

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**Agendamento de Atividades Apoiado por Sistema Multiagente em
Ambientes Virtuais de Aprendizagem**

MARCIA DA COSTA PIMENTA

Manaus
2010

MARCIA DA COSTA PIMENTA

**Agendamento de Atividades Apoiado por Sistema Multiagente em
Ambientes Virtuais de Aprendizagem**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Informática.

Orientador: Prof. José Francisco de Magalhães Netto, DSc.

Manaus
2010

MARCIA DA COSTA PIMENTA

**Agendamento de Atividades Apoiado por Sistema Multiagente em
Ambientes Virtuais de Aprendizagem**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Informática.

Banca Examinadora

Prof. José Francisco de Magalhães Netto, DSc. – Orientador
DCC - PPGI - UFAM

Prof. Alberto Nogueira de Castro Júnior, PhD.
DCC - PPGI - UFAM

Prof. Raimundo da Silva Barreto, DSc.
DCC - PPGI - UFAM

Prof. Crediné Silva de Menezes, DSc.
UFES / UFRGS

Manaus
2010

A Deus, acima de tudo.

Aos meus pais e meu esposo, os quais amo
incondicionalmente.

AGRADECIMENTOS

Antes de tudo a Deus, que sempre me proporciona tamanhas bênçãos, dando-me saúde, inteligência, força e perseverança na jornada da vida.

A meu esposo, Héber Martins, por seu amor sem medida, pela extraordinária paciência, carinho, colaboração e por todas as palavras de incentivo a mim dedicadas diariamente.

Aos meus pais, Agenor e Mariazinha, pela educação, dedicação, apoio e carinho de todas as horas; e ao meu irmão Agenor Filho, amigo sempre presente.

Ao meu orientador José Francisco M. Netto, pela oportunidade do aprendizado que me propiciou e pelos direcionamentos fundamentais para realização deste trabalho.

A amiga mais chegada que uma irmã, que suportou minha ausência sempre com paciência e palavras de ânimo e afeto: Anandria, obrigada!

A amiga Viviane Gomes pelo incentivo e motivação quanto ao meu ingresso neste curso. Obrigada pela ajuda nos primeiros e difíceis meses em que estudamos juntas, e nos demais também; pelos ensinamentos, lágrimas, alegrias e por tudo de bom você sempre compartilha comigo.

Aos amigos e colegas pelas conversas, momentos de aprendizado e de amizade, principalmente nos momentos de quase desesperança. Obrigada a: Márcio Alencar, Vitor Bremgartner, Marcelo Souza, Ricardo Câmara, Sionise Gomes, Edna Magalhães, Lady Daiana, Christophe Xavier, Leonardo Santos e demais colegas de curso pelas idéias compartilhadas e por sempre estarem dispostos, de uma forma ou outra, a colaborar para a realização deste curso.

A todo corpo docente do Mestrado pelos ensinamentos e atenção dispensada, em especial aos professores Edleno Moura, José Francisco Netto, Alberto Castro, Raimundo Barreto, Tayana Conte e Ruiteir Caldas.

A UFAM, em especial ao Departamento de Ciência da Computação pela oportunidade do curso e por sua secretaria representada pela pessoa de Elienai Nogueira.

Ao IFAM, pela liberação parcial; em especial agradeço aos colegas de trabalho: Antônio Ferreira, Roceli Lima, Viviane Gomes e a equipe de EAD desta instituição.

E a todos, que de forma direta ou indireta acreditaram e incentivaram a realização deste trabalho. Obrigada! Contem sempre comigo e nunca desistam dos seus objetivos!

“Em tudo somos atribulados, mas não angustiados; perplexos, mas não desanimados. Perseguidos, mas não desamparados; abatidos, mas não destruídos... Porque a nossa leve e momentânea tribulação produz para nós um peso eterno de glória mui excelente; Não atentando nós nas coisas que se vêem, mas nas que se não vêem; porque as que se vêem são temporais, e as que se não vêem são eternas.” II Cor. 4: 8-9 e 17-18.

RESUMO

A utilização de ambientes virtuais de aprendizagem usados em cursos à distância, está cada vez mais presente na Educação. No entanto, algumas questões precisam ser aperfeiçoadas nestes ambientes, como por exemplo, o acompanhamento das agendas. Neste trabalho relatamos uma investigação sobre o acompanhamento das atividades agendadas nestes ambientes de apoio a comunidades virtuais. Discutimos o problema referente ao acompanhamento de agendas, isto é, ao domínio e visualização dos compromissos no ambiente virtual de aprendizagem propondo um protótipo para este agendamento apoiado por Sistema Multiagente. O objetivo do modelo proposto é realizar o agendamento das atividades síncronas realizadas por participantes e mediadores. Cada usuário do ambiente poderá definir sua disponibilidade, a fim de realizar o agendamento de reuniões individuais ou em grupo, tendo como finalidade manter o usuário informado sobre os compromissos agendados, tornando suas tarefas diárias mais simples e visíveis no ambiente virtual de aprendizagem.

Palavras-chaves: Agentes; Ambientes Virtuais de Aprendizagem; Sistema Multiagente; Agendamento.

ABSTRACT

The use of virtual learning environments used in distance learning courses, is increasingly present in education. However, some issues need to be improved in these environments, for example, monitoring of schedules. We report an investigation on the monitoring of scheduled activities in these environments to support virtual communities. We discuss the problem with monitoring schedules, ie the domain and display of appointments in a virtual learning environment by proposing a prototype for this schedule supported by Multiagent System. The objective of the proposed model is to perform the scheduling of synchronous activities undertaken by participants and mediators. Each user can set their availability environment in order to scheduling individual or group meetings, with the purpose to keep the user informed of the appointments, making daily tasks easier and visible in the virtual learning environment.

Keywords: Agents, Virtual Learning Environments; Multiagent Systems; Scheduling.

LISTA DE FIGURAS

Figura 1. Página principal do AVA Moodle. Fonte: [MOODLE 2010]	12.
Figura 2. Uma instância do Moodle, com um curso e alguns blocos funcionais ativos	13.
Figura 3. Eventos no Calendário do Moodle	16.
Figura 4. Timetabling apresentado em um sistema de agenda	18.
Figura 5. Proposta de Berger <i>et al</i> 2008, apresentando o melhor percurso para realizar o agendamento.	20.
Figura 6. Árvore do Agendamento de Reuniões proposta por Yang 2009.	21.
Figura 7. Arquitetura utilizada por [Eijk 2002] em Scheduler Agent System for Traveling Users.	23.
Figura 8. Agentes Inteligentes dando suporte no AVA. [Xu 2006]	24.
Figura 9. Tela de configuração do agente no AVA. [Campana 2008]	25.
Figura 10. Diagrama de Classes dos Agentes. [Cuesta 2004]	26.
Figura 11. Agente Meeting Scheduler.jar em execução - JADE	27.
Figura 12. Processo de Interação do Agente com o Ambiente. Fonte: Apud [Russel e Norvig 2004].	37.
Figura 13. A Arquitetura Básica de um Agente. Fonte: [Russel e Norvig 2004]	42.
Figura 14. Arquitetura Básica do Protótipo Agendamento	46.
Figura 15. DbAgent2 - Código do Agente no JAVA	47.
Figura 16. Agente de Agendamento ativo no JADE	48.
Figura 17. Diagrama de Casos de Uso para o Agendamento	49.
Figura 18. Diagrama de Classes para o Agendamento	50.
Figura 19. Diagrama de Sequência para definição de preferências	51.
Figura 20. Aplicação phpMyAdmin ativa	54.
Figura 21. Página do Moodle para realizar ou atualizar o cadastro do usuário	56.
Figura 22. Página do Moodle para realizar o acesso com login e senha	57.
Figura 23. Página principal do curso no Moodle com o link para o Protótipo Agendamento	58.
Figura 24. Página do protótipo para o cadastro de disponibilidade de horário dos usuários	59.
Figura 25. Página que apresenta que o cadastro de disponibilidade já foi realizado	59.
Figura 26. Consulta de disponibilidade de horário dos usuários do AVA	60.

Figura 27. Tela da opção de Agendamento individual	61.
Figura 28. Tela de Agendamentos realizados	62.

LISTA DE TABELAS

Tabela 1.	Apresentação do resumo de cada trabalho relacionado	28.
-----------	---	-----

LISTA DE ABREVIACES

AVA	Ambiente Virtual de Aprendizagem
BMO	Bussiness Meeting Organizer
DER	Diagrama de Entidade Relacionamento
EAD	Educao a Distncia.
FIPA	Foundation of Intelligent Physical Agents
GNU	General Public License
GPL	General Public License
GPRS	General Packet Radio Service
GPS	General Problem Solver
HTML	Hypertext Markup Language
IA	Inteligncia Artificial
IFAM	Instituto Federal Amazonas
IUB	Instituto Universal Brasileiro
JADE	Java Agent Development Framework
LAN	Local Area Network
LMS	Learning Management System
MEB	Movimento de Educao de Base
MOODLE	Modular Object-Oriented Dynamic Learning Environment
PAGE	Perceptions Actions Goals Environment
PASA	Problema de Agendamento Semanal de Aulas
PHP	Hypertext Preprocessor
SGBD	Sistema Gerenciador de Banco de Dados
SMA	Sistema Multiagente
SMS	Short Message Service

SQL	Structured Query Language
UML	Unified Modeling Language
WWW	World Wide Web

SUMÁRIO

1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	3
1.2 OBJETIVOS	4
1.2.1 <i>Objetivo geral</i>	4
1.2.2 <i>Objetivos específicos</i>	4
1.3 METODOLOGIA	5
1.4 ORGANIZAÇÃO DO TRABALHO	5
2 REFERENCIAL TEÓRICO	7
2.1 EDUCAÇÃO A DISTÂNCIA	7
2.2 AMBIENTES VIRTUAIS DE APRENDIZAGEM (AVAs)	8
2.3 PLATAFORMA EDUCACIONAL MOODLE	10
2.3.1 <i>PRINCIPAIS ENTIDADES DO MOODLE</i>	11
2.3.2 <i>BLOCOS FUNCIONAIS</i>	13
2.3.3 <i>CALENDÁRIO</i>	15
2.4 AGENDAMENTO	17
3 AGENTES DE SOFTWARE	31
3.1 INTELIGÊNCIA ARTIFICIAL	31
3.1.1 <i>SISTEMAS QUE AGEM COMO HUMANOS</i>	33
3.1.2 <i>SISTEMAS QUE PENSAM COMO HUMANOS</i>	33
3.1.3 <i>SISTEMAS QUE PENSAM RACIONALMENTE</i>	34
3.1.4 <i>SISTEMAS QUE AGEM RACIONALMENTE</i>	34
3.1.5 <i>APLICAÇÕES</i>	35
3.1.6 <i>CONTROLE DE ROBOS</i>	35
3.1.7 <i>JOGOS E SIMULADORES</i>	35
3.1.8 <i>RECOMENDAÇÃO DE PRODUTOS</i>	36
3.2 AGENTES	36
3.2.1 <i>PROPRIEDADES DOS AGENTES DE SOFTWARE</i>	37
3.2.2 <i>DESCREVENDO UM AGENTE INTELIGENTE</i>	39
3.2.3 <i>ARQUITETURAS DE UM AGENTE</i>	42
4 PROTÓTIPO	44
4.1 DECISÕES DO PROJETO	44
4.2 DESCRIÇÕES DO PROTÓTIPO	45
4.3 ARQUITETURA DO PROTÓTIPO	46
4.4 MODELAGEM DO PROTÓTIPO	48
4.4.1 <i>Diagrama de caso de uso</i>	48
4.4.2 <i>Diagrama de Classes</i>	50
4.4.3 <i>Diagrama de Sequencia</i>	51
5 EXPERIMENTOS E VALIDAÇÃO	53
5.1 CENÁRIO DE UTILIZAÇÃO	53
5.2 CARACTERÍSTICAS GERAIS DO PROTÓTIPO	54
5.3 EXEMPLO DO AGENDAMENTO	55
6 CONSIDERAÇÕES FINAIS DO TRABALHO	63
6.1 CONTRIBUIÇÕES	64
6.2 TRABALHOS FUTUROS	64
7 REFERÊNCIAS BIBLIOGRÁFICAS	66
APÊNDICE A – ESTRUTURA DO PROTÓTIPO AGENDAMENTO	70
APÊNDICE B – AGENTE DO JADE MEDIADOR NO PROTÓTIPO	81
APÊNDICE C - TABELAS DO BANCO DE DADOS MYSQL DESIGNADAS AO PROTÓTIPO	86

CAPÍTULO 1

1. INTRODUÇÃO

A informática se converte como recurso inovador das atividades humanas, abrangendo vários setores da sociedade. Com a utilização de computadores, podemos desenvolver as mais difíceis tarefas, com rapidez e maior eficiência e segurança.

Nas últimas décadas a educação a distância, uma das modalidades da educação na qual o uso da informática tem tido grande influência, tomou um novo impulso com a integração de tecnologias tradicionais de comunicação associados aos materiais impressos enviados pelo correio, o que favoreceu a disseminação e a democratização do acesso à educação em diferentes níveis, permitindo atender a um amplo número de alunos.

Diante dos recursos disponíveis na Internet e da necessidade de ambientes educacionais que suportem adequadamente as atividades dos mediadores e aprendizes, a *Internet* ou simplesmente *Web*, se apresenta como um dos recursos capazes de atender parte das expectativas dos pesquisadores da área de Educação a Distância (EAD). Essas soluções buscam promover a educação e treinamento em larga escala, a custos mais acessíveis do que os cursos tradicionais, permitindo a publicação de material didático, a utilização de tutoriais, aplicação de exercícios e testes, além de facilitar a comunicação entre aprendizes e mediadores e a apresentação de aulas à distância, e até mesmo a utilização de simuladores.

No sentido de desenvolver facilidades para aprendizes e mediadores, uma diversidade de ambientes tem sido desenvolvida utilizando os recursos disponíveis na *Web*. No contexto das instituições de ensino, aprendizes e mediadores têm percebido, através da disponibilização de notas de aula, listas de exercícios e trabalhos na Internet, as vantagens de incorporar tais recursos ao processo de ensino-aprendizagem, seja ele presencial ou à distância.

Diante deste contexto, surgiram os ambientes virtuais de aprendizagem (AVAs) que representam a grande maioria de ferramentas existentes na web aplicadas a Informática na Educação, e nas subáreas da Educação a distância. Conforme [Xu 2006] os AVA's oferecem várias vantagens em cima de ambientes pedagógicos tradicionais em termos de conveniência e flexibilidade. Não há nenhuma limitação de limite geográfico pelo uso de AVA's. Eles são capazes de localizar os estudantes potenciais em áreas remotas ao redor do mundo a muito baixo custo. Por estas razões, os AVA's se tornam a cada dia uma das ferramentas crescentes mais rápidas em pesquisa de tecnologia educacional e desenvolvimento.

Baseados nesta idéia nos deparamos com os agentes, que representam um novo paradigma para desenvolvimento de aplicações de software, de tal forma que sua utilização tem acontecido nos mais variados tipos de aplicações, como por exemplo: aplicações na área médica, indústria, entretenimento, educação, dentre outras. Agente nada mais é que um elemento autônomo que representa, manipula, troca informações e conhecimento dependendo de um determinado espaço. Segundo Russel e Norvig, “um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.” [Russel e Norvig 2004]. O desenvolvimento de softwares utilizando agentes tem refletido algumas características que os agentes podem possuir (como: autonomia, reatividade, mobilidade, pro atividade, dentre outras). De acordo com [Netto 2006], muitos Sistemas Multiagente são simples plataformas de simulação, entretanto este paradigma é uma opção natural para modelar ambientes que possuem interações críticas, como sistemas de cadeia de suprimentos, suporte a rede de computadores, tráfego urbano e em aeroportos, controle de manufatura, ambientes robóticos, etc.

O desafio é fazer com que estes AVAs se tornem muito mais atraentes e eficazes, e que, o acompanhamento das atividades que estão ocorrendo nos mesmos, ajudem seus

usuários a gerir seu próprio trabalho, facilitando assim, a localização e o acesso a recursos distribuídos em várias comunidades de aprendizagem.

Inseridos no contexto destes ambientes virtuais, encontramos diversas atividades que podem ser realizadas a fim de facilitar o processo de ensino-aprendizagem. Nos Ambientes Virtuais de Aprendizagem, em geral, existe uma ferramenta que auxilia o usuário a marcar seus compromissos chamada Calendário. O Calendário do Moodle permite que o administrador agende eventos e avisos que serão visualizados na página inicial por todos que acessarem. Aos usuários também é permitido agendar informações, mas estas somente serão visualizadas por ele próprio. O problema de agendamento surge também pelo fato de que a maioria dos AVA's não tratam as atividades dos cursos simultaneamente, mas sim um curso de cada vez em suas agendas, o que nos motiva a elaborar um protótipo eficiente no que diz respeito a funcionalidade nos cursos, permitindo que os usuários agendem reuniões para realização das atividades propostas e visualizem o que está agendado.

1.1 Motivação

O objetivo principal de um curso a distância é a aprendizagem do aluno. Nos primeiros modelos de EaD, a metodologia aplicada envolvia grande esforço por parte do aprendiz, na qual basicamente eram disponibilizados materiais para auto-aprendizagem e posteriormente eram aplicados exercícios e provas, provocando um evidente distanciamento na relação professor-aluno. Atualmente, as novidades tecnológicas vêm colaborando para a mudança deste cenário e as atividades dos cursos atuais foram incrementadas com interações subjetivas e colaborativas, tornando os cursos mais interessantes e dinâmicos, aumentando a interação e participação dos alunos, porém aumentando também o trabalho dos professores em analisar tais atividades [Torres 2007].

O elemento motivador deste trabalho se dá nas pesquisas que vem sendo realizadas na área com o intuito de promover melhorias na qualidade de ensino. Existem diversos recursos educacionais que podem auxiliar o professor no processo de ensino-aprendizagem. Entretanto, em pesquisas iniciais notamos que há ainda limitações referentes à ambientes virtuais de aprendizagem tão utilizados como ferramenta educacional para ensino a distância. Por isto, a presente pesquisa nos motiva a trazer melhoria do ponto de vista educacional ao que se refere a ambientes virtuais de aprendizagem, dotando-os de um mecanismo de agendamento de reuniões baseado em SMA (Sistemas Multiagente) que possibilitam a combinação das agendas dos aprendizes e mediadores, com o objetivo da realização das atividades dentro do ambiente virtual de aprendizagem.

1.2 Objetivos

Os objetivos deste trabalho estão divididos em geral e específicos, como se segue:

1.2.1 Objetivo geral

Demonstrar que o uso de sistemas multiagente pode melhorar o estado da arte utilizando o agendamento de reuniões, para a realização de atividades desenvolvidas em um ambiente virtual de aprendizagem.

1.2.2 Objetivos específicos

Como objetivos específicos desta investigação citam-se:

- a) Integrar de conceitos como: Identificar problemas nos agendamentos de atividades em AVA's;
- b) Possibilitar aos usuários de AVA's apoio aos agendamentos de atividades nos cursos de EAD;

- c) Verificar na literatura como o problema de agendamento utilizando sistemas multiagente está sendo tratado;
- d) Construir um protótipo a partir de características definidas;
- e) Validar da abordagem proposta, através da sua utilização em um cenário real.

1.3 Metodologia

A metodologia utilizada neste projeto consiste em um conjunto de atividades típicas às investigações nessa área, como apresentado a seguir:

- a) Levantamento Bibliográfico – na etapa inicial da investigação, além da revisão realizada em trabalhos da literatura relacionada, buscando contextualizar adequadamente a pesquisa;
- b) Estudo das características dos softwares disponíveis para autoria de agentes;
- c) Definição e análise de cenários de uso para ambientes virtuais de aprendizagem que utilizam agentes em problemas de agendamento;
- d) Observação do uso do AVA;
- e) Prototipação de um sistema multiagente para o problema de agendamento;
- f) Validação funcional do protótipo: testes em cursos de que utilizam a plataforma educacional Moodle;
- g) Análise de resultados obtidos.

1.4 Organização do Trabalho

Esta dissertação está estruturada em seis capítulos, sendo o primeiro estas considerações iniciais. O Capítulo 2 é de referencial teórico, onde são apresentados conceitos, uma visão

geral sobre a educação a distância, suas aplicações, tecnologias atualmente utilizadas em ambientes virtuais de aprendizagem e agendamentos.

O Capítulo 3 descreve sobre histórico e aplicações da inteligência artificial, conceitos, propriedades e arquitetura de agentes de *software* e sistemas multiagente.

No Capítulo 4, aborda a construção e descreve a utilização do protótipo em um cenário real de uso, o modelo de dados e sua implementação.

No Capítulo 5, são apresentados os experimentos: o público-alvo, uma breve descrição do curso aplicado, a metodologia utilizada e a análise dos resultados obtidos após a aplicação do modelo de avaliação proposto nesta dissertação.

No Capítulo 6, são destacadas as conclusões, as contribuições desta dissertação, são indicadas as suas limitações e sugestões para trabalhos futuros.

CAPÍTULO 2

2. REFERENCIAL TEÓRICO

Este capítulo descreve conceitos fundamentais importantes sobre o qual se baseia esta pesquisa. Na primeira seção é comentado um breve histórico sobre Educação a Distância (EaD). Na segunda seção, apresentamos o Ambiente Virtual de Aprendizagem como ferramenta de ensino nesta modalidade de EAD. A terceira seção apresenta o AVA Moodle como ambiente escolhido para ser a base do sistema proposto. E a quarta seção, finalmente, apresenta alguns exemplos de Agendamentos com características representativas, formando um conjunto referencial para uso nas próximas etapas (análise e validação da proposta aqui apresentadas).

2.1 Educação a Distância

Embora o tema “educação a distância” esteja constantemente em pauta e sua utilização cada vez mais presente nos dias atuais, a EaD tem uma longa história de experimentações, fracassos e sucessos. Sua origem remonta ao final do século XVIII nas experiências de educação por correspondência, porém teve um desenvolvimento significativo a partir de meados do século XIX através da criação do primeiro curso por correspondência, *Correspondence Colleges*, pelo Sir Isaac Pitman no Reino Unido [Vidal 2002].

Ao longo de sua trajetória, foi fortemente influenciada pelas evoluções tecnológicas de comunicação de massa que se tornavam disponíveis: rádio e televisão, porém, a grande evolução em número de usuários e abrangência dos cursos foi alcançada com o desenvolvimento das tecnologias de comunicação por meio de redes eletrônicas.

Na primeira geração, os primeiros cursos utilizavam os correios como meio de comunicação, por isso eram chamados cursos por correspondência. No Brasil, as primeiras

iniciativas surgiram em 1904, onde instituições privadas ofertavam iniciações profissionais em áreas técnicas.

Com o advento do rádio e da televisão, a segunda geração foi marcada pela realização de programas educacionais e dos telecursos. No Brasil, os pioneiros foram: o Instituto Rádio Monitor, fundado em 1939 e logo após o Instituto Universal Brasileiro - IUB, em 1941, oferecendo cursos técnicos, tais como: técnico de rádio, mecânica, costura e outros do gênero. Porém, a descontinuidade de projetos governamentais também afetou os cursos a distância da época. Um exemplo concreto foi à criação do Movimento de Educação de Base – MEB, cujo objetivo era alfabetizar e apoiar a educação em massa de jovens e adultos – principalmente do Norte e Nordeste - através de escolas radiofônicas, destacando-se pela utilização do rádio como meio de comunicação e um programa de ensino favorável às classes populares. Infelizmente, a repressão política que se seguiu após o golpe de 64 desestruturou o projeto inicial que acabou sendo abandonado.

Atualmente, vive-se a terceira geração, caracterizada pelas comunicações de redes, especialmente a *Internet*. A sua propagação, aliada à crescente velocidade de banda das redes e à contínua evolução tecnológica – *software* e *hardware* – vêm possibilitando a implementação de diversos recursos educacionais (imagens, áudios, vídeos, *chats*, fóruns de debates, simuladores, dentre outros) empregando maior dinamismo e sincronismo aos cursos virtuais que em sua maioria, utilizam os ambientes virtuais de aprendizagem.

2.2 Ambientes Virtuais de Aprendizagem (AVAs)

Um AVA pode ser compreendido como um conjunto de funcionalidades, fornecidas através de suporte tecnológico de hardware e software adequados, que visam a simulação de um ambiente de aprendizagem, formado por comunidades virtuais que cooperam internamente e

entre si, com base na necessidade pedagógica de cada contexto educacional. [Spósito 2008] Trata-se de softwares que auxiliam na montagem de cursos acessíveis pela Internet, auxiliando professores e alunos no gerenciamento conteúdos e na administração destes cursos *on-line*.

Segundo [Santos 2009], existem vários ambientes virtuais baseados na Web. E os AVAs são os mais abundantes nesse meio. O mercado de AVAs no ano de 2005 possuía pelo menos 200 produtos. Muitos desses produtos surgiram ao longo dos anos e se desenvolveram, tornando-se populares e ganhando sempre novas versões. No entanto, outros são apenas citados, mas não houve um real desenvolvimento destes ambientes. Dois históricos sobre o desenvolvimento de AVAs baseados na web podem ser encontrados em (Wikipedia 2009) e (Moodle.org 2009). Baseado nestas duas listas e no surgimento de outros AVAs que se tem conhecimento, principalmente no cenário nacional, elaborou-se a seguinte lista, indicando o ano da aparição da primeira versão do dado ambiente.

Um dos grandes desafios consiste na adequação destes ambientes virtuais no que diz respeito ao agendamento de reuniões para realização de atividades e tarefas que são realizadas dentro dos ambientes virtuais de aprendizagem.

Baseados nisto, propomos um agendamento de atividades apoiados por sistemas multiagente em um ambiente virtual de aprendizagem. Para um melhor entendimento de nossa proposta, criamos um cenário para servir de exemplo. Consideremos um grupo de alunos participantes de um curso a distância, o qual utiliza nestas aulas, um ambiente virtual de aprendizagem. Existem professores e tutores monitorando o grupo e interagindo neste ambiente. Alguns problemas normalmente encontrados dizem respeito ao acompanhamento de agendas quanto ao domínio e visualização dos compromissos. O participante e o próprio mediador não acessam a agenda com a frequência desejada para visualizar as datas das atividades de seu grupo ou, simplesmente, esquecem de algum compromisso nela contido.

Além disso, podemos contar algumas vezes com problemas como: imprevistos, disponibilidade, prioridades e etc, por parte dos membros do curso, o que gera conflitos de horário e/ou data para agendar as atividades. Podemos verificar ainda que um usuário pode desejar acessar as informações sobre os compromissos que possui em outros cursos dentro do ambiente.

2.3 Plataforma Educacional Moodle

O Moodle, acrônimo de '*Modular Object-Oriented Dynamic Learning Environment*', é segundo a documentação disponível no sítio do projeto <http://moodle.org>, um sistema de gerenciamento de cursos [Moodle 2010], e tem sido utilizado por um número crescente de instituições de ensino, pela sua grande facilidade de uso e gerenciamento, pelas várias funcionalidades e recursos pedagógicos disponíveis e pela grande flexibilidade e adaptabilidade às diversas necessidades das instituições de ensino que o utilizam.

Baseado em código aberto, o Moodle é desenvolvido por uma comunidade de usuários distribuída pelo mundo todo e a sua adequação ou 'customização' a um contexto específico de utilização (organização/curso/aplicação) é uma prática comum dentre as instituições que o adotam.

Os usuários podem obter seu código-fonte a partir do sítio do projeto, que funciona como uma central de informações, discussões e colaborações, para que em seguida, possam usá-lo, modificá-lo ou distribuí-lo seguindo apenas os termos estabelecidos pela Licença Pública Geral (GPL) para software livre.

Atualmente encontrado na versão 1.9+ e traduzido para 50 idiomas diferentes, dentre eles, o português, o Moodle é executado em sistemas operacionais que suportam a linguagem

PHP¹ e possui suporte a diversos sistemas de gerenciamento de banco de dados, dentre eles, o MySQL², um dos mais utilizados [Moodle 2010].

Segundo a comunidade de desenvolvedores do projeto, a concepção do Moodle é norteada por uma filosofia de aprendizagem: a teoria sócio-construtivista. “A teoria sócio-construtivista defende a construção de idéias e conhecimentos em grupos sociais de forma colaborativa, uns para com os outros, criando assim uma cultura de compartilhamento de significados” [Moodle 2010].

No entanto, seu foco não é o usuário e sua ação, e sim os cursos que ele suporta, pois ele é essencialmente um sistema de gerenciamento de cursos, com pouca orientação para o suporte a colaboração. Os aprendizes não possuem meios para orientar suas ações, tornando-se apenas atores passivos, que cumprem as atribuições determinadas pelos tutores responsáveis por cada curso.

2.3.1 Principais Entidades do Moodle

Para facilitar a compreensão sobre como o Moodle está estruturado, a seguir é feita uma breve descrição das principais entidades que o compõem:

- a) Cursos: conjunto de cursos ou disciplinas que fazem parte de uma instituição de ensino;
- b) Administradores: usuários que possuem controle total sobre o ambiente ou sistema;
- c) Professores: conjunto de tutores que lecionam em cursos;

¹ PHP é o acrônimo recursivo para ‘PHP: Hypertext Preprocessor’. É uma linguagem de programação de computadores, interpretada, livre e muito utilizada para gerar conteúdo dinâmico na internet (PHP, 2008).

² MySQL é um sistema de gerenciamento de banco de dados de código aberto, que utiliza como interface a linguagem Structured Query Language (SQL), ou, Linguagem de Consulta Estruturada (MySQL, 2008).

- d) Alunos: conjunto de aprendizes ligados aos cursos;
- e) Atividades: conjunto de atividades designadas por administradores e tutores e utilizadas por todos os usuários dentro de um curso. Alguns exemplos: ‘Fórum’, ‘Wiki’, ‘Chat’, ‘Lição’, ‘Tarefa’, ‘Questionário’, ‘Pesquisa de Avaliação’ e ‘Wiki’;
- f) Recursos: conjunto de recursos designados por administradores e tutores e utilizados por todos os usuários dentro de um curso. Alguns exemplos: ‘Link a um Arquivo ou Site’, ‘Rótulo’, ‘Página de Texto Simples’, ‘Página Web’;
- g) Blocos Funcionais: conjunto de mecanismos que agregam diversas funcionalidades, designados por administradores e tutores e utilizados por todos os usuários dentro de um curso. Alguns exemplos: ‘Calendário’, ‘Etiquetas’, ‘Usuários Online’, ‘Mensagens’, ‘Últimas Notícias’, ‘Atividades Recentes’;
- h) Produções Acadêmicas: artefatos produzidos pelos aprendizes ao se utilizarem das atividades e recursos, por exemplo, textos e arquivos multimídia.

A figura 1 apresenta a página principal do Moodle com as suas funcionalidades.



Figura 1 – Página principal do AVA Moodle. Fonte: [MOODLE 2010]

2.3.2 Blocos Funcionais

As diversas funcionalidades existentes, a sua modularidade e facilidades de configuração dão ao Moodle a possibilidade de criação de AVAs bastante flexíveis. A figura 2 apresenta uma instância para o Moodle do AVA do Instituto Federal Amazonas - IFAM, visualizada em um *browser*, programa de navegação na Internet, cujo ambiente acessado por um usuário, possui quatro cursos e alguns blocos funcionais ativados: ‘Usuários Online’, ‘Calendário’, ‘Últimas Notícias’, ‘Próximos Eventos’, ‘Atividades Recentes’, ‘Mensagens’, ‘Categorias de Cursos’.



Figura 2 - Uma instância do Moodle, com um curso e alguns blocos funcionais ativos.

Os blocos funcionais, que irão fazer parte de uma instância do Moodle em um curso, podem ser definidos por administradores do ambiente ou por tutores do curso. Além dos blocos disponíveis para a versão atual do Moodle, muitos outros vem sendo concebidos por desenvolvedores e disponibilizados para *download* no sítio da comunidade do projeto.

Apesar da grande utilidade das ferramentas dos AVAs, os estudantes não costumam acompanhar os compromissos das atividades pré-definidas nas agendas do AVA. Com tantas

opções de ferramentas, disponibilizadas nos AVAs, o aluno pode facilmente ficar perdido, e de algum modo desinteressado no conteúdo e na utilização do ambiente. Algumas vezes por exemplo: não é possível saber quais atividades já foram realizadas e quais precisam ser concluídas, e ainda, qual a melhor forma de se aproveitar todos os recursos do sistema. Para benefício dos estudantes, [Dimitracopoulou 2005] afirma que, uma análise dos sistemas existentes de colaboração mostra que uma série de ferramentas e funções são projetadas e implementadas a fim de facilitar e melhorar o apoio ao processo de aprendizagem colaborativa.

Um Sistema Gerenciador de Cursos, “Learning Management System” (LMS) é um ambiente computacional que permite ao professor gerenciar um curso a distância, provendo o planejamento, implementação e gestão do aprendizado, permitindo inclusive o uso em cursos semipresenciais ou para a publicação de materiais que complementem os cursos presenciais.

Para este trabalho, destacamos o Modular Object Oriented Dynamic Learning Environment (Moodle) que é um AVA, regido pela General Public Licence (GPL), desenvolvido inicialmente por Martin Dougiamas. O Moodle [Moodle 2010], é um sistema gerenciador de cursos em que os dados são armazenados em um banco de dados MySQL ou PostgreSQL³, com interface PHP (Personal Home Page). Foi desenvolvido sob a teoria construtivista social, a qual defende a construção de idéias e conhecimentos em grupos sociais de forma colaborativa, uns para com os outros, criando assim uma cultura de compartilhamento de significados.

[Pulino 2005] menciona como vantagem em relação a outros sistemas gerenciadores de cursos, que o Moodle é um software livre que apresenta todas as funcionalidades e objetivos educacionais requeridos em um LMS. Esse importante fato implica na implantação sem qualquer ônus e a possibilidade de pesquisas acadêmicas relacionadas com o Moodle,

³ Disponível em: <<http://www.postgresql.org>>

permitindo melhorias no sistema, adequando-o às necessidades pedagógicas e operacionais, personalizando o ambiente segundo as necessidades de cada curso ou disciplina.

Segundo [Campana 2008], um ambiente de aprendizagem deve fornecer suporte às questões relativas à interatividade, cooperação, colaboração e mediação, além de facilitar o uso e a construção da aprendizagem de seus usuários. Essa seção busca explorar algumas necessidades encontradas em sistemas de educação à distância em relação a esses vários aspectos e definir formas de atendê-las.

Neste trabalho, optou-se por utilizar o ambiente Moodle, pois ele é uma plataforma educacional voltada ao gerenciamento de cursos ou disciplinas para a composição de AVAs, como solução tecnológica para o suporte ao modelo de agendamento. Desta forma, ambientes que forem concebidos por meio do Moodle, poderão usufruir do modelo de agendamento e seus usuários poderão utilizá-lo como mecanismo para monitorar o que vem acontecendo em seus cursos ou disciplinas.

2.3.3 Calendário

O Calendário do Moodle permite que o administrador agende eventos e avisos que serão visualizados na página inicial por todos que acessarem, compromissos como provas e atividades por exemplo. Aos usuários também é permitido agendar informações, mas estas somente serão visualizadas por ele próprio.

Estruturação dos eventos do calendário e o seu funcionamento obedecem a hierarquia dos eventos disponíveis. Um evento pode ser classificado em quatro níveis, conforme observamos na Figura 3:



Figura 3 – Eventos no Calendário do Moodle.

Global: Os eventos deste nível podem ser vistos por todos os usuários do site, ignorando o curso ou grupo onde ele está. Esse tipo de evento só poderá ser criado pelo Administrador.

Curso: Os eventos deste nível só serão visualizados por quem estiver dentro do curso onde o evento foi criado. Esse tipo de evento só poderá ser criado pelo Administrador.

Grupo: Os eventos deste nível só serão visualizados por quem estiver incluído no grupo selecionado. Esse tipo de evento só poderá ser criado pelo Administrador.

Usuário: Os eventos deste nível só serão visualizados por quem o criou (o mesmo se aplica ao administrador). Esse tipo de evento é criado pelo próprio usuário, não importando sua função (aluno, professor, administrador, etc).

Há duas opções de criação do evento, ele pode ser criado através do calendário geral (da *home*) ou de dentro de um curso (visualizado somente por alunos dentro de um certo grupo). Apesar desta descrição dos eventos, o calendário do Moodle tem poucas funcionalidades comparando com o protótipo apresentado nesta pesquisa. Mesmo assim, o calendário serviu de base para o desenvolvimento do protótipo por estar relacionado ao agendamento.

2.4 Agendamento

De acordo com o Dicionário Houaiss da Língua Portuguesa [Houaiss e Villar 2009], a palavra agendamento vem do verbo Agendar e significa: Marcar a data da realização de algo; Incluir (compromisso) em agenda. Neste trabalho é proposto o agendamento de reuniões para realização de atividades dentro de um AVA com um monitoramento realizados por agentes.

De acordo com [Martins 2010], os problemas de programação de horários (Timetabling) são problemas de alocação que constituem uma área de intensa atividade de pesquisa com aplicações em diversos contextos. A criação de tabelas de horários para jogos e a organização de escalas de horário de trabalho, são exemplos de situações que descrevem esse tipo de problema em ambiente esportivo e empresarial, respectivamente.

Uma característica marcante, compartilhada por diversas variantes de problemas de timetable é a sua natureza fortemente associativa, ou seja, soluções para tais problemas são obtidas a partir de associações entre um número de eventos (ex.: aulas, palestras, jogos), e um número limitado de recursos (ex.: tempo, pessoas, salas). Sendo o objetivo principal, determinar como utilizar os recursos da melhor forma possível.

A elaboração de quadros de horários para agendar uma atividade, apesar de bem definida, não é uma tarefa simples, e se torna mais complicada à medida que crescem o número de associações e as limitações impostas aos recursos. Um exemplo clássico de problema de timetable é encontrado em instituições de ensino, quanto ao agendamento de aulas. Para esta variante, existe uma grande quantidade de trabalhos desenvolvidos até mesmo competições, como a International Timetabling Competition, que têm como objetivo estimular o crescimento e a popularização de soluções automatizadas (software) para o problema. Na Figura 4 podemos visualizar um sistema de agendamento que utiliza timetabling.

The screenshot shows the 'Any School Day.EVE - TTMaker' software interface. It features a menu bar (File, Edit, View, Event, Email, Help) and a toolbar with various icons. The main window is divided into several sections:

- Venue Name, Location, Facilities, Styles:**

Room 1	Block A	Projector	
Room 2	Block A	TV	
Science Room 1	Block A	Science	
- Participant Name, Usual Role, Address, Phone, Fax, Email:**

Alex Harris	Teacher				
Anita Buckley	Teacher				
Gordon Ward	Teacher				
Nev Watson	Teacher				
Bob Harrison					
Cheryl Smith					
Eileen Steel					
Harriet Davies					
Ian Davies					
Ian Robinson					
Jason Stewart					
- Group Name, Facilities, Members:**

Advanced Maths		Nev Watson, Petr...
Art	Projector	Alex Harris, Mary...
English 1A		Alex Harris, Cher...
Home Economics 1B	Projector	Anita Buckley, Iar...
Journalism 2	TV	Gordon Ward, Ha...
Maths 1A		Gordon Ward, Iar...
Science 1A	Science	Anita Buckley, Tim...
Social Studies		Alex Harris, Bob H...
Sports Analysis 1	TV	Gordon Ward, Jol...
- Every Day Timetable:**

	Room 1	Science Room 1	Room 2
9:00am			
9:50am	English 1A	Science 1A	Sports Analysis 1
10:40am	Art	Maths 1A	Advanced Maths
11:30am	Home Economics 1B	Social Studies	Journalism 2
12:20pm	Social Studies	Maths 1A	Advanced Maths
1:10pm			
	Home Economics		

At the bottom left, it says '9 groups, 1 selected'. At the bottom right, there are buttons for 'NUM' and 'SCRL'.

Figura 4 – Timetabling apresentado em um sistema de agenda.

Contudo, apesar desse esforço, ainda existe uma grande distância entre o que é desenvolvido e o que, de fato, é utilizado na prática. Cada instituição possui peculiaridades que, em certos casos, impedem, ou, no melhor dos casos, dificultam a utilização de softwares não específicos. Como raramente algum esforço é estabelecido para o desenvolvimento de soluções específicas esse distanciamento tende a aumentar. Diversas técnicas foram desenvolvidas, estudadas e adaptadas ao longo dos anos, visando à solução eficiente de problemas de programação de horários. As primeiras baseavam-se principalmente em heurísticas desenvolvidas para a utilização na resolução manual, porém, com o aumento da importância de tais problemas, técnicas mais aprimoradas começaram a ser utilizadas.

Duas categorias podem ser identificadas quanto à classificação de tais técnicas. A primeira, composta por métodos exatos, tem como objetivo principal encontrar as melhores

soluções para o problema. Enquanto que a segunda categoria, composta por métodos heurísticos e meta-heurísticos, tem como objetivo a obtenção de boas soluções, não necessariamente as melhores.

O problema de agendamento semanal de aulas, abreviadamente PASA, representa, neste trabalho, os problemas de programação de horários em instituições de ensino e pode ser definido como a necessidade de se relacionar disciplinas, horários, professores, estudantes e salas, de modo que nenhum dos envolvidos esteja associado a mais de uma atividade em horários simultâneos.

Uma Aula, no contexto do PASA, é definida como uma associação entre uma disciplina, um professor e um horário. Esses dois tipos de restrições são de fundamental importância quando métodos de otimização são utilizados para a resolução do PASA. As restrições fortes são responsáveis por restringir o conjunto de soluções factíveis, limitando, assim, a quantidade de soluções a serem avaliadas. Já as restrições fracas, são utilizadas para quantificar a qualidade das soluções e seguem critérios de avaliação mais específicos a cada instância do problema.

Em contraposição à importância que possuem em processos de otimização, as restrições consideradas em cada instância do problema também contribuem para a complexidade de resolução do mesmo. Desta forma, a escolha das restrições se torna uma etapa importante da criação do método de resolução do problema. Estes problemas são, em sua maioria, pertencentes à classe de problemas NP-Completo, ou NP-Difícil (problemas de otimização), um cuidado adicional é necessário na escolha das restrições e resolução do problema, com o objetivo de favorecer, o quanto possível, o tempo de execução do método a ser utilizado.

Os problemas de alocação de horários são, em sua maioria, problemas pertencentes à classe NP-Difícil, desta forma, solucioná-los através da utilização de métodos exatos ainda é

um desafio para instâncias de tamanho razoável. De acordo com esse aspecto, a utilização de meta-heurísticas ou métodos híbridos tornam-se boas opções.

[Berger *et al* 2008], considera o agendamento com a seguinte definição: um conjunto de pessoas, onde alguns dos participantes desejam agendar uma reunião. Trata o problema de agendamento de reuniões, considerando o tempo e o espaço. Os participantes estão localizados em locais diferentes e necessitam encontrar um ponto em comum para a reunião ser realizada. Os participantes têm horários individuais, e por esse motivo foi pensada uma estratégia para agendar, conforme podemos observar na Figura 5. Cada usuário recebe um mapa com o local da reunião e dois caminhos mais curtos. O primeiro caminho, mostrado em vermelho, é o caminho da reunião atual para o local novo encontro. O segundo caminho, mostrado em verde, é o caminho a partir da localização nova reunião para a localidade próxima reunião.

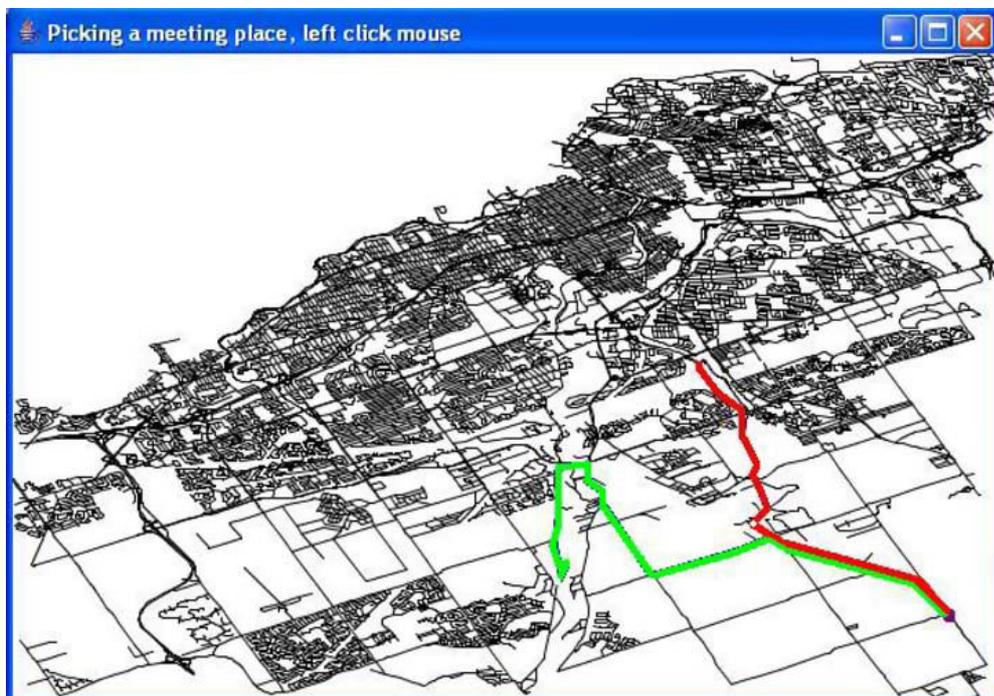


Figura 5 – Proposta de Berger *et al* 2008, apresentando o melhor percurso para realizar o agendamento.

O objetivo da proposta realizada por [Berger *et al* 2008] foi obter algoritmos eficientes para resolver problemas desta programação, com o objetivo de encontrar soluções aproximadas, e quando necessário implementar essas soluções e integrá-las em aplicações que permitam que os usuários conectados em uma rede, agendem reuniões.

Em contraste à definição de [Berger *et al* 2008], que inclui a solução do agendamento algoritmos eficientes em sua definição, [Yang 2009] afirma que a tarefa de organizar um reunião pode ser melhorada através da redução da carga de trabalho do agendamento, tornando a logística dessa reunião mais eficiente através do uso de um *mobile (iPhone)* onde agentes tomam decisões em nome do usuário. Nesta pesquisa foi elaborada uma árvore que apresenta a estrutura de uma agendamento de reunião, como podemos visualizar na figura 6.

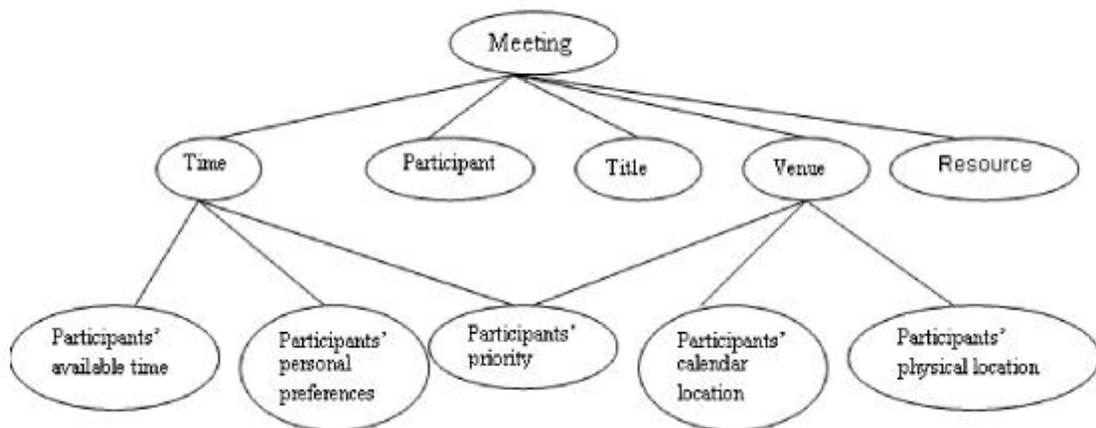


Figura 6 – Árvore do Agendamento de Reuniões proposta por Yang 2009.

Baseada nesta idéia, [Yang 2009] propôs o BMO (*Business Meeting Organizer*), que é um sistema multiagente de agendamento de reuniões, que automatizam decisões de hora e local. O BMO não está baseado em calendários, mas em agentes de software que servem como representantes de cada usuário, negociando um melhor momento para uma reunião em

particular, levando em consideração o contexto, a disponibilidade e as preferências pessoais quanto ao momento de agendar a reunião.

Outra definição clássica ocorre em [Crawford 2005], cuja principal idéia de agendamento é permitir que agentes de software personalizados realizem agendamento de reuniões. Esta pesquisa tem o objetivo de reduzir a carga cognitiva diária no computador dos usuários. [Crawford 2005] defende a idéia de que o agendamento de uma reunião pode ser um processo demorado exigindo muitas mensagens de correio eletrônico a serem compostas e lidas, e muitas vezes as reuniões podem dar lugar a novas reuniões. Potencialmente, os agentes de software podem remover essa carga totalmente, comunicando-se uns com os outros para agendar reuniões. Como usuário tem a posse de seus próprios calendários e preferências pessoais sobre o agendamento de reuniões, faz sentido abordar este problema de forma distribuída. Esta idéia foi aplicada numa analogia aplicada ao contexto de pequeno time de futebol de robôs (onde uma câmera e um computador *off-board* permitiram a coordenação da equipe de planejamento). O problema de negociação de reunião tem uma série de importantes características em comum com o pequeno time de futebol de robôs. Em ambos os domínios, o espaço de estratégias disponíveis é enorme e não é possível para os agentes adaptação on-line sem considerar o espaço de toda a estratégia. Além disso, o ambiente em ambos os domínios é dinâmica, os modelos dos "adversários" estão aprendendo o desconhecido e estar on-line é necessário para um bom desempenho. O objetivo foi mapear as execuções terminologia de futebol de robôs para o problema do agendamento de reuniões. As execuções correspondem a estratégias de negociação concluída, o adversário corresponde ao agente e o agente de aprendizagem está a negociar com o conjunto de estratégias de negociação disponíveis para o agente de aprendizagem. O agente tem de se adaptar a estratégia de aprendizagem de seleção para cada um dos diferentes agentes que negocia com outros simultaneamente. Em geral, foi proposto um agente para maximizar utilidade do usuário, facilitando assim o agendamento de

reuniões, minimizando o tempo de negociação do processo de acordo com as preferências de seus usuários.

No artigo proposto por [Eijk 2002] são utilizados agentes para agendamentos de viagens para usuários específicos, que utilizam a Internet, LAN (Local Area Network), GPRS (General Packet Radio Service) e outros serviços, para realizar a comunicação para o agendamento nestas viagens, para localização dos participantes ou para combinar as datas dos eventos das viagens; onde os agentes são responsáveis por este agendamento, conforme podemos observar na Figura 7. [Eijk 2002] diz que, embora os agentes representem o usuário e acessem os seus perfis, os usuários podem decidir não deixar os agentes compartilharem tal informação, em determinada tarefa, como por exemplo, expor toda a informação do usuário relacionada a outros agentes, e negociação entre agentes quanto a privacidade de usuário para escolha melhor sobre local e preferências quanto ao agendamento.

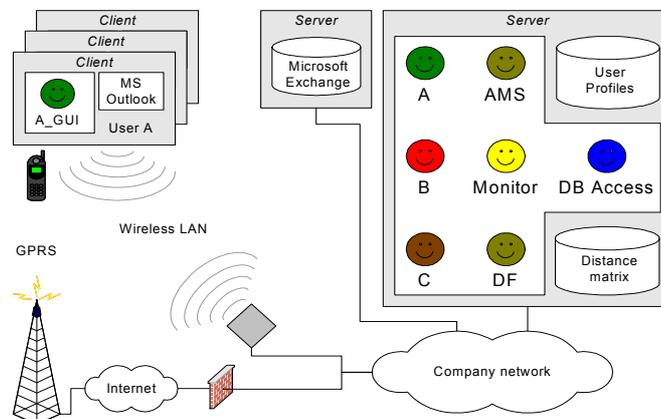


Figura 7 - Arquitetura utilizada por [Eijk 2002] em Scheduler Agent System for Traveling Users.

[Xu 2006] utiliza agentes inteligentes para dar suporte em ambientes virtuais de aprendizagem com o objetivo de analisar o comportamento proativo dos usuários dentro do ambientes com relação ao aprendizado; embora não realize um agendamento dentro do ambiente como podemos visualizar na Figura 8.

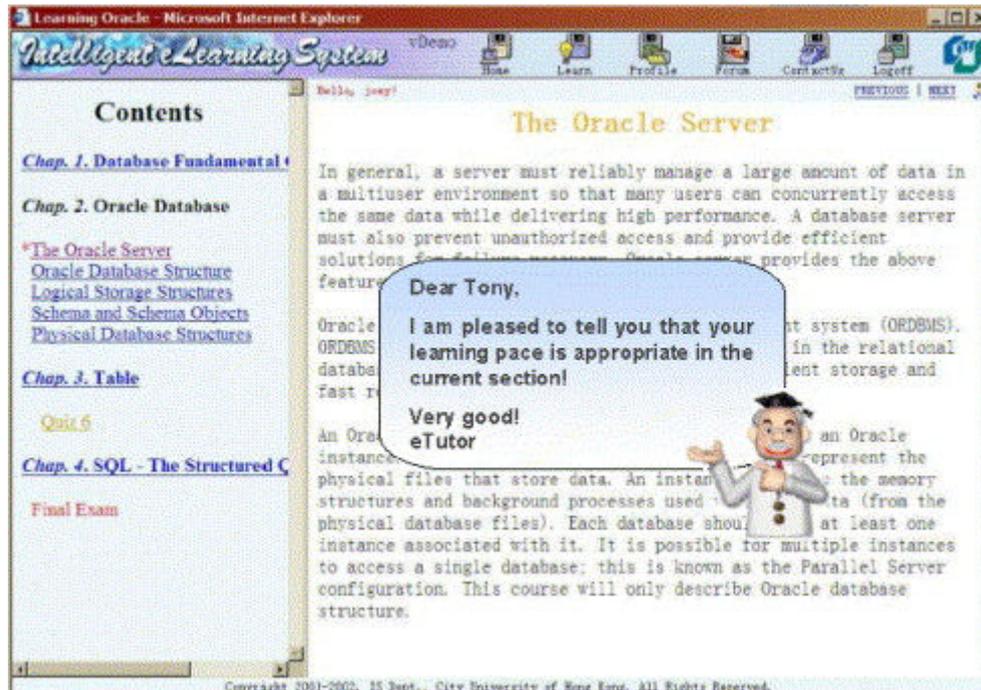


Figura 8 – Agentes Inteligentes dando suporte no AVA. [Xu 2006]

[Campana 2008] diz que para ser realizado um agendamento são necessários mecanismos para o agendamento e registro de reuniões, considerando o *quorum* mínimo e os participantes obrigatórios, bem como, o aviso sobre as reuniões agendadas pelo mediador. Os requisitos associados às facilidades de agendamento segundo [Campana 2008] são:

- agendar reuniões síncronas levando em consideração as disponibilidades de cada participante e, principalmente, dos participantes obrigatórios, quando for o caso;
- fornecer mecanismos para tratar a questão de *quorum* mínimo para os encontros síncronos, conforme os critérios estabelecidos pelo mediador;
- permitir ao participante definir ações para os atrasos nas agendas ou sobre a aproximação de datas importantes. Esta pesquisa apresentou o resultado da agentificação de um Ambiente Virtual de Aprendizagem, como podemos visualizar na Figura 9.

Meus Dados | Meus Grupos

::: Santinho Ferreira de Souza > Produção de texto e ensino 2008/1

Agentes do Ambiente

Agente Monitorador de Escaninhos dos Usuários dos Grupos

Avisar no intervalo de dia(s)

por E-mail e/ou Mensagem Instantanea

a inclusão de novos arquivos ou pastas que contenham conteúdo

Avisar independente do conteúdo.

É uma regra do grupo.

Adicionar Regra

REGRAS PESSOAIS

Intervalo de Alerta	Tipo Aviso	Conteúdo	Tipo Analise	Ultima Execução
1 dia(s)	E-mail	Qualquer...	-	19/08/2008 17:45:04

REGRAS DO GRUPO

Intervalo de Alerta	Tipo Aviso	Conteúdo	Tipo Analise	Ultima Execução
Não existem regras cadastradas para este grupo.				

1

2

3

4

Figura 9 – Tela de configuração do agente no AVA. [Campana 2008]

Foi acrescentada uma camada multiagente a um AVA para a monitoração e suporte aos participantes e mediadores dos grupos (ou cursos) desse ambiente, tornando suas atividades diárias mais simples, diminuindo o trabalho cognitivo e manual da utilização das ferramentas pelos usuários. Ela está relacionada a este trabalho, no que diz respeito a agendamento de reuniões e atividades síncronas, levando em consideração as disponibilidades pré-cadastradas, o que poderemos ver com detalhes no Capítulo 4 desta pesquisa.

Por sua vez [Cuesta 2004], apresenta um estudo de um caso experimental para implementação de um sistema multiagente programador de reuniões, que permite aos usuários

corrigir reuniões automaticamente com base nas agendas pessoais dos participantes, que foi apresentado através de um diagrama de classes, como é apresentado na Figura 10.

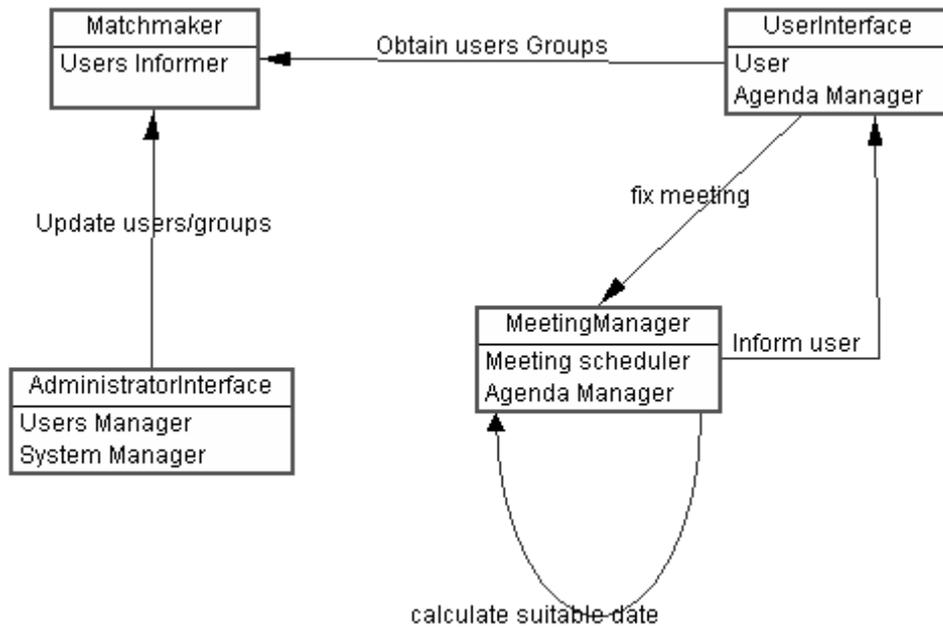


Figura 10 – Diagrama de Classes dos Agentes. [Cuesta 2004]

O objetivo deste trabalho foi colocar em prática uma metodologia orientada a agentes e utilizar uma plataforma de desenvolvimento utilizando o JADE (Java Agent Development Framework). O principal objetivo do sistema é a implementação de um programador de reuniões que ajuda os usuários dentro de uma organização a corrigir reuniões entre eles. A fim de alcançar este objetivo geral, o sistema fornece uma maneira de: gerenciar usuários, manter uma agenda para cada usuário, organizar reuniões e ao final informar os usuários. Uma vez que uma reunião foi estabelecida com sucesso, todos os usuários devem ser informados e sua agenda pessoal será atualizada em conformidade. O usuário será notificado de qualquer modificação que comprometam uma reunião na qual ele está envolvido.

O JADE (Java Agent Development Framework), conceitualiza um agente como um processo independente e autônomo que possui uma identidade, possivelmente persistente, que

requer comunicação com outros agentes para realizar sua tarefa. Logo, apenas define capacidades relacionadas à interação e a comunicação entre agentes. Além disso, ele apresenta em seus exemplos um agente de troca de mensagens que mostra mobilidade dos agentes e agendamento de múltiplas tarefas. Este exemplo de agente é o *Meeting Scheduler*, que se trata de uma aplicação de amostra, que usa JADE para troca de mensagem e para a implementação dos protocolos de interação, onde implementa um agente de agenda de reunião que ajuda um usuário programando reuniões com outros usuários, conforme é apresentado na Figura 11.

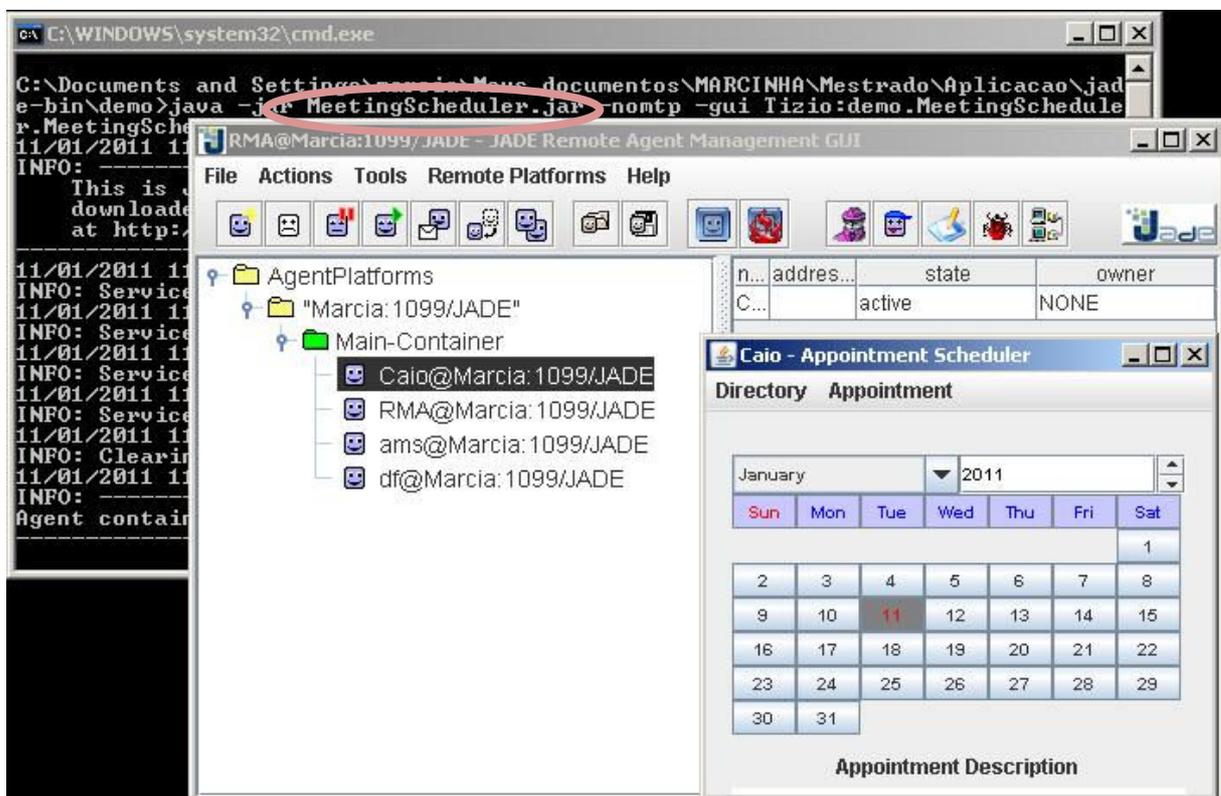


Figura 11 – Agente Meeting Scheduler.jar em execução - JADE

É uma aplicação de amostra baseada em FIPA (Foundation of Intelligent Physical Agents), que também contribui para este trabalho. Estes exemplos estão dispostos no próprio JADE, que encontra-se disponível para instalação em: <http://jade.cselt.it/>.

Todos os trabalhos correlatos citados estão relacionados com esta pesquisa: ora no que diz respeito a agendamento de reuniões, ora a sistema multiagente em AVA's, ora a agentes para marcar uma data para realização de alguma atividade. Estes trabalhos foram citados nesta pesquisa principalmente pela questão conceitual contribuindo com o amadurecimento da idéia proposta, pelo fato de não termos encontrado na literatura agentes específicos para agendar reuniões dentro de um AVA. Apresentamos na tabela 1 um resumo das propostas de cada trabalho relacionado ao nosso. O que nos trouxe maior segurança a nível da implementação, foram os exemplos de agentes do próprio JADE, que serviram de modelo na implementação desta aplicação.

Tabela 1 – Apresentação do resumo de cada trabalho relacionado.

TRABALHO	RESUMO
A meeting scheduling problem respecting time and space - Berger	Algoritmos eficientes para resolver problemas de agendamento encontrando soluções aproximadas, que permitam que os usuários conectados em uma rede, agendem reuniões.
Multi-agent Meeting Scheduling Using Mobile Context - Yang	Propõe o BMO (<i>Business Meeting Organize</i>), que é um sistema multiagente de agendamento de reuniões, que automatiza decisões de hora e local.
Learning to Select Negotiation Strategies in Multi-Agent Meeting Scheduling - Crawford	Defende a idéia de que o agendamento de uma reunião pode ser um processo demorado exigindo muitas mensagens de correio eletrônico a serem compostas e lidas, e muitas vezes as reuniões podem dar lugar a novas reuniões. Potencialmente, os agentes de software podem remover essa carga

	totalmente, comunicando-se uns com os outros para agendar reuniões.
Implementation of a Scheduler Agent System for Traveling Users - Eijk	São utilizados agentes para agendamentos de viagens para usuários específicos, que utilizam a Internet.
Intelligent agent supported personalization for virtual learning environments - Xu	Utiliza agentes inteligentes para dar suporte em ambientes virtuais de aprendizagem com o objetivo de analisar o comportamento proativo dos usuários dentro do ambientes com relação ao aprendizado.
Agentes para Apoiar o Acompanhamento das Atividades em Ambientes Virtuais de Aprendizagem - Campana	Apresenta uma camada multiagente no AVA para monitorar e dar suporte as atividades realizadas pelos usuários.
Developing a meeting scheduler multiagent system: an experimental case study – Cuesta	Apresenta um estudo de um caso experimental para implementação de um sistema multiagente programador de reuniões.
Meeting Scheduler - Jade	Este é um exemplo de aplicativo que usa JADE para troca de mensagens e para a execução dos protocolos de interação. Ele implementa um agente planejador de reunião que ajuda o usuário no agendamento de reuniões com outros usuários.

O que diferencia o nosso trabalho destes supracitados é que em nossa abordagem, quando os agentes são ativados a fim de monitorar o agendamento de reuniões, por exemplo, estes registros são realizados em um banco de dados mantendo desta forma, os usuários atualizados das disponibilidades de horário que já cadastrou e dos agendamentos que realizou.

Este fator é o que diferencia nossa pesquisa dos trabalhos relacionados. Uma versão inicial desta pesquisa está detalhada em [Pimenta e Netto 2009].

O presente capítulo também apresentou algumas características do Moodle, principalmente, com relação a visualização do ambiente onde está inserido o protótipo. Mais adiante, no Capítulo 4, será apresentado um protótipo para a abordagem de monitoramento proposta neste trabalho construído a partir do Moodle.

CAPÍTULO 3

3. AGENTES DE SOFTWARE

Este capítulo apresenta as características principais dos agentes e o modelo conceitual para a abordagem proposta para o agendamento em AVAs.

3.1. Inteligência Artificial

Há mais de 2000 anos filósofos têm tentado entender como o processo de aprendizado e racionalidade pode ser realizado, apontando que o estudo da inteligência seja uma das mais antigas disciplinas. Entretanto a Inteligência Artificial (IA) teve seu início formal apenas em 1956, e desde então vem englobando diversas áreas, desde questões genéricas como percepção e raciocínio lógico até tarefas específicas como jogar xadrez, provar teoremas matemáticos, diagnosticar doenças, dentre outras [Russel e Norvig 2004].

A IA é normalmente utilizada para resolver problemas de grande complexidade ou quando não existe uma solução viável pela computação convencional e requerem modelagem do comportamento de um ser inteligente ou quando as tarefas são melhores executadas por seres humanos que por máquinas.

A inteligência artificial pode ser definida, dentre outras formas, como: o estudo das faculdades mentais através do uso de modelos computacionais; ou a automação de atividades associadas com o processo de pensamento humano, tais como: tomada de decisões, resoluções de problemas e aprendizado; ou ainda, o estudo de processos computacionais que tornam possível perceber, raciocinar e agir [Winston 1992].

A IA está organizada em quatro categorias: sistemas que agem como humanos; sistemas que pensam como humanos; sistemas que pensam racionalmente e sistemas que agem racionalmente.

Um agente é uma entidade de software que exibe um comportamento autônomo¹ e proativo orientado a objetivos, que está situado em algum ambiente sobre o qual é capaz de realizar ações para alcançar seus próprios objetivos de projeto e a partir do qual percebe alterações [Wooldridge 2009].

Wooldridge [Wooldridge 2009] visualiza um agente como sendo uma entidade com capacidade de resolução de problemas encapsulada. Inserido nesta visão, define o agente como tendo as seguintes propriedades:

- autonomia² - executam a maior parte de suas ações sem interferência direta de agentes humanos ou de outros agentes computacionais, possuindo controle total sobre suas ações e estado interno;
- habilidade social - por impossibilidade de resolução de certos problemas ou por outro tipo de conveniência, interagem com outros agentes (humanos ou computacionais), para completarem a resolução de seus problemas, ou ainda para auxiliarem outros agentes. Disto surge a necessidade de que os agentes tenham capacidade para comunicar seus requisitos aos outros e um mecanismo decisório interno que defina quando e quais interações são apropriadas;
- capacidade de reação - percebem e reagem à alterações no ambientes em que estiverem inseridos.

¹Segundo Wooldridge [Wooldridge 2009], há um consenso geral de que autonomia é a idéia central da noção e agência. Entende-se autonomia como a capacidade do agente agir por seus próprios objetivos sem a intervenção de outrem [Russel 1995].

²Para ter autonomia, o agente deve ter um certo grau de inteligência, o que capacita-o a sobreviver em um ambiente dinâmico e por vezes não benigno. Segundo Russell [Russell 1995], o comportamento de um agente pode ser baseado tanto em sua própria experiência, quanto em seu conhecimento embutido.

- capacidade pró-ativa - agentes, do tipo deliberativo, além de atuar em resposta às alterações ocorridas em seu ambiente, apresentam um comportamento orientado a objetivos, tomando iniciativas quando julgarem apropriado.

Agentes operam e existem em um ambiente, que pode conter ou não outros agentes. Em algumas situações, agentes podem executar tarefas por si próprios, mas, em outras, têm a necessidade do auxílio de outros agentes. Isso acontece a todo o momento no mundo real, pois quando alguém tem a necessidade de resolver algum problema que envolva algum conhecimento ou habilidade que não possua, necessita o auxílio de outra pessoa apta a fazê-lo.

3.1.1 Sistemas que agem como humanos

O teste de Turing, proposto pelo cientista britânico Alan Turing na década de 50, foi elaborado para prover uma definição operacional satisfatória de inteligência. Para Turing, comportamento inteligente é a habilidade para realizar atividades humanas em todas as tarefas cognitivas suficientes para enganar um interrogador. O teste consiste em colocar um humano fazendo interrogatório através de um teclado para outra entidade que ele não esteja visualizando, se o interrogador não conseguir distinguir se a entidade é outro humano ou um computador, o sistema é aprovado pelo teste.

Nesta categoria o computador deverá ter as seguintes capacidades: habilidade para se comunicar em um idioma; armazenar conhecimento; usar o conhecimento armazenado para responder questões e alcançar novas conclusões; e se adaptar a novas circunstâncias.

3.1.2 Sistemas que pensam como humanos

Para ensinar um sistema a pensar como humanos existe primeiramente, o complexo desafio de entender o funcionamento da mente humana. O sistema tem que ter a habilidade de se

comportar de forma similar a um humano diante do problema apresentado, agindo para resolver um impasse ou um problema da mesma forma que o humano resolveria e conseguindo os mesmos resultados que uma mente humana teria.

São sistemas desenvolvidos para automação de atividades associadas com o pensamento humano, tais como a tomada de decisões, aprendizagem ou resolução de problemas. Um bom exemplo é o GPS – *General Problem Solver*, projetado em 1963 por Newell e Simon, que tinha o objetivo de simular o pensamento humano. Mapeando o estado atual e o estado objetivo, o GPS tenta minimizar a diferença entre os dois estados encontrando o melhor caminho para a solução.

3.1.3. Sistemas que pensam racionalmente

Baseado nos princípios da lógica formal, onde é possível prover uma notação para representar todos os objetos e seus relacionamentos, sistemas que pensam racionalmente são capazes de encontrar uma solução (se existir), dado um problema com notação lógica.

Nesta categoria, a IA utiliza padrões estruturados de argumentos baseados nos silogismos de Aristóteles [Aristoteles 2009], como por exemplo: “Todos os homens são mortais; Sócrates é um homem; então Sócrates é mortal.”

Esta abordagem possui dois problemas: a dificuldade de transformar conhecimentos informais em termos formais requeridos pela notação lógica e a grande diferença entre ter habilidade em resolver o problema na teoria e conseguir resolvê-lo na prática.

3.1.4 Sistemas que agem racionalmente

Nesta abordagem, a inteligência artificial é o estudo e construção de agentes racionais que percebam o ambiente e desenvolvam ações que permitam que o(s) objetivo(s) seja(m) atingido(s). Uma forma de agir racionalmente é raciocinar logicamente através de inferências para concluir que uma dada ação realizará um objetivo.

Através de contínuas aplicações de inferências, os sistemas são capazes de decidir e otimizar ações que maximizem as chances do ambiente ficar em estados que satisfaçam os objetivos propostos.

3.1.5 Aplicações

A Inteligência Artificial está disseminada em diversos segmentos da computação – *internet*, redes e sistemas distribuídos, banco de dados, engenharia de *software*, projeto e análise de *hardware*, robótica, computação gráfica, jogos, simuladores, dentre outros – e em várias áreas, como: arquitetura, artes, astronomia, finanças, negócios, engenharia, direito, medicina, forças armadas, telecomunicação e obviamente informática. Seguem alguns exemplos:

3.1.6 Controle de robôs

Outra aplicação de relevância nos dias atuais são os sistemas que manipulam robôs, seja para realizar tarefas na área médica como cortes cirúrgicos milimétricos que exigem extrema precisão, seja para realizar atividades mecânicas com inúmeras repetições na área industrial ou até mesmo na área da segurança, enfrentando ambientes inóspitos e atividades perigosas como desarmamento de bombas.

3.1.7 Jogos e simuladores

Também presente no campo do entretenimento, a inteligência artificial modela os ambientes físicos e simula os comportamentos e personalidades dos personagens na tentativa de tornar o mundo virtual cada vez mais parecido com o mundo real e assim, fornecer uma interessante interação com os usuários. Um exemplo clássico e de sucesso mundial são os jogos de futebol virtual, onde cada equipe, time ou seleção, possui características, estratégias e habilidades semelhantes aos times e seleções reais, inclusive com os próprios jogadores daqueles países.

3.1.8 Recomendação de produtos

Outra aplicação bastante presente nos dias atuais, principalmente utilizada pelos sítios de compras na *internet*, são os sistemas desenvolvidos especialmente para observar o comportamento de compras dos usuários e através de análise comparativa com outros perfis de usuários, fazer recomendações personalizadas de produtos afins, potencializando assim novas vendas para a empresa.

3.2 Agentes

Agente de *software* é uma entidade que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que seja capaz de intervir no seu ambiente, de forma flexível e inteligente, sem requerer intervenção humana [Bradshaw 1997].

Agentes possuem sensores que observam o ambiente e atuam com autonomia baseado em seus objetivos, explícitos ou implícitos, para interferir em algo no ambiente. Enquanto o agente humano possui os cinco sentidos para perceber e interagir com o ambiente, o agente de *software* pode fazer uso de câmeras, sensores infra-vermelho, laços indutivos, teclado e diversos outros dispositivos para monitoramento e outra variedade de mecanismos para interação, como por exemplo: vídeo, auto-falante ou impressora.

Um agente racional deve executar uma ação correta, fundamentada a partir das suas percepções do ambiente, porém o sistema pode oferecer várias ações corretas e neste caso o agente deve ter inteligência para escolher a melhor ação, dentre as possíveis, que torne o ambiente mais adequado com o objetivo estipulado e que maximize seu nível de sucesso [Russel e Norvig 2004].

Para que um agente possa realizar uma ação racional é preciso considerar quatro importantes fatores: possuir níveis de sucesso bem definidos; ter capacidade para observar seqüências de percepções; ter conhecimento sobre o ambiente e conhecer quais ações pode realizar.

A Figura 12 ilustra de forma resumida o processo de interação de um agente inteligente com o seu ambiente.

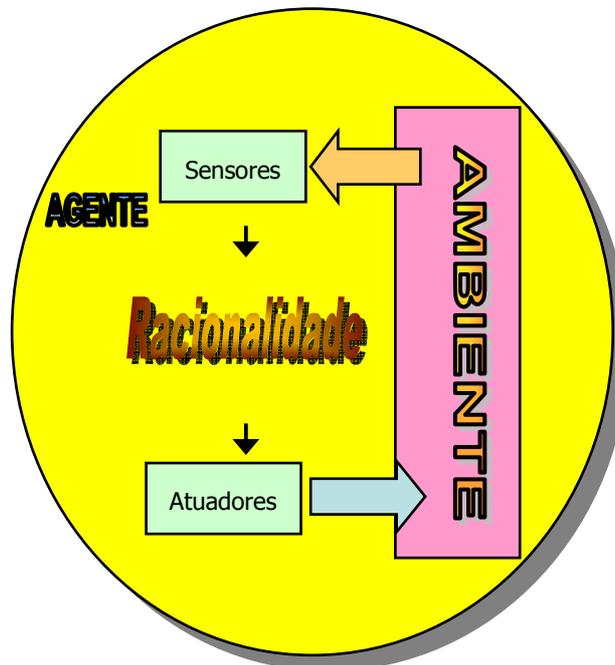


Figura 12 – Processo de Interação do Agente com o Ambiente. Fonte: Apud [Russel e Norvig 2004].

3.2.1 Propriedades dos agentes inteligentes

Um agente de software racional ou inteligente deve possuir uma ou mais das seguintes propriedades:

Autonomia

Essencial em sistemas especialistas, robótica e jogos.

O comportamento de um agente ideal deve combinar o seu domínio de conhecimento embutido com sua própria experiência adquirida. Comportamentos que são manipulados

apenas com o domínio do conhecimento perdem em autonomia. Por outro lado, à medida que o agente toma suas decisões baseado em sua própria experiência se utilizando de uma máquina de inferência, estará aumentando a sua autonomia.

Agentes que se utilizam apenas do seu domínio de conhecimento para realizar suas ações podem se deparar com dois problemas: dificuldade de criar o domínio de conhecimento – para um jogo de xadrez, por exemplo, seria necessária uma tabela de 35 elevado a 100 para mapear todas as possibilidades de percepções/ações - ou até mesmo impossibilidade de compor um domínio; e dificuldade em manter o domínio – em casos de domínio muito grande, alguma mudança que se faça necessária pode acarretar de deixar uma percepção sem ação ou até mesmo causar conflitos de ações para uma mesma percepção.

Adaptabilidade

Essencial em agentes na internet e interfaces amigáveis.

Capacidade do agente em se adaptar diante de uma nova situação, para a qual não foi fornecida nenhuma orientação no seu domínio de conhecimento.

Uma das abordagens que pode ser utilizada para prover adaptabilidade é a aprendizagem baseada em instâncias. Diversos algoritmos são utilizados na tarefa de classificação e treinamento dos dados: Os algoritmos são geralmente utilizados quando a aquisição do conhecimento é de difícil obtenção e neste caso se utilizam de uma base de dados armazenados. Uma vez que as instâncias estejam organizadas e treinadas, os algoritmos são capazes de classificar uma nova situação de acordo com a proximidade com todas as outras instâncias.

Comunicação

Essencial em sistemas multiagente, jogos e comércio eletrônico.

Capacidade do agente em se comunicar com o usuário, ou com ambiente ou, no caso de sistemas multiagentes, entre si. A troca de informações pode ser realizada de forma direta – comunicação explícita, ou indireta – emissão de sinais através do ambiente.

Para que haja interação, é necessária a utilização de protocolos padrão de comunicação. O protocolo é quem define as regras e impõe o formalismo necessário para que as mensagens sejam encaminhadas e compreendidas pelos agentes.

Personalidade

Essencial em sistemas de entretenimento, realidade virtual e interfaces amigáveis.

Para dotar um agente de personalidade é preciso realizar uma modelagem das suas atitudes e simulações das suas emoções. Quando o agente utiliza recursos multimídia para apresentar ao usuário um personagem, é denominado agente de *software* animado.

Continuidade temporal e persistência

Essencial em sistemas de filtragem, monitoramento e controle.

Agentes são processos que precisam estar continuamente em execução, seja de forma ativa ou passiva, no seu ambiente.

Para tornar suas percepções e ações armazenadas para futuras análises, monitoramentos e decisões é preciso que o agente interaja com um banco de dados diretamente ou com um protocolo de comunicação com o banco de dados.

3.2.2 Descrevendo um agente inteligente

Um agente inteligente pode ser descrito em termos do seu PAGE: *Perceptions* – percepções; *Actions* - ações; *Goals* – objetivos e *Environment* – ambiente.

Percepções

Corresponde a tudo que o agente será capaz de observar no ambiente que terá relevância para as suas ações.

Um agente guarda de trânsito, por exemplo, deverá observar: se o motorista está dirigindo sem cinto de segurança; se existe alguma criança no banco da frente; se o veículo estiver sem placa; se o veículo ultrapassou um sinal vermelho; dentre outras.

Ações

Corresponde a tudo que o agente será capaz de realizar no ambiente em busca de atingir seu(s) objetivo(s).

No caso do agente guarda de trânsito, exemplos de ações seriam: apitar para alertar o motorista sobre alguma irregularidade; solicitar que o motorista pare o veículo ou multar o motorista por alguma infração.

Objetivos

Um agente pode ter um ou mais objetivos a alcançar enquanto atua em um ambiente ou com os usuários.

Um exemplo de objetivo do agente guarda de trânsito seria fazer com que as leis sejam respeitadas.

Ambiente

O ambiente é o habitat de onde o agente terá a capacidade de captar todas as percepções necessárias para sua atuação. Pode ser um ambiente físico, para o caso de agentes robôs ou, mais comum, um *software*, para os agentes de *software*.

A diversidade de possibilidades diferentes de ambientes é imensa, porém existem algumas propriedades que ajudam a entender ou a definir o ambiente de um agente:

- **Acessível** – quando os sensores do agente conseguem perceber o estado completo do ambiente ou **inacessível** – quando os sensores do agente não têm condições de perceber tudo o que acontece no ambiente. Um agente humano, por exemplo, cujo domínio seja uma casa, não tem condições de perceber visualmente os objetos contidos dentro de um quarto fechado que ele não tem a chave para abrir;
- **Determinista** – quando o próximo estado do ambiente pode ser completamente determinado pelo estado atual e as ações selecionadas pelo agente ou **não determinista** – quando a combinação do estado atual e as ações do agente não são suficientes para determinar o próximo estado do ambiente;
- **Episódico** – quando o episódio de um agente não depende das ações ocorridas em episódios prévios. A experiência de um agente é dividida em episódios e cada episódio consiste numa possibilidade de percepção e ação ou **não-episódico** – quando o agente não pode decidir suas ações apenas baseado no episódio atual, ou seja, é necessário raciocinar entre o episódio atual e episódios prévios;
- **Estático** – quando o ambiente não sofre alterações enquanto o agente estiver escolhendo a ação a realizar ou **dinâmico** – quando há outros processos operando no qual poderá haver mudanças no ambiente além do controle do agente; e
- **Discreto** – quando existe um número definido de percepções e ações a cada episódio ou **contínuo** – quando as percepções e ações mudam em uma escala contínua de valores.

3.2.3 Arquiteturas de um agente

Os agentes podem ser implementados utilizando diferentes arquiteturas a depender das suas condições de perceber e agir, do(s) seu(s) objetivo(s) e das características do ambiente onde ele irá atuar. A Figura 13 ilustra a arquitetura básica de um agente.

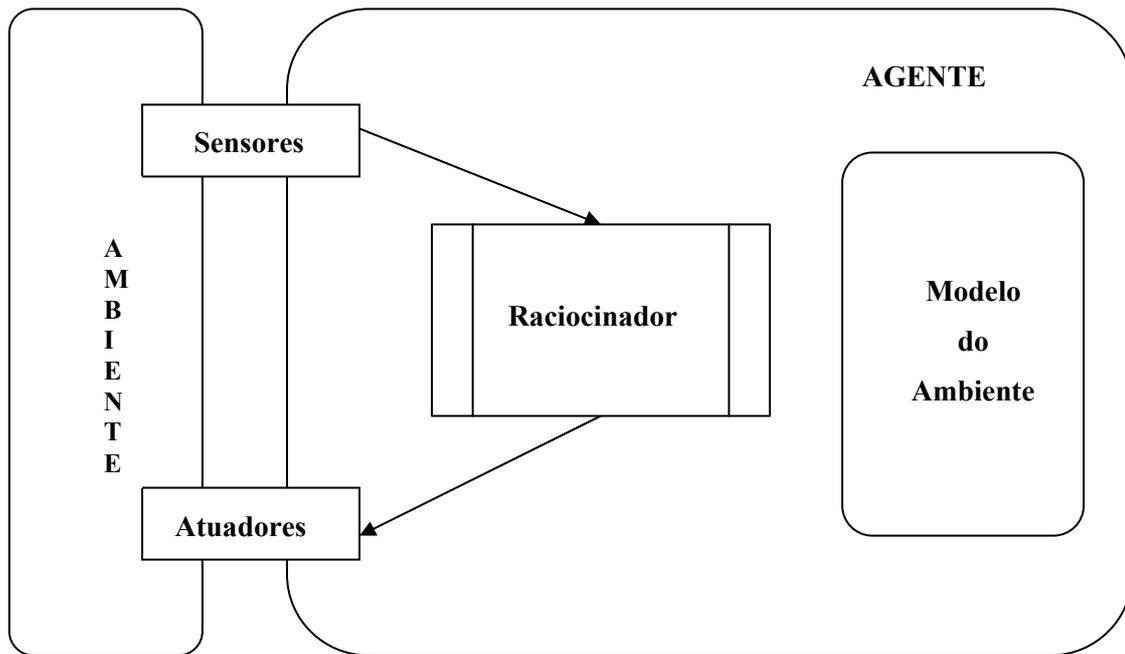


Figura 13 – Arquitetura Básica de um Agente. Fonte: [Russel e Norvig 2004].

As arquiteturas serão apresentadas por ordem de complexidade - da mais simples a mais complexa. À medida que a complexidade aumenta, o agente tem sua autonomia incrementada.

Inserido neste contexto de processo de agendamento, adotamos o conceito de agentes que, segundo [Russel e Norvig 2004], “é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.” Ainda nesta linha, [Wooldridge 2009], define agentes como “sistemas computacionais capazes de ações autônomas em algum ambiente, a fim de alcançarem seus objetivos de projeto.”. Neste trabalho é apresentado um protótipo de um sistema de

agendamento dentro do AVA Moodle e propomos um sistema multiagente que monitorará o processo de agendamento neste Ambiente. Esta arquitetura será apresentada no capítulo 4, que diz respeito ao protótipo desenvolvido.

CAPÍTULO 4

4. PROTÓTIPO

Este capítulo descreve o desenvolvimento de um protótipo para um mecanismo de suporte a agendamentos em AVA's.

4.1 Decisões do Projeto

Optou-se por utilizar o Moodle, por ser uma plataforma educacional voltada ao gerenciamento de cursos ou disciplinas (Moodle 2010) para a composição de AVAs. É uma solução tecnológica para o suporte ao agendamento de atividades dos usuários do AVA, além de ser um software livre, gratuito, que já possui uma grande comunidade de desenvolvedores em todo mundo.

Desta forma, ambientes que forem concebidos através do Moodle, poderão usufruir deste modelo de agendamento e seus usuários poderão utilizá-lo como mecanismo para realização dos compromissos e atividades, em seus cursos ou disciplinas.

A seguir são relacionadas algumas características do Moodle, que ratificam a sua escolha como arcabouço tecnológico para o suporte ao desenvolvimento do protótipo:

- a) É um software livre, de código aberto, mantido por um projeto que reúne desenvolvedores espalhados pelo mundo todo, o que traz facilidades, como a sua liberdade de utilização, alteração e disponibilização;
- b) Sua estrutura aceita que módulos sejam incorporados, o que permite o acoplamento do modelo com certa facilidade;
- c) Sua larga utilização por uma grande quantidade de instituições de ensino no Brasil e no exterior vem sendo bem sucedida, dando-lhe credibilidade para que seja continuamente utilizado como suporte do modelo de agendamento, e possa vir a contribuir ainda mais com eventuais aprimoramentos.

4.2 Descrições do Protótipo

O Sistema Web desenvolvido, visa facilitar a interação entre usuários, a partir do momento em que este irá possibilitar que qualquer usuário do ambiente virtual de aprendizagem, possa marcar uma reunião, com a finalidade de realizar um diálogo mais particular, permitindo a interação por parte dos mesmos.

Para que um usuário possa usufruir do sistema, será preciso que ele faça previamente um cadastro no AVA. Dentro do AVA existe um link que oferece acesso ao protótipo de agendamento. Por este motivo, o usuário deve cadastrar-se previamente no AVA a fim de posteriormente ter acesso ao sistema. As informações que deverão ser preenchidas são: nome de usuário, senha, endereço de e-mail, nome, sobrenome, cidade/município e país. Cada usuário possui um tipo de acesso, que será atribuído pelo administrador do sistema, que poderá ser: administrador, professor, estudante e visitante. O que modifica cada tipo de usuário é a permissão de acesso que cada um possui no AVA e em seus cursos.

A partir do momento em que o usuário cria seu cadastro e o sistema avisa que o cadastro foi concluído com sucesso, este está apto a utilizar o sistema. O próximo passo agora é fazer o login no AVA. Utilizando o nome de usuário e senha cadastrados previamente, o usuário consegue logar com facilidade no sistema e fazer uso de suas funcionalidades.

Em seguida, este usuário entra num curso do AVA, define sua disponibilidade e a partir de então, faz uso do sistema de agendamento. O usuário também poderá a visualizar a disponibilidade de outros usuários que estiverem no curso a fim de realizar agendamento de reunião com eles.

Somente usuários administradores têm a permissão para exclusão de outros usuários dos cursos do AVA.

4.3 Arquitetura do Protótipo

Apresentamos um protótipo de agendamento de reuniões dentro do AVA Moodle e propomos um Sistema Multiagente que monitorará o processo de agendamento neste ambiente. Essas atividades monitoradas por agentes de agendamento podem ser, por exemplo, reuniões para realização de chat, encontros presenciais, provas, trabalhos, etc. Antes de agendar um novo compromisso, referente a um determinado curso, o agente verifica em sua agenda, por meio de consultas ao banco de dados, a existência ou não, de incompatibilidades ou conflitos de horários e datas de cada usuário. O objetivo de utilizar uma camada multiagente é de deixar o sistema mais autônomo, auxiliando os usuários no processo de agendamento. Na Figura 14 podemos visualizar a arquitetura básica do sistema proposto.

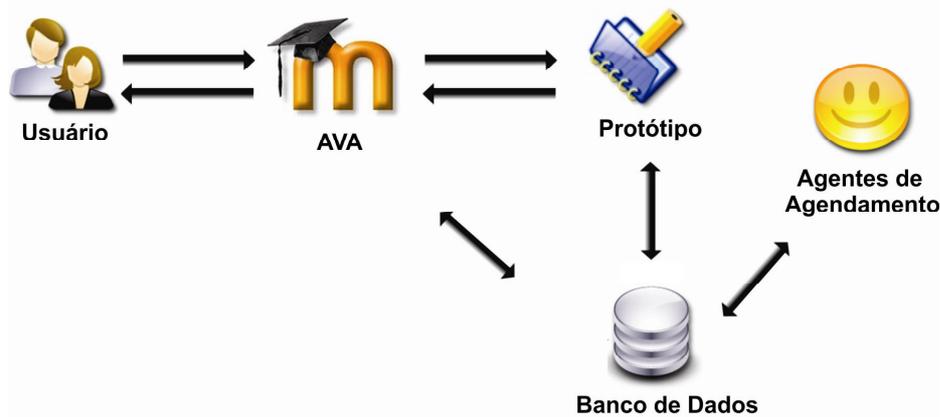


Figura 14 – Arquitetura Básica do Protótipo Agendamento.

Esta arquitetura apresenta uma visão macro do sistema de agendamento. Os usuários podem acessar o AVA e cadastrar suas informações para consultar posteriormente. No AVA está contido o protótipo de agendamento, que pode interagir com os usuários do AVA apresentando as informações disponíveis através de sua interface. O sistema de agendamento acessa o banco de dados a fim de trabalhar com as informações disponibilizadas pelos usuários do AVA e do sistema de agendamento. Em seguida, os agentes interagem

diretamente com o banco, que é o mesmo utilizado pelo protótipo e pelo Moodle, com o objetivo de realizar o monitoramento das opções pré-definidas pelos usuários, antes de confirmar o agendamento das atividades. A Figura 15 apresenta o código do agente em JAVA antes de ser compilado.

```

import jade.core.Agent;

public class DatabaseAgent extends Agent {
    private Connection conn;
    private MessageParser mp;
    private ACLMessage lastReceivedMsg;

    private MessageTemplate template = MessageTemplate.and(
        MessageTemplate.MatchPerformative(ACLMessage.QUERY_IF),
        MessageTemplate.MatchOntology("presence") );

    //Override
    protected void setup() {
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn = DriverManager.getConnection("jdbc:mysql://localhost");
            mp = new MessageParser();
        }
        catch (Exception e) {
            e.printStackTrace();
        }

        addBehaviour(new TickerBehaviour(this, 5000) {
            public void onTick() {
                createPerson();
                ACLMessage msg = myAgent.receive(template);
                if (msg != null) {
                    Logger.getLogger(DatabaseAgent.class).info(msg);
                }
            }
        });
    }
}

```

Figura 15 – DbAgent2 - Código dos Agentes no JAVA.

Ao ser compilado, o código deixará os agentes ativos para as opções de consulta e agendamento no AVA. Ao ser executado, o programa ativa os agentes no JADE, e os deixa em *stand by* para que estes agentes monitorem o agendamento quando solicitado, conforme podemos observar na Figura 16. Este código pode ser melhor visualizado ao final deste trabalho no apêndice B.

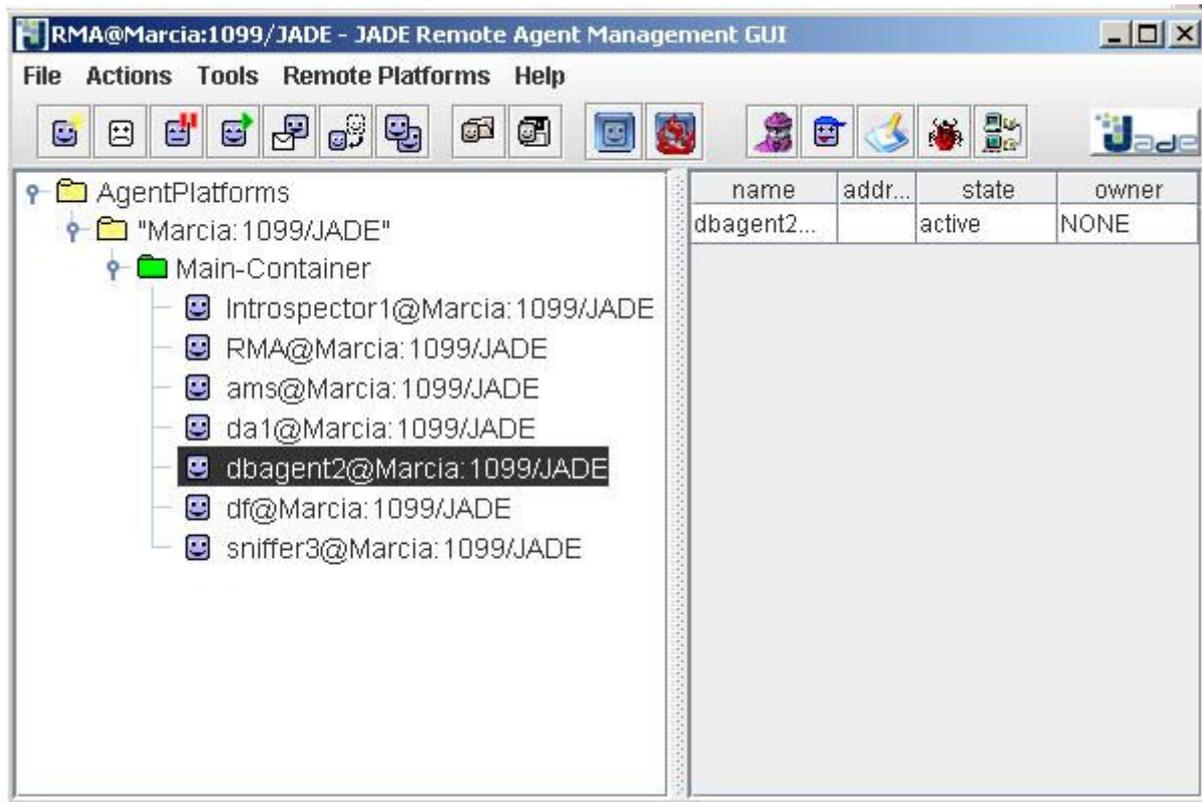


Figura 16 – Agente de Agendamento ativo no JADE

4.4 Modelagem do Protótipo

Esta subseção descreve a modelagem realizada para o protótipo através da utilização de Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Sequencia.

4.4.1 Diagrama de Casos de Uso

Um caso de uso descreve uma funcionalidade proposta para um novo sistema. Incorporado à *Unified Modeling Language* (UML) sua utilização é uma prática freqüente na identificação de requisitos de um sistema (Booch *et al.*, 2000 *apud* Lima 2006).

A Figura 17 apresenta o Diagrama de Casos de Uso para o protótipo do agendamento.



Figura 17 – Diagrama de Casos de Uso para o Agendamento.

O agendamento é acessado pelo usuário que interage com os casos de uso. Uma primeira versão deste diagrama pode ser encontrada em [Pimenta e Netto 2009]. Os casos de uso são apresentados na Figura 17 e descritos de forma resumida a seguir:

Definir Disponibilidade: Possui a responsabilidade de identificar a disponibilidade dos participantes do ambiente, a fim de obter as informações necessárias para que o agente as utilize quando necessário.

Consultar Agendamento: Possui a responsabilidade de manter o participante informado sobre as atividades agendadas pelo mediador para o grupo e sobre seus próprios agendamentos.

Realizar Agendamento: Possui a responsabilidade de efetivamente criar um novo agendamento, após verificar a disponibilidade de horário cadastrada.

Informar Agendamento: Possui a responsabilidade de manter os participantes e os mediadores informados sobre as atividades que estarão sendo agendadas no ambiente. Ele pode avisar a um determinado participante sobre as tarefas e alertar o mediador sobre o cumprimento das mesmas, pelos participantes do grupo.

4.4.2 Diagrama de Classes

A Figura 18 apresenta o Diagrama de Classes para o protótipo de Agendamento construído no Moodle 1.9+.

A classe *Agendamento* herda o Id do usuário e da Agenda, após este preenchimento o aluno irá cadastrar a sua disponibilidade utilizando os mesmos id's na classe disponibilidade. Esta, por sua vez, utiliza alguns métodos de outras classes com as quais possui relações de dependência para, juntamente com seus próprios métodos, fornecer as funcionalidades de monitoramento à consulta dos agentes, o que irá complementar o percurso do mecanismo de agendamento. Estas classes podem ser melhor visualizadas no Apêndice C deste trabalho.

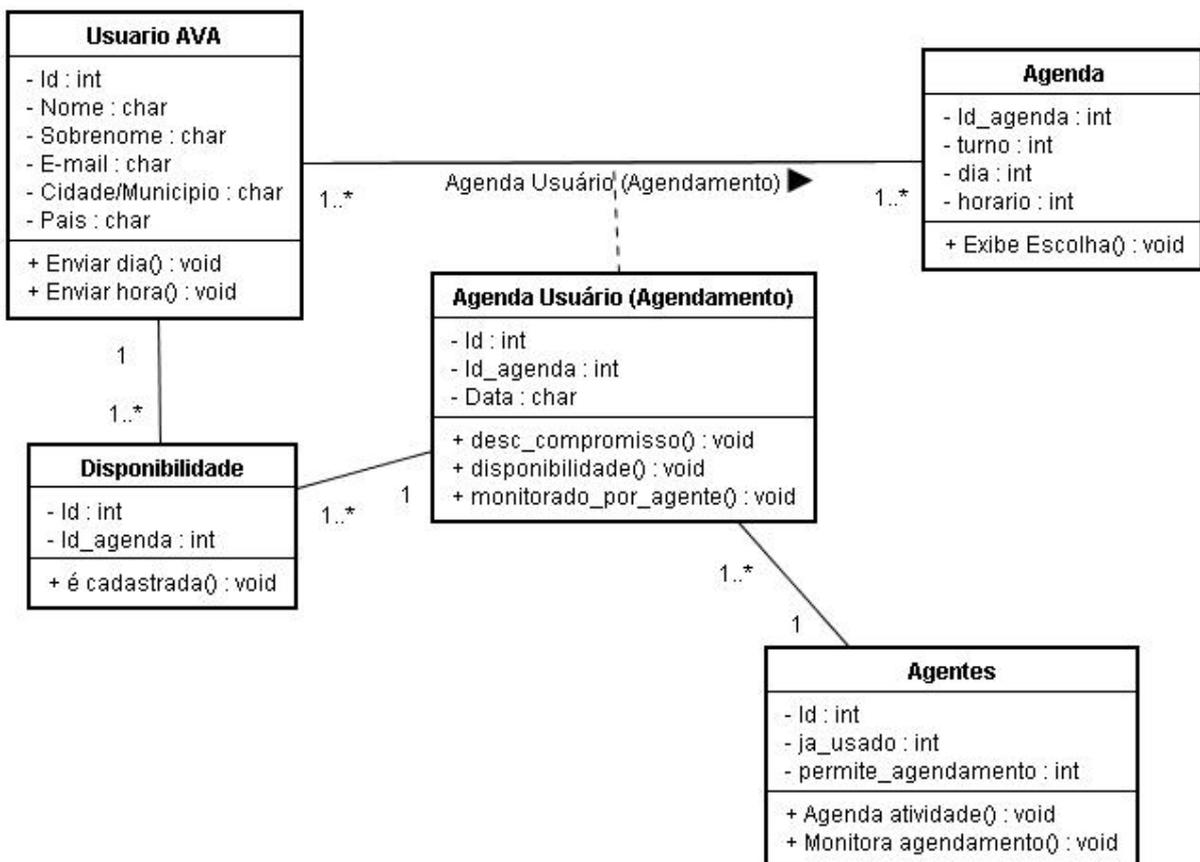


Figura 18 – Diagrama de Classes para o Agendamento.

4.4.3 Diagrama de Sequência

A Figura 19 apresenta o diagrama de seqüência para o cadastro de disponibilidade. Segundo [Furlan 1998], diagrama de seqüência mostra interações entre objetos organizados em uma seqüência de tempo e de mensagens trocadas, mas não trata de associações entre os objetos. A definição das mensagens trocadas é feita baseando-se na documentação dos casos de uso.

Para elaborar o diagrama de seqüência, pode-se pensar na implementação. Algumas classes, não criadas no diagrama de classes preliminar, serão criadas agora para modelar a troca de mensagens no sistema, tanto do usuário com o sistema como mensagens trocadas entre as classes. Como também estão sendo modeladas as mensagens trocadas com o usuário, no mínimo deverão ser criadas as classes de interface.

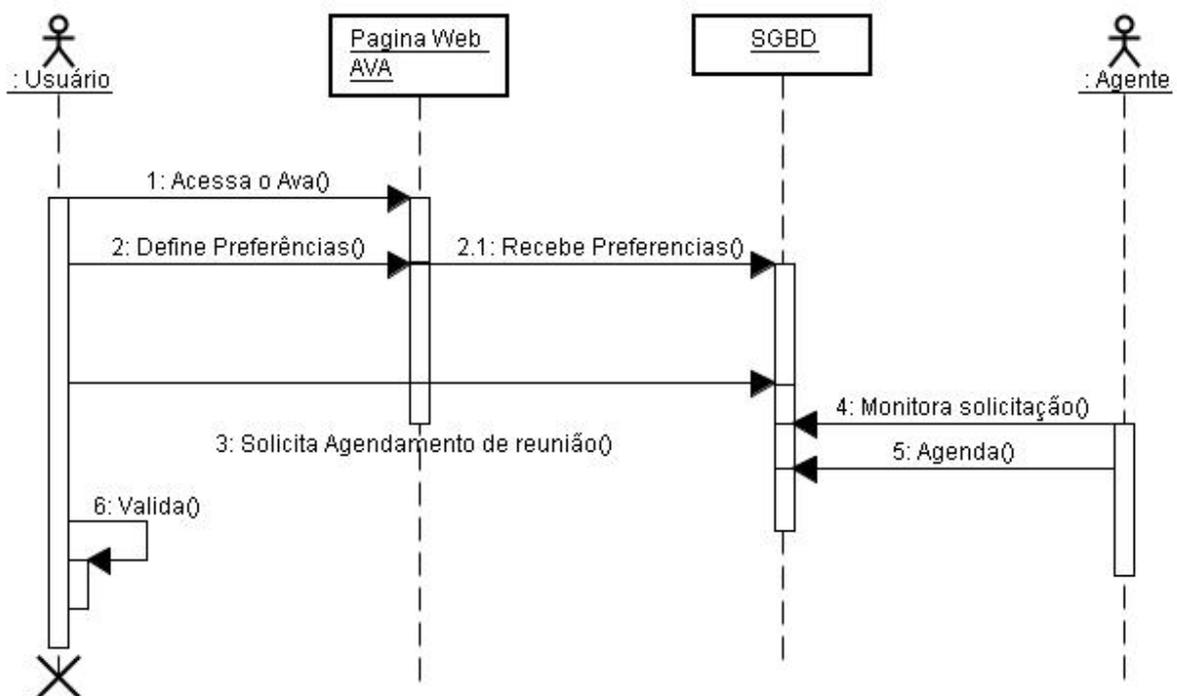


Figura 19 - Diagrama de Sequência para definição de preferências.

A realização do agendamento é prevista efetivamente após a validação da disponibilidade de horário pelo agente. Esta é pré-definida pelos usuários do AVA, onde posteriormente, o agente é responsável por verificar a disponibilidade do horário pré-cadastrado e, realizar o agendamento do usuário do AVA.

CAPÍTULO 5

5. EXPERIMENTOS E VALIDAÇÃO

A abordagem proposta para o monitoramento em AVAs, utilizada pelo protótipo descrito no capítulo anterior pôde ser aplicada num cenário real de utilização apresentado a seguir.

5.1 Cenário de Utilização

Com o protótipo implementado, foi possível realizar testes de agendamento. Este modelo foi testado com uma turma de 46 usuários que utilizaram o AVA Moodle, onde existem alunos, tutores, mantenedores de cursos e professores do curso técnico em Meio Ambiente na modalidade a distância. Podemos observar algumas vantagens de se utilizar o modelo como apoio a realização dos compromissos no AVA, entre elas, o desenvolvimento de um compromisso baseado em um cadastro dos horários disponíveis, que permitiu aos usuários da turma do AVA, a capacidade de planejar, a trabalhar em equipe ou dupla, e integração destes usuários uma vez que eles podem consultar dia e horário que os colegas têm disponibilidade de realizar uma atividade.

Com a finalidade de averiguar a funcionalidade da ferramenta, foi realizado um backup da turma supracitada, que são alunos reais do AVA Moodle, cadastrados no projeto Escola Técnica Aberta do Brasil – e-Tec Brasil, um projeto de expansão da educação profissionalizante do Ministério da Educação e Cultura, Secretaria de Educação a Distância e Secretaria de Educação Profissional e Tecnológica. [E-Tec Brasil 2010]. Este grupo de alunos, vinculados a este programa, estão relacionados a uma turma do curso técnico de meio ambiente, na modalidade a distância, da disciplina de Biologia Ambiental e foram escolhidos para testar o experimento e verificar a eficiência do serviço prestado pelo modelo de agendamento, sugerido neste trabalho.

5.2 Características Gerais do Protótipo

O protótipo concebido para um Ambiente Virtual de Aprendizagem baseia-se na estrutura de módulos do *Modular Object Oriented Dynamic Learning Environment* (Moodle).

Para a instalação do AVA Moodle, na versão 1.9+, no sistema operacional Windows XP, foi utilizado o pacote com WampServer [WampServer 2010] (conhecido anteriormente como WAMP5), que é um software publicado sob a General Public License (GNU4). Este pacote é usado para instalar rapidamente no computador os softwares PHP, banco de dados MySQL e servidor Apache, disponibilizando suporte ao uso de scripts PHP localmente no Windows. O WampServer dispõe ainda do phpMyAdmin, que é uma aplicação de fácil utilização que serve para administrar o banco de dados MySQL pelo *browser*. Podemos utilizá-lo para criar, copiar, deletar, renomear e alterar tabelas, fazer a manutenção de tabelas, deletar, editar e adicionar campos, exportar ou importar um banco de dados, executar códigos, dentre outras coisas, conforme podemos observar na Figura 20.

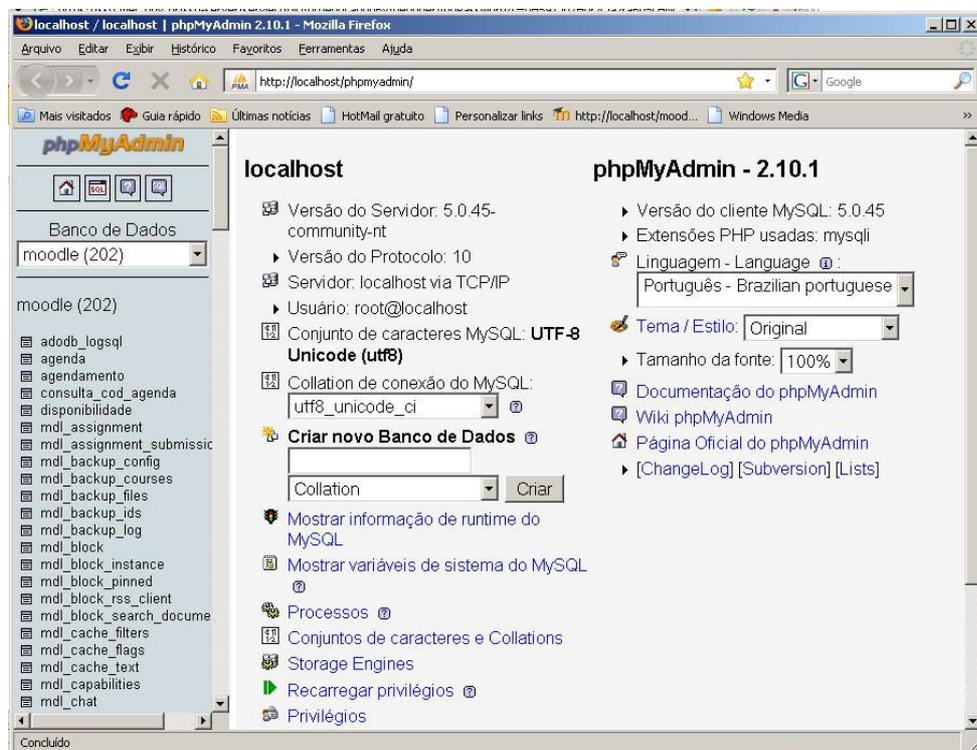


Figura 20 – Aplicação phpMyAdmin ativa.

⁴ Disponível em <<http://www.gnu.org>>

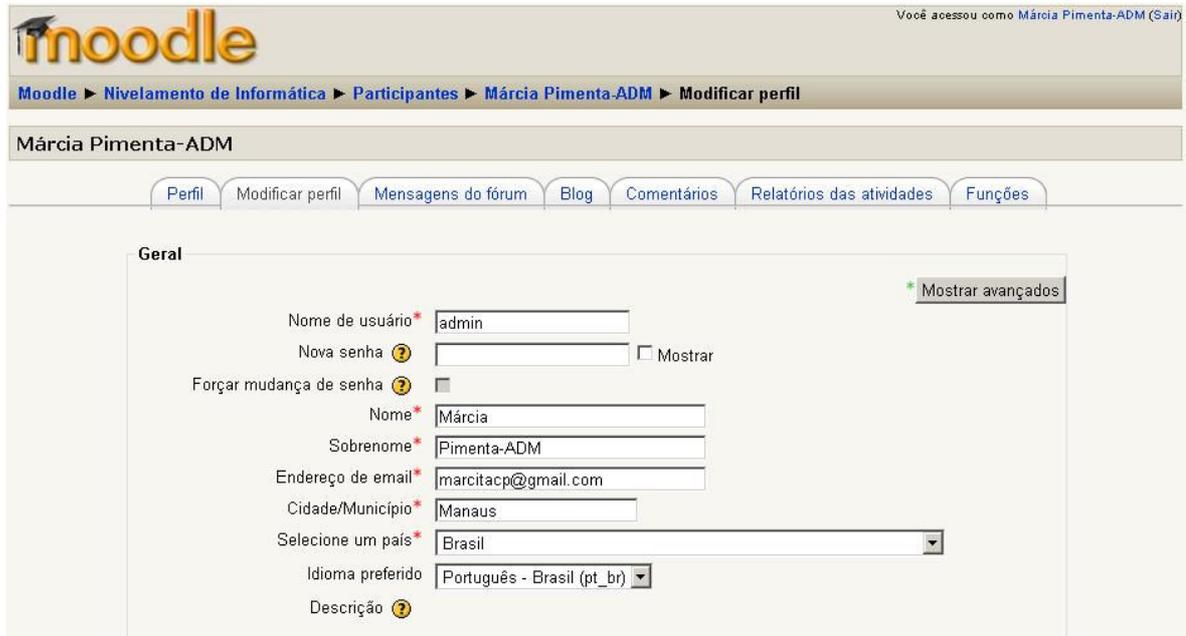
A implementação dos novos recursos é desenvolvida na linguagem de programação PHP. Para monitorar e realizar o agendamento efetivamente, agentes inteligentes que o intermediam, foram implementados na linguagem de programação JAVA, utilizando o *framework* JADE (*Java Agent Development Framework*) em sua versão 3.7. Os softwares supracitados podem ser adotados como requisitos mínimos para a utilização do protótipo proposto neste trabalho.

O agendamento terá início a partir do momento em que o usuário faz login no AVA. Dentro do ambiente virtual de aprendizagem foi implementado o sistema de verificação da agenda, baseado nas informações pré-definidas pelo próprio usuário. Este sistema possui a responsabilidade de manter o participante informado sobre as atividades agendadas pelo mediador ou reuniões entre os usuários do AVA. Essas atividades serão sempre síncronas, que podem ser, por exemplo, reuniões, chats, encontros presenciais, provas, trabalhos, etc, como já foi dito anteriormente e verificado nesta heurística. São necessários alguns passos no AVA para que o usuário realize o agendamento, o que iremos verificar no exemplo de agendamento apresentado na próxima seção.

5.3 Exemplo do Agendamento

Para que um usuário possa utilizar o sistema, será preciso que ele faça previamente um cadastro no AVA Moodle. Dentro do AVA existe um link que oferece acesso ao protótipo de agendamento. Por este motivo, o usuário deve cadastrar-se previamente no AVA a fim de posteriormente acessar o sistema. As informações que deverão ser preenchidas são: nome de usuário, senha, endereço de e-mail, nome, sobrenome, cidade/município e país, conforme podemos verificar na Figura 21. Cada usuário possui um tipo de acesso dentro do AVA, que será atribuído pelo administrador do sistema, que poderá ser: administrador, professor,

estudante e visitante. O que modifica cada tipo de usuário é a permissão de acesso que cada um possui no AVA e em seus cursos.



The screenshot shows the Moodle user profile modification page for Márcia Pimenta-ADM. The page title is "Módulo > Nivelamento de Informática > Participantes > Márcia Pimenta-ADM > Modificar perfil". The user's name "Márcia Pimenta-ADM" is displayed at the top. Below the name are several tabs: "Perfil", "Modificar perfil", "Mensagens do fórum", "Blog", "Comentários", "Relatórios das atividades", and "Funções". The "Modificar perfil" tab is active. The "Geral" section contains the following fields:

- Nome de usuário*: admin
- Nova senha: [input field] Mostrar
- Forçar mudança de senha:
- Nome*: Márcia
- Sobrenome*: Pimenta-ADM
- Endereço de email*: marcitacp@gmail.com
- Cidade/Município*: Manaus
- Selecione um país*: Brasil
- Idioma preferido: Português - Brasil (pt_br)
- Descrição: [input field]

A "Mostrar avançados" link is visible in the top right corner of the form area.

Figura 21 - Página do Moodle para realizar ou atualizar o cadastro do usuário.

A partir do momento em que o usuário cria seu cadastro e o sistema avisa que o cadastro foi concluído com sucesso, este receberá um e-mail para confirmação do cadastro e estará apto a utilizar o ambiente virtual de aprendizagem. O próximo passo é fazer o login no AVA, como visualizamos na Figura 22. Utilizando o nome de usuário e senha cadastrados previamente, o usuário consegue logar com facilidade no sistema e fazer uso de suas funcionalidades.

Você ainda não fez o acesso (Acesso)

Moodle ► Acesso ao site Português - Brasil (pt_br)

Retornando a este site?

Acesse aqui, usando seu Nome de Usuário e a sua Senha.
(O uso de Cookies deve ser permitido no seu navegador) ?

Nome de usuário

Senha

Alguns cursos podem permitir o acesso a visitantes

Esqueceu o seu nome de usuário ou a sua senha?

Você ainda não fez o acesso (Acesso)

[Home Page](#)

Figura 22 - Página do Moodle para realizar o acesso com login e senha.

Ao acessar o ambiente, o usuário irá encontrar na tela principal do seu curso, um link chamado Protótipo Agendamento onde ele poderá cadastrar a sua disponibilidade de horários para alguma possível atividade síncrona, bem como realizar consultas e os agendamentos. Dentro do ambiente virtual de aprendizagem foi implementado o sistema de verificação da agenda, baseado nas informações pré-definidas pelo próprio usuário. Este sistema possui a responsabilidade de manter o participante informado sobre as reuniões agendadas pelo mediador ou entre os usuários do AVA. É apresentada na Figura 23, a página principal de um curso no AVA Moodle, o qual utiliza o modelo de agendamento implementado.

The screenshot shows the Moodle interface for a course titled "Nivelamento de Informática". At the top, the Moodle logo is on the left, and the user's name "Márcia Pimenta-ADM" is on the right. Below the header, there's a navigation bar with "Moodle" and the course name. A sidebar on the left contains various menu items like "Participantes", "Atividades", "Pesquisar nos Fóruns", and "Administração". The main content area displays a message from "Márcia Pimenta" welcoming students and asking them to fill out a prototype scheduling form. Below the message, there are links for "Fórum de notícias" and "Protótipo AGENDAMENTO", with the latter being circled in red. On the right, there are sections for "Últimas Notícias", "Próximos Eventos", and "Atividade recente".

Figura 23 - Página principal do curso no Moodle com o link para o protótipo Agendamento.

Na Figura 24, podemos visualizar o cadastro de disponibilidades proposto no modelo de agendamento, que é o primeiro passo depois que o usuário clica no link supracitado. Após a escolha dos dias, horários e turnos pretendidos, o sistema irá informar o sucesso do cadastro e exibirá um link para fechar a tela. Cada aluno do AVA irá realizar essa tarefa apenas uma vez. Se o aluno que já cadastrou suas disponibilidades, clicar novamente no link de cadastro, o sistema irá avisar que este cadastro já foi realizado, conforme podemos observar na Figura 25.

SISTEMA MULTIAGENTE DE AGENDAMENTO

CADASTRO CONSULTA AGENDAMENTO AGENDAMENTO INDIVIDUAL

▶ CADASTRO DE DISPONIBILIDADE

Nome: Márcia Pimenta-ADM

TURNO MATUTINO	TURNO VESPERTINO	TURNO NOTURNO
<input type="checkbox"/> Segunda 8 as 10 horas	<input type="checkbox"/> Segunda 14 as 16 horas	<input type="checkbox"/> Segunda 18 as 20 horas
<input type="checkbox"/> Segunda 10 as 12 horas	<input type="checkbox"/> Segunda 16 as 18 horas	<input type="checkbox"/> Segunda 20 as 22 horas
<input type="checkbox"/> Terça 8 as 10 horas	<input type="checkbox"/> Terça 14 as 16 horas	<input type="checkbox"/> Terça 18 as 20 horas
<input type="checkbox"/> Terça 10 as 12 horas	<input type="checkbox"/> Terça 16 as 18 horas	<input type="checkbox"/> Terça 20 as 22 horas
<input type="checkbox"/> Quarta 8 as 10 horas	<input type="checkbox"/> Quarta 14 as 16 horas	<input type="checkbox"/> Quarta 18 as 20 horas
<input type="checkbox"/> Quarta 10 as 12 horas	<input type="checkbox"/> Quarta 16 as 18 horas	<input type="checkbox"/> Quarta 20 as 22 horas
<input type="checkbox"/> Quinta 8 as 10 horas	<input type="checkbox"/> Quinta 14 as 16 horas	<input type="checkbox"/> Quinta 18 as 20 horas
<input type="checkbox"/> Quinta 10 as 12 horas	<input type="checkbox"/> Quinta 16 as 18 horas	<input type="checkbox"/> Quinta 20 as 22 horas
<input type="checkbox"/> Sexta 8 as 10 horas	<input type="checkbox"/> Sexta 14 as 16 horas	<input type="checkbox"/> Sexta 18 as 20 horas
<input type="checkbox"/> Sexta 10 as 12 horas	<input type="checkbox"/> Sexta 16 as 18 horas	<input type="checkbox"/> Sexta 20 as 22 horas

GRAVAR

Figura 24 - Página do protótipo para o cadastro de disponibilidade de horário dos usuários.

SISTEMA MULTIAGENTE DE AGENDAMENTO

CADASTRO CONSULTA AGENDAMENTO AGENDAMENTO INDIVIDUAL

Prezado usuário, você já cadastrou 6 DISPONIBILIDADE(S)

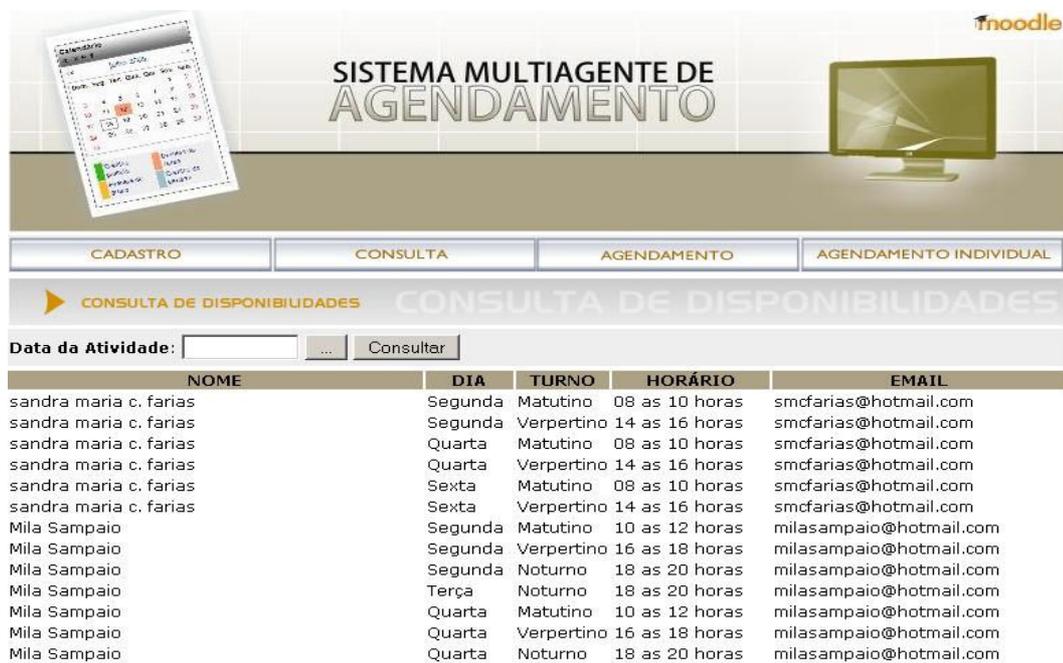
[Fechar](#)

SMAA - SISTEMA MULTIAGENTE DE AGENDAMENTO
COPYRIGHT 2010

Figura 25 - Página que apresenta que o cadastro de disponibilidade já foi realizado.

A partir do cadastro de disponibilidades realizado, o usuário aparecerá nos horários selecionados disponível para agendar. O modelo desenvolvido baseia-se na estrutura de módulo apresentado através de *link* para site do AVA Moodle. Além de

cadastrar a disponibilidade dos usuários, também realiza consultas a tabelas do banco de dados para prover as informações solicitadas por outros usuários do AVA, para quando estes precisarem de informações sobre a disponibilidade dos colegas ou mediadores para agendar uma reunião, por exemplo. Na Figura 26 são apresentados todos os usuários que realizaram o cadastro de disponibilidades e estão disponíveis para agendar, onde é possível também, realizar a consulta de datas específicas para realizar o agendamento posterior quando solicitado.



NOME	DIA	TURNO	HORÁRIO	EMAIL
sandra maria c. farias	Segunda	Matutino	08 as 10 horas	smcfarias@hotmail.com
sandra maria c. farias	Segunda	Verpertino	14 as 16 horas	smcfarias@hotmail.com
sandra maria c. farias	Quarta	Matutino	08 as 10 horas	smcfarias@hotmail.com
sandra maria c. farias	Quarta	Verpertino	14 as 16 horas	smcfarias@hotmail.com
sandra maria c. farias	Sexta	Matutino	08 as 10 horas	smcfarias@hotmail.com
sandra maria c. farias	Sexta	Verpertino	14 as 16 horas	smcfarias@hotmail.com
Mila Sampaio	Segunda	Matutino	10 as 12 horas	milasampaio@hotmail.com
Mila Sampaio	Segunda	Verpertino	16 as 18 horas	milasampaio@hotmail.com
Mila Sampaio	Segunda	Noturno	18 as 20 horas	milasampaio@hotmail.com
Mila Sampaio	Terça	Noturno	18 as 20 horas	milasampaio@hotmail.com
Mila Sampaio	Quarta	Matutino	10 as 12 horas	milasampaio@hotmail.com
Mila Sampaio	Quarta	Verpertino	16 as 18 horas	milasampaio@hotmail.com
Mila Sampaio	Quarta	Noturno	18 as 20 horas	milasampaio@hotmail.com

Figura 26 – Consulta de disponibilidade de horário dos usuários do AVA.

Na Figura 27 é apresentada a opção de agendamento individual dos usuários disponíveis em um curso do AVA. Neste momento o agente implementado no JADE, chamado DbAgent está ativo e irá monitorar se é possível agendar ou não reunião. Ao desempenhar este procedimento, o agente grava a consulta em uma tabela temporária. Nesta tabela, é realizada a consulta pelos agentes; em caso positivo, o agente grava os dados necessários na tabela do agendamento do banco de dados. Para efetuar o Agendamento individual, o usuário que fez o login no sistema aparece primeiramente na opção “Usuário que

está agendando”; as possíveis pessoas que ele poderá agendar aparecem na opção turma abaixo. Lembrando que só irão aparecer nesta guia, os usuários que já cadastraram sua disponibilidade, para não haver inconsistência nas informações.

Após escolher o usuário, o agendador ainda deverá selecionar a opção da data e horário a serem agendados. Se o usuário tiver a disponibilidade escolhida cadastrada, o agente verifica esta informação e o agendamento é realizado e agente apresenta a mensagem: “Agendamento realizado com sucesso!”, caso contrário, o sistema apresenta uma mensagem, avisando que não é possível realizar este agendamento. Isto será devido ao fato de não haver esta opção de disponibilidade escolhida cadastrada ou porque o usuário ainda não cadastrou a sua disponibilidade ao iniciar o AVA. Outra possibilidade é a do mediador tentar forçar um agendamento sem o monitoramento do agente, para que haja que renegociação pessoalmente ou por e-mail para uma data específica, em que não haja disponibilidade por parte do aluno. Esta idéia da renegociação estará melhor detalhada no capítulo 6 deste trabalho, na seção trabalhos futuros.



Figura 27 - Tela de opção de Agendamento individual.

Outra opção no protótipo é o da Figura 28, que apresenta os usuários que possuem disponibilidades para agendamentos específicos de dia e horário para verificação desta disponibilidade. Com data e horário equivalentes; é possível visualizar o usuário que já possui agendamento.

NOME	HORÁRIO	EMAIL
sandra maria c. farias	10 as 12 horas	smcfarias@hotmail.com
Maria Isabel	08 as 10 horas	miar@terra.com.br

SMAA - SISTEMA MULTIAGENTE DE AGENDAMENTO
COPYRIGHT 2010

Figura 28 - Tela de Agendamentos realizados.

Por ter sido implementado na forma de um novo módulo para o Moodle, o protótipo poderá ser adicionado a qualquer instalação do Moodle 1.9+ através de procedimentos simples e usuais de inserção de novos módulos na plataforma. Sua compatibilidade com outras versões não foi testada, no entanto estima-se que ela possa ser usada com versões próximas da 1.9+, obedecendo os requisitos mínimos de software já citados.

Para melhor utilização deste protótipo em um ambiente em produção Moodle, os arquivos de instalação poderão ser obtidos através do endereço eletrônico marcitacp@gmail.com. Outras informações sobre desenvolvimento de código e características técnicas específicas do Moodle podem ser disponibilizadas no mesmo endereço eletrônico supracitado.

CAPÍTULO 6

6. CONSIDERAÇÕES FINAIS DO TRABALHO

O modelo proposto permite a especificação, a execução e a evolução de estratégias de acompanhamento diante do dinamismo das atividades de mediação (realizadas por mediadores) e das ações executadas com a ajuda de estudantes (observação, construção, diálogo). A execução dessas estratégias deve apoiar as atividades de mediação reduzindo significativamente a sobrecarga de trabalho dos mediadores com tarefas repetitivas, além de incorporar características pró-ativas ao ambiente de aprendizagem, tornando-o mais interessante e atrativo ao estudante. Entre essas características pró-ativas, podemos citar avisos personalizados de acordo com agendamentos realizados pelos usuários do AVA. Com a integração do sistema multiagente que mediará agendamento no Moodle, o sistema ficará mais robusto e terá uma maior autonomia nesse processo, facilitando o trabalho do agendamento por parte dos usuários, por exemplo, estudantes e professores.

Assim, discutem-se o problema crescente das estratégias de acompanhamento do processo de aprendizagem na EAD, utilizando um AVA, e a necessidade de automatizar parte dessa gerência. Foi apresentado um modelo para um sistema de agendamento de reuniões baseado em um contexto multiagente, a fim de facilitar sua utilização e apoiar à mediação dos compromissos entre os usuários.

Resultados preliminares desta experiência encorajam-nos a aperfeiçoar cada vez mais a utilização de recursos computacionais no contexto tecnológico, como por exemplo, o Moodle, que pode ser utilizado em atividades do ensino presencial e certamente contribuirá muito para futuros cursos a distancia. Nesta experiência tivemos resultados empíricos pelo fatos dos experimentos terem dado continuidade mesmo após a finalização da disciplina.

Portanto, ficamos motivados a explorar cada vez mais a potencialidades desse sistema utilizados em novas disciplinas nestes cursos à distância.

6.1 Contribuições

Este projeto contribui para:

- a) Concepção de novas ferramentas e aplicações que sirvam de suporte às agendas de comunidades virtuais de aprendizagem;
- b) Adequação dos AVAs existentes à necessidade de um agendamento de reuniões para a realização das atividades no espaço de trabalho mais amplo e eficaz;
- c) Ajuda no desenvolvimento de novos mecanismos e módulos para o Moodle;

6.2 Trabalhos Futuros

A abordagem de agendamento apresentada neste trabalho pode ser incorporada em mecanismos de acompanhamento das atividades desenvolvidas em ambientes virtuais de aprendizagem utilizados na EAD. O objetivo maior é que este modelo de agendamento, seja utilizado com relativa facilidade e frequência por seus usuários, minimizando a falta de conhecimento das disponibilidades de horários dos usuários, ocasionadas pelo fato dos usuários não cadastrarem esta disponibilidade previamente e por isso, não agendarem em dias e horários que os colegas possam também. Com isto, poderemos até pensar em diminuir um problema eminente nos cursos na modalidade a distância chamado evasão, já que os compromissos agendados serão de acordo com a disponibilidade de cada usuário, pois algumas vezes os compromissos do dia-a-dia, no impedem de acessar o AVA diariamente no mesmo horário.

Sugerimos como trabalhos futuros indicações de agendamento entre usuários que possuam perfis semelhantes quando estiverem desenvolvendo trabalhos em comum, sendo estes perfis definidos como áreas de interesse dos usuários, seus objetivos e características individuais. Tal indicação será através da utilização de agentes de software específicos que seriam representantes de cada usuário para este fim, e que poderão informar o agendamento realizado através de e-mail ou SMS (Short Message Service).

Outra sugestão de trabalho futuro seria a renegociação do agendamento. Esta renegociação poderia ser realizada ora pelos próprios agentes, que representariam os usuários do AVA; ora sem os agentes, onde os próprios usuários fariam essa negociação em caráter de urgência em agendar uma reunião. Esta renegociação poderia ser feita dentro do próprio AVA ou por e-mail.

Recomendamos também, como trabalho futuro, tornar os mecanismos instalação do protótipo de agendamento, mais adaptáveis aos ambientes virtuais de modo geral, não somente ao Moodle como utilizado neste trabalho, acomodando assim, as necessidades e diferenças individuais de quem os utiliza.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ARISTOTELES – Disponível em www.mundodosfilosofos.com.br/aristoteles.htm, acesso em 27/01/2010.
- BERGER, Florian, Rolf Klein, Nussbaum Doron. “A Meeting Scheduling Problem Respecting Time and Space”. © Springer Science + Business Media, Bonn, Germany, 2008.
- BOOCH, G; RUMBAUGH, J e JACOBSON, I: UML, Guia do Usuário: tradução; Fábio Freitas da Silva, Rio de Janeiro, Campus, 2000.
- BRADSHAW, J. M. “An introduction to software agents In”: BRADSHAW, J. M. Ed. Software Agents. Massachusetts: MIT Press, 1997.
- BRITO, Silvana R.; SILVA, Aleksandra, FAVERO, Elói; “Modelo Baseado em Agentes para Acompanhamento do Processo de Aprendizagem em Ambientes Virtuais”. Anais do Simpósio Mineiro de Sistemas de Informação; Minas Gerais - MG: Sociedade Brasileira de Computação, Outubro de 2005.
- CAMPANA, Vitor Faiçal; Sanches, Diego Rodrigues; Tavares, Orivaldo de Lira; Souza, Santinho Ferreira de. “Agentes para Apoiar o Acompanhamento das Atividades em Ambientes Virtuais de Aprendizagem”, Anais Do XIX SBIE – Simpósio Brasileiro de Informática na Educação. Fortaleza - CE, 2008.
- CRAWFORD, Elisabeth and VELOSO, Manuela. "Learning to Select Negotiation Strategies in Multiagent Meeting Scheduling." In Proceedings of EPIA'05, the 12th Portuguese Conference on Artificial Intelligence, Covilhã, Portugal, December 2005.
- CUESTA, Pedro, Gómez, Alma, *et al.* "Developing a meeting scheduler multiagent system: an experimental case study". Iadis International Conference Applied Computing. Lisboa, Portugal, 2004.
- DIMITRACOPOULOU, Angelique. “Designing Collaborative Learning Systems: Current Trends & Future Research Agenda”, Learning Technology & Educational Engineering Laboratory. University of the Aegean, Greece, 2005.

- EIJK, J. R. van. "Implementation of a Scheduler Agent System for Traveling Users". Poster em workshop "Ubiquitous Agents on Embedded, Wearable and Mobile devices", Julho: 2002.
- E-TEC BRASIL, Disponível em: <http://www.etcbrasil.mec.gov.br> Acesso em 23 fev. 2010.
- FURLAN, José Davi. "Modelagem de objetos através da UML". Makron Books, 1998.
- HOUAISS, Antônio; VILLAR, Mário de Salles. "Minidicionário Houaiss da Língua Portuguesa". 3 ed.rev. e aum. Rio de Janeiro: Objetiva, 2009.
- JADE, Disponível em: jade.tilab.com/ Acesso em 15 fev. 2010.
- LIMA, P. S. R. "Um Ambiente Colaborativo de Aprendizagem Interdisciplinar Apoiado por Interfaces Adaptativas". Tese de Doutorado em Engenharia Elétrica. Universidade Federal do Pará. Belém. 2006.
- MARTINS, Jean Paulo. "O Problema do Agendamento Semanal de Aulas". 2010. 84 f. Dissertação (Mestrado em Computação). Universidade Federal de Goiás, Goiânia - GO.
- MESQUITA, Luciana Frasson (2003). "Mecanismos de Suporte à Percepção em Ambientes Cooperativos de Aprendizagem: Um Estudo de Caso no AMCOR". Programa de Pós Graduação em Engenharia Elétrica, Dissertação de Mestrado. Universidade Federal do Espírito Santo. Vitória - ES.
- MOODLE, moodle.org. disponível em: www.moodle.org. acesso em: 02 fev. 2010.
- MOODLE (2010). "Moodle, A Free, Open Source Course Management System for Online Learning". Disponível em: <http://docs.moodle.org>. Acesso em 15/07/2010.
- MYSQL (2010). "The World's Most Popular Open Source Database". Disponível em <http://www.mysql.com>. Acesso em 15/07/10.
- NETTO, José Francisco de Magalhães. "Uma Arquitetura para Ambientes Virtuais de Convivência: uma Proposta Baseada em Sistemas Multiagente". Tese de Doutorado, Universidade Federal do Espírito Santo. Vitória, 2006.
- PHP (2010). "PHP: Hipertext Preprocessor". Disponível em <http://www.php.net>. Acesso em 15/07/2010.
- PIMENTA, M. C. ; NETTO, J. F. M. "Agendamento de Atividades e Tarefas Apoiado por Sistemas Multiagente em Ambientes Virtuais de Aprendizagem". I Escola Regional de Informática - Regional Norte 1. Manaus:2009.

- PULINO, A. R., “Moodle, um sistema de gerenciamento de cursos”, Brasília/DF: Departamento de Engenharia Civil e Ambiental, Universidade de Brasília. Sob Licença da Creative Commons, 2005.
- RUSSEL, Stuart; NORVIG, Peter: “Inteligência Artificial”. Editora Campus, São Paulo, 2004.
- TORRES, Camila Costa. “A educação a distância e o papel do tutor: contribuições da ergonomia”. 198 f. Tese (Doutorado em Psicologia) – Instituto de Psicologia, Universidade de Brasília, Brasília, 2007.
- VIDAL, Elisabete – “Ensino a Distância vs Ensino Tradicional”. – Universidade Fernando Pessoa, Porto, 2002.
- SANTOS, Leonardo Nascimento dos. “Desenvolvimento de um Multi-Organizador Flexível de Espaços Virtuais”. 2009. 93 f. Dissertação (Mestrado em Informática). Universidade Federal do Amazonas, Manaus - AM.
- SILVA, A. S. ; Gonçalves, A. N. A. ; SILVA, C. N. ; BRITO, S. R. . “O Uso de Agentes de Software para Notificações no Ambiente Virtual de Aprendizagem Moodle”. II MoodleMoot, Universidade Mackenzie, São Paulo, 2008.
- SPÓSITO, Marcos André Fernandes. “Estação de Percepção: uma abordagem para o monitoramento em Ambientes Virtuais de Aprendizagem”. . 2008. 135 f. Dissertação (Mestrado em Informática). Universidade Federal do Amazonas, Manaus - AM.
- WAMPSEVER, Disponível em: www.wampserver.com. Acesso em 03 fev. 2010.
- WINSTON, P. H. “Artificial Intelligence.” Addison-Wesley, Reading, Massachusetts, third edition, 1992.
- WOOLDRIDGE, Michael. “An Introduction to Multiagent Systems”. Editora Wiley:England, 2002.
- XU, Dongming; WANG, Huaiqing. “Intelligent Agent Supported Personalization for Virtual Learning Environments”. Decision Support Systems: Volume 42, Issue 2, Pages 825-843, 2006.
- YANG, Kathleen, Pattan, Neha, *et al.* "Multi-agent Meeting Scheduling Using Mobile Context." Firt International ICST Conference. San Diego, CA, USA, 2009.

Apêndices

Apêndice A – Estrutura do protótipo Agendamento

Este apêndice apresenta algumas funções principais em PHP para realização do Agendamento. Por motivos de insuficiência de espaço não foi possível anexar a este trabalho todas as funções em PHP. Para tanto envie solicitação por correio eletrônico para *marcitacp@gmail.com*.

Para implementação do protótipo de Agendamento o mesmo é descrito em PHP, e sua validação é realizada por meio da estrutura de um banco de dados especificada no MySQL.

O Código abaixo apresenta a consulta principal que chama o agente na linguagem PHP.

```
<?php
// Funcao que retorna o dia da semana de uma data
function diasemana($data) {
    $ano = substr($data, 6, 4);
    $mes = substr($data, 3, 2);
    $dia = substr($data, 0, 2);
    $diasemana = date("w", mktime(0,0,0,$mes,$dia,$ano) );
    switch($diasemana) {
        case"0": $diasemana = "Domingo"; break;
        case"1": $diasemana = "Segunda"; break;
        case"2": $diasemana = "Terça"; break;
        case"3": $diasemana = "Quarta"; break;
        case"4": $diasemana = "Quinta"; break;
        case"5": $diasemana = "Sexta"; break;
        case"6": $diasemana = "Sbado"; break;
    }
    return($diasemana);
}
$data2=$_POST['date']; // data escolhida
$hora=$_POST['horario']; // hora escolhida
$codaluno=$_POST['aluno']; // cod. aluno
$semana=diasemana($data2); // dia da semana
$codtutor=$_POST['codtur']; // cod. tutor
/////data do agendamento
$ano = substr($data2, 6, 4);
$mes = substr($data2, 3, 2);
$dia = substr($data2, 0, 2);
$dataag=$ano."-".$mes."-".$dia;
//$curso=$COURSE->id; // cod. curso
require_once("cab.php");
```

```

require_once("config.php");
require_once("locallib.php");
// Definir o cod. curso que está trabalhando
$curso=2;
// Acesso banco
include "DB/conect.php";
// Mostrar todas as disp. cadastradas
$sql = "SELECT firstname, lastname, dia, turno, horario, email, data
      from mdl_user u, agenda a, agendamento ag
      where u.id=ag.id_aluno and ag.id_agenda=a.id_agenda";
$resultado = mysql_query($sql);
//procurar fazer o seguinte:
/*
- ao realizar a consulta, gravar a consulta cujo resultado estah abaixo em uma tabela temporária;
- nessa tabela, será feita a consulta pelos agentes;
- em caso positivo, gravar os dados necessários na tabela do agendamento ou equivalente;
- em caso negativo, gravar um registro nessa mesma tabela temporaria (ou em outra) indicativo que
nao deu certo
- nesse instante, será feita uma checagem se há ou não os registros especiais (dessa segunda
insercao)
para dizer se deu ou não certo
- deletar os registros das tabelas temporarias
*/
// Mostrar todos os usuarios do curso
$sqluser = "SELECT u.firstname, u.lastname, u.id
          FROM mdl_course_display c, mdl_user u
          WHERE c.userid=u.id and c.course='$curso'";
$resultado2 = mysql_query($sqluser);
// Consultar agenda do usuário agendado
$sqlagenda = "select d.id_agenda as cod
from mdl_user u, agenda a, disponibilidade d
where a.id_agenda=d.id_agenda and u.id=d.id_aluno and
a.dia='$semana' and a.horario='$hora' and u.id='$codaluno'";
/////
$resultado3 = mysql_query($sqlagenda);
$dados3=mysql_fetch_array($resultado3);

//passo 1: obter o codigo de agendamento do ususraio agendado

$codagenda=$dados3['cod'];

//passo 2: inserir na tabela auxiliar os dados que eu quero //consultar

/* Notar que o primeiro campo a ser inserido é justamente o ///$codagenda, que será comparado no
agente se é maior que 0 ou não. O segundo campo é o ja_usado, que pode assumir o valor 0 (não
usado) ou 1 (já usado). Isto serve para não haver manipulações repetidas no banco de dados,
consumindo mais tempo e evitando resultados errados. O terceiro campo, permite_agendamento,
indica se o agendamento pode ou não ser realizado (1 ou 0, respectivamente).
*/
$sqlaux = "insert into consulta_cod_agenda (cod, ja_usado, permite_agendamento) VALUES
('$codagenda', 0, 0) ";

```

```

$resultadoaux = mysql_query($sqlaux);

// passo 3: agora o sistema de agendamento espera um pouco //enquanto os agentes trabalham (o
passo 4 está no código dos //agentes, que ficam monitorando de 5 em 5 segundos)
//pode-se colocar uma mensagem de estah agendendo.....
sleep (20);

//passo 5: consultar o que o agente alterou, se ele permitiu ou //não permite agendamento

$sqlaux2 = "select cod from consulta_cod_agenda where ja_usado = 0 and permite_agendamento =
1 ";

$resultadoaux2 = mysql_query($sqlaux2);
$dadosaux3=mysql_fetch_array($resultadoaux2);
$codagenda2=$dadosaux3['cod'];

//passo 6 : atualizar tabela auxiliar para indicar que todos os //dados foram verificados e
trabalhados; não serão mais úteis

$sqlaux3 = " update consulta_cod_agenda set ja_usado = 1 where ja_usado = 0 ";

$resultadoaux3 = mysql_query($sqlaux3);

// passo 7: Verificar se existe disp. entre os usuários, pelos //resultados das consultas php e do
agente

if($codagenda = $codagenda2)
    $mensagem="";
else
    $mensagem="N&atilde;o &eacute; poss&iacute;vel agendar com essa pessoa";

// testes
// echo $codtutor."*".$codaluno."*".$codagenda."*".$dataag;
?>
<style type="text/css">
<!--
.style1 {font-weight: bold}
.style2 {color: #FF0000}
-->
</style>
<form action="consulta5.php" method="post">
<table width="777" border="0" align="center" bgcolor="#FFFFFF">
    <tr>
        <td colspan="5"></td>
    </tr>
    <input type="hidden" value="<?php echo $USER->id; ?>" name="codtur" >

</tr>

```

```

    <td height="26" colspan="4" bgcolor="#EEEEEE"><p class="style1"><strong>Usuario que esta
agendando</strong>: <span class="style15"><?php echo $USER->firstname." ".$USER->lastname;
?></span><br />
    <strong>Turma</strong>:
    <select name="aluno" id="aluno">
    <?php
// Transferir conteudo da query p/ combobox
while ($dados2=mysql_fetch_array($resultado2))
{
    $codigo=$dados2['id'];
    $nome=$dados2['firstname']." ".$dados2['lastname'];
    // Mostrar uma lista, sem o usuário logado
    if($codigo<>$codtutor)
        echo "<option value=$codigo>$nome</option>";
    }
    ?>
</select>
    <span class="style2"><?php echo $mensagem; ?></span><br/>
</td>
<input type="hidden" value="<?php echo $codagenda; ?>" name="codagenda" >
</tr>
<tr>
    <td height="32" colspan="5" bgcolor="#EEEEEE"><p class="style1"><strong>Data da
Atividade</strong>:

    <input name="date" type="text" id="campo_fecha" size="10" maxlength="10" />
    <input type="button" id="lanzador" value="..." />
    <strong>Horário</strong>:

    <select name="horario" id="horario">
    <option >Selecione ...</option>
    <option value="08 as 10 horas">08 as 10 horas</option>
    <option value="10 as 12 horas">10 as 12 horas</option>
    <option value="14 as 16 horas">14 as 16 horas</option>
    <option value="16 as 18 horas">16 as 18 horas</option>
    <option value="18 as 20 horas">18 as 20 horas </option>
    <option value="20 as 22 horas">20 as 22 horas </option>
    </select>
    <input type="submit" name="button" id="button" value="Agendar" /></td>
</tr>
<tr>
    <td width="303" bgcolor="#ACA185" class="style6"><div align="center" class="style7 style7">
    <div align="center"><span class="style3">NOME</span></div>
    </div></td>
    <td width="120" bgcolor="#ACA185" class="style6"><div align="center" class="style7 style7">
    <div align="center"><span class="style3">DATA</span></div>
    </div></td>
    <td width="113" bgcolor="#ACA185" class="style3"><div align="center" class="style7 style7">
    <div align="center"><span class="style3">HORÁRIO</span></div>
    </div></td>
    <td width="213" bgcolor="#ACA185" class="style3"><div align="center"
class="style3">EMAIL</div></td>

```

```

</tr>
<script type="text/javascript">
    Calendar.setup({
        inputField : "campo_fecha", // id del campo de texto
        ifFormat : "%d/%m/%Y", // formato de la fecha, cuando se escriba en el campo de texto
        button : "lanzador" // el id del botón que lanzará el calendario
    });
</script>
</form>
<?php
// Somente agendar se existir disp. entre usuarios
if($codagenda = $codagenda2)
{
    $qryagenda = "insert into agendamento(id_tutor, id_aluno, id_agenda, data)
values ('$codtutor','$codaluno','$codagenda','$dataag')";
    $qryagenda=mysql_query($qryagenda)
    or die ("Erro na gravação TABELA AGENDAMENTO!");
}
// Exibir resultado da consulta
while ($dados=mysql_fetch_array($resultado))
{
    $NOME=$dados['firstname']." ".$dados['lastname'];
    $DATA=$dados['data'];
    $HORARIO=$dados['horario'];
    $EMAIL=$dados['email'];
    $DATA = date("d/m/o", strtotime($DATA));
    // Mostrar valores
    echo"<tr class='style1'><td>$NOME</td>";
    echo"<td>$DATA</td>";
    echo"<td>$HORARIO</td>";
    echo"<td>$EMAIL</td></tr>";
}
?>
<tr>
    <td colspan="5" bgcolor="#ACA185"><div align="center"><span class="style3">SMAA - SISTEMA
MULTIAGENTE DE AGENDAMENTO<br />
    </span><span class="style11">COPYRIGHT 2010</span></div></td>
</tr>
</table>
</body>
</html>

```

O Código abaixo apresenta o cadastro das disponibilidades de horário na linguagem PHP.

```

<?php
require_once("cab.php");
require_once("config.php");
require_once("locallib.php");
// Acesso banco
include "DB/conect.php";
$codigo=$USER->id;
// Verificar se aluno j agendou
$qryverif = "select count(*) as qtd from disponibilidade where id_aluno='$codigo'";
$resultado = mysql_query($qryverif);
$res = mysql_fetch_array($resultado);
$num_reg = $res["qtd"];
if($num_reg>0)
{
echo "<div align='center'>";
echo "<p class='style3'>Prezado usu rio, voc  j cadastrou <b>$num_reg</b>
DISPONIBILIDADE(S)</p>";
echo "<p class='style3'><a href='#fechar' onClick='self.close()'>Fechar</a><br></div>";
}
else
{
echo"<table width='777' border='0' align='center' bgcolor='#FFFFFF'>";
echo"<tr><td colspan='3'><img src='ondeestou/disponibilidade.jpg' width='771' height='40'
/></td></tr>"
?>
<form action="gravar.php" method="post">
<tr>
<td height="26" colspan="3" bgcolor="#EEEEEE"><p class="style1"><strong>Nome</strong>:
<span class="style15"><?php echo $USER->firstname." ".$USER->lastname; ?></span><br />
</p> </td>
</tr>

<tr>
<td width="236" bgcolor="#ACA185"><div align="center" class="style7"><span
class="style10">TURNO MATUTINO</span></div></td>
<td width="284" bgcolor="#ACA185"><div align="center" class="style7"><span
class="style10">TURNO VESPERTINO</span></div></td>
<td width="243" bgcolor="#ACA185"><div align="center" class="style7"><span
class="style10">TURNO NOTURNO</span></div></td>
</tr>
<tr>
<input type="hidden" value="<?php echo $USER->id; ?>" name="codaluno" >
<td><input type="checkbox" name="semana[]" value="1">
<span class="style1">Segunda 8 as 10 horas</span></td>
<td><input type="checkbox" name="semana[]" value="3">
<span class="style1">Segunda 14 as 16 horas</span></td>
<td><input type="checkbox" name="semana[]" value="5">
<span class="style1">Segunda 18 as 20 horas</span></td>
</tr>
<tr bgcolor="#E7E3DA">
<td><span class="style1">
<input type="checkbox" name="semana[]" value="2">

```

```

Segunda 10 as 12 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="4">
    Segunda 16 as 18 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="6">
    Segunda 20 as 22 horas</span></td>
</tr>
<tr>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="7">
TerÃ§a 8 as 10 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="9">
    TerÃ§a 14 as 16 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="11">
    TerÃ§a 18 as 20 horas</span></td>
</tr>
<tr bgcolor="#E7E3DA">
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="8">
TerÃ§a 10 as 12 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="10">
    TerÃ§a 16 as 18 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="12">
    TerÃ§a 20 as 22 horas</span></td>
</tr>
<tr>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="13">
Quarta 8 as 10 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="15">
    Quarta 14 as 16 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="17">
    Quarta 18 as 20 horas</span></td>
</tr>
<tr bgcolor="#E7E3DA">
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="14">
Quarta 10 as 12 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="16">
    Quarta 16 as 18 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="18">
    Quarta 20 as 22 horas</span></td>
</tr>

```

```

<tr>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="19">
    Quinta 8 as 10 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="21">
    Quinta 14 as 16 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="23">
    Quinta 18 as 20 horas</span></td>
</tr>
<tr bgcolor="#E7E3DA">
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="20">
    Quinta 10 as 12 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="22">
    Quinta 16 as 18 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="24">
    Quinta 20 as 22 horas</span></td>
</tr>
<tr>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="25">
    Sexta 8 as 10 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="27">
    Sexta 14 as 16 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="29">
    Sexta 18 as 20 horas</span></td>
</tr>
<tr bgcolor="#E7E3DA">
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="26">
    Sexta 10 as 12 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="28">
    Sexta 16 as 18 horas</span></td>
  <td><span class="style1">
    <input type="checkbox" name="semana[]" value="30">
    Sexta 20 as 22 horas</span></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td colspan="3" bgcolor="#EEEEEE"><div align="center">
    <input type="submit" name="button" id="button" value="GRAVAR" />
  </div>
</td>
</tr>

```

```

    </form>
  </div> </td>
</tr>
<?php
}
?>
<tr>
  <td colspan="3" bgcolor="#ACA185"><div align="center" class="style9"><span
class="style1"><strong>SMAA - SISTEMA MULTIAGENTE DE AGENDAMENTO</strong><br />
  COPYRIGHT 2010</span></div></td>
</tr>
</table>
</body>
</html>

```

O Código abaixo apresenta a função que retorna dias e horários pra que o agendamento seja realizado.

```

<?php
// Funcao que retorna o dia da semana de uma data
function diasemana($data) {
  $ano = substr($data, 6, 4);
  $mes = substr($data, 3, 2);
  $dia = substr($data, 0, 2);
  $diasemana = date("w", mktime(0,0,0,$mes,$dia,$ano) );

  switch($diasemana) {
    case"0": $diasemana = "Domingo"; break;
    case"1": $diasemana = "Segunda"; break;
    case"2": $diasemana = "Tera"; break;
    case"3": $diasemana = "Quarta"; break;
    case"4": $diasemana = "Quinta"; break;
    case"5": $diasemana = "Sexta"; break;
    case"6": $diasemana = "Sbado"; break;
  }
  return($diasemana);
}
$data2=$_POST['date'];
$semana=diasemana($data2);
$hora=$_POST['horario'];
require_once("cab.php");
require_once("config.php");
require_once("locallib.php");

// Acesso banco
include "DB/conect.php";
// Consulta por data e hora
$sql = "SELECT firstname, lastname, dia, turno, horario, email
  from mdl_user u, agenda a, disponibilidade d

```

```

where a.id_agenda=d.id_agenda and u.id=d.id_aluno and
dia='$semana' and horario='$hora';

//and horario=$hora'

//Ejecutando consulta
$resultado = mysql_query($sql);
?>
<form action="consulta2.php" method="post">
<table width="777" border="0" align="center" bgcolor="#FFFFFF">
<tr>
<td colspan="5"></td>
</tr>
<tr>
<td height="32" colspan="5" bgcolor="#EEEEEE"><p class="style1"><strong>Data da
Atividade</strong>:

<input name="date" type="text" id="campo_fecha" size="10" maxlength="10" />
<input type="button" id="lanzador" value="..." />
<strong>Horário</strong>:

<select name="horario" id="horario">
<option >Selecione ...</option>
<option value="08 as 10 horas">08 as 10 horas</option>
<option value="10 as 12 horas">10 as 12 horas</option>
<option value="14 as 16 horas">14 as 16 horas</option>
<option value="16 as 18 horas">16 as 18 horas</option>
<option value="18 as 20 horas ">18 as 20 horas </option>
<option value="20 as 22 horas ">20 as 22 horas </option>
</select>
<input type="submit" name="button" id="button" value="Agendar" /></td>
</tr>
<tr>
<td width="303" bgcolor="#ACA185" class="style6"><div align="center" class="style7 style7">
<div align="center"><span class="style3">NOME</span></div>
</div></td>
<td width="120" bgcolor="#ACA185" class="style6"><div align="center" class="style7 style7">
<div align="center"><span class="style3">DATA</span></div>
</div></td>
<td width="113" bgcolor="#ACA185" class="style3"><div align="center" class="style7 style7">
<div align="center"><span class="style3">HORÁRIO</span></div>
</div></td>
<td width="213" bgcolor="#ACA185" class="style3"><div align="center"
class="style3">EMAIL</div></td>
</tr>
<script type="text/javascript">
Calendar.setup({
inputField : "campo_fecha", // id del campo de texto
ifFormat : "%d/%m/%Y", // formato de la fecha, cuando se escriba en el campo de texto
button : "lanzador" // el id del botón que lanzará el calendario
});
</script>

```

```
</form>
<?php
// Exibir resultado da consulta
while ($dados=mysql_fetch_array($resultado))
{
    $NOME=$dados['firstname']." ".$dados['lastname'];
    $DATA=$dados['data'];
    $HORARIO=$dados['horario'];
    $EMAIL=$dados['email'];
    $DATA = date("d/m/o", strtotime($DATA));
    // Mostrar valores
    echo"<tr class='style1'><td>$NOME</td>";
    echo"<td>$DATA</td>";
    echo"<td>$HORARIO</td>";
    echo"<td>$EMAIL</td></tr>";
}
?>
<tr>
    <td colspan="5" bgcolor="#ACA185"><div align="center"><span class="style3">SMAA - SISTEMA
MULTIAGENTE DE AGENDAMENTO<br />
    </span><span class="style11">COPYRIGHT 2010</span></div></td>
</tr>
</table>
</body>
</html>
```

Apêndice B – Agente do JADE mediador no protótipo

O código abaixo apresenta o agente *Databaseagent.java*, desenvolvido na linguagem de programação JAVA no framework JADE, que é responsável por mediar o agendamento realizado pelo banco de dados.

```
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import java.net.URLConnection;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

public class DatabaseAgent extends Agent {
    private Connection conn;
    private MessageParser mp;
    private ACLMessage lastReceivedMsg;

    private MessageTemplate template = MessageTemplate.and(
        MessageTemplate.MatchPerformative(ACLMessage.QUERY_IF),
        MessageTemplate.MatchOntology("presence") );
```

```

@Override
protected void setup() {
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/moodle","peter","secret");
        mp = new MessageParser();
    }
    catch (Exception e) {
        e.printStackTrace();
    }

    addBehaviour(new TickerBehaviour(this, 5000) {
        public void onTick() {
            createPerson();
            ACLMessage msg = myAgent.receive(template);
            if (msg != null) {
                Logger.getLogger(DatabaseAgent.class.getName()).log(Level.INFO,
                    "Msg. received: " + msg.getContent());
                // createPerson(mp.getFirstName(msg.getContent()),
                //     mp.getLastName(msg.getContent()),
                //     mp.getAge(msg.getContent()));

            }
            else {
                block();
            }
        }
    });
}

// private boolean createPerson(String firstName, String lastName, int age) {
private boolean createPerson() {

boolean rslt = true;

```

```

Integer maxIdx = null;
try {
    String resultado = " ";
    String resultado2 = " ";
    String resultado3 = " ";
    String resultado4 = " ";

    Statement stmt = conn.createStatement();
    //String stmtStr = "SELECT MAX(person_id) from person;";
    //consulta abaixo para saber o ultimo que acessou
    /*      String stmtStr = "SELECT firstname from mdl_user join mdl_user_lastaccess " +
        "on mdl_user_lastaccess.userid = mdl\_user.id where mdl_user_lastaccess.timeaccess =
        (select MAX(mdl_user_lastaccess.timeaccess)from mdl_user_lastaccess);";
    */
    /*  String stmtStr = "SELECT firstname FROM mdl_user JOIN mdl_user_lastaccess " +
        "ON mdl_user_lastaccess.userid = mdl\_user.id " +
        "WHERE mdl_user_lastaccess.timeaccess > ( SELECT MAX(
        mdl_user_lastaccess.timeaccess ) " +
        "FROM mdl_user_lastaccess ) -300;"; */

    //  String stmtStr = "SELECT max(id) from mdl_user";

    //passo de teste (não considerar)
    String stmtStr = "SELECT id_consulta_cod_agenda, cod from consulta_cod_agenda where " +
        " ja_usado = 0 and permite_agendamento = 0 ";

    ResultSet rs = stmt.executeQuery(stmtStr);
    /*  if (rs.first()) {
        // maxIdx = rs.getInt(1);
        resultado = rs.getString(1);

    } */

    if (rs.first()){

        resultado = rs.getString(1) + "\n";
    }
}

```

```

        resultado2 = rs.getString(2) + "\n";

        while(rs.next()){
            resultado = resultado + rs.getString(1) + "\n";
            resultado2 = resultado2 + rs.getString(2) + "\n";
        }

System.out.println(resultado);
System.out.println(resultado2);

//passo 4: agora, sobre os dados em que nao foram usados, permitir agendamento onde o cod > 0,
//indicando que pode haver um agendamento, pois o usuario solicitado tem disponibilidade
String stmtStr2 = " update consulta_cod_agenda set permite_agendamento = 1 where ja_usado = 0
    and permite_agendamento = 0 " +
" and cod > 0 ";

stmt.executeUpdate(stmtStr2)

// em seguida, o trabalho eh passado para o php
//String stmtStr3 = " update consulta_cod_agenda set ja_usado = 1 where ja_usado = 0 ";

//stmt.executeUpdate(stmtStr3);

        //int maxIndex = maxIdx.intValue();
        //maxIndex++;
        //maxIdx = new Integer(maxIndex);

        /*stmtStr = "INSERT INTO person VALUES(" + maxIdx.toString() +
            "," + "\"" + firstName + "\"" + "," +
            "\"" + lastName + "\"" + "," +
            age + ")";
*/
        // rslt = stmt.executeUpdate(stmtStr);

```

```
} catch (SQLException ex) {  
    Logger.getLogger(DatabaseAgent.class.getName()).log(Level.SEVERE, null, ex);  
    rslt=false;  
}  
  
return rslt;  
}  
}
```

Apêndice C – Tabelas do banco MySQL designadas ao protótipo

1. Tabela Agenda

Campo	Tipo	Nulo	Padrão
id_agenda	int(11)	Não	
dia	varchar(10)	Não	
turno	varchar(20)	Não	
horario	varchar(20)	Não	

Índices:

Nome chave	Tipo	Cardinalidade	Campo
PRIMARY	PRIMARY	30	id_agenda

Uso do espaço:

Tipo	Uso
Dados	1,224 Bytes
Índice	2,048 Bytes
Total	3,272 Bytes

Estatísticas da coluna:

Comandos	Valor
Formato	dinâmico
Colunas	30
Tamanho da coluna \emptyset	40
Tamanho do registro \emptyset	109 Bytes
Próximo Autoindex	31
Criação	Mai 03, 2010 as 11:03 AM
Última atualização	Mai 03, 2010 as 11:20 AM

2. Tabela Agendamento

Campo	Tipo	Nulo	Padrão
id_tutor	int(11)	Não	
id_aluno	int(11)	Não	
id_agenda	int(11)	Não	
data	date	Não	

Uso do espaço:

Tipo	Uso	
Dados	128	Bytes
Índice	1,024	Bytes
Total	1,152	Bytes

Estatísticas da coluna:

Comandos	Valor
Formato	fixo
Colunas	8
Tamanho da coluna \emptyset	16
Tamanho do registro \emptyset	144 Bytes
Criação	Mai 18, 2010 as 11:13 AM
Última atualização	Set 09, 2010 as 10:45 AM
Última verificação	Jun 18, 2010 as 11:57 AM

3. Tabela Consulta_Cod_Agenda

Campo	Tipo	Nulo	Padrão
id_consulta_cod_agenda	int(11)	Não	
cod	int(11)	Não	
ja_usado	int(11)	Não	
permite_agendamento	int(11)	Não	

Índices:

Nome chave	Tipo	Cardinalidade	Campo
PRIMARY	PRIMARY	4	id_consulta_cod_agenda

Uso do espaço:

Tipo	Uso	
Dados	68	Bytes
Índice	2,048	Bytes
Total	2,116	Bytes

Estatísticas da coluna:

Comandos	Valor
Formato	fixo
Colunas	4
Tamanho da coluna \emptyset	17
Tamanho do registro \emptyset	529 Bytes
Próximo Autoindex	5
Criação	Set 09, 2010 as 10:32 AM
Última atualização	Set 22, 2010 as 08:57 AM

4. Tabela Disponibilidade

Campo	Tipo	Nulo	Padrão
id_aluno	int(11)	Não	
id_agenda	int(11)	Não	

Índices:

Nome chave	Tipo	Cardinalidade	Campo
PRIMARY	PRIMARY	9	id_aluno
			id_agenda

Uso do espaço:

Tipo	Uso	
Dados	477	Bytes
Índice	2,048	Bytes
Sobrecarga	396	Bytes
Efetivo	2,129	Bytes
Total	2,525	Bytes

Estatísticas da coluna:

Comandos	Valor
Formato	fixo
Colunas	9
Tamanho da coluna \emptyset	9
Tamanho do registro \emptyset	281 Bytes
Criação	Mai 18, 2010 as 10:55 AM
Última atualização	Ago 05, 2010 as 04:18 PM