



HAL
open science

Uma Arquitetura de Agentes para Recomendação Contextualizada de Eventos Baseada em Propagação da Ativação

Régia Ana Mendonça Neves

► **To cite this version:**

Régia Ana Mendonça Neves. Uma Arquitetura de Agentes para Recomendação Contextualizada de Eventos Baseada em Propagação da Ativação. Computation and Language [cs.CL]. UNIVERSIDADE DE BRASÍLIA, 2013. Portuguese. NNT: . tel-01097148

HAL Id: tel-01097148

<https://hal.science/tel-01097148>

Submitted on 3 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tese de Doutorado

Uma Arquitetura de Agentes para Recomendação Contextualizada
de Eventos Baseada em Propagação da Ativação

Ana Régia de Mendonça Neves

Brasília, 11 junho de 2013

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

Tese de Doutorado

**Uma Arquitetura de Agentes para Recomendação Contextualizada
de Eventos Baseada em Propagação da Ativação**

Ana Régia de Mendonça Neves

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Doutor em Engenharia Elétrica*

Banca Examinadora

Prof ^a . Ph.D. Célia Ghedini Ralha, CIC/IE <i>Orientadora</i>	_____
Prof. D.Sc. Gilmar Silva Beserra, FGA <i>Examinador externo</i>	_____
Prof ^a . D.Sc. Marisa B. B. Medeiros, CIN/UFSC <i>Examinadora externa</i>	_____
Prof. Ph.D. Adolfo Bauchspiess, ENE/FT <i>Examinador interno</i>	_____
Prof. Ph.D. Ivan M. de T. Camargo, ENE/FT <i>Examinador interno</i>	_____

Dedicatória

Aos meus pais, Regina e Dávila, pelo apoio e incentivo.

Às minhas irmãs, Karina, Kelly, Karen e Joelma, por me darem força nesta longa caminhada.

Às minhas sobrinhas, Gabriela, Lorena, Letícia, Sarah e Ayana, pelos sorrisos que completam a minha vida.

Ana Régia de Mendonça Neves

Agradecimentos

Agradeço à minha mãe, Regina, e irmãs, Karina e Joelma, por acompanharem o meu árduo e longo caminho no doutorado, compartilhando todas as alegrias e tristezas que esta jornada trouxe, fazendo com que a distância geográfica seja apenas um detalhe. As palavras são poucas para expressar meu agradecimento.

Ao meu pai, Dávila, pelo apoio nos momentos necessários.

Agradeço aos meus amigos Heider Marconi & Família e Gilmar Beserra & Família pela amizade, incentivo, correções de artigos, por acreditarem em mim nos momentos que eu falhei e acima de tudo amenizarem um pouco a distância familiar da qual eu reclamo há cinco anos.

Aos meus amigos José Edil e Reinaldo Pimenta & Família pelo apoio, risadas, almoços e caronas valiosas porque depender do transporte público em Brasília é complicado.

Agradeço aos discentes Humphrey Fonseca e Danielli dos Santos por colaborarem com o desenvolvimento deste projeto de pesquisa.

Agradeço ao discente Álvaro Carvalho pela valiosa colaboração que foi fundamental para a finalização deste projeto.

Aos meus alunos, Felipe César e Elton Garcia, pelo interesse e participação no projeto.

Ao professor José Camargo por ceder gentilmente o espaço físico, onde pude realizar grande parte da minha pesquisa e conhecer os meus melhores amigos, por me apoiar em momentos de crise no doutorado e por sempre perguntar sobre a minha saúde física e mental, afinal de contas não somos meros fazedores de artigos.

Ao professor José Carlos Ralha pelas valiosas correções no LaTeX que sempre resolvia falhar quando estava perto de algum prazo de entrega.

Agradeço à minha orientadora, professora Célia Ghedini, por ter me aceitado para orientação, por me ajudar no momento crítico de definição de projeto após algumas trocas de orientador. Muito obrigada por ter aceitado este desafio!

Ana Régia de Mendonça Neves

RESUMO

As técnicas tradicionais de recomendação não consideram as preferências do usuário de acordo com o contexto no qual está inserido. Não obstante, os sistemas de recomendação que agregam elementos contextuais são geralmente baseados em descrições sintáticas ou em valores numéricos informados explicitamente pelo usuário, o que pode gerar recomendações fora do domínio desejado, além de limitar a transparência do sistema quanto a interação usuário-ambiente, um dos principais objetivos da Computação Pervasiva. Deste modo, para que o processo de recomendação seja personalizado e transparente ao usuário, faz-se necessária uma aplicação que filtre proativamente o conteúdo das informações conforme contexto corrente e às preferências do usuário. Neste sentido, o objetivo deste trabalho é propor uma arquitetura baseada no paradigma de agentes de software, que permita a extração incremental de padrões comportamentais semanticamente relacionados pela aplicação da Teoria de Propagação da Ativação da Memória, possibilitando a oferta de informações contextualizadas aos usuários em ambientes distintos, como em um campus universitário. A prova de conceito da arquitetura proposta foi realizada pelo desenvolvimento de uma aplicação denominada *eAgora?*, a qual foi utilizada em um cenário de recomendação de eventos acadêmicos e culturais no campus Darcy Ribeiro da Universidade de Brasília.

ABSTRACT

Traditional recommendation techniques don't consider user's preferences according to context. Nevertheless, the recommendation systems that add contextual elements are often based on syntactic descriptions and numerical values, which are explicitly informed by users. As a result, recommendation outside the desired domain can be generated. Also the Pervasive Computing major goal can be limited by the system transparency. Thus, for a transparent and customized recommendation process it is necessary an application that proactively filters the information content according to the current context and user's preferences. This research aims to propose an architecture based on the agents paradigm, that enables incremental extraction of semantically related behaviour patterns. Also, the use of Spreading Activation Theory of Memory enables that contextual information is offered in distinct environments, e.g., the university campus. The proposed architecture has been validated through the development of *eAgora?* application, which is used in academic and cultural events recommendation at the Darcy Ribeiro Campus of the Brasília University.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	JUSTIFICATIVA	3
1.3	OBJETIVOS	3
1.4	CONTRIBUIÇÕES	3
1.5	LIMITAÇÕES DO PROJETO	5
1.6	APRESENTAÇÃO DO MANUSCRITO	5
2	REVISÃO DA LITERATURA	6
2.1	METODOLOGIA	6
2.2	SENSIBILIDADE AO CONTEXTO	9
2.2.1	AQUISIÇÃO DO CONTEXTO	10
2.2.2	REPRESENTAÇÃO DO CONTEXTO	11
2.2.3	PROCESSAMENTO DA INFORMAÇÃO	12
2.2.4	ENTREGA DA INFORMAÇÃO	12
2.3	UNIVERSIDADE PERVASIVA OU <i>iCampus</i>	14
2.4	SISTEMAS DE RECOMENDAÇÃO	17
2.4.1	IDENTIFICAÇÃO DO USUÁRIO	19
2.4.2	COLETA DE DADOS	19
2.4.3	CONSTRUÇÃO DO PERFIL	19
2.4.4	COMPARAÇÃO	20
2.4.5	ENTREGA E APRESENTAÇÃO	21
2.4.6	AVALIAÇÃO	22
2.4.7	ESTRATÉGIA DE AJUSTE	22
2.4.8	RECOMENDAÇÃO SENSÍVEL AO CONTEXTO	22
2.5	TEORIA DA PROPAGAÇÃO DA ATIVAÇÃO DA MEMÓRIA	23
2.5.1	ALGORITMO DE PROPAGAÇÃO DA ATIVAÇÃO	24
2.5.2	SATM E INTELIGÊNCIA ARTIFICIAL	28
2.5.3	SATM EM SISTEMAS DE RECOMENDAÇÃO	30
2.6	SISTEMA MULTIAGENTE	30
2.6.1	AGENTES	31
2.6.2	PROJETO DE UM AGENTE	32
2.6.3	CARACTERÍSTICAS DE SMA	34

2.6.4	METODOLOGIAS	36
2.6.5	APLICAÇÕES DE SMA	39
2.7	ONTOLOGIA.....	39
2.8	TRABALHOS CORRELATOS.....	42
3	ARQUITETURA PROPOSTA	45
3.1	VISÃO GERAL	45
3.2	MODELO DE CONTEXTO	45
3.2.1	INFORMAÇÕES	50
3.2.2	PESSOA	52
3.2.3	CAMPUS	54
3.2.4	INTERVALO TEMPORAL	55
3.3	MODELO ARQUITETURAL	55
3.3.1	ESPECIFICAÇÃO DO SISTEMA.....	56
3.3.2	PROJETO ARQUITETURAL	58
3.3.3	PROJETO DE DESENVOLVIMENTO DETALHADO	58
3.4	RECOMENDAÇÃO POR PROPAGAÇÃO DA ATIVAÇÃO	60
3.5	INTEGRAÇÃO DA APLICAÇÃO <i>eAgora?</i>	62
3.5.1	APLICAÇÃO PRINCIPAL E <i>interface</i>	62
3.5.2	CONTROLE DE ACESSO.....	63
3.5.3	LOCALIZAÇÃO	63
3.5.4	AGREGAÇÃO E PROPAGAÇÃO DA ATIVAÇÃO	64
3.5.5	INTEGRAÇÃO.....	64
4	PROVA DE CONCEITO	68
4.1	CENÁRIOS	68
4.1.1	ONTOLOGIA DE APLICAÇÃO	69
4.1.2	RECOMENDAÇÃO	70
4.1.3	TESTES DO PROTÓTIPO.....	71
4.2	DISCUSSÃO DOS RESULTADOS	72
5	CONCLUSÕES	77
5.1	TRABALHOS FUTUROS	79
	REFERÊNCIAS BIBLIOGRÁFICAS	80
A	MODELO DO PROTOCOLO DE REVISÃO SISTEMÁTICA	92
B	FORMULÁRIO DE CONDUÇÃO DA REVISÃO SISTEMÁTICA	94
C	FORMULÁRIO DE SELEÇÃO DE ESTUDOS DA REVISÃO SISTEMÁTICA	96
D	FORMULÁRIO DE EXTRAÇÃO DE DADOS DA REVISÃO SISTEMÁTICA	97

E	ARTEFATOS DA METODOLOGIA PROMETHEUS	99
F	LISTA DE RELEVÂNCIA SEMÂNTICA PARA O CENÁRIO I	112
G	LISTA DE RELEVÂNCIA SEMÂNTICA PARA O CENÁRIO II	114
H	<i>Log</i> DO ALGORITMO DE PROPAGAÇÃO DA ATIVAÇÃO PARA O CENÁRIO I	118
I	<i>Log</i> DO ALGORITMO DE PROPAGAÇÃO DA ATIVAÇÃO PARA O CENÁRIO II	125

LISTA DE FIGURAS

2.1	Visão geral do conceito de <i>iCampus</i> (adaptado de Ng et al., 2010).....	15
2.2	Processo de recomendação personalizada (adaptado de Tuzhilin, 2009).....	18
2.3	Um exemplo de rede semântica (adaptado de Crestani, 1997).	24
2.4	Rede estruturada de um modelo de SA (Crestani, 1997).	24
2.5	Fluxo básico do algoritmo SA (adaptado de Crestani, 1997).	26
2.6	Exemplo de um pulso do algoritmo de propagação de ativação (a) e (b)	28
2.7	Pseudocódigo para o algoritmo SA.	29
2.8	Visão geral de um SMA (adaptado de Wooldridge, 2009).....	31
2.9	Agente percebendo e atuando em um ambiente (adaptado de Russell and Norvig, 2009).	32
2.10	Tipos de arquitetura (a) e (b) (adaptado de Wooldridge, 2009).....	34
2.11	Metodologias AOSE (Henderson-Sellers and Giorgini, 2005).	36
2.12	Fases da metodologia Prometheus (adaptado de Padgham and Winikoff, 2004).	37
2.13	Tipos de ontologias e seus relacionamentos (adaptado de Guarino, 1998).	40
2.14	Representação simplificada dos componentes básicos da ontologia (adaptado de Noy and McGuinness, 2001).....	41
3.1	Visão Geral do Modelo SeCoM (Bulcão Neto, 2006).	47
3.2	Visão Geral da OntoUnB.	49
3.3	Taxonomia de classes da OntoUnB.....	51
3.4	Taxonomia expandida da classe Informações.	52
3.5	Taxonomia expandida da classe Pessoa.	53
3.6	Taxonomia expandida da classe Campus.	54
3.7	Taxonomia expandida da classe Intervalo Temporal.	55
3.8	Diagrama de papéis do sistema (a), (b) e (c).....	58
3.9	Diagrama de visão geral do sistema (a), (b) e (c).	60
3.10	Arquitetura geral proposta baseada em agentes.	61
3.11	Fluxo básico para recomendação contextualizada e personalizada.	62
3.12	Modelo de classes para aplicação <i>eAgora?</i>	63
3.13	Modelo de classes para o módulo de localização externa.	64
3.14	Diagrama de sequência para localização externa.	65
3.15	Arquitetura geral de implementação.	66
3.16	Diagrama de comunicação geral.....	67

4.1	Ontologia de aplicação.....	70
4.2	Tela do protótipo de cadastro e publicação de eventos.....	71
4.3	Tela de <i>login</i> do protótipo <i>eAgora?</i>	72
4.4	Telas iniciais do protótipo <i>eAgora?</i> (a) e (b).....	73
4.5	Telas de recomendações do protótipo <i>eAgora?</i> (a) e (b).	74
4.6	Telas dos eventos próximos recomendados pelo <i>eAgora?</i> (a) e (b).....	75
4.7	Telas de apresentações dos eventos próximos recomendados pelo <i>eAgora?</i> (a) e (b)...	75
4.8	Parte dos conceitos ativados no cenário I.....	76
E.1	Diagrama de ligação de dados.....	101
E.2	Diagrama de relacionamento entre agentes.....	101
E.3	Diagrama de revisão do sistema.....	106
E.4	Diagrama de interação.....	107
E.5	Diagrama de revisão do agente.....	108
E.6	Diagrama de revisão da capacidade.....	109

LISTA DE TABELAS

2.1	Estratégias de busca nas bases selecionadas	8
2.2	Teste de relevância (adaptado de Pereira, 2006).....	8
3.1	Relação entre classes OntoUnB e modelo SeCoM.	49
3.2	Descrição das percepções e ações das entidades do sistema.....	59
4.1	Configurações do algoritmo de propagação da ativação.....	72
4.2	Estatísticas dos resultados sobre a ontologia de aplicação nos cenários I e II.....	73
C.1	Critérios de Inclusão e Exclusão	96

LISTA DE ABREVIATURAS

- ACM – Association for Computing Machinery
- AOSE – Agent Oriented Software Engineering
- CARS – Context-aware recommender systems
- CML – Conceptual Modelling Language
- DAML+OIL – DARPA Agent Markup Language
- EEES – Espaço Europeu de Ensino Superior
- FIPA ACL – Foundation for Intelligent Physical Agents - Agent Communication Language
- IBM – International Business Machines
- ICC – Instituto Central Ciências
- iCampus – Campus inteligente
- IEEE – Institute of Electrical and Electronics Engineers
- JADE – Java Agent DEvelopment
- KIF – Knowledge Interchange Format
- KQML – Knowledge Query and Manipulation Language
- MIT – Instituto de Tecnologia de Massachusetts
- OCML – Operational Conceptual Modelling Language
- OO – Orientação a Objeto
- OIL – Ontology Inference Layer
- OWL – Ontology WeB Language
- PDT – Prometheus Design Tool
- RDF – Resource Description Framework
- RFID – Radio-Frequency IDentification
- ReS – Revisão Sistemática
- RNAs – Redes Neurais Artificiais

RS - Rede semântica

RSSI – Receive Signal Strenght Indication

SA – Spreading activation

SATM – Teoria da Propagação da Ativação da Memória

SECOM – Secretaria de Comunicação

SeCoM – Semantic Context Model

SHOE – Simple HTML Ontology Extensions

SIGRA – Sistema de Gestão Acadêmica

SIPES – Sistema de Pessoal

SIPPOS – Sistema de Pos-Graduação

SMA – Sistema Multiagente

SOA – Service-Oriented Architecture

SR – Sistemas de Recomendação

TI – Tecnologia da informação

UML – Unified Modeling Language

UnB – Universidade de Brasília

WS – Web Service

W3C – World Wide Web Consortium

XML – eXtensible Markup Language

XOL – Ontology Exchange Language

Capítulo 1

Introdução

Este capítulo apresenta a principal motivação deste projeto. Em seguida, os objetivos são apresentados, a fim de satisfazer os propósitos do estudo que nortearam o desenvolvimento deste trabalho, e, por fim, a estrutura da tese é especificada.

1.1 Contextualização

A harmonização e evolução contínua das estruturas educativas superiores impulsionada, principalmente, pela incorporação e desenvolvimento tecnológico, fundamenta a criação de universidades inteligentes. Neste cenário, surge a Universidade Pervasiva (Tavangarian et al., 2009), também conhecida como Campus Inteligente (Ng et al., 2010), onde a instituição educacional é enriquecida por meio de uma infraestrutura digital disponível para todos, criando um campus acessível, integrado, que auxilia os usuários e suporta um novo tipo de processo educativo.

Um dos desafios para a construção da Universidade Pervasiva é a oferta de serviços e eventos de modo dinâmico, contextualizado, personalizado e automatizado. Considerando que em uma universidade existem diferentes tipos de informações a serem prestadas, como matrícula, horário de aula, palestras e cardápio do dia no restaurante universitário, podem ser citadas como questões básicas da vida estudantil: “Onde fica sala X?”, “O que está acontecendo no campus?” e “Quais os eventos acadêmicos e culturais que eu teria interesse em participar?”. Mas estas perguntas nem sempre são facilmente respondidas aos interessados durante todo o ciclo de vida acadêmico de forma automatizada, dinâmica, contextualizada e personalizada.

Neste caso, o problema não é a quantidade de dados, mas como a informação é disponibilizada, com uso de filtros personalizados por relevância de conteúdo, conforme o contexto corrente (Hussein and Neuhaus, 2010). Diante deste cenário, sistemas de recomendação sensível ao contexto destacam-se como uma solução adequada, porque geram recomendações contextualizadas para um indivíduo ou grupo considerando os interesses e necessidades de cada um (Adomavicius and Tuzhilin, 2005). Tais sistemas, diferem-se das abordagens tradicionais de recomendação por explorarem o contexto do usuário, melhorando a qualidade do processo de recomendação por meio de aspectos

de dinamicidade e personalização.

Porém, os sistemas de recomendação que agregam elementos contextuais, geralmente, são baseados em descrição sintática ou em valores numéricos informados explicitamente pelo usuário (Adomavicius and Tuzhilin, 2005), o que pode acarretar consequências indesejáveis, tais como:

- gerar recomendações fora do domínio desejado, já que a descrição sintática utiliza palavras-chave e não diferencia palavras sinônimas e homônimas; e
- limitar a transparência do sistema na interação usuário-ambiente, pois o sistema constantemente solicita ao usuário algum tipo de informação.

Conforme exposto, o objetivo deste trabalho é propor uma arquitetura de software, que permita a extração incremental de padrões comportamentais semanticamente relacionados e oferte informações contextualizadas e personalizadas aos usuários em um campus universitário. A idéia é facilitar o acesso à informação, auxiliando na interação do usuário com o ambiente acadêmico. É importante destacar que a recomendação não é apenas comunicar ao ator o que acontece no campus, e sim que as informações a serem apresentadas caracterizem o contexto analisado e sejam entregues em tempo hábil para a tomada de decisão.

Deste modo, a arquitetura utiliza o paradigma de agentes de software (Wooldridge, 2009) para especificar os atores e as interações sociais que ocorrem no ambiente pervasivo, além de suportar a natureza dinâmica e distribuída das aplicações sensíveis ao contexto. O paradigma de agentes é constituído de entidades que apresentam capacidade de percepção e ação, que permitem modularizar e abstrair a complexidade inerente das aplicações sensíveis ao contexto. Além de permitir a integração entre dispositivos e serviços, por meio de uma abordagem de cooperação entre os agentes do ambiente para desempenhar tarefas específicas ou globais (Qingfeng and Longqing, 2011; Ng et al., 2011).

Para identificar os padrões comportamentais, interesses ou necessidades é proposto o uso da Teoria de Propagação da Ativação da Memória (*Spreading Activation Theory of Memory* - SATM) (Collins and Loftus, 1975). SATM procura explicar como as informações são organizadas e recuperadas na memória humana, uma vez que os interesses podem mudar de acordo com o contexto corrente.

Em razão da heterogeneidade do ambiente acadêmico e para evitar recomendações ambíguas, é importante definir um vocabulário comum, que permita a interoperabilidade entre as entidades presentes. Este entendimento comum, pode ser alcançado com o uso de ontologias (Studer et al., 1998), as quais viabilizam a representação do conhecimento de modo semântico e estruturado, o que facilita a aplicação da SATM.

Com isso, espera-se que o processo de aprendizado e fornecimento de informações seja proativo, personalizado e transparente quanto à interação usuário-ambiente. Como prova de conceito da arquitetura proposta foi desenvolvida uma aplicação denominada *eAgora?*, a qual foi ilustrada neste trabalho com dois exemplos no campus Darcy Ribeiro da UnB, sendo um usuário calouro e outro veterano de um determinado curso de graduação.

1.2 Justificativa

A oferta de informações no ambiente universitário não é realizada de modo ideal, uma vez que é apresentada desorganizada, em excesso e não considera os interesses e necessidades específicos dos diversos atores envolvidos na comunidade, sejam estes estudantes, professores e/ou servidores técnicos administrativos. Acrescenta-se ainda, quando este processo é automatizado, a falta de transparência na interação usuário-sistema por solicitar explicitamente aos usuários descrições sintáticas ou valores numéricos para o conteúdo ofertado. Considerando o processo de recomendação sensível ao contexto em ambientes pervasivos, existe ainda a necessidade de fornecer serviços e informações que sejam relevantes de modo menos intrusivo para o usuário.

1.3 Objetivos

O objetivo geral deste trabalho é propor uma arquitetura baseada em agentes para a extração incremental de padrões comportamentais do usuário e oferta de informações contextualizadas, de modo que o processo de recomendação seja contextualizado, personalizado, proativo, transparente e dinâmico.

Os objetivos específicos são:

- proceder uma revisão sistemática na área de recomendação sensível ao contexto com foco na Universidade Pervasiva e Campus Inteligente;
- modelar o domínio – visa o desenvolvimento de um modelo de conhecimento que represente a semântica do ambiente sensível ao contexto da Universidade Pervasiva;
- modelar a aplicação – seleção dos conceitos e relações que estão envolvidos diretamente com a aplicação, pois, nem todos os conceitos presentes na ontologia de domínio precisam ser apresentados ao usuário;
- modelar os agentes – objetiva desenvolver um modelo de sociedade que represente os agentes, especificando papéis, comportamentos e relações, bem como as características do ambiente em que estão inseridos;
- desenvolver aplicação – visa a construção de uma aplicação que explore e valide a arquitetura proposta para recomendação personalizada de eventos.

Vale ressaltar que alguns destes objetivos específicos foram alcançados por outros discentes conforme apresentado na Seção 1.4.

1.4 Contribuições

As contribuições deste trabalho para o estado-da-arte de sistemas de recomendação sensíveis ao contexto são:

- definição de uma arquitetura distribuída e transparente ao usuário, baseada em métodos cognitivos para ambiente pervasivos;
- definição de uma arquitetura que permite a aprendizagem incremental e dinâmica de perfil e contexto corrente de usuários;
- propor uma solução viável para o problema de inicialização nos sistemas de recomendação (*cold-start*), já que a aplicação *eAgora?* consegue recomendar eventos acadêmicos e culturais para novos usuários; e
- definição de uma arquitetura que possa ser estendida para outros domínios além do cenário universitário.

Conforme apresentado nos trabalhos correlatos (Seção 2.8), não foi encontrado na literatura abordagem similar ao proposto, no sentido de utilizar os conceitos e técnicas de modo integrado na arquitetura de recomendação sensível ao contexto. Deste modo, as contribuições descritas tornam a abordagem adotada inovadora, tratando algumas limitações que existem, principalmente, pela combinação de diferentes conceitos e técnicas integradas na arquitetura proposta.

Além disso, alguns trabalhos foram realizados paralelamente considerando a proposta deste projeto:

- dissertação de mestrado do discente Humphrey Fonseca – onde foi definida e avaliada uma arquitetura baseada em agentes para a localização interna de usuários nas instalações da UnB por meio de redes neurais artificiais, que foram treinadas com dados de intensidade de sinal de três protocolos de redes sem fio: Bluetooth (IEEE 802.15.1), Zigbee (IEEE 802.15.4) e Wi-Fi (IEEE 802.11) (Fonseca, 2011);
- projeto de iniciação científica (PIBIC) da discente Danielli dos Santos – teve como resultado a construção da ontologia de domínio OntoUnB (Santos, 2012);
- monografia dos discentes Elton Garcia e Felipe César – que apresenta uma aplicação de geolocalização para usuários com dispositivos móveis para a localização externa e a interação entre os usuários no campus Darcy Ribeiro da UnB (Santana and César, 2013);
- monografia do discente Álvaro Fonseca – que apresentou um estudo sobre a viabilidade de aplicação do *eAgora?* (Carvalho, 2013).

Como resultados desta pesquisa foram publicados os seguintes artigos em veículos da área de Engenharia e Computação:

- Localização de usuários em ambiente interno utilizando abordagem de sistema multiagente (Neves et al., 2010a) – iSys - Revista Brasileira de Sistemas de Informação – Qualis B3 na Ciência da Computação e B5 nas Engenharias III e IV;

- *Prototype for indoor localization based on multi-agent system* (Neves et al., 2010b) – *Proceedings of the Third International Conference on Intelligent and Advanced Systems (ICIAS'10)* – Qualis B4 na Ciência da Computação;
- *A User Location Case Study Using Different Wireless Protocols* (Fonseca et al., 2011) – *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access (MobiWac'10)* – Qualis B3 na Ciência da Computação;
- *An Agent-Based Architecture for Personalized Recommendation Using Spreading Activation* (Neves et al., 2013b) – *Proceedings of the 8th International Conference on Information Processing, Management and Intelligent Information Technology (ICIPT'13)*;
- *Location Agent: A Study Using Different Wireless Protocols for Indoor Localization* (Neves et al., 2013c) – *International Journal of Wireless Communications and Mobile Computing* (Open access); e
- *Agent-based Architecture for Context-aware and Personalized Event Recommendation* (Neves et al., 2013a) – *Expert Systems With Applications* (no prelo) – Qualis A1 na Ciência da Computação e A2 nas Engenharias I, III e IV.

1.5 Limitações do projeto

Algumas limitações devem ser consideradas na implantação deste projeto e dependem da infraestrutura tecnológica de cada universidade, tais como:

- a definição dos parâmetros do algoritmo e do tamanho da rede semântica para aplicar a SATM;
- a instanciação e manutenção da ontologia de mais de trinta mil alunos, considerando ainda professores e servidores administrativos;
- a privacidade e segurança das informações de perfil cadastradas;
- a manutenção e gerência das ilhas de informações pelos diversos departamentos, com o objetivo de manter os cadastros, de eventos acadêmicos e culturais na universidade, atualizados.

1.6 Apresentação do manuscrito

No Capítulo 2 será apresentada a revisão teórica desta pesquisa, para a qual utilizou-se a técnica de revisão sistemática. No Capítulo 3 será descrito o modelo conceitual com foco na arquitetura proposta para a recomendação contextualizada, personalizada e proativa de eventos aos usuários na Universidade Pervasiva. No Capítulo 4 será apresentado o desenvolvimento da aplicação *eAgora?*, a qual serve como prova de conceito da arquitetura proposta; seguindo das conclusões e trabalhos futuros no Capítulo 5.

Capítulo 2

Revisão da Literatura

Este capítulo apresenta a metodologia de pesquisa que direcionou o desenvolvimento deste capítulo e os conceitos das áreas de pesquisa envolvidas, incluindo: Sensibilidade ao contexto, Universidade Pervasiva ou iCampus, Sistema Multiagente, Ontologia e Trabalhos correlatos.

2.1 Metodologia

A revisão teórica desta pesquisa foi baseada na técnica de Revisão Sistemática (ReS) como uma estratégia de pesquisa bem definida e documentada. A ReS auxilia a coleta e avaliação de estudos relevantes relacionados a um determinado tema, por meio de uma sequência bem definida de passos metodológicos, seguindo protocolos desenvolvidos com raciocínio a priori, o que diferencia de revisões comuns (Kitchenham, 2004).

A técnica de ReS divide-se em três passos metodológicos, a saber:

1. planejamento – é definido o problema a ser pesquisado, os critérios de seleção de estudos e o protocolo de revisão sistemática utilizado para o planejamento da revisão bibliográfica;
2. condução – é feita uma análise da pesquisa bibliográfica a partir dos critérios estabelecidos no protocolo, incluindo ou excluindo estudos e extraíndo informações detalhadas desta pesquisa;
e
3. análise de resultados – são documentados os resultados e conclusões obtidas da ReS.

O protocolo de ReS é usado para o planejamento (1) da revisão bibliográfica e garante a execução com rigor de pesquisa. O protocolo utilizado nesta tese foi baseado nos conceitos propostos por Kitchenham (2004) e Biolchini et al. (2005):

- objetivos: (i) analisar, por meio de pesquisas bibliográficas, as bases do conceito de Universidade Pervasiva e *iCampus*, quais as universidades estão em processo de implementação

deste conceito, os problemas encontrados e quais técnicas estão sendo utilizadas para extração de padrões comportamentais e oferta de serviços personalizados; e (ii) avaliar os estudos experimentais realizados recentemente envolvendo recomendação de serviços personalizados e análise de padrões comportamentais aplicado em Universidade Pervasiva ou *iCampus*;

- questões de pesquisa: “Como aplicar os conceitos de Computação Pervasiva no setor educacional?”; “Quais universidades estão na direção da Universidade Pervasiva?”; “Como definir e modelar o contexto do ambiente, incluindo também o usuário e dispositivos?”; “Quais técnicas podem ser utilizadas para descobrir interesses e padrões comportamentais dos usuários?”; “Como prover informações ao usuário de acordo com o contexto?”; “Quais técnicas de aprendizagem de máquina suportam a modelagem semântica?” e “Quais as abordagens que permitem representar o indivíduo, especificando comportamentos e interações sociais?”;
- fontes utilizadas: foram selecionadas bases importantes e conhecidas da área de engenharia e computação disponíveis na web, a saber – *IEEE Xplore Digital Library*, *ACM Digital Library*, *Springer* e *Science Direct*;
- critérios de inclusão: para serem incluídos na fundamentação teórica os trabalhos deveriam – (i) conceituar extratores de padrões comportamentais para recomendação de informações; (ii) modelar o contexto baseado em ontologia; (iii) conceituar técnicas de recomendação e (iv) aplicar tais técnicas em Universidade Pervasiva ou *iCampus*;
- critérios de exclusão: foram descartados trabalhos que – (i) eram baseados no conceito de *smart campus*; (ii) não abordavam modelagem do ambiente sensível ao contexto por ontologia; e (iii) apresentavam técnicas de aprendizagem de máquina que não são baseadas em modelo semântico;
- palavras-chave: as palavras utilizadas para conduzir esta revisão foram – *Pervasive University*; *Intelligent Campus*; *Context-Aware computing*; *Ontology*, *Machine Learning*, *Context-Aware Recommender Systems*, *MultiAgent System*.

Após a finalização da fase de protocolo de revisão, é iniciada a fase de condução ou execução da ReS (2). Nessa fase são realizadas as pesquisas nas fontes listadas, utilizando-se as combinações de palavras-chave referentes a cada uma delas, a fim de encontrar trabalhos relevantes relacionados ao tema. Inicialmente, a condução foi feita no segundo semestre de 2010, durante os anos de 2011 e 2012. Adicionalmente, no primeiro semestre de 2013 foi realizada mais uma pesquisa para atualização da base. A Tabela 2.1 exibe alguns exemplos de estratégias de busca efetuadas nas bases selecionadas.

Para documentação das fontes foram preenchidos Formulários de Condução, que registram os trabalhos encontrados. Foram analisados os títulos e os resumos para saber quais seriam incluídos ou descartados. Essa escolha é feita com base nos critérios de inclusão e exclusão citados anteriormente. Além disso, foi utilizado um teste de relevância (Pereira, 2006) aplicado às referências e resumos dos artigos. Este teste é composto por uma lista de perguntas claras e objetivas que

Tabela 2.1: Estratégias de busca nas bases selecionadas

Base de dados	Estratégias de busca	Número de artigos
IEEE	“Pervasive University” OR “Intelligent campus”	19
	“Context-Aware Recommender Systems” AND “Intelligent campus”	186
ACM	“Context-Aware computing” AND “University” OR “Campus”	296
	“Context-Aware Recommender Systems”	203
	“Context-Aware Recommender Systems” AND “Machine Learning”	95
Springer	“Intelligent campus” AND “MultiAgent System”	426
	“MultiAgent System” AND “Context-Aware Recommender Systems”	53
Science Direct	“Context-Aware computing” AND “Ontology”	84
	“Context-Aware computing” AND “Machine Learning”	58

geram respostas afirmativas ou negativas, as quais refinam os artigos que serão acessados na íntegra. A Tabela 2.2 apresenta os questionamentos feitos durante a pesquisa para seleção dos artigos recuperados na fase anterior.

Tabela 2.2: Teste de relevância (adaptado de Pereira, 2006).

Referência de estudo:		
Questões	Sim	Não
1. O estudo está de acordo com o tema investigado?		
2. O estudo foi publicado dentro do período definido no projeto?		
3. O estudo foi publicado no idioma definido no projeto?		
4. O estudo aborda a solução do problema que está sendo investigado?		
O estudo foi conclusivo?		

Os artigos selecionados no teste de relevância foram acessados na íntegra para análise. Os Apêndices A, B, C e D contêm os formulários que auxiliam o desenvolvimento da ReS, conforme Biolchini et al. (2005). Esses formulários auxiliaram a última fase da ReS, análise de resultados (3), por terem dados extraídos dos trabalhos de um modo mais detalhado. O resultado final da ReS incluiu o estudo de:

- cinquenta e um artigos em periódicos;
- cinquenta artigos em conferências internacionais;
- seis teses de doutorado;
- treze livros;
- oito relatórios técnicos; e
- sete trabalhos não-publicados.

Estas referências serviram como base para o desenvolvimento deste capítulo e serão descritas nas próximas seções.

2.2 Sensibilidade ao contexto

O crescimento do poder computacional, a ampliação do uso de dispositivos móveis aliado à expansão das redes sem fio forneceram a infraestrutura necessária para o desenvolvimento da Computação Pervasiva (Ye et al., 2012). Considerada por muitos como o novo paradigma computacional do século XXI (Saha and Mukherjee, 2003; Carvalho et al., 2006), compreende a computação e comunicação a qualquer hora, lugar e em qualquer dispositivo, exigindo a intervenção mínima do usuário e adequando a tecnologia ao cotidiano. Indica uma tendência crescente para dispositivos móveis ou embutidos no ambiente, dotados de poder computacional, totalmente integrados e conectados a uma infraestrutura de rede, com atuação imperceptível e uso tão natural que será invisível aos usuários. Neste mundo, a computação está em toda a parte provendo serviços, informações e entretenimento.

Em 1998, com cunho mercadológico, a *International Business Machines* (IBM) utilizou este termo, Computação Pervasiva, para denominar um mundo pós-computador pessoal¹, visando principalmente os dispositivos e componentes, como semicondutores que utilizam pouca energia elétrica.

Ambientes pervasivos são considerados: (i) intrinsecamente dinâmicos, em virtude da mobilidade do usuário; e (ii) heterogêneos, em razão da variedade de dispositivos presentes, o que define vários cenários de atuação. Assim, para suportar a percepção do ambiente, as aplicações devem ser sensíveis ao contexto (Satyanarayanan, 2001).

O termo sensibilidade ao contexto foi utilizado pela primeira vez por Schilit and Theimer (1994) e, segundo Malik et al. (2007), refere-se à habilidade de um dispositivo ou aplicação perceber e se adaptar à situação corrente do usuário. Também pode envolver a entrega de informações e serviços mais relevantes e voltados ao usuário, diferenciando-se das aplicações tradicionais que apresentam características estáticas e baseadas em *desktops*.

Diferentes definições são encontradas na literatura para o termo contexto. Elas podem englobar a referência de contexto com exemplos ou subgrupos, ou seja, qualquer atributo que descreva o usuário, o dispositivo, a rede física ou o ambiente, podendo ser o nível de bateria ou a temperatura do ambiente (Schilit and Theimer, 1994; Olsen, 2007). Uma definição operacional de contexto é encontrada em Abowd et al. (1999) e Dey (2000) onde:

“contexto é qualquer informação que caracteriza a situação de uma entidade, que pode ser pessoa, lugar ou objeto, considerada relevante em uma interação usuário-aplicação, incluindo o próprio usuário e a aplicação.”

Destas definições, compreende-se que a sensibilidade ao contexto é a capacidade da aplicação ser ciente ao que está em torno, atuando no ambiente conforme o contexto se modifica no tempo.

¹<http://www.nytimes.com/1998/12/28/business/technology-computing-s-next-wave-nearly-hand-imagining-the-future-post-pc-world.html?pagewanted=all&src=pm>

Também, observa-se que a caracterização da situação é dependente do domínio e a relevância está relacionada com o que o usuário está fazendo em um momento específico, considerando o objetivo desta atividade.

Aplicações pioneiras, como *Active Badge Location System* (Want et al., 1992) e *Active Maps* (Schilit and Theimer, 1994), utilizavam mecanismos emissores de sinais que determinavam a localização atual de usuários para prover serviços, como encaminhar as ligações telefônicas para o telefone mais próximo. Porém, há muitas outras informações que podem ser utilizadas além de localização e identificação de pessoas e objetos. Para auxiliar esta escolha, Abowd et al. (1999); Abowd and Mynatt (2000) e Truong et al. (2001) definiram um conjunto com seis dimensões semânticas para especificação mínima do contexto, conhecido como (5Ws + 1H):

- quem? (*who?*) – identificar o usuário que está utilizando a aplicação;
- onde? (*where?*) – localizar o usuário;
- quando? (*when?*) – o momento que a interação está ocorrendo;
- o quê? (*what?*) – a atividade corrente do usuário;
- por quê? (*why?*) – entender as razões das ações do usuário; e
- como? (*how?*) – de que maneira as informações contextuais serão capturadas.

É possível definir um fluxo básico de processamento de contexto (Loke, 2006; Baldauf et al., 2007; Hoh et al., 2008), o qual é similar ao processamento de informação de modo geral, a saber:

1. aquisição do contexto;
2. representação do contexto;
3. processamento da informação; e
4. entrega da informação.

As próximas subseções descrevem cada etapa envolvida no fluxo básico de processamento de contexto apresentado.

2.2.1 Aquisição do contexto

A aquisição dos dados e/ou informações de contexto está relacionada ao monitoramento e obtenção de informações contextuais de fontes heterogêneas pela aplicação. Essas fontes podem estar associadas ao ambiente físico por meio de:

- sensores físicos, lógicos ou virtuais que captam, por exemplo, a localização, temperatura e atividades;

- informadas pelo próprio usuário, como preferências; e
- inferidas de outras informações pela aplicação.

Henricksen and Indulska (2006) propuseram uma classificação de tipos de aquisição de contexto, a saber:

- percebida – provenientes de sensores físicos, lógicos ou virtuais;
- estáticas – adquiridas de fontes persistentes de dados, como bibliotecas digitais;
- providas – fornecidas explicitamente pelo usuário à aplicação; e
- derivadas – inferidas por algum processo de raciocínio a partir de dados e/ou informações existentes.

Um possível cenário para aquisições derivadas é apresentado em Castro and Favela (2005), onde Redes Neurais Artificiais (RNAs) são utilizadas para o rastreamento contínuo de usuários em um ambiente interno a partir de valores da intensidade do sinal recebido (*Receive Signal Strength Indication* - RSSI) da infraestrutura de rede sem fio existente.

Um ponto importante a ser destacado no momento da aquisição é que quanto mais transparente for, menos intrusiva será a consulta de contexto e a aplicação estará na direção da Computação Pervasiva, a qual visa a transparência na interação com o usuário. Contudo, para essas informações terem um sentido, elas precisam ter um valor e um significado agregado, isto é alcançado com a representação do contexto.

2.2.2 Representação do contexto

De modo geral, em um ambiente sensível ao contexto entidades heterogêneas trocam informações ou dados constantemente. O ambiente deve traduzí-los em abstrações estabelecidas em um modelo de contexto. Este modelo formaliza as entidades, dados e informações que serão utilizados no ambiente, além dos relacionamentos entre as entidades.

Conforme Seção 2.1, uma questão de pesquisa definida neste trabalho é “Como definir e modelar o contexto do ambiente, incluindo também o usuário e dispositivos?”. Os resultados decorrentes da pesquisa para esta pergunta foram baseados nos estudos de Strang and Popien (2004), que descrevem e comparam algumas abordagens que podem ser utilizadas para representar o contexto, as quais são:

- pares de chave-valor – é o modelo mais simples para representar a informação contextual, onde um atributo caracteriza uma propriedade de contexto com um valor associado;
- linguagens de marcação – utiliza a linguagem de marcação *eXtensible Markup Language* (XML) para representar o contexto em uma estrutura de dados hierárquica composta de pares atributo-valor;

- gráficos – modela o contexto por meio de linguagens gráficas, como a *Unified Modeling Language* (UML), aproveitando a natureza genérica de representação, porém são limitadas quanto à modelagem do comportamento dinâmico do contexto;
- orientado a objetos – utiliza as características do paradigma orientado a objetos, como encapsulamento e reusabilidade, para lidar com os problemas da dinâmica do contexto;
- lógica – apresenta um alto grau de formalismo e representa o contexto por meio de fatos, expressões ou regras; e
- ontologias – representam as informações contextuais e seus inter-relacionamentos por meio de um vocabulário comum, o qual pode ser compartilhado, inferido e reutilizado.

Como exposto na Seção 1.1, motivado pela heterogeneidade do ambiente sensível ao contexto, este trabalho utiliza ontologias para modelagem de contexto, com o intuito de alcançar um vocabulário comum que permita a interoperabilidade semântica entre as diversas entidades presentes. Este conceito é apresentado na Seção 2.7.

2.2.3 Processamento da informação

Após a etapa de representação do contexto, ocorre a manipulação do contexto adquirido, o qual inclui mecanismos de inferência que auxiliam o processo de tomada de decisão nas aplicações. Por exemplo, Lech and Wienhofen (2005) propõem o sistema *AmbieAgents* orientado a agentes que utiliza raciocínio baseado em casos a fim de determinar a situação do usuário e recomendar informações relevantes. Em Chen and Lin (2012), RNAs são aplicadas para estimar a localização do usuário em um ambiente interno a partir do padrão ZigBee.

Geralmente, informações separadas podem não ser expressivas para uma aplicação, sendo a composição do contexto mais significativa. Como apresentado em Panu et al. (2003), a localização interna ou externa do usuário é inferida a partir de sensores que mensuram luminosidade, som, temperatura e umidade do ambiente, acrescido da atividade corrente do usuário e posição do dispositivo.

Algumas aplicações verificam e tratam inconsistências, já que, o contexto pode ser coletado por diferentes fontes distribuídas no ambiente. Dados ruidosos, rápida mudança de contexto e falhas de conexão são alguns exemplos que podem levar a um estado inconsistente. Em Gu et al. (2004), as informações de contexto são organizadas em diferentes níveis de confiança, deste modo, o contexto definido é classificado como mais confiável que o percebido e o deduzido. É importante ressaltar que o tratamento de inconsistência de contexto é um tema que, por si só, gera várias teses nas áreas de redes Ad Hoc e Computação Pervasiva, não sendo o tema de foco desta pesquisa.

2.2.4 Entrega da informação

Depois do processamento, três situações podem ocorrer: (i) a entrega imediata ao usuário; (ii) a adaptação do conteúdo; e (iii) armazenamento das informações percebidas ou derivadas.

A entrega do conteúdo pode passar por um processo de adaptação em resposta às mudanças de contexto do usuário, como o tipo de dispositivo ou o estado da conexão. Esta adaptação também pode ser direcionada para um usuário específico, denominada personalização (Endler et al., 2010). Chang and Hsu (2012) utilizam a tecnologia RFID (*Radio-Frequency IDentification*), estratégias de recomendação personalizadas e *Web Services* para desenvolver um ambiente de *shopping* sensível ao contexto, onde serviços personalizados são providos.

Algumas aplicações requerem o armazenamento das informações para criação de um histórico de contexto, o qual servirá para analisar padrões comportamentais do usuário e estabelecer tendências. Hong et al. (2009) propuseram um *framework* baseado em agentes para oferta de serviços personalizados, onde as informações, como preferências do usuário, padrões e hábitos, são extraídas do histórico de contexto.

Desse modo, compreende-se que o contexto está relacionado com as variações de características de um ambiente no qual o usuário se encontra, incluindo também o próprio usuário e os dispositivos presentes. De modo geral, uma infraestrutura deve prover meios para coletar, armazenar, processar e disseminar informações de contexto obtidas a partir de fontes diversas. Outras questões também devem ser tratadas, como:

- segurança e privacidade, para troca de dados entre usuários e aplicações;
- distribuição das informações, especificando como os atributos de contexto serão distribuídos;
- comunicação entre as diversas entidades presentes no ambiente; e
- o gerenciamento das diferentes fontes de contexto e da qualidade dessas informações por causa da natureza dinâmica e heterogênea destes elementos.

Entende-se que a Computação Pervasiva beneficia-se da sensibilidade ao contexto para enriquecer a interação usuário-ambiente. Sob esse novo paradigma, as aplicações tendem a ser transparentes, proativas e a entregar serviços, informações e entretenimento relevantes ao usuário em tempo real. A Computação Pervasiva é aplicada em diferentes domínios, tais como:

- ambiente de vida assistida (Rodríguez and Favela, 2008; Tiberghien et al., 2012) – auxilia as atividades diárias de pessoas idosas, proporcionando um ambiente inteligente para melhorar a qualidade de vida e reduzindo os custos com a saúde;
- esportes (Baca et al., 2010) – monitora o desempenho dos atletas em treino ou competição, com o objetivo de melhorar o condicionamento do atleta;
- sistemas de comunicação – para prover uma comunidade social virtual (Ng et al., 2011); e
- educação (Tavangarian et al., 2009; Nelaturu et al., 2010) – a instituição educacional é enriquecida por meio de uma infraestrutura digital disponível para todos e com aplicações que sustentem essa relação, criando um campus acessível e integrado, principalmente, ao suporte pessoal e no processo educativo.

Dentre as áreas citadas, o domínio de interesse deste trabalho é o educacional, especificamente o ambiente universitário, o qual é descrito na Seção 2.3 e refere-se à questão de pesquisa definida na Seção 2.1: “Como aplicar os conceitos de Computação Pervasiva no setor educacional?”.

2.3 Universidade Pervasiva ou *iCampus*

Em 1999 foi assinado na Europa o Processo de Bolonha² que define um conjunto de etapas para a reforma, integração e harmonização do ensino superior europeu nos quarenta e seis países signatários. Compreende mudanças curriculares, de metodologias de ensino/aprendizagem, cooperação internacional e aumento da mobilidade de docentes e discentes. O que levou a formação do Espaço Europeu do Ensino Superior (EEES) em 2010.

Para controle e análise das mudanças ocorridas, o Artigo 66.-A do Decreto -Lei nº 107/2008 exige que os estabelecimentos de ensino superior elaborem um relatório anual, público, acerca do progresso na concretização dos objetivos do processo. Alguns relatórios foram analisados³ e verificou-se que, entre outros pontos importantes, existe a necessidade de modernização da infraestrutura tecnológica das universidades como atrativo para a mobilidade dos discentes e docentes.

A Computação Pervasiva, citada na Seção 2.2, quando aplicada ao domínio educacional, permite a modernização da estrutura tradicional de ensino e a adequação a um novo tipo de comportamento, onde os dispositivos móveis estão ativamente inseridos no cotidiano do usuário. Neste sentido, algumas definições foram apresentadas para a evolução da universidade tradicional impulsionada pelo uso da tecnologia (Zender and Tavangarian, 2009; Nelaturu et al., 2010; Bromuri et al., 2010; Mircea and Andreescu, 2012). Duas representativas definições são:

- “A Universidade Pervasiva é uma instituição educacional enriquecida por mecanismos e artefatos da Computação Pervasiva de modo segmentado. Apresenta suporte integrado de tecnologia da informação (TI) em todos os campos de atividade, por exemplo: *eAprendizado/eEnsino*, *ePesquisa* e *eAdministração*. Do ponto de vista técnico, é um ambiente pervasivo, onde os componentes e padrões de interação são adaptados para as características de uma universidade” (Tavangarian et al., 2009).
- “Campus inteligente (*iCampus*) é a evolução de sistemas que têm apenas a capacidade de adaptação às mudanças ocorridas no ambiente, para os que conseguem pensar, raciocinar, entender, aprender e adaptar aos diferentes contextos em uma universidade. É constituído de seis áreas principais: (i) aprendizado; (ii) gerenciamento; (iii) governança; (iv) social; (v) saúde; e (vi) sustentabilidade” (Ng et al., 2010).

Para complementar a definição especificada por Tavangarian et al. (2009), o autor considera que Computação Pervasiva é um alto grau de TI em um ambiente, com suporte inteligente para as necessidades dos usuários por meio de serviços proativos e sensíveis ao contexto.

²http://ec.europa.eu/education/higher-education/bologna_en.htm

³ http://www.uc.pt/governo/reitoria/observatorios/rel_bolonha/
http://portal.ipv.pt/portal/page/portal/ese/ese_escola/ese_rcpb

O termo *iCampus* foi inicialmente explorado pelo Instituto de Tecnologia de Massachusetts (MIT) em colaboração com a *Microsoft Research*, teve como objetivo criar uma melhor infraestrutura para a universidade e favorecer a educação por meio da TI⁴. A Figura 2.1 apresenta as seis áreas definidas por Ng et al. (2010).

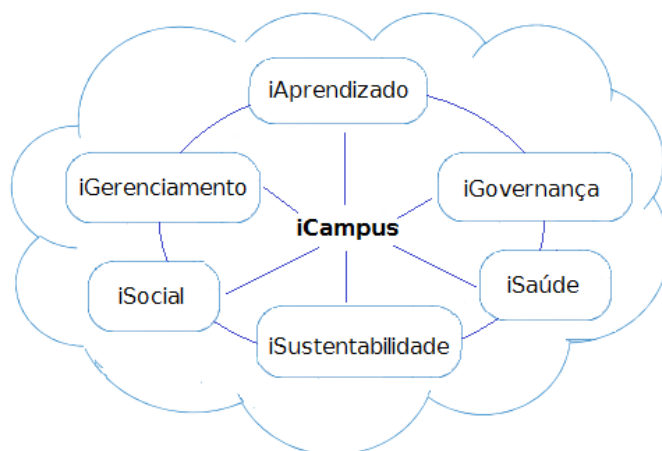


Figura 2.1: Visão geral do conceito de *iCampus* (adaptado de Ng et al., 2010).

O *i* na frente de cada área significa “inteligente”, com diversos graus e formas de inteligência, e “integrado”, onde a comunicação ocorre entre e dentro das áreas. São responsáveis por:

- aprendizado – alterações que ocorrem no modo de ensino, como professores e alunos não precisarem mais estar presentes no mesmo lugar e hora, acesso remoto, palestras a distância e cursos personalizados;
- gerenciamento – aspectos de instalações e infraestrutura nos prédios do campus, além do corpo de funcionários, estudantes e visitantes;
- governança – considera os recursos sociais e econômicos, a capacidade dos governos de planejar, formular e implementar políticas e cumprir funções;
- social – envolve atividades curriculares, extracurriculares e as atividades sociais em geral que ocorrem dentro do campus;
- saúde – garantir, rastrear e manter a saúde não só do campus, mas dos funcionários e estudantes. Esta ação não pode ser muito intrusiva, ou seja, a interação com a tecnologia deve ser transparente ou exigir a mínima atenção do usuário (*mobile health*); e
- sustentabilidade – aspectos sustentáveis do campus, como o uso de energia renovável.

Observa-se que algumas importantes características surgem das duas definições apresentadas, que são:

⁴<http://icampus.mit.edu/>

- modernização da universidade, por meio da tecnologia, a fim de adequá-la ao novo contexto emergente causado pelo avanço das tecnologias de redes sem fio e dispositivos móveis;
- um modelo integrado de universidade, onde todos os setores da universidade são colaborativos e ocorre comunicação entre estudantes, professores e gestão;
- suporte de todo o ciclo de vida estudantil na universidade de modo transparente e personalizado;
- recomendação personalizada de serviços e eventos que ocorrem no campus;
- o valor adicionado à universidade vai além do uso de dispositivos móveis e redes sem fio, incorporando também sensibilidade ao contexto, transparência e proatividade no ambiente;
- o alcance da universidade é estendido sem a expansão de instalações físicas; e
- enriquecimento da aprendizagem, ensino e pesquisa ultrapassando fronteiras.

Entende-se que, neste cenário, a universidade é definida como sendo:

- dinâmica – pela mobilidade dos indivíduos o que determina diversos contextos;
- heterogênea – considerando os diferentes dispositivos presentes, interações que podem ocorrer e serviços disponíveis;
- integrada – os setores da universidade estão integrados, como educacional, administrativo ou social;
- inteligente – pois deve ser proativa e capaz de perceber, aplicar mecanismos de raciocínio e realizar ações no ambiente; e
- contextualizada – pela sensibilidade ao contexto, onde um modelo define formalmente os conceitos relacionados ao cenário, possibilitando o entendimento das variáveis relevantes para a situação corrente, seguido do suporte ao usuário.

Outra questão de pesquisa definida na Seção 2.1 da ReS refere-se à: “Quais universidades estão na direção da Universidade Pervasiva?”. Neste sentido, alguns exemplos de aplicação são:

- MirRTLE (*Mixed Reality Teaching and Learning Environment*) (Callaghan et al., 2008; Dooley et al., 2011) da Universidade de Essex em Londres – projeto colaborativo para criar um ambiente de realidade integrada de ensino e aprendizado, onde professores e estudantes participam de uma aula online em tempo real. A interação ocorre por meio de avatares que representam professores e estudantes no sistema;
- ônibus *iCampus* (Mustapha et al., 2010) na Universidade de Kebangsaan na Malásia – um sistema é aplicado para monitorar o movimento do ônibus dentro da universidade;

- paciente virtual (Imison and Hughes, 2008) da Universidade Nova Gales do Sul na Austrália – pacientes virtuais permitem que estudantes de medicina desenvolvam habilidades de clínica e diagnóstico por meio de um cenário online;
- eCampus⁵ (Mitchell and Race, 2006) da Universidade de Lancaster – objetiva criar uma infraestrutura de comunicação pervasiva espalhada pelo campus;
- campus inteligente⁶ na Universidade Federal da Paraíba – representado por um campus virtual, onde são disponibilizados serviços voltados ao apoio de ensino, como ferramentas educativas, laboratórios remotos e de interesse social; e
- sistema de informação inteligente baseado em agentes IABUIS (*Intelligent Agent-Based University Information System*) (Cabukovski, 2011) implementado na Faculdade de Ciências Naturais e Matemática em Skopje na República de Macedônia.

Na Universidade Pervasiva, os serviços disponíveis devem ser ofertados de modo proativo, transparente e que seja mais próximo às preferências e situação corrente do usuário. Diante deste cenário, os Sistemas de Recomendação (SR) (Jannach et al., 2010; Lü et al., 2012; Deuk et al., 2012) destacam-se como uma possível solução para esse problema. A caracterização desse tipo de sistema é descrita na Seção 2.4.

2.4 Sistemas de recomendação

Ressalta-se na literatura que com a quantidade de conteúdo disponível na Internet e a facilidade de acessá-lo por meio de dispositivos móveis, muitas vezes o usuário tem dificuldade em encontrar algo de seu real interesse (Lorenzi, 2010; Davidsson and Moritz, 2011; Xu et al., 2012; Bouneffouf et al., 2012). Como exposto na Seção 1.1, este trabalho segue a visão de Hussein and Neuhaus (2010), onde afirmam que o problema não é a sobrecarga de informação, e sim encontrar um modo de filtrar este conteúdo para que seja relevante ao usuário na situação corrente em que ele se encontra. Uma das soluções é o uso de SR.

SR podem ser definidos como qualquer sistema que gere recomendações significativas de itens para um indivíduo ou grupo, de acordo com o interesse ou necessidade dos usuários (Jannach et al., 2010). Deste modo, o objetivo dos SR é auxiliar o usuário a encontrar facilmente o conteúdo que seja relevante para ele, quer sejam produtos, informações, serviços e pessoas. Neste trabalho, as palavras item e conteúdo são utilizadas para expressar este conjunto de opções que pode ser recomendado.

São aplicados em áreas como:

- comércio eletrônico (Dutta and Mukhopadhyay, 2011);
- turismo (Baltrunas, 2011);

⁵<http://ecampus.lancs.ac.uk/>

⁶<http://www.de.ufpb.br/labteve/projetos/campus.html>

- recomendação de filmes, como o provedor americano *Netflix*⁷, de livros na loja de compras online *Amazon*⁸ e de músicas, como encontrado no site *TasteKid*⁹; e
- educação (Sikka et al., 2012).

Os SR podem apresentar um recurso complementar conhecido como personalização. Refere-se à exploração do conhecimento sobre um indivíduo ou grupo mediante construção e análise de perfil, tem como objetivo adequar e recomendar o conteúdo disponível de acordo com as necessidades e preferências correntes. Por exemplo, em uma loja de livros o sistema pode guardar todos os livros vistos ou comprados pelo usuário para predição de interesses. Uma das definições encontradas na literatura é a de Choi and Han (2008), onde descrevem que “personalização é a integração das preferências e interesses do usuário no processo de entrega de conteúdo ou de um serviço”.

Neste trabalho, a arquitetura de um SR é especificada de acordo com a definição proposta por Adomavicius and Tuzhilin (2005) para recomendações personalizadas. A recomendação personalizada é apresentada como um processo iterativo caracterizando um ciclo de entendimento-entrega-medida (*Understand-Deliver-Measure*), conforme ilustrado na Figura 2.2, onde o ciclo inclui:

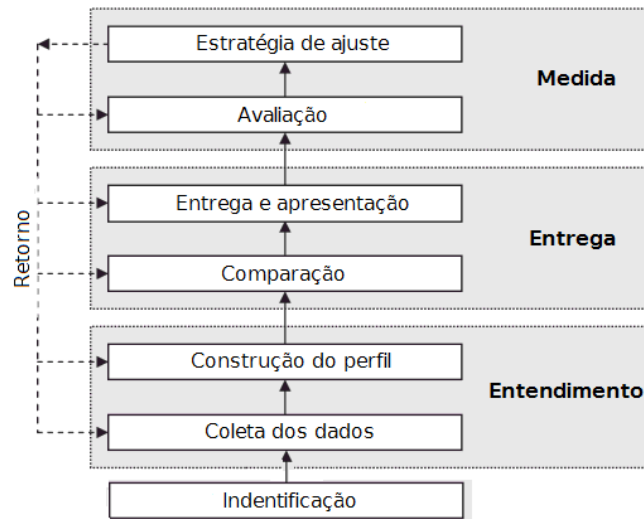


Figura 2.2: Processo de recomendação personalizada (adaptado de Tuzhilin, 2009).

- entendimento – coleta informações sobre o usuário e as armazena em um perfil;
- entrega – entrega de conteúdo personalizado baseado no perfil e nos itens disponíveis; e
- medida – mensura o impacto da personalização determinando quando o consumidor está satisfeito com o que lhe foi sugerido.

Ainda de acordo com a Figura 2.2, o ciclo contém seis etapas a partir da opção de identificação do usuário, a saber: (i) coleta de dados; (ii) construção do perfil; (iii) comparação; (iv) entrega

⁷<http://movies.netflix.com/WiHome>

⁸<http://www.amazon.com/>

⁹<http://www.tastekid.com/>

e apresentação; (v) avaliação do impacto; e (vi) estratégia de ajuste. Cada etapa é detalhada da Subseção 2.4.1 à 2.4.7.

2.4.1 Identificação do usuário

O processo de identificação é opcional e depende do sistema apresentar a capacidade de fazer recomendações personalizadas. Neste caso, o SR deve identificar o usuário para criar e manter uma base de dados que contenha informações pessoais sobre ele. A identificação e autenticação no sistema, geralmente ocorre por um *login* e senha.

Esta etapa é facultativa, pois depende do nível de personalização implementado pelo SR, podendo ser (Schafer, 2001):

- não recomendação – os mesmos itens são recomendados para todos os usuários, por exemplo, os produtos mais vendidos e os que estão em promoção;
- efêmera – é baseada na interação corrente do usuário com o sistema, sem armazenar qualquer tipo de informação anterior; e
- persistente – as recomendações são baseadas em um perfil construído do usuário.

2.4.2 Coleta de dados

Os SR trabalham com um conjunto de dados de entrada para recomendar o conteúdo. Estes dados são coletados de um indivíduo ou um grupo de modo explícito ou implícito (Schafer, 2001). Quanto mais informações o sistema coleta, o processo de recomendação se especializará de acordo com as preferências do usuário.

O modo explícito envolve a participação ativa do usuário mediante o preenchimento de questionários indicando preferências e dados pessoais para criação de um perfil. Também considera-se as avaliações quantitativas (escala numérica de pontuação) ou qualitativas (comentário sobre o item visitado ou adquirido) de itens. O processo implícito é menos intrusivo porque infere o padrão comportamental por meio da análise da interação com o sistema ou baseia-se em um histórico de ações. Tkalcivc et al. (2012) comparam o efeito de rotulagem afetiva implícita e explícita para um SR de imagem. O método implícito detecta as expressões faciais dos usuários e o explícito solicita ao usuário para determinar um rótulo afetivo para o conteúdo observado.

2.4.3 Construção do perfil

Os perfis são construídos de acordo com os dados coletados e são baseados no conhecimento ou comportamento do usuário. São construídos de modo explícito e implícito, como descrito anteriormente na etapa de *Coleta de dados*. Um exemplo de construção do perfil de modo explícito e interativo é encontrado em Rokach and Kisilevich (2012), onde por meio de um questionário o

usuário é solicitado a escolher o item de preferência entre dois apresentados. Assim, quanto mais respostas forem fornecidas, mais completo será o perfil.

2.4.4 Comparação

Após a construção do perfil, o sistema deve encontrar o conteúdo disponível mais relevante de acordo com as preferências descritas. Neste etapa, técnicas de recomendação são aplicadas. Segundo Burke (2002), um SR apresenta:

- dados do sistema – contemplam dados que o sistema contém antes do processo iniciar, como a relação de conteúdo que pode ser recomendado;
- dados de entrada – correspondem aos dados informados pelo usuário; e
- um algoritmo que combine estes dados para chegar a uma recomendação.

Quanto aos algoritmos de combinação, cinco diferentes técnicas podem ser identificadas (Burke, 2002; Jannach et al., 2010; Pagare and Shinde, 2012):

- filtragem colaborativa – considerada a técnica mais popular em SR (Resnick et al., 1994). Procura identificar indivíduos que possuam perfis comuns para formar grupos e recomendar itens baseado na opinião de cada grupo. Por exemplo, Chen and Chiang (2009) associaram esta técnica com ontologia para construção de um perfil semântico e oferta personalizada de documentos eletrônicos em um repositório. Alguns problemas deste tipo de filtragem devem ser considerados:
 - inicialização (*cold-start*) – (i) relacionado ao novo usuário no sistema, como ainda não existem avaliações realizadas por ele, o perfil de avaliações está vazio; e (ii) novos itens, um item deve ter avaliação mínima para poder ser recomendado;
 - esparsidade (*sparsity*) – geralmente, o número de avaliações já obtidas é muito pequeno quando comparado com as que ainda necessitam ser feitas; e
 - escalabilidade – relativo ao aumento de usuários e conteúdo disponível para que recomendações possam ser geradas com o mínimo de atraso.
- baseada em conteúdo – calcula a similaridade entre o perfil do usuário, que contém avaliações sobre itens comprados ou preferenciais, e itens disponíveis. Baseia-se no fato de que se o usuário gosta de um produto, provavelmente irá gostar de outros similares à ele. Lops et al. (2009) descrevem um SR, chamado de *FIRSt*, para a personalização do patrimônio cultural. O sistema *FIRSt* permite inferir os interesses dos usuários aplicando técnicas de aprendizagem de máquina. Além disso, emprega análise semântica para descobrir possíveis preferências em palavras-chave. Os problemas inerentes a esta abordagem são:
 - super especialização (*serendipity*) – o sistema recomenda apenas conteúdos que apresentam correspondência com o perfil do usuário;

- qualidade dependente do número de itens que constituem o perfil do usuário; e
 - limitações na representatividade do perfil.
- baseada em conhecimento – utiliza o conhecimento sobre os usuários e produtos para inferir preferências. Como descrito no trabalho de Lorenzi (2010), o qual apresenta a abordagem MATRES (*MultiAgent REcommenderS*). O sistema é baseado em suposições e confiança dos agentes para recomendação multiagente, onde é necessário conhecimento distribuído e específico. Neste caso, os agentes são colaborativos e geram recomendações de qualidade mesmo em situações críticas, como a falta de informação para gerar a recomendação. O cenário de aplicação é uma agência de viagem para oferta de pacotes turísticos, onde o conhecimento do domínio foi representado como casos (von Wangenheim and von Wangenheim, 2003). A principal limitação desta técnica é a aquisição do conhecimento, considerada o gargalo para esse tipo de sistema (Adomavicius and Tuzhilin, 2005);
 - filtragem demográfica – pode ser considerada uma extensão da filtragem colaborativa (Adomavicius and Tuzhilin, 2005) porque baseia-se no cálculo de similaridade entre usuários. Porém, utiliza dados demográficos, como idade, região e gênero, para construir o perfil, ao invés do histórico de avaliações. Em Chen and He (2009), um SR baseado em filtragem demográfica é descrito, onde a base de usuários é criada mediante os dados de idade, profissão e sexo; e
 - filtragem híbrida – agrega uma ou mais técnicas de recomendação para limitar os pontos fracos de cada abordagem. Um sistema pioneiro nesta abordagem é o FAB (Balabanović and Shoham, 1997), que recomenda páginas web utilizando a técnica colaborativa e baseada em conteúdo. Esmacili et al. (2012) também combinam estas técnicas para desenvolver uma abordagem híbrida de recomendação de comunidades virtuais e redes sociais considerando as preferências do usuário.

Estas técnicas podem ser classificadas em dois grandes grupos: (i) as que consistem nas características dos itens, como a baseada em conteúdo; e (ii) as que consideram os usuários, como as técnicas de filtragem colaborativa, demográfica e baseada em conhecimento. Para a especificação detalhada das técnicas descritas consultar Burke (2002); Jannach et al. (2010) e Pagare and Shinde (2012).

2.4.5 Entrega e apresentação

Etapa na qual é definido o momento mais apropriado e o meio a ser utilizado para a entrega, além de como o conteúdo será apresentado (Tuzhilin, 2009), sendo o menos intrusivo possível para o usuário.

Segundo Schafer (2001), a recomendação pode ser entregue via correio eletrônico ou mensagens de texto (*push technologies*), por solicitação do usuário (*pull technologies*) ou passivamente, quando conteúdos são recomendados em tempo real de acordo com o item corrente, por exemplo, “Quem se interessou por x, também se interessou por y”.

Os modos de visualização desta saída são:

- lista ordenada de acordo com critérios pré-estabelecidos, como itens mais vendidos ou uma categorização definida pelo sistema;
- lista não-ordenada para não influenciar a escolha do usuário ou possibilitar a perda de interesse;
- uma escala de classificação de acordo com a relevância do item e não por um critério estabelecido;
- um único conteúdo apenas, geralmente, são apresentadas como sugestões em frases do tipo: “Temos uma recomendação para você!” ou “Você vai gostar deste livro”;
- resenhas que expressam, textualmente ou por classificação, as opiniões dos usuários acerca de um item; e
- avaliações numéricas, onde os usuários estabelecem notas para o conteúdo.

2.4.6 Avaliação

Após a entrega, é aconselhável mensurar o impacto da recomendação a fim de melhorar este processo. Este cálculo é feito a partir do retorno (*feedback*) que o usuário fornece ao que foi sugerido, podendo ser por meio de avaliações explícitas, quantitativas e qualitativas, ou implícitas em observações comportamentais, como descrito na etapa de *Coleta de dados*. Em algumas abordagens, este processo serve para a construção de perfis (Samaras and Panayiotou, 2002). Geralmente, é possível distinguir dois tipos de retorno (Lops et al., 2011): (i) positivo, avalia características positivas feitas pelo usuário; e (ii) negativa, estima o que não foi considerado relevante. Em Jawaheer et al. (2010), é apresentado um estudo comparativo dos modos de retorno aplicado em um SR de músicas online. Oard and Kim (1998) descrevem os tipos de comportamento observáveis para o retorno implícito.

2.4.7 Estratégia de ajuste

Para finalizar o ciclo, após o cálculo do impacto efetuado, possíveis melhorias podem ser feitas nas etapas anteriores da recomendação. Caso o resultado das métricas não seja satisfatório, as causas do baixo desempenho devem ser identificadas e ajustadas de acordo com o retorno do usuário, como apresentado na Figura 2.2. Quando o retorno está integrado à recomendação, a qualidade do processo tende a melhorar com o tempo resultando em um “ciclo virtuoso de recomendação” (Tuzhilin, 2009).

2.4.8 Recomendação sensível ao contexto

Tradicionalmente, os SR tratam apenas das preferências do usuário, não considerando as mudanças de ambiente e da situação corrente, ou seja, o contexto; o qual influencia diretamente nos

interesses dele. Por exemplo, um indivíduo pode preferir um estilo de filme para ver sozinho e outro acompanhado. Neste caso, as preferências estão de acordo com o contexto corrente (Hong et al., 2009).

Esta necessidade intensificou-se com o acesso às tecnologias móveis, porque permitiu a disponibilização de informações a qualquer hora e lugar (Bouneffouf et al., 2012). Com isso, houve uma mudança de paradigma, onde informações de contexto são agregadas ao processo de recomendação surgindo os Sistemas de recomendação sensíveis ao contexto (*Context-aware recommender systems* - CARS) (Adomavicius and Tuzhilin, 2008; Asabere, 2013).

A noção de contexto foi descrita na Seção 2.2 e caracteriza-se como qualquer informação relevante do usuário e do ambiente para o sistema. Com isso, o processo de recomendação passa a considerar mudanças dinâmicas de contexto ocasionadas pela mobilidade do usuário. Neste sentido, são acrescentadas novas exigências, como (Arbanowski et al., 2004):

- manipulação de um conjunto extensível de preferências dos usuários;
- definição de múltiplos perfis associados às diferentes situações, já que as preferências variam com o contexto corrente;
- suporte para acessibilidade do conteúdo apresentado, caso necessário; e
- criação de diferentes perfis de rede e conteúdo, pois o tipo de conteúdo é dependente da situação.

A tendência em CARS é o uso de técnicas de aprendizagem de máquinas, como RNAs (Jiang, 2010), para permitir o acompanhamento das constantes mudanças de contexto que ocorrem no ambiente. Neste sentido, este projeto utiliza a Teoria de Propagação da Ativação da Memória (*Spreading Activation Theory of Memory* - SATM) (Collins and Loftus, 1975) para permitir o processo contínuo de adaptação, suportar a característica dinâmica do contexto e possibilitar o retorno implícito da recomendação feita. Esta teoria é descrita na Seção 2.5 e foi obtida como resposta para três questões de pesquisa definidas na Seção 2.1: (i) “Quais técnicas podem ser utilizadas para descobrir interesses e padrões comportamentais dos usuários?”; (ii) “Como prover informações ao usuário de acordo com o contexto e que permita o aprendizado dinâmico em diferentes situações?”; e (iii) “Quais técnicas de aprendizagem de máquina suportam a modelagem semântica?”;

2.5 Teoria da propagação da ativação da memória

A Teoria da Propagação da Ativação da Memória (SATM) procura explicar como funciona a recuperação de informações na memória humana. Propõe que a memória de longo prazo pode ser representada como uma rede estruturada de conceitos semânticos, onde os termos relacionados estão conectados, conforme ilustrado na Figura 2.3. Considera que quando um conceito é o foco da atenção, todos os conceitos a ele associados também são ativados (Anderson, 1983).

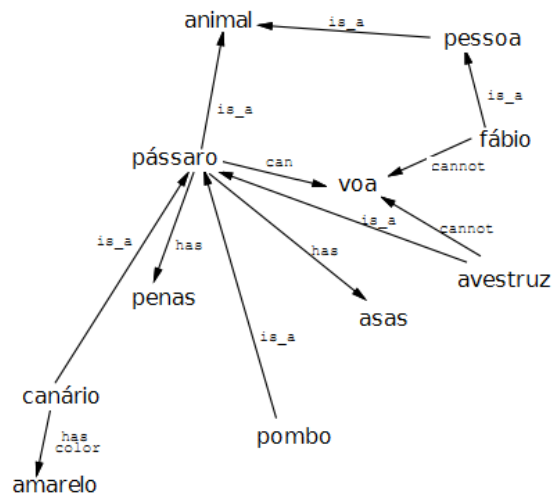


Figura 2.3: Um exemplo de rede semântica (adaptado de Crestani, 1997).

O conceito básico de SATM é que todas as informações importantes para um domínio são mapeadas em uma rede estruturada. Cada conceito na rede é representado por um nó i que está relacionado com outros nodos, por exemplo, o nó j . As relações podem ter um peso associado w_{ij} e indica a força ou grau da conexão, conforme apresentado na Figura 2.4. Quando um conceito é ativado, o respectivo nó do conceito também é, iniciando, assim, a propagação da ativação para os nós adjacentes que vai atenuando conforme se afasta do nó inicial ativado.

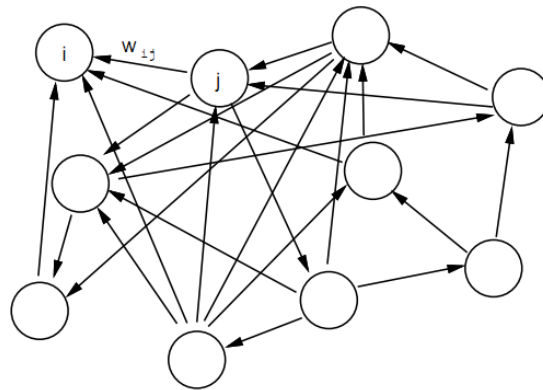


Figura 2.4: Rede estruturada de um modelo de SA (Crestani, 1997).

2.5.1 Algoritmo de propagação da ativação

O algoritmo *spreading activation* (SA) apresenta como estrutura uma rede semântica (RS)(NS, AS), onde:

- RS – rede semântica de tamanho finito e limitada pelo ambiente onde o algoritmo executa;
- NS – nós semânticos ou conjunto de nodos; e
- AS – arestas semânticas ou conjunto de arestas da RS.

Considerando a Figura 2.4, o algoritmo SA atua iterativamente sobre esta RS e apresenta oito elementos, os quais são:

1. nodo (i) – é um elemento do NS, ou seja, um nodo da RS. Representa um termo para o domínio e pode referenciar um conceito ou instância de conceito;
2. sementes (K) – o conjunto de nós que iniciarão a execução do algoritmo e a cada iteração é determinado um novo conjunto K a ser ativado;
3. peso da relação (w_{ij}) – uma aresta determina a associação semântica entre os nodos i e j . Esta relação possui um peso w_{ij} , que pode ser binário (0 ou 1) ou pertencer ao conjunto dos números reais e são dependentes do tipo da aplicação. Por exemplo, em uma aplicação com representação binária, caso não exista relação entre os nós i e j , então $w_{ij} = 0$;
4. valor de ativação (A_i) – refere-se ao valor atribuído ao nodo (i) em uma passagem do algoritmo por este nó, também podendo ser binário (0 ou 1) ou pertencer ao conjunto dos números reais. A cada pulso p todo nó i tem um nível de ativação A_i e uma saída O_i que é uma função do seu nível de ativação, conforme Equação 2.1 (Tsatsaronis et al., 2007).

$$O_i(p) = f(A_i(p)) \quad (2.1)$$

A saída do nó i afeta o próximo valor de ativação de qualquer nó j que está diretamente relacionado. Então, o valor de ativação de cada nó j da rede, no pulso p , é a soma da função de saída O_i do nó i no pulso $p - 1$ com o peso w_{ij} da relação, de acordo com a Equação 2.2 (Tsatsaronis et al., 2007).

$$A_j(p) = \sum_i O_i(p-1)w_{ij} \quad (2.2)$$

5. pulso (p) – corresponde ao ciclo de disparo do algoritmo SA, onde o nó i ou os nodos selecionados para ativação têm seus valores $A_i(p-1)$ atualizados para $A_i(p)$, considerando os pesos dos nodos adjacentes. A idéia básica de um pulso é avançar um passo na RS por meio das associações semânticas entre o nó i e seus adjacentes;
6. *threshold* – o ponto-chave no algoritmo é determinar quando um nó é ativado ou não, o que permite que o algoritmo espalhe a ativação sobre a rede (Kandfer and Shapiro, 2011). Uma das funções mais utilizadas é a limiar (Crestani, 1997), a qual determina se o nodo i deve ser considerado ativo e, portanto, propagar a ativação para outros nós adjacentes. Por exemplo, em um pulso p o nodo i tem como valor de ativação $A_i(p) = 0,79$, se *threshold* = 0,8, então o nodo i não será marcado para ativação e conseqüentemente não terá o seu valor de ativação atualizado;
7. fator de decaída (*decay factor*) (D) – este fator faz com que o valor de ativação A_j de um nodo diminua à medida que se afasta do nó inicial, caracterizando a diminuição da força

de ativação. Deste modo, define-se uma função h para penalizar a saída O_j de um nodo j , conforme Equação 2.3 (Rodríguez et al., 2013).

$$O_j = h(A_j) \quad (2.3)$$

8. relevância semântica – valor de ativação final de A_i de um nodo i , já que o algoritmo pode passar várias vezes pelo nodo i e atualizar A_i . Este valor de relevância semântica é retornado em uma lista junto a cada um dos nodos que tenham alcançado o valor de *threshold* aceito.

Uma vez apresentados os elementos do algoritmo SA, é descrito o fluxo de processamento clássico, de acordo com a Figura 2.5, onde cada iteração ou pulso compreende: (i) um ou mais pulsos; e (ii) condição de parada. Um pulso é composto por três fases:

- pré-ajuste – determina o valor dos pesos iniciais dos nós preparando a RS. Este passo determina o valor inicial dos nodos, podendo ser 1 para os nós pertencentes ao conjunto de sementes K e 0 para os demais;
- propagação – consiste na ação iterativa de difundir a ativação de um nó i para todos os nodos adjacentes, o qual pode ser interrompido por alguma condição de parada. A cada propagação, um conjunto de nodos é ativado e os seus valores de relevância semântica são ajustados. O cálculo do valor de ativação A_i do nó i é definido, como apresentado na Equação 2.2; e
- pós-ajuste – esta fase determina quais os nodos adjacentes de i que serão marcados para ativação no pulso seguinte.

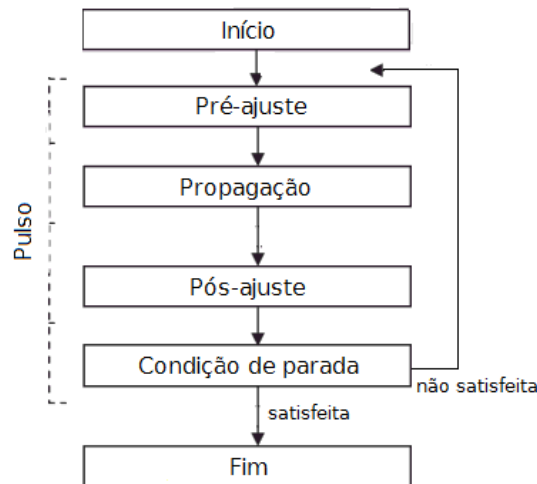


Figura 2.5: Fluxo básico do algoritmo SA (adaptado de Crestani, 1997).

As fases de pré e pós-ajuste são opcionais e utilizadas para evitar a retenção de pulsos anteriores, permitindo o controle da ativação dos nós individualmente e da rede (Crestani, 1997). Após a fase de pós-ajuste, a condição de parada é verificada, que pode ser:

- quando todos os nós são disparados;
- quando o número de iterações ou pulsos é alcançado; e
- a propagação alcança o mesmo nodo por arestas diferentes.

Deste modo, o algoritmo SA é iterativo, compreendendo uma sequência de pulsos e verificação de condição de parada. Pulso após pulso, a ativação se propaga pela rede atingindo os nodos que estão relacionados aos nós ativados inicialmente. Após a condição de parada ter sido alcançada, o algoritmo gera como resultado uma lista que contém os conceitos que foram ativados, ordenados pela medida de relevância semântica e representa os conceitos mais relacionados com o de interesse.

O algoritmo SA clássico apresenta algumas limitações, como a ativação ser propagada por toda a rede, por isso, restrições devem ser aplicadas, tais como (Crestani, 1997):

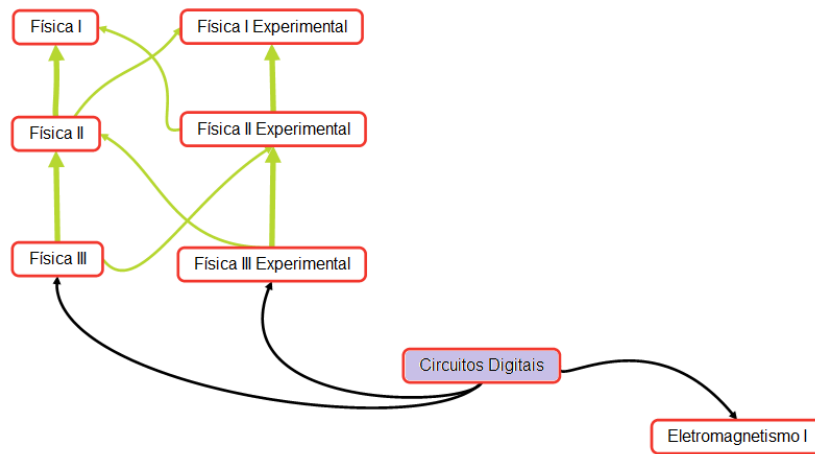
- de distância – determina a quantidade de níveis a serem ativados. Deve considerar relações de até terceira ordem;
- de propagação – a propagação deve parar ao encontrar nós com um grande número de associações, o que evita a ativação de grandes áreas. Considera que um termo com muitos relacionamentos é genérico e não deve ser expandido;
- de caminho – a difusão da ativação deve ser feita por caminhos mais significativos, de acordo com as regras de inferência da aplicação. Isso pode ser modelado por pesos nas arestas ou por rótulos; e
- de ativação – define um limiar para a ativação do nó, controlando a propagação na rede. Caso um nó não alcance este valor, o fluxo não prossegue neste nodo.

Estas estratégias de controle atuam na propagação e evitam que a difusão ocorra em toda a rede, diminuindo o esforço computacional.

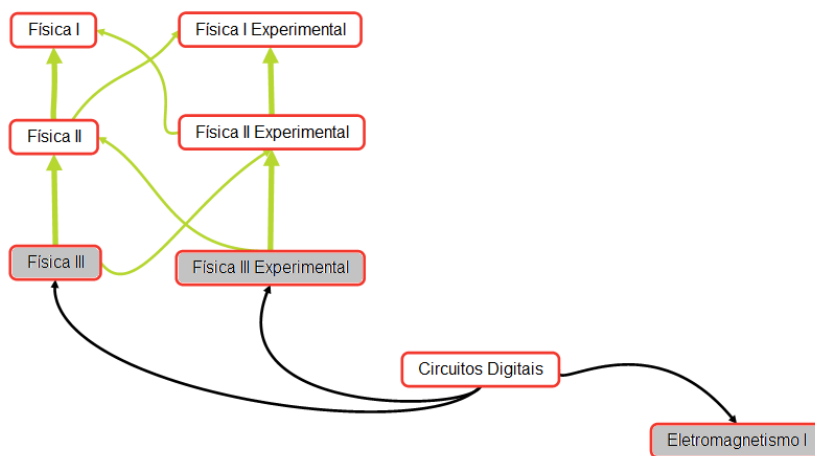
As Figuras 2.6(a) e 2.6(b) apresentam um trecho de uma rede semântica para os pré-requisitos da disciplina de Circuitos Digitais, ou seja, semente (K)=Circuitos Digitais. O pulso $p = 0$ indica que o algoritmo ainda não iniciou. Após o disparo do primeiro pulso ($p = 1$), o valor de ativação $A_{CircuitosDigitais}$ é atualizado e K é expandido para um conjunto de sementes K' , onde os conceitos são semanticamente relacionados na RS. Este conjunto K' contém os termos: Física III, Física III Experimental e Eletromagnetismo I. A Figura 2.6(b) ilustra os nós marcados para a propagação, repetindo assim o processo iterativo.

Um pseudocódigo para o algoritmo SA pode ser escrito definindo as fases de pré-ajuste e propagação, conforme ilustrado na Figura 2.7.

Candan and Sapino (2010) descrevem três algoritmos geralmente utilizados para implementar a SATM, que são: (i) rede de Hopfield; (ii) *branch-and-bound* (ramificação e corte); e (iii) *Constrained Leaky Capacitor Model*. Porém, todos seguem o mesmo processo: ativação do nó, propagação



(a) Rede semântica no pulso (p) = 0



(b) Rede semântica no pulso (p) = 1

Figura 2.6: Exemplo de um pulso do algoritmo de propagação de ativação (a) e (b)

do pulso pela rede e verificação da condição de parada. Este trabalho utiliza o *framework* OntoSpread¹⁰ desenvolvido em Java sob a licença GPL (GNU - General Public License). Ele foi escolhido porque fornece uma interface de programação de aplicativo que permite o desenvolvimento, configuração, personalização e execução do algoritmo original de SA, aplicando níveis de restrição, como a diminuição dos pesos de acordo com a distância do nó ativado. Além disso, permite o uso do algoritmo no campo da Web Semântica com as linguagens definidas pelo W3C, como RDF e OWL.

2.5.2 SATM e Inteligência Artificial

A SATM é mais análoga ao modelo conexionista da Inteligência Artificial (IA) (Rodriguez et al., 2010). Uma das técnicas da IA que pode se assemelhar à esta teoria é a Rede Neural Artificial (RNA), onde é definida como um processador massivamente paralelo distribuído, constituído de

¹⁰<https://code.google.com/p/ontospread/>

```

1. Início
2. nosSelecionadosParaAtivacao = {};
3. float thresholdDisparo, fatorDecaida;
4. para i ∈ K faça
    |
    |     /* Inicia o conjunto de sementes K */
5.     atribuirAtivacaoMaxima(i);
6.     marcarNoComoDisparado(i);
    |
    |     /* Prepara os termos adjacentes para disparo */
7.     Adjacentes ← retornaAdjacentesDe(i);
8.     marcarNosParaDisparo(Adjacentes);
9. fim
10. enquanto condição de término não for alcançada e nosMarcadosParaDisparo ≠ 0 faça
11.     para j ∈ nosMarcadosParaDisparo faça
12.         calcularAtivacao(j);
13.         dispararNo(j);
14.         nosMarcadosParaDisparo.remove(j);
15.         marcarComoDisparado(j);
15.     fim
16. fim

```

Figura 2.7: Pseudocódigo para o algoritmo SA.

unidades de processamento simples, que têm a propensão natural de armazenar conhecimento experimental e torná-lo disponível para uso (Haykin, 1998).

Porém, algumas diferenças podem ser destacadas, como:

- o algoritmo SA é aplicado em uma rede semântica, onde todos os nós da rede representam um conceito para um domínio;
- os nós da rede semântica estão conectados por relacionamento, ou seja, os termos que apresentam relação semântica estão conectados;
- a aplicação do algoritmo SA não é dividida em fase de treinamento e teste, como ocorre nas redes múltiplas camadas (Haykin, 1998); e
- os pesos da RNA são fixos após o treinamento, enquanto que os pesos na rede semântica são variáveis e indicam a relevância semântica dos termos. Deste modo, quando o usuário escolhe algum termo semântico indicado pelo algoritmo, o valor do peso da conexão é atualizado.

O algoritmo SA é para ser aplicado em redes semânticas e associativas. Segundo Teuffel et al. (2009), redes semânticas são consideradas como um modo de representação do conhecimento por notação gráfica, onde os nós armazenam conceitos e as arestas representam a relação entre esses conceitos; já as redes associativas são consideradas como sendo mais gerais do que as redes semânticas e as relações são ponderadas de acordo com as forças das associações.

2.5.3 SATM em sistemas de recomendação

Alguns trabalhos que aplicam SATM em sistemas de recomendação, como em Hussein et al. (2007); Trousov et al. (2009) e Alvarez et al. (2011), apresentam as seguintes características:

- aprendem iterativamente o comportamento do usuário de modo transparente, ou seja, sem solicitar o preenchimento de cadastros ou indexação de pontos;
- permite o acompanhamento dinâmico das preferências do usuário em diferentes contextos;
- ofertas personalizadas;
- uma interação mais natural com o sistema; e
- possibilita o retorno implícito para ajuste do processo de recomendação.

Hussein et al. (2007) ressaltam as vantagens de aplicar SATM em CARS, as quais são:

- os nós ativados estão de acordo com o contexto corrente do usuário;
- a rede construída é personalizada;
- possibilita a construção de um perfil dinâmico;
- facilita o aprendizado do sistema pelas preferências do usuário considerando o contexto; e
- permite o retorno implícito.

Como especificado na Seção 1.1, este trabalho baseia-se na proatividade e transparência da recomendação contextualizada, representando o indivíduo na sociedade e a própria sociedade, enfatizando o comportamento e a interação social, particularmente entre usuário e aplicação. Para alcançar este nível de representação, é proposto o uso de agentes (Wooldridge, 2009), já que outros paradigmas não conseguem representar esse tipo de modelo. Além disso, aplicações sensíveis ao contexto são intrinsecamente dinâmicas e distribuídas, por isso, precisam de uma abordagem que permita a definição de comportamentos autônomos e proativos para antecipar e adaptar às mudanças ocorridas no ambiente.

A próxima seção corresponde ao último questionamento definido na Seção 2.1: “Quais as abordagens que permitem representar o indivíduo, especificando comportamentos e interações sociais?”.

2.6 Sistema Multiagente

O paradigma de agentes de software fundamenta o estudo e definição de Sistema Multiagente (SMA). Uma definição intuitiva de SMA é um sistema que possui dois ou mais agentes inteligentes interagindo em um ambiente para satisfazer um conjunto de objetivos individuais ou coletivos. Para Wooldridge (2009), um SMA é composto de agentes autônomos que interagem e atuam segundo

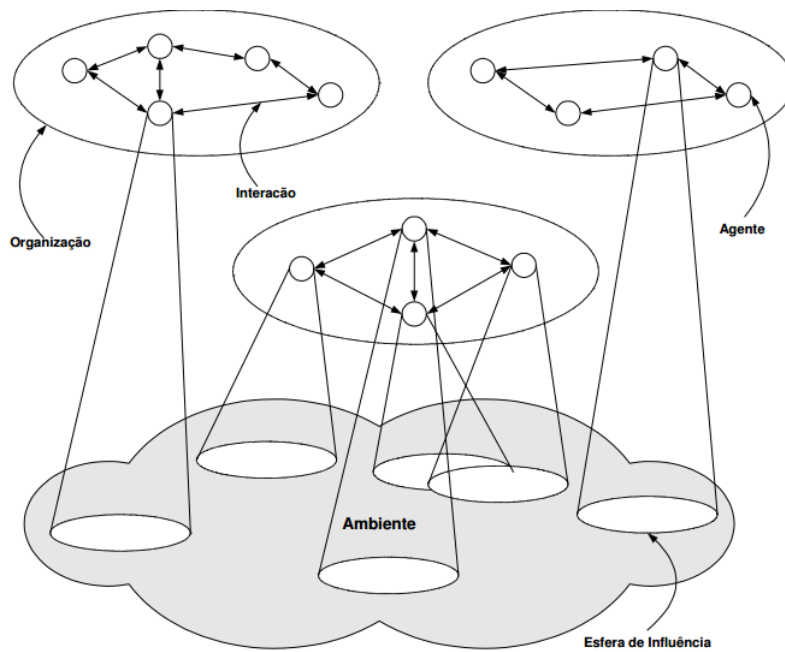


Figura 2.8: Visão geral de um SMA (adaptado de Wooldridge, 2009).

seus objetivos em um ambiente, apresentando diferentes esferas de influência e são ligados por relações organizacionais, como ilustrado na Figura 2.8.

Deste modo, cada agente em um SMA será composto de:

- diferentes capacidades de percepção e ação no ambiente;
- distintas esferas de influência, isto é, cada um influencia uma parte do ambiente, podendo ocorrer sobreposição; e
- várias organizações de acordo com as suas atividades e objetivos a serem alcançados.

2.6.1 Agentes

Um agente é uma entidade unitária de um SMA, é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e agir sobre ele por intermédio de atuadores (Russell and Norvig, 2009). A Figura 2.9 ilustra uma arquitetura de alto nível de um agente.

Um agente é considerado inteligente quando apresenta as seguintes características (Wooldridge, 2009):

- autonomia – exerce controle sobre suas ações;
- reatividade – respondem às mudanças que ocorrem no ambiente;
- proatividade – é capaz de ter iniciativa para satisfazer objetivos próprios; e
- sociabilidade – é capaz de comunicar-se, cooperar e raciocinar;

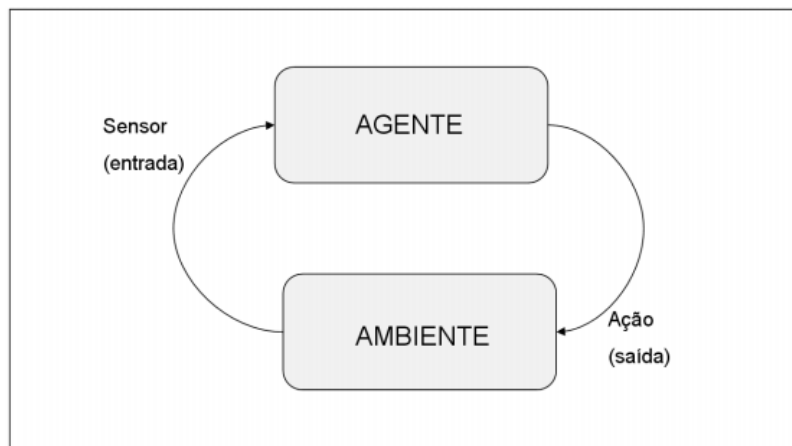


Figura 2.9: Agente percebendo e atuando em um ambiente (adaptado de Russell and Norvig, 2009).

Segundo Russell and Norvig (2009), um agente é classificado em quatro tipos básicos de acordo com o raciocínio empregado:

- agentes reativos simples – são considerados os agentes mais simples devido ao seu comportamento puramente reativo. Eles apenas escolhem uma ação baseada na sua percepção;
- agentes reativos baseados em modelos – mantêm um estado interno que depende de seu histórico de percepções e ações. São igualmente conhecidos como agentes reativos com estado e suas ações dependem não só da percepção, mas também do estado interno;
- agentes baseados em objetivos – mantêm o registro de estado interno e um conjunto de metas que tentam alcançar. As ações dos agentes são guiadas por objetivos, considerando também a análise do estado atual do ambiente; e
- agentes baseados em utilidades – além dos objetivos, utilizam uma função de utilidade para mensurar o grau de satisfação do agente a partir do estado no qual se encontra. A função de utilidade mapeia um estado ou uma sequência de estados em um número real, que descreve o grau de utilidade. O agente sempre escolherá a ação que leva a uma maior utilidade.

2.6.2 Projeto de um agente

Para desenvolver um projeto de agente inteligente deve-se especificar as percepções e ações possíveis de cada agente, os objetivos ou medidas de desempenho a serem alcançados, assim como o tipo de ambiente onde o agente irá atuar (Russell and Norvig, 2009). Logo, um agente inteligente pode ser caracterizado por percepções (*Percepts*), ações (*Actions*), objetivos (*Goals*) e ambiente (*Environment*) onde está inserido, também conhecido como PAGE.

2.6.2.1 Ambiente

Weyns and Holvoet (2004) definem que o ambiente neste paradigma é uma abstração de primeira classe que fornece as condições para os agentes existirem, além de mediar a interação entre eles e o acesso aos recursos. Por isso, definir suas propriedades é uma decisão de projeto que depende dos requisitos do domínio do problema e afeta diretamente a percepção do agente, além da complexidade do processo de concepção do agente. Segundo Russell and Norvig (2009), o ambiente pode ser caracterizado de acordo com os seguintes parâmetros:

- acessível vs. inacessível – determina se um agente consegue obter, por meio de sensores, informações completas e precisas sobre o ambiente. Caso isto ocorra, o ambiente é considerado acessível; do contrário, é considerado inacessível. Quanto mais acessível um ambiente for, mais simples será projetar e construir agentes para agirem eficazmente dentro dele;
- determinístico vs. não-determinístico – em um ambiente determinístico cada ação tem um efeito único garantido, não existindo qualquer incerteza quanto ao resultado da sua execução; já o não-determinístico pode encerrar ações falhas ou não apresentar um efeito único bem determinado;
- estático vs. dinâmico – um ambiente dinâmico pode mudar seu estado enquanto um agente está deliberando, pois existem outros agentes agindo neste ambiente simultaneamente; no estático é suposto permanecer inalterado enquanto o agente decide a próxima ação a executar;
- discreto vs. contínuo – um ambiente é discreto quando existe um número limitado de percepções e ações possíveis para o agente. Enquanto que no contínuo é impossível enumerar todos os estados possíveis; e
- episódico vs. não-episódico – em ambientes episódicos a escolha de cada ação depende somente do episódio em si, ou seja, é independente dos anteriores. Por exemplo, localizar peças defeituosas em etapas distintas de uma linha de montagem. Em contraste, nos não-episódicos a ação atual pode afetar decisões futuras, como em jogos de xadrez.

Analisando as propriedades dos ambientes pode-se concluir que os mais complexos de serem implementados são os inacessíveis, não-determinísticos, dinâmicos, contínuos e não-episódicos, por exemplo, o mundo real.

2.6.2.2 Arquitetura

O projeto de um agente também especifica a definição da arquitetura a ser utilizada, a qual contém a estrutura do modo de relacionamento dos agentes por meio do uso de protocolos de comunicação e interação. As arquiteturas podem ser classificadas como (Wooldridge, 2009; Weiss, 1999):

- baseada em lógica – os agentes tomam decisões por meio de deduções lógicas e o ambiente é simbolicamente representado e manipulado por formalização matemática;

- reativa – os agentes desta arquitetura mapeiam diretamente uma situação para ação;
- BDI (*Beliefs, Desires, Intentions*) – o estado interno de processamento de um agente é descrito por meio de um conjunto de estados mentais definidos, como crenças (*Beliefs*), desejos (*Desires*) e intenções (*Intentions*); e
- híbrida ou em camada – os agentes apresentam comportamento ativo e reativo; e as decisões são tomadas por múltiplas camadas, cada uma representando um nível de abstração.

A arquitetura híbrida pode ser classificada de acordo com o fluxo de controle, o qual pode ser horizontal (Figura 2.10(a)) e vertical (Figura 2.10(b)) (Wooldridge, 2009).

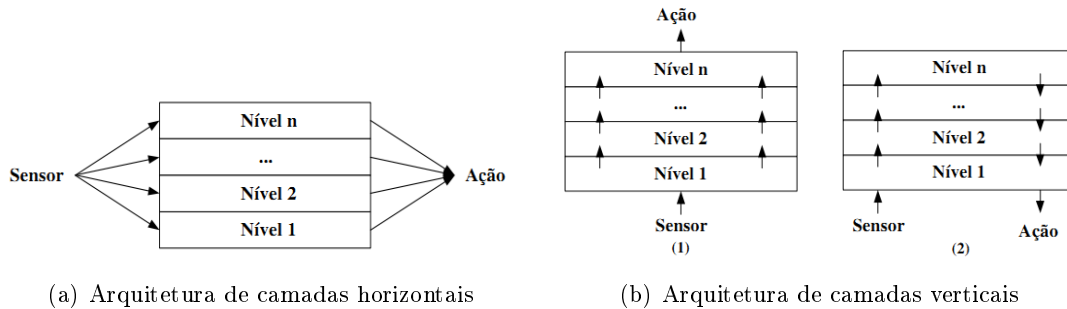


Figura 2.10: Tipos de arquitetura (a) e (b) (adaptado de Wooldridge, 2009).

A Figura 2.10(a) ilustra a arquitetura em camadas horizontais, onde cada camada é um agente e produz sugestões sobre que ação executar. Apresenta como vantagem a simplicidade conceitual, visto que, caso seja necessário que um agente tenha n tipos diferentes de comportamentos, então implementa-se n camadas diferentes. Porém, uma vez que as camadas competem entre si para gerar as ações, uma função mediadora deve ser incluída, o que pode ocasionar um gargalo no processo de decisão do agente.

A Figura 2.10(b) apresenta a arquitetura em camadas verticais, onde o sensor de entrada e a ação de saída estão localizados em no máximo uma camada. As camadas não competem entre si para gerar ações, porém para uma decisão ser tomada um fluxo de controle deve passar por todas as camadas e caso uma delas falhe, o agente terá problemas para gerar e executar uma ação. Pode conter:

- um fluxo de controle (1) – o fluxo passa sequencialmente por cada camada até a última, onde é gerada a ação a executar;
- dois fluxos de controle (2) – primeiro o fluxo sobe as várias camadas até o nível superior, e, depois, o fluxo segue o sentido inverso, dando origem à ação a executar.

2.6.3 Características de SMA

A partir do momento que dois agentes estão presentes em um ambiente, ocorre a necessidade de uma infraestrutura para comunicação e interação entre eles, que serão descritas nas próximas subseções.

2.6.3.1 Comunicação

Na definição apresentada na Seção 2.6.1, assume-se que um agente possui habilidade social para interagir com outros agentes presentes no ambiente. Neste sentido, uma importante característica do SMA é a comunicação de alto nível. Segundo Russell and Norvig (2009), comunicação é a troca intencional de informações estimulada pela produção e percepção de sinais extraídos de um ambiente compartilhado.

A capacidade de comunicação de um agente permite a troca de informações e a coordenação de atividades, resultando em sistemas mais coerentes. Esta interação ocorre por meio de uma linguagem comum dentro de um domínio específico e deve apresentar:

- sintaxe – parte da gramática que contém as regras relativas à disposição das palavras na frase e a das frases no discurso, bem como a relação das palavras e das frases entre si;
- semântica – parte que define o significado dos símbolos e de suas combinações;
- vocabulário – definição do conjunto de palavras que encerram uma linguagem ou língua;
- pragmática – conjunto de regras para interpretar os símbolos utilizados na comunicação; e
- modelo do domínio de discurso – significado de um conjunto de símbolos dentro de um contexto específico.

Algumas linguagens foram definidas no contexto da comunicação em SMA e pode-se destacar (Bellifemine et al., 2007):

- *KQML (Knowledge Query and Manipulation Language)* (Chalupsky et al., 1992) – é baseada em troca de mensagens e caracteriza-se por especificar toda a informação necessária à compreensão do conteúdo da mensagem, ou seja, a linguagem não está preocupada com o conteúdo da mensagem; e
- FIPA ACL - (*Foundation for Intelligent Physical Agents - Agent Communication Language*¹¹) – define essencialmente a estrutura exterior da mensagem e assemelha-se no formato à *KQML*, porém, contém uma semântica mais compreensível.

Observa-se que os agentes precisam de uma linguagem de comunicação que particularize as mensagens que podem ser trocadas e um vocabulário comum bem definido para entender o que está sendo dito. Neste sentido, justifica-se o uso de ontologias neste trabalho (Seção 2.7) para especificação dos conceitos relacionados.

¹¹<http://www.fipa.org/about/index.html>

2.6.3.2 Interação

Após a linguagem de comunicação ser definida, é preciso discutir sobre os mecanismos de interação. Nesta direção, os protocolos de interação estabelecem as etapas de troca de mensagem entre os agentes.

A interação dos agentes deve ser coordenada para resultar um sistema coerente. De acordo com Weiss (1999), *a coordenação é uma característica essencial para um sistema de agentes que realizam alguma atividade em um ambiente compartilhado*. Assim, coordenar é a capacidade de fazer com que os agentes trabalhem em conjunto e consigam combinar seus objetivos para alcançarem o objetivo final do sistema. Existem basicamente duas abordagens para o modo de interação (Wooldridge, 2009):

- cooperativa – todos os agentes do sistema trabalham juntos para alcançar um objetivo geral apoiados na noção de auxílio mútuo; e
- competitivo – os agentes apresentam os mesmos objetivos e usam estratégias individuais para alcançá-los. Geralmente, são aplicados protocolos de negociação, como *Contract Net* (Smith, 1980), para definir um acordo entre os agentes.

2.6.4 Metodologias

Há diversas metodologias de desenvolvimento de software orientado a agentes e uma comparação entre elas, como Gaia, Tropos, Prometheus, MaSE e Passi, é apresentada nos trabalhos de Dam and Winikoff (2003); Henderson-Sellers and Giorgini (2005) e Wooldridge (2009). De modo geral, são denominadas de metodologias AOSE (*Agent Oriented Software Engineering*) e muitas delas foram baseadas nos conceitos de Orientação a Objeto (OO), como exibe a Figura 2.11.

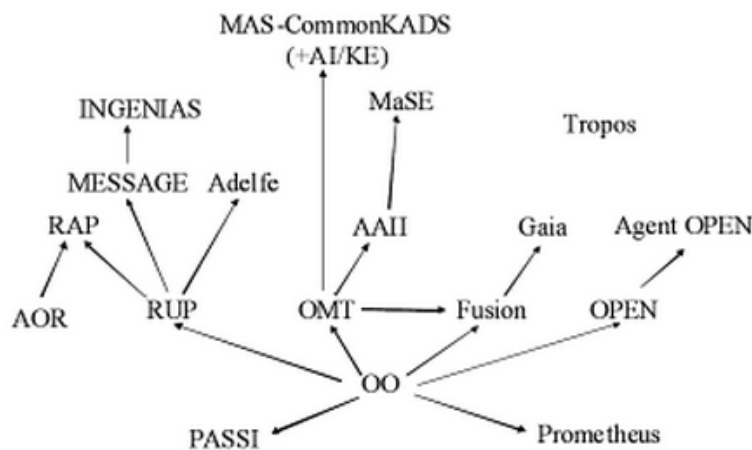


Figura 2.11: Metodologias AOSE (Henderson-Sellers and Giorgini, 2005).

Após estudo, optou-se utilizar a metodologia Prometheus (Padgham and Winikoff, 2002, 2004)

no desenvolvimento do projeto da arquitetura proposta, pois define um processo detalhado e iterativo para a especificação, projeto, implementação, teste e depuração de sistemas de software orientados a agentes. Inclui aspectos do ambiente que o agente está inserido e da organização. Permite o uso tanto por especialistas, quanto por usuários comuns. Oferece ferramentas de suporte gratuitas, como a *Prometheus Design Tool*¹² (PDT), que permite ao usuário verificar inconsistências, gerar automaticamente um conjunto de diagramas e a descrição do projeto, o que inclui descritores para cada entidade, um dicionário para o projeto e os diagramas gerados anteriormente; porém não são obrigatórias para a aplicação desta metodologia. Os artefatos que constituem a metodologia Prometheus são apresentados no Anexo E.

Apresenta três fases de desenvolvimento:

1. especificação do sistema – identifica os objetivos, as funcionalidades básicas do sistema, as ações e percepções dos agentes;
2. projeto arquitetural – utiliza as saídas da fase anterior para determinar os tipos de agentes do sistema e como irão interagir; e
3. projeto de desenvolvimento detalhado – define informações detalhadas sobre cada agente e como eles vão realizar as tarefas dentro do sistema global.

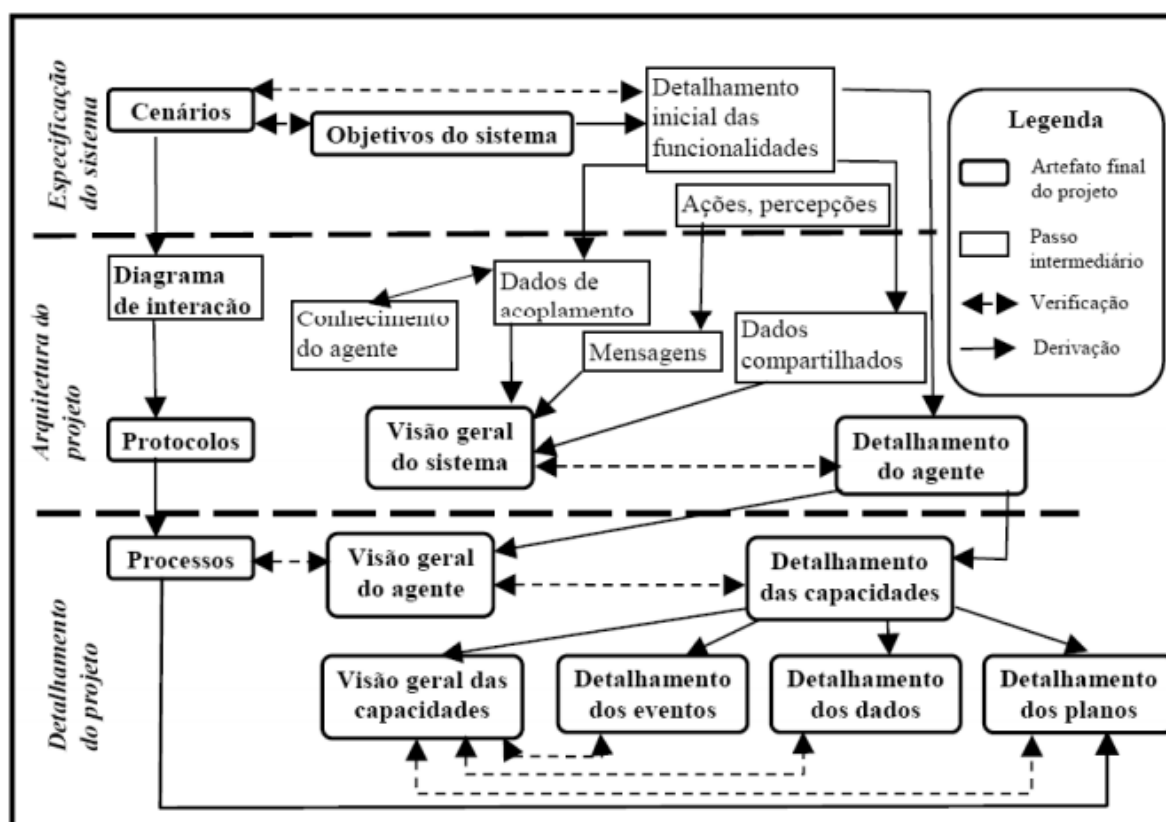


Figura 2.12: Fases da metodologia Prometheus (adaptado de Padgham and Winikoff, 2004).

¹²<http://www.cs.rmit.edu.au/agents/pdt/index.shtml>

A primeira fase, *especificação do sistema*, está diretamente ligada à importância do ambiente (Seção 2.6.2.1) e consiste nos seguintes passos:

- descrever o sistema em alto nível;
- desenvolver cenários de caso de uso; e
- especificar a interface do sistema.

A descrição de alto nível do sistema envolve a definição dos objetivos e funcionalidades. As operações do sistema são ilustradas por meio de casos de uso e a descrição da interface encerra a relação entre o sistema e o ambiente em termos de ações (saídas), percepções (entradas) e dados externos. Estes passos, não sequenciais, devem ser repetidos até obter o nível de detalhamento desejado.

A segunda fase, *projeto arquitetural*, foca principalmente na definição dos agentes (Seção 2.6.1). Segundo Padgham and Winikoff (2004), a maior decisão a ser tomada durante esta fase é determinar quais agentes devem existir; sendo constituída por três atividades:

- definir os tipos de agentes;
- projetar a estrutura global do sistema; e
- definir as interações entre os agentes.

A definição dos tipos de agentes é baseada no agrupamento das funcionalidades. Isto envolve analisar se as funcionalidades lidam com os mesmos dados ou não. Após a caracterização dos tipos, determina-se os descritores de agentes que contêm informações de alto nível, como o ciclo de vida, os dados que utiliza e produz, os eventos que o agente irá reagir, os objetivos, entre outros. A estrutura geral e estática do sistema é apresentada por um diagrama de visão do sistema (agentes + eventos + dados), enquanto que o comportamento dinâmico dos agentes é representado por meio de diagramas e protocolos de interação.

A última fase, *projeto de desenvolvimento detalhado*, corresponde à estrutura interna de cada agente e à realização das tarefas no sistema global; consiste das seguintes atividades:

- definir cada agente em termos de capacidade;
- desenvolver processo de especificação;
- modelar planos para tratar eventos; e
- definir detalhes dos eventos, planos e dados.

O refinamento progressivo em termos de capacidade gera um diagrama de visão geral dos agentes, focando na estrutura interna do agente fora do sistema. Também são definidos os descritores

que contêm informações sobre quais eventos são gerados e recebidos pelo agente, quais são as trocas de dados e objetivos individuais.

É uma fase que deve ser intercalada e repetida, onde ocorre a tradução iterativa da parte dinâmica do sistema, ou seja, protocolos de interação em especificações de processo. Inclui igualmente a determinação de planos para caracterizar o comportamento do agente frente aos eventos. Além da descrição dos detalhes para os eventos, planos e dados para poder iniciar o processo de implementação do sistema.

Deste modo, a metodologia Prometheus provê suporte completo e detalhado para cada fase de desenvolvimento de um projeto de SMA com a disponibilização de uma documentação bem definida, além de ferramentas para o projeto de diagramas.

2.6.5 Aplicações de SMA

SR podem utilizar agentes para inserir características, como autonomia, reatividade, proatividade e habilidade social, visando melhorar a qualidade da recomendação ao usuário. Por exemplo, Al Tair et al. (2012) apresentam uma arquitetura baseada em agentes para incluir proatividade em recomendações sensíveis ao contexto de pacotes turísticos. Vizcaíno et al. (2010) utilizam agentes para simular o comportamento social de um indivíduo em uma sociedade, onde uma arquitetura de agentes recomenda documentos em uma comunidade. Algumas iniciativas de pesquisa em CARS para Universidade Pervasiva, que se correlacionam com a proposta deste trabalho, são apresentadas na Seção 2.8.

2.7 Ontologia

Um ambiente sensível ao contexto é inerentemente heterogêneo e dinâmico, onde diferentes dispositivos devem interagir e ter mobilidade entre os ambientes. Esta interação deve ser o mais consensual possível para evitar interpretações ambíguas a respeito da semântica dos termos de um domínio. Assim, entende-se que para possibilitar a representação de um domínio, a interoperabilidade semântica e o conhecimento compartilhado de um grupo heterogêneo, é importante uma ferramenta que determine um vocabulário comum bem definido. O uso de ontologias permite a especificação formal e conceitual sobre um domínio de interesse, incluindo conceitos, relacionamentos, restrições e instâncias.

Na área da Ciência da Computação, uma definição clássica de ontologia é apresentada por Gruber (1993), em que ontologia é *uma especificação explícita de uma conceitualização*. Este trabalho adota a definição de Studer et al. (1998):

“uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada.”

Deste modo, as entidades relevantes para um domínio específico podem ser conceitualizadas, isto é, representadas e definidas. Após a conceitualização, a noção de domínio pelas entidades é

consensual; e para permitir o compartilhamento, os conceitos devem apresentar uma especificação formal e explícita. Com isso, é reduzida a ambiguidade na interpretação do vocabulário definido.

As ontologias podem ser classificadas de acordo com: (i) a função; (ii) o grau de formalismo do vocabulário; (iii) a aplicação; e (iv) a estrutura e conteúdo da conceitualização (Almeida and Bax, 2003). Segundo Almeida and Bax (2003), mesmo sem um consenso no critério de classificação, existem semelhanças entre suas funções. Isto permite a inserção simultânea de uma ontologia em mais de uma classificação. Conhecer os principais tipos e características de cada classificação possibilita definir uma que seja adequada à utilização desejada. A Figura 2.13 apresenta a classificação de acordo com a estrutura (Guarino, 1998), a qual serviu como fundamentação para o desenvolvimento da ontologia definida neste projeto de pesquisa.

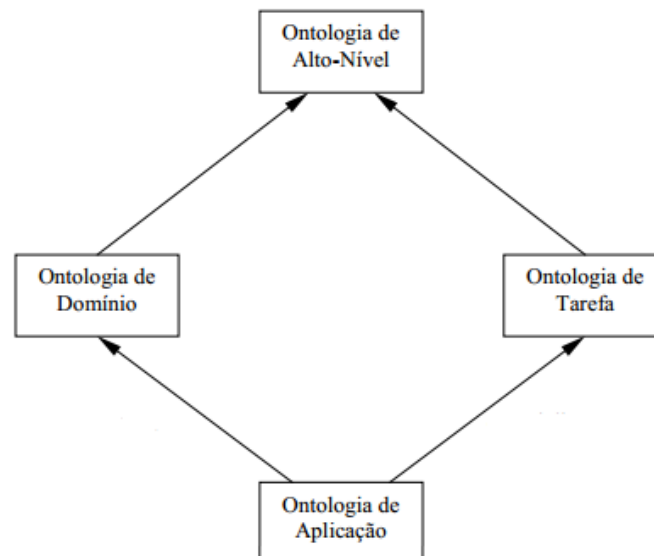


Figura 2.13: Tipos de ontologias e seus relacionamentos (adaptado de Guarino, 1998).

- alto nível – descreve conceitos gerais, como espaço, tempo, e são independentes de um problema particular ou domínio específico;
- domínio – descreve o vocabulário relacionado ao domínio geral por meio da especificação de conceitos introduzidos na ontologia de alto nível;
- tarefa – refere-se ao vocabulário de uma tarefa ou atividade geral por meio da especificação de conceitos introduzidos na ontologia de alto nível; e
- aplicação – ontologia mais específica que define os conceitos relativos a uma determinada aplicação.

Oberle et al. (2009) ressaltam que a ontologia de domínio é apenas uma referência para o domínio a ser representado. Em aplicações computacionais, as ontologias devem ser simplificadas para ontologias de aplicação. Deste modo, uma ontologia de aplicação é definida e utilizada nesta pesquisa conforme apresentado na Seção 4.

Noy and McGuinness (2001) identificam as características básicas de uma ontologia:

- classes – conceitos relevantes de um domínio organizados em uma hierarquia taxonômica;
- relações – tipo de associação entre os conceitos de um domínio particular;
- axiomas – usados para modelar sentenças sempre verdadeiras; e
- instâncias – representam elementos ou indivíduos em um domínio.

A Figura 2.14 ilustra os componentes citados em um domínio para vinhos, onde as classes são representadas em preto e as instâncias em vermelho. As arestas representam os relacionamentos dos tipos: relações entre classes e instâncias (rotuladas com *io* – *is of*) e propriedades (rotuladas com o nome das propriedades). As restrições são expressas em axiomas utilizando lógica descritiva e não são representadas graficamente.

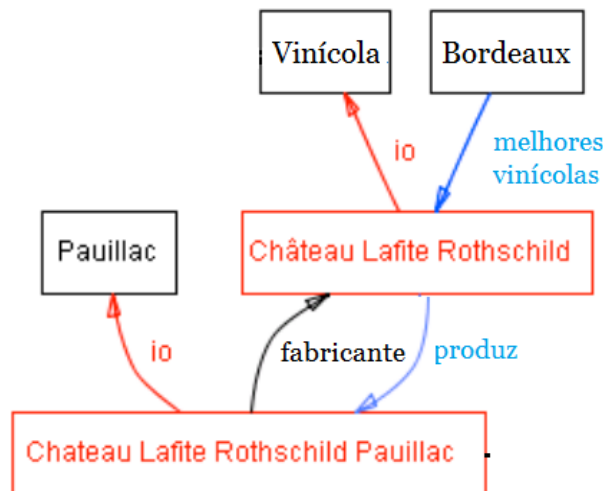


Figura 2.14: Representação simplificada dos componentes básicos da ontologia (adaptado de Noy and McGuinness, 2001).

Para que uma ontologia possa ser efetivamente utilizada, ela precisa ser especificada em uma determinada linguagem. Segundo Corcho et al. (2003), as linguagens para a representação formal de ontologias podem ser classificadas em três categorias, que diferem em graus de formalismo e poder de expressividade:

- linguagens tradicionais – CycL (Lenat and Guha, 1989), KIF (*Knowledge Interchange Format*) (Genesereth and Fikes, 1992), Ontolingua (Gruber, 1992), CML (*Conceptual Modeling Language*) Schreiber et al. (1994), OCML (*Operational Conceptual Modelling Language*) (Shadbolt et al., 1993);
- linguagens padrões para *Web* – XML (*Extensible Markup Language*) (Quin, 2001) e RDF (*Resource Description Framework*) (Beckett, 2004); e
- linguagens para ontologia no contexto da *Web Semântica* – OIL (*Ontology Inference Layer*) (Fensel et al., 2000), DAML+OIL (*DARPA Agent Markup Language*) (Harmelen et al., 2001),

SHOE (*Simple HTML Ontology Extensions*) (Luke and Heflin, 2000), XOL (*Ontology Exchange Language*) (Karp et al., 2000) e OWL (*Ontology WeB Language*) (McGuinness and Harmelen, 2004).

Uma descrição, comparação e avaliação entre as linguagens acima citadas pode ser encontrada em Su and Ilebrikke (2002) e Corcho et al. (2003).

Ante o exposto, ao representar o conhecimento de um domínio específico por meio de uma ontologia, obtém-se a representação semântica e explícita do conjunto das informações relevantes para o contexto, que podem ser manipuladas para direcionar o comportamento de aplicações sensíveis ao contexto. Além disso, permite que os agentes compartilhem e reutilizem as informações de contexto. Porém, Henricksen et al. (2004) apontam alguns problemas neste tipo de representação:

- os padrões nos quais as ontologias e seus motores de inferência se baseiam ainda são limitados;
- a linguagem OWL ainda não provê suporte direto para regras axiomáticas, o que limita os tipos de raciocínio;
- não suportam adequadamente raciocínio sobre informações de contexto consideradas imprecisas ou ambíguas; e
- criar ou estender ontologias para informações de contexto é complexo e sujeito a falhas.

A modelagem de contexto por ontologias permite além do compartilhamento e o reuso do conhecimento, a inferência de novas informações por meio de mecanismos de raciocínio, aproveitando o conhecimento implícito, ou seja, aquele que não está diretamente incluído em uma ontologia. Por exemplo, uma ontologia em que parente está definido como um relacionamento mais geral do que irmão; neste caso, se João é irmão de José, o sistema conclui que João é parente de José. Assim, se um usuário consultar esta ontologia perguntando quem é parente de João, o sistema responderá que José é parente de João, sem que esse fato tenha sido declarado.

Este trabalho utiliza a linguagem *OWL* para a descrição semântica do ambiente, uma vez que pretende disponibilizar informações no ambiente Web. Em relação à representação de contexto, foi utilizado o modelo ontológico denominado *Semantic Context Model* (SeCoM) (Bulcão Neto, 2006), o qual é baseado em ontologias e em padrões de Web Semântica. O modelo SeCoM foi definido para apoiar o desenvolvimento de aplicações sensíveis ao contexto, conforme será apresentado na Seção 3.2.

2.8 Trabalhos Correlatos

Os trabalhos de pesquisa apresentados nesta seção foram escolhidos durante o processo de ReS descrito na Seção 2.1, especificamente para responder a questão de projeto: “Como prover informações ao usuário de acordo com o contexto?”. Ressaltando que o domínio de oferta dessas

informações é o da Universidade Pervasiva. Principalmente, os que têm como objetivo auxiliar de modo personalizado o ciclo de vida estudantil dentro da universidade a qualquer hora e lugar.

Al Takroui et al. (2007) apresentam o eyeJOT, um sistema de informação sensível ao contexto baseado em proximidade. A principal motivação do desenvolvimento do projeto é prover um quadro de avisos interativo e comunitário inserindo sensibilidade ao contexto no ambiente. Os estudantes são capazes de recuperar e compartilhar informações sobre as atividades que ocorrem no campus, como eventos e calendário do semestre. O nível de proximidade com o quadro de avisos rege a zona de personalização que pode ser:

- módulo protetor de tela – nenhum estudante é detectado no raio de alcance do sistema;
- mensagens comunitárias – um aluno é detectado dentro de um raio de 1.5 metros a 3 metros e informações gerais são exibidas no quadro de avisos; e
- informações personalizadas – são apresentadas de acordo com o perfil cadastrado pelo estudante no sistema, neste caso, uma agenda detalhada do mês corrente.

O contexto é definido como: (i) identificação do estudante por meio de dispositivos com Bluetooth; (ii) localização física; (iii) hora e data corrente; (iv) agenda pessoal; e (v) calendário. As informações de perfil são configuradas manualmente pelos estudantes que acessam o sistema.

Furbach et al. (2007) e Maron et al. (2008) descrevem o sistema de informação *Campus News*¹³, o qual permite que os estudantes da Universidade de Koblenz-Landau, na Alemanha, encontrem e acessem informações relevantes de acordo com os interesses por eles definidos. Os autores consideram este sistema como uma parte do quebra-cabeça para a construção da Universidade Pervasiva. Nós de serviços estão espalhados pelo campus ofertando informações sobre a cafeteria, promoções no restaurante e eventos que ocorrem na universidade. Estas ofertas são enviadas por mensagens *broadcasting* para usuários próximos aos nós e que possuam dispositivos habilitados para Bluetooth. As informações são filtradas pelo servidor baseado no perfil do estudante, que foi cadastrado por meio de um questionário em uma página web indicando interesses próprios. A arquitetura do sistema consiste de três componentes: (i) uma aplicação web para interface com o usuário; (ii) o servidor; e (iii) os nós de serviço que entregam as informações nos dispositivos móveis.

Haron et al. (2010) apresentam o *R-CCANS*, um sistema de notificação sensível ao contexto baseado em tecnologia RFID para estudantes universitários. Uma das motivações dos autores coincide com a deste projeto de tese, pois afirmam que os canais de comunicação existentes em uma universidade, como o quadro de avisos, dificilmente alcançam a audiência pretendida e interessada devido ao limite de área de cobertura deste meio. Por isso, a necessidade de um sistema sensível ao contexto que informe de modo personalizado os estudantes. O contexto foi modelado como: (i) hora corrente; (ii) identificação, consiste no perfil do estudante composto de id, curso e preferências (livro, classes, esporte, eventos e miscelânea); e (iii) localização. As informações personalizadas são derivadas do perfil que é adquirido diretamente do estudante pelo preenchimento de um cadastro. O mecanismo de inferência do sistema é baseado em regras de produção. Como o ambiente real é

¹³<http://campusnews.uni-koblenz.de/>

contínuo e estocástico, o uso de regras pode prejudicar o desempenho do sistema de recomendação na ocorrência de uma situação não prevista pelo sistema.

Lin et al. (2013) propõem a recomendação de cursos dirigida por meio de um protocolo de negociação que considera as preferências e prioridades dos estudantes e da administração da instituição de ensino. Todo o sistema é baseado em agentes porque permitem a representação de um indivíduo autônomo com objetivos, a integração desses indivíduos, além da flexibilidade que apresentam para reagir às mudanças ocorridas no ambiente. Deste modo, os estudantes e a administração são modelados por agentes com interesses próprios. A interação entre os agentes ocorre quando uma lista de matérias é apresentada pelo agente de administração, esta ação inicia o processo de negociação com os agentes estudantes para a recomendação de matérias em um semestre específico, conforme as preferências e prioridades de cada um.

A análise destes trabalhos permitem a identificação de características importantes para a recomendação de serviços baseado em contexto em um ambiente pervasivo como:

- tipo de modelagem do perfil do usuário;
- definição do contexto de recomendação;
- o modo de interação usuário-sistema; e
- decisão de aplicação de métodos de aprendizagem.

Porém, diferente das iniciativas apresentadas, este trabalho foca na transparência de interação e proatividade da recomendação contextualizada e personalizada baseada em agentes, considerando o domínio da Universidade Pervasiva. O paradigma de agentes foi escolhido por ser capaz de representar o indivíduo, o ambiente e as interações sociais que ocorrem entre os indivíduos e indivíduo-aplicação em uma Universidade Pervasiva, onde a dinamicidade do domínio pode ser representada por meio de diferentes tipos de agentes.

Outra diferença da proposta desta tese dos trabalhos de Al Takroui et al. (2007); Furbach et al. (2007); Maron et al. (2008); Haron et al. (2010) e Lin et al. (2013) é o uso do algoritmo de propagação da ativação (Seção 2.5), que possibilita a recomendação de acordo com as diferentes situações de contexto de modo implícito. Permitindo que a aplicação se adapte às preferências do usuário em contextos variados e sem solicitar constantemente sua atenção. Estas características citadas permitem a transparência, proatividade e personalização da recomendação contextualizada em um ambiente pervasivo.

O Capítulo 3 descreve o modelo conceitual e arquitetural de recomendação sensível ao contexto baseada em agentes no contexto de uma Universidade Pervasiva.

Capítulo 3

Arquitetura Proposta

Este capítulo apresenta a modelagem semântica do contexto por meio de ontologia, a descrição arquitetural dos agentes baseado na metodologia Prometheus e o algoritmo de propagação da ativação utilizado neste trabalho.

3.1 Visão geral

O trabalho de pesquisa descrito nesta tese resultou no desenvolvimento do modelo arquitetural proposto, o qual inclui quatro etapas distintas, que serão detalhadas nas seções subsequentes, a saber:

1. modelo de contexto – a definição do modelo de contexto segue a estrutura da informação definida pelo modelo SeCoM, proposto por Bulcão Neto (2006), juntamente com a ontologia de domínio denominada OntoUnB, a qual foi definida utilizando o Método 101 (Noy and McGuinness, 2001);
2. modelo arquitetural – a modelagem dos agentes foi realizada utilizando a metodologia Prometheus (Padgham and Winikoff, 2002, 2004);
3. recomendação – módulo que agrega o *framework* OntoSpread (Rodríguez et al., 2013) aos demais elementos arquiteturais; e
4. integração – integração do modelo de contexto à arquitetura de agentes com o *framework* OntoSpread, os quais foram testados pelo desenvolvimento da aplicação *eAgora?*.

3.2 Modelo de contexto

Pelo exposto na Seção 2.7, o modelo de contexto deste trabalho conta com a definição de uma ontologia, a qual é importante para:

- a representação semântica do domínio específico, onde são definidos formalmente os conceitos e relações existente;
- a interoperabilidade semântica dos agentes, o que viabiliza o compartilhamento de informação; e
- a manipulação e recomendação de informação envolvida no domínio, a qual está sendo tratada pela aplicação.

O modelo de contexto foi definido utilizando o editor de ontologia Protégé versão 4.1 beta, de código aberto, uma vez que este permite a criação, visualização e manipulação de ontologias em diferentes formas de representação. O Protégé permite o desenvolvimento de ontologias em diversas linguagens reconhecidas pelo *World Wide Web Consortium* (W3C), mas foi utilizado neste trabalho a linguagem OWL, porque além de ser padrão W3C, ela facilita a manipulação de conteúdos na Web via aplicações.

Existem algumas metodologias que auxiliam na construção de uma ontologia, como a *OntoClean* (Guarino and Welty, 2002), *Enterprise Ontology* (Uschold et al., 1995) e Método 101 (Noy and McGuinness, 2001). Para a escolha da metodologia são identificados o que é abordado em cada uma delas, destacando as vantagens e limitações. A partir disso, é avaliada a melhor metodologia que atende às necessidades para a formalização da ontologia.

O processo de especificação da ontologia de domínio deste trabalho, doravante denominada OntoUnB, seguiu o Método 101 (Noy and McGuinness, 2001) por apresentar etapas bem definidas e detalhadas, proporcionando maior facilidade de especificação. Além disso, Noy and McGuinness (2001) destacam que não há uma forma ou metodologia correta para o desenvolvimento de ontologias. O Método 101 é composto por sete passos, a saber:

1. definir o domínio e o escopo da ontologia;
2. considerar o reuso de ontologias já existentes;
3. enumerar os termos importantes da ontologia;
4. definir as classes e as hierarquias;
5. definir as propriedades das classes;
6. definir os valores das propriedades; e
7. criar instâncias individuais das classes.

Após a definição do domínio e o escopo da ontologia voltado para o ambiente universitário, optou-se reutilizar o modelo SeCoM (Bulcão Neto, 2006) baseado em ontologia e em padrões da WeB Semântica. O modelo SeCoM foi escolhido por apoiar o desenvolvimento de aplicações sensíveis ao contexto e ser caracterizado como um modelo independente de domínio. Neste modelo, as informações de contexto são representadas com semântica explícita de modo a facilitar o seu

compartilhamento, reúso para diferentes domínios de aplicação e processamento de sistemas de software. Ele também é escrito na linguagem OWL e segue as dimensões semânticas definidas por Abowd and Mynatt (2000) e Truong et al. (2001) apresentadas na Seção 2.2: quem? (*who?*), onde? (*where?*), quando? (*when?*), o quê (*what?*), por quê? (*why?*) e como? (*how?*). O modelo SeCoM é dividido em duas camadas conforme Figura 3.1:

- camada superior – representa o modelo em si e contém as sete classes principais do modelo representado no formato oval escuro; e
- camada inferior – composta por seis classes de apoio representado no formato oval claro, construída com conceitos específicos da aplicação sensível ao contexto que está sendo desenvolvida.

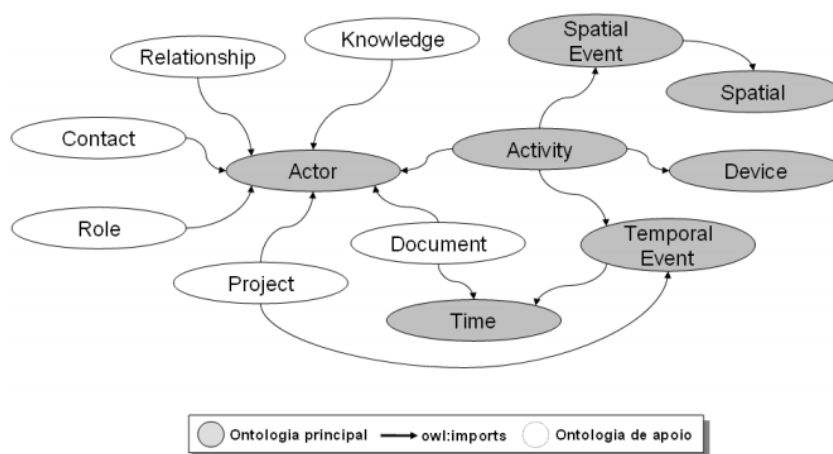


Figura 3.1: Visão Geral do Modelo SeCoM (Bulcão Neto, 2006).

Segue uma breve descrição semântica das classes principais (ovais escuras) e das classes de apoio (ovais claras) representadas na Figura 3.1:

- Ator (*Actor*) – representa informações de contexto de entidades que executam alguma ação em uma interação usuário-computador;
- Espaço (*Spatial*) – descreve informações de localização;
- Evento espacial (*Spatial Event*) – extensão da classe Espaço para representar eventos que ocorrem em algum espaço físico ou virtual e as relações espaciais e mereológicas entre esses elementos;
- Tempo (*Time*) – representa a idéia temporal no modo de instantes ou intervalos de tempo e as relações temporais entre esses elementos;
- Evento temporal (*Temporal Event*) – complementa as informações descritas na classe Tempo e representa eventos que possuem alguma extensão temporal;

- Atividade (*Activity*) – descreve as ações que os atores realizam em um ambiente sensível ao contexto;
- Dispositivo (*Device*) – refere-se aos dispositivos computacionais de captura e acesso das informações de um ambiente sensível ao contexto;
- Conhecimento (*Knowledge*) – descreve informações sobre o conhecimento e interesse de uma pessoa a respeito de uma área de conhecimento específica;
- Relacionamento (*Relationship*) – modela os tipos de relacionamentos sociais existentes entre pessoas, por exemplo, se as pessoas são amigas, então elas se conhecem;
- Papel (*Role*) – descreve informações sobre o papel social de um ator, por exemplo, aluno e empregado;
- Contato (*Contact*) – descreve informações de contato de um ator, como número de telefone e endereço;
- Documento (*Document*) – descreve informações de artefatos produzidos por um ator na forma de documentos físicos ou eletrônicos; e
- Projeto (*Project*) – modela projetos que atores podem estar envolvidos.

As classes de apoio do modelo SeCoM: (i) Conhecimento; (ii) Relacionamento; (iii) Papel; (iv) Contato; (v) Documento; e (vi) Projeto, suportam a descrição de perfis de atores. Por esse motivo, elas importam a classe Ator. Para maiores detalhes consultar Bulcão Neto (2006).

Após o estudo das noções conceituais das classes e dos relacionamentos existentes entre elas no modelo SeCoM, a OntoUnB utilizou e adaptou nove classes do modelo:

- Ator;
- Espaço;
- Evento Espacial;
- Tempo;
- Evento Temporal;
- Atividade;
- Conhecimento;
- Papel; e
- Contato.

As classes: (i) Dispositivo; (ii) Relacionamento; (iii) Documento; e (iv) Projeto foram suprimidas por não se aplicarem às informações de contexto definidas neste trabalho de pesquisa. A Figura 3.2 apresenta as principais classes e relações da OntoUnB. A Tabela 3.1 exibe a relação entre as classes da OntoUnB e o modelo SeCoM. A classe informações foi acrescentada na OntoUnB para representar: (i) eventos culturais; (ii) eventos acadêmicos; e (iii) serviço de localização.

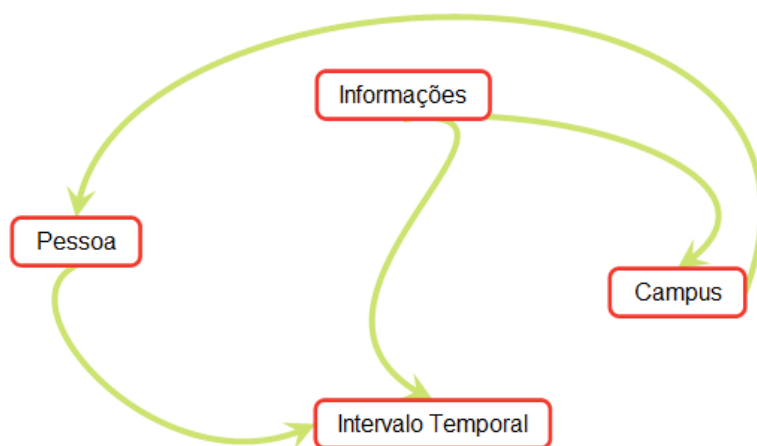


Figura 3.2: Visão Geral da OntoUnB.

Tabela 3.1: Relação entre classes OntoUnB e modelo SeCoM.

Classes OntoUnB	Classes SeCoM
Pessoa	Ator
	Atividade
	Conhecimento
	Papel
	Contato
Intervalo Temporal	Tempo
	Evento Temporal
Campus	Espaço
	Evento Espacial

Na terceira fase do Método 101, enumerar os termos importantes para a ontologia, determina-se neste trabalho que uma universidade pode ser definida de acordo com cinco entidades básicas, a saber (Lamas, 2009):

- pessoas – constituem os estudantes, funcionários e visitantes do campus. Sendo que os estudantes podem ser de graduação ou pós-graduação e os funcionários podem ser professores, servidores técnicos administrativos ou terceirizados;
- estrutura administrativa – reflete a estrutura hierárquica da universidade, incluindo, conselhos superiores, reitoria, unidades acadêmicas, órgãos complementares e centros;

- estrutura acadêmica – representa os institutos e as faculdades, bem como as atividades de ensino, pesquisa e extensão nas respectivas áreas;
- atividades – corresponde às tarefas que os usuários podem realizar no campus. Importante para a recomendação de eventos acadêmicos e culturais que ocorrem no campus;
- infraestrutura representada espacialmente – encerra as estruturas que apresentam uma representação geo-espacial no campus.

Deste modo, o modelo de contexto da OntoUnB foi conceituado de acordo com estas entidades e o estatuto da Universidade de Brasília (UnB¹) resultando em quatro entidades de contexto, a saber:

- Informações – representam os eventos acadêmicos e culturais que ocorrem no campus, além do serviço de localização;
- Pessoa – representa um papel importante e central para a aplicação. Define o vocabulário para descrever entidades que executam alguma interação usuário-computador, encerra os visitantes, o perfil e as atividades realizadas por ele;
- Campus – localização do usuário e dos eventos que ocorrem no campus; e
- Intervalo Temporal – descreve o tempo do usuário e dos eventos.

A delimitação dessas entidades de contexto facilita a aplicação identificar quais informações devem ser manipuladas durante um interação usuário-sistema para poder recomendar eventos; além de determinar as classes, hierarquias e propriedades do domínio, respectivamente quarta e quinta fase do Método 101.

Neste trabalho, a especificação de classes adota um processo de desenvolvimento *top-down*, ou seja, começa com a definição dos conceitos mais gerais do domínio e, posteriormente, é feita a especialização desses conceitos. Deste modo, o modelo completo da OntoUnB é composto por mais de duzentas classes, sendo que as classes principais são: (i) Informações; (ii) Pessoa; (iii) Campus; e (iv) Intervalo Temporal. A Figura 3.3 representa graficamente a taxonomia de classes por meio do *plugin* OWL Viz disponível no editor Protégé.

Cada uma das classes que compõem a OntoUnB representadas na Figura 3.3 são descritas a seguir.

3.2.1 Informações

A classe Informações é definida como um conjunto organizado de dados que descreve os eventos acadêmicos e culturais que são recomendados aos usuários do campus universitário. Ela foi baseada no informativo UnB Hoje, que divulga diariamente os eventos acadêmicos, científicos e culturais da UnB há 27 anos. Além disso, contém o serviço de localização disponível para todas as pessoas que circulam no campus, no caso de visitantes este serviço se mostra de grande utilidade.

¹http://www.unb.br/unb/transparencia/downloads/regimento_estatuto_unb.pdf

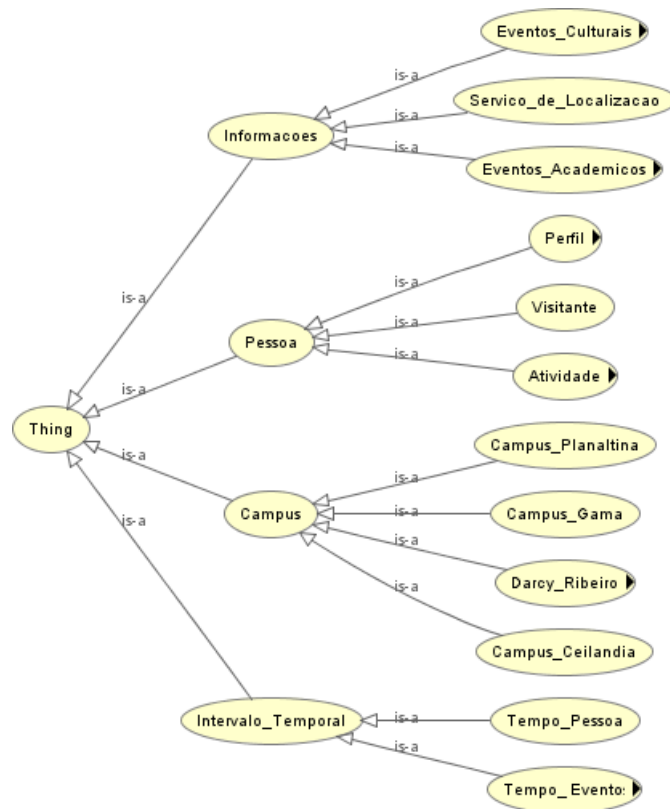


Figura 3.3: Taxonomia de classes da OntoUnB.

A classe Informações apresenta três subclasses: (i) Eventos Acadêmicos; (ii) Eventos Culturais; e (iii) Serviço de Localização, conforme Figura 3.4. A subclasse Eventos Acadêmicos descreve os eventos acadêmicos e científicos que acontecem no campus universitário e podem ser classificados em:

- congresso;
- cursos;
- palestra;
- seminários;
- defesas;
- conferências; e
- *workshop*.

A subclasse Eventos Culturais representa os eventos culturais que acontecem no campus universitário e podem ser classificados em:

- filme;



Figura 3.4: Taxonomia expandida da classe Informações.

- teatro;
- exposição; e
- show.

Por último, a subclasse Serviço de Localização está associada ao fornecimento da localização simbólica, interna ou externa, do usuário no campus universitário.

3.2.2 Pessoa

Derivada conceitualmente da classe Ator e suas extensões (Atividade, Conhecimento, Papel e Contato) do modelo SeCoM, a classe Pessoa é definida para representar as pessoas que circulam pelo campus universitário, como os estudantes, funcionários e visitantes, e, conseqüentemente, interagem com a aplicação. Conforme apresentado na Figura 3.5, esta classe possui três subclasses, a saber:

- Atividade – os usuários estão envolvidos em alguma atividade no campus, a qual ocorre em um intervalo de tempo e uma localização física;
- Perfil – representa um papel importante para a aplicação de recomendação sensível ao contexto, pois é nesta subclasse que estão contidas as informações que indicam as preferências dos usuários. Estas informações são:
 - necessidades especiais;

- interesses acadêmicos;
 - matrícula;
 - vinculação;
 - CPF;
 - interesses culturais;
 - e-mail;
 - grandes áreas; e
 - idioma.
- Visitante – representa as pessoas que frequentam o campus e não apresentam vinculação institucional.

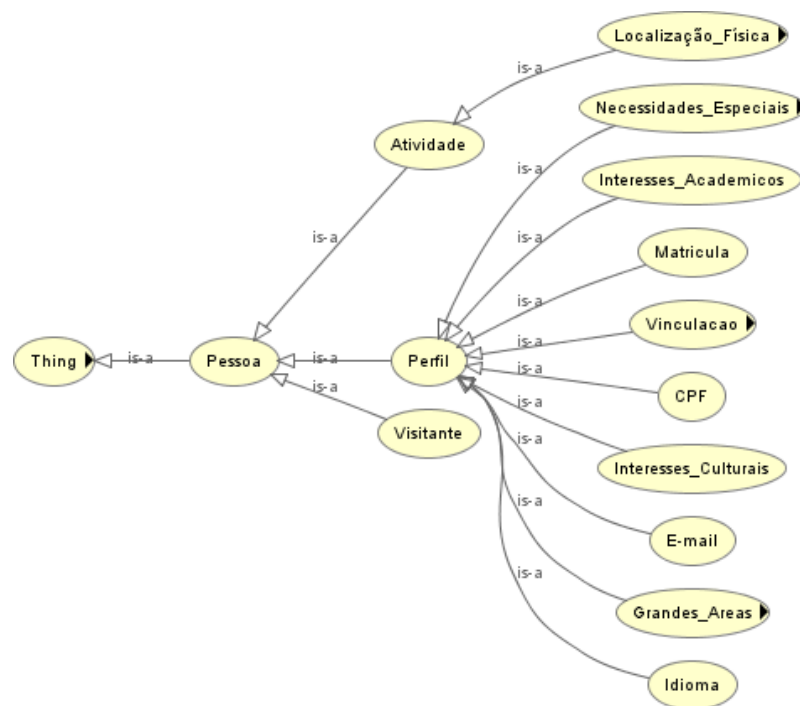


Figura 3.5: Taxonomia expandida da classe Pessoa.

Para relacionar as subclasses foram criadas as seguintes propriedades de objeto, a saber:

- *temAtividade* – relaciona a classe Pessoa com a subclasse Atividade;
- *temLocalizaçãoFísica* – relaciona a subclasse Localização Física com a classe Atividade;
- *temPerfil* – relaciona a subclasse Perfil com a classe Pessoa; e
- *temVisitante* – relaciona a subclasse Visitante com a classe Pessoa.

3.2.3 Campus

A classe Campus, derivada conceitualmente da classe Espaço e Evento Espacial do modelo SeCoM, define os locais físicos dentro do campus universitário. O local pode ser o próprio campus, o qual possui uma estrutura administrativa e acadêmica. A Figura 3.6 contém quatro subclasses que referenciam os campus Ceilândia, Planaltina, Gama e Darcy Ribeiro.

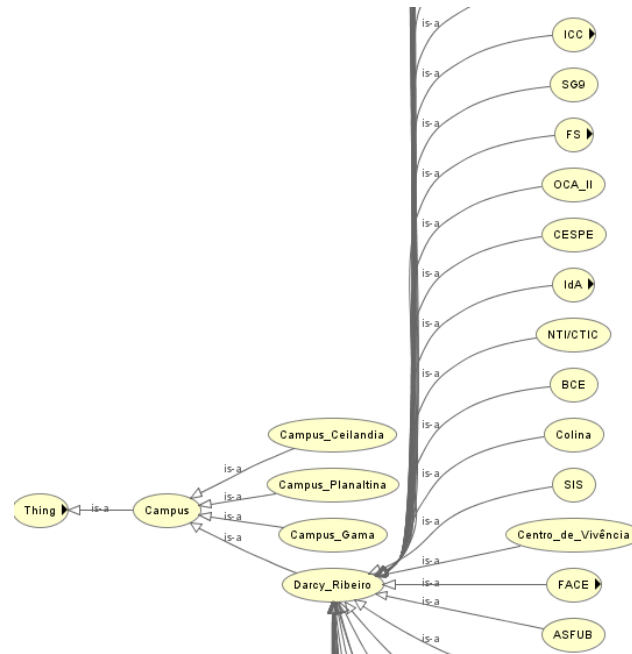


Figura 3.6: Taxonomia expandida da classe Campus.

A priori, apenas a subclasse Darcy Ribeiro foi mapeada e contém os principais prédios existentes no campus. Dentro de cada edificação, constam as estruturas acadêmicas ou administrativas. Esta subclasse está relacionada com a localização espacial do campus, de modo que cada edificação possui uma coordenada espacial.

A estrutura acadêmica e administrativa da UnB é composta pela: Reitoria, Vice-Reitoria, Decanatos, Centros, Órgãos Auxiliares, Órgãos Complementares, Diretorias, Faculdades, Institutos e Departamentos. Os diversos cursos com suas disciplinas são vinculados aos Departamentos, os quais estão integrados a uma específica Faculdade ou Instituto. Para relacionar os indivíduos existentes dentro da subclasse Darcy Ribeiro, foram criadas propriedades de objetos (*Object Properties*) para as principais edificações da instituição, como Instituto Central de Ciências (ICC), Faculdade de Tecnologia (FT) e Restaurante Universitário (RU). A Figura 3.6 foi editada por questões de formatação, mas todas as unidades contantes da estrutura da UnB estão definidas na OntoUnB, conforme estrutura organizacional².

²<http://www.dpo.unb.br/organogramaunb.php>

3.2.4 Intervalo Temporal

A classe Intervalo Temporal, derivada conceitualmente das classes Tempo e Evento Temporal do modelo SeCoM, define o tempo relacionado às atividades de cada usuário e o intervalo de tempo relacionado aos eventos no campus. É constituída de duas subclasses, conforme Figura 3.7: (i) Tempo Evento e (ii) Tempo Pessoa. Sendo que a subclasse Tempo Evento também apresenta duas subclasses: (i) Tempo Início e (ii) Tempo Término, relacionadas com o início e término de um evento, respectivamente. Esta classe facilita a validação de informações inconsistentes, por exemplo, recomendar um evento que já passou para um usuário.

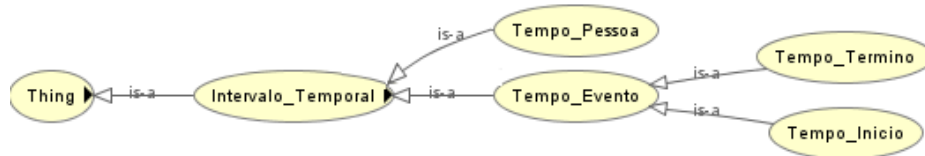


Figura 3.7: Taxonomia expandida da classe Intervalo Temporal.

Para relacioná-las, foram criadas as seguintes propriedades de objetos:

- *temTempoPessoa* que relaciona a classe Intervalo Temporal com a subclasse Tempo Pessoa;
- *temTempoEvento* que relaciona a classe Intervalo Temporal com a subclasse Tempo Evento;
- *temTempoInicio* que relaciona a subclasse Tempo Evento com a subclasse Tempo Início; e
- *temTempoTermino* que relaciona a subclasse Tempo Evento com a subclasse Tempo Término.

Após a definição das classes, propriedades e valores (Fases quatro, cinco e seis do Método 101), começa a última fase do desenvolvimento de uma ontologia que é a instanciação. Estas instâncias são especificadas na Seção 4.1 por meio de uma prova de conceito do modelo proposto.

Uma vez que as entidades de contexto de um ambiente tenham um modelo semântico, uniforme e padronizado de representação, o entendimento durante o processo de comunicação e manipulação de informações pelos diversos agentes da arquitetura é único. Depois deste modelo formalizado, é necessário uma arquitetura que a suporte e seja capaz de manipular tais informações para recomendar eventos em um campus universitário. Assim, a Seção 3.3 apresenta a arquitetura proposta nesta pesquisa.

3.3 Modelo arquitetural

Pelo exposto na Seção 2.6, SMA é utilizado neste trabalho para a modelagem dos agentes e as interações com o sistema. Por isso, esta seção apresenta os detalhes da modelagem da arquitetura baseada em agentes por meio da metodologia Prometheus (Seção 2.6.4). Utilizou-se a ferramenta

PDT para a especificação dos agentes. As três fases da metodologia são tratadas dentro da ferramenta e os respectivos diagramas utilizados nesta modelagem são apresentados nas Figuras 3.8(a), 3.8(b), 3.8(c), 3.9(a), 3.9(b), 3.9(c) e 3.10. O Apêndice E apresenta o detalhamento dos artefatos que constituem a metodologia Prometheus em suas três fases, a saber:

1. Especificação do sistema – responsável pelas questões do ambiente, onde o sistema é definido em termos de objetivos, funcionalidades, ações, percepções e cenários de casos de uso. Representado pelo diagrama:
 - papéis do sistema (*system roles*) (Figuras 3.8(a), 3.8(b) e 3.8(c)) – diagrama responsável por apresentar uma visão geral das principais funcionalidades do sistema.
2. Projeto arquitetural – determina os tipos de agentes e as interações entre eles. Contém os seguintes diagramas:
 - diagrama de visão geral do sistema (*system overview*) (Figuras 3.9(a), 3.9(b) e 3.9(c)) – ponto central do projeto, pois ilustra a convergência de todos os outros diagramas do sistema. Com este diagrama é possível ter uma visão geral de todo o sistema, mostrando os agentes, seus relacionamentos e interações.
3. Projeto de desenvolvimento detalhado (Figura 3.10) – última fase da metodologia, que descreve a estrutura interna de cada agente definido e como ele realiza tarefas específicas no sistema. As funcionalidades da fase de Especificação do sistema formam um conjunto de capacidades, as quais são definidas nesta fase.

Cada etapa apresentada é descrita nas próximas subseções.

3.3.1 Especificação do sistema

Com base na arquitetura definida foi desenvolvido o protótipo de uma aplicação denominada *eAgora?*. O *eAgora?* foi descrito por meio de um cenário de aplicação, requisitos do sistema e descrição básica dos papéis dos atores.

Um cenário típico (A1) para ilustrar o funcionamento da aplicação com a sequência de respostas para um evento específico é: *eAgora?* instalada na universidade e disponível para dispositivos móveis, com o objetivo de auxiliar os diversos atores presentes, como estudantes, professores, servidores administrativos e visitantes, a se localizar no ambiente e receber sugestões personalizadas de eventos que ocorrem no campus de acordo com o contexto corrente. A aplicação deve ter a capacidade de registrar padrões de comportamento pelo uso do algoritmo de propagação da ativação (SA).

Os requisitos gerais funcionais (RF) para o cenário A1 são:

- A1-RF1 – *login* do usuário, informação de identificação com um nome e senha;

- A1-RF2 – localizar usuário;
 - A1-RF2.1 – localizar usuário no ambiente externo; e
 - A1-RF2.2 – localizar usuário no ambiente interno por meio da infraestrutura de rede sem fio existente (WiFi, Bluetooth e ZigBee), usando mapeamento de intensidade sinais por redes neurais artificiais.
- A1-RF3 – coletar perfil do usuário;
- A1-RF4 – sugestões personalizadas – recomendações contextualizadas de eventos acadêmicos e culturais que ocorrem no campus;
 - A1-RF4.1 – recomendação de evento – funcionalidade do *eAgora?* que objetiva recomendar eventos culturais e acadêmicos que podem ser relevantes para o usuário no contexto corrente. O contexto definido é a entrada da função: perfil do usuário, data, hora e localização corrente dos eventos e do usuário, lista com as descrição dos eventos. Importante ressaltar que cada departamento é responsável por manter uma lista própria de eventos. No contexto coletado é aplicado o algoritmo SA para recomendações personalizadas de eventos para o usuário;
 - A1-RF4.2 – retorno (*feedback*) implícito possibilitando a interação do usuário com a aplicação mais dinâmica e transparente. Este processo enfatiza o conceito de proatividade e transparência de ambientes pervasivos; e
 - A1-RF4.3 – armazena e pontua as sugestões aceitas pelo usuário promovendo melhorias no processo de recomendação.

Os requisitos não-funcionais definidos para o cenário A1 são:

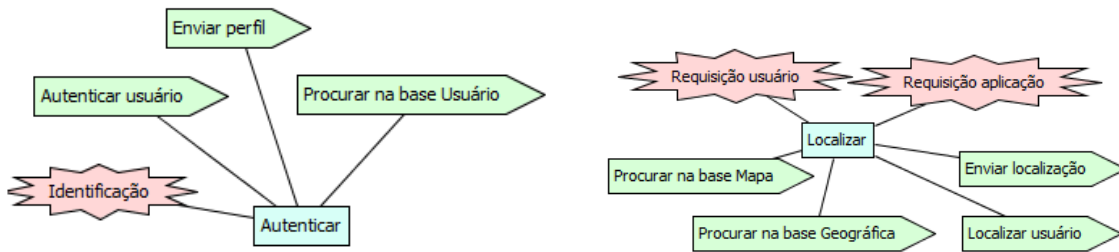
- A1-RNF1 – visão única de bases de dados distribuídas; e
- A1-RNF2 – interação com a aplicação deve ocorrer em tempo real.

Após a definição do cenário e das funcionalidades gerais, os papéis da aplicação devem ser caracterizados para refletir as funcionalidades descritas, ou seja, os atores e as interações com a aplicação são determinados em termos de percepções, ações e objetivos. Segundo Padgham and Winikoff (2004), atores podem ser qualquer pessoa ou entidade que irá interagir com o sistema.

Três papéis foram identificados, conforme ilustrado nas Figuras 3.8(a), 3.8(b) e 3.8(c).

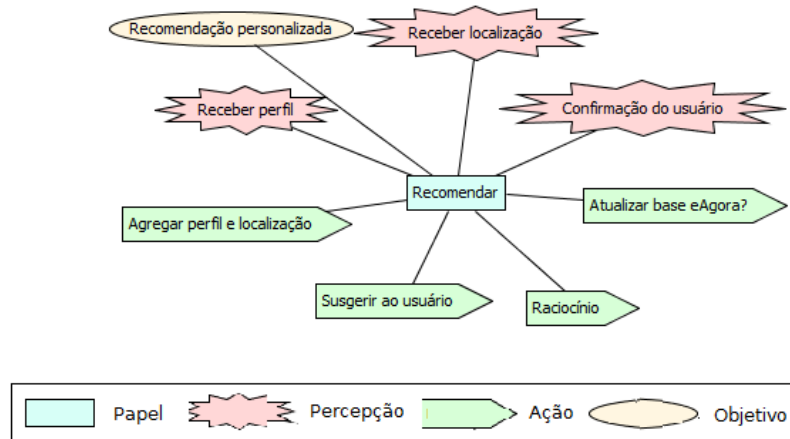
- autenticar – é responsável pela autenticação dos usuários na aplicação;
- localizar – tem como objetivo localizar os usuários no ambiente interno ou externo; e
- recomendar – responsável por alcançar o objetivo de recomendações personalizadas e para isso usa o algoritmo SA. Tem como entrada as informações dos papéis *Autenticar* e *Localizar*.

A Tabela 3.2 apresenta as entidades definidas nas Figuras 3.8(a), 3.8(b) e 3.8(c), em termos de percepções e ações.



(a) Diagrama do papel Autenticar.

(b) Diagrama do papel Localizar.



(c) Diagrama do papel Recomendar.

Figura 3.8: Diagrama de papéis do sistema (a), (b) e (c).

3.3.2 Projeto arquitetural

A ferramenta PDT utiliza como entrada os resultados produzidos na etapa de Especificação do sistema (Seção 3.3.1) para determinar os tipos de agentes que serão incluídos na aplicação e as interações entre esses agentes. As Figuras 3.9(a), 3.9(b) e 3.9(c) apresentam o diagrama detalhado dos agentes identificados.

3.3.3 Projeto de desenvolvimento detalhado

A última fase de especificação consiste no desenvolvimento da estrutura interna de cada agente e como ele realiza as tarefas no sistema. As Figuras 3.8(a), 3.8(b), 3.8(c), 3.9(a), 3.9(b) e 3.9(c), que representam as definições das fases de Especificação do sistema e Projeto arquitetural, respectivamente, serviram como base para esta etapa. A Figura 3.10 apresenta a arquitetura final proposta com os seguintes agentes:

- Agente de Localização – agente reativo que localiza o usuário no campus e envia esta informação para o Agente de Recomendação. Este agente é composto de dois módulos: (i) indoor; e (ii) outdoor, responsáveis por monitorar e coletar informações de localização nas

Tabela 3.2: Descrição das percepções e ações das entidades do sistema.

Papel	Percepção	Ação
<i>Autenticar</i>	- recebe solicitação de autenticação do usuário	- procura na base <i>Usuário</i> - autentica usuário - envia o perfil do usuário para <i>Recomendar</i>
<i>Localizar</i>	- recebe solicitação do usuário - recebe solicitação da aplicação	- procura nas bases <i>Mapa</i> e <i>Geográfica</i> - localiza o usuário no ambiente interno ou externo - envia a localização para <i>Recomendar</i>
<i>Recomendar</i>	- recebe perfil do usuário de <i>Autenticar</i> - recebe a localização do usuário de <i>Localizar</i> - recebe as sugestões aceitas pelo usuário	- agrega as informações recebidas - aplica o algoritmo SA - recomenda personalizadas eventos - atualiza a base <i>eAgora?</i>

bases *Mapa* e *Geográfica*, neste último caso por meio de uma arquitetura Web Service (WS³);

- Agente de Controle de acesso – agente reativo responsável pela autenticação do usuário. Consulta a base *Usuário* por meio de WS; e
- Agente de Recomendação – agente orientado a objetivo que infere a recomendação personalizada de acordo com o contexto. É composto de dois módulos: (i) agregação, que agrega as informações sobre o usuário (localização e perfil); e (ii) ativação, que aplica o algoritmo SA para se adaptar ao contexto corrente, recomendando eventos de modo contextualizado e personalizado.

A Figura 3.10 também apresenta todas as bases de dados da aplicação, a saber:

- base *Usuário* – informações que a UnB contém sobre estudantes, professores e servidores administrativos;
- base *eAgora?* – contém as informações de contexto dos usuários da aplicação;
- base *Mapa* – contém o mapeamento de intensidade de sinais do ambiente interno; e
- base *Geográfica* – contém informações sobre as estruturas externas de todo o campus universitário.

Um fluxo básico do processo de recomendação é ilustrado na Figura 3.11, onde o fluxo de ação é representado pela seta contínua e o retorno à recomendação do evento pela seta tracejada.

³<http://www.w3.org/TR/ws-arch/>

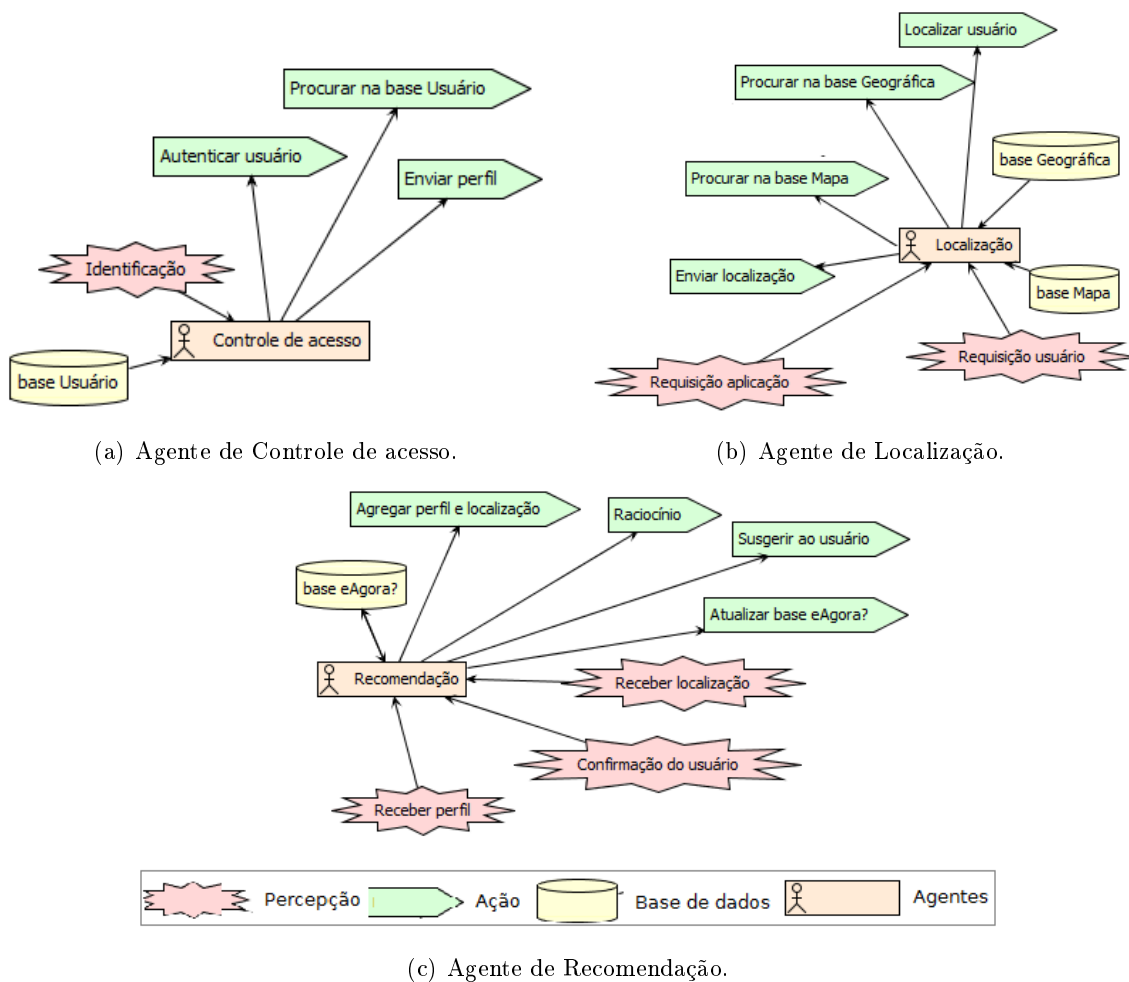


Figura 3.9: Diagrama de visão geral do sistema (a), (b) e (c).

O usuário efetua *login* na aplicação *eAgora?* e os dados de localização e informação de perfil são recuperados pelos agentes que tem acesso as bases de dados institucionais onde os usuários estão cadastrados. Após agregação dos dados do usuário, estes conceitos são os nós iniciais K da rede semântica, que serão considerados como as sementes para aplicação do algoritmo SA (vide Seção 2.5.1). Os conceitos relacionados às sementes iniciais são ativados e o ciclo de propagação se inicia. Quando a condição de parada é alcançada, uma lista ordenada de termos relacionados com K , de acordo com a relevância semântica, é gerada. A partir desta lista, os agentes buscam no departamento os eventos correspondentes e ofertam ao usuário. Quando o usuário aceita um evento recomendado, o valor da conexão desta relação é atualizado. Aumentando, deste modo, a relevância semântica dos termos relacionados ao contexto corrente.

3.4 Recomendação por propagação da ativação

O módulo de recomendação é o mais importante da arquitetura e objetiva recomendações contextualizadas e personalizadas por meio da aplicação do algoritmo SA (Seção 2.5.1). O uso do algoritmo é dividido em duas etapas:

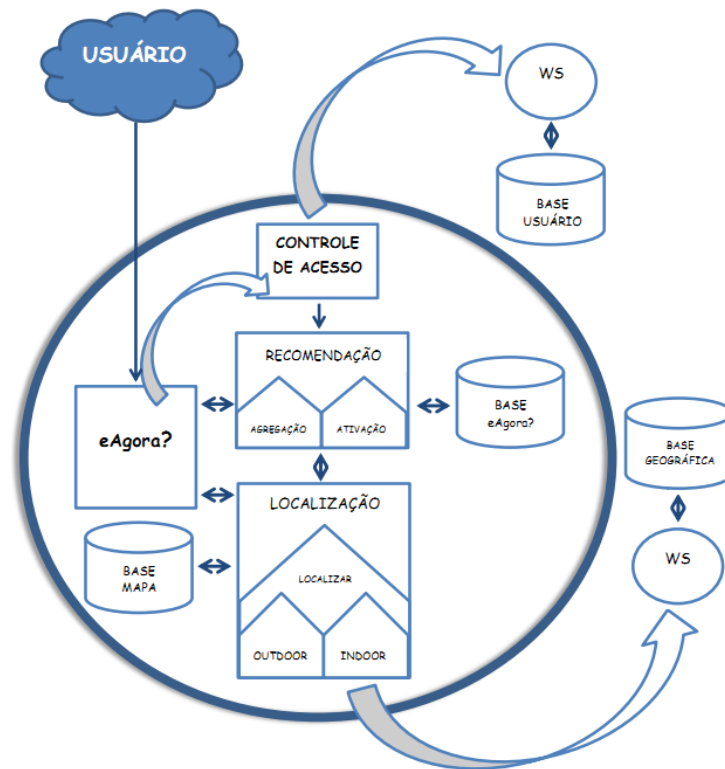


Figura 3.10: Arquitetura geral proposta baseada em agentes.

- construção do modelo de contexto para representar o domínio de conhecimento da aplicação (Seção 3.2);
- atuação do algoritmo na rede semântica definida para descobrir possíveis interesses de acordo com o contexto corrente.

Deste modo, o processo de recomendação consegue ofertar conteúdos mais relevantes para o usuário, pois os termos ativados pelo algoritmo na rede estão de acordo com o contexto e o perfil atual. Para este fim, o *framework* OntoSpread (Seção 2.5) foi utilizado.

Outras vantagens que este módulo oferece para o processo de recomendação sensível ao contexto por meio do algoritmo SA são:

- as recomendações só são ofertadas quando o usuário as solicita;
- todo o processo de recomendação é transparente ao usuário;
- o processo é proativo, pois se inicia no momento que o usuário efetua *login* na aplicação *eAgora?*; e
- o processo também pode ser considerado dinâmico porque acompanha as variações de contexto e perfil do usuário.

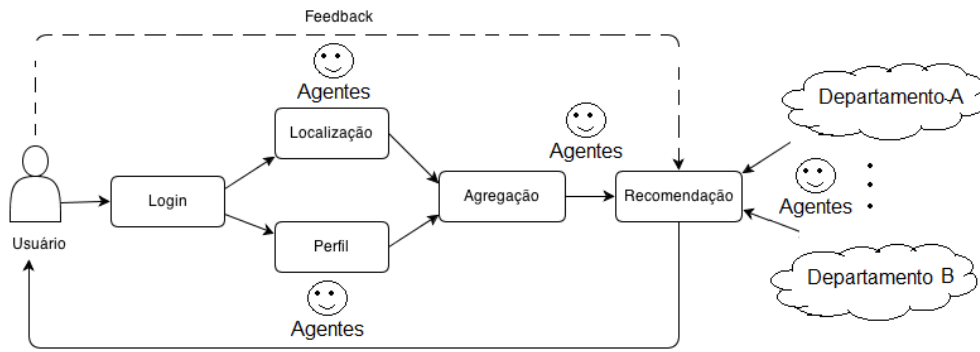


Figura 3.11: Fluxo básico para recomendação contextualizada e personalizada.

3.5 Integração da aplicação *eAgora?*

Na arquitetura proposta apresentada na Figura 3.10 são identificadas cinco questões a serem abordadas quando é considerado um ambiente real:

- Aplicação principal e *interface* – relacionado ao desenvolvimento da aplicação *eAgora?* e a *interface* de interação com o usuário;
- Controle de acesso – define o modelo de autenticação do usuário na aplicação;
- Localização – representa as técnicas a serem aplicadas para localizar o usuário no ambiente;
- Agregação e Propagação da ativação – define como as informações de contexto serão agregadas e o uso algoritmo de propagação da ativação para recomendações sensíveis ao contexto;
- e
- Integração – aborda a integração de todas as ferramentas utilizadas na aplicação.

Cada etapa apresentada é analisada a seguir.

3.5.1 Aplicação principal e *interface*

Seis classes foram definidas para a aplicação *eAgora?*, conforme Figura 3.12:

- usuário – representa o usuário;
- evento – descrição dos eventos culturais e acadêmicos que ocorrem no campus;
- área – representa as áreas de conhecimento relacionadas aos eventos;
- local – representa a localização simbólica de um evento dentro do campus;
- tipo evento – descrição do tipos de eventos que ocorrem no campus, por exemplo, uma palestra;

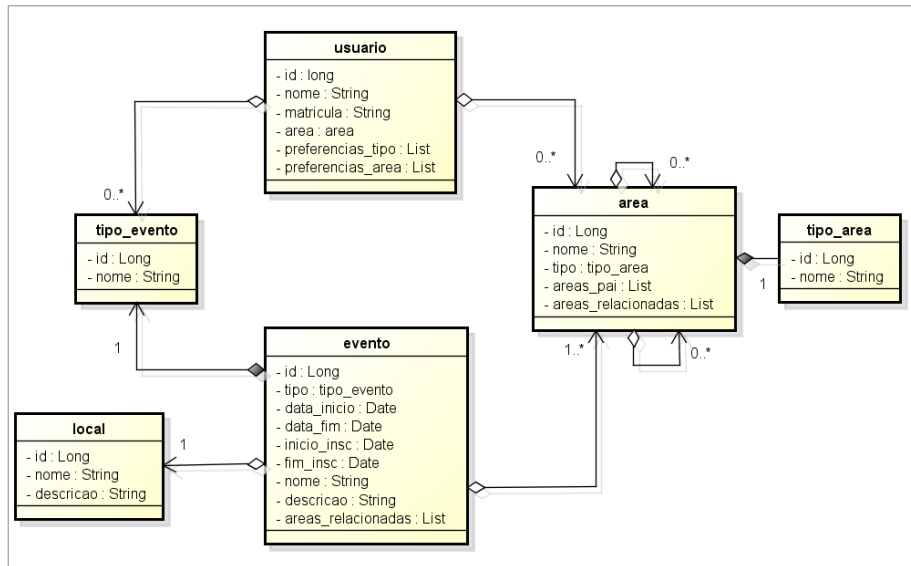


Figura 3.12: Modelo de classes para aplicação *eAgora?*.

- tipo área – descrição dos tipos das áreas de conhecimento.

A modelagem proposta permite o armazenamento dos dados relativos aos usuários da aplicação e das preferências de cada um deles, além da localização descoberta pelo *eAgora?*.

3.5.2 Controle de acesso

Esta etapa requer análises sobre os modos disponíveis na UnB. Optou-se utilizar o componente de autenticação integrado ao *framework JBoss Seam*, que fornece uma API para abstrair todo o desenvolvimento de uma camada de autenticação. Ao implementar um método responsável pelo *login* do usuário, os dados de contexto relacionados à identidade do usuário são automaticamente disponibilizados na camada de negócio da aplicação, o que facilita o desenvolvimento de regras relacionadas aos perfis de usuário.

3.5.3 Localização

A localização do usuário é a primeira etapa a ser realizada em um aplicação sensível ao contexto, a qual deve ser feita de modo dinâmico e inteligente. A localização em um ambiente interno foi descrita em Fonseca et al. (2011) e Neves et al. (2013c), onde três diferentes protocolos de redes sem fio, WiFi (IEEE 802.11), Bluetooth (IEEE 802.15.1) e ZigBee (IEEE 802.15.4), foram testados e seus resultados comparados para localização nos laboratórios da UnB, usando mapeamento de intensidade de sinais por RNAs e baseado em agentes. Deste modo, obteve-se um modo dinâmico e inteligente para localização interna. Este módulo ainda não está agregado à aplicação *eAgora?* e, por isso, o estudo de caso é apresentado apenas para localizações externas na universidade.

A Figura 3.13 apresenta as classes definidas para o módulo de localização externa. Duas entida-

des são representadas no domínio: (i) Local, pode representar uma sala, anfiteatro ou auditório; e (ii) Estrutura, pode representar um departamento ou instituto. A partir desta definição, é possível associar eventos acadêmicos e culturais às localizações dentro da universidade.

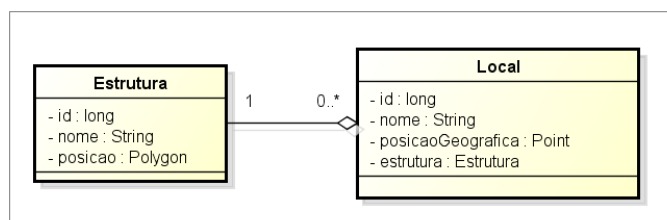


Figura 3.13: Modelo de classes para o módulo de localização externa.

Um cenário de interação do usuário e o agente de localização é exibido na Figura 3.14 por meio de um diagrama de sequência, onde podem ser identificadas três etapas:

- obter a localização geográfica (externa) do usuário;
- buscar as estruturas dentro de um raio definido que seja próximo ao usuário; e
- buscar eventos que ocorram no local destas estruturas identificadas.

A sequência da interação ilustrada na Figura 3.14 é: (1) o usuário efetua *login* na aplicação; (1.1) o controlador principal busca a posição geográfica deste usuário; (1.1.1) as estruturas que estão definidas nesta posição são retornadas para o usuário exibindo a localização externa na qual se encontra; (2) neste diagrama é o usuário que solicita os eventos; e (2.1) os eventos a serem ofertados são filtrados de acordo com o contexto corrente e uma lista de eventos baseado nesse contexto é recomendada ao usuário.

A priori, a localização geográfica do usuário é baseada na API *Geolocation*⁴ especificada pelo W3C, permitindo a apresentação da localização na tela do dispositivo móvel integrada com a interface do *Google Maps*⁵.

3.5.4 Agregação e Propagação da ativação

Esta etapa envolve o uso do modelo de contexto na linguagem OWL, isto é, da *OntoUnB*, como descrito na Seção 3.2, e do *framework* *OntoSpread*.

3.5.5 Integração

Entende-se que a recomendação contextualizada em uma Universidade Pervasiva é um problema que requer uma abordagem distribuída. Neste sentido, é proposta uma arquitetura onde os departamentos são responsáveis por publicar os eventos relacionados ao seu contexto. Sendo que

⁴<http://dev.w3.org/geo/api/spec-source.html>

⁵<https://maps.google.com.br/>

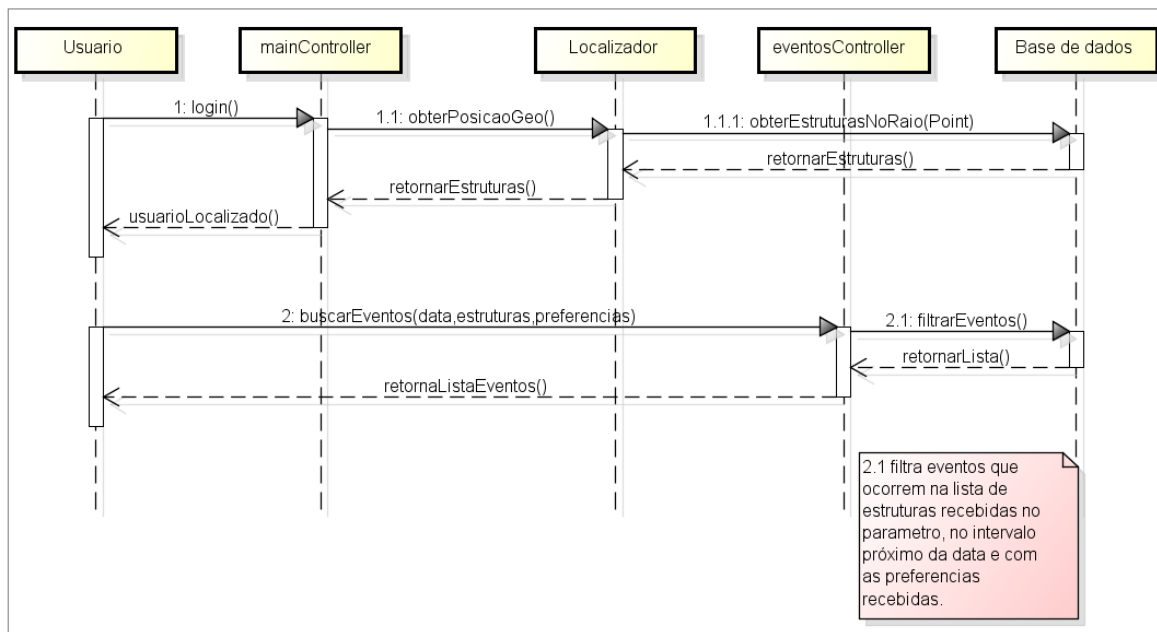


Figura 3.14: Diagrama de sequência para localização externa.

é de interesse de cada departamento manter um registro de controle sobre os eventos que estão relacionados. Considerando este caso, o uso da aplicação *eAgora?* também é justificado para a universidade, sendo um aplicação que pode buscar informações em outros sistemas, aumentando a eficiência e distribuindo as responsabilidades das entidades envolvidas.

O protótipo tem como requisito o acesso por dispositivos móveis e, por isso, optou-se por desenvolver uma aplicação Web adaptada para ser acessada por esses dispositivos. Esta decisão foi tomada para diminuir o número de tecnologias a serem integradas em uma solução completa. Foi desenvolvido na plataforma *Java Enterprise Edition*⁶ (JavaEE) versão 6, que oferece uma API e ambiente de execução para a programação de aplicações de grande porte, distribuídas, multicamada, escaláveis e confiáveis. O servidor *JBoss AS 7*⁷ foi escolhido para a execução da aplicação.

Além disso, foi utilizada a ferramenta *Apache Maven*⁸, que tem um arquivo XML (POM) para descrever o projeto de software, as dependências sobre os módulos e os componentes externos, a ordem de compilação, diretórios e *plugins* necessários. A Figura 3.15 apresenta o diagrama de componentes do *eAgora?*, que contém além dos componentes, as outras entidades no ambiente da UnB para implantação do projeto.

Baseada na arquitetura apresentada na Figura 3.15, agentes com capacidade cognitiva e reativa buscam as informações distribuídas no ambiente e aplicam mecanismos de inferência para tratá-las e recomendar eventos contextualizados ao usuários. O SMA foi desenvolvido com o uso do

⁶<http://www.oracle.com/technetwork/java/javasee/overview/index.html>

⁷<http://www.jboss.org/overview/>

⁸<http://maven.apache.org/>

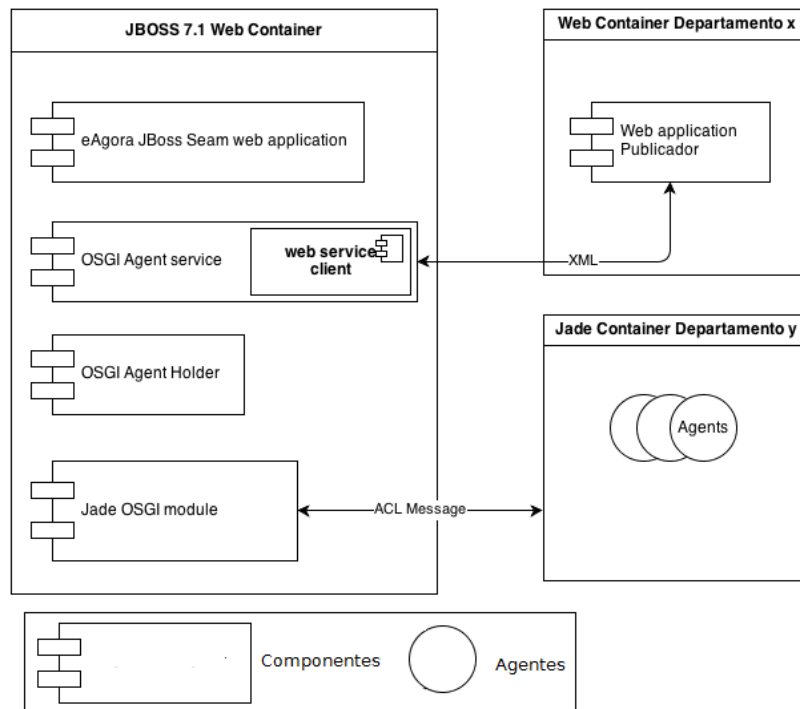


Figura 3.15: Arquitetura geral de implementação.

framework Java Agent DEvelopment versão 4.3.0 (JADE⁹).

Os agentes integrados ao sistema Web podem se comunicar com plataformas de agentes de outras aplicações ou consumir WS. Deste modo, a integração de uma plataforma Web com o JADE é feita por meio da tecnologia *Open Services Gateway Initiative* (OSGi¹⁰), que permite a instalação, remoção e atualização de componentes sem precisar reiniciar a aplicação. Para diminuir o acoplamento e permitir que esses componentes se comportem de maneira dinâmica, a plataforma foi projetada utilizando uma arquitetura orientada a serviços ou *Service Oriented Architecture* (SOA) (Bianco et al., 2007). Desse modo, os componentes podem publicar e descobrir serviços prestados por outros por intermédio da plataforma em tempo de execução. Esses componentes são chamados de *bundles*. A Figura 3.16 apresenta o diagrama integrado de todos os módulos identificados no ambiente.

O Capítulo 4 apresenta dois cenários de aplicação para prova de conceito da arquitetura proposta com o uso do aplicativo *eAgora*?

⁹<http://jade.tilab.com/>

¹⁰<http://www.osgi.org/Main/HomePage>

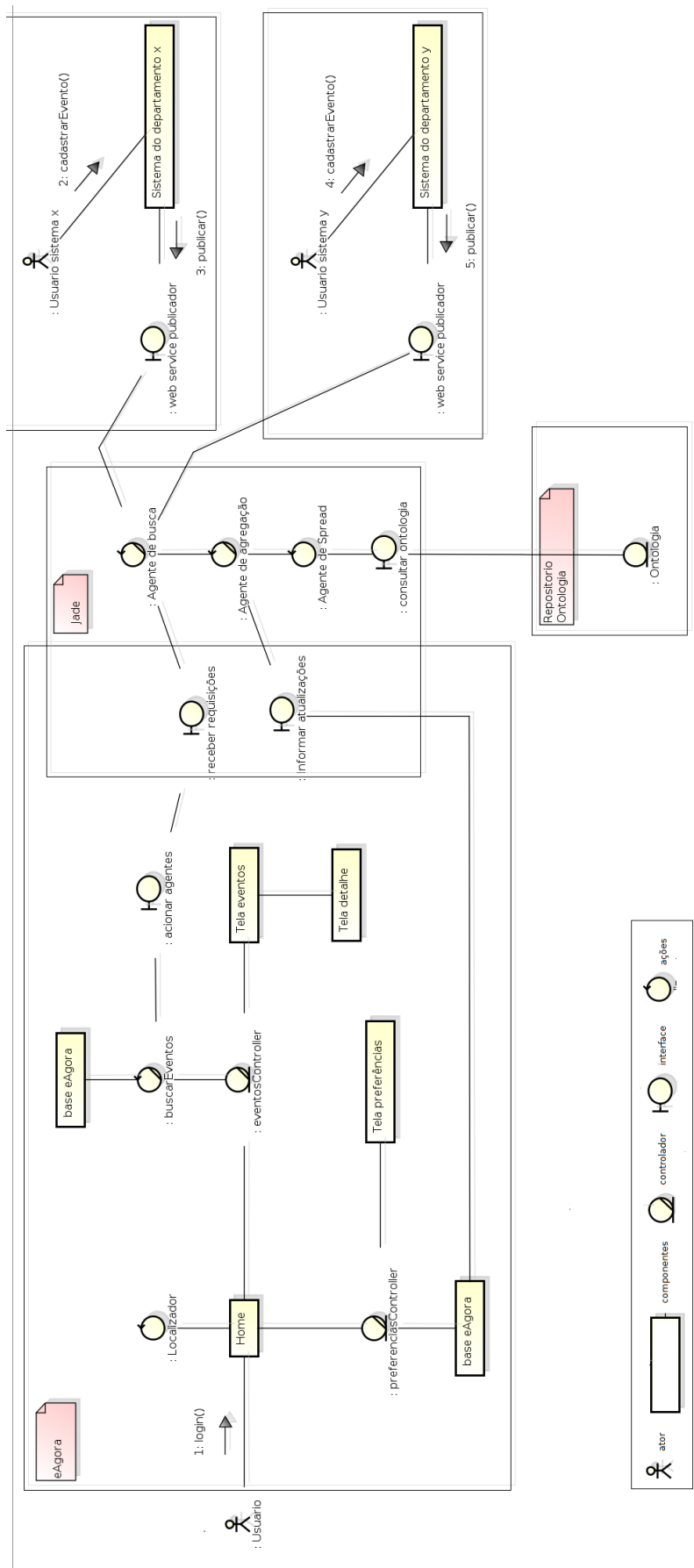


Figura 3.16: Diagrama de comunicação geral.

Capítulo 4

Prova de conceito

Este capítulo apresenta os cenários para a prova de conceito arquitetural por meio da aplicação eAgora? para recomendação sensível ao contexto de eventos no domínio da Universidade Pervasiva.

4.1 Cenários

Após definição e implementação da arquitetura, como descrito na Seção 3, foi realizada a prova de conceito da arquitetura proposta com a aplicação *eAgora?*. A prova envolve dois cenários diferentes com estudantes do curso de Bacharelado em Ciência da Computação da UnB. O estudante identificado como X está cursando o segundo semestre do curso, enquanto o estudante identificado como Y está no sétimo semestre.

Além do contexto corrente, que contém a localização e intervalo temporal, o *eAgora?* agrega o perfil básico gerado e mantido pelo sistema da UnB no momento da matrícula do aluno. O perfil do usuário é composto de: (i) curso; (ii) período corrente; e (iii) matérias cursadas. Este perfil foi definido apenas para a prova de conceito e pode ser modificado com o acréscimo de novas características, como estilo musical, preferências literárias e gêneros de filmes.

Para ilustrar os cenários testados, o perfil do estudante X inclui:

- curso – Bacharelado em Ciência da Computação;
- período – segundo semestre; e
- disciplinas cursadas –
 - Cálculo 2;
 - Probabilidade e Estatística;
 - Estrutura de dados;
 - Física 2; e
 - Física 2 Experimental.

Para o estudante Y:

- curso – Bacharelado em Ciência da Computação;
- período – sétimo semestre; e
- disciplinas cursadas –
 - Introdução a Teoria dos Grafos;
 - Tradutores;
 - Sistemas operacionais;
 - Análise e Projeto de Sistemas;
 - Modelagem orientada a objetos;
 - Projeto análise de algoritmos; e
 - Teleinformática e Redes 2.

O cenário I é aplicado ao aluno X, o qual foi localizado pela aplicação próximo ao Departamento de Física. O cenário II é aplicado ao estudante Y, que está localizado no Departamento de Ciência da Computação.

4.1.1 Ontologia de aplicação

Parte da ontologia de domínio definida, OntoUnB (Seção 3.2), foi instanciada criando-se a ontologia de aplicação. Esta ontologia está de acordo com a grade curricular da UnB para o curso de Bacharelado em Ciência da Computação e se relaciona com os seguintes departamentos e institutos (Figura 4.1):

- Matemática (MAT);
- Estatística (EST);
- Engenharia Elétrica (ENE);
- Línguas estrangeiras e Tradução (LET);
- Linguística, Português e Línguas Clássicas (LIP); e
- Instituto de Física (IF).

Esta relação semântica foi definida porque eventos acadêmicos e culturais em uma universidade estão ligados às áreas de conhecimento. Para a definição destas áreas, foi utilizada a classificação disponível na Capes¹, uma vez que trata-se de uma classificação amplamente aplicada no âmbito acadêmico no país. A rede semântica construída para aplicação do algoritmo SA contém setenta e sete conceitos.

¹<http://www.capes.gov.br/avaliacao/tabela-de-areas-de-conhecimento>

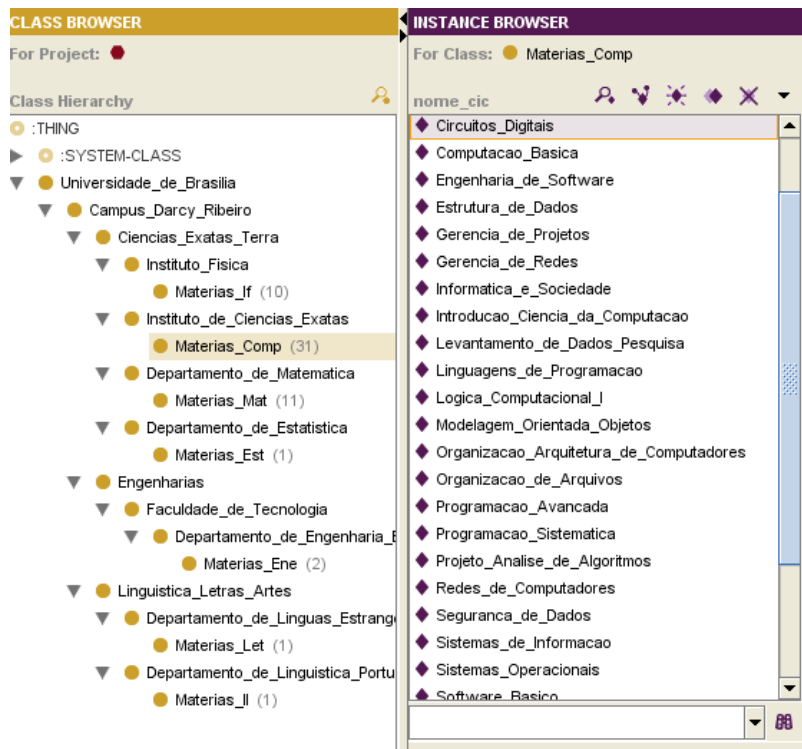


Figura 4.1: Ontologia de aplicação.

4.1.2 Recomendação

O módulo de recomendação é o mais importante da arquitetura, como descrito na Seção 3.4. Assim, para os cenários apresentados na etapa anterior são ofertados eventos acadêmicos e culturais que acontecem no campus, de acordo com as informações da Secretaria de Comunicação (SeCom²) da UnB. Para este fim, uma outra aplicação Web foi desenvolvida, onde os eventos são cadastrados e mantidos (Figura 4.2), e simula os serviços disponíveis de outros sistemas que os agentes irão consumir.

Após a realização do *login* na aplicação *eAgora?*, os estudantes X e Y são localizados no campus. As informações contextuais, localização, intervalo temporal e perfil, são agregadas e o algoritmo SA é aplicado para gerar recomendações de eventos acadêmicos relacionadas ao contexto corrente e perfil.

A Tabela 4.1 exhibe as configurações do algoritmo SA aplicadas nos dois cenários apresentados (Seção 2.5.1), que são:

- o valor mínimo da ativação corresponde ao *threshold* – determina o valor mínimo para um nó ser ativado;
- número máximo de conceitos propagados (M) – o processo de ativação e propagação será executado, na maioria, até M conceitos terem sido propagados;

²<http://www.secom.umb.br/>

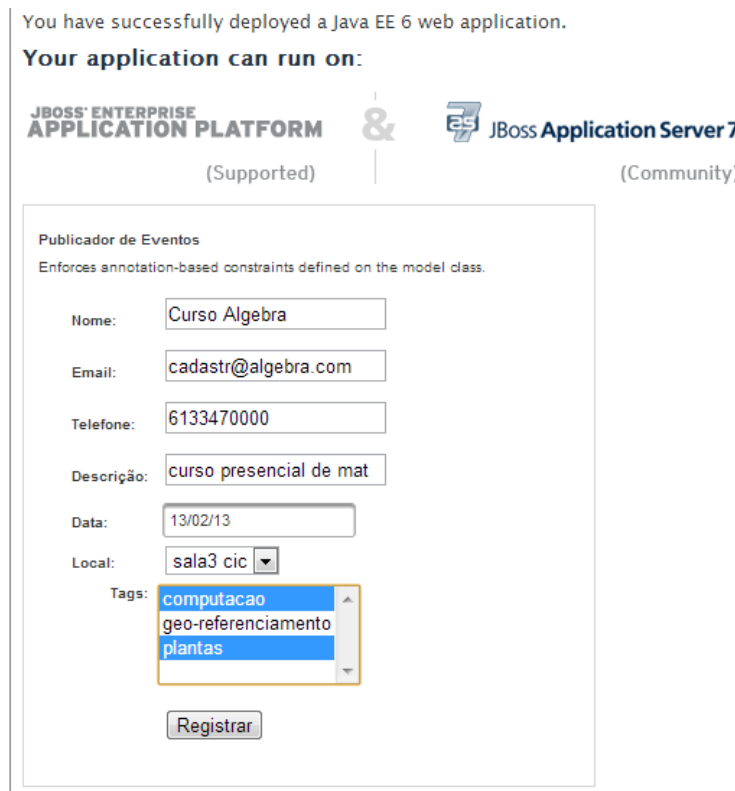


Figura 4.2: Tela do protótipo de cadastro e publicação de eventos.

- número mínimo de conceitos propagados (M_{min}) – o processo de ativação e propagação será executado, em pelo menos, M_{min} conceitos terem sido propagados;
- pesos entre conceitos – corresponde ao peso da relação entre os nodos; e
- fator de decaída – responsável por diminuir a força da propagação à medida que se afasta do nó inicial, conforme a Equação 2.3.

Esses valores foram mantidos constantes porque o interesse deste trabalho é analisar as sugestões ofertadas pelo algoritmo SA conforme as mudanças de contexto e perfil do usuário no ambiente, e não o desempenho do algoritmo com diferentes configurações. No entanto, sugere-se que em trabalhos futuros os parâmetros do algoritmo de SA sejam validados em testes mais extensos para avaliação do desempenho do algoritmo em estudos de casos com bases reais da UnB.

4.1.3 Testes do protótipo

Os testes foram executados com *browser Android*, celular *Samsung Galaxy Ace* com sistema operacional *Android 2.3.6*. A Figura 4.3 apresenta a tela de autenticação do protótipo para o aluno X com matrícula 0678325.

Após a autenticação, o estudante pode ser localizado no campus (Figura 4.4(a)) e interagir com a aplicação *eAgora?* por meio do menu de opções (Figura 4.4(b)).

Tabela 4.1: Configurações do algoritmo de propagação da ativação.

Configuração	Cenário I	Cenário II
Valor mínimo de ativação	1.0	1.0
Número máximo de conceitos propagados	3	3
Número mínimo de conceitos propagados	3	3
Pesos entre conceitos	1.0	1.0
Fator de decaída	h1	h1

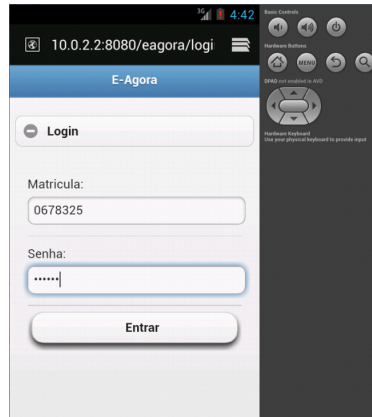


Figura 4.3: Tela de *login* do protótipo *eAgora*?

As Figuras 4.5(a) e 4.5(b) representam os perfis dos alunos X e Y, respectivamente, e as listas de sugestões geradas após aplicação do algoritmo SA. Essas listas contêm as áreas que apresentam eventos relacionados ao perfil e contexto corrente dos estudantes.

O protótipo Web desenvolvido também apresenta uma interface que simula a descrição dos eventos próximos recomendados, conforme as Figuras 4.6(a) e 4.6(b). *Tags* semanticamente relacionadas são utilizadas para oferta contextualizada de eventos próximos (físico e temporal) de acordo com o perfil do usuário. Neste caso, cada evento está relacionado com um ou mais assuntos nomeados como *tags*.

Além disso, o estudante pode verificar detalhes sobre os eventos. Informações, como categoria, data e hora do evento, são exibidas para o usuário, conforme as Figuras 4.7(a) e 4.7(b).

4.2 Discussão dos resultados

A Tabela 4.2 apresenta algumas estatísticas extraídas referentes aos resultados ilustrados nas Figuras 4.5(a) e 4.5(b). A principal diferença apresentada na Tabela 4.2 é referente aos números de conceitos ativados. Isso ocorre porque a rede semântica do estudante Y apresenta mais termos, que já foram ativados com o passar do ciclo de vida estudantil, do que a rede do estudante X, que ainda está no início da sua vida acadêmica. Deste modo, o algoritmo consegue acompanhar as mudanças de contexto que ocorrem, de modo transparente, definindo dinamicamente a importância de cada

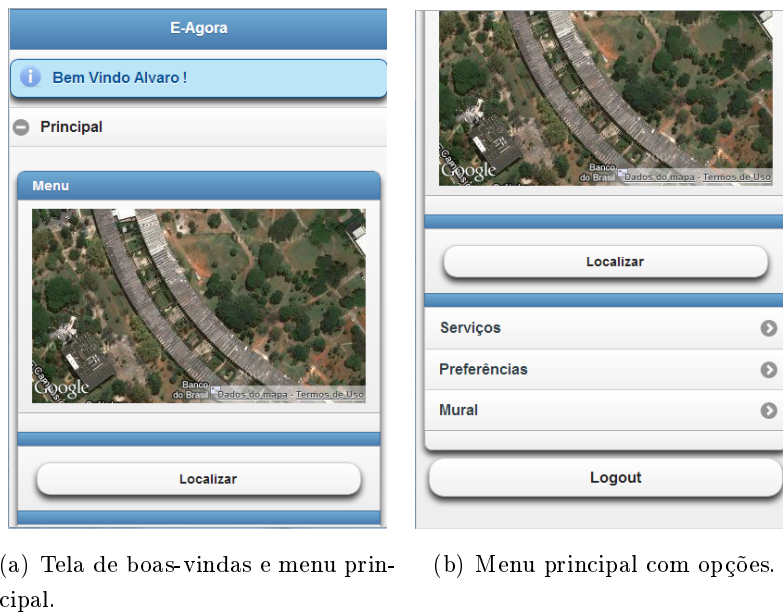


Figura 4.4: Telas iniciais do protótipo *eAgora*? (a) e (b).

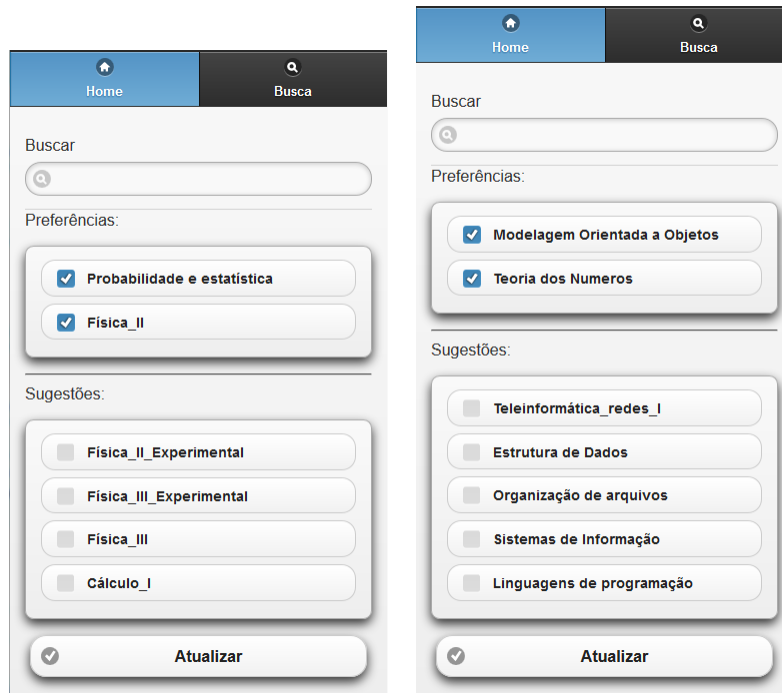
Tabela 4.2: Estatísticas dos resultados sobre a ontologia de aplicação nos cenários I e II.

Contexto de ativação	Cenário I	Cenário II
Nós ativados	14	34
Nós propagados	3	3
Maior valor de ativação	8	8
Profundidade do caminho propagado	2	2
Conceitos (name:value)	#Materias_If:8.0, #Fisica_II_Experimental:2.0	#Materias_Comp:8.0, #Teleinformatica_Redes_I:2.0

conceito para um contexto. Essa transparência ocorre, principalmente, porque o algoritmo faz esse mapeamento sem que seja necessária a intervenção direta do usuário.

Uma característica importante a ser ressaltada é a atualização da relevância semântica entre os conceitos, a qual pode ser enfraquecida ou reforçada com o passar do tempo, por exemplo, termos que não são mais utilizados e, por isso, não apresentam alterações de pesos, não são mais recomendados. Agregando esta característica com a capacidade de adaptação dinâmica, o algoritmo consegue recomendar conteúdo mais relevante e útil conforme a situação corrente.

Dos quatorze conceitos ativados na rede do estudante X no cenário I, treze são ilustrados em verde na Figura 4.8. O conceito que não aparece refere-se à *Materias_If*, que corresponde ao conceito onde estão contidas todas as matérias ofertadas pelo Instituto de Física, conforme grade curricular da UnB. Esta rede representa parte dos setenta e sete conceitos definidos na rede semântica, conforme a classificação da Capes para as áreas de conhecimento (Seção 4.1.1), que são: Ciências Exatas e da Terra, Ciências Biológicas, Engenharias, Ciências da Saúde, Ciências Agrárias, Ciências Sociais Aplicadas, Ciências Humanas, Linguística, Letras e Artes e Multidisciplinar.



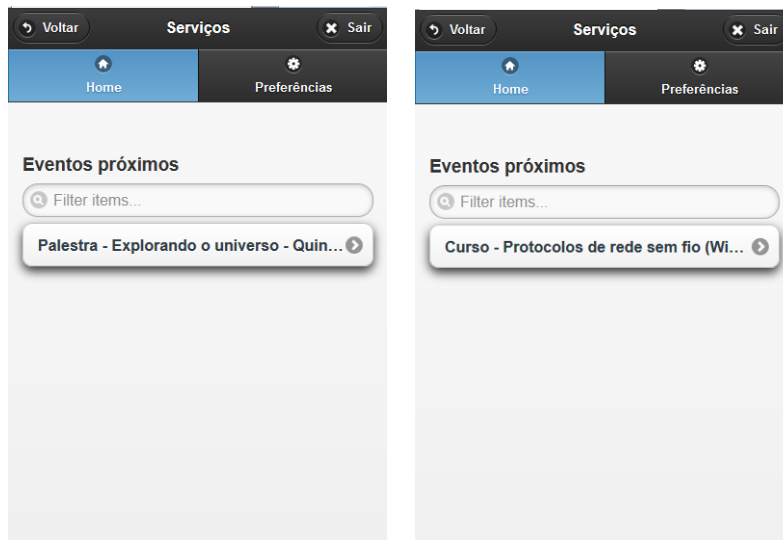
(a) Tela de sugestões para o estudante X no cenário I. (b) Tela de sugestões para o estudante Y no cenário II.

Figura 4.5: Telas de recomendações do protótipo *eAgora?* (a) e (b).

Os conceitos ativados na Figura 4.8 correspondem à lista de relevância semântica (Apêndice F) gerada após execução do algoritmo SA para o cenário I. O Apêndice G contém a lista de relevância semântica para o cenário II. A partir dessa lista, os agentes buscam os eventos relacionados. Os conceitos que tiverem eventos disponíveis serão recomendados ao usuário, conforme Figura 4.5(a).

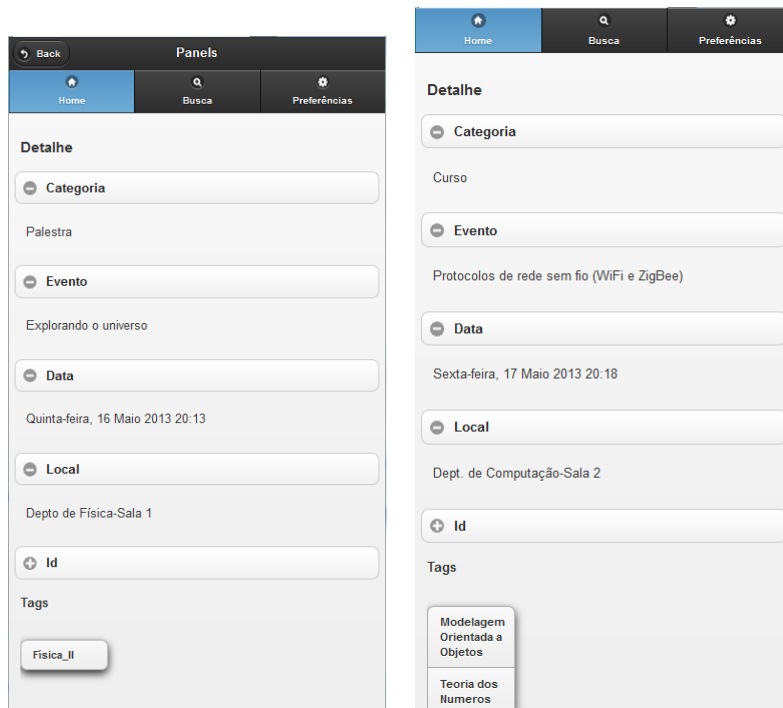
A Tabela 4.2 também apresenta o número de nós propagados igual a três e corresponde ao número máximo de conceitos que foram propagados, como definido nas configurações (Tabela 4.1). Além disso, demonstra que após a atualização dos pesos, o maior valor de ativação alcançado foi oito para o conceito *Materias_If*. Outra informação presente na tabela é a restrição imposta de distância (Seção 2.5.1), ou seja, a definição da quantidade de níveis que serão ativados. Nos cenários I e II, a profundidade do caminho propagado foi de dois níveis na rede. A condição de parada para o algoritmo foi definida como sendo a restrição de distância.

Os Apêndices H e I apresentam os *logs* do algoritmo SA para os cenários I e II, respectivamente. Correspondem às três fases do algoritmo, conforme apresentado na Seção 2.5.1, que são: pré-ajuste (*pre-adjustment*), propagação (*spreading*) e pós-ajuste (*post-adjustment*). Deste modo, indicam os conceitos que foram inicializados, ativados e propagados; com as respectivas atualizações de pesos e nível de profundidade da rede.



(a) Tela de eventos para a opção Física II Experimental para o estudante X no cenário I. (b) Tela de eventos para a opção Teoria da Informática e Redes I para o estudante Y no cenário II.

Figura 4.6: Telas dos eventos próximos recomendados pelo *eAgora?* (a) e (b).



(a) Tela de apresentação dos detalhes do evento selecionado para o estudante X no cenário I. (b) Tela de apresentação dos detalhes do evento selecionado para o estudante Y no cenário II.

Figura 4.7: Telas de apresentações dos eventos próximos recomendados pelo *eAgora?* (a) e (b).

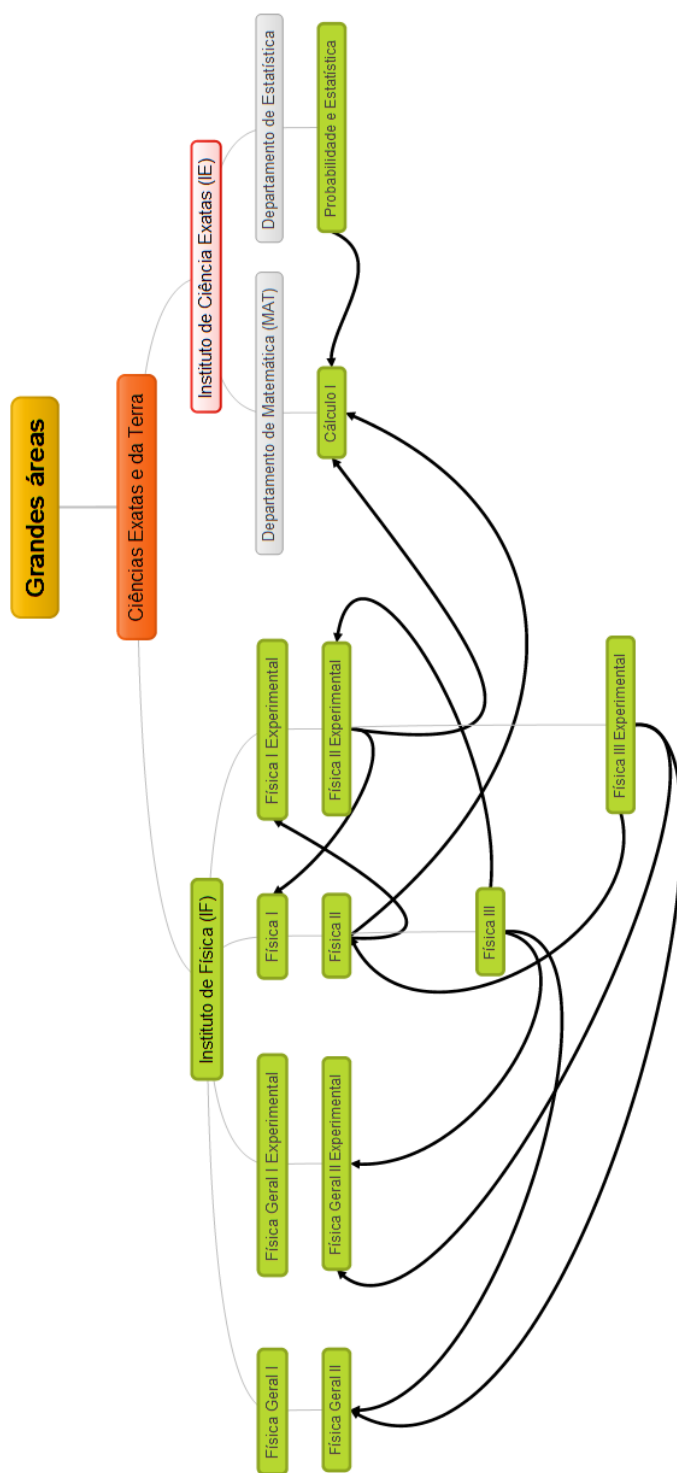


Figura 4.8: Parte dos conceitos ativados no cenário I.

Capítulo 5

Conclusões

A recomendação sensível ao contexto no domínio pervasivo não é um problema trivial, pois envolve muitos aspectos relacionados à dinamicidade do ambiente, à mobilidade dos usuários e aos limites tecnológicos, a saber:

- suportar as diversas fontes de serviços espalhadas pelo campus;
- lidar com a mobilidade do usuário e as constantes mudanças de contexto;
- prover meios para uma comunicação homogênea entre as diferentes entidades no ambiente;
- suportar a heterogeneidade do ambiente;
- disponibilizar meios que auxiliem estudantes, professores e servidores administrativos em suas atividades; e
- dispor meios que filtrem de modo contextualizado itens relevantes e que sejam entregues em tempo hábil para a tomada de decisão.

Conforme apresentado no Capítulo 2, as abordagens tradicionais de recomendação no domínio da Universidade Pervasiva apresentam algumas restrições:

- o modelo do perfil do usuário é gerado por preenchimento de questionários e não prevê as mudanças de preferências de acordo com a dinamicidade do contexto;
- requisição ativa do usuário pelo sistema de recomendação para avaliação de itens recomendados ou por construção manual de perfil;
- uso de avaliações quantitativas ou descrições sintáticas podendo gerar ambiguidades na recomendação; e
- ausência de um método de adaptação dinâmica do contexto para tornar a interação usuário-sistema mais transparente e menos intrusiva.

Conforme exposto, esta pesquisa tem como objetivo geral propor uma arquitetura baseada em agentes para a extração incremental de padrões comportamentais do usuário e oferta de informações contextualizadas, de modo que o processo de recomendação seja personalizado, proativo, transparente e dinâmico.

Como prova de conceito para a proposta arquitetural foi desenvolvida a aplicação *eAgora?*, que utiliza o algoritmo SA para acompanhar as mudanças de contexto e perfil do usuário, já que modelos estáticos de recomendação não conseguem mapear estas modificações. Com isso, recomendações mais relevantes são geradas para o usuário, pois refletem o contexto corrente. Além disso, as interações com o sistema de recomendação tornam-se mais transparentes e menos intrusivas porque com o uso do algoritmo são evitadas as constantes solicitações para a construção manual do perfil, por exemplo.

A abordagem proposta ainda permite a exploração de características intrínsecas aos agentes, como habilidade social, autonomia de interação e atuação, de acordo com os objetivos definidos no contexto da Universidade Pervasiva. Neste sentido, a abordagem orientada a agentes permite a definição de indivíduos, como elementos dentro de uma sociedade, a própria sociedade, formada por todos os seus elementos, o ambiente pervasivo, representando também o comportamento autônomo dos agentes que se adaptam às mudanças ocorridas no ambiente de modo dinâmico. Além da modularização da solução dentro do ambiente pervasivo, a qual permite o tratamento de ambientes complexos inerentes ao mundo real.

Para tratar a heterogeneidade de ambientes pervasivos, ontologias foram utilizadas para a construção da OntoUnB no contexto da Universidade Pervasiva. Provendo um vocabulário formalizado e comum entre as diversas entidades presentes, o que permite a interoperabilidade semântica e a diminuição da ambiguidade no processo de recomendação.

A contribuição principal deste trabalho é o desenvolvimento de uma arquitetura modularizada, que permite a definição de comportamentos autônomos e cognitivos para recomendações sensíveis ao contexto no âmbito da Universidade Pervasiva. Esta arquitetura pode se adaptar às mudanças de contexto ocorridas no ambiente, o que gera recomendações relevantes, contextualizadas e personalizadas para o usuário. Ressaltando ainda a interação transparente usuário-aplicação, um dos objetivos da Computação Pervasiva.

Em visitas realizadas nas universidades de Londres (Universidade de Essex e *College London*), de Bordeaux e Coimbra, percebeu-se o interesse crescente na área de Universidade Pervasiva, com a formação de grupos ativos de pesquisa que aplicam na própria universidade os projetos desenvolvidos, objetivando a melhoria da estrutura tradicional de ensino. Esta área é promissora, com pesquisas importantes porque adequam as universidades a uma tendência mundial de modernização da estrutura de ensino e aos novos paradigmas tecnológicos, como Computação Pervasiva e Computação sensível ao contexto.

A autora observou que a Universidade Pervasiva é considerada como a evolução da estrutura tradicional de ensino superior, por isso, o interesse da continuidade dos estudos nessa área. Neste sentido, a Universidade de Melbourne (UoM), na Austrália, está com chamada aberta para pós-doutorado no projeto "*Living Campus*", que visa transformar o campus da UoM em um ambiente

de testes para pesquisas com participação do cidadão, engajamento da comunidade, redes sociais, dinâmica populacional, entre outros. É um projeto que conta com a colaboração da IBM *Research*.

A Universidade Pervasiva é muito mais do que um problema tecnológico, pois representa uma mudança de paradigma no ensino superior, onde a preocupação principal está na aprendizagem com qualidade e na viabilização do bem estar de todos os atores envolvidos. Espera-se que este trabalho de pesquisa tenha viabilizado a reflexão de pontos importantes neste cenário. Principalmente, para a importância do papel das competências, pois ele não se restringe apenas aos aspectos de passagem de conteúdos disciplinares, mas é a universidade integrada com a participação ativa e consciente de todos. É certo que este é o caminho para ser alcançada a educação pervasiva no mundo globalizado e inteligente.

5.1 Trabalhos Futuros

Este trabalho deixa muitas questões em aberto para prospecção futura. Portanto, para não ser exaustivo, seguem algumas sugestões:

- integrar o código desenvolvido para o módulo de Localização interna e externa com a aplicação *eAgora?*;
- implantar a aplicação *eAgora?* na UnB, inclusive analisando os sistemas de segurança utilizados pela universidade para compor o módulo de Controle de acesso da aplicação;
- integrar a aplicação *eAgora?* com as redes sociais viabilizando um rede social interna à UnB, onde os atores possam compartilhar informações de acordo com categorias de anúncio, por exemplo, compra e venda;
- integrar a aplicação com os bancos de dados institucionais e outros sistemas legados (e.g., SIGRA, SIPPOS, SIPES), além da plataforma Lattes do CNPq; e
- fazer um levantamento sobre o desenvolvimento de sistemas e aplicativos construídos por meio de Computação em nuvem. Considerando a automatização da universidade na direção da Universidade Pervasiva.

Referências Bibliográficas

- G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput. Hum. Interact.*, 7(1):29–58, 2000. ISSN 1073-0516.
- G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, 1999. Springer-Verlag. ISBN 3-540-66550-1.
- G. Adomavicius and A. Tuzhilin. Personalization technologies: a process-oriented perspective. *Commun. ACM*, 48(10):83–90, 2005. ISSN 0001-0782.
- G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-093-7.
- H. Al Tair, M. J. Zemerly, M. Al-Qutayri, and M. Leida. Architecture for context-aware pro-active recommender system. *International Journal Multimedia and Image Processing (IJMIP)*, 2(1/2): 15–133, march/june 2012.
- B. Al Takrouri, A. Canonico, L. Gongora, M. Janiszewski, C. Toader, and A. Schrader. eyejot - a ubiquitous context-aware campus information system. In *Proceedings of the 2nd International Conference on Pervasive Computing and Applications*, ICPCA 2007, pages 122–127, 2007. doi: 10.1109/ICPCA.2007.4365424.
- M. Almeida and M. Bax. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Revista Ciência da Informação*, 32:7–20, 2003.
- J. M. Alvarez, L. Polo, W. Jimenez, P. Abella, and J. E. Labra. Application of the spreading activation technique for recommending concepts of well-known ontologies in medical systems. In *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '11, pages 626–635, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0796-3.
- J. R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295, 1983.
- S. Arbanowski, P. Ballon, K. David, O. Droegehorn, H. Eertink, W. Kellerer, H. van Kranenburg, K. Raatikainen, and R. Popescu-Zeletin. I-centric communications: personalization, ambient

- awareness, and adaptability for future mobile services. *Communications Magazine, IEEE*, 42(9):63–69, 2004. ISSN 0163-6804.
- N. Y. Asabere. Towards a viewpoint of context-aware recommender systems (cars) and services. *International Journal of Computer Science and Telecommunications*, 4(1):10–29, january 2013.
- A. Baca, P. Kornfeind, E. Preuschl, S. Bichler, M. Tampier, and H. Novatchkov. A server-based mobile coaching system. *Sensors*, 10(12):10640–10662, 2010. ISSN 1424-8220.
- M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the Association for Computing Machinery*, 40(3):66–72, march 1997. ISSN 0001-0782.
- M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.
- L. Baltrunas. *Context-Aware Collaborative Filtering Recommender Systems*. PhD thesis, Faculty of Computer Science, Free University of Bozen-Bolzano (FUB), Italy, april 2011.
- D. Beckett. Rdf/xml syntax specification (revised)– w3c recommendation, agosto 2004. Acessado em 22/08/2011, <http://www.w3.org/TR/REC-rdf-syntax/>.
- F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 1st edition, 2007. ISBN 978-0470057476.
- P. Bianco, R. Kotermanski, and P. Merson. Evaluating a service-oriented architecture. Technical report, Carnegie Mellon University, 2007. Acessado em 10/02/2013, <http://www.sei.cmu.edu/reports/07tr015.pdf>.
- J. Biolchini, P. G. Mian, A. C. C. Natalli, and G. H. Travassos. Systematic review in software engineering. Technical report, Systems Engineering and Computer Science Department, COPPE/UFRJ, 2005.
- D. Bouneffouf, A. Bouzeghoub, and A. L. Gancarski. Following the user’s interests in mobile context-aware recommender systems: The hybrid-e-greedy algorithm. In *Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops*, WAINA ’12, pages 657–662, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4652-0.
- S. Bromuri, V. Urovi, and K. Stathis. icampus: A connected campus in the ambient event calculus. *IJACI*, 2(1):59–65, 2010.
- R. de F. Bulcão Neto. *Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis ao contexto*. PhD thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (USP), novembro 2006.
- R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, november 2002. ISSN 0924-1868.

- V. Cabukovski. An integrated intelligent university information system. In *Computer Science and Society (ISCCS), 2011 International Symposium on*, pages 393–395, 2011. doi: 10.1109/ISCCS.2011.105.
- V. Callaghan, M. Gardner, B. Horan, J. Scott, L. Shen, and M. Wang. A mixed reality teaching and learning environment. In *Proceedings of the 1st international conference on Hybrid Learning and Education*, ICHL '08, pages 54–65, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85169-1.
- K. S. Candan and M. L. Sapino. *Data Management for multimedia retrieval*. Cambridge University Press, 1st edition, 2010.
- A. C. P. de L. F. de Carvalho, A. Brayner, A. Loureiro, A. L. Furtado, A. von Staa, C. J. P. de Lucena, C. S. de Souza, C. M. B. Medeiros, C. L. Lucchesi, E. S. Silva, F. R. Wagner, I. Simon, J. Wainer, J. C. Maldonado, J. P. M. de Oliveira, L. Ribeiro, L. Velho, M. A. Gonçalves, M. C. C. Baranauskas, M. Mattoso, N. Ziviani, P. O. A. Navaux, R. S. Torres, V. A. F. Almeida, W. Jr. Meira, and Y. Kohayakawa. Grandes desafios da pesquisa em computação no brasil – 2006 – 2016. Technical report, Sociedade Brasileira de Computação, maio 2006.
- A. M. G. Carvalho. eAgora: definição arquitetural e protótipo baseado em agentes e contexto do usuário. Monografia de graduação, Instituto de Ciências Exatas, Departamento de Ciência da Computação, Universidade de Brasília, 2013.
- L. A. Castro and J. Favela. Continuous tracking of user location in wlans using recurrent neural networks. In *Proceedings of the Sixth Mexican International Conference on Computer Science*, ENC'05, pages 174–181, Washington, DC, USA, 2005. IEEE Computer Society.
- H. Chalupsky, T. Finin, R. Fritzson, D. McKay, S. Shapiro, and G. Wiederhold. An overview of kqml: A knowledge query and manipulation language, 1992.
- C. C. Chang and S. T. Hsu. Personalized service provision in a context-aware shopping environment. In *Proceedings of the Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 103–108, 2012.
- D. N. Chen and Y. C. Chiang. Combining personal ontology and collaborative filtering to design a document recommendation system. *Journal of Service Science and Management*, 2(4):322–328, 2009.
- R. C. Chen and Y. H. Lin. Using zeebee sensor network with artificial neural network for indoor location. In *Natural Computation (ICNC), 2012 Eighth International Conference on*, pages 290–294. IEEE, 2012.
- T. Chen and L. He. Collaborative filtering based on demographic attribute vector. In *Proceedings of the International Conference on Future Computer and Communication*, FCC' 09, pages 225–229. Institute of Electrical and Electronics Engineers (IEEE), 2009.
- O. Choi and S. Y. Han. Personalization of rule-based web services. *Sensors*, 8(4):2424–2435, 2008. ISSN 1424-8220.

- A. M. Collins and E. F. Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428, 1975.
- O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data and Knowledge Engineering*, 4(1):41–64, 2003. ISSN 0169-023X.
- F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.
- K. Dam and M. Winikoff. Comparing agent-oriented methodologies’. In P. Giorgini, B. Henderson-Sellers, and M. Winikoff, editors, *Agent-Oriented Information Systems*, volume 3030 of *Lecture Notes in Computer Science*, pages 78–93. Springer Berlin Heidelberg, 2003.
- C. Davidsson and S. Moritz. Utilizing implicit feedback and context to recommend mobile applications from first use. In *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, CaRR ’11, pages 19–22, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0625-6.
- H. P. Deuk, K. K. Hyea, II Y. Choi, and J. K. Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, september 2012. ISSN 0957-4174.
- A. K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgia Institute of Technology, november 2000.
- J. Dooley, V. Callaghan, H. Hagra, M. Gardner, M. Ghanbari, and D. Al-Ghazzawi. The intelligent classroom: Beyond four walls. In *Ambient Intelligence and Smart Environments*, volume 10, pages 457–468, 2011.
- R. Dutta and D. Mukhopadhyay. Offering a product recommendation system in e-commerce. *Computing Research Repository (CoRR)*, abs/1109.4257, 2011.
- M. Endler, J. Viterbo, and H. Fonseca. Contexto, inferência e adaptação na internet do futuro. Technical report, Laboratory of Advanced Collaboration, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), maio 2010.
- L. Esmacili, B. Minaei-Bidgoli, H. Alinejad-Rokny, and M. Nasiri. Hybrid recommender system for joining virtual communities. *Research Journal of Applied Sciences, Engineering and Technology*, 4(5):500–509, 2012.
- D. Fensel, I. Horrocks, F. Harmelen, S. Decker, M. Erdmann, and M. C. A. Klein. Oil in a nutshell. In *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, EKAW, pages 1–16, London, UK, 2000. Springer-Verlag. ISBN 3-540-41119-4.
- H. C. Fonseca. Avaliação do uso de diferentes protocolos para localização com abordagem de sistema multiagente e rede neural. Dissertação de mestrado, Instituto de Ciências Exatas, Departamento de Ciência da Computação, Universidade de Brasília, 2011.

- H. C. da Fonseca, A. R. M. Neves, and C. G. Ralha. A user location case study using different wireless protocols. In *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac 2011, pages 1–4. ACM, 2011.
- U. Furbach, M. Maron, and K. Read. Campus news – an information network for pervasive universities. In *GI Jahrestagung (1)*, pages 508–512, 2007.
- M. Genesereth and R. Fikes. Knowledge interchange format. Technical report, Computer Science Department – Stanford University, 1992.
- T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical report, Knowledge Systems Laboratory – Stanford University, 1992.
- T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, june 1993. ISSN 1042-8143.
- T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 270–275, 2004.
- N. Guarino. Formal ontology and information systems. pages 3–15. IOS Press, 1998.
- N. Guarino and C. Welty. Evaluating ontological decisions with ontoclean. *Commun. ACM*, 45(2):61–65, feb 2002. ISSN 0001-0782.
- F. Harmelen, P. F. Patel-Schneider, and I. Horrocks. Reference description of the daml+oil ontology markup language, agosto 2001. Acessado em 20/08/2011, <http://www.daml.org/2001/03/reference.html>.
- N. S. Haron, N. S. Saleem, M. H. Hasan, M. M. Ariffin, and I. A. Aziz. A rfid-based campus context-aware notification system. *Journal of Computing*, 2(3), march 2010. ISSN 2151-9617.
- S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998. ISBN 0132733501.
- B. Henderson-Sellers and P. Giorgini. *Agent-Oriented Methodologies*. Idea Group Inc (IGI), 1st edition, 2005.
- K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive Mob. Comput.*, 2(1):37–64, february 2006. ISSN 1574-1192.
- K. Henricksen, S. Livingstone, and J. Indulska. Towards a hybrid approach to context modelling, reasoning and interoperation. In *Proceedings of First International Workshop on Advanced Context Modelling, Reasoning And Management*, 2004.
- S. Hoh, A. Devaraju, and C. C. Wong. A context aware framework for user centered services. In *Proceedings of the 21st International Symposium Human Factors in Telecommunication*, pages 1–8, 2008.

- J. Hong, E. H. Suh, J. Kim, and S. Kim. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36(4):7448–7457, 2009. ISSN 0957-4174.
- T. Hussein and S. Neuhaus. Explanation of spreading activation based recommendations. In *Proceedings of the 1st international workshop on Semantic models for adaptive interactive systems*, SEMAIS '10, pages 24–28, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0000-1.
- T. Hussein, D. Westheide, and J. Ziegler. Context-adaptation based on ontologies and spreading activation. In A. Hinneburg, editor, *LWA 2007: Lernen - Wissen - Adaption, Halle, September 2007, Workshop Proceedings*, pages 361–366. Martin-Luther-University Halle-Wittenberg, 2007. ISBN 978-3-86010-907-6.
- M. Imison and C. Hughes. The virtual patient project : using low fidelity, student-generated online cases in medical education. In *In Hello? Where are you in the landscape of educational technology? Proceedings of the 25th annual ASCILITE*, pages 441–445, 2008. ISBN 9780980592719.
- D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems An Introduction*. Cambridge University Press, 2010.
- G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, pages 47–51, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0407-8.
- F. Jiang. Personalized context-aware rating prediction model and recommendation approach based on neural network. In *Proceedings of the International Conference on Internet Technology and Applications*, pages 1–4. Institute of Electrical and Electronics Engineers (IEEE), 2010.
- M. Kandefer and S.C. Shapiro. Evaluating spreading activation for soft information fusion. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8, 2011.
- P. Karp, V. Chaudhri, and J. Thomere. Xol ontology exchange language, agosto 2000. Acessado em 21/08/2011, <http://www.ai.sri.com/pkarp/xol1>.
- B. Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.
- A. R. Lamas. Uma arquitetura para o desenvolvimento de sistemas de informações geográficas móveis sensíveis ao contexto. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Viçosa, 2009.
- T. C. Lech and L. W. M. Wienhofen. Ambieagents: a scalable infrastructure for mobile and context-aware information services. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 625–631, New York, NY, USA, 2005. ACM. ISBN 1-59593-093-0.

- D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201517523.
- F. Lin, A. Newcomb, and A. J. Armstrong. A mas approach to course offering determination. In *Academic and Professional Development Fund Report 2011-2012*, volume 5, pages 1–6. Athabasca University Library, march 2013.
- S. Loke. *Context Aware Pervasive Systems : the architecture of a new breed of applications*. Taylor and Francis Ltd, 1st edition, 2006.
- P. Lops, M. de Gemmis, G. Semeraro, P. Gissi, C. Musto, and F. Narducci. Content-based filtering with tags: The first system. In *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications*, ISDA '09, pages 255–260. Institute of Electrical and Electronics Engineers (IEEE), 2009.
- P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems handbook*, volume XXIX, chapter 3, pages 73–105. Springer US, 2011.
- F. Lorenzi. *Uma abordagem multiagente de recomendação baseada em suposições e confiança para cenários dinâmicos*. PhD thesis, Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), novembro 2010.
- L. Lü, M. Medo, C. H Yeung, Y-C Zhang, Z.-K. Zi-Ke Zhang, and T. Zhou. Recommender systems. *Physics Reports*, 519(1):1–49, 2012. ISSN 0370-1573.
- S. Luke and J. Heflin. Shoe: Simple html ontology extensions, agosto 2000. Acessado em 20/08/2011, <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>.
- N. Malik, U. Mahmud, and Y. Javed. Future challenges in context-aware computing. In *International Conference WWW/Internet*, IADIS, 2007.
- M. Maron, K. Read, and M. Schulze. Campus news - artificial intelligence methods combined for an intelligent information network. In M. Muhlhauser, A. Ferscha, and E. Aitenbichler, editors, *Constructing Ambient Intelligence*, volume 11 of *Communications in Computer and Information Science*, pages 44–52. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85378-7.
- D. L. McGuinness and F. Harmelen. Owl web ontology language overview – w3c recommendation, agosto 2004. Acessado em 21/08/2011, <http://www.w3.org/TR/owl-features>.
- M. Mircea and A. I. Andreescu. Service-oriented university: changes and opportunities towards innovation. *Procedia - Social and Behavioral Sciences*, 31:251–256, 2012. ISSN 1877-0428.
- K. Mitchell and N. J. P. Race. Oi: capturing user attention within pervasive display environments. In *Proceedings of Workshop on Pervasive Mobile Interaction Devices - Mobile Devices as Pervasive User Interfaces and Interaction Devices*, PERMID'06, pages 1–5, may 2006.

- A. M. Mustapha, M. A. Hannan, A. Hussain, and H. Basri. Ukm campus bus monitoring system using rfid and gis. In *Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on*, pages 1–5, 2010. doi: 10.1109/CSPA.2010.5545246.
- S. C. B. Nelaturu, R. Kambham, N. J. Karna, R. Parupalli, and K. Mandula. Building intelligent campus environment utilizing ubiquitous learning. In *Technology for Education (T4E), 2010 International Conference on*, pages 230–231, july 2010.
- A. R. M. Neves, H. Fonseca, L. Zoby, and C. G. Ralha. Localização de usuários em ambiente interno utilizando abordagem de sistema multiagente. *iSys - Revista Brasileira de Sistemas de Informação*, 3, 2010a. Site: <http://www.seer.unirio.br/index.php/isys/article/view/1309>.
- A. R. M. Neves, L. T. Maia, C. G. Ralha, and R. P. Jacobi. Prototype for indoor localization based on multi-agent system. In *Proceedings of the Third International Conference on Intelligent and Advanced Systems, ICIAS 2010*, pages 14–17. IEEE Computer Society, 2010b.
- A. R. M. Neves, A. M. G. Carvalho, and C. G. Ralha. Agent-based architecture for context-aware and personalized event recommendation. *Expert Systems With Applications*, 2013a. doi: 10.1016/j.eswa.2013.07.081. No prelo.
- A. R. M. Neves, A. M. G. Carvalho, and C. G. Ralha. An agent-based architecture for personalized recommendation using spreading activation. In *Proceedings of the 8th International Conference on Information Processing, Management and Intelligent Information Technology, ICIPT 2013*, pages 1–5, 2013b.
- A. R. M. Neves, H. C. Fonseca, and C. G. Ralha. Location agent: A study using different wireless protocols for indoor localization. *International Journal of Wireless Communications and Mobile Computing*, 1(1):1–6, may 2013c. doi: 10.11648/j.wcmc.20130101.11.
- J. W. P. Ng, N. Azarmi, M. Leida, F. Saffre, A. Afzal, and P. D. Yoo. The intelligent campus (icampus): End-to-end learning lifecycle of a knowledge ecosystem. In *Proceedings of the 2010 Sixth International Conference on Intelligent Environments, IE '10*, pages 332–337. IEEE Computer Society, 2010. ISBN 978-0-7695-4149-5.
- J. W. P. Ng, M. J. Zemerly, and O. A. Al Hammadi. Context-aware collaborative mlearning in an intelligent campus environment. In *GCC Conference and Exhibition, 2011 IEEE*, pages 63–64, 2011.
- N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory – Stanford University, 2001. Acessado em 14/08/2011, <http://protege.stanford.edu/publications/>.
- D. Oard and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the Association for the Advancement of Artificial Intelligence Workshop on Recommender Systems*, pages 81–83, 1998.

- D.l Oberle, S. Grimm, and S. Staab. An ontology for software. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 383–402. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-70999-2.
- R. L. Olsen. Challenges and solutions in achieving personalisation through context adaptation. *Teletronikk*, 103(1):70–84, 2007. ISSN 0085-7130.
- L. Padgham and M. Winikoff. Prometheus: a methodology for developing intelligent agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, AAMAS '02, pages 37–38, New York, NY, USA, 2002. ACM. ISBN 1-58113-480-0.
- L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, 1st edition, 2004.
- R. Pagare and A. Shinde. A study of recommender system techniques. *International Journal of Computer Applications*, 47(16):1–4, june 2012.
- K. Panu, M. Jani, K. Juha, K. Heikki, and M. Esko-Juhani. Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3):42–51, july 2003. ISSN 1536-1268.
- A. L. Pereira. Revisão sistemática da literatura sobre produtos usados no tratamento de feridas. Master’s thesis, Universidade Federal de Goiás, 2006.
- Z. Qingfeng and S. Longqing. Construction of intelligent agent-based university personalized information service system. In *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*, volume 1, pages 190–192, aug. 2011.
- L. Quin. Extensible markup language (xml) – w3c ubiquitous web domain, agosto 2001. Acessado em 22/08/2011, <http://www.w3.org/XML/>.
- P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM. ISBN 0-89791-689-1.
- J. M. A. Rodríguez, J. E. L. Gayo, and P. O. Pablos. Ontospread: A framework for supporting the activation of concepts in graph-based structures through the spreading activation technique. In M. D. Lytras, D. Ruan, R. D. Tennyson, P. Ordonez De Pablos, F. J. García Peñalvo, and L. Rusu, editors, *Information Systems, E-learning, and Knowledge Management Research*, volume 278 of *Communications in Computer and Information Science*, pages 454–459. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-35878-4.
- M. Rodríguez and J. Favela. An agent middleware for ubiquitous computing in healthcare. In Margarita Sordo, Sachin Vaidya, and Lakhmi Jain, editors, *Advanced Computational Intelligence Paradigms in Healthcare*, volume 107 of *Studies in Computational Intelligence*, pages 117–149. Springer Berlin / Heidelberg, 2008.

- M. A. Rodriguez, A. Pepe, and J. Shinavier. The dilated triple. In L. C. Jain and X. Wu, editors, *Advanced Information and Knowledge Processing*, pages 3–14. Springer-Verlag, 2010.
- L. Rokach and S. Kisilevich. Initial profile generation in recommender systems using pairwise comparison. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1854–1859, 2012. ISSN 1094-6977.
- S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Cambridge, MA, 2009. ISBN 978-0136042594.
- D. Saha and A. Mukherjee. Pervasive computing: A paradigm for the 21st century. *Computer*, 36(3):25–31, march 2003. ISSN 0018-9162.
- G. Samaras and C. Panayiotou. Personalized portals for the wireless user based on mobile agents. In *Proceedings of the 2nd international workshop on Mobile commerce, WMC '02*, pages 70–74, New York, NY, USA, 2002. ACM. ISBN 1-58113-600-5.
- E. G. Santana and F. M. César. Uma aplicação de informação geográfica sensível ao contexto. Monografia de graduação, Departamento de Engenharia e Ciência da Computação, Instituto de Educação Superior de Brasília, 2013.
- D. M. Santos. Construção de uma ontologia para ambiente acadêmico – ontounb. 18 congresso de iniciação científica da unb, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília, 2012.
- M. Satyanarayanan. Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10–17, august 2001.
- J. B. Schafer. *MetaLens: a Framework for Multi-source Recommendations*. PhD thesis, University of Minnesota, july 2001.
- B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, sep/oct 1994. ISSN 0890-8044.
- G. Schreiber, B. J. Wielinga, H. Akkermans, W. V. Velde, and A. Anjewierden. Cml: The common-kads conceptual modelling language. In *Proceedings of the 8th European Knowledge Acquisition Workshop on A Future for Knowledge Acquisition*, pages 1–25, London, UK, 1994. Springer-Verlag. ISBN 3-540-58487-0.
- N. Shadbolt, E. Motta, and A. Rouge. Constructing knowledge-based systems. *IEEE Softw.*, 10(6):34–38, november 1993. ISSN 0740-7459.
- R. Sikka, A. Dhankhar, and C. Rana. A survey paper on e-learning recommender system. *International Journal of Computer Applications*, 47(9):27–30, june 2012.
- R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.*, 29(12):1104–1113, dec 1980. ISSN 0018-9340.

- T. Strang and C. L. Popien. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, september 2004.
- R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998.
- X. Su and L. Ilebrekke. A comparative study of ontology languages and tools. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering, CAiSE '02*, pages 761–765, London, UK, 2002. Springer-Verlag. ISBN 3-540-43738-X.
- D. Tavangarian, U. Lucke, and U. Rostock. Pervasive university – a technical perspective. *Information Technology*, 51(1):6–13, january 2009.
- P. Teufl, U. Payer, and P. Parycek. Automated analysis of e-participation data by utilizing associative networks, spreading activation and unsupervised learning. In *Proceedings of the 1st International Conference on Electronic Participation, ePart '09*, pages 139–150, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03780-1.
- T. Tiberghien, M. Mokhtari, H. Aloulou, and J. Biswas. Semantic reasoning in context-aware assistive environments to support ageing with dementia. In *The Semantic Web – ISWC 2012, Lecture Notes in Computer Science*, pages 212–227. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35172-3.
- M. Tkalcivc, A. Odi, A. Ko, and J. F. Tasic. Impact of implicit and explicit affective labeling on a recommender system’s performance. In *Proceedings of the 19th international conference on Advances in User Modeling, UMAP'11*, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-28508-0.
- A. Troussov, D. Parra, and P. Brusilovsky. Spreading activation approach to tag-aware recommenders: Modeling similarity on multidimensional networks. In *Proceedings of Workshop on Recommender Systems and the Social Web at the 2009 ACM Conference on Recommender systems, RecSys '09*. ACM, october 2009.
- K. N. Truong, G. D. Abowd, and J. A. Brotherton. Who, what, when, where, how: Design issues of capture & access applications. In *UbiComp*, volume 2201 of *Lecture Notes in Computer Science*, pages 209–224. Springer, 2001. ISBN 3-540-42614-0.
- G. Tsatsaronis, M. Vazirgiannis, and I. Androutsopoulos. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, pages 1725–1730, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- A. Tuzhilin. Personalization: the state of the art and future directions. In G. Adomavicius and A. Gupta, editors, *Business Computing, Handbooks in Information Systems*, volume 3, chapter 1, pages 3–40. Emerald group publishing limited, Howard House, Wagon Lane, Bingley BD 16 1 WA, UK, 1st edition, january 2009.

- M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13:31–89, 1995.
- A. Vizcaíno, J. P. Soto, J. Portillo-Rodríguez, and M. Piattini. A multi-agent recommender system to suggest documents in communities of practice. In Y. Demazeau, editor, *Trends in practical applications of agents and multiagent systems*, volume 71, chapter 3, pages 339–346. Springer-Verlag, 2010.
- C. G. von Wangenheim and A. von Wangenheim. *Raciocínio baseado em casos*. Manole, Barueri, São Paulo, Brasil, 1st edition, 2003.
- R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, january 1992. ISSN 1046-8188.
- G. Weiss. *Multiagent Systems: a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, MA, 1st edition, 1999. ISBN 978-0262232036.
- D. Weyns and T. Holvoet. A formal model for situated multi-agent systems. *Fundam. Inf.*, 63(2–3):125–158, may 2004. ISSN 0169-2968.
- M. Wooldridge. *An introduction to Multi-Agent Systems*. John Wiley & Sons, Ltd., 2nd edition, 2009. ISBN 978-0470519462.
- B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 21–30, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5.
- J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1):36–66, february 2012.
- R. Zender and D. Tavangarian. Service-oriented university: Infrastructure for the university of tomorrow. In D. Tavangarian, T. Kirste, D. Timmermann, U. Lucke, and D. Versick, editors, *Intelligent Interactive Assistance and Mobile Multimedia Computing*, volume 53 of *Communications in Computer and Information Science*, pages 73–84. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-10262-2.

Apêndice A

Modelo do Protocolo de Revisão Sistemática

1. Objetivo:

.....
.....

2. Formulação da pergunta:

- intervenção;
- controle;
- população;
- resultados; e
- aplicação.

3. Critérios de seleção de fontes:

.....
.....

4. Métodos de busca de fontes:

.....
.....

5. Palavras-chave:

.....

.....

6. Listagem de fontes:

.....

.....

7. Tipos dos artigos:

.....

.....

8. Idioma dos artigos:

.....

.....

9. Critérios de inclusão e exclusão dos artigos:

.....

.....

10. Critérios de qualidade dos estudos primários:

.....

.....

11. Processo de seleção dos estudos primários:

.....

.....

12. Avaliação da qualidade dos estudos primários:

.....

.....

13. Estratégia de extração de informação:

.....

.....

14. Sumarização dos resultados:

.....

.....

Apêndice B

Formulário de Condução da Revisão Sistemática

1. Fonte (fonte na qual a busca foi conduzida):

.....
.....

2. Data de busca:

.....
.....

3. Palavras-chave utilizadas:

.....
.....

4. *Strings* de busca utilizadas (combinação de palavras-chave utilizadas):

.....
.....

5. Lista de artigos encontrados (referências dos artigos encontrados pela busca):

.....
.....

6. Lista dos artigos incluídos:

• Nome do artigo:

– autores:

– data de publicação:

– veículo de publicação:

- Critérios de inclusão e exclusão:
- Justificativa:

7. Lista dos artigos excluídos:

- Nome do artigo:
 - autores:
 - data de publicação:
 - veículo de publicação:
- Critérios de inclusão e exclusão:
- Justificativa:

Apêndice C

Formulário de Seleção de Estudos da Revisão Sistemática

1. Nome do artigo:

- autores:
- data de publicação:
- veículo de publicação:
- fonte (na qual o artigo foi obtido):
- situação (incluído ou excluído):

2. Critérios de Inclusão e Exclusão:

Tabela C.1: Critérios de Inclusão e Exclusão

Critérios	Resultado
1. Os artigos devem estar disponíveis na Web?	Sim ou Não
2. Os artigos devem apresentar textos completos dos estudos em formato eletrônico?	Sim ou Não
3. Os artigos devem estar descritos em inglês?	Sim ou Não
4. Os artigos devem contemplar técnicas de inspeção de documentos de requisitos descritos em linguagem natural?	Sim ou Não
5. Os artigos devem contemplar a execução de estudos experimentais investigando técnicas de inspeção de documentos de requisitos?	Sim ou Não

3. Justificativa:

Apêndice D

Formulário de Extração de Dados da Revisão Sistemática

1. Nome do artigo:

- autores:
- data de publicação:
- veículo de publicação:
- fonte:
- abstract:
- resumo (feito pelo pesquisador):

2. Estudo

- data de execução:
- local:
- tipo (experimento, estudo de caso):

3. Descrição:

.....
.....

4. Hipóteses avaliadas:

.....
.....

5. Variáveis independentes:

.....
.....

6. Variáveis dependentes:

.....
.....

7. Participantes:

.....
.....

8. Material:

.....
.....

9. Projeto do estudo:

.....
.....

10. Ameaças à validade:

.....
.....

11. Resultados:

.....
.....

12. Comentários adicionais (comentário do pesquisador acerca do estudo):

.....
.....

13. Referências relevantes (lista das referências relevantes, explicando porque ela são relevantes):

.....
.....

Apêndice E

Artefatos da Metodologia Prometheus

A seguir são apresentados detalhadamente os artefatos que constituem a metodologia Prometheus.

1. Objetivos e Funcionalidades.

Identificar os objetivos do sistema e as funcionalidades necessárias.

Objetivos:

.....
.....

Funcionalidades:

Objetivo Associado #

.....
.....

2. Descritor de Funcionalidade.

Descrição de projeto de alto nível para cada funcionalidade.

Nome:

.....
.....

Descrição (2-3 sentenças):

.....
.....

Percepções:

.....
.....
Ações:

.....
.....
Mensagens enviadas:

.....
.....
3. Descritor de cenários.

Nome:

.....
.....
Descrição:

.....
.....
Contexto:

.....
.....
Passos do cenário: (Percepção/ Ação/ Mensagem/ Atividade com as funcionalidades associadas e os dados)

Passo	Descrição	Funcionalidades	Dados lidos	Dados escritos
-------	-----------	-----------------	-------------	----------------

.....
.....
Variações dos principais casos:

4. Diagrama de ligação de dados.

Utilizar os descritores de funcionalidade para auxiliar na identificação de possíveis agrupamentos de agentes e conseqüentemente na elaboração deste diagrama (Figura E.1)

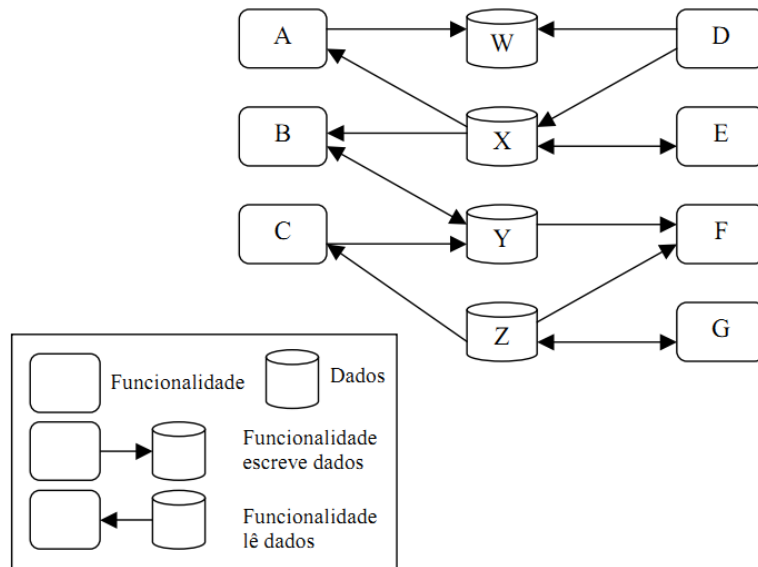


Figura E.1: Diagrama de ligação de dados

5. Diagrama de relacionamento do agente.

Detalhar (Figura E.2)

- tipo (classes) de agentes;
- número esperado de cada tipo de agente; e
- a interação entre agentes de diferentes tipos.

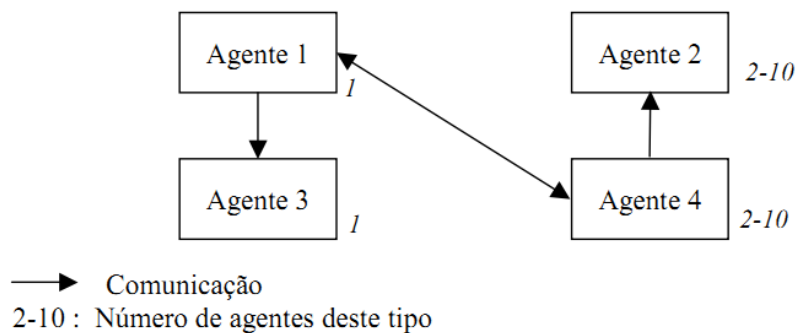


Figura E.2: Diagrama de relacionamento entre agentes

6. Descritores de percepção/ ação/ mensagem.

Descritores de percepção

Nome:

.....
.....

Descrição:

.....
.....

Origem:

.....
.....

Estrutura:

.....
.....

Eventos importantes da percepção:

.....
.....

Dados auxiliares:

.....
.....

Descritores de ação

Nome:

.....
.....

Descrição:

.....
.....

Parâmetros:

.....
.....
Tempo (duração):

.....
.....
Detecção de falha:

.....
.....
Mudanças parciais:

.....
.....
Falhas:

Descritor de mensagem

Nome:

.....
.....
Descrição:

.....
.....
Informação transportada:

.....
.....
Protocolo:

.....
.....
Agente de origem:

.....
.....
Agente de destino:

.....
.....
Finalidade:

7. Descritor de tipo de agente.

.....
.....
Descrição:

.....
.....
Objetivos:

.....
.....
Número de instâncias:

.....
.....
Capacidades necessárias:

.....
.....
Ciclo de vida do agente:

.....
.....
Inicialização:

.....
.....
Término:

.....
.....
Percepção:

.....
.....
Ação:

.....
.....
Mensagens enviadas:

.....
.....
Mensagens recebidas:

.....
.....
Interações:

.....
.....
Informação lida:

.....
.....
Informação escrita:

8. Diagrama de revisão do sistema

O diagrama de revisão de sistema (Figura E.3) ilustra os agentes, as percepções, ações, as mensagens entre os agentes e o uso de arquivos de dados compartilhados ou externos. Além disso, exibe as interfaces com as partes que não são agentes do sistema ou outros sub-sistemas.

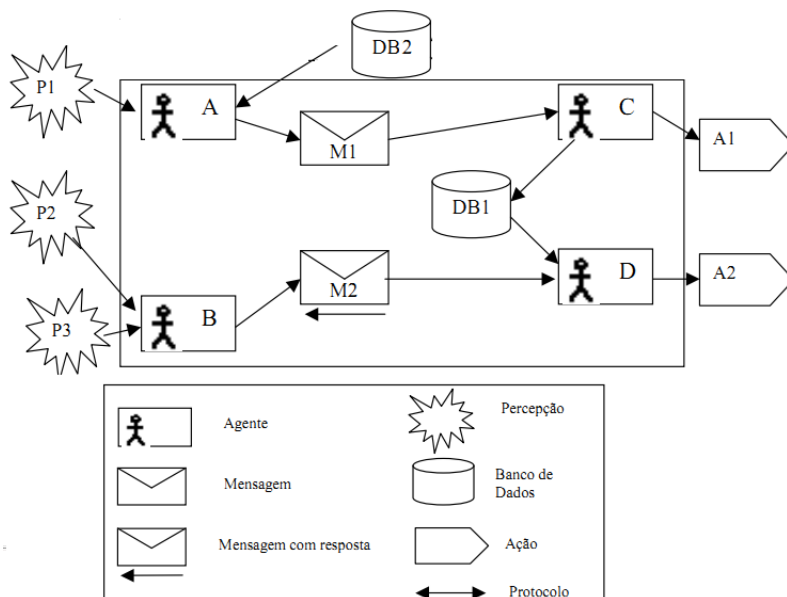


Figura E.3: Diagrama de revisão do sistema

9. Diagrama de interação

Para desenvolver o diagrama de interação (Figura E.4) é utilizado os cenários e as mensagens sobre quais funcionalidades pertencem aos agentes. Deste modo, são identificados os agentes e sobre como as mensagens são trocadas entre os agentes.

10. Diagrama de revisão do agente

Cada agente pode possuir uma ou várias capacidades. As capacidades são blocos construtivos que podem ser reutilizados pelos agentes. Para aumentar a funcionalidade da capacidade, elas podem se incorporadas a outras. O diagrama de revisão do agente (Figura E.5) ilustra as capacidades de cada agente, o evento gerado e manipulado de cada capacidade.

11. Diagrama de revisão da capacidade (Dado-Evento-Plano)

O digrama de revisão de capacidade (Figura E.6) ilustra os relacionamento entre as capacidades.

Documentação:

.....

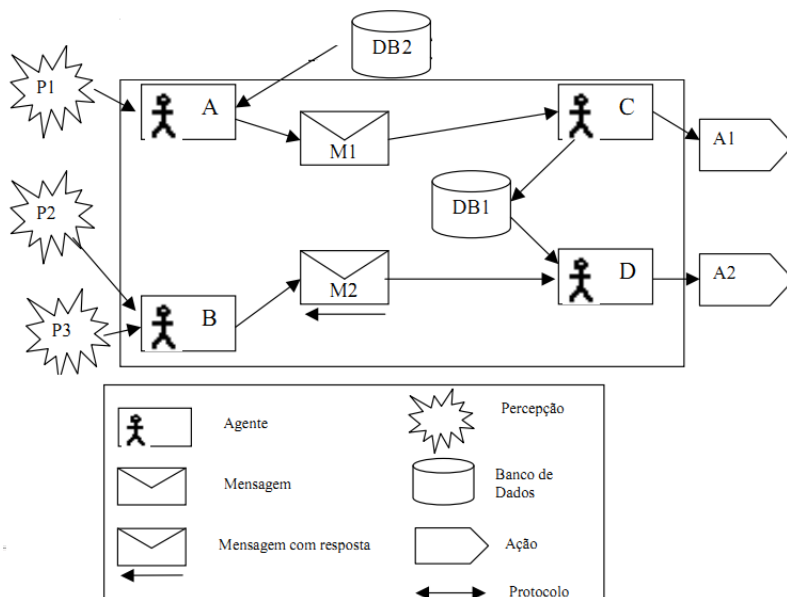


Figura E.4: Diagrama de interação

12. Descritor de tipo de plano

Documentação:

.....

.....

Tipo de evento manipulado:

.....

.....

Contexto:

.....

.....

Fracasso do plano:

.....

.....

Método de contingência:

.....

.....

Sub-tarefas geradas:

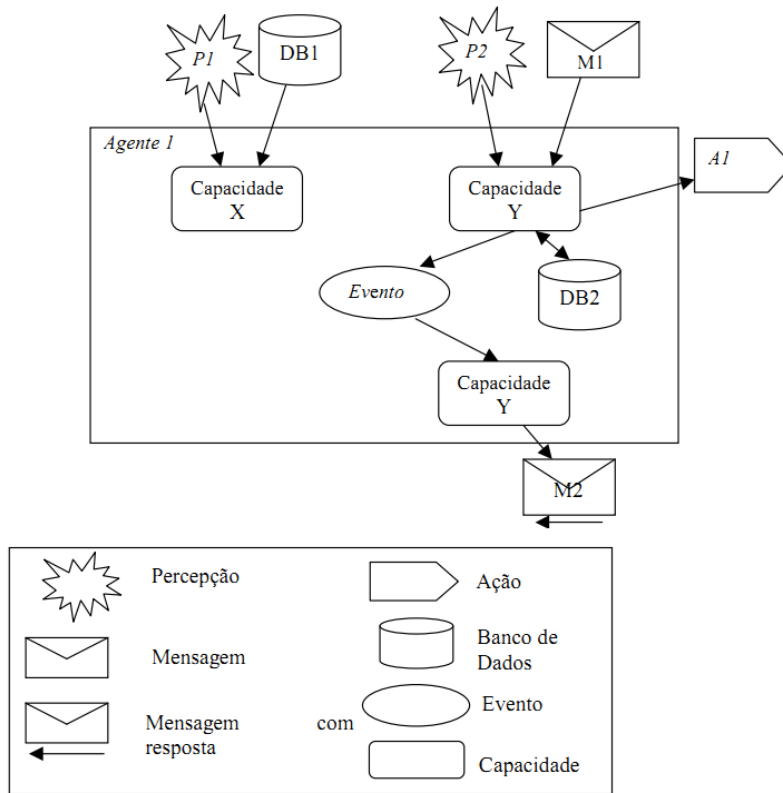


Figura E.5: Diagrama de revisão do agente

Eventos da mensagem enviada:

Novos objetivos gerados:

Procedimentos:

Informação lida:

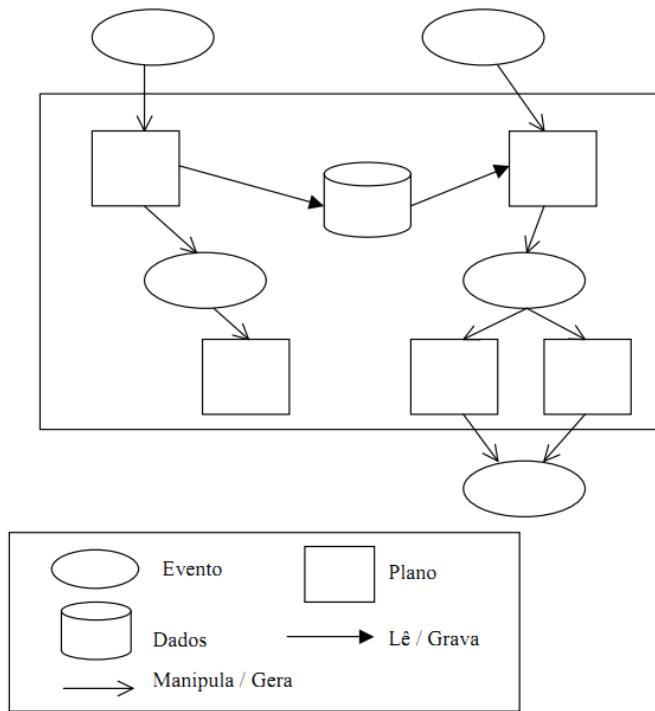


Figura E.6: Diagrama de revisão da capacidade

Informação escrita:

.....

.....

13. Descritor de tipo de evento

Nome:

.....

.....

Finalidade:

.....

.....

Argumentos:

.....

.....

Cobertura de segurança:

.....

.....
Sobreposição de respostas para um evento:
.....
.....

14. Crenças e Estruturas de dados

Descrição de projeto detalhado da estrutura de crenças, o qual inclui:

- plataforma de implementação;
- linguagem de programação; e
- especificação dos dados por meio de diagramas apropriados.

15. Dicionário de projetos

Lista de nomes que contém:

Agentes:
.....
.....

Capacidades:
.....
.....

Planos:
.....
.....

Ações:
.....
.....

Eventos:
.....
.....

Mensagens:

.....
.....

Percepções:

.....
.....

Campos de dados:

.....
.....

Apêndice F

Lista de relevância semântica para o cenário I

A lista de relevância semântica gerada após execução do algoritmo SA para o cenário I, que corresponde ao estudante X do segundo período do curso de Bacharelado em Ciência da Computação e localizado próximo ao Instituto de Física. Esta lista contém além das instâncias, o valor do peso final e a profundidade da rede, respectivamente. As quatorze instâncias representam as seguintes disciplinas:

- Materias_If;
- Física I Experimental;
- Física I;
- Física II;
- Física III Experimental;
- Física III;
- Instituto_Fisica;
- Física II Experimental;
- Física Geral II Experimental;
- Física Geral II;
- Física Geral I Experimental;
- Física Geral I;
- Probabilidade e Estatística;
- Cálculo I.

(ConceptTO([http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28](http://www.owl-ontologies.com/unnamed.owl#Materias>If,
null, null), 8.0, 1
(ConceptTO(<a href=),
null, null), 3.0, 1
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25,
null, null), 3.0, 1 =
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24,
null, null), 3.0, 1
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_31,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_30,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_29,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_27,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_26,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_23,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_22,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_83,
null, null), 1.0, 1
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72,
null, null), 1.0, 1

Apêndice G

Lista de relevância semântica para o cenário II

A lista de relevância semântica gerada após execução do algoritmo SA para o cenário II, que corresponde ao estudante Y do sétimo período do curso de Bacharelado em Ciência da Computação e localizado próximo ao Departamento de Ciência da Computação. Esta lista contém além das instâncias, o valor do peso final e a profundidade da rede, respectivamente. As trinta e quatro instâncias representam as seguintes disciplinas:

- Materias_Comp;
- Linguagem de programação;
- Engenharia de software;
- Modelagem orientada a objetos;
- Organização de arquivos;
- Teleinformática e Redes II;
- Sistemas de informação;
- Estrutura de dados;
- Trabalho de graduação II;
- Trabalho de graduação I;
- Análise e projeto de sistemas;
- Autômatos;
- Software básico;
- Circuitos digitais;

- Transmissão de dados;
- Organização e arquitetura de computadores;
- Lógica computacional I;
- Segurança de dados;
- Informática e Sociedade;
- Levantamento de dados e Pesquisa;
- Gerência de projetos;
- Tradutores;
- Teleinformática e Redes I;
- Instituto_de_Ciencias_Exatas;
- Programação sistemática;
- Introdução a Ciência da Computação;
- Projeto e Análise de algoritmos;
- Computação básica;
- Gerência de redes;
- Redes de computadores;
- Sistemas operacionais;
- Banco de dados;
- Programação avançada; e
- Teoria dos números I.

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
null, null), 8.0, 1

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53,
null, null), 5.0, 2

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61,
null, null), 3.0, 1

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
null, null), 3.0, 1

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_50,
null, null), 2.0, 2

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_54,

null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_51,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_52,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_47,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_46,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_45,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_44,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_49,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_48,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_40,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_66,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_41,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_60,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_42,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_43,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_63,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_64,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_65,
 null, null), 2.0, 2
 (ConceptTO(<http://www.owl-ontologies.com/unnamed.owl#>
 Instituto_de_Ciencias_Exatas,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_56,
 null, null), 2.0, 2
 (ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_38,
 null, null), 2.0, 2

(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_55,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_37,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_58,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_39,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_57,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_59,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_36,
null, null), 2.0, 2
(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_70,
null, null), 1.0, 1

Apêndice H

Log do algoritmo de propagação da ativação para o cenário I

O cenário I corresponde ao estudante X do segundo período do curso de Bacharelado em Ciência da Computação e foi localizado próximo ao Instituto de Física.

```
Init ontospread
Jena model for is null: creating new Jena Model
Opening resource input stream for filename: relation-weights.rdf
Loading 1 resources into the model
Loading RDF relation-weights.rdf
RDF stream loaded, model contains 49 triplets
Loaded Jena Rdf model
Applying pre adjustment
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28
to value 1.0
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_83
to value 1.0
Jena model for is null: creating new Jena Model
Add location: LocatorFile
Add location: ClassLoaderLocator
Failed to find configuration: file:location-mapping.rdf;file:location-mapping.n3;
file:location-mapping.ttl;file:etc/location-mapping.rdf;file:etc/location-mapping.n3;
file:etc/location-mapping.ttl
Add location: LocatorFile
Add location: LocatorURL
Add location: ClassLoaderLocator
Add location: LocatorFile
Add location: LocatorURL
Add location: ClassLoaderLocator
```

```

Found: file:etc/ont-policy.rdf (ClassLoaderLocator)
Opening resource input stream for filename: eAgora20.owl
Loading 1 resources into the model
Loading OWL eAgora20.owl
Loaded eAgora20.owl
Loaded Jena Owl model
Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28
Discarding instance concept Fisica_II^http://www.w3.org/2001/XMLSchema#string
conversion to Individual is not possible because is
class com.hp.hpl.jena.rdf.model.impl.LiteralImpl
Processing concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28
Adding http://www.w3.org/1999/02/22-rdf-syntax-ns#type object
http://www.owl-ontologies.com/unnamed.owl#Materias_If
Registering relation http://www.w3.org/1999/02/22-rdf-syntax-ns#type
(http://www.w3.org/1999/02/22-rdf-syntax-ns#type)
with concept http://www.owl-ontologies.com/unnamed.owl#Materias_If
Adding http://www.owl-ontologies.com/unnamed.owl#requisito_if object
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
Registering relation http://www.owl-ontologies.com/unnamed.owl#requisito_if
(http://www.owl-ontologies.com/unnamed.owl#requisito_if)
with concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
Adding http://www.owl-ontologies.com/unnamed.owl#requisito_if object
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72
Registering relation http://www.owl-ontologies.com/unnamed.owl#requisito_if
(http://www.owl-ontologies.com/unnamed.owl#requisito_if)
with concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72
Adding http://www.owl-ontologies.com/unnamed.owl#requisito_if
object http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25
Registering relation http://www.owl-ontologies.com/unnamed.owl#requisito_if
(http://www.owl-ontologies.com/unnamed.owl#requisito_if)
with concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25
SPREADING http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28
DEPTH 1 SCORE 1.0 related 5
Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_If
from http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 on
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_If
to value 1.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_If
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24 from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 on
http://www.owl-ontologies.com/unnamed.owl#requisito_if

```

Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
 to value 1.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72 from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 on
http://www.owl-ontologies.com/unnamed.owl#requisito_if
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72
 to value 1.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25 from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 on
http://www.owl-ontologies.com/unnamed.owl#requisito_if
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25
 to value 1.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25
 Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_If from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 on
http://ontospread.sf.net#INSTANCE_OF
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_If from
 value 1.0 to 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Adding relations org.ontospread.xmlbind.Relations@2814a18d
 Adding superclasses com.hp.hpl.jena.util.iterator.UniqueExtendedIterator@4929b0e1
 Creating relations with SUPERCLASS
http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica
 Added relation with hierarchy SUPERCLASS on property null
 End relations with SUPERCLASS
 Creating relations with SUBCLASS
 End relations with SUBCLASS
 Relations 1
 Processing concept http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Adding null object http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica
 Adding subclass http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica
 SPREADING http://www.owl-ontologies.com/unnamed.owl#Materias_If
 DEPTH 2 SCORE 2.0 related 11
 Activation for http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica from
http://www.owl-ontologies.com/unnamed.owl#Materias_If on
<http://ontospread.sf.net#SUPERCLASS>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica

Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24 from
value 1.0 to 3.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_31 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_31
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_31
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28 from
value 1.0 to 3.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25 from
value 1.0 to 3.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_22 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_22
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_22
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_29 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_29
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_29
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_23 from
<http://www.owl-ontologies.com/unnamed.owl#Materias>If> on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_23
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_23

Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_30 from
http://www.owl-ontologies.com/unnamed.owl#Materias_If on
<http://ontospread.sf.net#INSTANCE>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_30
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_30
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_27 from
http://www.owl-ontologies.com/unnamed.owl#Materias_If on
<http://ontospread.sf.net#INSTANCE>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_27
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_27
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_26 from
http://www.owl-ontologies.com/unnamed.owl#Materias_If on
<http://ontospread.sf.net#INSTANCE>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_26
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_26
 Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
 Discarding instance concept Fisica_I[^]<http://www.w3.org/2001/XMLSchema#string> conversion to
 Individual is not possible because is class `com.hp.hpl.jena.rdf.model.impl.LiteralImpl`
 Processing concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
 Adding <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> object
http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Registering relation [http://www.w3.org/1999/02/22-rdf-syntax-ns#](http://www.w3.org/1999/02/22-rdf-syntax-ns#type)
[type\(http://www.w3.org/1999/02/22-rdf-syntax-ns#type\)](http://www.w3.org/1999/02/22-rdf-syntax-ns#type)
 with concept http://www.owl-ontologies.com/unnamed.owl#Materias_If
 SPREADING http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24
 DEPTH 2 SCORE 3.0 related 2
 Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_If from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24 on
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_If from
 value 2.0 to 5.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_If from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24 on
http://ontospread.sf.net#INSTANCE_OF
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_If from
 value 5.0 to 8.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_If
 Applying post adjustment


```

SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
null, null),
8.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_31,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_30,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#Instituto_Fisica,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#SUPERCLASS}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_83,
null, null),
1.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_83, {}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_29,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
null, null),
3.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28, {}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_72,
null, null),
1.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.owl-ontologies.com/unnamed.owl#requisito_if}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_27,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),

```

```

PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_26,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_25,
null, null),
3.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.owl-ontologies.com/unnamed.owl#requisito_if}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_24,
null, null),
3.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.owl-ontologies.com/unnamed.owl#requisito_if}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_23,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_22,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_28,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_If,
{http://ontospread.sf.net#INSTANCE}))

```

Apêndice I

Log do algoritmo de propagação da ativação para o cenário II

O cenário II corresponde ao estudante Y do sétimo período do curso de Bacharelado em Ciência da Computação e foi localizado próximo ao Departamento de Ciência da Computação.

```
Init ontospread
Jena model for is null: creating new Jena Model
Opening resource input stream for filename: relation-weights.rdf
Loading 1 resources into the model
Loading RDF relation-weights.rdf
RDF stream loaded, model contains 49 triplets
Loaded Jena Rdf model
Applying pre adjustment
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62
to value 1.0
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_70
to value 1.0
Jena model for is null: creating new Jena Model
Add location: LocatorFile
Add location: ClassLoaderLocator
Failed to find configuration: file:location-mapping.rdf;file:location-mapping.n3;
file:location-mapping.ttl;file:etc/location-mapping.rdf;file:etc/location-mapping.n3;
file:etc/location-mapping.ttl
Add location: LocatorFile
Add location: LocatorURL
Add location: ClassLoaderLocator
Add location: LocatorFile
Add location: LocatorURL
Add location: ClassLoaderLocator
```

Found: file:etc/ont-policy.rdf (ClassLoaderLocator)
Opening resource input stream for filename: eAgora20.owl
Loading 1 resources into the model
Loading OWL eAgora20.owl
Loaded eAgora20.owl
Loaded Jena Owl model
Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62
Discarding instance concept Modelagem_Orientada_Objeto^{^^}
<http://www.w3.org/2001/XMLSchema#string>
conversion to Individual is not possible because is
class `com.hp.hpl.jena.rdf.model.impl.LiteralImpl`
Processing concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62
Adding <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> object
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
Registering relation <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
(<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>)
with concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
Adding http://www.owl-ontologies.com/unnamed.owl#requisito_cic object
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
Registering relation http://www.owl-ontologies.com/unnamed.owl#requisito_cic
(http://www.owl-ontologies.com/unnamed.owl#requisito_cic)
with concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
SPREADING http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62
DEPTH 1 SCORE 1.0 related 3
Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
from http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62 on
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
to value 1.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62 on
http://www.owl-ontologies.com/unnamed.owl#requisito_cic
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
to value 1.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62 on
http://ontospread.sf.net#INSTANCE_OF
Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
value 1.0 to 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_Comp

Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Adding relations [org.ontospread.xmlbind.Relations@2814a18d](http://www.ontospread.xmlbind.org)
 Adding superclasses [com.hp.hpl.jena.util.iterator.UniqueExtendedIterator@4929b0e1](http://www.com.hp.hpl.jena.util.iterator.UniqueExtendedIterator@4929b0e1)
 Creating relations with SUPERCLASS
http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas
 Added relation with hierarchy SUPERCLASS on property null
 End relations with SUPERCLASS
 Creating relations with SUBCLASS
 End relations with SUBCLASS
 Relations 1
 Processing concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Adding null object http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas
 Adding subclass http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas
 SPREADING http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 DEPTH 2 SCORE 2.0 related 32
 Activation for http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#SUPERCLASS>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Instituto_de_Ciencias_Exatas
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62 from
 value 1.0 to 3.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_65 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_65
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_65
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_60 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
 Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_60
 to value 2.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_60
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_48 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on

http://ontospread.sf.net#INSTANCE
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_48
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_48
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_50 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_50
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_50
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_63 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_63
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_63
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_46 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_46
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_46
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_44 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_44
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_44
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_36 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_36
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_36
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_49 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_49
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_49
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_42 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on

http://ontospread.sf.net#INSTANCE
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_42
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_42
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_51 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_51
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_51
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_64 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_64
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_64
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_58 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_58
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_58
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_66 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_66
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_66
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_37 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_37
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_37
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_56 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_56
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_56
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_52 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on

http://ontospread.sf.net#INSTANCE
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_52
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_52
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_54 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_54
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_54
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_38 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_38
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_38
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_59 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_59
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_59
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_57 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_57
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_57
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 from
value 1.0 to 3.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_40 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_40
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_40
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on

http://ontospread.sf.net#INSTANCE
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_55 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_55
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_55
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_47 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_47
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_47
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_39 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_39
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_39
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_45 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_45
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_45
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_41 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_41
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_41
Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_43 from
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp on
<http://ontospread.sf.net#INSTANCE>
Initializing score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_43
to value 2.0
Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_43
Creating concept for resource http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
Discarding instance concept Engenharia_de_Software^^

<http://www.w3.org/2001/XMLSchema#string> conversion to
 Individual is not possible because is class `com.hp.hpl.jena.rdf.model.impl.LiteralImpl`
 Processing concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
 Adding <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> object
http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Registering relation <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 (<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>)
 with concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Adding http://www.owl-ontologies.com/unnamed.owl#requisito_cic object
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53
 Registering relation http://www.owl-ontologies.com/unnamed.owl#requisito_cic
 (http://www.owl-ontologies.com/unnamed.owl#requisito_cic)
 with concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53
 SPREADING http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61
 DEPTH 2 SCORE 3.0 related 3
 Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 on
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
 value 2.0 to 5.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Activation for http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53 from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 on
http://www.owl-ontologies.com/unnamed.owl#requisito_cic
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53 from
 value 2.0 to 5.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53
 Activation for http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61 on
http://ontospread.sf.net#INSTANCE_OF
 Updating score of concept http://www.owl-ontologies.com/unnamed.owl#Materias_Comp from
 value 5.0 to 8.0
 Adding to stack http://www.owl-ontologies.com/unnamed.owl#Materias_Comp
 Applying post adjustment
 SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_50,
 null, null),
 2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
 {<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
 PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
 {<http://ontospread.sf.net#INSTANCE>}}))
 SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_53,
 null, null),

5.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_54,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_51,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_52,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
null, null),
8.0, 1 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_47,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_46,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{<http://ontospread.sf.net#INSTANCE>}})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_45,
null, null),
2.0, 2 = {PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>}),

```

PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_44,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_49,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_48,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_40,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_66,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_41,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_60,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),

```

```

PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_42,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_43,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_61,
null, null),
3.0, 1 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.owl-ontologies.com/unnamed.owl#requisito_cic}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
null, null),
3.0, 1 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62, {}}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_63,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_64,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_65,
null, null),
2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathTO(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}))
SpreadedConceptTO(ConceptTO(http://www.owl-ontologies.com/unnamed.owl#
Instituto_de_Ciencias_Exatas,
null, null), 2.0, 2 ={PathTO(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,

```

```

{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#SUPERCLASS}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_70,
null, null), 1.0, 1 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_70,
{}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_56,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_38,
null, null), 2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_55,
null, null), 2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_37,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_58,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_39,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE}}))
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_57,
null, null),

```

```
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE})})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_59,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE})})
SpreadedConceptT0(ConceptT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_36,
null, null),
2.0, 2 ={PathT0(http://www.owl-ontologies.com/unnamed.owl#eAgora2_Instance_62,
{http://www.w3.org/1999/02/22-rdf-syntax-ns#type}),
PathT0(http://www.owl-ontologies.com/unnamed.owl#Materias_Comp,
{http://ontospread.sf.net#INSTANCE})})
```