



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



LUIZ JÁCOME JÚNIOR

**MobiLE+: UM AMBIENTE DE SUPORTE À APRENDIZAGEM
UBÍQUA**

MOSSORÓ – RN

2014

LUIZ JÁCOME JÚNIOR

**MobiLE+: UM AMBIENTE DE SUPORTE À APRENDIZAGEM
UBÍQUA**

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. Dr. Francisco Milton Mendes Neto - UFRSA.

Coorientadora: Profa. Dra. Cecília Dias Flore - UFCSPA.

MOSSORÓ – RN

2014

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Jácome Júnior, Luiz.

MobilE+: um ambiente de suporte à aprendizagem ubíqua. / Luiz Jácome Júnior. – Mossoró, RN, 2014

99 f.

Orientador(a): Prof. Dr. Francisco Milton Mendes Neto.

Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Aprendizagem móvel - Dissertação. 2. Aprendizagem ubíqua - Dissertação. 3. Sistemas multiagente - Dissertação. I. Mendes Neto, Francisco Milton. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/BC

CDD 004.678

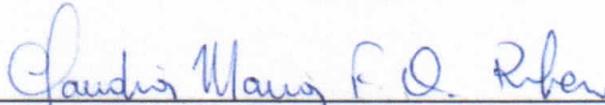
LUIZ JÁCOME JÚNIOR

**MobiLE+: UM AMBIENTE DE SUPORTE À APRENDIZAGEM
UBÍQUA**

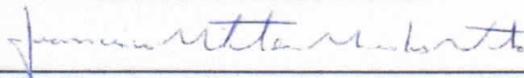
Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: 25/02/2014.

BANCA EXAMINADORA



Prof. Dra. Cláudia Maria Fernandes Araújo Ribeiro – UFERSA
Presidente e Primeiro Membro



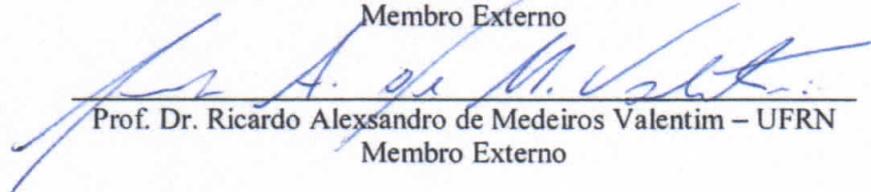
Prof. Dr. Francisco Milton Mendes Neto – UFERSA
Orientador e Segundo Membro



Prof. Dra. Cecília Dias Flores – UFCSPA
Coorientadora e Terceiro Membro



Prof. Dr. Aquiles Medeiros Filgueira Burlamaqui – UFRN
Membro Externo



Prof. Dr. Ricardo Alexandro de Medeiros Valentim – UFRN
Membro Externo

Aos meus irmãos Neto, Canindé, Diá e Das Chagas (In memoriam) pelo companheirismo e cumplicidade em tudo que passamos juntos.

Aos meus pais Luiz Jácome e Neide Linhares Jácome, por tudo que me ensinaram e pelo exemplo e apoio que sempre me deram, à minha esposa Adya, que tanto me incentivou, e em especial ao meu filho, Luiz Neto, por todos os momentos de descontração e felicidade.

AGRADECIMENTOS

Aproveito a oportunidade para externar minha gratidão a pessoas que foram essenciais em minha vida e nessa caminhada árdua do mestrado, sem as quais eu não teria alcançado todos os meus objetivos.

Agradeço ao meu pai, por todos os esforços realizados para contribuir com minha formação e pelo exemplo de vida e dedicação com a família, sempre nos guiando para o melhor caminho, e a minha mãe, pelo incentivo e por sempre ter se mostrado uma guerreira diante das dificuldades.

Gostaria de agradecer especialmente ao meu filho, que sempre me proporciona momentos felizes e ajudou-me a ter uma compreensão diferente da vida. Acho brilhante como suas atitudes simples marcam minha vida.

Agradeço à minha esposa Adya Willyana, pelo constante incentivo e compreensão, que tem se mostrado uma verdadeira companheira, sempre conseguindo me acalmar nos momentos difíceis, e por todos os momentos que já vivemos. Sem você, meus dias não seriam os mesmos, nem as vitórias teriam os mesmos sabores.

Aos meus irmãos Sales e Jorge, por compreenderem minha ausência nas questões familiares. Agradeço, também, às minhas irmãs Carla e Carminha, pelo carinho e por todos os momentos felizes.

Ao meu orientador e amigo Milton Mendes, que sempre se mostrou preocupado e dedicado em passar seu conhecimento, por todas as oportunidades que me foram proporcionadas e pelos conselhos pessoais, que se mostraram muito importante em momentos difíceis. A minha orientadora Cecília Dias Flores que, apesar da distância, vem colaborando com este trabalho, sempre disposta a contribuir.

Gostaria de agradecer a todos os professores que contribuíram para a construção de um curso de tão alto nível e que sempre mostraram dedicação e prontidão ao curso e aos alunos.

Aos amigos e colegas do LES, em especial Ferdinandy e Antonio Farias por todas as proezas que conseguimos realizar juntos, pela colaboração, amizade e incentivo.

Aos colegas e professores do mestrado, com os quais adquiri não apenas conhecimentos técnicos necessários para realização deste trabalho, mas também valores que irão me nortear durante toda a vida.

À CAPES, pelo apoio financeiro que viabilizou a realização deste trabalho.

“Conhecer o homem - esta é a base de todo o sucesso” (Charles Chaplin).

RESUMO

Paralelo ao desenvolvimento e popularização dos recursos tecnológicos, surgiu a preocupação, por parte dos educadores, de buscar meios de adotar novas tecnologias no âmbito educacional, o que ocasionou consequências relevantes para os processos de ensino e aprendizagem. Essa evolução promoveu o surgimento de novas abordagens para o ensino não presencial, sendo uma delas a aprendizagem móvel. Esta última traz grande contribuição para aprendizagem informal, pois favorece a autoaprendizagem, as relações sociais, o compartilhamento de experiência, a colaboração e o acesso a uma diversidade de conteúdos educacionais. Contudo, por facilitar o acesso a diversos conteúdos, essa modalidade de ensino frequentemente fornece informações muitas vezes inadequadas ao contexto do usuário. A modalidade de ensino-aprendizagem denominada de aprendizagem ubíqua minimiza esse problema ao utilizar-se de técnicas computacionais para suporte à aprendizagem móvel sensível ao contexto do usuário. No entanto, ao elaborar um meio personalizado de aquisição de conhecimento, é necessário obter informações relevantes para traçar o perfil dos usuários e, assim, identificar suas necessidades. Os chamados traços digitais possuem um grande potencial a ser explorado, trazendo experiências reais do usuário, de maneira a viabilizar a personalização do ambiente. Desta forma, este trabalho apresenta um ambiente de aprendizagem ubíqua, chamado de MobiLE+. Esse ambiente é capaz de fornecer conteúdos educacionais adequados às características particulares dos usuários, levando em consideração seu perfil, o contexto atual e o histórico de interações no meio virtual. Essas informações são capturadas de forma dinâmica, autônoma e transparente para os usuários, através de um Sistema Multiagente (SMA).

Palavras-Chave: Aprendizagem Móvel. Ambiente Sensível ao Contexto. Aprendizagem Ubíqua. Aprendizagem Informal. Sistemas Multiagente. Enriquecimento Semântico.

ABSTRACT

Parallel to the development and popularization of technological resources, emerged a concern on the part of educators to seek ways to add new technologies in the educational field, which led to important consequences for the processes of teaching and learning. This evolution has promoted the emergence of new approaches to non-presence education, one of them mobile learning. The last brings great contribution to informal learning because it facilitates the self-learning, social relationships, experience sharing, collaboration and access to a variety of educational contents. However, by facilitating the access to several contents, this teaching modality often provides inappropriate information to the context of the user. The modality of teaching and learning called ubiquitous learning minimizes this problem by utilizing computational techniques to support mobile learning context-sensitive the user. However, when developing a personalized means of acquiring knowledge is necessary obtain relevant information of profile the users and thus identify their needs. The so-called digital traces have great potential to be explored, bringing real user experiences in order to enable the customization of the environment. Thus, this work presents a ubiquitous learning environment, called Mobile+. This environment is able to provide appropriate content the particular characteristics of users, taking into account their profiles, the current context and history of interactions in the virtual environment. This information is captured in a dynamic, autonomous and transparent to users way, through a Multi-Agent System (MAS).

Keywords: Mobile Learning. Context-sensitive Environment. Ubiquitous Learning. Informal Learning. Multi-agent Systems. Semantic Enrichment.

LISTA DE TABELAS

Tabela 1 - Dimensões de Estilos de Aprendizagem. Fonte: Adaptado de (GRAF; KINSHUK, 2010).....	29
Tabela 2 - Planilha de cálculo do ILS. Fonte: Adaptado de (FELDER; SOLOMAN, 2004). .	30
Tabela 3 – Principais Entidades do Modelo de Dados do MobiLE+	62
Tabela 4 - URIs de acesso ao recurso <i>Person</i>	72
Tabela 5 - <i>Template</i> textual do Agente Coletor.....	83
Tabela 6 - <i>Template</i> textual do Agente Usuário.....	85
Tabela 7 - <i>Template</i> textual do Agente Recomendador.	86
Tabela 8 - <i>Template</i> textual do Agente de Interface.....	88
Tabela 9 - <i>Template</i> textual do Agente DF	89

LISTA DE FIGURAS

Figura 1 - Representação Conceitual da Aprendizagem Ubíqua.....	26
Figura 2 - Dimensões do ILS. Fonte: Adaptado de (FELDER; SOLOMAN, 2004).	31
Figura 3 - Visão Genérica de Agente. Fonte: Adaptado de (ARTERO, 2009).	32
Figura 4 - Diagrama de componentes da arquitetura YouUbi.....	38
Figura 5 - Diagrama de Componentes do cliente.	42
Figura 6 - Menu principal da aplicação.	43
Figura 7 - Tela de Login.....	44
Figura 8 - Notificação de Conteúdo Recomendado.....	44
Figura 9 - Declaração do <i>BroadcastReceiver</i> que monitora mudança no estado da bateria. ..	47
Figura 10 - Trecho de código da classe <i>BatteryStatus</i>	47
Figura 11 - Permissão para Leitura do Histórico de Navegação e Favoritos.	48
Figura 12 - Trecho de código da classe <i>VisitHistory</i>	48
Figura 13 - Permissões de Acesso à Localização.	50
Figura 14 - Trecho de código da classe <i>CurrentLocation</i>	51
Figura 15 - Permissão para leitura de informações do telefone.	52
Figura 16 - Trecho de código da classe <i>DeviceInfo</i>	53
Figura 17 - Permissões relacionadas à conectividade.	54
Figura 18 - Declaração do <i>ConnectivityReceiver</i> que monitora alterações no <i>status</i> de conexão.....	54
Figura 19 - Trecho de código da classe <i>ConnectivityStatus</i>	54
Figura 20 - Trecho de código da classe <i>YouTubaManager</i>	55
Figura 21 - Tela de Visualização de Vídeos.....	56
Figura 22 - Tela de Comentário e Avaliação sobre o Vídeo.	56
Figura 23 - Aplicação do Questionário ILS.	57
Figura 24 - Representação dos Métodos Padrões do Serviço Rest.....	59
Figura 25 - Modelo do Banco de Dados Relacional.....	61
Figura 26 - Diagrama de Classes da Implementação dos Recursos.	71
Figura 27 - Exemplo de uma Requisição para Consumo do Serviço	72
Figura 28 - Trecho de Código da Classe Abstrata <i>Resource</i>	73
Figura 29 - Trecho de Código da Classe <i>PersonResource</i>	73
Figura 30 - Trecho de Código da Classe <i>Facade</i>	74

Figura 31 - Trecho de Código nº 1 da Classe <i>ConnectionWebService</i>	75
Figura 32 - Trecho de Código nº 2 da Classe <i>ConnectionWebService</i>	76
Figura 33 - Trecho de Código nº 3 da Classe <i>ConnectionWebService</i>	77
Figura 34 - Mensagem de Saída do Servidor de uma Operação bem sucedida.....	78
Figura 35 - Mensagem de Saída do Servidor de um Erro Durante a Operação	78
Figura 36 - Diagrama de Tarefas do SMA.	81
Figura 37 - Modelo de Interação do SMA.....	82
Figura 38 - Diagrama de Atividades do Agente Coletor.	84
Figura 39 - Diagrama de Atividades do Agente Usuário.	86
Figura 40 - Diagrama de Atividades do Agente Recomendador.....	87

LISTA DE SIGLAS

ACL	<i>Agent Communication Language</i>
API	<i>Application Programming Interface</i>
APs	<i>Agent Platforms</i>
CAg	<i>Collector Agent</i>
DF	<i>Directory Facilitator</i>
EaD	Educação a Distância
ESOA	Engenharia de Software Orientada a Agentes
FIPA	<i>Foundation for Intelligent Physical Agents</i>
FSLSM	<i>Felder-Silverman Learning Styles Model</i>
GCM	<i>Google Cloud Messaging</i>
GPS	<i>Global Position System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IAg	<i>Interface Agent</i>
ILS	<i>Index of Learning Styles</i>
IP	<i>Internet Protocol</i>
JADE	<i>Java Agent Development Framework</i>
Java EE	<i>Java Enterprise Edition</i>
JSON	<i>JavaScript Object Notation</i>
JPA	<i>Java Persistence API</i>
JSR	<i>Java Specification Requests</i>
LBC	Lista Baseada em Conteúdo
LHU	Lista Híbrida Ubíqua
LPC	Lista Colaborativa Ponderada
<i>M-Learning</i>	<i>Mobile Learning</i>
MAC	<i>Media Access Control</i>
MEC	Ministério da Educação
MILOS	<i>Multiagent Infrastructure for Learning Object Support</i>
MobiLE	<i>Mobile Learning Environment</i>
MTP	<i>Message Transport Protocol</i>
MTS	<i>Message Transport Service</i>
OAs	Objetos de Aprendizagem

OBAA	Objetos de Aprendizagem Baseados em Agentes
PDA	<i>Personal Digital Assistant</i>
SCROLL	<i>System for Capturing and Reminding of Learning Log</i>
SEED	Secretaria de Educação a Distância
SES	Sistema de Enriquecimento Semântico
SRHU	Sistema de Recomendação Híbrida Ubíqua
SMA	Sistema Multiagente
SR	Sistema de Recomendação
SRBC	Sistema de Recomendação Baseada em Conteúdo
SRH	Sistema de Recomendação Híbrida
SW	Serviço Web
RAg	<i>Recommender Agent</i>
TDs	Traços Digitais
<i>U-Learning</i>	<i>Ubiquitous Learning</i>
UAg	<i>User Agent</i>
UPLE	<i>Ubiquitous Personal Learning Environment</i>
URI	<i>Uniform Resource Identifier</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	CONTEXTUALIZAÇÃO	16
1.2	PROBLEMÁTICA	17
1.3	OBJETIVO	19
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO.....	20
2	REFERENCIAL TEÓRICO	21
2.1	APRENDIZAGEM INFORMAL.....	21
2.2	APRENDIZAGEM UBÍQUA	22
2.2.1	Educação a Distância	22
2.2.2	Aprendizagem Móvel	23
2.2.3	Ambiente Sensível ao Contexto	23
2.2.4	Aprendizagem Ubíqua	25
2.3	ENRIQUECIMENTO SEMÂNTICO	26
2.4	SISTEMAS DE RECOMENDAÇÃO	27
2.5	ESTILOS DE APRENDIZAGEM	28
2.5.1	Dimensões de Estilos de Aprendizagem	28
2.5.2	Índice de Estilos de Aprendizagem.....	30
2.6	AGENTES E SISTEMA MULTIAGENTE.....	32
2.6.1	Tipos de Agentes.....	32
2.6.2	Propriedades dos Agentes.....	33
2.6.3	Tipos de Ambientes	34
2.6.4	Sistemas Multiagente	35
3	MOBILE+: UM SISTEMA DE SUPORTE À APRENDIZAGEM UBÍQUA.....	37
3.1	DESCRIÇÃO GERAL DO SISTEMA	37
3.2	DESCRIÇÃO DO CLIENTE ANDROID.....	42
3.2.1	Informações Capturadas do Usuário.....	45
3.3	IDENTIFICAÇÃO DO ESTILO DE APRENDIZAGEM.....	57
3.4	ARQUITETURA ORIENTADA A SERVIÇO.....	58
3.5	CAMADA DE PERSISTÊNCIA DE DADOS	60
4	MODELAGEM E IMPLEMENTAÇÃO DO SERVIÇO WEB.....	71

4.1	PROVEDOR DO SERVIÇO	71
4.2	CONSUMIDOR DO SERVIÇO	74
4.3	PROTOCOLO DE COMUNICAÇÃO.....	77
5	MODELAGEM DO SISTEMA MULTIAGENTE	79
5.1	MODELO DE TAREFAS	80
5.2	MODELO DE INTERAÇÃO.....	81
5.3	MODELO DO AGENTE COLETOR.....	83
5.4	MODELO DO AGENTE USUÁRIO.....	84
5.5	MODELO DO AGENTE RECOMENDADOR.....	86
5.6	MODELO DO AGENTE DE INTERFACE	88
5.7	MODELO DO AGENTE DF	88
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	91
7	REFERÊNCIAS.....	93
	ANEXO A – QUESTIONÁRIO ORIGINAL DO ILS	99

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Os avanços das tecnologias móveis e de comunicação sem fio contribuíram para demarcar um novo horizonte aos processos de ensino e aprendizagem já estabelecidos. Os dispositivos móveis possibilitam o acesso a ambientes virtuais de ensino e aprendizagem, caracterizando, assim, a aprendizagem móvel, do inglês *Mobile Learning* ou *m-learning* (SACCOL et al., 2010).

A aprendizagem móvel permite que a aprendizagem ocorra a qualquer hora e em qualquer lugar. Porém, apesar de prover mobilidade, ela não fornece uma aprendizagem capaz de considerar informações sensíveis ao contexto do estudante (MANDULA et al., 2011). Ao considerarmos 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 (também conhecida como *u-learning*, de *ubiquitous learning*) (SACCOL et al., 2010).

Esse meio de oferecer ensino e aprendizagem permite que estudantes e professores possam tirar vantagens dos recursos oferecidos pelas tecnologias móveis, dentre as quais se destaca a possibilidade de acessar, visualizar e prover conteúdo independentemente do horário e a partir de qualquer localidade (AHMED; CHANG, 2012). Entretanto, as tecnologias móveis promovem o acesso a uma ampla quantidade e diversidade de conteúdo educacional, o que pode acarretar em perda de desempenho dos estudantes, por provocar a dispersão destes. Para lidar com essa situação, ao fornecer conteúdo educacional, devem-se levar em consideração as características particulares de cada estudante, como também as restrições dos seus dispositivos móveis, uma vez que os mesmos possuem recursos distintos e limitados. Também deve-se considerar a localização do estudante, visto que as oportunidades de aprendizagem podem surgir no momento que ocorre a exploração de diversos ambientes físicos.

Modelos de aprendizagem modernos consideram experiências do mundo real. As mídias sociais facilitam a obtenção de dados do mundo real, ao fornecer experiências reais dos seus usuários, experimentadas através de um meio digital (THAKKER et al., 2012). Esses conteúdos gerados pelos usuários incorporam, pela sua própria natureza, diferentes pontos de

vista sobre determinados aspectos do mundo real. A captura semântica e identificação de diversos pontos de vista possuem um grande potencial para auxiliar a personalização de ambientes de aprendizagem (DESPOTAKIS et al., 2012).

A fim de desenvolver um ambiente de aprendizagem que melhor atenda as necessidades dos estudantes e que busque o aprimoramento do processo de ensino-aprendizagem, é fundamental considerar os aspectos relacionados a situações reais do cotidiano do estudante, transcendendo as barreiras do formalismo educacional. A aprendizagem ubíqua pode auxiliar a educação informal fornecendo um meio favorável para autoaprendizagem e interações do estudante com o ambiente real, através de experiências obtidas pelas mídias sociais.

1.2 PROBLEMÁTICA

A aprendizagem móvel é uma modalidade de ensino-aprendizagem não presencial muito conveniente, pois ultrapassa as restrições de tempo e espaço. Podemos citar, também, seu grande potencial de aplicação, tendo em vista a enorme disseminação dos dispositivos móveis em todo o mundo (AHMED; CHANG, 2012). Um dos desafios da aprendizagem móvel é a adequação do conteúdo ao estudante de acordo com as características cognitivas deste. No ensino presencial, esta já é uma tarefa difícil de ser realizada, visto que é necessário que o professor possua certo conhecimento das necessidades de cada um dos estudantes.

Por facilitar o acesso a diversos conteúdos educacionais, essa modalidade de ensino frequentemente fornece conteúdo inadequado ao contexto do estudante. Uma forma de minimizar esse problema é a utilização de técnicas computacionais para suporte à aprendizagem ubíqua. Essa modalidade de ensino-aprendizagem se beneficia das características proporcionadas pelo *m-learning*, além de possuir a capacidade de oferecer recursos educacionais adequados às características particulares do estudante.

As mídias sociais fornecem uma grande quantidade de informações de experiências reais vivenciadas por seus usuários, denominadas de Traços Digitais (TDs). Os TDs podem se tornar recursos valiosos ao campo educacional, fornecendo uma compreensão de um determinado domínio explorado durante o processo de aprendizagem (THAKKER et al., 2012).

Os ambientes virtuais de aprendizagem, em geral, dedicam maior atenção aos aspectos relacionados à aprendizagem formal, ou seja, relacionados à distribuição de conteúdos especificamente confeccionados para cursos fornecidos por instituições de ensino. No entanto, quando falamos de aprendizagem ubíqua, não podemos desconsiderar a importância da aprendizagem informal, ou seja, aquela obtida através do acesso a conteúdos não relacionados a cursos formais, uma vez que esta faz parte da vida cotidiana do estudante. Neste cenário, oportunidades pedagógicas podem surgir à medida que o estudante interage com as mídias sociais e se desloca por diferentes ambientes físicos. Estes ambientes podem possuir informações com potencial de auxiliar o estudante a alcançar seus objetivos pedagógicos, no contexto tanto da educação formal como da educação informal.

O projeto de ambientes de aprendizagem ubíqua já é um tema bastante abordado em pesquisas acadêmicas. LI et al. (2012) desenvolveram o sistema SCROLL (*System for Capturing and Reminding of Learning Log*). Este permite aos usuários registrarem suas experiências de aprendizagem, associando a elas fotos, áudio, vídeos, localização, dados de sensores e outras informações. O SCROLL tem o propósito de ajudar o usuário a manter suas experiências de aprendizagem, registrá-las por meio do contexto, compartilhá-las com outros usuários, identificar hábitos de aprendizagem e apoiar a aprendizagem de acordo com os hábitos pessoais de aprendizagem dos usuários. Atualmente, o SCROLL está focado no campo de aprendizagem de línguas.

Gluz e Vicari (2010) propuseram a MILOS (*Multiagent Infrastructure for Learning Object Support*), uma infraestrutura, combinando ontologias¹ e agentes, que implementa as funcionalidades necessárias aos processos de autoria, gerência, busca e disponibilização de OAs compatíveis com a proposta de padrão de metadados de Objetos de Aprendizagem (OAs) OBAA (Objetos de Aprendizagem Baseados em Agentes). Outras iniciativas vêm sendo realizadas para aperfeiçoar o MILOS. Como em Jardim da Silva (2013), que desenvolveu um mecanismo de busca semântica de objetos de aprendizagem integrado ao MILOS.

Já Taraghi (2012) apresenta o UPLE (do inglês *Ubiquitous Personal Learning Environment*). Trata-se de um ambiente que favorece a autonomia do aprendiz, por meio de uma interface baseada em *widgets*². O usuário possui total liberdade para gerenciar e personalizar o ambiente, escolhendo produtos e serviços de que necessite para conseguir cumprir com suas metas de aprendizagem. A ubiquidade do ambiente ocorre devido sua

¹ Especificação formal explícita de um determinado domínio de conhecimento, com estruturas e detalhes semânticos (MOORE et al., 2009).

² *Widgets* representam recursos independentes, serviços e aplicações que são integradas no UPLE.

disseminação em várias plataformas, inclusive móvel, e sua personalização acontece por intervenção direta do usuário e não de forma automatizada, como proposto neste trabalho.

GICQUEL et al. (2013) apresentam um projeto de um ambiente de aprendizagem ubíqua chamado CALM (*Contextualized Learning in Mobility*), com a finalidade de prover aprendizagem durante visitas escolares a museus. Utiliza modelos semânticos para representar o domínio da aprendizagem (patrimônio cultural) e o contexto no qual ela ocorre, para promover experiências mais contextualizadas aos alunos. No entanto, o CALM está limitado a uma aplicação específica e não atende aos critérios da aprendizagem informal

Ao projetar um ambiente virtual de aprendizagem que atenda à nova dinâmica educacional, imposta pelo avanço tecnológico, é essencial considerar a infinidade de conteúdos aos quais o estudante vem sendo exposto. Desta forma, não é interessante se limitar à utilização de OAs ou a uma aplicação específica, mas sim buscar atender a diversos recursos e aplicações, uma vez que estes podem contribuir para que o estudante alcance seus objetivos. As mídias sociais possuem um grande potencial a ser explorado, de maneira a tornar o ambiente de aprendizagem mais próximo da realidade do usuário, favorecendo a aprendizagem informal.

1.3 OBJETIVO

Tendo em vista a problemática apresentada, o presente trabalho propõe o desenvolvimento de um ambiente de aprendizagem ubíqua capaz de adequar-se às características particulares dos estudantes, fornecendo conteúdo adequado às suas necessidades pedagógicas, sem interferir na sua rotina de estudo, interação social e profissional.

Para isso, foi desenvolvido um ambiente de aprendizagem ubíqua capaz de capturar os diversos conteúdos gerados pelos usuários, através de experiências cotidianas, para identificar seu contexto e auxiliar no processo de enriquecimento semântico (*semantic augmentation*), de modo a aperfeiçoar o processo de recomendação de recursos educacionais e, conseqüentemente, a personalização do ambiente.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

O Capítulo 2 apresenta o referencial teórico necessário para o entendimento desse trabalho, abordando os seguintes temas: Aprendizagem Informal, Aprendizagem Ubíqua, Enriquecimento Semântico, Sistemas de Recomendação, Estilos de Aprendizagem e Agentes e Sistema Multiagente. No Capítulo 3, é feita uma descrição geral do MobiLE+, abordando toda a arquitetura do ambiente e posicionando o escopo deste trabalho. Ainda neste capítulo são apresentados os componentes e funções para o desenvolvimento da aplicação Android, o processo de identificação dos estilos de aprendizagem, a arquitetura orientada a serviço, a camada de persistência e as tecnologias utilizadas. O Capítulo 4 é dedicado à modelagem e implementação do serviço web. O Capítulo 5 apresenta a modelagem dos agentes presentes no MobiLE+, detalhando os aspectos conceituais, bem como as atividades que cada um realiza no sistema. Finalmente, o Capítulo 6 apresenta as considerações finais acerca do que foi exposto nesta dissertação, bem como apresenta os trabalhos futuros identificados.

2 REFERENCIAL TEÓRICO

2.1 APRENDIZAGEM INFORMAL

Os ambientes de aprendizagem podem ser classificados em dois tipos: formal e informal. A aprendizagem formal ocorre sob a supervisão de um professor, enquanto que na aprendizagem informal o aluno é responsável pelo seu aprendizado, utilizando-se de informações cotidianas do seu ambiente (JIUGEN et al., 2011; WANG; SHEN, 2012). Portanto, a aprendizagem informal ocorre em todas as fases da vida, tendo como principais características (JIUGEN et al., 2011):

- **Autonomia:** o estudante é responsável por sua aprendizagem, onde ele determina o objetivo a ser alcançado, o conteúdo a ser abordado e como este será processado.
- **Conhecimento:** obtido por meio das interações sociais e profissionais.
- **Diversidade:** a fonte de informação se apresenta de várias formas, sem que seja necessário um local determinado. A aquisição de conhecimento pode ser obtida através de experiências diárias, recursos do ambiente, bibliotecas, redes sociais, dentre outros.

As tecnologias móveis, quando utilizadas em ambientes de aprendizagem formal, reforçam a ligação entre o aluno, o conteúdo e o professor, fornecendo estímulo e motivação. Neste cenário, os educadores têm o desafio de elaborar conteúdos educativos para os dispositivos móveis dos alunos, com o objetivo de maximizar os resultados da aprendizagem.

Como a aprendizagem informal ocorre, em determinadas ocasiões, de forma ocasional e inconsciente, as tecnologias móveis podem auxiliar o processo de aprendizagem fornecendo um ambiente que favoreça a autoaprendizagem, relações sociais, o compartilhamento de experiências e a aprendizagem colaborativa. Através dos dispositivos móveis, os alunos podem adquirir conhecimento para tomar decisões e resolver problemas do contexto real.

2.2 APRENDIZAGEM UBÍQUA

Nesta seção serão apresentados alguns conceitos necessários para o entendimento de um ambiente de aprendizagem ubíqua. Serão mostrados também conceitos relacionados à Educação a Distância (EaD) e algumas formas de se prover este tipo de ensino. A Seção 2.2.1 traz conceitos básicos relacionados à EaD. A Seção 2.2.2 mostra conceitos e aspectos relacionados à aprendizagem móvel, uma das formas de se prover a EaD. A Seção 2.2.3 mostra conceitos inerentes a ambientes sensíveis ao contexto. A Seção 2.2.4 explica o que vem a ser a aprendizagem ubíqua propriamente dita.

2.2.1 Educação a Distância

Diferentes conceitos de EaD podem ser encontrados na literatura. Entretanto, para o presente trabalho, será adotada uma definição elaborada pela Secretaria de Educação a Distância (SEED) do MEC (Ministério da Educação), a qual pode ser encontrada no Decreto 5.622/2005 (BRASIL, 2005). Com o objetivo de instruir os sistemas de ensino, esse decreto define, em seu artigo 1º, a EaD como sendo uma

[...] modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos (BRASIL, 2005).

De fato, conforme percebe-se através da definição apresentada, não há um modelo único de EaD, podendo ser consideradas como tal várias formas de ensino e aprendizagem. A natureza do curso, as condições reais do cotidiano e as necessidades dos estudantes são os elementos que irão definir a tecnologia e a metodologia mais adequadas, assim como a definição dos momentos presenciais necessários e obrigatórios (previstos em lei), estágios supervisionados, práticas em laboratórios de ensino, tutorias presenciais nos polos descentralizados de apoio presencial e outras estratégias (SEED/MEC, 2007). Porém, dentre as várias formas de EaD, pode-se destacar a aprendizagem realizada através do uso de dispositivos móveis.

2.2.2 Aprendizagem Móvel

O advento do dispositivo móvel trouxe novas perspectivas para a humanidade e tornou possível ampliar o domínio da educação mediada por computador, onde aprender fora dos limites do ambiente educacional tradicional tornou-se uma nova realidade (FAZLINA et al., 2013).

A aprendizagem móvel refere-se ao uso de tecnologias portáteis e móveis, tais como computadores de bolso, PDAs (*Personal Digital Assistants*), telefones celulares, *smartphones*, *tablets*, em atividades de aprendizagem, na organização de ideias e no acesso a informações de forma eficiente (MANDULA et al., 2013). Essas atividades podem ser realizadas a partir de qualquer lugar e a qualquer hora, trazendo informação e conhecimento para a situação e o lugar onde a aprendizagem acontece (FAZLINA et al., 2013).

Os dispositivos móveis estão cada vez mais presentes no nosso dia-a-dia, podendo ser empregados no trabalho, no ensino e no lazer. De acordo com Sharples (2000), citado por Fazlina et al. (2013), os dispositivos móveis podem ser visto especialmente a partir da perspectiva de aprendizagem ao longo da vida.

As tecnologias móveis têm apreciado um momento de grandes avanços, superando as limitações de outrora, não mais ficando a desejar quanto à capacidade dos computadores pessoais. No entanto, outros desafios surgem para mitigar as lacunas provenientes desse avanço. Desafios sociais, pedagógicos e educacionais, incluindo o acesso a materiais de aprendizagem fora da sala de aula, desenvolvimento de conteúdo e suporte a dispositivos passam a ser considerados antes de se desenvolver qualquer aplicação para o *m-learning* (MANDULA et al., 2013).

2.2.3 Ambiente Sensível ao Contexto

Sensibilidade ao contexto descreve um paradigma no qual o contexto de um usuário é levado em consideração para definir o seu perfil (MOORE et al., 2009; MOORE; PHAM, 2012). Não existe um consenso a respeito da definição de “contexto”, sendo este específico da

aplicação e da intenção desejadas, requerendo a identificação das funções e propriedades dos domínios dos indivíduos (MOORE et al., 2009). Não há como definir de forma precisa o que seria o contexto. Dessa forma, as características que serão consideradas para construção do contexto dependem da situação específica.

O contexto pode ser definido de acordo com informações relativas a propriedades que se combinam para descrever e caracterizar uma entidade e seu papel de uma forma legível pelo computador (MOORE et al., 2009; MOORE; PHAM, 2012). A localização do estudante, por exemplo, é uma característica importante para a definição do seu contexto em um ambiente para aprendizagem móvel. Entretanto, o contexto inclui mais do que apenas a localização. De fato, quase todas as informações disponíveis no momento da interação podem ser vistas como informações contextuais, dentre as quais se destacam (MOORE et al., 2009):

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

A lista acima, embora não contenha exatamente todas as informações que podem ser consideradas, serve para demonstrar a complexidade inerente ao contexto, sua natureza de domínio específico e a dificuldade em defini-lo e medi-lo (MOORE, et al., 2009). Na tentativa de diminuir essa complexidade, Moore et al. (2008) definem dois tipos gerais de contexto:

- 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.
- 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, 2010). Entretanto, os trabalhos encontrados na literatura não levam

em consideração os recursos físicos dos dispositivos móveis, o que compromete uma definição mais precisa do contexto dos estudantes e, conseqüentemente, o acesso e a navegação adequados nos conteúdos recomendados, uma vez que isto é diretamente influenciado pelas características dos dispositivos móveis utilizados pelos estudantes.

2.2.4 Aprendizagem Ubíqua

Mark Weiser (1991) idealizou o mundo da computação ubíqua, onde o computador deixa de ser usado como um dispositivo de propósito geral, passando a ter uma nova visão de sua funcionalidade, atendendo as necessidades específicas de cada usuário. Neste cenário, um usuário não se restringe a acessar um único computador, mas vários dispositivos computacionais interligados entre si e disseminados de forma quase imperceptível pelo ambiente.

A partir do conceito apresentado pode-se definir a aprendizagem ubíqua como sendo a utilização de dispositivos móveis, tecnologias de comunicação móvel sem fio, sensores e mecanismo de localização, com o objetivo de auxiliar o processo educacional, levando em consideração características particulares dos estudantes (SACCOL et al., 2010).

Desta forma, 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 fornece um serviço de forma transparente aos estudantes. A Figura 1 mostra um diagrama representando a aprendizagem ubíqua como a intersecção entre a aprendizagem móvel e a sensibilidade ao contexto.

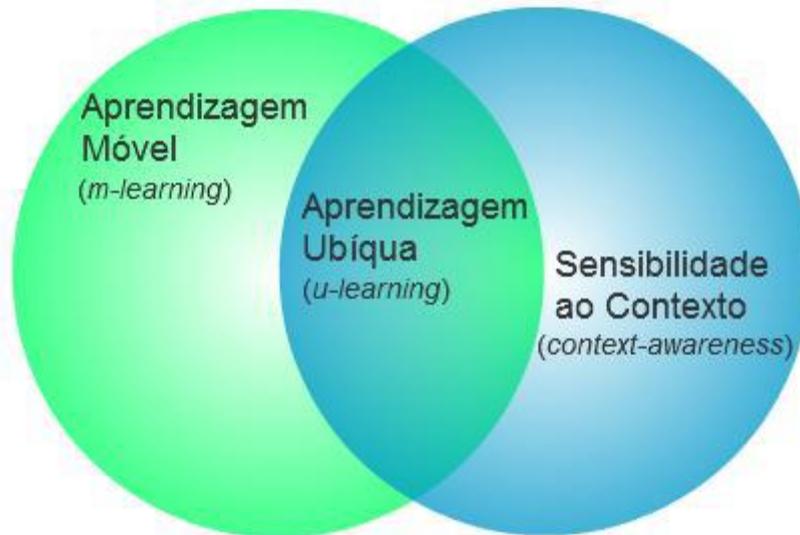


Figura 1 - Representação Conceitual da Aprendizagem Ubíqua

A mobilidade permitida ao usuário devido 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.) (SACCOL et al., 2010).

2.3 ENRIQUECIMENTO SEMÂNTICO

O notório crescimento das mídias sociais proporciona o aumento da imensa quantidade de conteúdos gerados por seus usuários. Esses conteúdos têm influenciado radicalmente as técnicas aplicadas em muitas áreas.

Segundo Redecker et al. (2010) citado por Thakker et al. (2012), tendências apontam que a mídia social irá ter um forte impacto sobre a aprendizagem informal, fornecendo conteúdo gerado pelo usuário com o objetivo de auxiliar o processo de aprendizagem. Redes Sociais podem oferecer diversas experiências reais:

- As pessoas podem comentar sobre experiências com outras pessoas, serviços ou lugares (shoppings, hotéis, praias etc.);
- Compartilhar histórias (*blogs*, livro, *twitter*);
- Comentar situações vivenciadas (vídeo, livro, viagem etc.).

A essas experiências reais experimentadas através das mídias sociais dá-se o nome de Traços Digitais (TDs) (THAKKER et al., 2012). Quando selecionados com atenção, os TDs possuem grande potencial para a aprendizagem. Ambientes de aprendizagem modernos podem explorar conteúdo gerado pelo usuário para proporcionar situações relacionadas com sua experiência do mundo real.

Os conteúdos gerados pelas redes sociais, geralmente, não possuem o domínio de aprendizagem bem definido, podendo apresentar múltiplas interpretações. No entanto, têm surgido novas arquiteturas para criar, agregar, organizar e explorar os TDs em situações de aprendizagem informal (THAKKER et al., 2012).

O enriquecimento semântico possibilita estender a compreensão do domínio de um determinado TD. Para isso, são realizados a identificação e o enriquecimento de termos-chave a partir do conteúdo textual (comentários), considerando termos relacionados. As formas superficiais enriquecidas podem ser semanticamente associadas a conceitos oriundos de uma ontologia (podendo ser de um contexto específico), a fim de realizar uma marcação semântica (DESPOTAKIS et al., 2012).

2.4 SISTEMAS DE RECOMENDAÇÃO

De acordo com Schafer et al. (1999), citado por Vieira e Nunes (2012), o aumento de meios de disponibilização de conteúdo, podendo ser produtos ou informação, através de sistemas Web, provoca uma situação onde o usuário possui muitas opções de escolha antes mesmo de estar apto a selecionar uma opção que atenda suas necessidades. Os Sistemas de Recomendação (SR) buscam amenizar os impactos gerados por essa sobrecarga de informação.

Os SRs utilizam repositórios de informação e dados de preferência dos usuários para direcionar conteúdos aos indivíduos com potenciais interesses. Um dos desafios dos SRs é realizar a indicação de produtos, serviços e/ou informação que melhor atendam as expectativas dos usuários (e ao seu perfil) (CAZELLA et al., 2011).

Segundo Vieira e Nunes (2012), os SRs podem ser classificados em três tipos: Sistema de Recomendação Baseada em Conteúdo (SRBC), Sistema de Recomendação Colaborativa (SRC) e Sistema de Recomendação Híbrida (SRH). A recomendação baseada em conteúdo é realizada através de informações recuperadas do usuário (perfil,

comportamento) e/ou do seu histórico de escolhas. Já a recomendação por filtragem colaborativa leva em consideração escolhas realizadas por usuários com características similares. Por último, o SRH efetua a recomendação através da aplicação das técnicas utilizadas pelos SRBC e SRC, de forma a atingir um maior número de possibilidades e sugerir objetos que diretamente não aparentam estar relacionados (MARTINS et al., 2011).

No âmbito da educação ubíqua, os alunos são expostos a uma grande quantidade de recursos educacionais e podem levar um bom tempo para realizar escolhas difíceis. Tendo em vista que os estudantes estão em processo de formação, estes não estão aptos o suficiente para realizar tais escolhas. Um SR vem a suprir esta carência recomendando recursos educacionais apropriados e tirando parte da responsabilidade de escolha do estudante, que passará a lidar com um volume menor informação.

2.5 ESTILOS DE APRENDIZAGEM

Existem na literatura diversos modelos de estilos de aprendizagem, contudo pesquisas como as de Kuljis e Liu (2005), Carver et al. (1999), citado por Graf e Kinshuk (2010), têm mostrado que, ao se tratar de ambientes educacionais adaptativos, o Modelo de Estilos de Aprendizagem Felder-Silverman (do inglês *Felder-Silverman Learning Styles Model - FLSM*) (FELDER; SILVERMAN, 1988) pode ser visto como o mais utilizado para classificação de estilos de aprendizagem.

2.5.1 Dimensões de Estilos de Aprendizagem

De acordo com o modelo FLSM, o aluno pode ser caracterizado em quatro dimensões de estilos de aprendizagem. Estas dimensões podem ser identificadas respondendo-se às seguintes perguntas (GRAF; KINSHUK, 2010):

- Que tipo de informação o aluno preferencialmente percebe? (sensitivo ou intuitivo);
- Através de qual canal sensorial uma informação externa é mais eficientemente recebida? (visual ou verbal);

- Como o aluno prefere processar a informação? (ativamente ou reflexivamente); e
- Como o aluno progride seu entendimento no assunto? (sequencialmente ou globalmente).

Na Tabela 1 são apresentadas as quatro dimensões de estilos de aprendizagem, segundo o modelo FSLSM.

Tabela 1 - Dimensões de Estilos de Aprendizagem. Fonte: Adaptado de (GRAF; KINSHUK, 2010).

DIMENSÃO	ESTILO DE APRENDIZAGEM	PERFIL DO ALUNO
Percepção	Sensitivo	Gosta de aprender a partir de materiais concretos e tende a ser mais prático.
	Intuitivo	Prefere materiais abstratos, como teorias e seus significados, e costuma ser mais inovador.
Entrada	Visual	Lembra-se melhor daquilo que usualmente vê, como imagens, gráficos, animações, etc.
	Verbal	Maior facilidade em lembrar-se de palavras, sejam elas escritas ou orais.
Processamento	Ativo	Prefere aprender por meio de suas experiências. Geralmente trabalha bem com outros alunos.
	Reflexivo	Diferentemente do ativo, prefere aprender sozinho refletindo sobre o assunto em questão.
Entendimento	Sequencial	Prefere aprender por meio de passos lineares, levando a um processo gradual de aprendizagem.
	Global	Aprende através de passos largos. É holístico, isto é, prefere entender primeiro o todo e em seguida suas partes.

2.5.2 Índice de Estilos de Aprendizagem

O Índice de Estilos de Aprendizagem, do inglês *Index of Learning Styles*® (ILS), é um questionário de 44 (quarenta e quatro) questões com o propósito de avaliar e indicar preferências de estilos de aprendizagem, baseando-se nas quatro dimensões do modelo FLSM (LITZINGER et al., 2007).

O ILS foi desenvolvido por Felder e Soloman (2004) na Universidade do Estado da Carolina do Norte em 1991. Desde então, inúmeros pesquisadores interessados em estilos de aprendizagem passaram a colaborar com o seu desenvolvimento, bem como passaram a utilizá-lo em suas pesquisas.

Com o aumento do interesse pelo método ILS, inúmeros trabalhos foram realizados com o intuito de validá-lo e verificar sua confiabilidade (ZYWNO, 2003; FELDER; SPURLIN, 2005; LITZINGER et al., 2007). Os resultados desses trabalhos mostraram que o ILS é estatisticamente confiável na tarefa de identificar estilos de aprendizagem. A Tabela 2 apresenta a planilha utilizada para auxiliar no cálculo do ILS.

Tabela 2 - Planilha de cálculo do ILS. Fonte: Adaptado de (FELDER; SOLOMAN, 2004).

Ativo/Reflexivo			Sensitivo/Intuitivo			Visual/Verbal			Sequencial/Global		
Q	a	b	Q	a	b	Q	a	b	Q	a	b
1	___	___	2	___	___	3	___	___	4	___	___
4	___	___	6	___	___	7	___	___	8	___	___
9	___	___	10	___	___	11	___	___	12	___	___
13	___	___	14	___	___	15	___	___	16	___	___
17	___	___	18	___	___	19	___	___	20	___	___
21	___	___	22	___	___	23	___	___	24	___	___
25	___	___	26	___	___	27	___	___	28	___	___
29	___	___	30	___	___	31	___	___	32	___	___
33	___	___	34	___	___	35	___	___	36	___	___
37	___	___	38	___	___	39	___	___	40	___	___
41	___	___	42	___	___	43	___	___	44	___	___
Soma Total											

a	B	a	b	a	b	a	b
_____	_____	_____	_____	_____	_____	_____	_____
(Maior – Menor) + Letra do Maior (Ex. 7a – 4b = 3a)							
_____	_____	_____	_____	_____	_____	_____	_____

A metodologia do ILS consiste em três etapas: (i) inicialmente o indivíduo analisado responde um questionário de 44 (quarenta e quatro) perguntas, as quais estão listadas em seu formato original no Anexo A. Cada pergunta possui apenas duas alternativas (*a* e *b*); (ii) em seguida, o sistema coleta as respostas, preenchendo uma planilha similar à apresentada na Tabela 2. Ao término do cálculo, ter-se-á um valor ímpar que varia de 1 a 11 seguido da letra predominante (*a* ou *b*); (iii) de posse do resultado, o avaliador o dispõe em um formulário similar ao apresentado na Figura 2.

Resultados entre 1 e 3 indicam uma preferência ligeiramente equilibrada entre os estilos dessa dimensão. Caso o resultado fique entre 5 e 7, há uma tendência moderada por um dos estilos. Já se o resultado ficar entre 9 e 11, o método defende que o avaliado possui uma preferência muito alta por um dos estilos, indicando uma dificuldade de aprendizagem em ambientes que não suportam esse estilo dominante.

ACT		REF
11a 9a 7a 5a 3a 1a		1b 3b 5b 7b 9b 11b
SEN		INT
11a 9a 7a 5a 3a 1a		1b 3b 5b 7b 9b 11b
VIS		VRB
11a 9a 7a 5a 3a 1a		1b 3b 5b 7b 9b 11b
SEQ		GLO
11a 9a 7a 5a 3a 1a		1b 3b 5b 7b 9b 11b

Figura 2 - Dimensões do ILS. Fonte: Adaptado de (FELDER; SOLOMAN, 2004).

2.6 AGENTES E SISTEMA MULTIAGENTE

Devido à grande variedade de paradigmas de agentes existentes na literatura, torna-se difícil conceituá-los de forma generalista, de modo que tal definição seja aceita nos mais diversos contextos.

Contudo, ao longo dos anos, surgiram na literatura várias iniciativas com esse intuito, como o trabalho de Franklin e Graesser (1997), que propôs uma definição formal de agente autônomo como sendo um sistema situado em um ambiente ao qual faz parte, percebendo-o e agindo sobre o mesmo, em função do tempo, e perseguindo sua própria agenda, bem como percebendo os efeitos que suas ações irão causar no futuro.

Em (RUSSELL e NORVIG (2010), um agente é tratado como qualquer entidade capaz de perceber seu ambiente através de sensores e atuar sobre ele por meio de atuadores.

Já Artero (2009), além de concordar com Russell e Norvig (2010), acrescenta que os agentes são programas que, de posse de uma representação do conhecimento e objetivos de um usuário, executam um conjunto de tarefas em seu lugar.

De modo geral, nesse trabalho, um agente pode ser genericamente descrito como qualquer entidade de hardware ou software que perceba seu ambiente, por meio de seus sensores, e execute ações sobre o mesmo, por meio de seus atuadores, de forma autônoma e proativa.

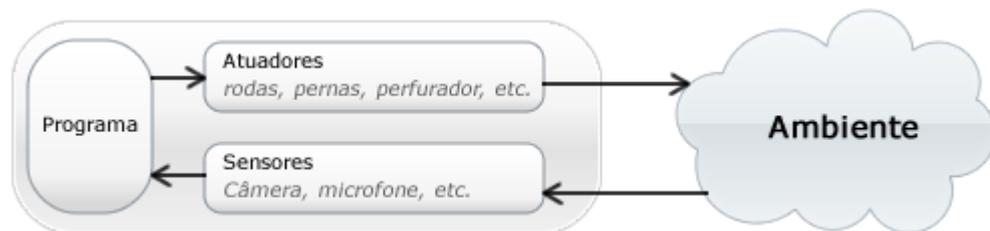


Figura 3 - Visão Genérica de Agente. Fonte: Adaptado de (ARTERO, 2009).

2.6.1 Tipos de Agentes

Trabalhos como (RUSSELL; NORVIG, 2010) e (ARTERO, 2009) classificam agentes nos seguintes tipos:

- **Agente Tabela:** é o tipo de agente mais simples. Ele mantém uma tabela interna com todas as possíveis percepções e as ações que elas devem disparar;
- **Agente Reativo Simples:** aquele que, baseado somente em sua percepção atual do ambiente, escolhe a ação mais adequada. A diferença em relação ao Agente Tabela é que, ao invés de ter uma tabela percepção-ação, ele possui um conjunto de regras condição-ação.
- **Agente Reativo Baseado em Modelo:** para compensar sua observação parcial do ambiente, esse agente possui um modelo interno, que mantém o histórico de suas ações e percepções. A deliberação é fruto da análise da percepção atual em função de seu modelo de mundo.
- **Agente Baseado em Objetivo:** aquele que delibera avaliando suas ações e seus impactos, observando se tais ações o aproximam de seus objetivos.
- **Agente Baseado em Utilidade:** aquele que escolhe suas ações também visando se aproximar de um objetivo, porém sempre tentando maximizar seu grau de “felicidade”, mensurado por meio de uma função de utilidade. Ou seja, tenta maximizar a eficiência de suas ações.
- **Agente com Aprendizagem:** agente capaz de se adaptar às mudanças no ambiente, por meio de mecanismos de aprendizagem.

2.6.2 Propriedades dos Agentes

Alguns pesquisadores, ao definirem agentes, utilizam-se de duas noções: uma fraca e outra forte. Segundo Wooldridge e Jennings (1995), a noção fraca é talvez a mais utilizada, a qual se refere a sistemas de hardware ou software que apresentam as seguintes propriedades:

- **Autonomia:** característica que confere ao agente o poder de executar suas próprias ações sem nenhuma intervenção humana direta;
- **Habilidade Social:** permite que os agentes de um ambiente se comuniquem através de uma Linguagem de Comunicação entre Agentes (*Agent Communication Language* - ACL) (FIPA, 2002) ou, por exemplo, por meio de um mecanismo conhecido como Quadro Negro, do inglês *blackboard*, o qual é explicado na Seção 2.6.4;

- **Reatividade:** como mencionado anteriormente, os agentes sentem o seu ambiente e reagem às mudanças nele ocorridas;
- **Proatividade:** os agentes não necessariamente atuam sobre um ambiente somente em resposta a um estímulo. Eles podem apresentar comportamentos orientados a objetivos, bem como ter iniciativa em certas ocasiões.

Já a noção forte de agente explora as propriedades listadas acima de uma forma mais profunda, utilizando-se de certas capacidades mentais encontradas nos humanos, como o conhecimento de domínio (ontologias), crenças, intenções, comprometimento, obrigações, etc (WOOLDRIDGE; JENNINGS, 1995).

Além dessas características básicas, outras características mais específicas podem ser atribuídas aos agentes (ARTERO, 2009):

- **Confiabilidade:** reflete a credibilidade que os agentes apresentam ao realizar suas tarefas, somada à certeza de que processará apenas informações verdadeiras;
- **Cooperatividade:** capacidade de trabalhar em conjunto com outros agentes, em prol da realização de uma tarefa em comum;
- **Degradação:** capacidade que o agente possui em resistir às anomalias no ambiente e concluir suas tarefas;
- **Inteligência:** capacidade de negociar em situações novas e de incertezas;
- **Mobilidade:** habilidade de poder se transportar de um ambiente para outro;
- **Persistência:** reflete o nível de coerência do agente ao longo do tempo. Ou seja, manter-se preciso ao longo do tempo;
- **Personalidade:** habilidade que o agente possui de poder demonstrar sua opinião.

2.6.3 Tipos de Ambientes

No contexto de agentes, ambiente é o espaço, físico ou virtual, no qual um ou mais agentes estão inseridos. Tais agentes sentem esse ambiente através de seus sensores e o modificam através de seus atuadores. Segundo Russell e Norvig (2010), o ambiente pode ser classificado como:

- **Acessível/Inacessível:** quando um agente possui total acesso ao estado atual de um ambiente, sem que para isso ele precise manter alguma estrutura ou memória interna, considera-se este ambiente acessível ou, caso contrário, inacessível.
- **Determinístico/Não Determinístico:** um ambiente é determinístico, do ponto de vista do agente, se seu próximo estado puder ser determinado pelo estado atual e pelas ações que este agente venha a escolher.
- **Episódico/Não Episódico:** em um ambiente episódico, toda a interação agente-ambiente é distribuída em episódios (passos, iterações, etc.), de modo que tais episódios sejam independentes entre si.
- **Estático/Dinâmico:** para distinguir um ambiente estático de um dinâmico, basta verificar se o estado desse ambiente muda com o passar do tempo. Havendo essa mudança, o ambiente é considerado dinâmico. Caso o ambiente não mude, mas alguma propriedade do agente varie com o passar do tempo, considera-se tal ambiente semi-dinâmico.
- **Discreto/Contínuo:** ambientes discretos são aqueles onde há um número finito de possíveis percepções e ações envolvidas (ex. Jogo de Xadrez). Caso essas percepções e ações variem sobre uma escala contínua de valores, o chamamos de ambiente contínuo (ex. Controle de tráfego aéreo).

Ainda segundo Russell e Norvig (2010), não existe um agente ideal que se adapte bem a qualquer ambiente, ou seja, cada tipo de ambiente requer agentes que se adaptem bem às suas restrições e especificidades. Dessa forma, os agentes devem ser modelados de acordo com o ambiente no qual irão habitar. Quanto mais complexo o ambiente for, mais flexível, adaptável e dinâmico deve ser o agente. De acordo com os tipos de ambientes supracitados, é fácil observar que o tipo de ambiente mais difícil de lidar é o que possui atributos que o qualificam como inacessível, não determinístico, não episódico, dinâmico e contínuo.

2.6.4 Sistemas Multiagente

Os Sistemas Multiagente (SMAs) representam uma subárea da Inteligência Artificial Distribuída que, segundo Hubner (2003), estuda o comportamento de um grupo organizado de

agentes autônomos que cooperam na resolução de problemas que estão além das capacidades de resolução de cada um individualmente.

A estratégia de tratar problemas complexos como sendo um conjunto de problemas menores tem sido bastante utilizada. Associar agentes a cada um desses problemas menores aumenta as chances de se obter uma solução mais rápida e confiável (ARTERO, 2009).

O projeto de sistemas dessa natureza depende fortemente do padrão arquitetural escolhido. Um dos padrões arquiteturais mais simples é o quadro negro. Nele, os agentes realizam suas tarefas cooperativamente, sem estabelecerem uma comunicação direta entre si. Para isso, eles utilizam, para leitura e escrita, uma estrutura de dados central, sendo ela o único vínculo existente entre eles. Outra abordagem bastante utilizada é o padrão baseado em agentes comunicativos. Nele, os agentes estabelecem um protocolo de comunicação para que possam trocar mensagens durante seus ciclos de execução (ARTERO, 2009).

Os sistemas multiagente podem estar organizados de forma: (i) hierárquica, na qual se estabelecem níveis de hierarquia no controle e tomada de decisão; (ii) comunidade de especialistas, em que todos os agentes estão no mesmo nível de decisão, porém cada um é especialista em um dado domínio; (iii) comunidade científica, na qual as soluções para os problemas são definidas localmente e validadas socialmente com outros agentes solucionadores de problemas; e (iv) federado, em que um grupo de agentes é supervisionado por um agente especial encarregado de coordenar as atividades do grupo e realizar a comunicação com os outros supervisores, formando assim grupos semiautônomos (ARTERO, 2009).

Para desenvolver um sistema multiagente é importante seguir alguma metodologia de Engenharia de Software Orientada a Agentes (ESOA). Nos últimos anos, foram propostas várias metodologias para modelagem de SMAs. Em Morais II (2010), é feito um levantamento das principais metodologias, comparando-as e apresentando suas qualidades e deficiências. Nesse mesmo trabalho, é proposta uma metodologia que estende a MAS-CommonKADS, a qual foi denominada MAS-CommonKADS+ e foi utilizada para modelagem do SMA apresentado nesta dissertação (MORAIS II, 2010).

3 MOBILE+: UM SISTEMA DE SUPORTE À APRENDIZAGEM UBÍQUA

Este capítulo apresenta o MobiLE+ (do inglês, *Mobile Learning Environment*), um sistema de suporte à aprendizagem ubíqua. O “+” (*plus*) foi adicionado pelo fato deste trabalho ter sido baseado na ideia apresentada em Silva (2012), embora nenhum código tenha sido reutilizado, cujo objetivo era fornecer objetos de aprendizagem adequados ao contexto do estudante, se limitando a aprendizagem formal. Porém, Silva (2012) desconsidera a aprendizagem informal, por não considerar os aspectos cotidiano dos usuários, e diversas características relevantes que são abordadas neste trabalho. Desta forma, o MobiLE foi totalmente reformulado para atender as tendências tecnológicas e aprimorar a personalização do ambiente.

3.1 DESCRIÇÃO GERAL DO SISTEMA

O MobiLE+ tem como objetivo central ser capaz de fornecer conteúdo adequado ao usuário, de modo a auxiliá-lo a alcançar determinado objetivo de aprendizagem, levando em consideração suas características particulares e o ambiente que está inserido. Ele foi projetado de maneira modular para se integrar a outros componentes, desenvolvidos por outros membros do grupo de pesquisa no qual o MobiLE+ está inserido.

O trabalho do grupo de pesquisa visa desenvolver um ambiente de aprendizagem ubíqua que possibilita o acesso, a visualização e o fornecimento de conteúdo educacional independentemente do horário e a partir de qualquer localidade e que direcione os estudantes de modo a maximizar seu aprendizado, mitigando a ausência do professor nos ambientes virtuais de aprendizagem. Além disso, este ambiente pode beneficiar-se da mobilidade do estudante para fornecer oportunidades educacionais presentes no ambiente real, assim como, obter informações de experiências reais, através do meio digital, providas pelas redes sociais, a fim de favorecer a autoaprendizagem e interações entre os estudantes. Desta forma, o estudante terá acesso às vantagens didático-pedagógicas oferecidas por um ambiente de aprendizagem formal, considerando, também, a importância da aprendizagem informal, uma vez que esta faz parte da vida cotidiana do estudante.

O ambiente MobiLE+ foi desenvolvido sobre a arquitetura YouUbi³ (MONTEIRO; JÁCOME JÚNIOR, 2014), composta por 9 (nove) componentes principais. Cada componente possui uma especialidade e colabora para construção do ambiente de aprendizagem ubíqua. Os componentes deste ambiente podem ser observados no Diagrama de Componentes apresentado na Figura 4.

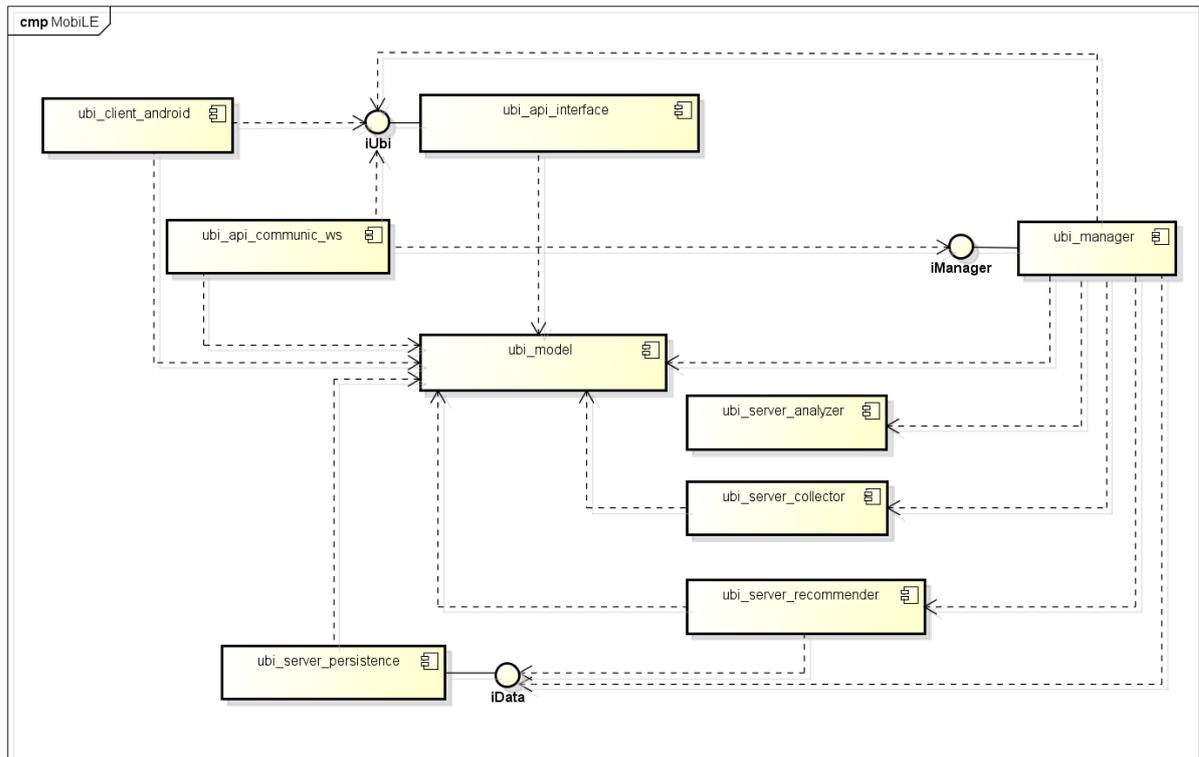


Figura 4 - Diagrama de componentes da arquitetura YouUbi.

Uma breve descrição da especialização e colaboração de cada componente é apresentada a seguir:

- *ubi_client_android*: Aplicativo desenvolvido para a plataforma Android, responsável pela interação do usuário com os conteúdos educacionais e mídias sociais (GOOGLE ANDROID DEVELOPER, 2014). Trata-se da interface da arquitetura do MobiLE+ que realiza comunicação com os demais componentes da arquitetura, através do Serviço Web (SOMMERVILLE, 2011). Além de servir como interface de interação, este componente é responsável por capturar e monitorar algumas informações, tais como: a localização geográfica, as características e estado do dispositivo e dados gerados pela interação do usuário com o dispositivo móvel. Toda transmissão do cliente ao servidor, seja

³ YouUbi: <http://www.youubi.com>.

informação capturada através do monitoramento ou por alguma ação realizada pelo usuário, segue um protocolo concebido para a arquitetura do MobiLE+.

- *ubi_api_interface*: Representa uma interface compartilhada entre o cliente e o servidor. Desta forma, esta descreve os métodos que o servidor disponibiliza para o cliente acessar via serviço web.
- *ubi_api_communic_ws*: Implementa o serviço web responsável por receber e tratar as requisições realizadas pelo cliente. Cada requisição é direcionada para ser atendida por uma classe especialista do componente *ubi_manager*, descrito posteriormente. Assim como o cliente, o serviço web segue as especificações do protocolo de comunicação criado para o MobiLE+. Uma descrição detalhada do protocolo de comunicação utilizado neste projeto será apresentada posteriormente. Este componente implementa, também, o serviço *Google Cloud Messaging – GCM* (GOOGLE ANDROID GCM, 2014). Com o serviço GCM é possível enviar dados do servidor para o aplicativo cliente mesmo quando este não está em execução e, conseqüentemente, sem a necessidade de uma requisição inicial. Logo, um agente de software pode se beneficiar dessa característica para enviar uma mensagem de notificação diretamente para o cliente.
- *ubi_model*: Este componente compreende as classes que constituem as regras de negócio do sistema, assim como seu modelo de dados. A lógica do sistema é representada por essas classes e suas relações. Também implementa o modelo de dados do sistema, através uso de anotações da API (*Application Programming Interface*) do *Java Persistence API - JPA*, refletindo diretamente na estrutura das entidades do banco de dados (DEITEL, 2005; ORACLE JPA, 2014).
- *ubi_manager*: Gerencia o fluxo de trabalho para o conjunto de classes do *ubi_model*. Desta forma, controla a instanciação e a colaboração (comunicação) entre as classes, levando em consideração as relações já definidas no modelo de dados.
- *ubi_server_collector*: Realiza o monitoramento e a coleta de informações geradas pelos usuários através da sua interação com mídias sociais. Leva em consideração que essa interação não será realizada, exclusivamente, por meio de um dispositivo móvel, mas poderá ocorrer mediante outros recursos

computacionais. Portanto, este componente monitora determinadas atividades, buscando traçar um histórico de ações ocorridas fora dos limites do ambiente MobiLE+. Vale ressaltar que essas atividades são de suma importância para acompanhar a evolução do perfil do usuário, apesar de não conterem informações contextuais deste. Este módulo atua também no refinamento de informações coletadas, transformando coordenadas geográficas em informações legíveis a humanos, identificando pontos de interesses próximos à localização atual do usuário, buscando metadados de *sites* acessados, entre outras funções.

- *ubi_server_analyzer*: Fornece uma interface genérica de enriquecimento semântico de perfil de usuário e recursos com base em interações de redes sociais, armazenando as informações coletadas em ontologias (DESPOTAKIS et al., 2012). Este componente possui interfaces tanto para sistemas de recomendação de conteúdo quanto para sistemas de captura de dados de usuários e recursos. Ele recebe as informações capturadas pela interface móvel e os recursos (metadados de vídeos, comentários, endereços de páginas web etc.) utilizados pelo usuário e aplica o processo de enriquecimento semântico (DEITEL, 2005). Os perfis e recursos enriquecidos semanticamente são armazenados em ontologias formando uma base de dados enriquecida, que será acessada pelo componente *ubi_server_recommender*, explicado a seguir, para realizar a recomendação de conteúdo.
- *ubi_server_recommender*: Consiste de um Sistema de Recomendação Híbrida Ubíqua (SRHU), ou seja, que aplica técnicas de Recomendação Colaborativa e Recomendação Baseada em Conteúdo, levando em consideração os perfis e conteúdos enriquecidos semanticamente e o contexto do usuário no momento da recomendação.
- *ubi_server_persistence*: É formado pelas classes que realizam o acesso ao banco de dados, fornecendo uma interface única de acesso. Tem como objetivo separar as regras de acesso ao banco de dados das regras de negócio da aplicação, contribuindo para diminuir o acoplamento do sistema como um todo.

Podemos identificar três subsistemas que compõem o MobiLE+, a saber:

- a) Interface Móvel e Serviço Web: Tem como finalidade: i) fornecer uma interface de interação do usuário com os conteúdos educacionais e mídias sociais; ii) monitorar e capturar as informações dos usuários; iii) prover toda a arquitetura necessária para o funcionamento do MobiLE+; e iv) criar uma arquitetura multiagente para monitorar os usuários, enriquecer os perfis e conteúdos gerados pelos usuários (utilizando o SES) e realizar a recomendação no momento apropriado (utilizando o SRHU);
- b) Sistema de Enriquecimento Semântico (SES): A solução desenvolvida consiste em um sistema de enriquecimento semântico de perfil de usuário com base nos acessos do mesmo a conteúdos relacionados com domínios de assuntos registrados no sistema através de ontologias, baseando-se em técnicas de processamento de linguagem natural, ontologias e algoritmos de seleção e marcação de conteúdo. Através do processo de enriquecimento, sabemos: i) qual a relação entre um determinado recurso e um determinado assunto; e ii) qual a relação de interesse de um usuário para com um domínio específico; e
- c) Sistema de Recomendação Híbrida Ubíqua (SRHU): Este componente tem como funções principais: i) Realizar análise periódica na base de dados e no Sistema de Enriquecimento Semântico (SES) gerando pré-listas de conteúdos potenciais à recomendação, sendo estas a Lista Colaborativa Ponderada (LCP) e a Lista Baseada em Conteúdo (LBC); e ii) Com as pré-listas preparadas, ao ser solicitado pelo componente do cliente móvel, gerar a Lista Híbrida Ubíqua (LHU), considerando os dados do contexto, e por fim a Recomendação Conjunta ou Final, que será exposta para o usuário através da Interface Móvel.

Como podemos observar, devido à complexidade do problema, modelou-se um sistema formado por três subsistemas, sendo: (i) um componente capaz de capturar as interações dos usuários com os dispositivos móveis, assim como as características desses dispositivos e do ambiente onde estão sendo utilizados, fornecendo toda a arquitetura necessária para integração e funcionamento do MobiLE+; (ii) um componente que, a partir dos dados coletados no item anterior, enriquece semanticamente os perfis dos usuários e do conteúdo acessado; e (iii) um componente que realiza a recomendação de conteúdos para usuários. No contexto do problema citado, o objetivo desse trabalho se refere apenas ao item (i), Interface Móvel e Serviço Web, sendo esse o escopo dessa dissertação.

3.2 DESCRIÇÃO DO CLIENTE ANDROID

A interface do MobiLE+ permite que os usuário acessem vídeos e outros conteúdos educacionais utilizando-se da praticidade dos dispositivos móveis. Além da facilidade de acesso, os dispositivos móveis podem fornecer diversas informações a respeito dos seus usuários. Esta seção descreverá o funcionamento da aplicação cliente e todas as informações que são capturadas com o objetivo de prover conteúdo de forma personalizada.

O Diagrama de Componentes da Figura 5 apresenta de forma detalhada a composição do componente *ubi_client_android* e sua relação com os demais componentes do ambiente. Antes de descrever as informações capturadas pelo cliente é importante compreender o funcionamento da sua arquitetura.

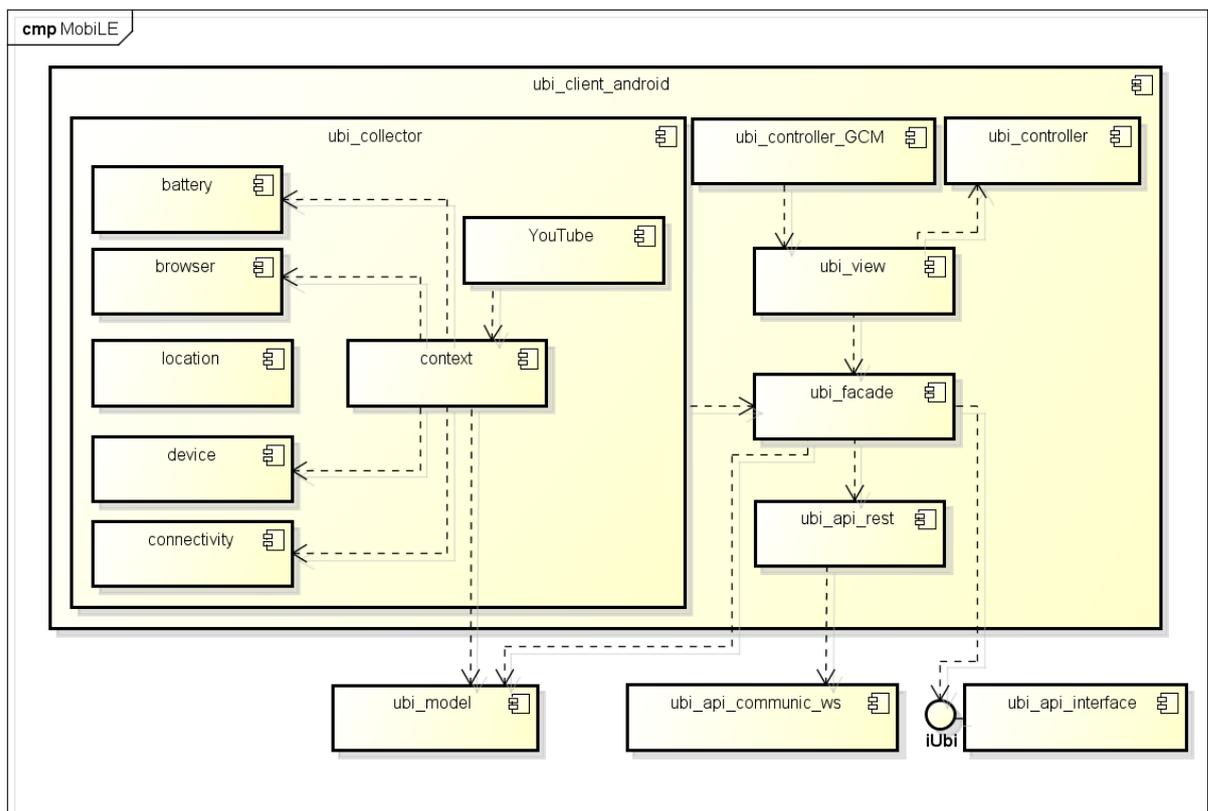


Figura 5 - Diagrama de Componentes do cliente.

O componente *ubi_view* implementa a interface gráfica com o usuário e se relaciona com *ubi_facade* para executar as funções de usuário (casos de uso) (Figura 6). Logo, o componente *ubi_facade* limita o conhecimento do cliente com relação às partes do sistema referente à lógica de negócio, contribuindo para manter o sistema desacoplado. Incumbe ao *ubi_facade* realizar a comunicação com o componente *ubi_api_rest* para enviar requisições ao

serviço web seguindo o protocolo designado. Desta forma, toda a comunicação do cliente com o serviço web é centralizada e padronizada pelo componente *ubi_api_rest*.

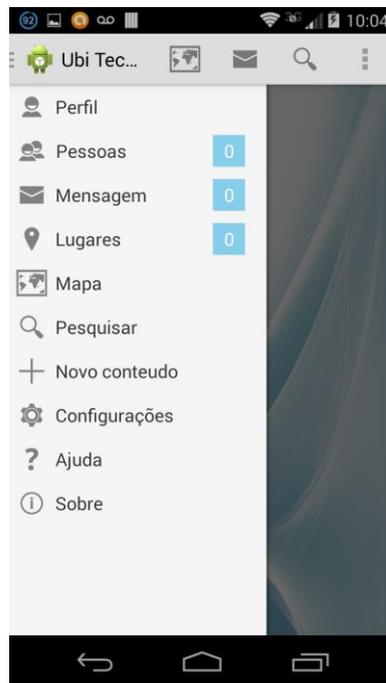


Figura 6 - Menu principal da aplicação.

O controle de sessão é realizado pelo componente *ubi_controller*, onde para cada elemento da interface será verificado se o usuário possui uma sessão válida (autenticada e autorizada). Caso não possua uma sessão válida, ele intervirá no *ubi_view* impedindo acesso e direcionando para a tela de *login* (Figura 7). Este componente é responsável também por solicitar autorização para utilização de serviços externos, por meio do protocolo *OAuth* (HAMMER-LAHAV, 2014).



Figura 7 - Tela de Login.

Já o componente *ubi_controller_GCM* implementa o cliente GCM (*Google Cloud Messaging*), que é responsável por receber mensagens enviadas pelo servidor e notificar o usuário. O serviço GCM lida com todos os aspectos de enfileiramento e entrega de mensagens, mesmo que o aplicativo não esteja em execução. Um exemplo de notificação pode ser visto na Figura 8.

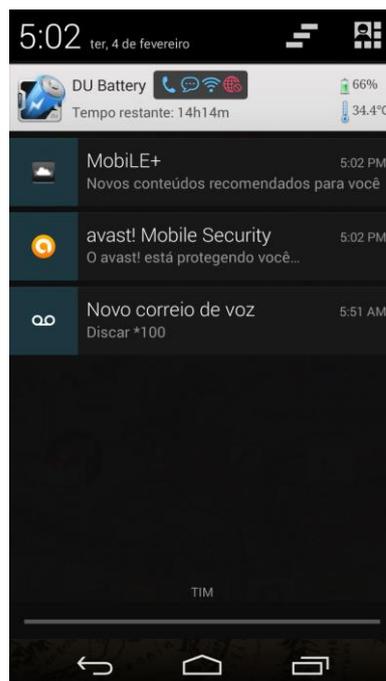


Figura 8 - Notificação de Conteúdo Recomendado.

3.2.1 Informações Capturadas do Usuário

Ao se cadastrar no sistema, o usuário preenche algumas informações básicas (nome, e-mail, data de nascimento e sexo), mas em um outro momento ele poderá optar por acrescentar outros dados ao seu perfil. Essas informações formarão o contexto estático do usuário.

O componente *ubi_collector*, apresentado na Figura 5, tem a função de coletar informações do usuário que serão úteis para compor o contexto dinâmico e melhor direcionar a recomendação dos conteúdos. Ele é atualmente formado por sete componentes, cada um responsável por coletar alguma informação específica. Essas informações irão compor o contexto do usuário em um determinado momento. Portanto, para cada ação do usuário no sistema, ou mudança de estado do dispositivo móvel, serão capturadas informações contextuais deste.

Algumas ações do sistema que ocasionam a captura do contexto do usuário são:

- Acessar ou incluir aos favoritos determinado *link*, utilizando o navegador padrão;
- Pesquisar e/ou assistir um vídeo do YouTube;
- Se cadastrar no sistema;
- Realizar *login* ou *logout*;
- Editar perfil;
- Visualizar pessoa;
- Enviar solicitação de amizade;
- Aceitar amizade;
- Enviar mensagem para outro usuário;
- Pesquisar por conteúdo;
- Visualizar conteúdo;
- Comentar conteúdo; e
- Comentar localização.

As mudanças de estado, que ocasionam a coleta de informações contextuais, são desencadeadas pelos seguintes eventos:

- O usuário aproximou-se fisicamente de um ou mais lugares (museu, teatro, cinema, universidade etc.), podendo ser de potencial interesse ou não;
- Foi identificada alteração na atividade física do usuário, entre os estados: andando, dirigindo e parado;
- O nível da bateria do dispositivo móvel foi alterado, entre os estados: crítico, baixo (menor que 30%) e normal;
- O dispositivo foi conectado ou desconectado à rede elétrica;
- Ocorreu alguma alteração no status de conectividade do dispositivo, podendo ser: conectado, desconectado, conectando, conectado em rede *wifi* e conectado em rede de dados móvel;
- O usuário instalou o cliente do MobiLE+ em outro celular; e
- O sistema multiagente possui conteúdo pendente para ser recomendado.

Para melhor compreender o funcionamento da coleta de informações que ocorre na aplicação cliente, será detalhada a função de cada elemento do componente *ubi_collector*.

3.2.1.1 Bateria

O nível da bateria é um fator a ser considerado no momento da recomendação. Não é interessante recomendar algum conteúdo (principalmente vídeo) para o usuário caso seu dispositivo esteja com um nível crítico de quantidade de energia. Um exemplo seria enviar um vídeo de 40 minutos, quando o usuário só teria cerca de 15 minutos restante de bateria. Por isso, é importante deixar todo o ambiente MobiLE+ consciente do estado da bateria.

O momento antes de enviar uma notificação para o usuário informando que existe conteúdo recomendado é a ocasião apropriada para verificar o estado da bateria. Neste caso, a verificação será solicitada pelo Sistema Multiagente (SMA) somente quando a última leitura tiver ocorrido a mais de 30 minutos após o instante da requisição. O SMA não inicia o processo de recomendação quando o nível da bateria está crítico (abaixo de 15%). Quando o nível da bateria está baixo (abaixo de 30% e acima de 15%), o Sistema de Recomendação (SR) prioriza conteúdos textuais.

Entretanto, essa verificação ocorre também em outras ocasiões que possam afetar a conduta do usuário, como apresentado a seguir:

- O estado da bateria do dispositivo foi alterado de normal para baixo ou crítico;
- O estado da bateria do dispositivo foi alterado de baixo ou crítico para normal;
- O dispositivo foi conectado à rede elétrica; e
- O dispositivo foi desconectado à rede elétrica.

Para identificar essas alterações no estado da bateria, o componente *battery* (Figura 5) implementa um *BroadcastReceiver*, que é capaz de receber mensagens padrões nativas da plataforma Android (GOOGLE ANDROID DEVELOPER, 2014). Para isso, foram especificados os filtros como apresentado na Figura 9.

```

124 <receiver android:name=".BatteryReceiver" >
125 <intent-filter>
126 <action android:name="android.intent.action.ACTION_BATTERY_LOW" />
127 <action android:name="android.intent.action.ACTION_BATTERY_OKAY" />
128 </intent-filter>
129 </receiver>

```

Figura 9 - Declaração do *BroadcastReceiver* que monitora mudança no estado da bateria.

O *BroadcastReceiver* intercepta o momento que ocorreu alguma alteração, informando se está carregando e se o nível da bateria está baixo ou normal. A identificação da natureza da mudança e o percentual da bateria fica a cargo da classe *BatteryStatus*, parte do seu código é apresentado na Figura 10.

```

21 public boolean isCharging(){
22     int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
23     return status == BatteryManager.BATTERY_STATUS_CHARGING || status == BatteryManager.BATTERY_STATUS_FULL;
24 }
25
26 public boolean isUsbCharge(){
27     int chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
28     return chargePlug == BatteryManager.BATTERY_PLUGGED_USB;
29 }
30
31 public boolean isAcCharge(){
32     int chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
33     return chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
34 }
35
36 public float getBatteryLevel(){
37     int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
38     int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);
39     return (level / (float)scale);
40 }

```

Figura 10 - Trecho de código da classe *BatteryStatus*.

Na Figura 10, o método *isCharging*, linha 21, retorna um valor lógico (verdadeiro ou falso) informando se a bateria do dispositivo está sendo carregada ou não. Já os métodos *isUsbCharge* (linha 26) e *isAcCharge* (linha 31) anunciam se o dispositivo está sendo

carregado via USB ou rede elétrica convencional, respectivamente. O método *getBatteryLevel* é responsável por informar o percentual do nível da bateria do dispositivo móvel.

3.2.1.2 Navegador (*Browser*)

Os *sites* que os usuários acessam têm um grande potencial para o processo de recomendação, pois nos diz muito a respeito das preferências destes, o que possibilita aprimorar a customização do ambiente de aprendizagem.

O componente *browser* (Figura 5) é o responsável por capturar os últimos acessos dos usuários, bem como sua lista de favoritos. O acesso de leitura ao histórico de navegação e à lista de favoritos requer permissão especial, que pode ser concedida como apresentado na Figura 11.

```
11 <uses-permission android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS" />
```

Figura 11 - Permissão para Leitura do Histórico de Navegação e Favoritos.

A Figura 11 apresenta a permissão que deve ser incluída no arquivo de manifesto para conceder o acesso de leitura do histórico e dos favoritos à aplicação cliente.

```
20 private static String[] proj = {
21     Browser.BookmarkColumns._ID,
22     Browser.BookmarkColumns.URL,
23     Browser.BookmarkColumns.TITLE,
24     Browser.BookmarkColumns.VISITS,
25     Browser.BookmarkColumns.DATE,
26     Browser.BookmarkColumns.CREATED,
27     Browser.BookmarkColumns.BOOKMARK,
28     Browser.BookmarkColumns.FAVICON
29 };
30
31 public Cursor getAllVisitedUrls(ContentResolver cr) throws IllegalStateException {
32     return cr.query(Browser.BOOKMARKS_URI, proj, "bookmark = 0", null, null);
33 }
34
35 public Cursor getAllBookmarks(ContentResolver cr) throws IllegalStateException {
36     return cr.query(Browser.BOOKMARKS_URI, proj, "bookmark = 1", null, null);
37 }
```

Figura 12 - Trecho de código da classe *VisitHistory*.

A Figura 12 exibe um trecho de código da classe *VisitHistory*, na qual podemos identificar dois métodos, *getAllVisitedUrls* e *getAllBookmarks*, responsáveis por capturar

todos os acessos do usuário e todos *sites* favoritos, respectivamente. Para cada item da lista são obtidas as seguintes informações:

- ID: identificador único do item da lista;
- URL: o endereço atribuído ao favorito ou ao item do histórico;
- TITLE: o título atribuído ao favorito ou ao item do histórico;
- VISITS: o número de vezes que o item foi visitado;
- DATE: a última data que o item foi visitado (histórico);
- CREATED: a data em que o item foi criado (favorito);
- BOOKMARK: *flag* para identificar se o item é um favorito (valor 1) ou histórico (valor 0); e
- FAVICON: o *favicon*⁴ do marcador.

Os endereços capturados na aplicação cliente são transmitidos para o servidor, onde o componente *ubi_server_collector* realizará um processo de busca para identificar seus metadados. Os metadados fornecem informações essenciais para o processo de enriquecimento semântico aplicado pelo SES.

3.2.1.3 Localização

Quando falamos em ambiente de aprendizagem ubíqua, onde este se favorece das vantagens da aprendizagem móvel, podemos destacar a importância da localização do usuário. Ambientes virtuais de ensino e aprendizagem podem ser potencializados ao proporcionar experiências mais contextualizadas. Quando o usuário está se locomovendo e explorando o ambiente físico, oportunidades educacionais surgem e essas podem guiá-lo a conseguir obter sucesso na realização de uma tarefa específica.

Reconhecer a atividade física atual do usuário, tais como andar, dirigir ou ficar parado, permite identificar o melhor momento para a recomendação. Não seria prudente, por exemplo, acessar um conteúdo ao dirigir um veículo. Então, não deve-se recomendar conteúdo durante esta atividade. Seguindo este princípio, o componente *location* (Figura 5), além de obter as coordenadas geográficas da posição do usuário, busca identificar a atividade física atual deste.

⁴ Pequenas imagens que são mantidas em *sites* para a exibição no navegador.

Estas são informações que compõem, também, o contexto do usuário e são úteis durante o processo de recomendação.

A captura das coordenadas do usuário será realizada constantemente, considerando um intervalo fixo de atualização, com o objetivo de identificar alterações na atividade física e aproximação de lugares de interesse. O reconhecimento da atividade física nos fornece um parâmetro útil para determinar se o momento será apropriado para a recomendação ou não. Já a identificação de um lugar (estabelecimentos, localizações geográficas ou pontos de interesse conhecidos) abre mais possibilidades, como exposto a seguir:

- i) A recomendação de conteúdo com base nas características do lugar atual (proximidades) e que seja tema de interesse do usuário; e
- ii) Ao chegar em um determinado local, o usuário pode receber recomendação de conteúdos que foram indicados e/ou acessados por pessoas que frequentaram o mesmo lugar e que possuem perfis similares.

As coordenadas capturadas pelo cliente serão utilizadas pelo componente *ubi_server_collector* para realizar a Geocodificação Reversa⁵ (utilizando o serviço do Google Geocoding API) e para identificar lugares, através da Google Place API. Desta forma, a localização contextual do usuário será mantida no banco de dados e possíveis lugares de interesse serão identificados (GOOGLE GEOCODING API, 2014; GOOGLE PLACE API, 2014).

A precisão das coordenadas depende das tecnologias disponíveis, podendo ser sensor GPS (*Global Position System* – Sistema de Posicionamento Global), torres de celular e *wifi*. Então é possível obter a localização mesmo quando não se possui um dispositivo com GPS integrado. No entanto, para utilizar essas tecnologias, é necessário fornecer algumas permissões especiais, como apresentado na Figura 13.

```
20 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
21 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Figura 13 - Permissões de Acesso à Localização.

A permissão exibida na linha 20, da Figura 13, concede acesso à localização aproximada derivada de fontes de localização de rede, tais como torres de celular e *wifi*. Já a

⁵ Processo onde uma localização no mapa é traduzida em um endereço legível para seres humanos (GOOGLE GEOCODING API, 2014).

permissão da linha 21 fornece ao aplicativo o acesso à localização mais precisa proveniente de fontes como GPS, torres de celular e *wifi*.

```

24 private Location getLocation(){
25     Location location;
26     if(!locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
27         && !locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)){
28         return null;
29     }else if(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)){
30         location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
31     }else if(locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)){
32         location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
33     }else{
34         location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
35     }
36     locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
37         LOCATION_REFRESH_TIME, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
38     return location;
39 }
40
41 public double getLatitude(){
42     if(location==null){
43         location = this.getLocation();
44     }
45     return location.getLatitude();
46 }
47
48 public double getLongitude(){
49     if(location==null){
50         location = this.getLocation();
51     }
52     return location.getLongitude();
53 }
54
55 public double getAltitude(){
56     if(location==null){
57         location = this.getLocation();
58     }
59     return location.getAltitude();
60 }

```

Figura 14 - Trecho de código da classe *CurrentLocation*.

O trecho de código da Figura 14 representa a classe *CurrentLocation*, encarregada de capturar as coordenadas geográficas, quando ocorre atualizações na localização. Essa classe implementa a interface *LocationListener* para monitorar atualização das informações de localização. Os parâmetros considerados na atualização da localização são determinados por duas constantes que definem o tempo e a distância mínima relevante. O método *getLocation*, da linha 24, utiliza o provedor de localização disponível no dispositivo móvel para obter a última localização conhecida. Os métodos *getLatitude* (linha 41), *getLongitude* (linha 48) e *getAltitude* (linha 55) capturam a latitude, longitude e altitude, respectivamente, da última localização válida.

3.2.1.4 Dispositivo

As características e recursos do dispositivo móvel podem limitar o acesso a recursos educacionais. Pensando nisso, o componente *ubi_device* (Figura 5) identifica várias características do dispositivo e informações sobre o serviço de telefonia disponível. Todas essas informações são enviadas para o componente *ubi_server_collector* que, através do modelo do dispositivo, verifica em uma base de dados, chamada WURFL (SCIENTIAMOBILE, 2014), quais recursos aquele dispositivo móvel suporta.

A Figura 15 mostra a permissão utilizada pelo componente *ubi_device* para obter informações do dispositivo.

```
18 | <uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Figura 15 - Permissão para leitura de informações do telefone.

Na Figura 16 é possível ver um trecho do código da classe *DeviceInfo*, responsável por capturar as diversas características do dispositivo móvel.

```

23 public String getProducId(){
24     return Build.ID;
25 }
26 public String getProducSerial(){
27     return Build.SERIAL;
28 }
29 public String getModel(){
30     return Build.MODEL;
31 }
32 public String getBrand(){
33     return Build.BRAND;
34 }
35 public String getProductFormalName(){
36     return Build.PRODUCT;
37 }
38 public String getNetworkOperatorName(){
39     return telephonyManager.getNetworkOperatorName();
40 }
41 public String getSimOperatorName(){
42     return telephonyManager.getSimOperatorName();
43 }
44 public String getDeviceId(){
45     return telephonyManager.getDeviceId();
46 }
47 public int getHeightPixels(){
48     DisplayMetrics display = new DisplayMetrics();
49     windowManager.getDefaultDisplay().getMetrics(display);
50     return display.heightPixels;
51 }
52 public int getWidthPixels(){
53     DisplayMetrics display = new DisplayMetrics();
54     windowManager.getDefaultDisplay().getMetrics(display);
55     return display.widthPixels;
56 }

```

Figura 16 - Trecho de código da classe *DeviceInfo*.

3.2.1.5 Conectividade

A conectividade é um fator primordial para um ambiente ubíquo. Não é possível conceber tal ambiente sem a presença de uma conexão e interoperabilidade entre dispositivos. Quando falamos em aprendizagem informal, é fundamental considerar a diversidade de conteúdos disponíveis na web. Logo, para o ambiente de aprendizagem ubíqua é fundamental a existência de conexão com a internet, uma vez que sem ela não é possível acessar e receber recomendação de conteúdos educacionais, interagir com outras pessoas, acessar redes sociais, dentre outras atividades.

O componente *connectivity* (Figura 5) tem a função de fornecer informações sobre o *status* da conexão do dispositivo. O cliente deve saber dessa informação para determinar se

um processo de recomendação é realmente viável, a depender do tipo (*wifi* e Dados da Telefonia Móvel) e qualidade da internet, além da disposição do usuário para assumir os custos. Já o servidor, através do componente *ubi_server_collector*, pode priorizar conteúdos educacionais menores quando a conexão com a internet for limitada.

As permissões de acesso às informações de conectividade são apresentadas na Figura 17.

```

13     <uses-permission android:name="android.permission.NETWORK" />
14     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
15     <uses-permission android:name="android.permission.INTERNET" />

```

Figura 17 - Permissões relacionadas à conectividade.

A Figura 17 exhibe as permissões referente à conectividade. A declaração da linha 13 fornece acesso a serviços de rede. A permissão da linha 14 possibilita o acesso a informações sobre redes. Já a linha 15 permite que o aplicativo estabeleça uma conexão com a internet.

```

124<⊖   <receiver android:name=".ConnectivityReceiver" >
125<⊖       <intent-filter>
126<⊖           <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
127<⊖       </intent-filter>
128<⊖   </receiver>

```

Figura 18 - Declaração do *ConnectivityReceiver* que monitora alterações no *status* de conexão.

O *ConnectivityReceiver* detecta alterações no *status* da conectividade, informando se ocorreu mudanças entre dados móveis e *wifi* gratuita. A classe *ConnectivityStatus* (Figura 19) fornece mais detalhes sobre a natureza da alteração.

```

19<⊖   public boolean isConnected(){
20<⊖       return activeNetwork != null && activeNetwork.isConnectedOrConnecting();
21<⊖   }
22<⊖
23<⊖   public boolean isWiFi(){
24<⊖       boolean result = false;
25<⊖       if(activeNetwork != null) {
26<⊖           result = (activeNetwork.getType() == ConnectivityManager.TYPE_WIFI);
27<⊖       }
28<⊖       return result;
29<⊖   }
30<⊖
31<⊖   public boolean isMobileData(){
32<⊖       boolean result = false;
33<⊖       if(activeNetwork != null) {
34<⊖           result = (activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE);
35<⊖       }
36<⊖       return result;
37<⊖   }

```

Figura 19 - Trecho de código da classe *ConnectivityStatus*.

O método da linha 19 identifica se o dispositivo está conectado ou conectando. Os métodos *isWiFi* (linha 23) e *isMobileData* (linha 31) identificam se a conexão é do tipo *wifi* gratuita ou dados móveis, respectivamente.

3.2.1.6 YouTube

O YouTube permite acessar uma vasta gama de vídeos educacionais que vão desde palestras acadêmicas a tutoriais, discursos, aulas e muito mais. Essa ferramenta, quando utilizada de maneira adequada, possui um grande potencial educacional, pois se trata de um meio eficaz de disseminação de informação.

O MobiLE+ possui integração com o YouTube, onde é possível: buscar, acessar, avaliar, comentar e associar a localização atual ao vídeo. Essas informações são enviadas para o componente *ubi_server_collector*, que utilizará o *id* do vídeo para obter outras informações deste, como metadados, descrição e comentários. Em seguida, estas informações são enriquecidas semanticamente antes de serem armazenadas para futura recomendação.

Para a integração com o cliente foi utilizada a API de Dados do YouTube para a plataforma Android. Com ela é possível buscar, visualizar, avaliar e realizar comentários aos vídeos.

```

32 private List<SearchResult> search(String term) {
33     List<SearchResult> searchResultList = null;
34     try {
35         youtube = new YouTube.Builder(HTTP_TRANSPORT, JSON_FACTORY, new HttpRequestInitializer() {
36             public void initialize(HttpRequest request) throws IOException {}
37         }).setApplicationName("youtube-cmdline-search-sample").build();
38
39         YouTube.Search.List search = youtube.search().list("id,snippet");
40
41         search.setKey(DeveloperKey.DEVELOPER_KEY);
42         search.setQ(term);
43         search.setType("video");
44
45         search.setFields("items(id/kind,id/videoId,snippet/title,snippet/description,snippet/thumbnails/default/url)");
46         search.setMaxResults(NUMBER_OF_VIDEOS_RETURNED);
47         SearchListResponse searchResponse = search.execute();
48
49         searchResultList = searchResponse.getItems();
50     } catch (GoogleJsonResponseException e) {
51         System.err.println("There was a service error: " + e.getDetails().getCode() + " : " + e.getDetails().getMessage());
52     } catch (IOException e) {
53         System.err.println("There was an IO error: " + e.getCause() + " : " + e.getMessage());
54     } catch (Throwable t) {
55         t.printStackTrace();
56     }
57     return searchResultList;
58 }

```

Figura 20 - Trecho de código da classe *YouTubaManager*.

O trecho de código da classe *YouTubeManager* (Figura 20) demonstra o método que realiza a busca por vídeos a partir de um termo informado, instanciando classes da API fornecida pelo YouTube.

A Figura 21 apresenta a tela onde se pode visualizar o vídeo escolhido ou recomendado.



Figura 21 - Tela de Visualização de Vídeos.

Após a execução de um vídeo será apresentada uma interface, demonstrada na Figura 22, onde o usuário poderá realizar comentário, avaliar o vídeo e associar este à sua posição atual.

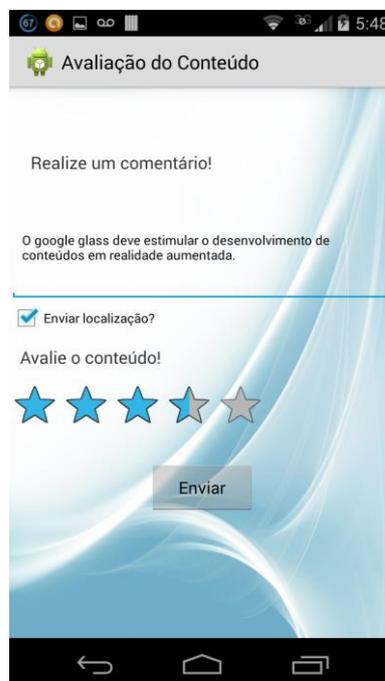


Figura 22 - Tela de Comentário e Avaliação sobre o Vídeo.

3.3 IDENTIFICAÇÃO DO ESTILO DE APRENDIZAGEM

A identificação dos estilos de aprendizagem dos alunos, no MobiLE+, é inicialmente realizada com o auxílio do método ILS (descrito na Seção 2.5). O usuário, ao se cadastrar no MobiLE+, é convidado a responder um formulário com 44 (quarenta e quatro) perguntas (Anexo A). Em seguida, suas dimensões de estilo de aprendizagem são calculadas e persistidas no banco de dados.

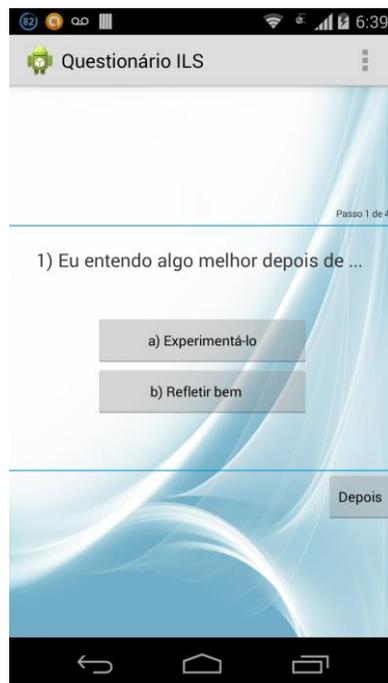


Figura 23 - Aplicação do Questionário ILS.

Na Figura 23 é possível ver o questionário ILS sendo aplicado no aplicativo Android. Após essa identificação participativa, ou seja, com a intervenção direta do aluno, do estilo de aprendizagem, é iniciado um ciclo de adaptação, no qual o estilo de aprendizagem passará por frequentes análises e possíveis atualizações, de acordo com seu histórico de preferências.

Essa etapa é conduzida pelo Agente Usuário (UAg). Inicialmente, o agente acessa o estilo de aprendizagem armazenado no banco, ou seja, aquele proveniente do método ILS. Em seguida, o agente obtém o histórico de escolhas do usuário e realiza uma análise comparativa com seu estilo de aprendizagem armazenado no banco. Com isso, será possível identificar desvios e realizar pequenos ajustes no estilo de aprendizagem.

Para realizar os ajustes, o UAg, inicialmente, verifica os metadados dos conteúdos com a finalidade de identificar a dimensão de estilo de aprendizagem que este se enquadra. Desta forma, um conteúdo que possui a palavra-chave “tutorial” tende a ser acessado por um

usuário que possui maior facilidade em aprender através de passos lineares, ou seja, de maneira gradual. Já um conteúdo em formato de vídeo ou imagem tende a ser acessado por usuários que preferem aprender por meio da percepção visual.

Em um segundo momento, o UAg busca identificar a frequência de acessos a conteúdos que não se enquadram no estilo de aprendizagem armazenado no banco (inicialmente obtida pelo questionário ILS). Portanto, será realizada uma análise do histórico de acessos dos usuários para identificar o percentual de conteúdos que divergem do estilo de aprendizagem obtido em cada dimensão. O ajuste, para cada dimensão, será realizado tomando como base a escala descrita na Figura 2, de modo que:

- Para um percentual de divergência entre 50% e 59%, o estilo de aprendizagem será deslocado em 1 (um) ponto para o lado do estilo divergente;
- Para um percentual de divergência entre 60% e 79%, o estilo de aprendizagem será deslocado em 2 (dois) pontos para o lado do estilo divergente;
- Para um percentual de divergência entre 80% e 100%, o estilo de aprendizagem será deslocado em 3 (três) pontos para o lado do estilo divergente.

3.4 ARQUITETURA ORIENTADA A SERVIÇO

Serviço Web é uma das soluções mais utilizadas para integração de sistemas e comunicação entre aplicações diferentes. Esta tecnologia permite a compatibilidade de softwares desenvolvidos em diferentes plataformas. Desta forma, os Serviços Web fornecem uma solução que define um padrão para troca de mensagens entre as aplicações consumidoras (clientes) e um fornecedor do serviço (SOMMERVILLE, 2011).

Para o desenvolvimento do Serviço Web utilizou-se a arquitetura RESTful (ORACLE REST, 2006). Trata-se de uma arquitetura de transferência de estado representacional que fornece serviços de acesso a recursos. As informações da aplicação e de transações podem ser estruturadas nos formatos XML (*eXtensible Markup Language*) (W3C, 2014), JSON (*JavaScript Object Notation*) (JSON, 2014), entre outros. Esta arquitetura possui um pequeno conjunto de operações para manipulação das informações dos recursos. As aplicações da rede se comunicam através do protocolo HTTP (*Hypertext Transfer Protocol*) (HTTP, 2014), com o objetivo de trocar representações de recursos (Figura 24). Desta forma, o REST estende os

atributos do Protocolo HTTP e *Identificador Uniforme de Recursos (Uniform Resource Identifier - URI)* para recuperar ou modificar o estado de um recurso.

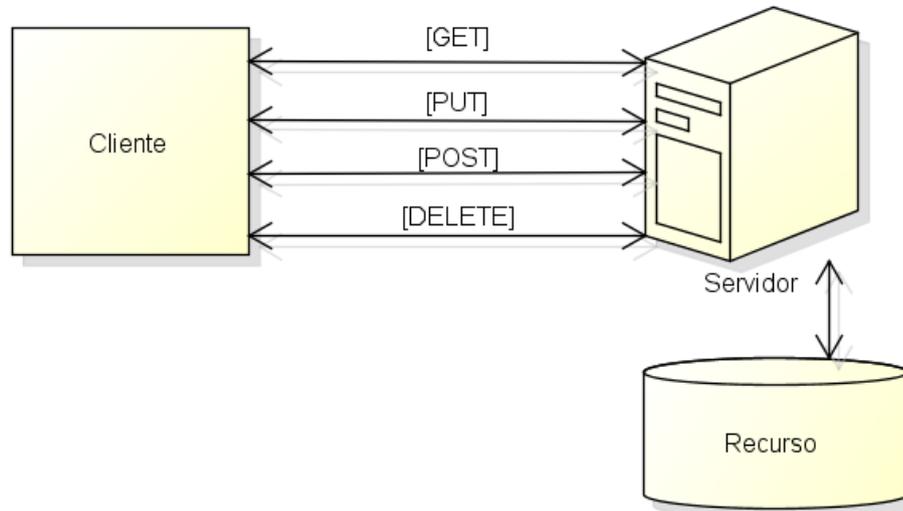


Figura 24 - Representação dos Métodos Padrões do Serviço Rest.

A Figura 24 apresenta os métodos que devem ser utilizados para manipular um recurso. Estes métodos são:

- PUT: para criar um recurso;
- GET: para obter um recurso;
- POST: para atualizar um recurso; e
- DELETE: para excluir um recurso.

O Java EE (*Enterprise Edition*) (ORACLE JAVA EE, 2014) integra o JSR-311 (*Java Specification Requests*) (ORACLE REST, 2006) para fornecer um conjunto de especificações que define a criação de um Serviço Web com a arquitetura RESTful. Para a implementação de referência do JSR-311, utilizou-se o Jersey (JERSEY, 2014), que é um projeto *open source* mantido pela *Oracle Corporation*. Com ele é possível realizar anotações em classes e métodos para definir recursos e suas operações.

O Serviço Web desenvolvido tem como objetivo fornecer um meio de troca de recursos entre o cliente Android e o servidor. Os recursos definidos para o MobiLE+ são representações de classes/entidades pertencentes ao componente *ubi_model* (Figura 4). No Capítulo 4 serão detalhados os aspectos de modelagem e implementação do serviço, bem como o protocolo de comunicação desenvolvido.

3.5 CAMADA DE PERSISTÊNCIA DE DADOS

O Mobile+ faz uso da camada de persistência e do modelo de dados da arquitetura YouUbi (MONTEIRO; JÁCOME JÚNIOR, 2014). Esta camada é implementada a nível de aplicação utilizando classes de entidades com anotações JPA e a nível de banco de dados foi utilizado o PostgreSql (POSTGRESQL, 2014).

Na Figura 25 é apresentado o modelo de banco de dados com as entidades do MobiLE+. Nele é possível visualizar todas as entidades que dão suporte ao sistema, bem como seus relacionamentos.

A Tabela 3 exibe uma classificação e descrição das principais entidades, do modelo de dados do YouUbi, utilizadas pelo SMA.

Tabela 3 – Principais Entidades do Modelo de Dados do MobiLE+

Tipo	Nome	Descrição
Básicas	<i>Person</i>	Representa os usuários cadastrados no sistema e que irão interagir e receber recomendações de conteúdos.
	<i>Content</i>	Representa todo o tipo de conteúdo do sistema que pode servir de objeto de aprendizagem para um usuário e possui um endereço web.
	<i>Location</i>	Representa uma posição no globo, longitude e latitude, e os detalhes que identificam o que está localizado nessa posição.
Relações	<i>RelationPersonPerson</i>	Representa as formas de relacionamento entre os usuários.
	<i>RelationPersonContent</i>	Representa as formas de relacionamento entre os usuários e os conteúdos.
	<i>RelationPersonLocation</i>	Representa as formas de relacionamento entre os usuários e os locais.
	<i>RelationContentLocation</i>	Representa as formas de relacionamento entre os conteúdos e os locais.
Auxiliares	<i>Device</i>	Representa os aparelhos ou dispositivos, e suas características e limitações, pelos quais o usuário pode acessar o sistema.
	<i>Context</i>	Representa um “retrato” de um instante de um usuário, com seu respectivo dispositivo, em um determinado local, interagindo de alguma forma com o sistema, acessando ou não um conteúdo.
	<i>Learning</i>	Representa uma extensão da entidade <i>Person</i> com características específicas que identificam a

		formação e a forma de aprendizagem do usuário.
	<i>Frequency</i>	Representada por quatro histogramas, sendo classificados em diários (dois) ou semanais (dois), positivos ou negativos. Os positivos representam o ato de aceitação ou visualização de conteúdo, enquanto que os negativos representam a rejeição a uma recomendação de conteúdo. Se relacionam a atributos estendidos das entidades básicas <i>person</i> , <i>content</i> e <i>location</i> .
	<i>Tags</i>	Representadas por uma lista de termos que em conjunto possuem alguma relevância e identificam os interesses ou assuntos relacionados a uma das entidades básicas <i>Person</i> , <i>Content</i> ou <i>Location</i> .

A seguir será realizado um detalhamento das entidades, apresentando uma breve descrição dos seus campos:

a) *Person*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *email*: Representa o e-mail do usuário, único e utilizado como *login*.
- *pass*: Armazena a senha do usuário para acessar o sistema.
- *nameFirst*: Identifica o primeiro nome do usuário.
- *nameLast*: Identifica o último nome do usuário.
- *dateRegister*: Representa a data de registro do usuário no sistema.
- *dateBirth*: Representa a data de nascimento do usuário.
- *genre*: Identifica o sexo do usuário.
- *phone*: Representa o número do telefone do usuário.
- *income*: Identifica a renda do usuário, expressa em reais.
- *religion*: Identifica a religião do usuário.
- *race*: Identifica a cor ou raça do usuário.
- *profession*: Identifica a profissão do usuário.
- *rateDurationContents*: Indica, em segundos, a média de duração dos

conteúdos visualizados pelo usuário.

- *learning*: Representa uma instância da entidade *Learning* e caracteriza a formação e modo de aprendizado do usuário. Seus atributos são melhor detalhados no item que explica a própria entidade *Learning*.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários do usuário. Seus atributos são melhor detalhados no item que explica a própria entidade *Frequency*.
- *listContexts*: Representa o histórico de contextos vividos pelo usuário. Seu último registro representa o momento atual. Seus atributos são melhor detalhados no item que explica a própria entidade *Context*.
- *listDevices*: Representa a lista de aparelhos do usuário. Seus atributos são melhor detalhados no item que explica a própria entidade *Device*.
- *listTags*: Representa a lista de *tags* de um usuário. Seus atributos são melhor detalhados no item que explica a própria entidade *Tag*.
- *listRelatePersonPerson*: Representa todas as relações entre os usuários. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationPersonPerson*.
- *listRelatePersonContent*: Representa todas as relações entre usuários e conteúdos. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationPersonContent*.
- *listRelatePersonLocation*: Representa todas as relações entre usuários e locais. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationPersonLocation*.

b) *Content*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica o tipo de conteúdo (postagem, evento, grupo ou desafio). Apenas as postagens estão sendo consideradas nesse primeiro momento para o sistemas de recomendação.
- *subtype*: Identifica o subtipo do conteúdo. Para o tipo Postagem, por exemplo, o subtipo pode assumir os valores de vídeo, áudio, página, imagem ou texto.
- *visibility*: Define os níveis de privacidade de um conteúdo. Pode estar visível a nível de todos, amigos ou privado. Apenas os visíveis a cada

usuário são utilizados no sistema de recomendação.

- *title*: Representa o título do conteúdo.
- *description*: Representa a descrição do conteúdo.
- *dateCreation*: Representa a data de criação de um conteúdo.
- *urlOnline*: Representa o endereço web vinculado ao conteúdo.
- *bytesOnline*: Representa o tamanho (em kbytes) do conteúdo.
- *secondsOnline*: Representa a duração (em segundos) necessária para conhecer todo o conteúdo.
- *rating*: Representa a média das avaliações dos usuários em relação ao conteúdo.
- *rateAcceptance*: Representa o percentual de aceitação dos usuários em relação ao conteúdo, quando este é recomendado.
- *person*: Representa o usuário que inseriu este conteúdo no sistema.
- *author*: Representa o autor do conteúdo.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários do conteúdo. Seus atributos são melhor detalhados no item que explica a própria entidade *Frequency*.
- *listTags*: Representa a lista de *tags* de um conteúdo. Seus atributos são melhor detalhados no item que explica a própria entidade *Tag*.
- *listRelatePersonContent*: Representa todas as relações entre os conteúdos e os usuários. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationPersonContent*.
- *listRelateContentLocation*: Representa todas as relações entre os conteúdos e os locais. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationContentLocation*.

c) *Location*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *title*: Representa o título do local ou nome que o identifica.
- *description*: Representa a descrição do local.
- *altitude*: Representa, em metros, a altitude do local em relação ao nível do mar.
- *longitude*: Representa a coordenada geográfica de longitude onde está situado o local.

- *latitude*: Representa a coordenada geográfica de latitude onde está situado o local.
- *country*: Representa a parte do endereço que identifica o país onde está localizado o local.
- *state*: Representa a parte do endereço que identifica o estado onde está localizado o local.
- *city*: Representa a parte do endereço que identifica a cidade onde está localizado o local.
- *neighborhood*: Representa a parte do endereço que identifica o bairro onde está localizado o local.
- *streetName*: Representa a parte do endereço que identifica o nome da rua onde está localizado o local.
- *streetNumber*: Representa a parte do endereço que identifica o número onde está localizado o local.
- *zipCode*: Representa a parte do endereço que identifica o código de endereçamento postal onde está localizado o local.
- *idGoogle*: Representa o identificador do local no sistema de mapeamento utilizado.
- *rating*: Representa a média das avaliações dos usuários em relação ao local.
- *rateAcceptanceHere*: Representa o percentual de aceitação dos usuários em relação aos conteúdos recomendados nesse local.
- *rateRatingHere*: Representa a média das avaliações dos usuários em relação aos conteúdos recomendados nesse local.
- *person*: Representa o usuário que inseriu este local no sistema.
- *frequency*: Representa os histogramas de aceitação e rejeição, semanais e diários de conteúdo no local. Seus atributos são melhor detalhados no item que explica a própria entidade *Frequency*.
- *listTypes*: Representa os tipos de funcionalidades relativas ao local, sendo extraídas do sistema de mapeamento.
- *listTags*: Representa a lista de *tags* de um local. Seus atributos são melhor detalhados no item que explica a própria entidade *Tag*.
- *listRelatePersonLocation*: Representa todas as relações entre usuários e locais. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationPersonLocation*.

- *listRelateContentLocation*: Representa todas as relações entre os conteúdos e os locais. Seus atributos são melhor detalhados no item que explica a própria entidade *RelationContentLocation*.

d) *RelationPersonPerson*:

- *person1*: Identifica o primeiro usuário da relação.
- *person2*: Identifica o segundo usuário da relação.
- *status*: Identifica o tipo de relação existente entre os usuários. Pode assumir os valores: amigos, solicitou amizade, amizade solicitada, rejeitou amizade, amizade rejeitada, deletou amizade, amizade deletada, visualizou perfil e perfil visualizado.
- *affinityRate*: Identifica o grau de afinidade entre *person1* e *person2*. Este índice é calculado com base em vários atributos das entidades e dos históricos de cada um dos usuários.
- *commomPersons*: Representa o número de amigos em comum entre *person1* e *person2*.
- *dateRelation*: Representa a data da criação da relação.

e) *RelationPersonContent*:

- *person*: Identifica o usuário da relação.
- *content*: Identifica o conteúdo da relação.
- *status*: Identifica qual o tipo de relação existente entre o usuário e o conteúdo. Pode assumir os valores: favoritos, sugeridos, visualizados, rejeitados ou avaliados.
- *dateRelation*: Representa a data da criação da relação.
- *countRelation*: Representa quantas vezes essa relação ocorreu com um determinado *status*.
- *ratePerson*: Avaliação de *person* sobre esse conteúdo. Caso o conteúdo ainda não tenha sido avaliado por *person*, *ratePerson* assumi o valor de 5,0.
- *i1*: No caso de conteúdo sugerido a um usuário, *i1* representa o peso da recomendação colaborativa, desse conteúdo para o usuário.
- *i2*: No caso de conteúdo sugerido a um usuário, *i2* representa o peso da recomendação baseada em conteúdo, desse conteúdo para o usuário.
- *i3*: No caso de conteúdo sugerido a um usuário, *i3* representa o peso da

recomendação baseada na ubiquidade⁶, desse conteúdo para o usuário.

f) *RelationPersonLocation*:

- *person*: Identifica o usuário da relação.
- *location*: Identifica o local da relação.
- *status*: Identifica qual o tipo de relação existente entre usuário e conteúdo nesse local. Pode assumir os valores: favoritos, sugeridos, visualizados, rejeitados ou avaliados.
- *finality*: Identifica a finalidade do uso do local pelo usuário nessa relação. Pode assumir os valores: educação, trabalho, casa/lar, lazer ou outro.
- *ratePerson*: Para o caso do status ser “avaliado”, identifica a avaliação do usuário sobre esse local, caso contrário recebe o valor padrão de 5,0.
- *dateRelation*: Representa a data da criação da relação.

g) *RelationContentLocation*:

- *content*: Identifica o conteúdo da relação.
- *location*: Identifica o local da relação.
- *dateRelation*: Representa a data da criação da relação.
- *status*: Identifica qual o tipo de relação ocorreu para o conteúdo nesse local. Pode assumir os valores: sugerido, rejeitado, criado, visualizado, relacionado, mencionado, rejeitados ou avaliado.

h) *Device*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica o tipo de dispositivo. Pode assumir os valores: *smartphone*, *tablet*, *notebook* ou *desktop*.
- *os*: Identifica qual o sistema operacional utilizado pelo dispositivo.
- *osVersion*: Identifica a versão do sistema operacional utilizado pelo dispositivo.
- *hasWifi*: Informa se o dispositivo tem o recurso de *wifi*.
- *hasGPS*: Informa se o dispositivo tem o recurso de GPS.
- *hasBluetooth*: Informa se o dispositivo tem o recurso de *Bluetooth*⁷.

⁶ Processo de recomendação que leva em consideração o contexto do usuário.

⁷ Tecnologia que utiliza ondas de rádio de curto alcance para conexão e troca de informações entre dispositivos como *notebook*, *smartphone*, computadores e videogames (BLUETOOTH, 2014).

- *displayWidth*: Informa a largura, em *pixel*, do *display* do dispositivo.
- *displayHeight*: Informa a altura, em *pixel*, do *display* do dispositivo.
- *productBrand*: Informa a marca do dispositivo.
- *productModel*: Informa o modelo do dispositivo.
- *mac*: Informa o endereço MAC (*Media Access Control*) do dispositivo.
- *description*: Informa a descrição do dispositivo.
- *person*: Identifica o usuário relacionado ao dispositivo.

i) *Context*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *type*: Identifica o tipo de evento que disparou a criação do contexto. Pode ser uma alteração de local, um recebimento, criação, ou postagem de um conteúdo, ou alguma outra ação do usuário no dispositivo.
- *timeStamp*: Identifica o momento da criação do contexto.
- *status*: Identifica se o usuário está *online* ou *off-line* no sistema.
- *battery*: Informa o percentual da carga da bateria do dispositivo no momento da criação do contexto.
- *ip*: Identifica o endereço IP (*Internet Protocol*) do dispositivo no momento da criação do contexto.
- *connectionSpeed*: Informa a velocidade da conexão, em kbps, no momento da criação do contexto.
- *personSpeed*: Informa a velocidade do usuário, em m/s, no momento da criação do contexto.
- *person*: Identifica o usuário relacionado ao contexto.
- *location*: Identifica o local da criação do contexto.

j) *Learning*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *educationalLevel*: Identifica o nível educacional do usuário. Pode ser: fundamental completo ou incompleto, médio completo ou incompleto, superior completo ou incompleto, ou pós-graduação.
- *formationCourse*: Para usuário com nível educacional superior incompleto ou acima, informa o nome do curso de graduação.
- *formationBigArea*: Para usuário com nível educacional superior incompleto

ou acima, informa a grande área do curso de graduação.

- *formationArea*: Para usuário com nível educacional superior incompleto ou acima, informa a área do curso de graduação.
- *formationSubarea*: Para usuário com nível educacional superior incompleto ou acima, informa a subárea de trabalho ou de pesquisa do usuário.
- *styleDimenPerception*: Informa o estilo de aprendizagem quanto à dimensão de percepção do usuário. Pode ser: sensorial ou intuitiva.
- *styleDimenRetention*: Informa o estilo de aprendizagem quanto à dimensão de retenção do usuário. Pode ser: visual ou verbal.
- *styleDimenProcessing*: Informa o estilo de aprendizagem quanto à dimensão de processamento do usuário. Pode ser: ativo ou reflexivo.
- *styleDimenOrganization*: Informa o estilo de aprendizagem quanto à dimensão organizacional do usuário. Pode ser: sequencial ou global.

k) *Frequency*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *h24Positive*: É composto de vinte e quatro posições que representam as horas do dia, indicando o percentual de aceitação de um conteúdo por parte de um usuário em um determinado local, para cada uma dessas horas.
- *h24Negative*: É composto de vinte e quatro posições que representam as horas do dia, indicando o percentual de rejeição de um conteúdo por parte de um usuário em um determinado local, para cada uma dessas horas.
- *d7Positive*: É composto de sete posições que representam os dias da semana, indicando o percentual de aceitação de um conteúdo por parte de um usuário em um determinado local, para cada um desses dias.
- *d7Negative*: É composto de sete posições que representam os dias da semana, indicando o percentual de rejeição de um conteúdo por parte de um usuário em um determinado local, para cada um desses dias.

l) *Tag*:

- *id*: Identificador padrão da instância e chave primária da tabela no banco de dados.
- *name*: Representa o termo que identifica a *tag*.
- *count*: Informa o número de vezes que a *tag* foi representada.

4 MODELAGEM E IMPLEMENTAÇÃO DO SERVIÇO WEB

Este capítulo busca esclarecer como foi projetado e desenvolvido o Serviço Web (SW), detalhando a implementação do servidor (provedor do serviço) e do cliente (consumidor do serviço), bem como o protocolo utilizado por estes.

4.1 PROVEDOR DO SERVIÇO

Este é responsável por disponibilizar o serviço de acesso e manipulação de recursos. Ao receber uma requisição de um cliente, são identificados o recurso e a natureza da operação. Depois o SW realiza uma chamada a uma classe específica do componente *ubi_manager* com o objetivo de atender à requisição do cliente, obtendo, atualizando, deletando ou criando um recurso.

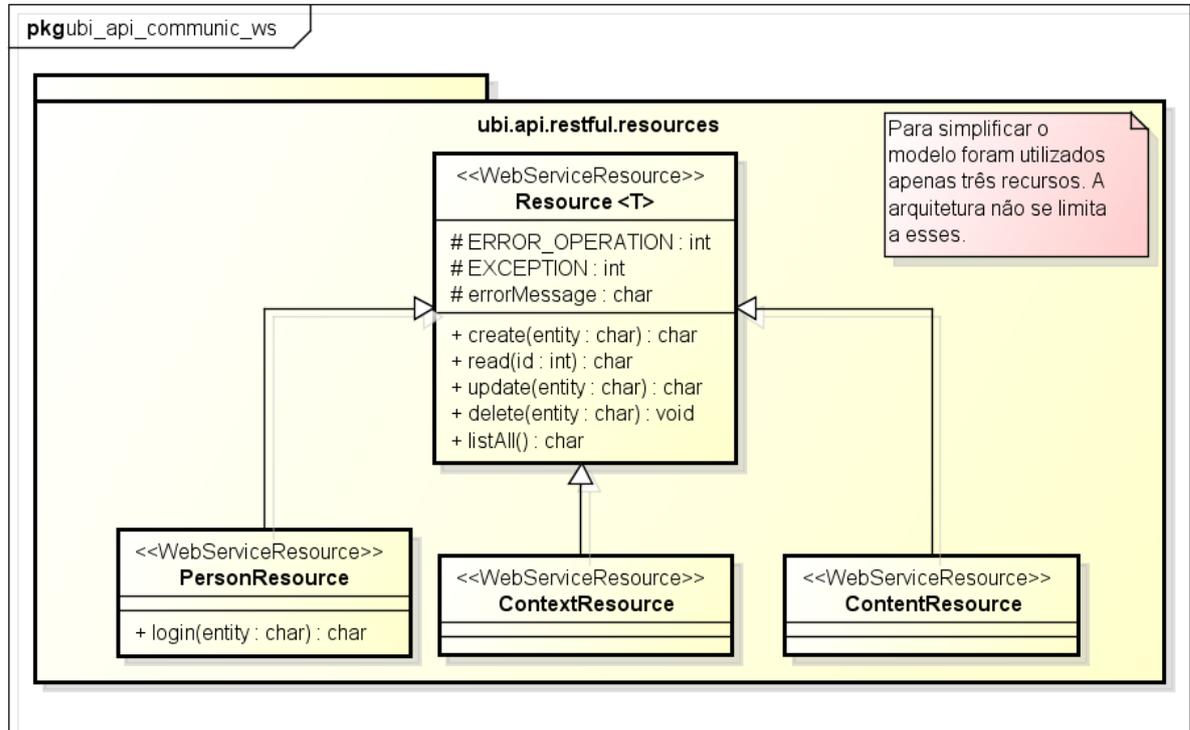


Figura 26 - Diagrama de Classes da Implementação dos Recursos.

Como pode ser visto na Figura 26, os recursos especificados no SW herdam as características de uma classe mais genérica denominada *Resource*. Ela possui uma

implementação padrão para acesso aos recursos, onde é possível criar (*create*), obter (*read*), atualizar (*update*), deletar (*delete*) e listar (*listAll*) recursos. Os recursos herdam as operações da classe *Resource*, mas podem implementar métodos específicos, como é o caso da classe *PersonResource*, que possui um método (*login*) para autenticar no sistema. Para simplificar, foram apresentados somente três recursos, pois os demais seguem o mesmo padrão.

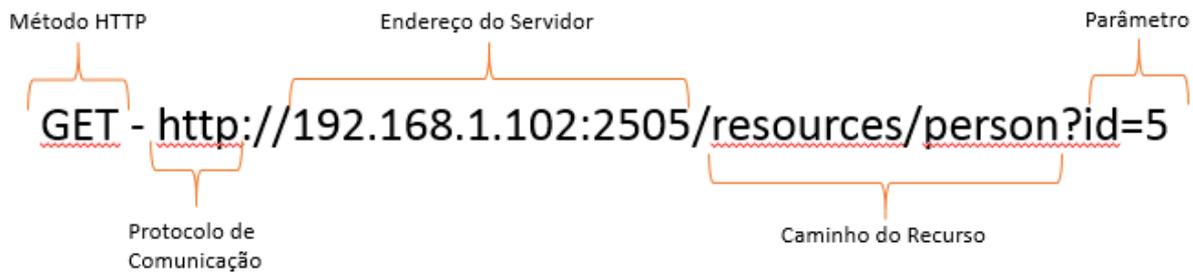


Figura 27 - Exemplo de uma Requisição para Consumo do Serviço

A Figura 27 demonstra o padrão de URL utilizado pelo SW. Nela é possível identificar o protocolo de comunicação utilizado, o método HTTP, o endereço do servidor, o caminho do recurso e os parâmetros. A Tabela 4 descreve as URIs disponíveis para o recurso *Person*.

Tabela 4 - URIs de acesso ao recurso *Person*.

URI do Recurso	Método	Formato	Efeito
/resource/person/{id}	GET	<i>Person</i>	Busca um objeto <i>Person</i> .
/resource/person/{id}	POST	<i>Person</i>	Atualiza o objeto <i>Person</i> .
/resource/person	PUT	<i>Person</i>	Cria objeto <i>Person</i> .
/resource/person	DELETE	<i>Person</i>	Remove objeto <i>Person</i> .
/resource/person/listAll	GET	<i>Person</i>	Lista todos os objetos <i>Person</i> .
/resource/person/login	POST	<i>Person</i>	Autentica o usuário.

A Figura 28 apresenta um trecho de código da classe *Resource*. Como pode ser visto em sua declaração (linha 16), trata-se de uma classe abstrata, ou seja, é um modelo genérico que não pode ser instanciado, mas pode ser compartilhado por um grupo de classes derivadas. Então essa classe fornece um padrão a ser seguido pelas demais classes de recurso, na arquitetura do MobiLE+. Nas linhas 14 e 15, são utilizadas anotações da especificação JSR-311 para indicar os tipos de dados que serão aceitos e fornecidos, respectivamente, pelo recurso. Já a anotação da linha 27 especifica o método HTTP que será utilizado para a chamada da operação (linha 28).

```

14 @Consumes(MediaType.APPLICATION_JSON+";charset=utf-8")
15 @Produces(MediaType.APPLICATION_JSON+";charset=utf-8")
16 public abstract class Resource <T> implements IResource<T>{
17     protected Class<T> clazz;
18     protected final static int ERROR_OPERATION = 1;
19     protected final static int EXCEPTION = 2;
20     protected final static String errorMessage="Service temporarily unavailable. Soon we're back!";
21
22     protected Resource(Class<T> clazz) {
23         this.clazz = clazz;
24     }
25
26     @SuppressWarnings("finally")
27     @PUT
28     public String create(String entity){
29         String result="";
30         try {
31             Context context = null;
32             Object object = new GsonBuilder().excludeFieldsWithoutExposeAnnotation().create()
33                 .fromJson(new JsonParser().parse(entity).getAsJsonObject().get("entity"), this.clazz);
34             if(new JsonParser().parse(entity).getAsJsonObject().has("context")){
35                 context = new GsonBuilder().excludeFieldsWithoutExposeAnnotation().create()
36                     .fromJson(new JsonParser().parse(entity).getAsJsonObject().get("context"), Context.class);
37             }
38             if(new Facade_ws<T>(clazz).createEntity(object, context)){
39                 result = "{\data\":{\message\":true, \operation\":\create\", \entity\":\"+clazz.getSimpleName()+\"}, \success\":true}";
40             }else{
41                 result="{\data\":{\message\":\"+errorMessage+\", \code\":+ERROR_OPERATION+, \operation\":\create\", \"
42                     + \"\entity\":\"+clazz.getSimpleName()+\"}, \success\":false}";
43             }
44         } catch (Exception e) {
45             result="{\data\":{\message\":\"+errorMessage+\", \code\":+EXCEPTION+, \operation\":\create\", \"
46                 + \"\entity\":\"+clazz.getSimpleName()+\"}, \success\":false}";
47             e.printStackTrace();
48         }finally{
49             return result;
50         }
51     }

```

Figura 28 - Trecho de Código da Classe Abstrata *Resource*.

Ainda na Figura 28, é possível localizar o método *create* (linha 28), responsável por criar um recurso. Esse método recebe como parâmetro uma *String* que representa a entidade formada na especificação JSON e retorna uma outra *String* referente ao protocolo de comunicação que será explicado na Seção 4.3. Como pode-se observar na figura, o método obtém um objeto (linha 32), referente a uma entidade do componente *ubi_model*, e o contexto do cliente (linha 35) no momento dessa operação. O resultado dessa operação é uma *String* que segue o protocolo especificado e que será serializada no formato JSON.

```

14 @Path("/person")
15 public class PersonResource extends Resource<Person>{
16     public PersonResource() {
17         super(Person.class);
18     }
19
20     @SuppressWarnings("finally")
21     @POST
22     @Path("/login")
23     public String login(String entity) { //TODO utilizar contexto seguro sobre SSL
24         String result="";
25
26         JsonObject json = new JsonParser().parse(entity).getAsJsonObject().get("entity").getAsJsonObject();
27         Context context = new GsonBuilder().excludeFieldsWithoutExposeAnnotation().create()
28             .fromJson(new JsonParser().parse(entity).getAsJsonObject().get("context"), Context.class);
29
30         try {
31             if(new Facade_ws<Person>(clazz).login(json.get("email").getAsString(), json.get("pass").getAsString(), context)){
32                 result = "{\data\":{\message\":true, \operation\":\login\", \entity\":\"+clazz.getSimpleName()+\"}, \success\":true}";
33             }else{
34                 result="{\data\":{\message\":\"+errorMessage+\", \code\":+ERROR_OPERATION+, \operation\":\login\", \"
35                     + \"\entity\":\"+clazz.getSimpleName()+\"}, \success\":false}";
36             }
37         } catch (Exception e) {
38             result = "{\data\":{\message\":\"+errorMessage+\", \code\":+EXCEPTION+, \operation\":\login\", \"
39                 + \"\entity\":\"+clazz.getSimpleName()+\"}, \success\":false}";
40             e.printStackTrace();
41         }finally{
42             return result;
43         }
44     }
45 }

```

Figura 29 - Trecho de Código da Classe *PersonResource*.

A Figura 29 ilustra a relação de herança, apresentada no diagrama da Figura 26, através da classe *PersonResource*. Como pode ser observado na figura, esta classe é responsável por fornecer o acesso à entidade *Person*. A linha 14 mostra uma anotação de referência ao caminho do recurso. Já a linha 22 apresenta a declaração do caminho complementar de acesso ao método *login* da linha 23. O método HTTP é indicado na linha 21.

4.2 CONSUMIDOR DO SERVIÇO

Uma vez que o serviço esteja disponível, é necessário fazer o cliente manipular as solicitações e tratar as resposta de acordo com o protocolo estabelecido pelo serviço. Para isso foi criada a classe *ConnectionWebService*, no cliente Android. Ela é uma classe genérica que permite manipular qualquer recurso disponibilizado pelo SW. Na Figura 30, são apresentadas algumas chamadas a essa classe para acessar o recurso *Person*.

```

37 @Override
38 public Person getPerson(long idPerson) {
39     return new ConnectionWebService<Person>(androidContext, "person", Person.class).get(idPerson+"");
40 }
41
42 @Override
43 public Person getPerson(String email) {
44     return new ConnectionWebService<Person>(androidContext, "person", Person.class).get(email);
45 }
46
47 @Override
48 public boolean createPerson(Person person, Context contexto) {
49     person.setPass(SecurityUbi.encodeMD5(person.getPass()));
50     return new ConnectionWebService<Person>(androidContext, "person", Person.class).put(person, contexto);
51 }
52
53 @Override
54 public List<Person> getAllPerson() {
55     return new ConnectionWebService<Person>(androidContext, "person/listAll", Person.class)
56         .list(new TypeToken<ArrayList<Person>>(){}.getType());
57 }

```

Figura 30 - Trecho de Código da Classe *Facade*.

Como se pode observar na Figura 30, é bastante simples utilizar a classe *ConnectionWebService*, pois ela abstrai toda a complexidade da comunicação com o SW.

```

26 public class ConnectionWebService <T>{
27     private enum RequestMethod {
28         DELETE, GET, POST, PUT
29     }
30
31     private final String CATEGORIA = "CONNECTION";
32     private String resource;
33     private RequestMethod method;
34     private Class<T> clazz;
35     private String params;
36     private Context context;
37     private String result;
38
39     public ConnectionWebService(Context context, String resource, Class<T> clazz) {
40         this.context = context;
41         this.resource = resource;
42         this.clazz = clazz;
43         result = "";
44         params = "";
45     }
46
47     @SuppressWarnings("finally")
48     public T get(String id){
49         T entity = null;
50         this.setMethod(RequestMethod.GET);
51         this.setParams(id);
52         try {
53             this.setResult(new ConnectionTask().execute().get());
54             if(this.getResult()!=null && !this.getResult().equals("")) entity = getResponseEntity();
55         }catch(Exception e){
56             e.printStackTrace();
57         }finally {
58             return entity;
59         }
60     }

```

Figura 31 - Trecho de Código nº 1 da Classe *ConnectionWebService*.

A Figura 31 apresenta parte da classe *ConnectionWebService*. Seu construtor (linha 39) recebe três parâmetros: o contexto da aplicação Android, o caminho e o tipo da classe do recurso em questão. O método *get*, da linha 48, retorna um objeto do tipo especificado de acordo com o *id* informado. Nas linhas 50 e 51, são informados o método HTTP e os parâmetros necessários para o consumo do serviço, respectivamente. Na linha 53, é lançada a requisição para o consumo do SW, através de uma tarefa assíncrona. O resultado obtido é tratado e transformado em um objeto pelo método *getResponseEntity()*, utilizado na linha 54.

```

200 public String requestProcess(){
201     String result = "";
202     try{
203         HttpClient httpClient = new DefaultHttpClient();
204         String uri = "http://" + Constant.server + ":" + Constant.port + "/" + Constant.resource + "/" + this.getResource();
205         HttpRequestBase request = null;
206         switch (method) {
207             case GET:
208                 if(getParams()!=null)uri+="?entity="+getParams();
209                 request = new HttpGet(uri);
210                 break;
211             case POST:
212                 request = new HttpPost(uri);
213                 ((HttpPost)request).setEntity(new StringEntity(getParams(), "UTF-8"));
214                 break;
215             case PUT:
216                 request = new HttpPut(uri);
217                 ((HttpPut)request).setEntity(new StringEntity(getParams(), "UTF-8"));
218                 break;
219             case DELETE:
220                 request = new HttpDelete(uri);
221         }
222         request.addHeader("Accept", "application/json;charset=UTF-8");
223         request.addHeader("Content-type", "application/json;charset=UTF-8");
224         request.addHeader("Accept-Charset", "utf-8");
225
226         HttpResponse response = httpClient.execute(request);
227         HttpEntity entity = response.getEntity();
228         InputStream in = entity.getContent();
229         if(in !=null){
230             result = ConnectionWebService.read(in);
231             setResult(result);
232         }
233     }catch (Exception e){
234         e.printStackTrace();
235     }finally {
236         return result;
237     }

```

Figura 32 - Trecho de Código nº 2 da Classe *ConnectionWebService*.

O método *requestProcess()*, apresentado na Figura 32, é responsável por realizar a requisição ao SW. Como esse processo pode ser demorado, ele é chamado dentro de uma tarefa assíncrona e o cliente é notificado do progresso da requisição. A URI da requisição (linha 204) é formada por alguns valores constantes, que possuem o endereço e a porta do servidor, e o caminho do recurso, informado no construtor da classe. Então, é identificado o método HTTP e definidos seus parâmetros. O tipo de dados das transações entre o cliente e o servidor são adicionados ao cabeçalho da requisição (linhas 222 e 223). Posteriormente, o cliente envia a solicitação (linha 226), recebe a resposta (linha 227) e a transforma para *String* (linha 230). O resultado será uma *String* no formato JSON, seguindo o protocolo aqui estabelecido.

```

294 private boolean isSuccess(){
295     boolean result = false;
296     JsonObject json = new JsonParser().parse(this.getResult()).getAsJsonObject();
297
298     if(json.get("success").getAsBoolean()){
299         result = json.getAsJsonObject("data").get("message").getAsBoolean();
300     }else{
301         this.setMsg(json.getAsJsonObject("data").get("message").getString());
302     }
303     return result;
304 }
305
306 private T getResponseEntity(){
307     T entity=null;
308     JsonObject json = new JsonParser().parse(this.getResult()).getAsJsonObject();
309     if(json.get("success").getAsBoolean()){
310         entity = new Gson().fromJson(json.getAsJsonObject("data").getAsJsonObject("object"), clazz);
311     }else{
312         this.setMsg(json.getAsJsonObject("data").get("message").getString());
313     }
314     return entity;
315 }

```

Figura 33 - Trecho de Código nº 3 da Classe *ConnectionWebService*.

Os métodos *isSuccess()* e *getResponseEntity()*, apresentados na Figura 33, interpretam o protocolo de comunicação resultante da solicitação, informando se a transação foi bem sucedida e retornando o objeto do resultado, respectivamente. Caso ocorra uma falha interna no servidor, será enviada uma mensagem de notificação ao usuário, capturada nas linhas 301 e 312, em ambos os métodos.

4.3 PROTOCOLO DE COMUNICAÇÃO

As informações recebidas e oferecidas pelo SW são apresentadas por meio de estruturas de dados serializadas no formato JSON. O JSON é um formato leve de troca de dados, de fácil leitura por humanos, fácil processamento em sistemas e baseado na linguagem *JavaScript*. Por ser facilmente manipulado em diversas plataformas, torna-se prático especificar uma padrão de mensagens a fim de definir um protocolo de comunicação. Os padrões de mensagens utilizados no ambiente *MobiLE+* serão apresentados a seguir.

A Figura 34 representa uma mensagem de resposta do servidor indicando que a operação foi bem sucedida e fornecendo o recurso solicitado.

```

1  {
2      data:{
3          message:    true,
4          operation:  'NOME DA OPERAÇÃO',
5          entity:     'NOME DA ENTIDADE',
6          object:     OBJETO
7      },
8      success:true
9  }

```

Figura 34 - Mensagem de Saída do Servidor de uma Operação bem sucedida

Já a Figura 35 demonstra o formato de uma mensagem que o servidor envia quando acontece algum erro durante a requisição, identificando ao cliente a natureza desse erro através de uma mensagem padrão de notificação ao usuário.

```

1  {
2      data:{
3          message:    'MENSAGEM DE ERRO',
4          code:       'CÓDIGO DO ERRO',
5          operation:  'NOME DA OPERAÇÃO',
6          entity:     'NOME DA ENTIDADE',
7          object:     null
8      },
9      success:    false
10 }

```

Figura 35 - Mensagem de Saída do Servidor de um Erro Durante a Operação

Os atributos utilizados nas mensagens apresentadas nas figuras 34 e 35 são:

- *data*: objeto composto por elementos da resposta do servidor;
- *success*: indica se a transação foi realizada com sucesso ou não;
- *message*: mensagem de notificação ao usuário;
- *operation*: operação realizada (*read*, *create*, *update*, *delete*, *list*);
- *entity*: nome da entidade referente ao recurso;
- *object*: recurso solicitado pelo cliente;
- *code*: código numérico indicando a natureza do erro.

5 MODELAGEM DO SISTEMA MULTIAGENTE

A organização do SMA é do tipo comunidade de especialistas, pois cada um dos cinco tipos de agentes criados para este trabalho encontra-se no mesmo nível, sendo cada um deles especialista em determinada tarefa. Os agentes interagem entre si através de um protocolo de comunicação previamente estabelecido pelo FIPA (*Foundation for Intelligent Physical Agents*) (FIPA, 2002). O SMA foi desenvolvido utilizando a plataforma de desenvolvimento e execução de agentes de software conhecida como JADE (*Java Agent Development Framework*) (BELLIFEMINE et al., 2007).

O SMA é composto de cinco tipos de agentes: Agente Usuário (*User Agent* - UAg), Agente Coletor (*Collector Agent* - CAg), Agente Recomendador (*Recommender Agent* - RAg), Agente de Interface (*Interface Agent* - IAg) e Agente DF (*Directory Facilitator*).

Os CAg realizam o monitoramento e a captura de informações contextuais do usuário, a atualização do contexto, o refinamento e enriquecimento semântico (utilizando o SES) dos dados capturados e a detecção de outras atividades ocorridas fora dos limites do ambiente MobiLE+, com a finalidade de melhor traçar o perfil do usuário. Eles registram o serviço contendo o contexto do usuário atualizado.

Os UAg são responsáveis por recuperar o contexto e o perfil do usuário, acompanhar e sugerir o preenchimento do perfil e/ou questionário ILS, realizar análise do histórico de escolhas para ajustar o estilo de aprendizagem e identificar horário mais apropriado para a recomendação, por meio de uma análise do histórico de atividades. O processo de recomendação será desencadeado por este agente, ao analisar as informações contextuais e o momento mais conveniente ao usuário. Para isso, ele registra um serviço no DF indicando a intenção da recomendação e fornecendo o contexto do usuário, capturado pelo CAg.

O RAg tem o intuito de detectar conteúdos educacionais adequados ao contexto do estudante, levando em consideração as informações consultadas no agente DF e as informações da base de dados dos recursos e perfis enriquecidos semanticamente. Internamente o RAg utiliza o SRHU para selecionar os conteúdos a serem recomendados. Após identificar os conteúdos educacionais adequados, o RAg cadastra um serviço no agente DF com novas informações, as quais servirão ao IAg.

O IAg é responsável por notificar o usuário sobre a disponibilidade de conteúdo recomendado, a necessidade de completar o perfil ou preencher o formulário ILS. Desta

forma, o IAg utiliza o serviço fornecido pelo RAg para obter os recursos recomendados e realizar a notificação ao usuário, por meio do servidor GCM.

O agente DF possui o comportamento voltado para a mediação da comunicação entre os outros agentes. Sua função principal é fornecer uma arquitetura do tipo “quadro-negro”, onde agentes escrevem informações, procuram por informações escritas por outros agentes e conseguem, através do serviço provido pelo DF, se comunicar com o agente que escreveu aquela informação.

Para a modelagem do SMA presente no MobiLE+, optou-se por seguir a metodologia MAS-CommonKADS+, devido à sua maior expressividade na tarefa de definição dos agentes, seus comportamentos e técnicas de IA utilizadas (MORAIS II, 2010).

Devido ao fato dos agentes não se comunicarem diretamente, alguns modelos da metodologia foram suprimidos, pois, no contexto desse trabalho, não apresentaria prejuízo ao entendimento do SMA desenvolvido.

5.1 MODELO DE TAREFAS

Segundo Morais II (2010), o modelo de tarefas descreve as atividades necessárias para atingir determinada meta. No caso específico do MobiLE+, a tarefa principal do SMA é a adaptação do ambiente em relação aos usuários. Porém, para que essa tarefa seja concluída com sucesso, uma série de subtarefas necessitam ser realizadas a priori. O modelo de tarefas para o SMA do MobiLE+ pode ser visto na Figura 36.

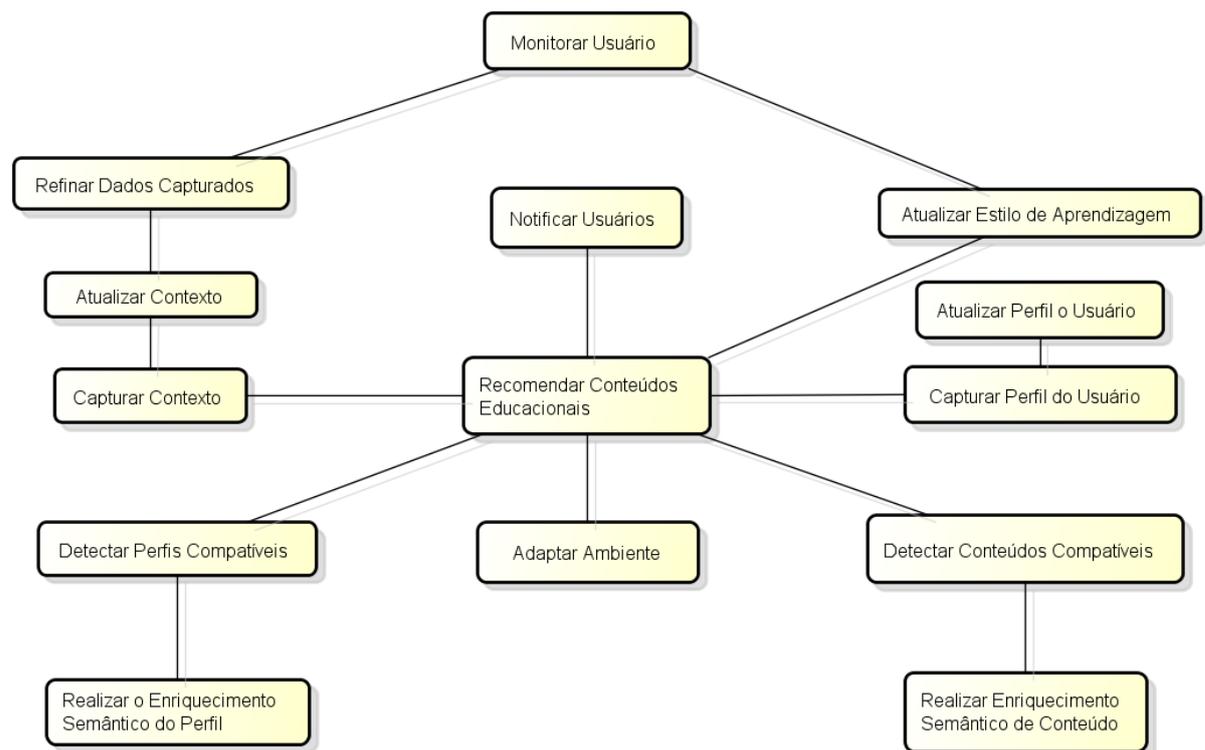


Figura 36 - Diagrama de Tarefas do SMA.

5.2 MODELO DE INTERAÇÃO

O modelo de interação fornece uma visão do comportamento dinâmico do SMA, onde são representadas as trocas de mensagens entre os agentes. A Figura 37 mostra o modelo de interação entre os agentes que compõem do SMA do MobiLE+.



Figura 37 - Modelo de Interação do SMA.

No modelo apresentado na Figura 37, é possível visualizar toda a interação que ocorre entre os agentes, o banco de dados, SES e o SRHU. O diagrama mostra que a comunicação realizada entre os agentes é intermediada pelo Agente DF, responsável por estabelecer uma arquitetura de “quadro-negro”. A comunicação com os componentes de enriquecimento semântico (SES) e de recomendação (SRHU) é realizada exclusivamente pelo CAg e RAg, respectivamente.

5.3 MODELO DO AGENTE COLETOR

Para modelar os agentes de um SMA, a metodologia indica o uso de *templates* textuais, os quais descrevem uma série de informações sobre esses agentes. Na Tabela 5 é apresentado o *template* textual do CAg. O CAg é um agente Reativo Simples, pois suas ações dependem exclusivamente da percepção atual.

Tabela 5 - *Template* textual do Agente Coletor.

Agente: CAg
<p>Objetivo</p> <p>Monitorar e capturar informações contextuais do usuário, refinar e enriquecer semanticamente dados capturados e atualizar o contexto.</p>
<p>Parâmetro de Entrada</p> <p>Identificação do usuário no sistema.</p>
<p>Parâmetros de Saída</p> <p>Dados coletados (ver Subseção 3.2.1) relacionados ao contexto do usuário e ao estado do dispositivo.</p>
<p>Condição de Ativação</p> <p>O agente percebe que o usuário realizou alguma atividade ou o estado (localização, bateria e conectividade) do dispositivo foi alterado (ver Subseção 3.2.1).</p>
<p>Condição de Finalização</p> <p>Quando o agente registra um serviço no DF disponibilizando a última leitura de contexto obtida.</p>
<p>Informações Associadas</p>

Possui um comportamento de execução única para cadastro de serviço com o contexto no DF, sendo criado, portanto, um CAg para cada usuário ativo no sistema. Cada CAg também possui um comportamento cíclico para ficar monitorando por requisições do UAg. O processo de enriquecimento semântico, dos dados capturados, será realizado pelo SES.

Descrição

Este agente possui o comportamento totalmente voltado para monitoramento das atividades dos usuários, capturando e enriquecendo os dados coletados.

Para entender melhor o fluxo de execução do CAg é apresentado, na Figura 38, o diagrama de atividades desse agente.

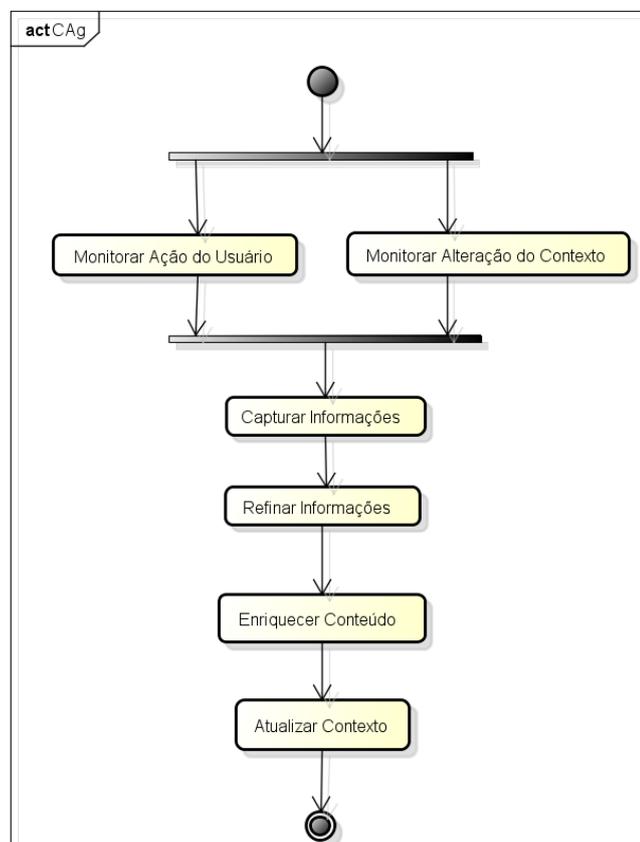


Figura 38 - Diagrama de Atividades do Agente Coletor.

5.4 MODELO DO AGENTE USUÁRIO

Na Tabela 6 é exposto o *template* textual do UAg. Este agente pode ser classificado como Reativo Baseado em Modelo, pois realiza uma análise do histórico de ações do usuário

(modelo interno) para, em conjunto com suas percepções atuais, identificar qual o melhor horário da recomendação.

Tabela 6 - *Template* textual do Agente Usuário.

Agente: UAg
<p>Objetivo</p> <p>Recuperar contexto, capturar e atualizar o perfil do usuário, acompanhar e sugerir o preenchimento do perfil e/ou questionário ILS, ajustar o estilo de aprendizagem e identificar o horário de disponibilidade para a recomendação.</p>
<p>Parâmetro de Entrada</p> <p>Identificador do usuário no sistema.</p>
<p>Parâmetros de Saída</p> <p>Perfil do usuário e estilo de aprendizagem atualizados e horário que deverá ocorrer a recomendação.</p>
<p>Condição de Ativação</p> <p>Quando o agente percebe que chegou a hora preferida de estudo do usuário, existe um contexto disponível (C_{Ag}) e ainda não foi recomendado nenhum conteúdo.</p>
<p>Condição de Finalização</p> <p>Quando o agente registra um serviço no DF para iniciar o processo de recomendação.</p>
<p>Informações Associadas</p> <p>Possui um comportamento cíclico para identificar o melhor horário para a recomendação e registrar um serviço notificando ao R_{Ag} que inicie o processo de recomendação. Também informa ao I_{Ag} sobre a necessidade de notificar o usuário para completar seu perfil ou preencher o questionário ILS.</p>
<p>Descrição</p> <p>Este agente busca fornecer o perfil e o estilo de aprendizagem do usuário atualizados, assim como identificar o melhor momento para a recomendação.</p>

Para entender melhor o fluxo de execução do UAg é apresentado, na Figura 39, o diagrama de atividades desse agente.

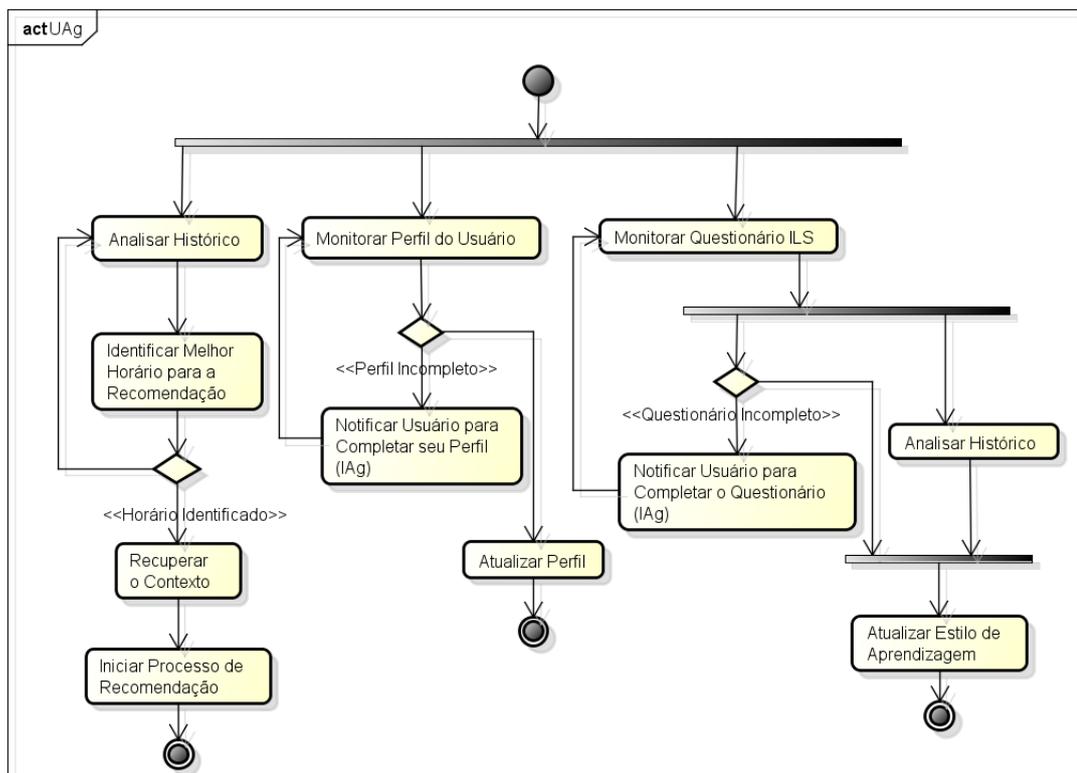


Figura 39 - Diagrama de Atividades do Agente Usuário.

5.5 MODELO DO AGENTE RECOMENDADOR

A Tabela 7 exibe o *template* textual do RAg. Este é um Agente Reativo Baseado em Modelo, pois, ao perceber o serviço no DF, ele utiliza informações mantidas em uma base de dados para decidir quais recursos recomendar.

Tabela 7 - *Template* textual do Agente Recomendador.

Agente: RAg	
Objetivo	Detectar conteúdos educacionais adequados ao perfil e contexto do usuário, utilizando o SRHU.
Parâmetro de Entrada	Perfil do usuário, contexto, estilo de aprendizagem e base de dados dos recursos e perfis enriquecidos semanticamente.
Parâmetros de Saída	

Conteúdos educacionais adequados ao usuário e identificador do usuário no sistema.
<p>Condição de Ativação</p> <p>Quando o agente percebe, no DF, um serviço cadastrado para iniciar o processo de recomendação.</p>
<p>Condição de Finalização</p> <p>Quando o agente registra um serviço no DF informando quais conteúdos serão recomendados.</p>
<p>Informações Associadas</p> <p>Possui um comportamento cíclico para monitorar o serviço fornecido pelo UAg e iniciar o processo de recomendação.</p>
<p>Descrição</p> <p>Este agente é responsável por recomendar conteúdos educacionais que atendam as necessidades dos usuários.</p>

Para entender melhor o fluxo de execução do UAg é apresentado, na Figura 40, o diagrama de atividades desse agente.

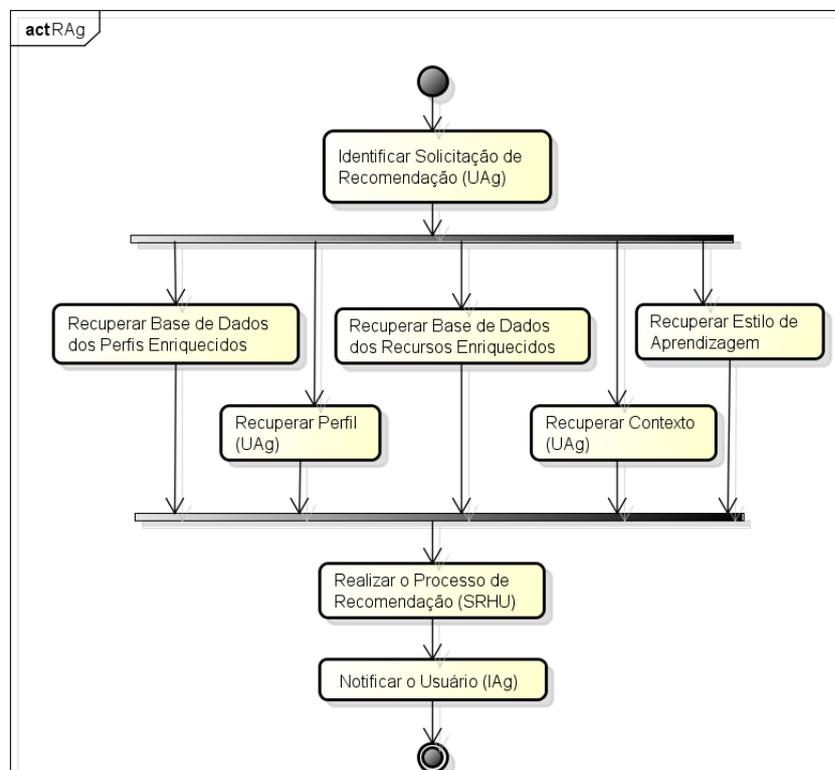


Figura 40 - Diagrama de Atividades do Agente Recomendador.

5.6 MODELO DO AGENTE DE INTERFACE

A Tabela 8 exibe o *template* textual do IAg. Esse agente é classificado como Agente Reativo Simples, visto que suas ações dependem somente da sua percepção atual, ou seja, da necessidade de notificar o usuário.

Tabela 8 - *Template* textual do Agente de Interface.

Agente: IAg
<p>Objetivo</p> <p>Notificar o usuário quanto à existência de conteúdo recomendado, para completar o formulário ou preencher o questionário ILS.</p>
<p>Parâmetro de Entrada</p> <p>Conteúdo educacional fornecido pelo RAg.</p>
<p>Parâmetros de Saída</p> <p>Notificação enviada para o dispositivo móvel do usuário.</p>
<p>Condição de Ativação</p> <p>Quando o agente percebe um serviço no DF informando a existência de conteúdo a ser recomendado ou outra necessidade de notificação.</p>
<p>Condição de Finalização</p> <p>Notificação recebida pelo usuário.</p>
<p>Informações Associadas</p> <p>Possui um comportamento cíclico para monitorar o serviço fornecido pelo RAg e enviar a notificação ao usuário.</p>
<p>Descrição</p> <p>Este agente possui o comportamento voltado para a verificação da necessidade de notificar o usuário.</p>

5.7 MODELO DO AGENTE DF

O Agente DF realiza o papel de Mediador entre os agentes. A Tabela 9 mostra o seu *template* textual.

Tabela 9 - *Template* textual do Agente DF

Agente: DF
<p>Objetivo</p> <p>Possibilitar que os agentes possam cadastrar e procurar por serviços oferecidos por outros agentes (serviço de páginas amarelas).</p>
<p>Parâmetro de Entrada</p> <p>Registros de serviços por parte dos agentes.</p>
<p>Parâmetros de Saída</p> <p>Lista de serviços e os respectivos agentes responsáveis pelos mesmos.</p>
<p>Condição de Ativação</p> <p>Quando a plataforma JADE é inicializada.</p>
<p>Condição de Finalização</p> <p>Caso a plataforma JADE seja finalizada.</p>
<p>Informações Associadas</p> <p>É um agente inicializado pela própria plataforma JADE como um dos componentes integrantes do padrão FIPA.</p>
<p>Descrição</p> <p>Este agente possui o comportamento voltado para a mediação entre os outros agentes. Sua função principal é fornecer uma arquitetura do tipo “quadro-negro”, onde agentes escrevem informações, procuram por informações escritas por outros agentes e conseguem, através do serviço provido pelo DF, se comunicar com o agente que escreveu aquela informação.</p>

A principal vantagem em utilizar o JADE reside no fato de que é possível utilizar todos os componentes da especificação FIPA, visto que o JADE atende todas as especificações deste padrão e ainda o estende. Um desses componentes é o agente DF.

Outra vantagem dessa plataforma consiste na possibilidade de consultar e utilizar, através do MTS (*Message Transport Service*), DFs que estejam sendo executados em outras plataformas de agentes (*Agent Platforms* - APs). Assim, o DF utilizado na plataforma JADE deste trabalho pode, por exemplo, ser consultado, por meio da internet, por um agente que esteja sendo executado em outro local, através do protocolo MTP (*Message Transport Protocol*). Essa característica permite a criação de uma rede de colaboração entre os agentes,

mesmo que estejam localizados, por exemplo, em redes e servidores distintos. No presente trabalho, este protocolo foi utilizado para comunicação entre agentes que se encontravam na mesma plataforma, visto que os quatro agentes criados e o DF se encontravam no mesmo servidor.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Os dispositivos móveis estão cada vez mais presentes no nosso dia-a-dia, chegando a desempenhar um papel ativo ao longo da nossa vida. Isso tem se tornado uma realidade para um número crescente de pessoas, o que evidencia seu grande potencial de aplicação em diversas áreas. Nos últimos anos, esses dispositivos vêm se tornando o centro das atenções do desenvolvimento tecnológico, fazendo com que antigas limitações sejam parte do passado e, desta forma, ampliando seu horizonte de possibilidades.

No âmbito da educação, muito se tem pesquisado a respeito de novos meios de prover aprendizagem, de forma a estimular cada vez mais os estudantes. Logo, novas metodologias e abordagens educacionais têm sido criadas, com o intuito de alcançar melhores resultados pedagógicos em um público cada vez mais dinâmico e diversificado.

A comunidade científica tem dedicado esforços para tornar os ambientes virtuais de aprendizagem mais próximos da realidade do ensino presencial. A utilização de dispositivos móveis como meio de prover ensino, apesar da sua característica motivadora, trouxe novos desafios.

A aprendizagem móvel permite que o usuário tenha acesso a conteúdos ao longo do dia, acarretando uma exposição maior a recursos educacionais. Esse cenário pode ocasionar perda de desempenho do usuário, levando a situações onde terá que realizar escolhas difíceis, mesmo ainda não estando apto para isto. Um ambiente de aprendizagem ubíqua é capaz de adaptar-se às necessidades específicas dos usuário. Essa adaptação pode ser explorada por meio da adequação de conteúdo educativo ao usuário, levando em consideração suas preferências de aprendizagem.

Fornecer conteúdo ao usuário de acordo com suas características cognitivas não é uma tarefa trivial, visto que é necessário conhecer as necessidades e preferências de aprendizagem deste. As mídias sócias, quando exploradas de forma correta, podem nos fornecer muitas informações a respeito do usuário, o que vem a auxiliar a traçar seu perfil cognitivo.

Outro fator que deve ser levado em consideração é a localização do usuário, uma vez que oportunidades de aprendizagem podem surgir no momento que o usuário explora o ambiente físico.

Pensando nisso, o presente trabalho apresentou um ambiente de aprendizagem ubíqua que monitora e captura informações dos usuários, enriquece semanticamente os perfis e conteúdos e realiza a recomendação personalizada de conteúdos. Para tanto, o MobiLE+ leva

em consideração uma série de características, conforme mostrado ao longo deste trabalho. Seu intuito é aperfeiçoar o processo de ensino-aprendizagem de forma transparente aos usuários, direcionando e provocando um maior interesse no estudo por parte destes.

Apesar do trabalho feito até o momento, há algumas melhorias que podem ser realizadas. Assim, como trabalho futuro, pretende-se:

- Migrar o agente de interface para o dispositivo móvel, permitindo uma comunicação com o usuário mais expressiva;
- Identificar o estado de sobrecarga cognitiva do usuário, utilizando o método de avaliação NASA-TLX (desenvolvido pela NASA), para melhor adaptar o ambiente de aprendizagem ao estado atual do usuário (HART, 2006);
- Classificar os conteúdos com relação à sua carga cognitiva e dimensão do estilo de aprendizagem;
- Disponibilizar a aplicação em forma de API, para que possa ser utilizada por outros desenvolvedores;
- Testar o ambiente desenvolvido em áreas específicas (área médica, petrolífera, etc.) para verificar a adequabilidade da recomendação em contextos específicos; e
- Realizar um estudo de caso, com uma turma de um dos cursos de especialização fornecido pela UnA-SUS (Universidade Aberta do Sistema Único de Saúde), para validar o ambiente e mensurar o quanto este contribui para o processo de ensino-aprendizagem.

7 REFERÊNCIAS

- AHMED, S. F.; CHANG, M. Recommend computer studies courses for students taken based on supported mobile learning modes. **2012 IEEE International Conference on Technology Enhanced Education (ICTEE)**, p. 1–5, 2012. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6208652>>. .
- ARTERO, A. O. **INTELIGENCIA ARTIFICIAL - TEORICA E PRATICA**. LIVRARIA DA FISICA, 2009.
- BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. **Developing multi-agent systems with JADE**. 2007.
- BLUETOOTH, O. Bluetooth. Disponível em: <<https://www.bluetooth.org/en-us>>. Acesso em: 25/1/2014.
- BRASIL. **Decreto nº 5622, de 19 de dezembro de 2005. Regulamenta o art. 80, da Lei nº 9.394, de 20 de dezembro de 1996, que estabelece as diretrizes e bases da educação nacional**. Brasília, DF, 2005.
- CARVER, C. A.; HOWARD, R. A.; LANE, W. D. Enhancing student learning through hypermedia courseware and incorporation of student learning styles. **IEEE Transactions on Education**, v. 42, n. 1, p. 33–38, 1999. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=746332>>. Acesso em: 7/1/2014.
- CAZELLA, S. C.; SILVA, K. K.; BHEAR, P.; SCHNEIDER, D.; FREITAS, R. Recomendando objetos de aprendizagem baseado em competências em EAD. **RENOTE - Revista Novas Tecnologias na Educação**, v. 9, 2011. Disponível em: <<http://seer.ufrgs.br/renote/article/view/25123>>. Acesso em: 7/1/2014.
- DEITEL, H. M. **Java: como programar**. 6th ed. São Paulo, 2005.
- DESPOTAKIS, D.; THAKKER, D.; DIMITROVA, V. Capturing the semantics of individual viewpoints on social signals in interpersonal communication. **Semantic Web Journal**, , n. Special Issue on Personal and Social Semantic Web, 2012. Disponível em: <http://semantic-web-journal.org/sites/default/files/swj207_0.pdf>. Acesso em: 7/1/2014.
- FAZLINA, S.; MANAP, A. A.; RIAS, R. M. Mobile Learning Awareness among Students at Higher Learning Institutes: A Case Study. 2013 International Conference on Informatics and Creative Multimedia. **Anais...** p.226–229, 2013. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6702814>>. Acesso em: 16/1/2014.
- FELDER, R. M.; SILVERMAN, L. K. Learning and teaching styles in engineering education. **Engineering education**, v. 78, p. 674–681, 1988.

FELDER, R. M.; SOLOMAN, B. A. INDEX OF LEARNING STYLES (ILS). Disponível em: <<http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html>>. .

FELDER, R. M.; SPURLIN, J. Applications, Reliability, and Validity of the Index of Learning Styles. **International Journal of Engineering Education**, v. 21, p. 103–112, 2005.

FIPA. **FIPA ACL Message Structure Specification**. 2002.

FRANKLIN, S.; GRAESSER, A. Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages. **Anais...** p.21–35, 1997. London, UK: Springer-Verlag.

GICQUEL, P.-Y.; LENNE, D.; MOULIN, C. Design and use of CALM : An ubiquitous environment for mobile learning during museum visit. 2013 Digital Heritage International Congress (DigitalHeritage). **Anais...** p.645–652, 2013. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6744831>>. Acesso em: 26/3/2014.

GOOGLE ANDROID DEVELOPER, I. Android. Disponível em: <<http://developer.android.com/index.html>>. Acesso em: 14/1/2014.

GOOGLE ANDROID GCM, I. GCM - Google Cloud Messaging. Disponível em: <<http://developer.android.com/google/gcm/index.html>>. Acesso em: 14/1/2014.

GOOGLE GEOCODING API, I. Google Geocoding API. Disponível em: <<https://developers.google.com/maps/documentation/geocoding/?hl=pt-br>>. Acesso em: 13/1/2014.

GOOGLE PLACE API, I. Google Place API. Disponível em: <<https://developers.google.com/places/documentation/?hl=pt-br>>. Acesso em: 13/1/2014.

GRAF, S.; KINSHUK. Using Cognitive Traits for Improving the Detection of Learning Styles. 2010 Workshops on Database and Expert Systems Applications. **Anais...** p.74–78, 2010. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5592009>>. Acesso em: 7/1/2014.

HAMMER-LAHAV, E. OAuth. Disponível em: <<http://oauth.net/>>. Acesso em: 14/1/2014.

HART, S. G. NASA-Task Load Index (NASA-TLX); 20 Years Later. Proceedings of the Human Factors and Ergonomics Society Annual Meeting. **Anais...** p.904–908, 2006.

HTTP, W. HTTP - Hypertext Transfer Protocol. Disponível em: <<http://www.w3.org/Protocols/>>. Acesso em: 16/1/2014.

HUBNER, J. F. **Um modelo de reorganização de sistemas multiagentes**, 2003. Universidade de São Paulo.

JARDIM DA SILVA, L. R. **MSSearch: Busca Semântica de Objetos de Aprendizagem OBAA com Suporte a Alinhamento Automático de Ontologias**, 2013. Universidade do Vale do Rio dos Sinos.

JERSEY, O. Jersey - RESTful Web Services in Java. Disponível em: <<https://jersey.java.net/>>. Acesso em: 16/1/2014.

JIUGEN, Y.; RUONAN, X.; XIAOQIANG, H. Constructing informal learning mode based on social software. 2011 6th International Conference on Computer Science & Education (ICCSE). **Anais...** p.1227–1230, 2011. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6028853>>. Acesso em: 7/1/2014.

JSON, O. JSON - JavaScript Object Notation. Disponível em: <<http://www.json.org/>>. Acesso em: 16/1/2014.

KULJIS, J.; LIU, F. A comparison of learning style theories on the suitability for learning. Web Technologies, Applications, and Services. **Anais...** p.191–197, 2005.

LI, M.; OGATA, H.; HOU, B.; UOSAKI, N.; YANO, Y. Personalization in Context-aware Ubiquitous Learning-Log System. 2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education. **Anais...** p.41–48, 2012. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6185078>>. Acesso em: 7/1/2014.

LITZINGER, T. A.; LEE, S. H.; WISE, J. C.; FELDER, R. M. A Psychometric Study of the Index of Learning Styles©. **Journal of Engineering Education**, v. 96, n. 4, p. 309–319, 2007. Disponível em: <<http://doi.wiley.com/10.1002/j.2168-9830.2007.tb00941.x>>. Acesso em: 7/1/2014.

MANDULA, K.; MEDA, S. R.; JAIN, D. K.; KAMBHAM, R. Implementation of Ubiquitous Learning System Using Sensor Technologies. 2011 IEEE International Conference on Technology for Education. **Anais...** p.142–148, 2011. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6004374>>. Acesso em: 30/12/2013.

MANDULA, K.; MEDAY, S. R.; MURALIDHARAN, V.; PARUPALLI, R. A student centric approach for mobile learning video content development and instruction design. Advanced Communication Technology (ICACT), 2013 15th International Conference on. **Anais...** p.386–390, 2013. IEEE.

MARTINS, H. N. J.; COSTA, E. B.; OLIVEIRA, T. T. M.; BITTENCOURT, I. I. Sistema de Recomendação Híbrido para Bibliotecas Digitais que Suportam o Protocolo OAI-PMH. **Anais do XXII Simpósio Brasileiro de Informática na Educação**, p. 140–149, 2011. Disponível em: <<http://ceie-sbc.tempsite.ws/pub/index.php/sbie/article/view/1584>>. Acesso em: 7/1/2014.

MONTEIRO, B.; JÁCOME JÚNIOR, L. YouUbi. Disponível em: <<http://www.youbi.com/youbi/index.php/pt/>>. Acesso em: 15/1/2014.

MOORE, P.; HU, B.; JACKSON, M.; WAN, J. Intelligent Context for Personalised M-Learning. 2009 International Conference on Complex, Intelligent and Software Intensive Systems. **Anais...** p.247–254, 2009. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5066794>>. Acesso em: 16/1/2014.

MOORE, P.; HU, B.; WAN, J. Smart-Context: A Context Ontology for Pervasive Mobile Computing. **The Computer Journal**, v. 53, n. 2, p. 191–207, 2008. Disponível em: <<http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/bxm104>>. Acesso em: 16/1/2014.

MOORE, P.; PHAM, H. V. Personalized Intelligent Context-Aware E-Learning on Demand. 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems. **Anais...** p.965–970, 2012. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6245718>>. Acesso em: 16/1/2014.

MORAIS II, M. J. D. O. **MAS-CommonKADS+: Uma Extensão à Metodologia Mas-CommonKADS para Suporte ao Processo Detalhado de Sitemas Multiagentes Racionais**, 2010. Universidade Estadual do Ceará - UECE.

ORACLE JAVA EE, C. Java Enterprise Edition. Disponível em: <<http://www.oracle.com/technetwork/pt/java/javase/overview/index.html>>. Acesso em: 16/1/2014.

ORACLE JPA, C. JPA - Java Persistence API. Disponível em: <<http://www.oracle.com/technetwork/java/javase/tech/persistence-jsp-140049.html>>. Acesso em: 14/1/2014.

ORACLE REST, C. **RESTful Web Services**. 2006.

POSTGRESQL, O. PostgreSQL. Disponível em: <<http://www.postgresql.org/>>. Acesso em: 16/1/2014.

REDECKER, C.; ALA-MUTKA, K.; PUNIE, Y. Learning 2.0: The impact of social media on learning in Europe. **JRC Scientific and Technical Report**, 2010. Disponível em: <<http://www.ict-21.ch/com-ict/IMG/pdf/learning-2.0-EU-17pages-JRC56958.pdf>>. Acesso em: 7/1/2014.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2010.

SACCOL, A.; SCHLEMMER, E.; BARBOSA, J. **m-learning e u-learning - Novas perspectivas da aprendizagem móvel e ubíqua**. Person ed. São Paulo, 2010.

SCHAFER, J. BEN; KONSTAN, J.; RIEDI, J. Recommender systems in e-commerce. Proceedings of the 1st ACM conference on Electronic commerce - EC '99. **Anais...** p.158–166, 1999. New York, New York, USA: ACM Press. Disponível em: <<http://portal.acm.org/citation.cfm?doid=336992.337035>>. Acesso em: 7/1/2014.

SCIENTIAMOBILE, I. WURFL (Wireless Universal Resource FiLe). Disponível em: <<http://wurfl.sourceforge.net/>>. Acesso em: 14/1/2014.

SEED/MEC. **Referencias de Qualidade para Educação Superior a Distância**. Brasília, DF, 2007.

SHARPLES, M. The design of personal mobile technologies for lifelong learning. **Computers & Education**, v. 34, n. 3-4, p. 177–193, 2000. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0360131599000445>>. Acesso em: 16/1/2014.

SILVA, L. C. **MobiLE – Um Ambiente Multiagente de Aprendizagem Móvel para Apoiar a Recomendação Ubíqua de Objetos de Aprendizagem.**, 2012. Universidade Federal Rural do Semi-Árido.

SOMMERVILLE, I. Arquitetura orientada a serviços. **Engenharia de Software**. 9th ed., p.355–368, 2011. São Paulo: Person Prentice Hall.

TARAGHI, B. Ubiquitous Personal Learning Environment (UPLE). 2012 15th International Conference on Interactive Collaborative Learning (ICL). **Anais...** p.1–8, 2012. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6402139>>. Acesso em: 16/1/2014.

THAKKER, D.; DESPOTAKIS, D.; DIMITROVA, V. Taming digital traces for informal learning: a semantic-driven approach. ... **Learning for 21st Century ...**, v. 7563, p. 348–362, 2012. Disponível em: <http://link.springer.com/chapter/10.1007/978-3-642-33263-0_27>. Acesso em: 7/1/2014.

VIEIRA, F.; NUNES, M. DICA: Sistema de Recomendação de Objetos de Aprendizagem Baseado em Conteúdo. **Scientia Plena**, v. 8, p. 1–10, 2012. Disponível em: <<http://www.scientiaplena.org.br/ojs/index.php/sp/article/view/464>>. Acesso em: 7/1/2014.

W3C, O. Extensible Markup Language (XML). Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 16/1/2014.

WANG, M.; SHEN, R. Message design for mobile learning: Learning theories, human cognition and design principles. **British Journal of Educational Technology**, v. 43, n. 4, p. 561–575, 2012. Disponível em: <<http://doi.wiley.com/10.1111/j.1467-8535.2011.01214.x>>. Acesso em: 3/1/2014.

WEISER, M. The Computer for the 21st Century. **Scientific American**, v. 265, p. 66–75, 1991.

WOOLDRIDGE, M.; JENNINGS, N. R. Addressing different learning styles through course hypermedia. Knowledge Engineering Review. **Anais...** p.115–152, 1995.

YAU, J. Y.-K.; JOY, M. A Context-Aware Personalized M-learning Application Based on M-learning Preferences. 2010 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education. **Anais...** p.11–18, 2010. IEEE. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5476532>>. Acesso em: 16/1/2014.

ZYWNO, M. S. A Contribution to Validation of Score Meaning for FelderSoloman's Index of Learning Styles. Index of Learning Styles. **Anais...** p.16, 2003.

ANEXO A – QUESTIONÁRIO ORIGINAL DO ILS

- 1) I understand something better after I**
 - (a) try it out.
 - (b) think it through.
- 2) I would rather be considered**
 - (a) realistic.
 - (b) innovative.
- 3) When I think about what I did yesterday, I am most likely to get**
 - (a) a picture.
 - (b) words.
- 4) I tend to**
 - (a) understand details of a subject but may be fuzzy about its overall structure.
 - (b) understand the overall structure but may be fuzzy about details.
- 5) When I am learning something new, it helps me to**
 - (a) talk about it.
 - (b) think about it.
- 6) If I were a teacher, I would rather teach a course**
 - (a) that deals with facts and real life situations.
 - (b) that deals with ideas and theories.
- 7) I prefer to get new information in**
 - (a) pictures, diagrams, graphs, or maps.
 - (b) written directions or verbal information.
- 8) Once I understand**
 - (a) all the parts, I understand the whole thing.
 - (b) the whole thing, I see how the parts fit.
- 9) In a study group working on difficult material, I am more likely to**
 - (a) jump in and contribute ideas.
 - (b) sit back and listen.
- 10) I find it easier**
 - (a) to learn facts.
 - (b) to learn concepts.
- 11) In a book with lots of pictures and charts, I am likely to**
 - (a) look over the pictures and charts carefully.

(b) focus on the written text.

12) When I solve math problems

(a) I usually work my way to the solutions one step at a time.

(b) I often just see the solutions but then have to struggle to figure out the steps to get to them.

13) In classes I have taken

(a) I have usually gotten to know many of the students.

(b) I have rarely gotten to know many of the students.

14) In reading nonfiction, I prefer

(a) something that teaches me new facts or tells me how to do something.

(b) something that gives me new ideas to think about.

15) I like teachers

(a) who put a lot of diagrams on the board.

(b) who spend a lot of time explaining.

16) When I'm analyzing a story or a novel

(a) I think of the incidents and try to put them together to figure out the themes.

(b) I just know what the themes are when I finish reading and then I have to go back and find the incidents that demonstrate them.

17) When I start a homework problem, I am more likely to

(a) start working on the solution immediately.

(b) try to fully understand the problem first.

18) I prefer the idea of

(a) certainty.

(b) theory.

19) I remember best

(a) what I see.

(b) what I hear.

20) It is more important to me that an instructor

(a) lay out the material in clear sequential steps.

(b) give me an overall picture and relate the material to other subjects.

21) I prefer to study

(a) in a study group.

(b) alone.

22) I am more likely to be considered

- (a) careful about the details of my work.
- (b) creative about how to do my work.

23) When I get directions to a new place, I prefer

- (a) a map.
- (b) written instructions.

24) I learn

- (a) at a fairly regular pace. If I study hard, I'll "get it".
- (b) in fits and starts. I'll be totally confused and then suddenly it all "clicks".

25) I would rather first

- (a) try things out.
- (b) think about how I'm going to do it.

26) When I am reading for enjoyment, I like writers to

- (a) clearly say what they mean.
- (b) say things in creative, interesting ways.

27) When I see a diagram or sketch in class, I am most likely to remember

- (a) the picture.
- (b) what the instructor said about it.

28) When considering a body of information, I am more likely to

- (a) focus on details and miss the big picture.
- (b) try to understand the big picture before getting into the details.

29) I more easily remember

- (a) something I have done.
- (b) something I have thought a lot about.

30) When I have to perform a task, I prefer to

- (a) master one way of doing it.
- (b) come up with new ways of doing it.

31) When someone is showing me data, I prefer

- (a) charts or graphs.
- (b) text summarizing the results.

32) When writing a paper, I am more likely to

- (a) work on (think about or write) the beginning of the paper and progress forward.
- (b) work on (think about or write) different parts of the paper and then order them.

33) When I have to work on a group project, I first want to

- (a) have "group brainstorming" where everyone contributes ideas.

(b) brainstorm individually and then come together as a group to compare ideas.

34) I consider it higher praise to call someone

(a) sensible.

(b) imaginative.

35) When I meet people at a party, I am more likely to remember

(a) what they looked like.

(b) what they said about themselves.

36) When I am learning a new subject, I prefer to

(a) stay focused on that subject, learning as much about it as I can.

(b) try to make connections between that subject and related subjects.

37) I am more likely to be considered

(a) outgoing.

(b) reserved.

38) I prefer courses that emphasize

(a) concrete material (facts, data).

(b) abstract material (concepts, theories).

39) For entertainment, I would rather

(a) watch television.

(b) read a book.

40) Some teachers start their lectures with an outline of what they will cover.

Such outlines are

(a) somewhat helpful to me.

(b) very helpful to me.

41) The idea of doing homework in groups, with one grade for the entire group,

(a) appeals to me.

(b) does not appeal to me.

42) When I am doing long calculations,

(a) I tend to repeat all my steps and check my work carefully.

(b) I find checking my work tiresome and have to force myself to do it.

43) I tend to picture places I have been

(a) easily and fairly accurately.

(b) with difficulty and without much detail.

44) When solving problems in a group, I would be more likely to

(a) think of the steps in the solution process.

(b) think of possible consequences or applications of the solution in a wide range of areas.