- 2. Professor abre o aplicativo pela UniVirtual: neste caso o aplicativo é acionado apenas no ato da autenticação da UniVirtual executando diretamente o serviço que testa se o lugar já está cadastrado ou não (igual a situação anterior);

A Figura 16 contém telas do aplicativo de localização, simulando as situações 1 e 2 em que o usuário é o professor e quando o local ainda não foi cadastrado. Na tela [1] os dados ainda não foram capturados, já na tela [2] o usuário pressionou o botão *obter coord*, então os campos foram preenchidos com as informações geográficas de localização do dispositivo.

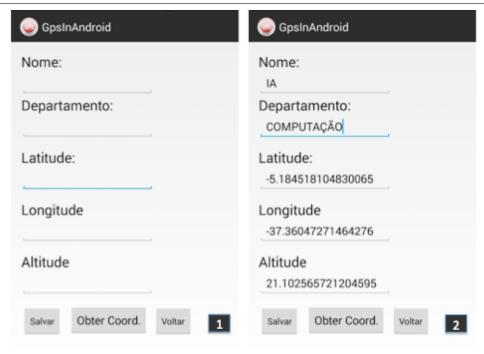


Figura 16 – Telas de cadastro de departamento via GPS. Fonte: Autoria própria.

3. Estudante: em ambos os casos, seja abrindo diretamente o aplicativo ou pela UniVirtual, o estudante estará limitado a apenas o serviço de verificação de local. Então se o local estiver cadastrado o estudante será reconhecido e um registro será efetuado na base de dados do Moodle em uma tabela específica. Porém, se o lugar não estiver cadastrado o serviço é ignorado e não acontecerá nada.

Quando identificada a proximidade do estudante a um local cadastrado, esta será uma informação bastante influente para o algoritmo genético que a utilizará no momento da seleção de OAs para a recomendação. Em momentos que o estudante esteja fisicamente (localização real) em um determinado departamento e virtualmente (na UniVirtual) em um departamento da mesma área, as chances de recomendações de OAs que se relacionam com esta área de estudo dos locais correntes serão ainda maiores que dos demais OAs.

Devido à extensão do código da classe *VerificandoDistancia* responsável pela verificação da proximidade do estudante, a mesma está disponível no Apêndice A.

4.4 DESCRIÇÃO DA UNIVIRTUAL

A UniVirtual é uma arquitetura multiagente integrada ao LMS Moodle através de um banco de dados. Essa arquitetura ainda contém uma interface gráfica tridimensional acessível na plataforma Android e em *desktops* através de navegadores web com o plugin *Unity Web Player*. Este ambiente 3D permite que o estudante matriculado em cursos do Moodle receba conteúdos

personalizados com base no seu perfil de usuário e sua localidade geográfica e virtual e, a partir de qualquer lugar (com a versão *mobile*) e horário.

Como pré-requisito para a entrada neste ambiente, os estudantes precisam, inicialmente, estarem cadastrados no LMS Moodle. Neste cadastro são fornecidas informações bastante relevantes que caracterizam o seu perfil e que servirão futuramente para a criação de um determinado contexto. Apenas após essa etapa é possível se autenticar à UniVirtual.

A comunicação com a base de dados do Moodle no ato da autenticação é efetuada a partir de *scripts* PHP executados pela UniVirtual. Já a atuação dos agentes funciona através do SMA acionado por mensagens, e que necessita dessa mesma base para acessar os metadados do estudante. Esse quadro básico de comunicação inicial entre os componentes encontra-se esboçado na Figura 17.

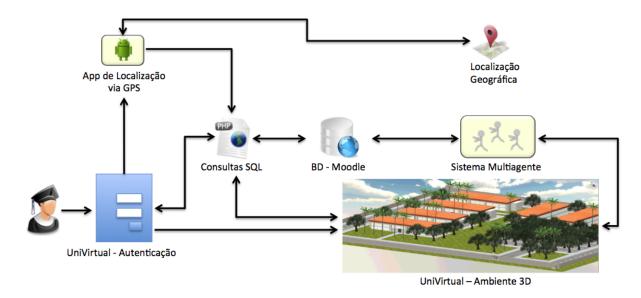


Figura 17 – Comunicação inicial entre os componentes. Fonte: Autoria própria.

Caso o estudante esteja em um dispositivo móvel com Android, ainda no ato da autenticação, será executado o aplicativo de localização via GPS em segundo plano, para identificar se o estudante encontra-se próximo a algum laboratório cadastrado no sistema, se ele estiver, essa informação será armazenada no banco de dados novamente através de *scripts* PHP, para somar ainda mais na criação do contexto deste estudante. Se ele não estiver próximo a um laboratório cadastrado, o sistema de localização é simplesmente ignorado.

A UniVirtual deve ser utilizada juntamente com o Moodle (MOODLE, 2014), pois ela funciona como um componente extra que deve ser integrado à infraestrutura deste LMS. No momento de inscrição no sistema Moodle, o estudante deve fornecer algumas informações que serão úteis para a definição do seu contexto, dentre as quais pode-se citar: idioma, tipo de arquivo de preferência, curso, entre outras. A Figura 18 exibe a interface do Moodle adaptada para o cadastramento dessas informações sobre o estudante.

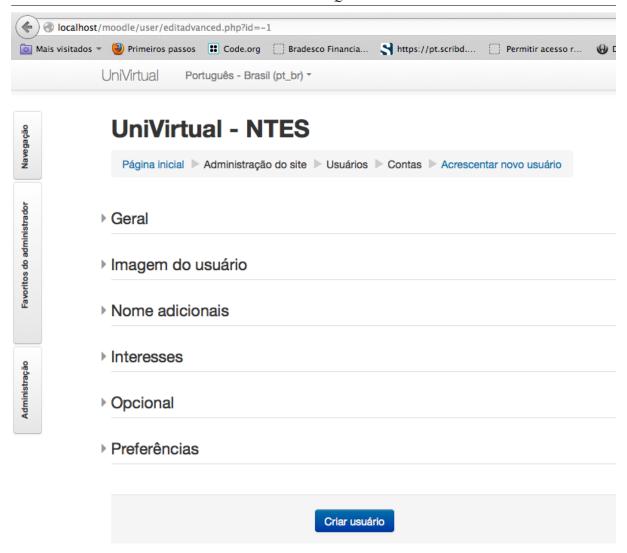


Figura 18 – Tela de cadastro de estudantes no Moodle. Fonte: Autoria própria.

Na parte correspondente às preferências do estudante encontra-se a opção de criação de uma senha específica para a UniVirtual. Após o cadastramento de todos estes dados, inclusive da senha, o estudante estará apto a utilizar o ambiente. Na Figura 19 estão ilustrados dois cenários. Primeiramente a tela de autenticação [1], da qual o estudante fornecerá o mesmo usuário do Moodle e a senha que ele cadastrou para a UniVirtual, em seguida, o cenário inicial do ambiente UniVirtual [2] carregado após a autenticação.

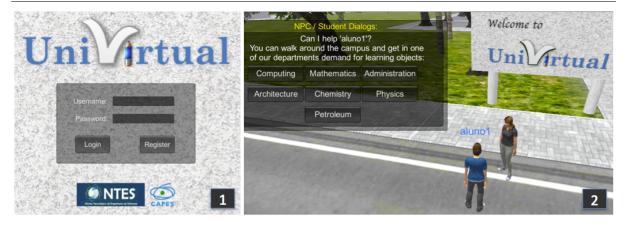


Figura 19 – Tela de autenticação [1] e cenário inicial da UniVirtual [2]. Fonte: Autoria própria.

Ao carregar a cena inicial o estudante terá posse do controle de locomoção do avatar que o representará neste universo digital. Este controle atende aos comandos tradicionais do teclado: W = frente, A = esquerda, D = direita, Q = girar à esquerda, e E = girar à direita, no caso do uso em computadores, caso o usuário esteja em um dispositivo móvel irá aparecer o *joystick* virtual que atende ao toque conforme a Figura 20.



Figura 20 – UniVirtual no Android. Fonte: Autoria própria.

Todo o ambiente virtual contém ao menos um NPC em cada cena. Nesta cena inicial, por exemplo, o estudante poderá se deparar com uma personagem virtual que orienta-o quanto aos prédios e seus respectivos cursos. Já os demais NPCs encontrados no interior de cada prédio são responsáveis pela recomendação dos OAs. Eles identificam a proximidade do estudante e, com auxílio do SMA, ofertam os OAs mais apropriados com base no contexto atual deste estudante.

Se faz necessário destacar que o foco principal do uso da UniVirtual é que a mesma funcione como um recurso adicional de suporte à aprendizagem do estudante de maneira flexível

e estimulante ao mesmo tempo. Portanto, vai além da recomendação de conteúdo sensível ao contexto.

Outro ponto importante de ressaltar é que outro objetivo do presente trabalho consiste na disponibilidade de um ambiente altamente interativo e em 3D que funcione, principalmente em dispositivos móveis, e que proporcione uma infraestrutura necessária para a recomendação adequada de OAs.

4.5 ARQUITETURA DO AMBIENTE

A arquitetura da UniVirtual encontra-se ilustrada na Figura 21 contendo os seus principais componentes e as suas relações.

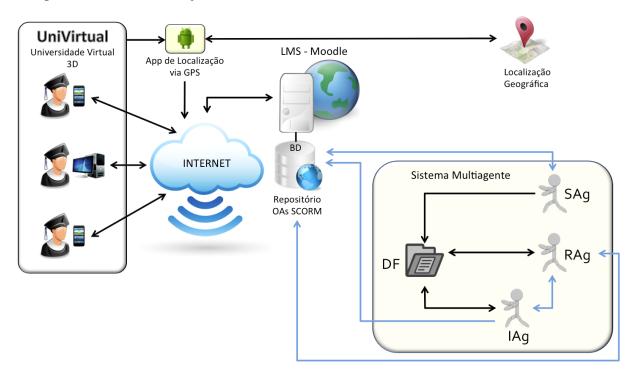


Figura 21 – Arquitetura da UniVirtual. Fonte: Autoria própria.

Abaixo a listagem dos principais componentes que compõem essa arquitetura:

- 1. Sistema multiagente (SMA): Desenvolvido para proporcionar uma maior autonomia ao sistema e prover a recomendação de OAs de forma automática e personalizada. Para o auxílio na atividade de seleção de OAs, neste SMA encontra-se embutido o AG.
- 2. LMS Moodle: O Moodle é imprescindível para o funcionamento da UniVirtual, pois ambos se comunicam. A autenticação do estudante na UniVirtual, por exemplo, é feita a partir do cadastro prévio no Moodle, assim como o repositório de OAs que se encontra na base de dados desse LMS. Sua relevância neste projeto também está atrelada a oferta dos cursos a distância, deste modo, sem esse componente seria inviável.

- 3. Repositório de OAs SCORM: Este é o repositório utilizado para armazenamento dos OAs. Ele fica localizado na base de dados do Moodle. Seus OAs devem ter sido desenvolvidos seguindo o padrão SCORM. Através das informações fornecidas no SCORM e dos metadados do padrão LOM, que é parte integrante do SCORM, os agentes serão capazes de comparar as informações dos OAs com os perfis dos estudantes.
- 4. UniVirtual: Consiste em um ambiente 3D interativo com NPCs que representam os agentes de software. É nele que ocorre a oferta de OAs. Este sistema está inteiramente ligado ao banco do Moodle para prover as suas funcionalidades.

4.6 SISTEMA MULTIAGENTE

No trabalho aqui apresentado foi necessária a implementação de três tipos de agentes que comporão o SMA. A organização deste SMA é do tipo comunidade de especialistas, com base em Artero (2009), visto que cada um dos tipos de agentes criados possuem o mesmo nível, entretanto, cada um deles é especialista em determinada tarefa. Foram utilizadas as bibliotecas do JADE para a criação dos seguintes agentes:

- Agente Estudante (SAg Student Agent);
- Agente de Interface (IA Interface Agent);
- Agente Recomendador (RAg *Recommender Agent*).

As próximas subseções tratam, primeiramente, da modelagem geral do SMA, em seguida, do modelo de cada um dos agentes que estão pré-definidos na Subseção 4.6.1.

4.6.1 Modelagem do SMA

Para a modelagem do SMA da UniVirtual foi utilizada a metodologia de modelagem orientada a agentes MAS-COMMONKADS+ (MORAIS, 2010). A modelagem da arquitetura do SMA é fundamental para compreensão das atividades que esse deve prover ao ambiente UniVirtual. Porém, nem todos os diagramas dos respectivos modelos são necessários para a compreensão da arquitetura multiagente dessa abordagem, visto que alguns deles acabam fornecendo informações redundantes. Logo, decidiu-se representar cada modelo através de um diagrama, contribuindo assim para a compreensão do SMA. A escolha dos modelos a serem inseridos na modelagem depende do tamanho e da complexidade do SMA sendo modelado (SILVA, 2012; FONTES, 2013).

Com base nas necessidades da UniVirtual, o SMA deve realizar uma série de tarefas específicas. Para isto o Modelo de Tarefas exibido na Figura 22 foi desenvolvido para auxiliar na delimitação das responsabilidades de cada agente.

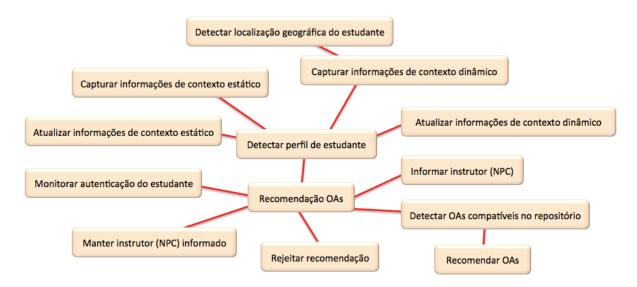
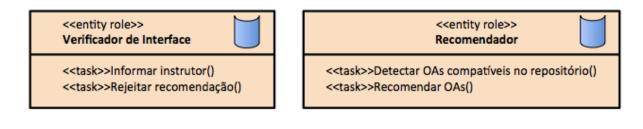


Figura 22 – Modelo de tarefas do SMA. Fonte: Autoria própria.

O objetivo do SMA da UniVirtual é a recomendação de OAs, logo esta pode ser considerada a principal tarefa, tendo portanto que ser decomposta em subtarefas que irão ajudar na realização da tarefa principal. Com base nesse contexto foram definidos os papéis e seus agentes responsáveis no SMA. Cada papel pode realizar várias tarefas exibidas em um Modelo de Tarefas. Portanto, com base nas tarefas do Modelo de Tarefas, foram elencados os seguintes papéis presentes no Modelo de Papéis na Figura 23.



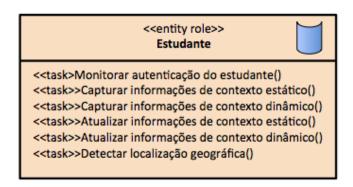


Figura 23 – Modelo de Papéis do SMA. Fonte: Autoria própria.

Baseando-se pela Figura 23, fica claro que os papéis podem ser responsáveis por várias tarefas. Existem agora três tipos de papéis, são eles: estudantes, recomendador e verificador de interface. Estes três papéis foram alocados, respectivamente, aos seguintes agentes: Agente Estudante, Agente Recomendador e Agente de Interface.

Com a definição desses papéis explícitos no Modelo de Papéis, se torna possível a elaboração de um Modelo de Organização para descrever a estrutura organizacional dos papéis no sistema, ou seja, como eles se relacionam. Este modelo está destacado na Figura 24.

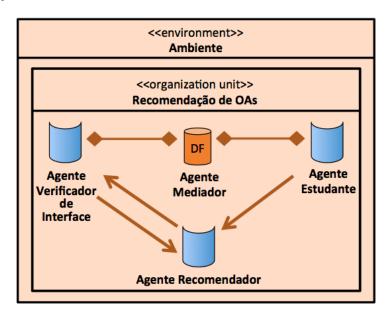


Figura 24 – Modelo de Organização do SMA. Fonte: Autoria própria.

Este modelo permite a percepção de um outro componente, o Agente Mediador que consiste no Quadro Negro (*Directory Facilitator*) (ARTERO, 2009) que é utilizado na arquitetura da UniVirtual. Um outro ponto notório é a relação do Agente Mediador que está a cooperar apenas entre os agentes IAg e SAg. O objetivo deste Agente Mediador é prover a comunicação entre esses dois agentes mesmo que, inicialmente, eles não se conheçam. O motivo pelo qual este componente não aparece no Modelo de Papéis é que o mesmo não realiza tarefas que compõem o Modelo de Tarefas. Na verdade, ele é um intermediador, que funciona centralizando as informações a respeito dos serviços dos agentes, de modo a permitir uma comunicação através de registros e pesquisas feitas por esses agentes.

Com a elaboração do Modelo de Organização há a possibilidade de visualizar a organização interna e externa dos papéis. Entretanto, o SMA deste trabalho possui apenas uma unidade organizacional, logo não existe uma representação externa.

Ainda sobre o Modelo de Organização da Figura 24, é importante frisar que o RAg representado pelo Agente Recomendador, apesar de não estar em contato com o Agente Mediador, ele se relaciona diretamente com os demais agentes. Essa relação será mais detalhada nas próximas subseções que tratam sobre cada agente.

Seguindo com os passos da modelagem do SMA, com a absorção de noção estática alcançada com o Modelo de Organização, chega o momento de se ter uma visão do comportamento

dinâmico do SMA. Neste caso a Figura 25 mostra um modelo de interação entre os agentes que compõem este sistema proposto para ajudar na compreensão do SMA.

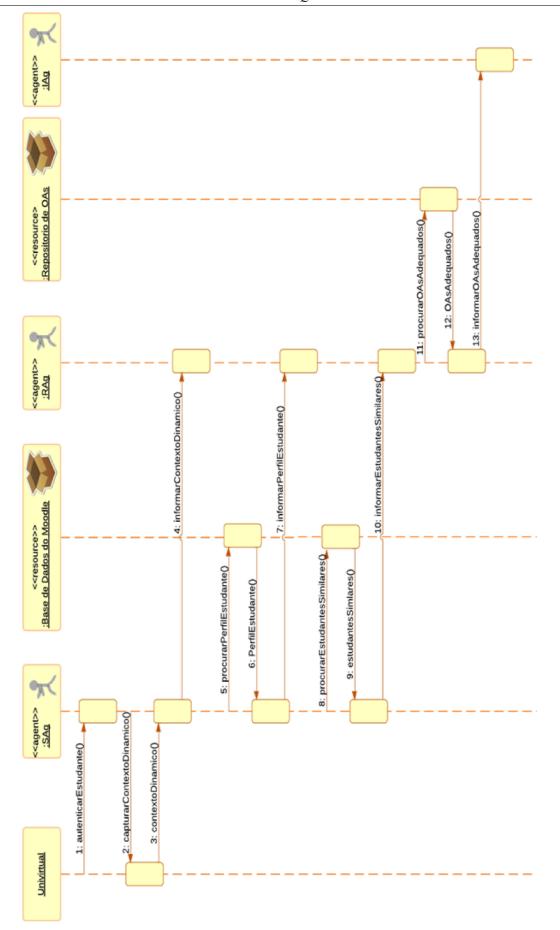


Figura 25 – Modelo de Interação dos agentes. Fonte: Autoria própria.

O modelo da Figura 25 permite a visualização do relacionamento dos agentes e dos recursos que são requisitados por eles. Neste diagrama encontra-se desde o comportamento do SAg no momento de autenticação do estudante na UniVirtual até a informação chegar no IAg, que trata de realizar ou rejeitar a recomendação.

4.6.2 Modelo do Agente Estudante - SAg

A responsabilidade dos SAgs consiste no monitoramento das atividades dos estudantes e recuperação das suas preferências, obtidas através dos seus perfis de usuários. O seu comportamento é realizado a partir da autenticação do estudante na UniVirtual. Estes agentes são capazes de capturar as informações do contexto dinâmico do estudante, além de sua localização geográfica já armazenada na base de dados através do aplicativo de localização real. Por fim, todas essas informações são disponibilizadas para o RAg por meio da base de dados.

A Tabela 3 mostra o seu template textual.

Tabela 3 – *Template* do Agente Estudante

AGENTE: SAg

Objetivo: Monitorar as informações do estudante, capturando o contexto dinâmico do mesmo, e registrando na base de dados.

Parâmetros de entrada: Identificador do estudante no sistema.

Parâmetros de saída: Dados relativos ao contexto do estudante.

Condição de Ativação: Quando o agente percebe que um estudante se autenticou no LMS ou na UniVirtual.

Condição de Finalização: Quando o agente se comunica com o RAg e envia para este informações sobre o contexto do estudante.

Descrição: Este agente possui o comportamento totalmente voltado para o monitoramento das atividades dos estudantes, recuperando e atualizando na base de dados as informações de preferências de aprendizagem que compõem os perfis dos estudantes, e a localização geográfica.

4.6.3 Modelo do Agente Recomendador - RAg

Por sua vez o RAg, de acordo com as informações disponibilizadas pelos SAgs e as obtidas dos OAs disponíveis no repositório, detecta os OAs que correspondem ao contexto atual do estudante. Ele se responsabiliza em encontrar o OA mais adequado de acordo com a

Recomendação Baseada em Conteúdo. Depois de selecionar os OAs adequados, o RAg repassa estes para o IAg.

A Tabela 4 mostra o seu template textual.

Tabela 4 – *Template* do Agente Recomendador

AGENTE: RAg

Objetivo: Receber as informações enviadas pelo SAg, aplicar o algoritmo genético e enviar resultados para o IAg.

Parâmetros de entrada: Informações sobre os estudantes enviadas pelo SAg, e informações dos OAs disponíveis no repositório de OAs SCORM.

Parâmetros de saída: Possíveis OAs adequados ao contexto do estudante.

Condição de Ativação: Quando o SAg reconhece uma autenticação na UniVirtual.

Condição de Finalização: Quando o agente se comunica com o Agente de Interface e envia para este as informações do OA a ser recomendado e para qual estudante.

Descrição: Este agente possui o comportamento voltado para a recomendação de OAs. É responsável por detectar e realizar a recomendação de OAs.

4.6.4 Modelo do Agente de Interface - IAg

O IAg recebe os OAs recomendados pelo RAg, em seguida, armazena-os na base de dados. Ao disponibilizar os OAs na base de dados o tutor virtual (NPC) consulta o conteúdo que foi sugerido e oferece de fato ao usuário estudante na UniVirtual. A Figura 26 apresenta o cenário correspondente com o momento da apresentação dos OAs, ação espontânea efetuada pelo NPC que captura os OAs enviados pelo IAg.

A Tabela 5 mostra o template textual do agente IAg, descrevendo suas características.

Tabela 5 – *Template* do Agente de Interface

AGENTE: IAg

Objetivo: Uma vez em posse das informações enviadas pelo RAg, o IAg realiza a sua atualização na base de dados, para o NPC fazer a consulta futuramente.

Parâmetros de entrada: Informações sobre a recomendação de OAs enviadas pelo RAg.

Parâmetros de saída: Dados sobre os OAs recomendados inseridos na base de dados.

Condição de Ativação: Quando o agente recebe, por parâmetros, os dados necessários do RAg.

Condição de Finalização: Quando confirmada a saída do usuário estudante da UniVirtual.

Descrição: Este agente possui o comportamento voltado ao repasse de OAs para que estes sejam disponibilizados para os NPCs.

4.6.4.1 Oferta dos OAs efetuada pelo NPC

Componentes não apresentados na modelagem do SMA por não fazerem parte do projeto deste sistema multiagente, mas que fazem parte do processo de recomendação, são os NPCs. Eles agem com o auxílio da base de dados do LME Moodle. Possuem características de agentes inteligentes por se comportarem com autonomia dentro da UniVirtual e reagirem conforme algum contexto.

Estes agentes virtuais percorrem um cenário da UniVirtual de forma aleatória e ao se depararem próximos do avatar estudante suas ações mudam. Dentro de um certo espaço de proximidade do avatar estudante, os NPCs param e, em segundo plano, consultam as informações disponibilizadas pelo IAg.

Com esses dados capturados, são disponibilizados na tela três opções de OAs. Caso o estudante mova o avatar, este NPC acompanhará o estudante virtual até que ele ultrapasse o limite de proximidade, e aí ele volta a se locomover aleatoriamente. Na Figura 26 pode-se ver o personagem que representa o estudante autenticado prestes a entrar no departamento de computação e, em seguida, o momento em que ocorre a recomendação de OAs.

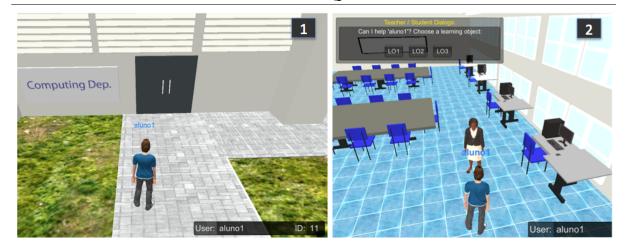


Figura 26 – Cenário inicial[1] e sugestão de OAs[2]. Fonte: Autoria própria.

Quando o estudante clicar em um botão de OAs sugeridos, o NPC se responsabilizará por invocar, no diretório correspondente ao repositório SCORM dos OAs, o OA correspondente ao botão acionado. Ainda por meio do NPC, será executado o arquivo (pdf, mpeg ou ppt) fora da UniVirtual, que poderá ser aberto pelos programas que os reconheçam. A partir daí a UniVirtual ficará rodando em segundo plano, podendo o estudante desfrutar do OA e retornar ao sistema logo após.

5 VALIDAÇÃO E RESULTADOS

Com a finalidade de avaliar a eficácia do algoritmo de recomendação desenvolvido foram realizados testes em laboratório. Como será apresentado neste capítulo, o mecanismo de recomendação de conteúdo apresentou resultados satisfatórios. O processo de validação dos resultados foi realizado como descrito a seguir.

5.1 CENÁRIO DE TESTE

Esta seção trata de apresentar todo o cenário utilizado para o experimento, a sua execução e os resultados obtidos.

5.1.1 Definição dos Cursos e o Repositório

Para realização do experimento, foram criados três cursos no Moodle, nas seguintes áreas: Inteligência Artificial, Matemática e Administração. O repositório de conteúdos utilizado continha 270 objetos de aprendizagem (OAs) no padrão SCORM.

5.1.2 Estruturação do Repositório

Levando em consideração o metadado que identifica o contexto do OA, os OAs foram dispostos da seguinte forma:

- (i) 1/3 do total de OAs relacionado ao curso de "Inteligência Artificial";
- (ii) 1/3 do total de OAs relacionado ao curso de "Matemática";
- (iii) 1/3 do total de OAs relacionado ao curso de "Administração".

Levando em consideração a variável nível de dificuldade do conteúdo do OA, os OAs foram dispostos da seguinte forma:

- (i) 1/3 do total de OAs com nível "Fácil";
- (ii) 1/3 do total de OAs com nível "Médio";
- (iii) 1/3 do total de com nível "Difícil".

5.1.3 Estruturação dos Perfis dos Estudantes

Com a finalidade de avaliar quantitativamente a eficácia da recomendação, foram definidos alguns perfis de estudantes, sendo esses estruturados da seguinte forma:

- (i) 3 Estudantes do curso de Inteligência Artificial com níveis baixo, médio e alto de aproveitamento no curso respectivamente;
- (ii) 3 Estudantes do curso de Matemática com níveis baixo, médio e alto de aproveitamento no curso respectivamente;
- (iii) 3 Estudantes do curso de Administração com níveis baixo, médio e alto de aproveitamento no curso respectivamente;
- (iv) 3 Estudantes matriculados nos cursos de Inteligência Artificial e Matemática com níveis baixo, médio e alto de aproveitamento no curso, respectivamente, em ambos os cursos;
- (v) 1 Estudante não matriculado em nenhum curso.

5.1.4 Execução e Resultados do Experimento

Os conteúdos educacionais foram inseridos no repositório do Moodle, relacionados com os cursos de Inteligência Artificial, Matemática e Administração e o projeto foi alocado no servidor.

Os OAs foram recomendados de acordo com o curso e o nível de dificuldade, portanto, foram analisados os resultados obtidos para as duas características, os quais foram analisados tanto separadamente como em conjunto. Para cada perfil foram feitas 1000 recomendações. Como cada recomendação contém 3 OAs, o número total de OAs recomendados para cada perfil foi de 3000 OAs.

Os perfis e os resultados dos testes de recomendação são apresentados a seguir:

(a) Para o Perfil 1, cujo estudante está matriculado no Curso de Inteligência Artificial, com baixo nível de aproveitamento. Foram recomendados: 1916 OAs com nível de dificuldade "baixo", 754 OAs com nível de dificuldade "médio" e 330 OAs com nível de dificuldade "difícil", correspondendo a 63,86% do total de recomendações para o nível de dificuldade "baixo". Quanto ao contexto do curso, foram recomendados: 1963 OAs do curso de Inteligência Artificial, 523 OAs do curso de Administração e 514 OAs do curso de Matemática, correspondendo a 65,43% do total de recomendações para o conteúdo de Inteligência Artificial.

- (b) Para o Perfil 2, cujo estudante está matriculado no Curso de Inteligência Artificial, com nível médio de aproveitamento. Foram recomendados: 645 OAs com nível de dificuldade "baixo", 1701 OAs com nível de dificuldade "médio" e 654 OAs com nível de dificuldade "difícil", correspondendo a 56,7% do total de recomendações para o nível de dificuldade "médio". Quanto ao contexto do curso, foram recomendados: 1996 OAs do curso de Inteligência Artificial, 539 OAs do curso de Administração e 465 OAs do curso de Matemática, correspondendo a 66,53% do total de recomendações para o conteúdo de Inteligência Artificial.
- (c) Para o Perfil 3, cujo estudante está matriculado no Curso de Inteligência Artificial, com um alto nível de aproveitamento. Foram recomendados: 316 OAs com nível de dificuldade "baixo", 751 OAs com nível de dificuldade "médio" e 1933 OAs com nível de dificuldade "difícil", correspondendo a 64,43% do total de recomendações para o nível de dificuldade "difícil". Quanto ao contexto do curso, foram recomendados: 1949 OAs do curso de Inteligência Artificial, 531 OAs do curso de Administração e 520 OAs do curso de Matemática, correspondendo a 64,96% do total de recomendações para o conteúdo de Inteligência Artificial.
- (d) Para o Perfil 4, cujo estudante está matriculado no Curso de Administração, com baixo nível de aproveitamento. Foram recomendados: 1960 OAs com nível de dificuldade "baixo", 757 OAs com nível de dificuldade "médio" e 283 OAs com nível de dificuldade "difícil", correspondendo a 65,33% do total de recomendações para o nível de dificuldade "baixo". Quanto ao contexto do curso, foram recomendados: 516 OAs do curso de Inteligência Artificial, 1954 OAs do curso de Administração e 530 OAs do curso de Matemática, correspondendo a 65,13% do total de recomendações para o conteúdo de Administração.
- (e) Para o Perfil 5, cujo estudante está matriculado no Curso de Administração, com nível médio de aproveitamento. Foram recomendados: 667 OAs com nível de dificuldade "baixo", 1714 OAs com nível de dificuldade "médio" e 619 OAs com nível de dificuldade "difícil", correspondendo a 57,13% do total de recomendações para o nível de dificuldade "médio". Quanto ao contexto do curso, foram recomendados: 475 OAs do curso de Inteligência Artificial, 2050 OAs do curso de Administração e 475 OAs do curso de Matemática, correspondendo a 68,33% do total de recomendações para o conteúdo de Administração.
- (f) Para o Perfil 6, cujo estudante está matriculado no Curso de Administração, com alto nível de aproveitamento. Foram recomendados: 332 OAs com nível de dificuldade "baixo", 757 OAs com nível de dificuldade "médio" e 1911 OAs com nível de dificuldade "difícil", correspondendo a 64,43% do total de recomendações para o nível de dificuldade "difícil". Quanto ao contexto do curso, foram recomendados: 488 OAs do curso de Inteligência

- Artificial, 1974 OAs do curso de Administração e 538 OAs do curso de Matemática, correspondendo a 65,8% do total de recomendações para o conteúdo de Administração.
- (g) Para o Perfil 7, cujo estudante está matriculado no Curso de Matemática, com baixo nível de aproveitamento. Foram recomendados: 1893 OAs com nível de dificuldade "baixo", 775 OAs com nível de dificuldade "médio" e 332 OAs com nível de dificuldade "difícil", correspondendo a 63,1% do total de recomendações para o nível de dificuldade "baixo". Quanto ao contexto do curso, foram recomendados: 534 OAs do curso de Inteligência Artificial, 508 OAs do curso de Administração e 1958 OAs do curso de Matemática, correspondendo a 65,26% do total de recomendações para o conteúdo de Matemática.
- (h) Para o Perfil 8, cujo estudante está matriculado no Curso de Matemática, com nível médio de aproveitamento. Foram recomendados: 638 OAs com nível de dificuldade "baixo", 1765 OAs com nível de dificuldade "médio" e 597 OAs com nível de dificuldade "difícil", correspondendo a 58,83% do total de recomendações para o nível de dificuldade "médio". Quanto ao contexto do curso, foram recomendados: 472 OAs do curso de Inteligência Artificial, 490 OAs do curso de Administração e 2038 OAs do curso de Matemática, correspondendo a 67,93% do total de recomendações para o conteúdo de Matemática.
- (i) Para o Perfil 9, cujo estudante está matriculado no Curso de Matemática, com alto nível de aproveitamento. Foram recomendados: 328 OAs com nível de dificuldade "baixo", 763 OAs com nível de dificuldade "médio" e 1909 OAs com nível de dificuldade "difícil", correspondendo a 63,63% do total de recomendações para o nível de dificuldade "difícil". Quanto ao contexto do curso, foram recomendados: 489 OAs do curso de Inteligência Artificial, 523 OAs do curso de Administração e 1988 OAs do curso de Matemática, correspondendo a 66,26% do total de recomendações para o conteúdo de Matemática.
- (j) Para o Perfil 10, cujo estudante está matriculado nos cursos de Inteligência Artificial e Administração, com baixo nível de aproveitamento. Foram recomendados: 1985 OAs com nível de dificuldade "baixo", 733 OAs com nível de dificuldade "médio" e 282 OAs com nível de dificuldade "difícil", correspondendo a 66,16% do total de recomendações para o nível de dificuldade "baixo". Quanto ao contexto do curso, foram recomendados: 1378 OAs do curso de Inteligência Artificial, 1351 OAs do curso de Administração e 271 OAs do curso de Matemática, correspondendo a 45,93% do total de recomendações para o conteúdo de Inteligência Artificial, e 45,03% do total de recomendações para o curso de Administração.
- (k) Para o Perfil 11, cujo estudante está matriculado nos cursos de Inteligência Artificial e Administração, com nível médio de aproveitamento. Foram recomendados: 615 OAs com nível de dificuldade "baixo", 1779 OAs com nível de dificuldade "médio" e 606 OAs com nível de dificuldade "difícil", correspondendo a 59,03% do total de recomendações para o nível de dificuldade "médio". Quanto ao contexto do curso, foram recomendados:

1410 OAs do curso de Inteligência Artificial, 1336 OAs do curso de Administração e 254 OAs do curso de Matemática, correspondendo a 47% do total de recomendações para o conteúdo de Inteligência Artificial, e 44,53% do total de recomendações para o curso de Administração.

- (l) Para o Perfil 12, cujo estudante está matriculado nos cursos de Inteligência Artificial e Administração, com alto nível de aproveitamento. Foram recomendados: 283 OAs com nível de dificuldade "baixo", 750 OAs com nível de dificuldade "médio" e 1967 OAs com nível de dificuldade "difícil", correspondendo a 65,56% do total de recomendações para o nível de dificuldade "difícil". Quanto ao contexto do curso, foram recomendados: 1356 OAs do curso de Inteligência Artificial, 1327 OAs do curso de Administração e 317 OAs do curso de Matemática, correspondendo a 45,2% do total de recomendações para o conteúdo de Inteligência Artificial, e 44,23% do total de recomendações para o curso de Administração.
- (m) Para o Perfil 13, cujo estudante não está matriculado em nenhum curso. Foram recomendados: 1037 OAs (X%) com nível de dificuldade "baixo", 947 OAs (Y%) com nível de dificuldade "médio" e 1016 OAs (Z%) com nível de dificuldade "difícil". Quanto ao contexto do curso, foram recomendados: 1016 OAs (X%) do curso de Inteligência Artificial, 1021 OAs (Y%) do curso de Administração e 963 OAs (Z%) do curso de Matemática.

Com exceção do perfil 13, no qual o estudante não estava matriculado em nenhum curso, tinha-se 12 perfis divididos em 3 tipos de níveis de aproveitamento do aluno no curso, que é o critério que indica o nível do aluno, que pode ser: Baixo, Médio e Alto. Sendo assim, a Tabela 6 apresenta o total de acertos para estes perfis segundo o critério do nível do aluno. Vale ressaltar que como foram 4 perfis com níveis "baixo" (Perfil 1, 4, 7 e 10), "médio" (Perfil 2, 5, 8 e 11) e "alto" (Perfil 3, 6, 9 e 12), e como para cada perfil foram realizadas 1000 recomendações, sendo que cada recomendação contém 3 OAs, o total de OAs recomendados para cada nível foi de 12.000 OAs.

A tabela 6 a seguir apresenta de forma resumida o percentual da recomendação segundo o critério do nível de dificuldade.

Nível de Dificuldade	Quantidade (Recomend	(%) Acertos	Total	
	Verdadeiro-Positivo	Falso-Positivo	Accitos	
Fácil	7754	4246	64,61%	12.000
Médio	6959	5041	57,99%	12.000
Difícil	7720	4280	64,33%	12.000

Tabela 6 – Resultados da recomendação por nível de dificuldade.

Vale ressaltar que os dados da recomendação referente ao perfil apresentado no item *m* não foram considerados para a Tabela 6, pois, como este perfil de estudante não está associado a dificuldade alguma, não há então como quantificar a relação da dificuldade dos itens recomendados com o perfil do estudante.

Quanto ao critério do contexto do OA por curso, foram testados 13 perfis de estudantes. Estes 13 perfis estavam divididos em 5 tipos, que correspondem ao estudante matriculado nos cursos de: Inteligência Artificial, Administração, Matemática, Inteligência Artificial e Matemática, e, por fim, o estudante não estando associado a nenhum curso. Sendo assim, a Tabela 7 apresenta o total de acertos para estes perfis segundo o critério de relação do OA com o curso no qual o estudante está matriculado.

Curso	Quantidade de OAs	(%)	Total	
Curso	Verdadeiro-Positivo Falso-Positivo		Acertos	Iotai
Inteligência	5908	3092	65 6101	9.000
Artificial	3908	3092	65,64%	
Administração	5978	3022	66,42%	9.000
Matemática	5984	3016	66,48%	9.000
Inteligência Artificial /	8158	842	90,64%	9.000
Administração	0130	042	90,04%	9.000
Nenhum Curso	0	3000	100%	3.000

Tabela 7 – Resultados da recomendação por curso.

Para este critério, o total de OAs recomendados por curso foi de 9.000, pois correspondem aos perfis com níveis baixo, médio e alto de aproveitamento do estudante no curso. Como foram realizadas 1000 recomendações para cada perfil, sendo 3 OAs por recomendação, foram recomendados 9.000 OAs para cada curso. Não foi contemplado o perfil que retrata o aluno não matriculado em nenhum curso, pois, como o mesmo não tem matrícula em curso, não há atribuição de nota para o mesmo, não sendo possível identificar seu índice de aproveitamento no curso.

Quanto ao critério do nível de dificuldade dos OAs e o conteúdo do curso, os resultados dos testes de recomendação são apresentados na Tabela 8.

Curso	Dificuldade do OA	Recomendações		(%) Total	
Curso		VP	FP	Acertos	Total
Inteligência	Fácil (Perfil 1)	1223	1777	40,76%	3.000
Artificial	Médio (Perfil 2)	1113	1887	37,10%	3.000
Aitiliciai	Difícil (Perfil 3)	1210	1790	40,33%	3.000
	Fácil (Perfil 4)	1248	1752	41,60%	3.000
Administração	Médio (Perfil 5)	1141	1859	38,03%	3.000
	Difícil (Perfil 6)	1218	1782	40,60%	3.000
	Fácil (Perfil 7)	1199	1801	39,96%	3.000
Matemática	Médio (Perfil 8)	1182	1818	39,40%	3.000
	Difícil (Perfil 9)	1217	1783	40,56%	3.000
Inteligência Artificial /	Fácil (Perfil 10)	1789	1211	59,63%	3.000
Administração	Médio (Perfil 11)	1618	1382	53,93%	3.000
Aummstração	Difícil (Perfil 12)	1724	1276	57,46%	3.000

Tabela 8 – Resultados da recomendação por curso e nível de afinidade.

Com a finalidade de efetuar uma validação sobre sistemas de recomendação, muitos pesquisadores têm utilizado métricas como *precision* (p) e *recall* (r) (SALEHI; KMALABADI, 2012; SALEHI; KMALABADI; GHOUSHCHI, 2012). Estas correspondem a métricas amplamente utilizadas com a finalidade de avaliar a utilidade das recomendações produzidas por um sistema de recomendação.

A métrica *precision* corresponde à proporção de resultados relevantes levando em consideração alguma definição de recomendação relevante para o domínio do problema (CARACIOLO, 2011). A Figura 27 apresenta a fórmula da métrica *precision*.

$$p = \frac{|\{documentos \ relevantes\} \cap \{documentos \ recomendados\}|}{|\{documentos \ recomendados\}|}$$

Figura 27 – Equação da métrica *Precision*. Fonte: Autoria própria.

A métrica *recall* corresponde à proporção de todos os resultados relevantes que estão inseridos no topo das recomendações relevantes (CARACIOLO, 2011). A Figura 28 apresenta a fórmula de cálculo desta métrica.

```
r = \frac{|\{documentos \ relevantes\} \ \cap \ \{documentos \ recomendados\}|}{|\{documentos \ relevantes\}|}
```

Figura 28 – Equação da métrica *Recall*. Fonte: Autoria própria.

Efetuando um comparativo entre as duas métricas, elas podem ser definidas da seguinte forma: *precision* corresponde à proporção de boas recomendações, ou seja, quanto do total que

foi recomendado deveria realmente ser recomendado e *recall* corresponde à proporção de boas recomendações que aparecem no topo das recomendações, ou seja, quanto do que deveria ser recomendado foi realmente recomendado.

Uma vez que é comum alcançar *recall* igual a 100%, onde todos os documentos recomendados são relevantes, o *recall* em si não é suficiente para efetuar uma validação sobre o sistema de recomendação. Por este motivo, é necessário calcular também *precision*, que corresponde ao número de documentos não relevantes que foram recomendados. E, além disso, aplicar a métrica *F-measure* (ou *F1 score*) que combina *recall* e *precision* (SALEHI; KMALABADI, 2012; SALEHI; KMALABADI; GHOUSHCHI, 2012). A Figura 29 exemplifica o cálculo desta métrica.

$$F1 = 2x \frac{precision \ x \ recall}{precision + recall}$$

Figura 29 – Equação da métrica *F-meansure*. Fonte: Autoria própria.

Aplicando essas métricas sobre os resultados apresentados na Tabela 8, obtêm-se os resultados que podem ser vistos na Tabela 9.

Tabela 9 – Resultados das métricas <i>Precision</i> ,	Recall e F-meansure da recomendação por curso e nível de
afinidade.	

Carres	D:611- 1- 1- 04	Recomendações			
Curso	Dificuldade do OA	Precision	Recall	F-meansure	
Inteligência	Fácil (Perfil 1)	0,4076	1,00	0,5791	
Artificial	Médio (Perfil 2)	0,3710	1,00	0,5412	
Arunciai	Difícil (Perfil 3)	0,4033	1,00	0,5747	
	Fácil (Perfil 4)	0,4160	1,00	0,5875	
Administração	Médio (Perfil 5)	0,3803	1,00	0,5510	
	Difícil (Perfil 6)	0,4060	1,00	0,5775	
	Fácil (Perfil 7)	0,3996	1,00	0,5710	
Matemática	Médio (Perfil 8)	0,3940	1,00	0,5652	
	Difícil (Perfil 9)	0,4056	1,00	0,5771	
Intoliaônaia Autificial /	Fácil (Perfil 10)	0,5963	1,00	0,7471	
Inteligência Artificial /	Médio (Perfil 11)	0,5393	1,00	0,7007	
Administração	Difícil (Perfil 12)	0,5746	1,00	0,7298	

Como pode ser visto em Pazos-Arias et al. (2008), o sistema de recomendação proposto obteve resultados compatíveis com resultados de *precision* e *recall* de diferentes estratégias para recomendação de cursos. Os resultados da avaliação mostram que a abordagem aqui apresentada fornece grande qualidade nas recomendações em *F-meansure*.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Para concretizar este trabalho foi implementada uma arquitetura baseada em agentes para recomendação de OAs em um ambiente virtual 3D. A solução construída objetivou tornar a aprendizagem da modalidade EaD, personalizada e adequada conforme as necessidades de cada estudante, visando estimular o estudo em cursos desta modalidade. Alguns componentes foram criados para contemplar todo o sistema. Primeiramente foi desenvolvido o ambiente 3D empregando a realidade virtual, em seguida, um SMA composto por um mecanismo de recomendação, e por fim, um aplicativo capaz de identificar a localização geográfica do usuário.

Em termos de contribuições, este trabalho se destaca pelo desenvolvimento do SMA que pode ser utilizado em outros seguimentos, pois sua arquitetura se adéqua a qualquer área de aplicação, bem como o desenvolvimento do ambiente virtual 3D que encontra-se com uma estrutura adaptável a outras ferramentas de gestão de aprendizagem *open source* além do Moodle, e, por fim, o desenvolvimento do aplicativo de localização geográfica que também funciona de forma independente.

Mesmo com a realização do trabalho, alguns ajustes são imprescindíveis para a sua melhoria. Portanto, como trabalhos futuros, pretende-se:

- Ampliar os tipos de OAs suportados pela UniVirtual;
- Implementar técnicas que permitam a abertura dos OAs de modo bastante interativo e específico conforme os cenários, aumentando cada vez mais a imersão neste ambiente virtual tridimensional;
- Implementar uma função *multiplayer* ao ambiente, que permita uma interação entre vários estudantes, de modo a tornar esta ferramenta cada vez mais atrativa e estimulante para os estudantes;
- Implementar a captura da localização geográfica a partir do própria ambiente 3D, pois próximo a finalização desta primeira versão da UniVirtual foi descoberta a possibilidade de efetuar isto através da API do Unity para Android. Diminuindo a necessidade de se instalar componentes extras que influenciam no funcionamento do sistema;
- Aprimorar o algoritmo de recomendação com a intenção de melhorar mais ainda os resultados da validação e consequentemente proporcionar uma entrega mais precisa de conteúdos de aprendizagem.

Apesar da pretensão de melhorias a versão atual da UniVirtual proporcionou um resultado satisfatório na validação efetuada, entretanto, ainda é necessário executar testes em contexto real para auxiliar no aprimoramento da ferramenta. Contudo, a UniVirtual demonstrou cumprir com o foco principal da recomendação de OAs com base no contexto que o estudante se encontra em

um dado momento, com o propósito de oferecer-lhe um meio estimulante dentro da modalidade EaD através do ambiente 3D desenvolvido.

REFERÊNCIAS

ADL. ADL Guidelines for Creating Reusable Content with SCORM 2004. Advanced Distributed Learning. [S.l.]: 4ed, 2008. Citado 2 vezes nas páginas 36 e 39.

ALLEGORITHMIC. Substance: simply the most powerful texturing toolset out there. 2014. Disponível em: http://www.allegorithmic.com. Citado na página 52.

ANDROID. Site oficial do sistema android. 2014. Disponível em: http://www.android.com/>. Citado 2 vezes nas páginas 17 e 58.

ARTERO, A. O. Inteligência artificial-teoria e prática. *Livraria da Física*, 2009. Citado 7 vezes nas páginas 23, 24, 25, 27, 28, 65 e 67.

AZEVEDO, C. E. F. de; ELIA, M. d. F. Proposta de uma aplicação de mundos virtuais na educação usando o open simulator com diferentes requisitos tecnológicos. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1. Citado na página 21.

BEHAR, P. A. Competências em educação a distância. [S.1.]: Penso Editora, 2013. Citado na página 17.

BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. *Developing multi-agent systems with JADE*. [S.l.]: John Wiley & Sons, 2007. Citado na página 53.

BLENDER. Site oficial do blender. 2014. Disponível em: http://www.blender.org/>. Citado na página 50.

BOS, A. S. et al. A história da computação através de um museu virtual interativo utilizando mundos virtuais 3d. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2013. v. 1, n. 1. Citado na página 21.

CABALLE, S. et al. Architectures for Distributed and Complex M-Learning Systems: Applying Intelligent Technologies. [S.l.]: Information Science Reference, 2010. Citado na página 31.

CARACIOLO, M. Artificial intelligence in motion. evaluating recommender systems - explaining f-score, recall and precision using real data set de apontador. 2011. Acesso em: 30 dez. Disponível em: http://aimotion.blogspot.com.es/2011/05/evaluating-recommender-systems.html. Citado na página 80.

CARVALHO, J. Ensino-aprendizagem do ple em modalidade híbrida. *Revista Animação e Educação*, 2010. Citado na página 53.

CASTILLO, S.; AYALA, G. Armoleo: An architecture for mobile learning objects. In: IEEE. *Electronics, Communications and Computers*, 2008. CONIELECOMP 2008, 18th International Conference on. [S.l.], 2008. p. 53–58. Citado 3 vezes nas páginas 30, 31 e 35.

CAZELLA, S. C. et al. Desenvolvendo um sistema de recomendação de objetos de aprendizagem baseado em competências para a educação: relato de experiências. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2012. v. 23, n. 1. Citado na página 34.

CAZELLA, S. C.; NUNES, M.; REATEGUI, E. B. A ciência da opinião: estado da arte em sistemas de recomendação. In: *XXX Congresso da SBC Jornada de Atualização da Informática*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 34 e 35.

COMMITTEE, L. T. S. et al. *Draft Standard for Learning Object Metadata. IEEE Standard 1484.12. 1, New York: Institute of Electrical and Electronics Engineers.* 2002. Citado 2 vezes nas páginas 35 e 37.

CREIGHTON, R. H. *Unity 3D Game Development by Example: Beginner's Guide.* [S.l.]: Packt Publishing, 2010. Citado na página 50.

CUNHA, F.; MAINENTE, A. Utilização de ambientes virtuais 3d no ensino de ciência da computação: estado da arte. *São Caetano do Sul*, 2011. Citado na página 19.

DIAS, C. et al. Padrões abertos: aplicabilidade em objetos de aprendizagem (oas). *Anais do XX Simpósio Brasileiro de Informática na Educação, SBIE, Florianópolis, SC*, 2009. Citado na página 37.

ECLIPSE. Site oficial do eclipse. 2014. Disponível em: http://www.eclipse.org. Citado na página 58.

FERBER, J. *Multi-agent systems: an introduction to distributed artificial intelligence*. [S.l.]: Addison-Wesley Reading, 1999. Citado na página 23.

FIALHO, F. A. P. Gestão do conhecimento e aprendizagem: as estratégias competitivas da sociedade pós-industrial. [S.l.]: Visual Books, 2006. Citado na página 19.

FONTES, L. M. *Uma Arquitetura Multiagente de Apoio à Aprendizagem Baseada em Problema*. Dissertação (Mestrado) — Universidade do Estadual do Rio Grande do Norte–UERN e Universidade Federal Rural do Semi-Árido-UFERSA. Mossoró, RN, 2013. Citado 4 vezes nas páginas 28, 30, 53 e 65.

FRANCISCATO, F. T.; MEDINA, R. D. M-learning e android: um novo paradigma? *RENOTE*, v. 6, n. 1, 2008. Citado na página 58.

FUSE. Fuse character creator. create high-quality 3d characters in minutes, not days. 2014. Disponível em: https://www.mixamo.com/fuse. Citado na página 52.

GPL. The gnu general public license v3.0 - gnu project - free software foundation (fsf). gnu general public license. 2007. Disponível em: http://www.gnu.org/copyleft/gpl. Citado na página 53.

HENDERSON-SELLERS, B.; GIORGINI, P. Agent-oriented methodologies. [S.l.]: IGI Global, 2005. Citado na página 23.

HERLOCKER, J. L. *Understanding and improving automated collaborative filtering systems*. Tese (Doutorado) — University of Minnesota, 2000. Citado 2 vezes nas páginas 34 e 35.

JADE. Java development framework: an open-source platform for peer-to-peer agent based applications. 2014. Disponível em: http://jade.tilab.com. Citado na página 53.

JÚNIOR, L. J. et al. Uma extensão do moodle para recomendação ubíqua de objetos de aprendizagem. *RENOTE*, v. 10, n. 3, 2012. Citado na página 34.

KEMCZINSKI, A. et al. Repositório de objetos de aprendizagem para a área de computação e informática-roai. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1. Citado na página 37.

KIRNER, C. Prototipagem rápida de aplicações interativas de realidade aumentada. *Tendências e Técnicas em Realidade Virtual e Aumentada*, v. 1, n. 1, p. 29–54, 2011. Citado na página 21.

- KOZA, J. R. Genetic programming: on the programming of computers by means of natural selection. [S.l.]: MIT press, 1992. Citado 2 vezes nas páginas 26 e 27.
- LI, S.; YANG, Z.; LIU, Q. Research of metadata based digital educational resource sharing. In: IEEE. *Computer Science and Software Engineering*, 2008 International Conference on. [S.l.], 2008. v. 5, p. 828–831. Citado na página 37.
- LI, Z.; YUE, J.; JÁUREGUI, D. A. G. A new virtual reality environment used for e-learning. In: IEEE. *IT in Medicine & Education*, 2009. *ITIME'09*. *IEEE International Symposium on*. [S.l.], 2009. v. 1, p. 445–449. Citado 2 vezes nas páginas 9 e 21.
- LINDEN, R. *Algoritmos Geneticos Uma Importante Ferramenta da Inteligência Computacional (3a edicao)*. [S.l.]: Brasport, 2012. Citado 5 vezes nas páginas 26, 27, 28, 30 e 45.
- LITTO, F. M.; FORMIGA, M. *Educação a distância: o estado da arte*. [S.l.]: Pearson, 2012. Citado 3 vezes nas páginas 19, 21 e 22.
- LOUREIRO, A. A. F. et al. Computação ubíqua ciente de contexto: Desafios e tendências. *Anais do XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SRBC 09)*, 2009. Citado 2 vezes nas páginas 32 e 33.
- LUGER, G. F. *Inteligência Artificial-: Estruturas e estratégias para a solução de problemas complexos*, 6^a *Edição*. [S.l.]: Bookman, 2014. Citado na página 25.
- MANDULA, K. et al. Implementation of ubiquitous learning system using sensor technologies. In: IEEE. *Technology for Education (T4E)*, *2011 IEEE International Conference on*. [S.l.], 2011. p. 142–148. Citado 2 vezes nas páginas 32 e 33.
- MARCELINO, R. Ambiente virtual de aprendizagem integrado a mundo virtual 3D ea experimento remoto aplicados ao tema resistência dos materiais. 2010. 124f. Tese (Doutorado) Tese (Doutorado em Engenharia de Minas, Metalúrgica e de Materiais)—Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010. Citado na página 20.
- MARTINS, H. N. et al. Sistema de recomendação híbrido para bibliotecas digitais que suportam o protocolo oai-pmh. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1. Citado na página 35.
- MIXAMO. Site oficial da mixamo. 2014. Disponível em: https://www.mixamo.com/. Citado 3 vezes nas páginas 9, 52 e 57.
- MOODLE. Site oficial do moodle. 2014. Disponível em: https://moodle.org/>. Citado 3 vezes nas páginas 22, 53 e 61.
- MOORE, P. et al. Intelligent context for personalised m-learning. In: IEEE. *Complex, Intelligent and Software Intensive Systems*, 2009. CISIS'09. International Conference on. [S.l.], 2009. p. 247–254. Citado 2 vezes nas páginas 31 e 32.
- MOORE, P.; JACKSON, M.; HU, B. Constraint satisfaction in intelligent context-aware systems. In: IEEE. *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on.* [S.1.], 2010. p. 75–80. Citado na página 17.

MORAIS, M. I. *MAS-CommonKADS+: Uma Extensão à Metodologia Mas-CommonKADS para Suporte ao Processo Detalhado de Sitemas Multiagentes Racionais*. Dissertação (Mestrado) — Universidade Estadual do Ceará-UECE. Fortaleza, CE, 2010. Citado na página 65.

- MYSQL, M. Site oficial do mysql. 2014. Disponível em: http://www.mysql.com. Citado na página 53.
- NETO, S. P.; FAJ, U. Computação evolutiva: desvendando os algoritmos genéticos. *Revista Ubiquidade*, v. 1, n. 1, p. 34, 2011. Citado 4 vezes nas páginas 26, 27, 28 e 30.
- NORVIG, P.; RUSSELL, S. *Inteligência Artificial*, 3^a Edição. [S.l.]: Elsevier Brasil, 2014. Citado 2 vezes nas páginas 22 e 24.
- OLIVEIRA, E. A.; TEDESCO, P. i-collaboration: Um modelo de colaboração inteligente personalizada para ambientes de ead. *Revista Brasileira de Informática na Educação*, v. 18, n. 01, p. 17, 2010. Citado 2 vezes nas páginas 16 e 17.
- OSÓRIO, F. S. et al. Ambientes virtuais interativos e inteligentes: Fundamentos, implementação e aplicações práticas. In: *XXIV Congresso da SBC–JAI 2004*. [S.l.: s.n.], 2004. Citado na página 21.
- PAZOS-ARIAS, J. J. et al. Provision of distance learning services over interactive digital tv with mhp. *Computers & Education*, Elsevier, v. 50, n. 3, p. 927–949, 2008. Citado na página 81.
- PHP. Site oficial do php. 2014. Acesso em: 15 dez. Disponível em: http://www.php.net>. Citado na página 53.
- RODOLPHO, E. R. Convergência digital de objetos de aprendizagem scorm. *Universidade Estadual de São Paulo Júlio de Mesquita Filho (UNESP), São José do Rio Preto, SP*, 2009. Citado 2 vezes nas páginas 35 e 37.
- ROSSI, L. H. L. et al. Tradução de metadados educacionais, um passo importante para a interoperabilidade. *Conferencias LACLO*, v. 4, n. 1, 2013. Citado na página 37.
- SACCOL, A. et al. *M-learning e u-learning: novas perspectivas da aprendizagem móvel e ubíqua*. [S.l.]: São Paulo, SP: Pearson, 2010. Citado 2 vezes nas páginas 16 e 32.
- SALEHI, M.; KMALABADI, I. N. A hybrid attribute—based recommender system for e—learning material recommendation. *IERI Procedia*, Elsevier, v. 2, p. 565–570, 2012. Citado 2 vezes nas páginas 80 e 81.
- SALEHI, M.; KMALABADI, I. N.; GHOUSHCHI, M. B. G. A new recommendation approach based on implicit attributes of learning material. *IERI Procedia*, Elsevier, v. 2, p. 571–576, 2012. Citado 2 vezes nas páginas 80 e 81.
- SALES, A. F. de A. *Um Sistema de Recomendação para Aprendizagem Ubíqua no Contexto da Educação Formal e Informal*. Dissertação (Mestrado) Universidade do Estadual do Rio Grande do Norte–UERN e Universidade Federal Rural do Semi-Árido-UFERSA. Mossoró, RN, 2014. Citado 2 vezes nas páginas 34 e 35.
- SANDES, T. A. d. C.; SILVA, E. R. d. Os softwares livres na produção do jogo estrada real digital, ea utilização do mesmo pelos professores de história. In: *Anais do Congresso Nacional Universidade, EAD e Software Livre*. [S.l.: s.n.], 2012. v. 2, n. 2. Citado na página 17.

SECONDLIFE. Site oficial do secondlife. 2014. Disponível em: http://secondlife.com/>. Citado na página 22.

- SILVA, L. *MobiLE–Um Ambiente Multiagente de Aprendizagem Móvel para Apoiar a Recomendação Ubíqua de Objetos de Aprendizagem*. Dissertação (Mestrado) Universidade do Estadual do Rio Grande do Norte–UERN e Universidade Federal Rural do Semi-Árido-UFERSA. Mossoró, RN, 2012. Citado 9 vezes nas páginas 9, 26, 27, 33, 35, 37, 39, 49 e 65.
- SILVA, L. C. N. da; NETO, F. M. M.; JÚNIOR, L. J. Mobile: Um ambiente multiagente de aprendizagem movel para apoiar a recomendacao sensivel ao contexto de objetos de aprendizagem. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2011. v. 1, n. 1. Citado na página 33.
- SILVA, M. Avaliação da aprendizagem em educação online: fundamentos, interfaces e dispositivos, relatos de experiências. [S.l.]: Edicoes Loyola, 2006. Citado na página 19.
- SILVA, T. R. da et al. Oba-mc: um modelo de objeto de aprendizagem centrado no processo de ensino-aprendizagem utilizando o padrão scorm. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2012. v. 23, n. 1. Citado na página 39.
- SOUZA D. F. L.; CASTRO, A. F. M. H. W. Ufersa virtual campus virtual inteligente para navegação e integração de serviços acadêmicos. *XIX Seminário de Iniciação Científica SEMIC*, 2013. Citado na página 54.
- UNITY. Site oficial da unity. 2014. Disponível em: http://unity3d.com/unity. Citado na página 50.
- VIEIRA, F. J. R.; NUNES, M. A. S. N. Dica: Sistema de recomendação de objetos de aprendizagem baseado em conteúdo. *Scientia Plena*, v. 8, n. 5, 2012. Citado na página 34.
- WIEDEMANN, T.; BARBOSA, J. L.; RIGO, S. J. Um modelo para recomendação de objetos de aprendizagem baseado em similaridade de sessões. In: *Anais do Simpósio Brasileiro de Informática na Educação*. [S.l.: s.n.], 2013. v. 24, n. 1. Citado na página 49.
- WILEY, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. [S.l.: s.n.], 2003. Citado na página 16.
- WOOLDRIDGE, M. *An introduction to multiagent systems*. [S.l.]: John Wiley & Sons, 2009. Citado na página 23.
- YAU, J. Y.-K.; JOY, M. A context-aware personalised m-learning application based on m-learning preferences. *International Journal of Mobile Learning and Organisation*, Inderscience, v. 5, n. 1, p. 1–14, 2011. Citado 5 vezes nas páginas 16, 30, 31, 32 e 49.
- YESSAD, A.; LABAT, J.; KERMORVANT, F. Segae: A serious game authoring environment. In: IEEE. *Advanced Learning Technologies (ICALT)*, 2010 IEEE 10th International Conference on. [S.l.], 2010. p. 538–540. Citado 2 vezes nas páginas 16 e 17.
- ZHAO, R.; WANG, J. Visualizing the research on pervasive and ubiquitous computing. *Scientometrics*, Springer, v. 86, n. 3, p. 593–612, 2011. Citado na página 33.
- ZINI, E. D. O. C. *Algoritmo Genético Especializado na Resolução de Problemas com Variáveis Contínuas e Altamente Restritos*. Dissertação (Mestrado) Universidade Estadual Paulista–UNESP. Solteira–SP, Ilha, 2009. Citado 3 vezes nas páginas 9, 28 e 29.

APÊNDICE A - CLASSE VERIFICANDODISTANCIA

```
1 public class VerificandoDistancia extends AsyncTask<String, integer,
      Double>{
2.
     @Override
3
4
     public Double doInBackground(String... uri) {
        estaPerto = null;
5
        distancia = null;
6
        Log.i("Verifica Distancia", "Vai entrar no try");
7
        String urlPost = "http://"+ip+"/android/calcularDistancia.php";
8
9
        String urlPost1 = "http://"+ip+"/android/pegarcoord.php";
        ArrayList<NameValuePair> parametrosPost = new ArrayList<
10
           NameValuePair>();
11
        parametrosPost.add(new BasicNameValuePair("ID", IdAluno.toString
           ()));
12
        parametrosPost.add(new BasicNameValuePair("Longitude", longitude
           .toString()));
        parametrosPost.add(new BasicNameValuePair("Latitude", latitude.
13
           toString()));
        String respostaRetornada = null;
14
        String respostaRetornada1 = null;
15
16
        try {
           respostaRetornada = ConexaoHttpClient.executaHttpPost(urlPost
17
              , parametrosPost);
           String resposta = respostaRetornada.toString();
18
           Log.i("Verifica Distancia", "resposta = "+resposta);
19
           estaPerto = Double.parseDouble(resposta.toString());
20
           Log.i("Verifica Distancia", "Verifica Distancia = "+estaPerto
21
              );
22
           if (estaPerto == 1.0) {
23
               distancia = "Perto";
24
           try {
25
               respostaRetornada1 = ConexaoHttpClient.executaHttpPost(
26
                  urlPost1, parametrosPost);
27
               String resposta1 = respostaRetornada1.toString();
28
               char separadaror='#';
29
30
               for(int i=0;i<respostal.length();i++)</pre>
31
                    if (separadaror == respostal.charAt(i))
32
                        contador++;
33
34
               coordenadas = new String[contador];
35
36
               char caracter_lido=respostal.charAt(0);
37
               String coord="";
38
               for (int i=0; caracter_lido != '^'; i++)
39
40
                    caracter_lido = resposta1.charAt(i);
41
```

```
Log.i("chars dos usuarios", ""+caracter_lido);
42
                     if (caracter_lido != '#')
43
                          coord+= (char) caracter_lido;
44
                     else
45
46
                     {
                         Log.i("Nome", ""+coord);
47
                         coordenadas[posicao] = ""+coord;
48
                         Log.i("Nome posicao ["+posicao+"]",""+coordenadas
49
                             [posicao]);
                         posicao++;
50
                         coord="";
51
                     }
52
                 }
53
                Log.i("FIM", "FIM da leitura de coordenadas");
54
55
56
            }
57
            catch (Exception erro)
58
                Log.i("erro", "erro = "+erro);
59
60
            }
61
62
63
            if(estaPerto == 0.0){
64
                distancia = "Longe";
65
            }
66
       resposta = resposta.replaceAll("\\s+", "resposta = "+resposta);
67
68
       catch (Exception erro) {
69
            distancia = "Error";
70
            Log.i("erro", "erro = "+erro);
71
            error = erro.toString();
72
73
       }
74
       return estaPerto;
75
76
     protected void onPostExecute(Double result) {
77
       if (distancia == "Perto") {
78
79
            locationManager.removeUpdates(locationListener);
80
            notificar();
81
            criandotoast("vc esta Perto");
            NaArea();
82
83
84
       if (distancia == "Longe") {
85
86
            criandotoast("vc esta longe");
87
       if (distancia == "Error") {
88
            criandotoast(error);
89
90
       }
91
92
```