

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE  
ÁREA DE CIÊNCIA DA COMPUTAÇÃO

**JACLASON MACHADO VERAS**

**MODELAGEM PARA O SOFTWARE VIRTUAL-TANEB BASEADO NA TEORIA  
DA RESPOSTA AO ITEM PARA AVALIAR O RENDIMENTO DOS ALUNOS**

São Luís

2010

**JACLASON MACHADO VERAS**

**MODELAGEM PARA O SOFTWARE VIRTUAL-TANEB BASEADO NA TEORIA  
DA RESPOSTA AO ITEM PARA AVALIAR O RENDIMENTO DOS ALUNOS**

Dissertação de Mestrado apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Eletricidade, na área de Ciência da Computação.

Orientador: Prof. Dr. Sofiane Labidi

São Luís

2010

Veras, Jaclason Machado

Modelagem para o software virtual-taneb baseado na teoria da resposta ao item para avaliar o rendimento dos alunos/ Jaclason Machado Veras. – São Luís, 2010.

110 f.

Orientador: Prof. Dr. Sofiane Labidi

Dissertação (Mestrado) – Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Maranhão, 2009.

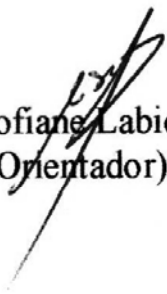
1. Modelagem de sistema de computação. 2. Aprendizagem - avaliação. 3. Modelagem de programas. 4. Educação básica - Matemática I. Título.

CDU 004.414.23:371.26

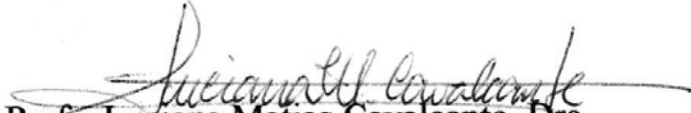
**MODELAGEM PARA O SOFTWARE VIRTUAL-TANEB  
BASEADO NA TEORIA DA RESPOSTA AO ITEM PARA  
AVALIAR O RENDIMENTO DOS ALUNOS**

**Jaclason Machado Veras**

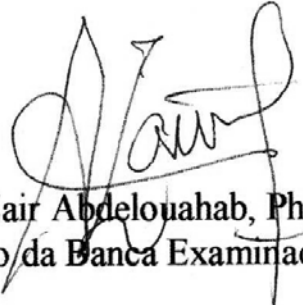
Dissertação aprovada em 26 de fevereiro de 2010.



Prof. Sofiane Labidi, Dr.  
(Orientador)



Prof. Luciana Matias Cavalcante, Dra.  
(Membro da Banca Examinadora)



Prof. Zair Abdelouahab, Ph.D.  
(Membro da Banca Examinadora)

Ao meu PRIMO INESQUECÍVEL  
Gerson Fontenele Machado

## **AGRADECIMENTOS**

Em primeiro lugar a Deus, por tudo de MARAVILHOSO que me tem concedido nos últimos anos.

Aos meus pais Sebastião e Alzira, por toda ajuda que me deram para estar realizando este SONHO que é de ser MESTRE.

Ao Prof. Dr. Nilson Santos, MEU AMIGO, pelos momentos de apoio durante as aflições enfrentadas NESTA LUTA.

Ao Prof. Dr. Sofiane Labidi, MEU ORIENTADOR E AMIGO, por ter me acolhido e confiado no meu trabalho a ser DESENVOLVIDO junto ao LSI.

À minha NAMORADA, Lílian Gomes, pela COMPREENSÃO, AMOR, DEDICAÇÃO e FORÇA demonstrada nos momentos difíceis vividos aqui nesta etapa de VIDA.

Ao meu PADRINHO, Vilson Fontenele, e minha MADRINHA, Maria de Fátima, pelo apóio dado ao chegar a São Luís para uma nova vida que seria construída.

Ao meu TIO, Osmar Fontenele, pela FORÇA e AJUDA dada no início da minha vida PROFISSIONAL.

Aos meus AMIGOS que aqui fiz, Lucas Drumond, Djeffersson Smith, Geraldo Abrantes.

Aos meus AMIGOS do LSI, Christian Diniz, Carlos Eduardo, Pedriana, Pedro Brandão, Paulo Cardoso, Adriano, Jonatas e Fernando.

Aos meus AMIGOS, Araújo, Nonato, Álvaro, Juçara, Janaína, Dona Teresa, Luzinária, Zé Pedro, Sérgio e Neide, que me deram FORÇA nos momentos difíceis antes de ingressar neste mestrado.

Ao meu AMIGO DE LONGOS DESAFIOS, Flávio Barros, pelo APOIO e DEDICAÇÃO nos momentos finais desse trabalho.

A todos os MEUS atuais e ex-colegas de trabalho.

A todos os AMIGOS ESPECIAIS, José Anchieta, Valdinéia, Lislely, Maria do Carmo, Lélia, Camila, Laércio, conquistados em São Luís e Teresina.

À “Baixinha” pelo café e pelo HUMOR FANTÁSTICO transmitido todas as tardes.

Enfim, a todas as pessoas, que DIRETA ou INDIRETAMENTE, contribuíram para a concretização deste SONHO.

"Mais importante que as riquezas naturais  
são as riquezas artificiais da educação e  
tecnologia."

Roberto Campos

## RESUMO

Esse trabalho propõe uma modelagem geral para um software educacional que atuará na verificação do conhecimento de geometria de alunos da quarta série do ensino fundamental. O Virtual-TANEB estará baseado no Sistema de Avaliação da Educação Básica (SAEB) que é um instrumento avaliativo usado no Brasil originado a partir das diretrizes do National Assessment of Educational Progress (NAEP), que, atualmente, é aplicado de forma manual por meio de cadernos de provas. Uma das motivações para o desenvolvimento dessa ferramenta foi reduzir o tempo de entrega dos resultados obtidos, através da resolução dos cadernos de provas, obedecendo à Matriz Curricular da Educação Básica. Outra motivação foi a inclusão da Teoria da Resposta ao Item (TRI) como quantificadora e qualificadora da avaliação educacional, permitindo uma análise estatística a partir de itens como elementos centrais da prova, visto a necessidade de acompanhar o processo de evolução dos sistemas de avaliação educacional. Porém, existem fatores que suprimem a utilização da TRI em larga escala, tais como: a personalização de cada aplicação e a complexidade matemática dos métodos envolvidos. Este protótipo de software educacional será aplicado às escolas do ensino fundamental no Estado do Maranhão, Brasil. Ele se adaptará ao nível cognitivo dos alunos através da inclusão de recursos provenientes da área de Inteligência Artificial e, além disso, possibilitará uma análise mais efetiva do conhecimento dos alunos usando a TRI sobre as respostas efetuadas. O Virtual-TANEB irá proporcionar alguns benefícios para a sociedade, como favorecer a longo prazo a redução de investimento econômico no processo de avaliação, resultando em melhorias para gerações presentes e futuras. Outro benefício é a adequação desse software a outros sistemas de avaliação, por exemplo, Programa Internacional de Avaliação de Alunos (PISA), Exame Nacional do Ensino Médio (ENEM), Prova Brasil, dentre outros. No futuro, mediante uma adaptação no software Virtual-TANEB, haverá a possibilidade de realizar a inclusão de outras disciplinas à prova.

**Palavras-Chave:** Educação. Avaliação Educacional. Aprendizagem. Sistemas Multiagentes. Teoria da Resposta ao Item.



## ABSTRACT

This work proposes a general modeling for an educational software which will serve to verify the knowledge of geometry of fourth grade students of elementary school. The Virtual-Taneb will be based on Basic Education Evaluation System (SAEB) which is an evaluative instrument used in Brazil originated from the guidelines of the National Assessment of Educational Progress (NAEP), which is currently applied manually by written tests. One of the motivations for developing this tool was to reduce the delivery time of obtained results through the resolution of written tests, obeying Basic Education Courseworks. Another motivation was the inclusion of the Item Response Theory (IRT) as an educational evaluation quantifier and qualifier, allowing a statistical analysis from items such as central elements of tests, since the need to monitor the development process of educational evaluation systems. However, there are factors that suppress the use of IRT on a large scale, such as customizing each application and the mathematical complexity of the methods involved. This prototype of educational software will be applied to elementary schools in the State of Maranhão, Brazil. It will be adapted to the cognitive level of students through the inclusion of recourses from the area of Artificial Intelligence and besides it will enable an effective analysis of students knowledge using the IRT on the responses made. Virtual-Taneb will provide some benefits to the Brazilian society, as to favor a long-term reduction of economic investment in the evaluation process, resulting in improvements for present and future generations. Another benefit is the adequation possibility of this software to other evaluation systems, as an example, the International Program of Students Assessment (PISA), the National Examination of High School (ENEM), Prova Brasil, among others. In the future, though the adaptation of the Virtual-Taneb software, there will be the possibility of adding other subjects to the test.

**Keywords:** Education. Educational Evaluation. Learning. Multiagent Systems. Item Response Theory.

## LISTA DE FIGURAS

|  |    |
|--|----|
| <b>Figura 1:</b> ICC para Item 1 .....   | 35 |
| <b>Figura 2:</b> Agente Genérico .....   | 40 |
| <b>Figura 3:</b> Peculiaridades de um Agente de Software .....                 | 41 |
| <b>Figura 4:</b> Comunicação “Quadro Negro” de um Agente de Software .....     | 43 |
| <b>Figura 5:</b> Interação na Troca de Mensagens.....                          | 44 |
| <b>Figura 6:</b> Mensagem KQML (ask-one) .....                                 | 46 |
| <b>Figura 7:</b> Mensagem FIPA-ACL (query-if) .....                            | 47 |
| <b>Figura 8:</b> Mensagem FIPA-ACL (inform) .....                              | 48 |
| <b>Figura 9:</b> Fases e Passos da Metodologia PASSI.....                      | 51 |
| <b>Figura 10:</b> Fases e Passos da Metodologia Gaia.....                      | 53 |
| <b>Figura 11:</b> Interface da Ferramenta AgentTool.....                       | 56 |
| <b>Figura 12:</b> Arquitetura do Virtual-TANEB .....                           | 61 |
| <b>Figura 13:</b> Fluxo de Informações.....                                    | 62 |
| <b>Figura 14:</b> Visão Geral do Virtual-TANEB .....                           | 63 |
| <b>Figura 15:</b> Diagrama de Casos de Uso “Cadastrar Itens” .....             | 64 |
| <b>Figura 16:</b> Diagrama de Casos de Uso “Selecionar Itens” .....            | 65 |
| <b>Figura 17:</b> Diagrama de Casos de Uso “Realizar Prova” .....              | 67 |
| <b>Figura 18:</b> Diagrama de Seqüência “Cadastrar Escola” .....               | 69 |
| <b>Figura 19:</b> Diagrama de Seqüência “Cadastrar Gestor” .....               | 70 |
| <b>Figura 20:</b> Diagrama de Seqüência “Cadastrar Func. Administrativos”..... | 71 |
| <b>Figura 21:</b> Diagrama de Seqüência “Cadastrar Professor e Pedagogo” ..... | 72 |
| <b>Figura 22:</b> Diagrama de Seqüência “Cadastrar Aluno”.....                 | 73 |
| <b>Figura 23:</b> Diagrama de Seqüência “Cadastrar Disciplinas”.....           | 74 |
| <b>Figura 24:</b> Diagrama de Seqüência “Cadastrar Itens”.....                 | 75 |
| <b>Figura 25:</b> Diagrama de Seqüência “Selecionar Itens” .....               | 75 |
| <b>Figura 26:</b> Trecho do Código para Cadastrar Usuário.....                 | 78 |
| <b>Figura 27:</b> Interface de Cadastro de Usuários no Virtual-TANEB.....      | 79 |
| <b>Figura 28:</b> Interface do Login no Virtual-TANEB .....                    | 80 |
| <b>Figura 29:</b> Interface dos Usuários Cadastrados no Virtual-TANEB .....    | 80 |
| <b>Figura 30:</b> Interface para Recuperar Senha no Virtual-TANEB.....         | 81 |
| <b>Figura 31:</b> Trecho do Código para Cadastrar Itens .....                  | 82 |
| <b>Figura 32:</b> Interface para Cadastro Itens no Virtual-TANEB.....          | 82 |

|   |    |
|---|----|
| <b>Figura 33:</b> Trecho do Código para Visualizar os Itens.....                | 83 |
| <b>Figura 34:</b> Interface dos Itens Cadastrados no Virtual-TANEB .....        | 83 |
| <b>Figura 35:</b> Trecho do Código do Agente Seletor .....                      | 84 |
| <b>Figura 36:</b> Trecho do Código sobre o Comportamento do Agente Seletor..... | 85 |

## LISTA DE TABELAS

|   |    |
|---|----|
| <b>Tabela 1:</b> Comparativo entre o SAEB e a Prova Brasil.....       | 28 |
| <b>Tabela 2:</b> Documentação do Caso de Uso “Cadastrar Itens” .....  | 65 |
| <b>Tabela 3:</b> Documentação do Caso de Uso “Selecionar Itens” ..... | 66 |
| <b>Tabela 4:</b> Documentação do Caso de Uso Realizar Prova.....      | 68 |

## LISTA DE ABREVIATURAS E SIGLAS

|                   |  |
|-------------------|--|
| <b>ACER</b>       | Australian Council for Educational Research                              |
| <b>ACL</b>        | Agent Communication Language   |
| <b>BD</b>         | Banco de Dados   |
| <b>CCI</b>        | Curva Característica do Item   |
| <b>CII</b>        | Curva de Informação de um Item   |
| <b>CITO</b>       | The Netherlands National Institute for Educational Measurement           |
| <b>CNE</b>        | Conselho Nacional de Educação  |
| <b>CONSED</b>     | Conselho Nacional do Livro Didático                                      |
| <b>DAEB</b>       | Diretoria de Avaliação da Educação Básica                                |
| <b>ENCCEJA</b>    | Exame Nacional para Certificação de Competências de Jovens e Adultos     |
| <b>ENEM</b>       | Exame Nacional do Ensino Médio   |
| <b>ETS</b>        | Educational Testing Service  |
| <b>FUNDESCOLA</b> | Fundo de Fortalecimento da Escola  |
| <b>GUI</b>        | Interface Gráfica com o Usuário  |
| <b>INEP</b>       | Instituto Nacional de Estudos e Pesquisas Educacionais “Anísio Teixeira” |
| <b>JADE</b>       | Java Agent Development Framework   |
| <b>JESS</b>       | Java Expert System Shell   |
| <b>JSP</b>        | JavaServer Pages   |
| <b>KQML</b>       | Knowledge Query and Manipulation Language                                |
| <b>LSI</b>        | Laboratório de Sistemas Inteligentes                                     |
| <b>MEC</b>        | Ministério da Educação   |
| <b>NIER</b>       | Japanese Institute for Educational Research                              |
| <b>OCDE</b>       | Organização para a Cooperação e Desenvolvimento Econômico                |
| <b>OMG</b>        | Object Management Group  |
| <b>PASSI</b>      | Process for Agents Societies Specification and Implementation            |
| <b>PDE</b>        | Plano de Desenvolvimento da Escola                                       |
| <b>PISA</b>       | Programa Internacional de Avaliação de Alunos                            |
| <b>PNLD</b>       | Programa Nacional do Livro Didático                                      |
| <b>ProUni</b>     | Programa Universidade para Todos   |
| <b>PTK</b>        | PASSI Tool Kit   |

|               |   |
|---------------|---|
| <b>PU</b>     | Processo Unificado  |
| <b>SAEB</b>   | Sistema de Avaliação da Educação Básica                           |
| <b>SARESP</b> | Sistema de Avaliação de Rendimento Escolar do Estado de São Paulo |
| <b>SMA</b>    | Sistemas Multiagentes   |
| <b>TCM</b>    | Teoria Clássica de Medidas  |
| <b>TRI</b>    | Teoria da Resposta ao Item  |
| <b>UML</b>    | Unified Modeling Language   |
| <b>VT</b>     | Virtual-TANEB   |

## SUMÁRIO

|   |    |
|---|----|
| LISTA DE FIGURAS .....  | 10 |
| LISTA DE TABELAS .....  | 12 |
| LISTA DE ABREVIATURAS E SIGLAS .....  | 13 |
| 1 INTRODUÇÃO .....  | 17 |
| 1.1 Contexto e Problemática.....  | 17 |
| 1.2 Relevância e Motivação .....  | 17 |
| 1.3 Objetivos.....  | 18 |
| 1.4 Estruturação.....   | 18 |
| 2 AS AVALIAÇÕES DA EDUCAÇÃO BÁSICA BRASILEIRA.....                                      | 20 |
| 2.1 Considerações Iniciais .....  | 20 |
| 2.2 Programa Internacional de Avaliação de Alunos (PISA).....                           | 21 |
| 2.3 Exame Nacional do Ensino Médio (ENEM) .....   | 22 |
| 2.4 Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA)..... | 24 |
| 2.5 Sistema de Avaliação da Educação Básica (SAEB).....                                 | 24 |
| 2.6 Prova Brasil.....   | 26 |
| 2.7 Estudo Comparativo entre o SAEB e a Prova Brasil.....                               | 27 |
| 2.8 Considerações Finais .....  | 29 |
| 3 TEORIAS DE MEDIDAS PARA AVALIAÇÃO EDUCACIONAL .....                                   | 30 |
| 3.1 Considerações Iniciais .....  | 30 |
| 3.2 Teoria Clássica de Medidas (TCM).....   | 31 |
| 3.3 Teoria da Resposta ao Item (TRI).....   | 33 |
| 3.4 Considerações Finais .....  | 38 |
| 4 AGENTES E SISTEMAS MULTIAGENTES.....  | 39 |
| 4.1 Considerações Iniciais .....  | 39 |
| 4.2 Agente de Software .....  | 40 |
| 4.3 Categoria de Agentes .....  | 42 |
| 4.4 Comunicação entre Agentes.....  | 43 |
| 4.4.1 Quadro de Avisos .....  | 43 |
| 4.4.2 Troca de Mensagens .....  | 44 |
| 4.5 Linguagens de Comunicação entre Agentes .....                                       | 45 |
| 4.5.1 KQML .....  | 46 |

|       |  |    |
|-------|--|----|
| 4.5.2 | FIPA-ACL .....   | 47 |
| 4.6   | Sistemas Multiagentes .....  | 48 |
| 4.7   | Metodologias e Ferramentas para o Desenvolvimento de Sistemas Multiagentes ..... | 50 |
| 4.7.1 | PASSI .....  | 50 |
| 4.7.2 | Gaia .....   | 52 |
| 4.7.3 | Tropos .....   | 54 |
| 4.7.4 | AgentTool .....  | 56 |
| 4.8   | Software na Educação que usam Agentes de Software .....                          | 57 |
| 4.9   | Considerações Finais .....   | 58 |
| 5     | MODELAGEM DO SOFTWARE VIRTUAL-TANEB .....  | 59 |
| 5.1   | Considerações Iniciais .....   | 59 |
| 5.2   | Arquitetura Geral do Virtual-TANEB .....   | 59 |
| 5.3   | Diagramas do Virtual-TANEB .....   | 62 |
| 5.3.1 | Diagrama de Casos de Usos .....  | 63 |
| 5.3.2 | Diagrama de Seqüência .....  | 68 |
| 5.4   | Considerações Finais .....   | 76 |
| 6     | IMPLEMENTAÇÃO DO SOFTWARE VIRTUAL-TANEB PARA A AVALIAÇÃO DA APRENDIZAGEM .....   | 77 |
| 6.1   | Considerações Iniciais .....   | 77 |
| 6.2   | Cadastrar no Virtual-TANEB .....   | 77 |
| 6.3   | Considerações Finais .....   | 85 |
| 7     | CONCLUSÃO .....  | 86 |
| 7.1   | Resultados Alcançados .....  | 86 |
| 7.2   | Principais Contribuições .....   | 87 |
| 7.3   | Perspectivas Futuras .....   | 87 |
|       | REFERÊNCIAS .....  | 89 |
|       | APÊNDICE .....   | 96 |



## **1 INTRODUÇÃO**

### **1.1 Contexto e Problemática**

Os sistemas de avaliações no Brasil são vistos como elementos de coleta de dados e informações estatístico-educacionais capazes de subsidiar a elaboração de propostas para a melhoria da qualidade do ensino. Dentre os sistemas de avaliação existentes no Brasil, encontramos o SAEB – Sistema de Avaliação da Educação Básica – que é um sistema responsável por verificar o nível de conhecimentos dos alunos, da quarta e oitava série do ensino fundamental, como dos alunos da terceira série do ensino médio, sobre determinados conhecimentos na Língua Portuguesa e Matemática.

As teorias de medidas, por sua vez, introduzem modelos estatísticos que são um modo bastante interessante de se analisar a aprendizagem dos alunos e as falhas existentes na estrutura escolar através de uma amostragem, mas em razão da escassez de soluções computacionais que façam esta avaliação, este processo de avaliação em alguns casos se torna inviável.

Diante deste contexto, nesta dissertação de mestrado propomos a execução da parte inicial do Projeto Virtual-TANEB (VT), tratando se então de uma modelagem para o software VT e a implementação do Agente Seletor, cuja finalidade do VT será, baseada na Teoria da Resposta ao Item, avaliar o rendimento dos alunos em relação ao ensino da geometria.

### **1.2 Relevância e Motivação**

No plano geral, a Teoria da Resposta ao Item é um tópico que merece ser profundamente pesquisado porque constitui, sem dúvida alguma, uma forma promissora de se utilizar modelos matemáticos no desenvolvimento de aplicações, visando um aperfeiçoamento aos sistemas de avaliação da educação básica existentes aqui no Brasil, mais precisamente no Sistema de Avaliação da Educação Básica (SAEB).

Já de maneira particular, pesquisar a Teoria da Resposta ao Item juntamente com o SAEB é de interesse do LSI (Laboratório de Sistemas Inteligentes – Departamento de Pós Graduação em Engenharia Elétrica da UFMA) porque

representam um complemento aos demais trabalhos que já vêm sendo atualmente desempenhados pelo grupo. No ponto em que se chegou, já há disponíveis estudos avançados para o desenvolvimento de um software baseado na Teoria da Resposta ao Item para fazer avaliações sobre o ensino da matemática na quarta série do ensino fundamental.

Desta forma, nossa principal motivação para a realização deste trabalho, tornando-o distinto de tantos outros em estágios relativamente mais avançados, é o fato da necessidade de recursos computacionais eficientes, de âmbito nacional, para aferir o nível de conhecimento dos alunos à área de geometria ensinada no ensino fundamental, independentemente da rede de ensino, seja ela pública ou particular.

### **1.3 Objetivos**

O objetivo geral do presente trabalho é desenvolver uma modelagem geral para o software Virtual-TANEB e implementar um agente que fará a seleção dos itens usados na elaboração da prova.

No sentido de alcançar o objetivo pretendido, buscar-se-á atingir os seguintes objetivos específicos:

- a) Avaliar a estrutura do Ambiente de Gestão Educacional dos Sistemas de Avaliação da Educação Básica;
- b) Aprimorar aspectos de avaliação relacionados ao SAEB.

### **1.4 Estruturação**

A estrutura dessa dissertação compreende sete capítulos, sendo eles, além desta introdução, os seguintes:

No Capítulo 2 traz uma revisão referente aos Sistemas de Avaliação Brasileiros para a Educação Básica, englobando desde a parte histórica até a forma de aplicação destas avaliações, passando ainda por uma análise detalhada dos diversos tipos de sistemas de avaliações realizados aqui no Brasil para avaliar o ensino brasileiro, são eles: Programa Internacional de Avaliação de Alunos (PISA), Exame Nacional do Ensino Médio (ENEM), Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA), Prova Brasil e o SAEB, este último

se configurando como um dos tópicos mais importantes para o embasamento do presente trabalho.

Já o Capítulo 3, aborda as Teorias de Medidas usadas por um sistema de avaliação, assunto igualmente fundamental no contexto da nossa pesquisa, onde se explica noções sobre os mecanismos de medidas executados através de uma amostragem; sintetiza-se como avaliar um aluno, identificando o seu rendimento através da análise das questões, em vez de analisar uma prova por completo; discute-se a Teoria da Resposta ao Item como um aperfeiçoamento a Teoria Clássica de Medidas.

No Capítulo 4, aborda os agentes de software e os sistemas multiagentes, assunto fundamental no contexto dessa pesquisa, onde há explicações básicas sobre os agentes de software e de sistemas multiagentes; sintetiza-se as categorias dos agentes; explica-se os mecanismos e as linguagens de comunicação entre os agentes; apresenta-se levantamento de metodologias e ferramentas para a construção de sistemas multiagentes; e, por último, exhibe alguns software que usam agentes para solucionar problemas complexos na área educacional.

O capítulo 5 por sua vez, descreve a modelagem geral do software Virtual-TANEB e que se caracteriza como uma das principais contribuições científica resultante deste trabalho.

Em seguida, o capítulo 6 apresenta uma implementação executada com o intuito de avaliar a modelagem apresentada no capítulo anterior, respectivamente, no setor de cadastrar todas as informações sobre os usuários, na etapa de cadastrar os itens em um banco de dados e por último, na fase do agente seletor.

Por fim, o Capítulo 7 apresenta as nossas conclusões acerca do trabalho, com ênfase nas principais contribuições, nos resultados alcançados e nas perspectivas futuras.

## **2 AS AVALIAÇÕES DA EDUCAÇÃO BÁSICA BRASILEIRA**

### **2.1 Considerações Iniciais**

Segundo (Soares e Alves 2003), mesmo antes de a Constituição brasileira de 1988 consagrar o princípio de que o acesso ao ensino básico é direito público subjetivo, tanto o governo federal como os estaduais e municipais vinham priorizando programas de construção de escolas e contratação de professores para atendimento de crianças de 7 a 14 anos.

A noção de qualidade no ensino traduz uma idéia complexa. Em todas as sociedades, a educação básica assume múltiplos objetivos, que são definidores desse princípio. Em geral, esses objetivos têm sido agregados em quatro grandes domínios: (1) o cognitivo, abrangendo a aquisição de competências intelectuais e domínio de diferentes áreas do conhecimento; (2) o vocacional, que inclui a aquisição das informações e habilidades necessárias à inserção no mundo do trabalho produtivo; (3) o social, relacionado com o preparo para a participação ética em uma sociedade plural e complexa; e (4) o pessoal, enfatizando o desenvolvimento de talentos pessoais, por exemplo, artísticos ou desportivos (Soares e Alves 2003).

O Instituto Nacional de Estudos e Pesquisas Educacionais “Anísio Teixeira” (INEP) é uma autarquia federal vinculada ao Ministério da Educação (MEC), cuja missão é promover estudos, pesquisas e avaliações sobre o Sistema Educacional Brasileiro com o objetivo de subsidiar a formulação e implementação de políticas públicas para a área educacional a partir de parâmetros de qualidade e equidade, bem como produzir informações claras e confiáveis aos gestores, pesquisadores, educadores e público em geral. Para gerar seus dados e estudos educacionais, o INEP realiza levantamentos estatísticos e avaliativos em algumas etapas da educação básica, assim como na modalidade de educação de jovens e adultos (Brasil 2008).

Com objetivo de aprimorar a qualidade nas instituições escolares a nível fundamental e médio, criou-se aqui no Brasil a partir do final década de 80, sistemas de avaliação com abrangência nacional para avaliar a qualidade no ensino básico. Com base neste aprimoramento, o Ministério da Educação do Brasil (MEC) tendo como parte integrante da estrutura organizacional do INEP a Diretoria de Avaliação

da Educação Básica (DAEB), tendo sob sua responsabilidade, além dos dois instrumentos fundamentais, SAEB e Prova Brasil, as seguintes avaliações para medir a qualidade do ensino: Programa Internacional de Avaliação de Alunos (PISA), Exame Nacional do Ensino Médio (ENEM) e Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA).

O objetivo deste capítulo é apresentar uma visão geral sobre os Sistemas de Avaliação da Educação Básica Brasileira. Além desta seção com as considerações iniciais, temos a seção 2.2 que apresenta o Programa Internacional de Avaliação de Alunos (PISA). A seção 2.3 mostra o Exame Nacional do Ensino Médio (ENEM). Em seguida, a seção 2.4 exhibe o Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA), enquanto a seção 2.5 relata a Prova Brasil. Já a seção 2.6 exhibe o Sistema de Avaliação da Educação Básica (SAEB). A seção 2.7 traz os softwares usados nestes sistemas de avaliação. A seção 2.8 demonstra um estudo comparativo entre o SAEB e a Prova Brasil. E por último, a seção 2.9 apresenta as considerações finais do capítulo.

## **2.2 Programa Internacional de Avaliação de Alunos (PISA)**

O Programa Internacional de Avaliação de Alunos (PISA) foi desenvolvido conjuntamente pelos países-membros da Organização para a Cooperação e Desenvolvimento Econômico (OCDE). Este programa é uma avaliação internacional de habilidades e conhecimentos de jovens de 15 anos, que visa aferir até que ponto os alunos próximos do término da educação obrigatória adquiriram conhecimentos e habilidades essenciais para a participação efetiva na sociedade (Inep 2001).

O PISA é implementado por um consórcio internacional liderado pelo *Australian Council for Educational Research* (ACER), da Austrália, que inclui *The Netherlands National Institute for Educational Measurement* (CITO), da Holanda, a *Westat* e o *Educational Testing Service* (ETS), dos Estados Unidos, o *Japanese Institute for Educational Research* (NIER), do Japão e no Brasil, ele é coordenado pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP), autarquia vinculada ao Ministério da Educação (MEC). O consórcio PISA 2000 reuniu especialistas de 30 países, com larga experiência internacional no desenho e execução de sistemas de avaliação (Inep 2001).

No PISA, além dos países membros da OCDE, alguns outros países são convidados a participar deste programa, dentre eles, podemos citar o Brasil e o México. Conforme (Brasil 2008), as avaliações do PISA são realizadas a cada três anos, tendo como objetivo central a capacidade de mensurar e comparar o desempenho dos jovens, as áreas de Linguagem, Matemática e Ciências, não somente quanto ao domínio curricular, mas também quanto aos conhecimentos relevantes e às habilidades necessárias à vida adulta.

Segundo (Inep 2001), a primeira avaliação foi realizada em 2000, com ênfase na Leitura, posteriormente novas avaliações foram executadas, enfocando respectivamente, a proficiência dos alunos as áreas de Matemática e de Ciências. A prova do PISA 2000 foi composta de um conjunto de questões (itens de teste) apresentado aos alunos na prova aplicada por meio de cadernos e questionários impressos. Os alunos leram vários tipos de textos e responderam a questões sobre os mesmos. Dentre as questões algumas foram de múltipla escolha e outras exigiram que os alunos construíssem suas próprias respostas. Já os testes objetivavam determinar se os alunos eram capazes de refletir e raciocinar sobre a área avaliada e não apenas se conseguiam reproduzir o conhecimento adquirido.

Portanto, O Programa Internacional de Avaliação de Alunos (PISA) visa produzir um conjunto abrangente de informações e análises, capazes de fornecer um perfil do desempenho dos alunos em associação com seu contexto social e educacional, e de indicar como esse desempenho evoluirá ao longo do tempo. Os relatórios emitidos destinam-se a dirigentes e agências governamentais, que poderão fazer uso dessas informações e análises para subsidiar suas decisões políticas (Inep 2001).

### **2.3 Exame Nacional do Ensino Médio (ENEM)**

Conforme (Maynart 2009), o Exame Nacional do Ensino Médio (Enem) foi criado em 1998 pelo Instituto Nacional de Estudos e Pesquisas Educacionais “Anísio Teixeira” (INEP), parte integrante do Ministério da Educação. Segundo o (Inep 2005), este exame tem como objetivo principal possibilitar uma referência para auto-avaliação, a partir das competências e habilidades que o estruturam.

O modelo de avaliação do Enem foi desenvolvido com ênfase na aferição das estruturas mentais com as quais construímos continuamente o conhecimento e não apenas na memória, que, importantíssima na constituição dessas estruturas, sozinha não consegue fazer-nos capazes de compreender o mundo em que vivemos. Há uma dinâmica social que nos desafia, apresentando novos problemas, questiona a adequação de nossas antigas soluções e exige um posicionamento rápido e adequado ao cenário de transformações imposto pelas mudanças sociais, econômicas e tecnológicas com as quais nos deparamos nas últimas décadas. Este cenário permeia todas as esferas de nossa vida pessoal, mobilizando continuamente nossa reflexão acerca dos valores, atitudes e conhecimentos que pautam a vida em sociedade (Inep 2005).

Neste exame podem participar os alunos que estão concluindo ou que já concluíram o ensino médio em anos anteriores, sendo assim, atualmente o Enem é utilizado como critério de seleção para estudantes que pretendem concorrer a uma bolsa no Programa Universidade para Todos (ProUni) e além disso, cerca de 500 universidades já usam o resultado do exame como critério de seleção para o ingresso no ensino superior, seja complementando ou substituindo o vestibular (Maynard 2009).

O Ministério da educação (MEC) estuda mudanças no Exame Nacional de Ensino Médio (Enem), o novo sistema deve possibilitar uma relação positiva entre o ensino médio e o ensino superior, proporcionando um exame nacional unificado, desenvolvido com base numa concepção da prova focada em habilidades e conteúdos mais relevantes, passaria a ser um importante instrumento de política nacional permitindo que milhões de alunos optem pelo ingresso a uma Universidade Pública Federal.

O novo Enem – 2009 será constituído das provas de Português, Matemática, Ciências Naturais e Ciências Humanas, além da prova de redação, sob recomendação do (MEC) a partir da edição de 2010, os conhecimentos de Física, Química e Biologia, associados à matriz de referência de Ciências da Natureza e suas Tecnologias, devem expressar integração crescente entre áreas, adequando-se à perspectiva interdisciplinar das competências e habilidades adotadas na matriz de referência correspondente. E, deliberou-se ainda, que as competências de conhecimento e uso estrangeira moderna só deverá ser objeto de avaliação a partir do Exame de 2010.

## **2.4 Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA)**

O Exame Nacional de Certificação de Competências de Jovens e Adultos (ENCCEJA) foi criado em 2002, para construir uma referência de avaliação para jovens e adultos que não puderam concluir os estudos em idade apropriada, atendendo à solicitação do Conselho Nacional de Educação (CNE) e do Conselho Nacional de Secretários de Educação (CONSED). Este exame é aplicado a brasileiros residentes no Brasil e no exterior, tendo por objetivo fundamental avaliar competências e habilidades básicas de jovens e adultos que não tiveram oportunidade de acesso à escolaridade regular na idade própria.

O participante faz uma prova e, atingindo a pontuação mínima exigida, recebe o certificado de proficiência, de instituições credenciadas, da etapa de ensino na qual se inscreveu (fundamental ou médio) (Brasil 2008). Nesta prova, os participantes poderão se inscrever para todas as áreas de cada etapa ou para apenas uma das áreas avaliadas e, dessa forma, obterem a certificação em cada prova separadamente. No nível fundamental, as disciplinas são: Língua Portuguesa, Língua Estrangeira Moderna, Educação Artística e Educação Física; História e Geografia; Matemática; Ciências Naturais. No ensino médio, a certificação abrange: Linguagens, Códigos e suas Tecnologias; Ciências Humanas e suas Tecnologias; Matemática e suas Tecnologias; e Ciências da Natureza e suas Tecnologias (Inep 2008). Além disso, o ENCCEJA visa sinalizar, para educadores, estudantes e interessados, a natureza e a função de uma avaliação de competências fundamentais ao exercício pleno da cidadania (Brasil 2008).

Diante do contexto apresentado, essa nova proposta de avaliação visa organizar e oferecer, aos parceiros que desejarem um exame que avalie competências e habilidades de jovens e adultos, situadas no conhecimento necessário para que eles possam conviver, participar e transformar o mundo em que vivem.

## **2.5 Sistema de Avaliação da Educação Básica (SAEB)**

No âmbito do Estado brasileiro, o interesse pela avaliação sistêmica na organização do setor educacional já se manifestava nos anos 30. Por outro lado,



(Azevedo 2000 e Waisenfisz 1991 apud Coelho 2008) esclarece que foi nos esboços de pesquisa e de planejamento educacional desenvolvido na gestão pública da educação brasileira que surgiu uma proposta inicial de um sistema nacional de avaliação ocorrendo no final dos anos 80, embora fosse objeto de interesse já na reforma dos anos 30 e estivesse presente, desde então, nos esboços de pesquisa e de planejamento educacional foram desenvolvidos.

De acordo com (Brasil 2001 apud Coelho 2008), foi nos meados dos anos 90 que a avaliação da educação básica foi implantada e se foi consolidando pela avaliação externa da escola pelo SAEB – Sistema de Avaliação da Educação Básica, com base em resultados da aprendizagem aferidos por recursos quantitativos, assim, ele foi desenvolvido para substituir o Sistema de Avaliação do Ensino Público de 1º Grau – SAEP, criado em 1987. A implantação do SAEB ocorreu com a primeira aplicação de provas e o levantamento de dados em nível nacional, em 1990. A segunda aplicação do SAEB foi em 1993 estruturada em três eixos de estudo: (1) rendimento do aluno; (2) perfil e prática docentes; (3) perfil dos diretores e formas de gestão escolar.

A partir de 1995, o SAEB incluiu em sua amostra o ensino médio e a rede particular de ensino com a aceitação de técnicas mais modernas de medição do desempenho dos alunos, também incorporou instrumentos de levantamento de dados sobre as características socioeconômicas, culturais e sobre os hábitos de estudo dos alunos e por fim, redefiniram as séries avaliadas, quarta e oitava séries do ensino fundamental e terceira série do ensino médio. Nesse período, também ocorreu à reestruturação e modernização do sistema de estatísticas e indicadores educacionais e a ampliação dos meios operacionais de centralização da avaliação educacional com a inclusão de exames nacionais: o Exame Nacional do Ensino Médio (ENEM) e o Exame Nacional de Certificação de Competências de Jovens e Adultos (ENCCEJA), citados anteriormente nas seções 2.3 e 2.4. A ação avaliativa também se estendeu aos programas governamentais como: o Programa Nacional do Livro Didático (PNLD) e o Fundo de Fortalecimento da Escola (FUNDESCOLA), neste especialmente com o projeto Plano de Desenvolvimento da Escola (PDE) (Brasil 2001 apud Coelho 2008).

Diante da necessidade de mecanismos que afirmam a qualidade real do ensino brasileiro básico, (Castro 1998) cita que a ênfase em processos de avaliação é hoje considerada estratégica como subsídio indispensável no monitoramento das

reformas e das políticas educacionais. Não há País no mundo preocupado em aumentar a eficiência, a equidade e a qualidade do seu sistema educacional que tenha ignorado a importância da avaliação como mecanismo de acompanhamento dos processos de reforma. Cada vez mais, atribui-se relevância tanto à avaliação institucional em suas diferentes dimensões (condições da infra-estrutura das instituições escolares; processos de gestão; formação, qualificação e produtividade dos recursos humanos, etc.), como em relação à avaliação de resultados (o que e como os alunos aprendem, quais os fatores associados ao rendimento escolar, impactos de fatores extra e intra – escolares na aprendizagem, etc.).

Então, baseado no contexto da concepção de sistemas de avaliação para a educação básica, (Brasil 2002 apud Coelho 2008) relata que o SAEB foi implantado e tem-se consolidado no campo das políticas públicas com o “principal objetivo de contribuir para a melhoria da qualidade da educação brasileira e para a universalização do acesso à escola, oferecendo subsídios concretos para a formulação, reformulação e o monitoramento das políticas públicas voltadas para a educação básica”.

Portanto, (Castro 1998) relata em seu trabalho que o Sistema Nacional de Avaliação da Educação Básica (SAEB), é responsável por uma avaliação em larga escala, aplicada em amostras de alunos de quarta e oitava séries do ensino fundamental e de terceira série do ensino médio, representativas de todas as Unidades da Federação, das regiões e do País. Além de identificar os níveis de desempenho e aprendizagem efetivamente alcançados pelos alunos, os resultados do SAEB permitem estabelecer algumas correlações importantes sobre os fatores associados ao rendimento escolar. Esta avaliação é feita utilizando-se uma grande quantidade de questões, o que lhe confere maior validade curricular, pois contempla uma amplitude maior de conteúdos e habilidades, abrangendo grande parte daquilo que é proposto nos currículos estaduais.

## **2.6 Prova Brasil**

Além dos exames citados anteriormente, existe ainda no Brasil, a Prova Brasil que se compõe de um conjunto de testes que avaliam Língua Portuguesa (competência leitora) e Matemática. A primeira edição, em 2005, foi realizada em 5.387 municípios de todas as unidades da federação. Mais de três milhões de

alunos, distribuídos em cerca de 40 mil escolas públicas urbanas, foram avaliados. Além dos testes, os alunos respondem a um questionário com informações sobre seu contexto social e capital cultural (Delmanto 2007).

A Prova Brasil é utilizada para avaliar alunos da quarta e oitava séries do ensino fundamental, contendo itens que permitem medir a competência de leitura em Língua Portuguesa e a competência em resolução de problemas em Matemática. Os resultados obtidos são apresentados em escalas de desempenho, por níveis. Pode-se pensar nas escalas da Prova associando-as a um termômetro: em ambos os casos, não se começa a contar do zero e, quanto mais alto o nível a que se chega, maior é o desempenho. Sendo assim, a interpretação pedagógica da Prova Brasil, ou seja, a análise dos resultados sobre os níveis de aprendizagem em que se encontra a maioria dos alunos traz pistas valiosas. Sua interpretação, junto com a análise dos resultados das avaliações aplicadas pelos professores, permite que equipes escolares revejam projetos pedagógicos e que os docentes possam definir mais claramente metas de aprendizagem e objetivos de ensino (Delmanto 2007).

Portanto, segundo (Brasil 2008), a Prova Brasil tem por objetivo oferecer aos professores, diretores, coordenadores e gestores das redes de ensino um instrumento para diagnosticar o nível de alfabetização dos alunos, ainda no início da educação básica, viabilizando, assim, a elaboração de ações que visem sanar as possíveis insuficiências apresentadas nas áreas de leitura e escrita.

## **2.7 Estudo Comparativo entre o SAEB e a Prova Brasil**

A partir do estudo dos dois principais sistemas de avaliação para este trabalho, o SAEB e Prova Brasil, abordados anteriormente nas seções 2.5 e 2.6, apresentaremos aqui um comparativo entre eles. Para realizar um comparativo é fundamental a escolha de critérios de comparação relevantes e que sirvam para evidenciar as características pelas quais os processos se diferenciam e as que possuem em comum. Esses critérios e correspondente avaliação foram definidos a partir de uma adaptação feita, seguindo um estudo comparativo já realizado por (Condé 2007) em seu trabalho. O comparativo é apresentado na Tabela 1.

A Tabela 1 apresenta as características dos sistemas de avaliação referentes à estruturação e forma de aplicação. Esse critério se refere ao conjunto de ações que são realizadas para produzir uma avaliação do ensino. As

características analisadas para esse critério são o tipo das disciplinas avaliadas (língua portuguesa e matemática); as escolaridades que as avaliações cobrem (quarta e oitava séries do ensino fundamental e terceira série do ensino médio); forma de obtenção dos resultados; estrutura dos cadernos de provas e o suporte de uma teoria de medida para auxiliar a realização da análise sobre a resolução da prova.

**Tabela 1 – Comparativo entre os Sistemas de Avaliação SAEB e Prova Brasil**

| <b>SISTEMAS DE AVALIAÇÃO DA EDUCAÇÃO BÁSICA</b> |  |   |
|---|--|---|
|   | <b>SAEB</b>  | <b>PROVA BRASIL</b>   |
| <b>Disciplinas</b>                              | Língua Portuguesa e Matemática   | Língua Portuguesa e Matemática  |
| <b>Escolaridade</b>                             | quarta e oitava série do Ensino Fundamental e terceira série do Ensino Médio | quarta e oitava série do Ensino Fundamental                           |
| <b>Resultados</b>                               | Por Estratos Amostrais   | Por Escola  |
| <b>Caderno de Provas</b>                        | Contém testes de apenas uma das disciplinas                                  | Contém testes de Língua Portuguesa e de Matemática                    |
| <b>Quantidade de Cadernos</b>                   | 26 Cadernos – 13 Blocos de 13 itens para todas as séries                     | 21 Cadernos – 7 blocos de itens de 10 itens (quarta) e de 12 (oitava) |
| <b>Quantidade de Itens no Caderno</b>           | 39 itens por caderno   | 40 itens para a quarta e 48 itens para a oitava por caderno           |
| <b>Resolução da Prova</b>                       | Cada aluno responde a um teste de uma única disciplina                       | Cada aluno responde a testes de ambas as disciplinas                  |
| <b>Teoria de Medida</b>                         | Teoria da Resposta ao Item (TRI)   | Teoria da Resposta ao Item (TRI)                                      |

Na tabela, pode-se observar que ambos os sistemas avaliam o ensino fundamental, com exceção do SAEB que avalia também o ensino médio. Quanto às disciplinas a serem cobertas pelos sistemas, todos cobrem as disciplinas de Língua Portuguesa e Matemática. Os resultados da avaliação estão presentes em forma de estratos amostrais pelo SAEB e por escola no caso da Prova Brasil. Quanto à estrutura dos cadernos de provas, no caso do SAEB ele traz 39 itens por caderno e

a Prova Brasil traz 40 itens para a quarta e 48 itens para a oitava por caderno. E também, no SAEB este caderno contém testes de apenas uma das disciplinas e já na Prova Brasil os cadernos contém testes de Língua Portuguesa e de Matemática.

E por último, quanto à teoria de medida vemos que ambos os sistemas utilizam a Teoria da Resposta ao Item no auxílio da análise de desempenho na resolução dos itens na prova.

## **2.8 Considerações Finais**

Este capítulo apresentou os principais sistemas de avaliação para a educação básica utilizados aqui no Brasil. Assim, em primeiro lugar, foram discutidos conceitos relacionados ao Programa Internacional de Avaliação de Alunos (PISA). Em seguida, o Exame Nacional do Ensino Médio (ENEM) foi analisado, já que se constitui em uma forma importante de se promover a avaliação da educação brasileira. O Exame Nacional para Certificação de Competências de Jovens e Adultos (ENCCEJA), bem como o Sistema de Avaliação da Educação Básica (SAEB) e a Prova Brasil, também foram detalhadas. Por fim, foram descritas um comparativo entre o SAEB e a Prova Brasil.

Os tópicos acima listados são importantes como referencial teórico para o desenvolvimento do restante do trabalho, posto que todo ele tenha embasamento sobre os sistemas de avaliação da educação brasileira, focando especificamente a abordagem do Sistema de Avaliação da Educação Básica (SAEB) com a utilização de uma teoria de medida para sondar a capacidade dos alunos em responder questionamentos (provas) relacionados ao ensino da Língua Portuguesa e da Matemática.

O próximo capítulo tratará das teorias de medidas, outro assunto fundamental para a evolução do tema proposto.

### 3 TEORIAS DE MEDIDAS PARA AVALIAÇÃO EDUCACIONAL

#### 3.1 Considerações Iniciais

A avaliação educacional brasileira tem procurado alternativas que possam melhorar o diagnóstico, tanto em relação ao nível de conhecimento dos alunos em relação a um determinado assunto, como também, apontar falhas na infra-estrutura escolar. Assim, esta análise proporciona um direcionamento mais eficaz nas soluções as serem aplicadas nos problemas apontados.

O manuseio de teorias de medidas, representadas por modelos matemáticos, surgido nas últimas décadas, apresenta-se como uma interessante forma de se lidar com as análises a serem feitas sobre a representação do desempenho dos alunos na realização de uma prova do ensino fundamental, visto que, elas permitem um melhor entendimento e uma maior eficiência na observação das respostas dos alunos. Além disso, essa abordagem torna-se bastante adequada para facilitar o intercâmbio de informações entre os professores, pedagogos e os alunos nos domínios da matemática e da língua portuguesa do ensino básico brasileiro.

Segundo (Andrade et al. 2000), os resultados obtidos em provas, expressos apenas por seus escores brutos ou padronizados, têm sido tradicionalmente utilizados nos processos de avaliação e seleção de indivíduos. No entanto, os resultados encontrados dependem do particular conjunto de itens (questões) que compõem o instrumento de medida, ou seja, as análises e interpretações estão sempre associadas à prova como um todo, o que é a característica principal da Teoria Clássica de Medidas (TCM). Assim, torna-se inviável a comparação entre indivíduos que não foram submetidos às mesmas provas, ou pelo menos, ao que se denomina de formas paralelas de testes. Pensando nesse problema, que surgiu a Teoria da resposta ao Item (TRI) trazendo complemento a teoria anterior citada como: a comparação entre indivíduos que foram submetidos a diferentes provas, ou seja, as análises e interpretações estão sempre associadas aos itens (questões), o que é a característica principal da Teoria da Resposta ao Item (TRI).

Assim, várias questões de interesse prático na área da Educação podem ser respondidas. É possível, por exemplo, avaliar o desenvolvimento de uma

determinada série de um ano para outro ou comparar o desempenho entre escolas públicas e privadas (Andrade et al. 2000).

O objetivo deste capítulo é apresentar uma visão geral sobre as Teorias de Medidas, em particular, a Teoria Clássica de Medidas (TCM) e a Teoria da Resposta ao Item (TRI). Além desta seção com as considerações iniciais, temos a seção 3.2 que apresenta a Teoria Clássica de Medidas (TCM). A seção 3.3 mostra a Teoria da Resposta ao Item. E por último, a seção 3.4 apresenta as considerações finais do capítulo.

### 3.2 Teoria Clássica de Medidas (TCM)

A Teoria Clássica da Medida (TCM) é o modelo mais usual para medição de traços latentes em ciências sociais, psicologia, educação e áreas afins (Embretson e Reise 2000, Pasquali 2003 e Reise et al. 2005 apud Tezza et al. 2009). Segundo (Hayes 1992 apud Tezza et al. 2009), a equação básica da TCM, conhecida como modelo clássico de medida, descreve a relação entre os escores observados, os escores verdadeiros e o erro:

$$X = T + E$$

onde: X é o escore observado;

T é o escore verdadeiro;

E é o erro de medida.

Dentro da TCM, (Weiss 1983 apud Tezza et al. 2009) afirma que o escore verdadeiro (T) representa o traço latente. O erro (E), segundo (Hambleton et al. 1991 apud Tezza et al. 2009) pode estar associado aos seguintes fatores:

- Impossibilidade de incluir um número infinito de itens no conjunto de itens;
- Impossibilidade de aplicar o conjunto de itens um número infinito de vezes.
- Impossibilidade de submeter o conjunto de itens a um número infinito de organizações.

Na TCM, informações do item e suas correlações, bem como a confiabilidade dos resultados (consistência interna da escala), validade e erros-

padrões, são dependentes das características particulares da população. Sendo assim, o escore observado ( $X$ ) é definido por um conjunto particular de itens em uma simples medida.

Esta dependência direta dos resultados do teste com a amostra e o instrumento configura uma preocupação do ponto de vista da validade do modelo gerado. Conforme (Thurstone 1928) já atentava para esta limitação: — um instrumento de medida, na sua função de medir, não pode ser seriamente afetado pelo objeto de medida. Na extensão em que sua função de medir for assim afetada, a validade do instrumento é prejudicada ou limitada. Segundo (Pasquali e Primi 2003) afirmam que, na elaboração de um instrumento de medida utilizando-se a TCM, os parâmetros dos itens dependem da amostra em que eles foram calculados. Assim, um item qualquer do conjunto se torna mais difícil ou mais fácil dependendo da amostra e, portanto, o parâmetro de dificuldade do item vai variar de pesquisa para pesquisa em função da amostra. Um exemplo desta limitação pode ser verificado na avaliação de intangíveis de uma empresa (Vargas et al. 2008 apud Tezza et al. 2009), se o instrumento não for independente da amostra, um conjunto qualquer de itens que avalie, por exemplo, o grau de desenvolvimento do capital humano será considerado muito fácil se a empresa possuir uma estrutura de treinamento e muito difícil se a empresa não possuir. Todavia, este problema configura mais um problema de amostragem do que de análise. Segundo (Pasquali 1996 apud Tezza et al. 2009), um problema mais grave ocorre com o cálculo do parâmetro de discriminação do item. Na TCM, a estimação deste parâmetro é feita baseada no escore total de um teste, seja utilizando grupos-critérios ou coeficientes de correlação.

Sendo assim, ao se utilizar o escore total, supõe-se que todos os itens sejam adequados, o que nem sempre é verdade. (Hambleton e Swaminathan 1985 apud Tezza et al. 2009) levantam outra limitação da TCM, que é a suposição de que a variação dos erros de medida ( $E$ ) é a mesma para todos os respondentes, o que parece ser uma suposição de difícil sustentação. (Embretson e Reise 2000 apud Tezza et al. 2009) exemplificam que um conjunto com itens medianamente fáceis poderá diferenciar mais a prática organizacional ou o sujeito com habilidade média, mas não irá diferenciar da mesma maneira os que tiverem habilidade superior, que provavelmente obterão escores perto dos mais altos. Conseqüentemente, o erro de medida ( $E$ ) neste segundo grupo será maior que no primeiro.



Tendo em vista estas limitações, (Pasquali e Primi 2003) argumentam que, dentro dos moldes da TCM, a tarefa de comparar os sujeitos ou organizações de forma vertical ou longitudinal é difícil, uma vez que obter um escore total em um conjunto de itens fáceis não é a mesma coisa do que obter o mesmo escore em um conjunto de itens difíceis. Para equacionar estes e outros problemas da TCM, tem-se trabalhado, mas mais diversas áreas do conhecimento com a Teoria da Resposta ao Item, a apresenta várias vantagens em relação à TCM.

### **3.3 Teoria da Resposta ao Item (TRI)**

Os estudos sobre a Teoria da Resposta ao Item (TRI) surgiram na década de 1960 com objetivo de disponibilizar recursos não encontrados em outras teorias, como por exemplo, na Teoria Clássica de Medidas (TCM), trazendo informações mais ricas para análise de testes (Andrade e Justino 2007).

A Teoria da Resposta ao Item (TRI) é um conjunto de modelos matemáticos e estatísticos que são utilizados para na análise de itens e escalas, para criar e administrar medidas, e para medir indivíduos ou organizações em um construto (traço latente) de interesse. A TRI é sustentada basicamente por três fundamentos, são eles: função resposta de um item, função de informação e invariância (Reise et al. 2005 apud Tezza et al. 2009 ).

No Brasil, a TRI foi usada pela primeira vez em 1995 na análise dos dados do Sistema Nacional de Ensino Básico (SAEB). A introdução da Teoria da Resposta ao Item permitiu que os desempenhos de alunos de quarta e oitava séries do Ensino Fundamental e de terceira série do Ensino Médio pudessem ser comparados e colocados em uma escala única de conhecimento. Foi, a partir dos resultados obtidos no SAEB, que outras avaliações em larga escala, como por exemplo, o Sistema de Avaliação de Rendimento Escolar do Estado de São Paulo (SARESP), também foram planejadas e implementadas de modo a serem analisadas através da TRI (Andrade et al. 2000).

Na área educacional, houve o crescimento do interesse em avaliar e selecionar alunos observando suas habilidades individuais através de um embasamento teórico mais consistente. A TCM (Andrade et al. 2000 apud Fernandes et al. 2008), propõe uma solução a este problema. Contudo, os resultados encontrados dependem de um conjunto de elementos que compõem o

instrumento de medida, as análises e interpretações são sempre associadas à prova em sua totalidade, o que é a característica principal desta metodologia, tornando inviável a comparação entre indivíduos que não foram submetidos às mesmas provas (Fernandes et al. 2008).

De acordo com (Alexandre et al. 2002), a TRI permite a comparação entre populações submetidas a provas diferentes, desde que essas provas tenham algumas questões comuns, ou, ainda, a comparação entre indivíduos da mesma população que tenham sido submetidos a provas totalmente diferentes. Isso porque uma das características principais da Teoria da Resposta ao Item é analisar como elemento central as questões que compõem uma prova. Como exemplo, suponha-se que seja preciso comparar o nível de conhecimento entre alunos de duas séries distintas (quarta e oitava séries do ensino fundamental, por exemplo). Na TCM essa comparação somente será possível caso seja aplicada a mesma prova para as duas turmas. Na TRI, são necessárias apenas algumas questões em comum nas provas aplicadas às duas séries. Sendo assim, a TRI é um modelo complementar a TCM (Andrade et al. 2000).

Conforme (Tezza et al. 2009), além da área educacional, a Teoria da Resposta ao Item tem sido aplicada em outros segmentos, como: serviços, gestão pela qualidade total, qualidade de vida, avaliação de intangíveis nas organizações, usabilidade em sites de e-commerce.

A TRI assume a existência de um traço latente, isto é, uma habilidade que não pode ser diretamente observável, sendo difícil de mensurá-la. Entretanto, a habilidade pode ser estimada a partir de um modelo matemático que leva em consideração características das questões e a probabilidade do indivíduo acertar uma determinada questão dado o seu grau de habilidade (Andrade e Justino 2007).

Para (Pasquali et al. 2003), o traço latente é a causa e a resposta do indivíduo é o efeito. Agora, para se poder estimar, a partir da resposta dada pelo sujeito, o seu nível no traço latente, é preciso que se hipotetizem relações entre as respostas observadas do sujeito e o seu nível neste mesmo traço latente. Quando estas relações são expressas numa equação matemática, constando de variáveis e de constantes, temos um modelo ou teoria do traço latente. Como tanto as variáveis e constantes que entram numa tal equação, quanto às formas matemáticas que as curvas, que expressam a relação hipotetizada, podem ser as mais variadas, segue que, em princípio, existe um número sem fim de tais equações possíveis. A TRI se

decidiu por algumas destas equações que achou mais adequadas ou produtivas, como veremos mais adiante.

De qualquer forma, o fundamental da teoria do traço latente consiste em expressar numa fórmula matemática a relação existente entre variáveis observadas e variáveis hipotéticas, chamadas estas de traços latentes. Assim, se conhecemos as características das variáveis observadas (como os itens de um teste), estas se tornam constantes na equação e esta se torna solucionável, permitindo que se estime então o nível do traço latente ou a aptidão do sujeito e vice-versa, isto é, se for conhecido o nível do traço latente é possível serem estimadas as características dos itens respondidos por este sujeito (Pasquali e Primi 2003).

Assim, de acordo com (Pasquali e Primi 2003), a TRI faz dois postulados básicos, são eles:

1º - desempenho do sujeito numa tarefa (item de um teste) pode ser predito a partir de um conjunto de fatores ou variáveis hipotéticas, ditos aptidões ou traços latentes (identificados na TRI com a letra grega teta:  $\theta$ ); o teta sendo a causa e o desempenho o efeito. Trata-se de modelagem latente (*latent trait modeling*). Ou seja, comportamento = função (traço latente)

2º - a relação entre o desempenho e os traços latentes pode ser descrita por uma equação matemática monotônica crescente, chamada de *Curva Característica do Item – CCI* (Vide Figura 1).

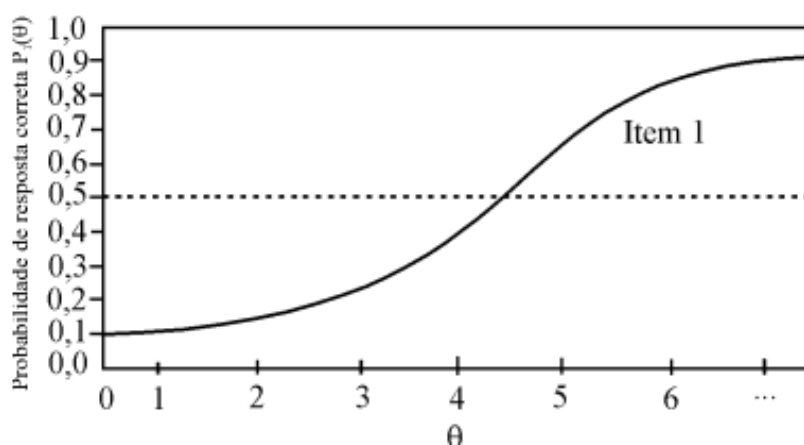


Figura 1 – CCI para Item 1.

A Figura 1 apresentada anteriormente mostra que à medida usada para aumentar o teta, também serve para aumentar a probabilidade de acertar o item (relação monotônica crescente entre aptidão e probabilidade de acerto).

Na Teoria da Resposta ao Item, existem vários modelos não-lineares. Entretanto, os modelos logísticos de respostas para itens dicotômicos são os mais utilizados. Este modelo possui os seguintes tipos de parâmetros: o de número 1 que representa somente a dificuldade do item, o de número 2 que simboliza a dificuldade e discriminação e, por último, o de número 3 que denota a dificuldade, discriminação e probabilidade de resposta correta dada por indivíduos de baixa habilidade (Andrade et al. 2000).

Sendo assim, é discutido no presente trabalho o modelo logístico de três parâmetros, que segundo (Andrade e Justino 2007), o modelo matemático de função logística de três parâmetros é geralmente o mais utilizado, pois, caracteriza cada questão em três parâmetros, conforme pode ser visto em na equação (1):

$$P(\phi) = c + (1 - c) \frac{1}{1 + e^{-a(\phi - b)}}$$

onde:  $b$  é o parâmetro de dificuldade;

$a$  é o parâmetro de discriminação;

$c$  é o parâmetro de acerto ao acaso;

$\phi$  é o nível de habilidade/traço latente/proficiência.

O parâmetro  $b$  é capaz de informar o grau de dificuldade da questão, enquanto o parâmetro  $a$  é utilizado para identificar o poder que a questão tem em discriminar a habilidade dos estudantes. Já o parâmetro  $c$  é utilizado para informar à chance que um indivíduo tem em acertar a questão ao acaso (Andrade e Justino 2007).

A Teoria da Resposta ao Item, além de fornecer o gráfico de *Curva Característica do Item – CCI*, também disponibiliza o gráfico da *Curva de Informação de um Item – CII*, a qual representa quanto um item possui de informação em toda a escala e está, segundo (Baker 2001 apud Tezza et al. 2009), associada à precisão do instrumento. A *CII* é representada pela equação (2).

$$I_i(\theta) = \frac{[P_i'(\theta)]^2}{P_i(\theta)Q_i(\theta)}$$

onde:  $I_i(\theta)$  é a informação fornecida pelo item  $i$  ao nível da aptidão  $\theta$ ;

$P_i(\theta)$  é o modelo de probabilidade;

$P_i'(\theta)$  é a derivada de  $P_i(\theta)$  em relação a  $\theta$  e  $Q_i(\theta) = 1 - P_i(\theta)$ .

Segundo (Pasquali e Primi 2003), a Teoria da Resposta ao Item (TRI) trouxe grandes avanços complementares à Teoria Clássica de Medidas, são eles:

a) o cálculo do nível de aptidão do sujeito independe da amostra de itens utilizados: diz-se que a habilidade do sujeito é independente do teste (*not test-dependent*). Na Clássica, o escore do sujeito dependia e variava segundo o teste aplicado fosse mais fácil ou mais difícil, ou produzisse maiores ou menores erros. Assim, tais escores não eram comparáveis e, mesmo aplicando ajustes, os escores ainda continuavam não comparáveis, sobretudo porque os testes produziam diferenças nas variâncias de erros de medida. No caso da TRI, não importa que itens ou conjunto de itens que você utilize, obviamente que estejam medindo o mesmo traço latente, irão produzir o mesmo nível de aptidão do sujeito, dentro, é óbvio, dos sempre presentes erros de medida em qualquer ramo da ciência;

b) o cálculo dos parâmetros dos itens (dificuldade e discriminação) independe da amostra de sujeitos utilizada: diz-se que os parâmetros são independentes dos sujeitos (*not group-dependent*). Na clássica, os parâmetros dependiam muito dos sujeitos amostrados possuem maior ou menor aptidão;

c) a TRI permite emparelhar itens com a aptidão do sujeito. Isto quer dizer que se avalia a aptidão de um sujeito, utilizando itens com dificuldade tal que se situam em torno do tamanho da aptidão do sujeito, sendo, assim, possível utilizar itens mais fáceis para sujeitos com habilidades inferiores e itens mais difíceis para sujeitos mais aptos, produzindo escores comparáveis em ambos os casos. Na clássica sempre era aplicado o mesmo teste, hermeticamente fechado, para todos os sujeitos, de sorte que, se o teste fosse fácil, avaliaria bem sujeitos de aptidão menor e mal sujeitos de aptidão superior e, se o teste fosse difícil, faria o contrário. Esta história ficará mais clara ao falarmos de testes adaptativos e montagem de testes otimizados;

d) a TRI constitui um modelo que não precisa fazer suposições que aparentam serem improváveis, tais como os erros de medida serem iguais para todos os testados;

e) a TRI não necessita trabalhar com testes estritamente paralelos como exige a clássica.

Diante do contexto apresentado anteriormente, a Teoria da Resposta ao Item representa um avanço analítico para todas as áreas do conhecimento, incluindo a educação, onde algumas destas vantagens são fundamentais para facilitar a

análise do nível de conhecimento dos discentes. Entre estas vantagens, destacam-se: (a) poder de posicionar indivíduos (alunos) de diferentes grupos em uma escala comum, mesmo que estes tenham respondido a itens diferentes, (b) permitir uma mais precisa avaliação das propriedades dos itens e seus resultados e, conseqüentemente, permitir maior precisão na aplicação de técnicas estatísticas, (c) possibilidade de desenvolvimento de indicadores mais eficientes para avaliar diferenças individuais de sistemas ou indivíduos, (d) maiores robustez dos resultados.

### **3.4 Considerações Finais**

Este capítulo apresentou as duas teorias de medidas usadas na avaliação da educação básica utilizadas aqui no Brasil. Assim, em primeiro lugar, foram discutidos conceitos relacionados à Teoria Clássica de Medidas (TCM). Em seguida, a Teoria da Resposta ao Item (TRI) foi analisada, já que se constitui em uma forma importante de se promover uma análise detalhada do nível de conhecimento dos alunos.

Assim, de uma forma geral, a TRI é uma alternativa muito mais elaborada que a TCM para construção de instrumentos, principalmente quando se necessitam medidas consistentes em toda a escala, e em contextos em que se decisões de grande impacto são tomadas com base na pontuação de um indivíduo ou sistema.

O próximo capítulo tratará dos agentes de software e sobre os sistemas multiagentes, assuntos essenciais para o progresso deste trabalho proposto aqui.

## 4 AGENTES E SISTEMAS MULTIAGENTES

### 4.1 Considerações Iniciais

O paradigma baseado em agentes, surgido nos últimos anos, apresenta-se como uma interessante forma de se lidar com as abstrações do mundo real e com a representação do conceito de organização, visto que ele permite um melhor entendimento e uma maior comunicabilidade dos requisitos de usuários, dos artefatos de projeto, da arquitetura do sistema e da implementação final (Lindoso 2006).

Segundo (Silva 2006), não há um conceito unânime e uniforme para o termo “agente”. O que existe na realidade é uma convergência para características fundamentais que uma “peça de software” deve possuir para ser considerado um agente.

Um agente é uma entidade de software que funciona de forma contínua e autônoma em um ambiente. Ele também deve ser capaz de perceber e atuar no seu ambiente, de forma flexível e inteligente, sem requerer intervenção ou orientação humana constantes. Idealmente, um agente deve aprender através da experiência, comunicar-se e cooperar com outro agente que porventura co-exista no mesmo ambiente. Ainda, um agente pode mover-se de um local para outro a fim de satisfazer os seus objetivos, este tipo de agente é conhecido como “Agentes Móveis” (Bradshaw 1997 apud Silva 2006).

Os Sistemas Multiagentes vieram com o objetivo de fornecer soluções para abordagens com uma crescente complexidade nos sistemas de computação que geralmente devem operar em ambientes abertos e que modifica-se dinamicamente. Conforme (Silva 2006), a abordagem Multiagentes tem sido apontada por muitos pesquisadores como adequada ao desenvolvimento de sistemas de software complexos devido ao seu alto nível de abstração, o que torna mais conveniente a análise e o projeto de tais sistemas.

Uma das vantagens dos sistemas multiagentes, segundo (Brenner et al. 1998 apud Silva 2006), é o fato deles permitirem a integração de agentes já existentes. Isso faz com que a solução do problema não demande o desenvolvimento de novos e especializados agentes, ou seja, o conhecimento dos agentes existentes pode ser utilizado para resolver determinado problema.

O objetivo deste capítulo é apresentar uma conceitualização ampla sobre agentes de software e sistemas multiagentes, observando conceitos primordiais como algumas metodologias e ferramentas usadas na implementação desses agentes. Além desta seção com as considerações iniciais, temos a seção 4.2 que relata sobre o agente de software. A seção 4.3 mostra a categorias de agentes. Já a seção 4.4 apresenta os tipos de comunicação entre os agentes. A seção 4.5 mostra as linguagens de comunicação entre agentes. A seção 4.6 traz explicações sobre sistemas multiagentes. A seção 4.7 apresenta as possíveis metodologias e ferramentas utilizadas na construção dos sistemas multiagentes. Já na seção 4.8 software que foram desenvolvidos usando agentes para a área educacional. E por último, a seção 4.9 traz as considerações finais do capítulo.

## 4.2 Agente de Software

A abordagem de agentes de software pode ser vista como complementar ao paradigma de objetos na implementação de sistemas com um nível de complexidade elevado. Esta nova tecnologia propõe um nível de abstração mais alto que aquele proposto pela tecnologia de objetos uma vez que o comportamento dos agentes situa-se mais próximo do comportamento de seres humanos cujo trabalho em geral é substituído ou suportado por sistemas de software.

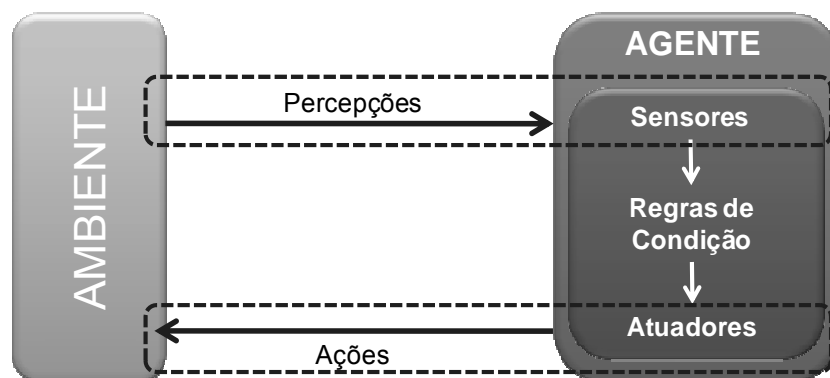


Figura 2 – Agente Genérico.

Na Figura 2 é exibido um agente de software genérico com sua estrutura básica que tem uma forma bem simples, possuindo uma memória interna que irá atualizar-se com a chegada de novas percepções, que são capturadas através de sensores. Assim, essa memória é utilizada nos procedimentos de tomada de



decisão, os quais irão gerar ações que serão executadas através de executores (Lindoso 2006).

De acordo com (Wooldridge et al. 1995), os agentes são artefatos de software tradicionalmente caracterizados por apresentarem traços bastantes peculiares, como os seguintes:

- **Autonomia:** funcionam sem intervenção humana e executam ações por iniciativa própria segundo o conhecimento que possuem do ambiente e sua racionalidade;
- **Sociabilidade:** implica no estabelecimento de canais de comunicação de uns com os outros, através de sofisticados protocolos de interação, a fim de cooperar, competir, negociar, dentre outras;
- **Percepção:** eles percebem o que acontece no ambiente físico ou computacional a sua volta e reagem a isso atuando sobre ele, podendo inclusive provocar-lhe alterações;
- **Pró-atividade:** seus comportamentos são direcionados ao alcance de objetivos, além da mera reatividade a estímulos do ambiente;
- **Aprendizagem:** são capazes de aprender a partir das ações realizadas, considerando sucessos e fracassos, de forma a melhorar seu desempenho.

A Figura 3 a seguir, ilustra estas características peculiares do agente exposto no parágrafo anterior.



Figura 3 – Peculiaridades de um Agente de Software.

Segundo (Pont et al. 1996 apud Lindoso 2006), os agentes são similares a objetos, mas algumas diferenças podem ser observadas, são elas:

- **Objetos:** são essencialmente passivos, reagindo a estímulos externos, mas sem apresentar comportamento voltado para objetivos como os agentes possuem;

- **Agentes:** tipicamente utilizam uma linguagem comum para a troca de mensagens, diferente das mensagens passadas para objetos que dependem da classe da qual eles são instâncias.

Portanto, segundo (Russell et al. 1995), um agente caracteriza-se também como uma entidade que percebe seu ambiente através de sensores e atua neste ambiente através de executores. Por exemplo, nos agentes humanos, os olhos e os ouvidos são sensores e as mãos e a boca são executores.

### 4.3 Categoria de Agentes

Nos últimos anos, o paradigma de agentes tem aumentado significativamente sua popularidade. Segundo (Hayes 1999 apud Franceschi et al. 2000) deve-se as características como flexibilidade, modularidade e aplicabilidade na solução de uma variedade de problemas.

Segundo (Nwana 1996 apud Silva 2006), é possível fazer uma classificação nas categorias dos agentes da seguinte maneira:

- **Agentes Móveis** – são agentes que tem a mobilidade como característica principal, isto é, uma capacidade de mover-se seja por uma rede interna local (*intranet*) ou até mesmo pela Internet, transportando-se pelas plataformas levando dados e códigos;

- **Agentes Situados ou Estacionários** – são agentes opostos aos móveis, ou seja, são fixos em um mesmo ambiente e/ou plataforma. Não se movimentam em uma rede e muito menos na Internet;

- **Agentes Competitivos** – são agentes que “competem” entre si para a realização de seus objetivos ou tarefas, isto é, não há colaboração entre os agentes;

- **Agentes Coordenados ou Colaborativos** – estes agentes têm a finalidade de alcançar um objetivo maior, realizando tarefas específicas, porém, coordenando-as entre si de forma que suas atividades se complementem;

- **Agentes Reativos** – são agentes que reagem a estímulos sem ter memória do eu já foi realizado no passado e nem previsão da ação a ser tomada no

futuro. Não tem representação do seu ambiente ou de outros agentes e são incapazes de prever e antecipar ações;

- **Agentes Cognitivos** – são agentes opostos aos reativos, pois, podem raciocinar sobre as ações tomadas no passado e planejar ações a serem tomadas no futuro. Portanto, os agentes cognitivos são capazes de solucionar problemas através dele mesmos. Eles possuem uma grande flexibilidade na forma de expressar seu comportamento.

#### 4.4 Comunicação entre Agentes

De acordo com (Silva 2006), a comunicação é o processo pelo qual são trocadas as informações entre os agentes e segundo (Brenner et al. 1998 apud Silva 2006), os métodos de comunicação podem ser divididos em dois grupos: **Quadro de Avisos e Troca de Mensagens**.

##### 4.4.1 Quadro de Avisos

Nesse método os agentes podem trocar informações, conhecimentos e dados através de um local compartilhado por todos. Este meio de comunicação também é conhecido como **quadro negro** (Almeida 2007).

Neste método de acordo com (Silva 2006), uma ação de comunicação é iniciada quando um agente escreve alguma coisa no quadro negro. Assim, os agentes podem acessar o quadro negro a qualquer momento e verificar se alguma informação foi modificada ou adicionada (*Vide* Figura 4).

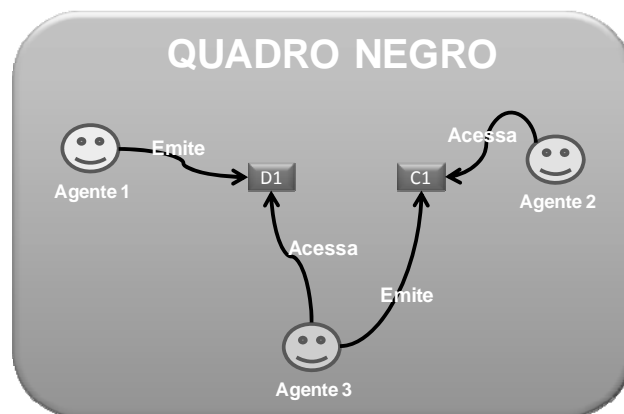


Figura 4 – Comunicação “Quadro Negro” de um Agente de Software.

(Brenner, 1998) afirma que “Quando um sistema possui um grande número de agentes, a quantidade de informações escritas no quadro negro cresce exponencialmente e, portanto, a busca por informações relevantes se torna cada vez mais pesada”.

Para resolver isso, dividem-se as áreas dos quadros negros em subáreas, cada qual alocada a um agente ou a um grupo deles. Assim, um agente precisará monitorar apenas a região a ele alocada (Almeida 2007).

Portanto, (Brenner 1998) afirma que uma sobrecarga do sistema pode ser causada pela leitura de informações no quadro de avisos. Com isso, os sistemas baseados em troca de mensagens, que será visto na próxima seção, vêm sendo mais utilizados.

#### 4.4.2 Troca de Mensagens

Para (Silva 2006), a comunicação usando mensagens forma uma base flexível para a implementação de estratégias de coordenação complexas.

Na troca de mensagens as informações são transmitidas diretamente de um agente para outro, no caso há um repasse entre um agente emissor para um receptor. Assim, um agente, emissor, envia uma mensagem direta a outro agente, o receptor.

Na Figura 5 a seguir, é mostrada a troca de mensagens entre dois agentes.



Figura 5 – Interação na Troca de Mensagens.

Conforme (Finin et al. 1997 apud Silva 2006), a troca de mensagens é classificada em:

- **Ponto-a-ponto** – as mensagens são enviadas para um agente específico que ser conhecido pelo emissor. Uma grande vantagem da abordagem ponto-a-ponto é que o agente emissor sempre sabe para onde uma mensagem está sendo enviada, o que permite a fácil introdução de controles de segurança;

- **Broadcast** – a mensagem é enviada para todos os agentes da sociedade. Na abordagem *broadcast*, um agente pode ser substituído por outro equivalente e o processamento do sistema será inalterado. A passagem de mensagem por *broadcast* não é segura, já que qualquer agente pode examinar o conteúdo de qualquer mensagem;

- **Multicast** – a sociedade de agentes é dividida em grupos. Desta forma, as mensagens podem ser enviadas somente aos agentes de determinados grupos.

Podemos concluir que a maioria dos Sistemas Multiagentes utiliza os sistemas baseados em troca de mensagens, por proporcionarem mais flexibilidade e dinamismo do que a arquitetura do quadro de avisos apresentada anteriormente. Entretanto, para isso acontecer faz-se necessário a definição de um protocolo de comunicação que especifique o exato processo de comunicação, um formato para as mensagens e a linguagem escolhida. Neste aspecto, as linguagens de comunicação de agentes são um recurso poderoso e importante para o processo de comunicação entre agentes. Estas linguagens serão apresentadas na seção 4.5.

#### 4.5 Linguagens de Comunicação entre Agentes

A tecnologia de agentes de software despertou o interesse como ferramenta para criação de um novo modelo para sistemas de software complexos. No projeto de agentes, misturam-se muitas das propriedades tradicionais de inteligência artificial (como “raciocínio” no nível de conhecimento, flexibilidade, etc.) com as experiências obtidas nas áreas de sistemas distribuídos, educação, teorias sobre negociação, dentre outras (Fialho 2003).

Para que haja uma cooperação entre os agentes, é necessária uma comunicação entre eles. De acordo com (Fialho 2003), uma coleção de agentes trabalhando juntos em cooperação pode ser vista como uma pequena sociedade, e o funcionamento de qualquer sociedade coerente precisa de uma linguagem comum e um meio de comunicação, portanto, para que os atos de comunicação e a cooperação entre agentes sejam possíveis, precisa-se de uma Linguagem de Comunicação entre Agentes (*Agent Communication Language – ACL*). Assim, (Silva 2006) relata em seu trabalho que uma *ACL* determina um meio através do qual uma

atitude (por exemplo, afirmação, requisição e consulta) a respeito do conteúdo da troca de conhecimentos é comunicada.

#### 4.5.1 KQML

A KQML (*Knowledge Query and Manipulation Language*) é uma linguagem e um protocolo para troca de informações e conhecimento entre agentes (Darpa 2006 apud Silva 2006).

Segundo (Fialho 2003), a KQML é uma linguagem versátil de propósito geral que suporta a comunicação entre vários agentes com um conjunto de primitivas reservadas chamadas *performative*. Ela adotou o termo *performative* para referir-se a uma declaração feita pelo usuário emissor e que é executada pelo agente receptor, simplesmente porque o emissor declara ou afirma.

Esta linguagem possui uma característica interessante, sendo ela responsável por oferecer uma forma de acesso à informação mesmo para programas que não sejam com agentes (Fialho 2003).

De acordo com (Silva 2006), a Figura 6 a seguir, apresenta um trecho de uma mensagem KQML, onde um agente comprador (:*sender*) perguntando (:*content*) a um agente vendedor (:*receiver*) o preço de uma mercadoria. O agente vendedor dá como resposta (:*reply-with*) o preço da mercadoria.

```
(ask-one
  :sender agente-comprador
  :receiver agente-vendedor
  :content (PRECO MERCADORIA ?preco)
  :reply-with preço-mercadoria
  :linguagem LPROLOG
  :ontology compra-e-venda)
```

**Figura 6 – Mensagem KQML (ask-one)**

**Fonte: Silva 2006.**

#### 4.5.2 FIPA-ACL

Para (Silva 2006), a FIPA-ACL é uma linguagem de comunicação entre agentes que vem ganhando espaço e, ela foi apresentada inicialmente como uma alternativa bem fundamentada para a KQML que com o passar do tempo, surgiu dois padrões fundamentais para uma implementação disponível e fortemente competitiva a KQML, são eles: padrão FIPA 2000 e a plataforma FIPA-OS.

A estrutura das mensagens FIPA-ACL é muito parecida com a estrutura das mensagens KQML. Sendo assim, cada mensagem é composta por um ato comunicativo (*performative*) e um conjunto de parâmetros. O campo que representa o ato comunicativo é o único que é obrigatório, pois, este campo indica qual ato está sendo transportado pela mensagem. Só que, a grande maioria das mensagens contém um emissor, um receptor e um campo de conteúdo. Os parâmetros podem ser alocados em qualquer posição dentro da mensagem. Além disso, as implementações são livres para definirem novos parâmetros, desde que, seja utilizado o prefixo “x-” (Silva 2006).

Nas Figuras 7 e 8 são mostrados exemplos de mensagens FIPA-ACL. Temos na Figura 7, o agente *i* perguntando se o agente *j* está registrado no domínio *d1*. já na Figura 8 o agente *j* responde que não.

```
(query-if
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :content "((registered (server d1) (agent j)))"
  :reply-with r09
)
```

**Figura 7 – Mensagem FIPA-ACL (query-if)**

**Fonte: Silva 2006.**

```
(inform
  :sender (agent-identifier :name j)
  :receiver (set (agent-identifier :name i))
  :content "((not (registered (server d1) (agent j))))"
  :in-reply-to r09
)
```

**Figura 8 – Mensagem FIPA-ACL (inform)**

**Fonte: Silva 2006.**

#### **4.6 Sistemas Multiagentes**

Sistemas multiagentes são aqueles em que vários agentes trabalham ou se interagem em grupo, de forma a realizar um determinado conjunto de tarefas ou objetivos. Os agentes, nos sistemas multiagentes, podem ser heterogêneos ou homogêneos, colaborativos ou competitivos, ou seja, a definição dos tipos de agentes depende da finalidade da aplicação que o sistema está inserido (Silva 2003 apud Tives 2007).

Para (Hübner e Schman 2003 apud Tives 2007), os sistemas multiagentes homogêneos são compostos por um único tipo de agente, já nos sistemas multiagentes heterogêneos são vários tipos de agentes existentes no mesmo sistema. Os sistemas multiagentes colaborativos são grupos de agentes que cooperam na solução de problemas que estão além da capacidade da resolução individual. Os agentes realizam tarefas específicas, porém, coordenando-as entre si de forma que suas atividades se completem. Os sistemas multiagentes competitivos são sistemas onde os agentes não têm um objetivo comum, cada um busca trabalhar apenas para o seu próprio bem, negociando com outros agentes quando necessário. Com isso verifica-se que esses tipos de agentes competem entre si para a realização de seus objetivos ou tarefas, não havendo colaboração entre os agentes.

Para a (OMG 2000), os sistemas multiagentes possuem algumas particularidades, são elas:



- Não se deve criar agentes que façam tudo sozinhos, pois dessa forma, tais agentes representarão problemas de performance, confiabilidade, manutenção. Dividindo as funcionalidades entre vários agentes, obtêm-se maiores modularidade, flexibilidade, extensibilidade e manutenção facilitada.

- O conhecimento que está espalhado em vários agentes pode ser integrado para se obter uma visão geral, quando necessário.

- Aplicações que requerem processamento distribuído são mais bem suportados por Sistemas Multiagentes. Para isso, os agentes podem ser projetados como pequenos componentes autônomos que trabalham em paralelo.

Para os diferentes autores, os sistemas multiagentes possuem diversas características. Segundo (Glutz 2003 apud Tives 2007), as três principais características dos sistemas multiagentes são:

- **Comunicação:** para negociar e comunicar intenções e planos. Tem como principal objetivo a troca de conhecimentos com outros agentes e a coordenação de atividade entre si, resultando em um sistema coerente.

- **Coordenação:** para a execução de atividades conjuntas. O compartilhamento de conhecimento entre os agentes a fim de tornar as ações individuais de cada agente coordenadas para se atingir o objetivo final do sistema multiagentes.

- **Interação:** define as formas de interação entre os agentes, ou seja, a comunicação e comportamento individual, a fim de criar um conjunto de agentes que trabalham juntos de forma competente.

Diante das explicações apresentadas anteriormente sobre os sistemas multiagentes, algumas das principais áreas de atuação dos sistemas multiagentes são: controle de tráfego aéreo, indústria, educação, gerência de negócios, interação humano-computador, ambientes de aprendizagem, entretenimento, aplicações para a internet, dentre outras.

Na seção 4.7 a seguir, serão exibidas as metodologias e ferramentas usadas na construção dos sistemas multiagentes e posteriormente, tabelas comparando as metodologias e as ferramentas estudadas.

## 4.7 Metodologias e Ferramentas para o Desenvolvimento de Sistemas Multiagentes

O advento dos sistemas multiagentes trouxe uma nova forma de se abordar à construção de aplicações distribuídas, inteligentes e robustas, que provêem soluções para problemas em ambientes complexos (Jennings 2000 apud Lindoso 2006, mas que suscitam problemas relativos ao comportamento individual dos agentes (Dileo et al. 2002 apud Lindoso 2006).

Na abordagem dos agentes, intenta-se fugir da tradição de se elaborar técnicas de desenvolvimento dirigidas pela tecnologia de implementação, como já ocorreu ditado pelas linguagens de programação estruturadas e orientadas a objetos, para baseá-las nos requisitos, no sentido que os conceitos usados na análise de requisitos são empregados adiante no projeto e na implementação (Mylopoulos e Castro 2000).

O projeto de sistemas baseados em agentes se difere do convencional porque o conceito de agente envolve noções como a de autonomia, cooperação etc. Assim, para uma técnica ser apropriada a esse desenvolvimento, ela deve aproveitar as lições adquiridas das abordagens clássicas, mas precisa ir além para abranger também aquelas peculiaridades, bem como questões relativas a ontologias, comunicação, mobilidade, dentre outras (Cossentino et al. 2002).

Diante dessa realidade apresentada, faz-se necessária uma explicação detalhada de algumas metodologias como a PASSI, a Gaia, Tropos e a AgentTool e respectivas ferramentas disponíveis para o desenvolvimento baseado em agentes.

### 4.7.1 PASSI

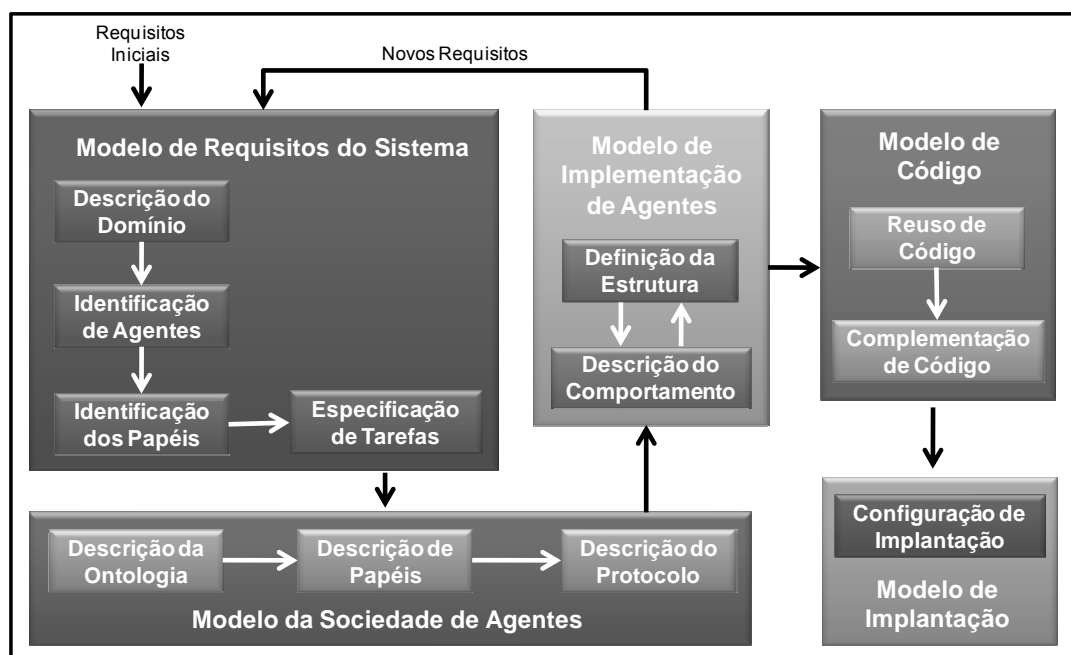
A PASSI (*Process for Agents Societies Specification and Implementation*) é uma metodologia para o projeto e desenvolvimento de sociedades multiagentes, abrangendo desde a especificação de requisitos até a codificação, através da integração de modelos e conceitos tanto da Engenharia de Software Orientada a Objetos quanto da Inteligência Artificial por meio da notação UML – *Unified Modeling Language* (Cossentino et al. 2002).

Segundo (Lindoso 2006), na PASSI, um agente é uma unidade de software que representa uma entidade autônoma capaz de perseguir seus objetivos

através de decisões, ações e relacionamentos sociais. Isso vale tanto no nível abstrato – quando a modelagem é distante da implementação, sendo orientado ao problema – quanto no concreto – quando os modelos são próximos do código, sendo dirigido à solução.

Assim, um agente é a implementação de software de uma entidade autônoma capaz de perseguir objetivos através de decisões, ações e relações sociais autônomas, podendo se ocupar de vários papéis funcionais durante suas interações com outros agentes com o intuito de alcançar objetivos, sendo um papel uma coleção de tarefas desempenhadas por um agente para atingir um sub-objetivo. Uma tarefa, por sua vez, é uma unidade motivada de comportamento individual ou interativo (Lindoso 2006).

A Figura 9 a seguir, exemplifica a metodologia PASSI com suas respectivas cinco fases, onde cada uma resulta de um diferente modelo, gerando um total de doze passos, que parte de uma representação informal de requisitos, seguida pela modelagem da sociedade de agentes e da solução arquitetural, as quais precedem a produção ou reuso do código necessário para a construção dos agentes, a serem configurados para uso na etapa final de implantação (Cossentino et al. 2002).



**Figura 9 – Fases e Passos da Metodologia PASSI.**

Fonte: Adaptado de Lindoso 2006.

A PASSI é uma metodologia adequada para o desenvolvimento de sistemas multiagentes, onde (Lindoso 2006) relata em seu trabalho, que esta relevância está associada a fatores estratégicos, tais como: o uso de uma linguagem de projeto padronizada e bem conhecida; o suporte de uma ferramenta CASE específica para facilitar o trabalho projetista (como a PTK, descrita a seguir); e atenção à geração automática de grande parte do código.

A ferramenta PTK (*PASSI Tool Kit*) é uma ferramenta especialmente concebida para prestar total suporte à metodologia PASSI, sendo composta conceitualmente por duas partes completamente integradas: um *plugin* para o Rational Rose destinado à construção dos modelos; e uma ferramenta Java, que permite o reuso de padrões tanto na plataforma JADE quanto no FIPA-OS através da representação do código dos agentes em uma meta-linguagem baseada em XML (Cossentino et al. 2002).

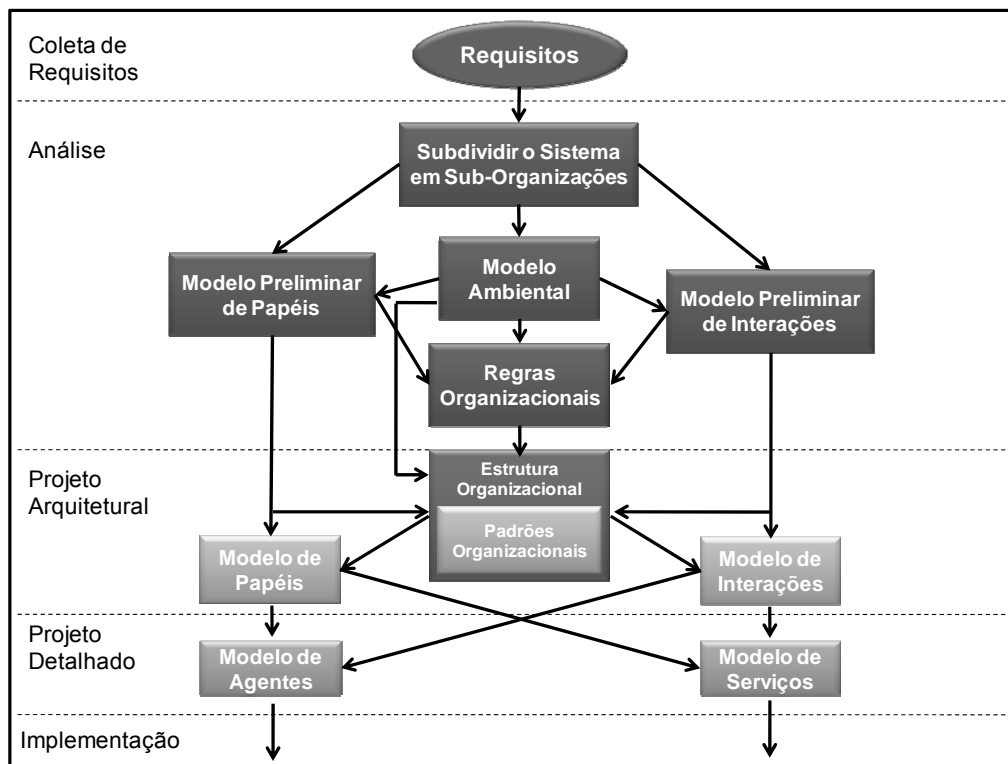
Portanto, a ferramenta PASSI Tool Kit visa dar suporte à PASSI, permitindo que os artefatos de software definidos em cada fase da metodologia possam ser construídos integrados a uma reconhecida ferramenta de modelagem, Rational Rose.

#### **4.7.2 Gaia**

A Gaia é uma metodologia geral para a análise e o projeto orientados a agentes que suporta tanto o nível micro do desenvolvimento, ou seja, a estrutura dos agentes, quanto o macro, isto é, a sociedade multiagentes e a estrutura da organização. A sua motivação é tentar representar com sucesso as características naturais dos agentes, tais como a autonomia e a capacidade de resolver problemas, além das formas como eles realizam interações e criam organizações. Assim, tem-se como resultado um projeto elaborado a partir dos requisitos do sistema e pronto para ser implementado (Wooldridge et al. 2000 apud Lindoso 2006).

A metodologia Gaia é interessante no desenvolvimento de sistemas multiagentes em domínios fechados, mas ela apresenta limitações que a torna pouco útil em ambientes abertos e inesperados como, por exemplo, o domínio das aplicações para Internet. Então, (Lindoso 2006) relata em seu trabalho que, para superar tais restrições, (Zambonelli 2003) propôs uma extensão da metodologia tomando por base a especificação original da Gaia. A Figura 10 ilustra essa versão

estendida e as novas abstrações organizacionais que são exploradas para se ampliar de maneira significativa a sua aplicabilidade.



**Figura 10 – Fases e Passos da Metodologia Gaia.**

Fonte: Adaptada de Lindoso 2006.

Segundo (Lindoso 2006), na fase de Análise, o objetivo da Gaia é compreender as funcionalidades do sistema, concebido como uma organização composta de um conjunto de papéis que interagem entre si, e representá-las em modelos de papéis e de interações. Ela é composta por dois passos: Modelagem de Papéis e Modelagem de Interações.

Em primeiro lugar, são identificados os papéis que participarão do sistema, os quais possuem quatro atributos, que são: responsabilidades, que podem obrigar o papel a realizar algo de bom ou proibi-lo fazer algo de ruim para o escopo do sistema; permissões, que representam o que o papel pode executar e, em particular, que dados pode acessar; atividades, que são tarefas que o papel desempenha sem precisar interagir com outros papéis; e protocolos, que são padrões específicos de interação. A Gaia define formalmente operadores e modelos para representar papéis e seus atributos, assim como esquemas para as interações (Lindoso 2006).

Já durante o projeto, são realizados três passos: Modelagem de Agentes; Modelagem de Serviços e Modelagem de Comunicação. No primeiro, ocorre o mapeamento dos papéis em tipos de agentes, para então ser criado o número adequado de instâncias de cada tipo de agente. No segundo, busca-se determinar o modelo de serviços requerido para satisfazer um papel, seja ele desempenhado por um ou vários agentes. No terceiro e último passo, os agentes são tornados familiares entre si através da criação de um modelo de comunicação (Lindoso 2006).

Conforme foi apresentado anteriormente sobre a metodologia Gaia, ela passou a ser mais pautada na noção de organização, considerando as abstrações a ela inerentes como sendo fundamentais para se analisar e projetar sistemas multiagentes. Isso permite definir uma seqüência ordenada de passos, compondo as fases da metodologia, os quais produzem um conjunto de modelos, os relacionados entre eles, e regras sobre como e quando utilizá-los para o desenvolvimento de um sistema multiagentes (Zambonelli et al. 2003).

#### **4.7.3 Tropos**

A Tropos é uma metodologia de desenvolvimento de software orientado a agentes que se baseia em duas características-chave: as noções de agente, objetivo, plano, bem como vários outros conceitos em nível de conhecimento. Tais são primitivas fundamentais e uniformemente usadas durante todo o processo; um papel crucial é atribuído à análise e especificação de requisitos quando o futuro sistema é analisado em relação ao seu ambiente alvo (Giunchiglia et al. 2002).

De acordo com (Lindoso 2006), a metodologia se inicia com um modelo do ambiente em que o sistema irá operar que é descrito em termos de atores, de seus objetivos e suas interdependências. Através de refinamentos incrementais, o modelo é estendido para incluir o próprio sistema e seus subsistemas, também representados como atores aos quais objetivos a alcançar são delegados, planos para executar e recursos a fornecer.

A linguagem da Tropos é baseada em um pequeno conjunto de conceitos e provê ferramentas e técnicas para construir os modelos que representam atores (agentes, posições e papéis), seus objetivos e suas interdependências. Tais modelos são usados para capturar as intenções dos interessados (usuários, proprietários, gerentes etc.), as responsabilidades do novo sistema com relação a

esses interessados, bem como sua arquitetura e os detalhes de seu projeto. Esses modelos oferecem uma interface comum para varias fases do desenvolvimento, desde o levantamento de requisitos até a implementação, podendo também ser usados como parte da documentação durante a operação e manutenção do software (Lindoso 2006).

As cinco principais fases do desenvolvimento de sistemas multiagentes nos termos da metodologia Tropos são descritas a seguir (Bresciani et al. 2004 apud Lindoso 2006):

- **Requerimentos Iniciais:** é por onde se começa a reunir o conjunto de requisitos funcionais e não-funcionais do sistema em desenvolvimento. Os principais interessados do sistema são identificados, sendo representados como atores, bem como se estabelecem seus respectivos objetivos;

- **Requerimentos Finais:** o sistema propriamente dito é introduzido também como um ator, sendo devidamente relacionado com os atores que representam os interessados em termos de dependências, que indicam as obrigações do sistema para com o seu ambiente e, em contrapartida, o que aquele pode esperar deste;

- **Projeto Arquitetural:** alguns outros atores do sistema são acrescentados e a eles são atribuídos sub-objetivos e sub-tarefas derivados dos objetivos e das tarefas designadas para o sistema;

- **Projeto Detalhado:** atores do sistema são definidos em mais detalhes, sendo especificadas as comunicações e protocolos de coordenação estabelecidos entre eles;

- **Implementação:** a especificação produzida nas fases precedentes é, então, transformada em um esqueleto para a implementação, o que é feito através do mapeamento dos elementos da metodologia aos conceitos de uma plataforma de programação de agentes escolhida.

Assim, conforme (Mylopoulos et al. 2000 apud Lindoso 2006), a Tropos confere fundamental importância aos requisitos porque é durante seu levantamento que as principais considerações técnicas têm que ser confrontadas com outras sociais e pessoais, sendo nessa fase em que são cometidos os piores e mais custosos erros no desenvolvimento de software. Pois é nela que é respondida a recorrente pergunta: “o que se pretende que o sistema faça?”.

#### 4.7.4 AgentTool

O AgentTool foi desenvolvido pelo *Air Force Institute of Technology*, cujo objetivo dele é permitir a projetistas de sistemas baseados em agentes especificar formalmente a estrutura e o comportamento requerido para um sistemas multiagentes e semi-automaticamente sintetizar sistemas multiagentes. O comportamento de alto nível do sistema é definido graficamente usando a Metodologia de Engenharia de Sistemas Multiagentes MaSE (*Multiagent Systems Engineering Methodology*) (Deloach 2001).

A Figura 11 exhibe a interface com o usuário do AgentTool, onde se pode acessar através de menus tabulares os passos para a análise e projeto de um sistema.

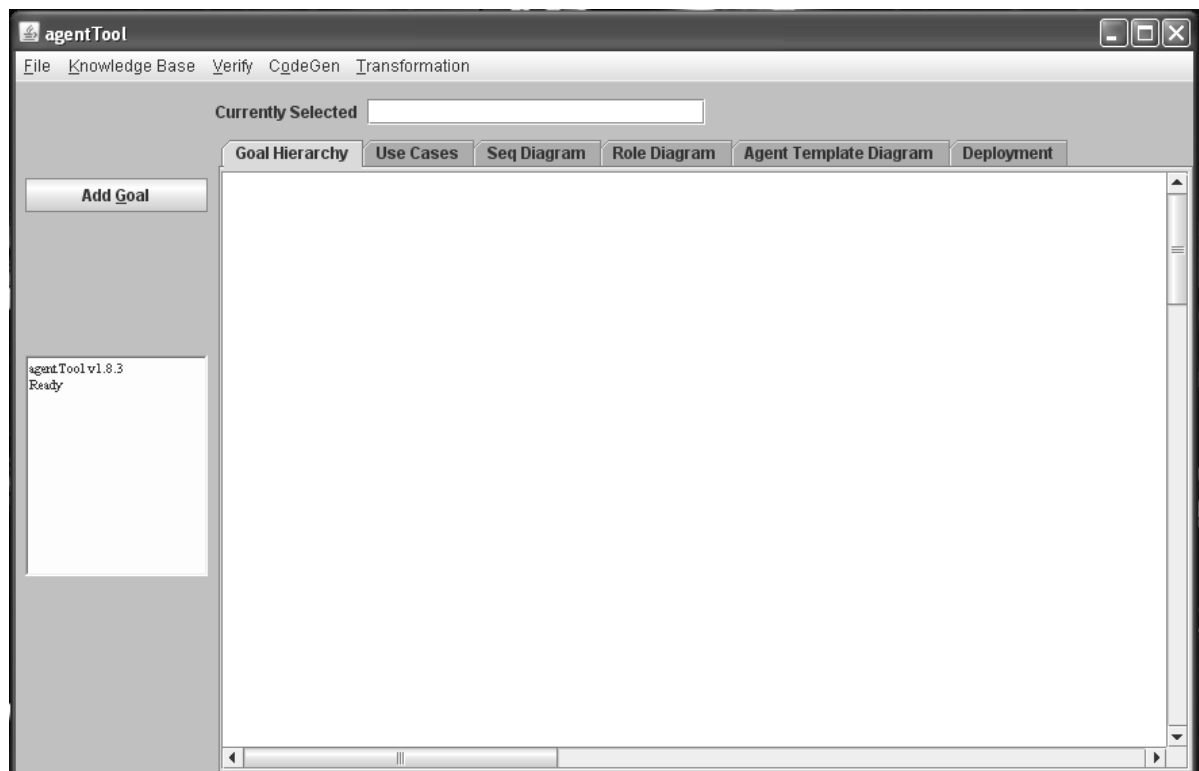


Figura 11 – Interface da Ferramenta AgentTool.

Segundo (Deloach 2001 apud Perotto 2002), dentre as ferramentas disponíveis atualmente para o desenvolvimento de Sistemas Multiagentes, a AgentTool é a que apresenta adequadamente um ciclo de vida completo e um meio complementar para análise, projeto, e desenvolvimento de Sistemas Multiagentes heterogêneos.



Portanto, A relação do usuário com a ferramenta AgentTool se dá através de menus suspensos que permitem acesso a diversas funções do sistema, incluindo uma verificação persistente da conversação da base de conhecimento e a geração do código.

#### 4.8 Software na Educação que usam Agentes de Software

Os software educacionais apoiados pela Inteligência Artificial (IA) tem a possibilidade de solucionar problemas complexos que antes não eram capazes de serem resolvidos. Esses sistemas do campo educacional, através de Tutores Inteligentes (Frezza et al. 2007) e de Agente Software, podem integrar um recurso computacional a uma proposta de ensino já solidificada e confirmada pela escola.

Assim, alguns software já desenvolvidos possuem como principal objetivo a integração de conceitos educacionais e fundamentos da Inteligência Artificial para melhorar a aprendizagem dos alunos e conseqüentemente melhorar a qualidade do ensino brasileiro. Podemos analisar como exemplos de software educacionais que usam recursos da IA os seguintes: o agente pedagógico **Fred** – responsável por acompanhar o percurso do aluno durante toda sua navegação no software, sugerindo atividades, questionando sua compreensão sobre os textos, chamando sua atenção e elogiando-o. Este agente está amparado por concepções pedagógicas (Inatista, Empirista e Interacionista) e busca respeitar o desenvolvimento individual, o processo cognitivo e o conhecimento que cada aluno traz em suas experiências. O software educacional desenvolvido é sobre a História dos Jesuítas no Rio Grande do Sul, tendo como público alvo crianças da quarta série do Ensino Fundamental (Mello e Frezza 2007).

Outro software educacional é um Sistema Tutor Inteligente (STI) baseado em estilos cognitivos de raciocínio, com a atuação de dois agentes pedagógicos que desempenham diferentes funções – o **Agente Tutor** (*Agente Dóris*) e o **Agente Companheiro** (*Agente Dimi*). O primeiro agente possui um módulo de tomada de decisão, que é responsável por analisar as características do aluno e relacioná-las com as táticas de ensino a serem utilizadas para a apresentação do material instrucional. Tem sua ativação através de regras de produção, na qual as condições são as características dos alunos e as ações são as táticas de ensino. Já o segundo agente tem a função de atuar no ambiente como um colega do aluno, emitindo

mensagens de desafios em relação ao conteúdo, por exemplo. Este STI utiliza como foco de estudo o domínio da Geografia (Frezza et al. 2007).

Por último, temos um sistema que utiliza **Agentes de Interface** para serem aplicados ao ensino de equações do primeiro grau. No sistema proposto, foi definida uma sociedade de com quatro agentes, são eles: **Pepe** (papagaio), **Merlin** (mago), **Jonas** (gênio da lâmpada) e **Bob** (robô), podendo ser selecionados de acordo com sua preferência e cada um possuindo funções específicas. Portanto, No sistema proposto, são usados personagens com as mais variadas ações no que se refere à comunicação com o usuário. Estes podem atuar dando as boas vindas ao usuário, apresentar dicas e diálogos pertinentes através de recursos escritos e orais. Estes agentes representam o componente interativo do Sistema Tutor Inteligente (Rabello et al. 2007).

#### **4.9 Considerações Finais**

Este capítulo apresentou os principais conceitos sobre agentes de software e sistemas multiagentes. Assim, no primeiro momento, foram discutidos conceitualizações sobre as categorias dos agentes. Em seguida, os possíveis mecanismos de comunicação entre agentes foram analisados, já que se estabelece em uma forma importante de se obter o funcionamento correto desses agentes. As linguagens de comunicação, sistemas multiagentes (SMA), metodologias e ferramentas utilizadas no desenvolvimento desses SMA, também foram detalhadas. Por fim, foram descritos alguns software que fazem uso de agentes na área da educação.

O próximo capítulo tratará da modelagem geral do software Virtual-TANEB, uma das contribuições do trabalho proposto.

## **5 MODELAGEM DO SOFTWARE VIRTUAL-TANEB**

### **5.1 Considerações Iniciais**

Para (Guedes 2008), a UML (*Unified Modeling Language – Linguagem de Modelagem Unificada*) tornou-se, nos últimos anos, a linguagem-padrão de modelagem adotada internacional pela indústria de Engenharia de Software. Assim, a UML oferece um grande número de diagramas que enfocam tanto características estruturais como comportamentais de um software. No entanto, no desenvolvimento de uma modelagem, não se faz necessário a utilização de todos os diagramas propostos pela linguagem, pois cada um deles possui uma função específica e, algumas vezes, alguns não são necessários em determinadas situações ou domínios.

Para demonstrar a funcionalidade do sistema Virtual-TANEB através dos diagramas UML, foi idealizada a presente modelagem, o qual consiste na construção dos diagramas de casos de usos e dos diagramas de seqüência para este software.

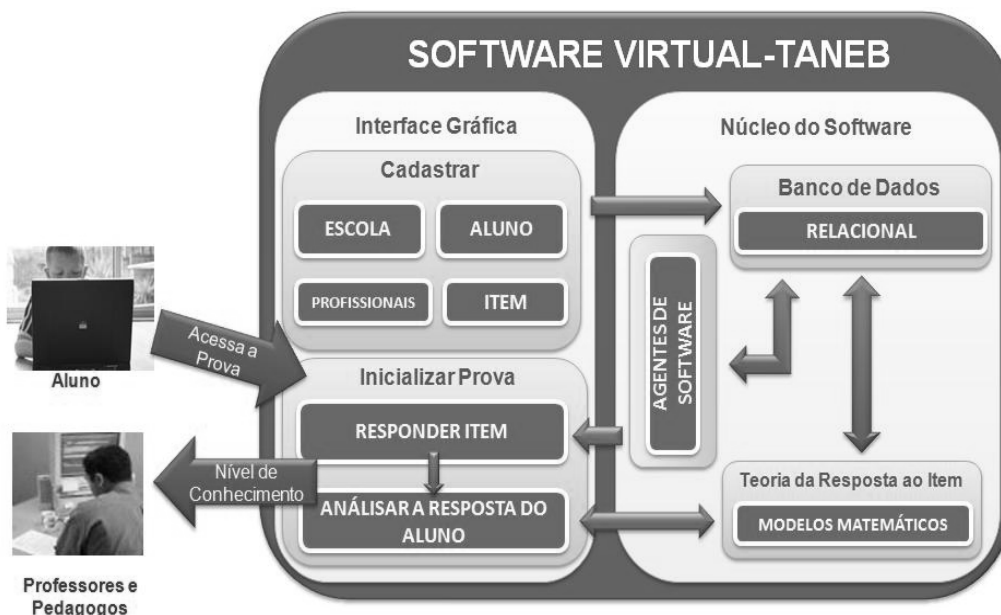
O objetivo deste capítulo é apresentar uma proposta de modelagem sobre o Virtual-TANEB, observando conceitos primordiais da UML e do Processo Unificado no desenvolvimento desta modelagem. Além desta seção com as considerações iniciais, temos a seção 5.2 que relata a arquitetura geral do Virtual-TANEB. A seção 5.3 mostra os diagramas de casos de usos e de seqüência usados para exibir algumas funcionalidades básicas do software aqui mostrado neste trabalho. Já a seção 5.4 apresenta as considerações finais do capítulo.

### **5.2 Arquitetura Geral do Virtual-TANEB**

Alguns software têm sido desenvolvidos com base nas Teorias Clássica de Medidas e de Resposta ao Item para aprimorar o processo de avaliação educacional. Desse modo, este capítulo traz a modelagem dos módulos do Virtual-TANEB (VT). Na fase inicial, ele proporcionará um suporte aos professores e pedagogos na avaliação do conhecimento de matemática dos alunos da quarta série do ensino fundamental, especificamente no domínio da geometria. Este software determinará o nível de desempenho dos alunos, a partir das questões respondidas corretamente, aplicando a Teoria da Resposta ao Item para avaliar as respostas.

A arquitetura do Virtual-TANEB, conforme mostra a Figura 12, é constituída pela Interface Gráfica com o Usuário (GUI) e o Núcleo do Software. A Interface com o Usuário irá auxiliar o estudante na realização de sua prova e, posteriormente, permitindo que professores e pedagogos possam avaliar o nível de conhecimento destes alunos sobre uma determinada área. Esta interação possibilita que através dessas respostas, pode-se fazer uma análise, e através do resultado da análise inferir a qualidade do ensino fundamental na área de matemática em uma determinada escola. Portanto, a GUI é responsável por permitir o contato do aluno com as provas formuladas a partir da seleção das questões armazenadas em um banco de dados através de agente de software. E, também, possibilita o contato dos professores e pedagogos com os rendimentos dos alunos através dos resultados obtidos conforme os modelos matemáticos baseados na Teoria da Resposta ao Item.

O Núcleo do Software é composto por Agentes de Software, uma Base de Dados e da Teoria da Resposta ao Item. Os agentes de software, advindos da Inteligência Artificial, ficarão responsáveis pela escolha dos itens armazenados em um banco de dados para a elaboração de uma prova, pelo monitoramento em relação à resolução dos itens, como por exemplo, o tempo gasto para responder cada item e, por último, terá um agente responsável por disponibilizar informações (dicas) sobre um determinado assunto, abordado no item, estas dicas estarão representadas em uma ontologia. A base de dados será responsável pelo armazenamento dos dados cadastrais das pessoas (Profissionais da Educação, Funcionários Administrativos e Alunos) envolvidas no processo de avaliação educacional como um todo, também das informações sobre a escola, e dos itens que constituirão a prova a ser criada. De um modo geral, essa arquitetura apresenta como o sistema será estruturado, as interações entre os elementos e uma idéia de como ele atingirá seus objetivos.



**Figura 12 – Arquitetura do Virtual-TANEB.**

De acordo com a arquitetura definida acima, o sistema tem três abordagens fundamentais: a primeira proverá ao estudante um suporte amigável e interativo através de sua interface, guiando-os pela atividade de responder os itens de uma prova. A segunda é a utilização da Teoria da Resposta ao Item (TRI) para analisar quais as respostas foram efetuadas corretamente e qual o conhecimento do aluno sobre aquele item. A última abordagem define que os professores e os pedagogos realizarão um diagnóstico após as atividades de resolução dos itens, identificando o que é preciso fazer para impulsionar a aprendizagem dos alunos sobre o respectivo assunto analisado.

A Figura 13 exhibe o fluxo de informações a serem analisadas a partir da entrada que contém os dados referentes aos alunos e aos itens da prova, seguindo para uma avaliação através da Teoria da Resposta ao Item e, por fim, retorna como saída o desempenho do estudante sobre a resolução da prova, em forma de amostra visualizada por meio de gráficos.



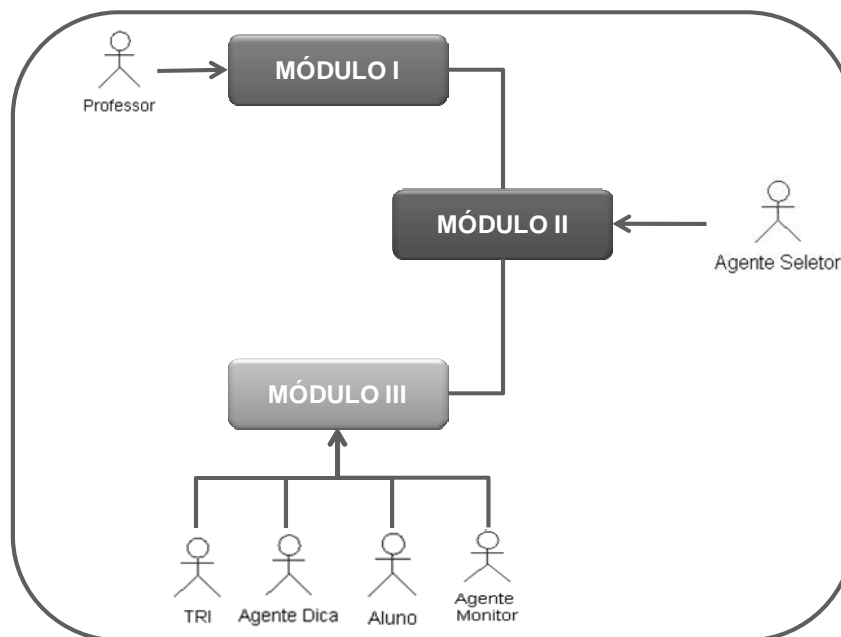
**Figura 13 – Fluxo de informações.**

É importante salientar que a aplicação da Teoria da Resposta ao Item nesse processo é fundamental, pois através dela é possível determinar o modelo matemático a ser adotado na avaliação da capacidade de conhecimento do aluno em relação à resolução de provas sobre a área de matemática. Assim, o Virtual-TANEB levará em consideração o modelo logístico de três parâmetros, a fim de possibilitar a construção de uma estimativa (um nível de conhecimento) de uma única população analisando a afinidade e as habilidades contidas em cada aluno.

### 5.3 Diagramas do Virtual-TANEB

No software Virtual-TANEB (VT), já estão definidos os agentes participantes do sistema (seletor, monitor, dica) e seus respectivos papéis. Sendo assim, para o desenvolvimento do nosso software, definimos primeiramente o contexto no qual o processo de avaliação ocorrerá a partir do VT através da Visão Organizacional do Software. Em seguida, utilizamos o Processo Unificado (Booch et al. 2000) com a notação UML (Guedes 2008) para descrever as tarefas e interações dos agentes envolvidos no processo de elaboração e resolução dos itens da prova.

A Figura 14, a seguir, exibe uma visão geral, dividida em módulos do funcionamento do VT, analisando a estrutura organizacional em que os agentes de software (seletor, monitor, dica) estão inseridos e definindo os relacionamentos de familiaridade que estes agentes possuem.



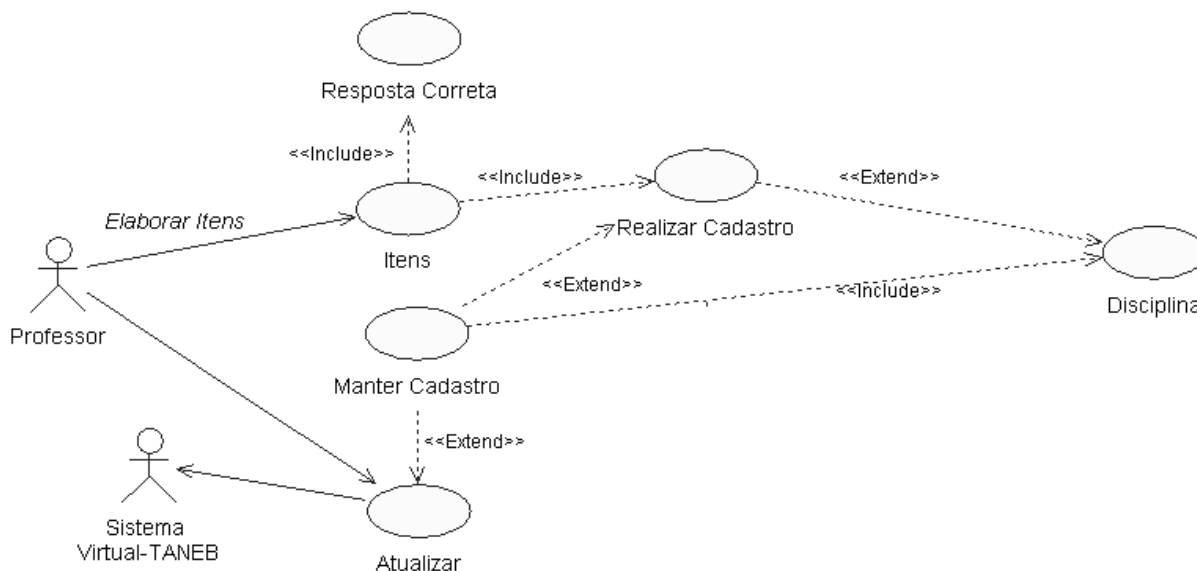
**Figura 14 – Visão Geral do Virtual-TANEB.**

O Módulo I é a parte do sistema que fica direcionado ao cadastramento dos dados pessoais dos profissionais envolvidos neste software, dentre eles, estão os professores, pedagogos, alunos, gestores e funcionários administrativos envolvidos na formulação e resolução dos itens, também cadastrados nesta seção. Já o Módulo II, é parte integrante do sistema responsável pela elaboração da prova, aonde temos uma interação entre o Agente Seletor e um Repositório de Dados no processo de selecionar itens para a formulação da prova a ser realizada pelos alunos. E, por último, participam ainda o Agente Monitor e a Agente Dica no Módulo III. O primeiro agente fazendo o monitoramento do desempenho do aluno, desde o tempo gasto para responder cada item, até a verificação do nível de desempenho por itens acertados e, o segundo agente aqui citado, ficando responsável por disponibilizar informações que possam auxiliar o discente na realização da prova. Todos fazendo uso de alguns repositórios de dados, dentre eles, uma ontologia que será usada pelo Agente Dica.

### 5.3.1 Diagrama de Casos de Usos

A partir da Visão Geral do VT exibida anteriormente, identificaram-se os Casos de Uso, do Módulo I, para cadastrar itens demonstrados na Figura 15. Este cadastro será feito a partir da observação de alguns aspectos como, por exemplo, os

descritores que conforme (Brasil 1997), são responsáveis por expressar a totalidade dos indicadores necessários para a orientação da construção de itens de prova que irão compor a Base de Itens do Virtual-TANEB.



**Figura 15 – Diagrama de Casos de Uso “Cadastrar Itens”.**

O processo de cadastramento dos itens, somente será visto como concluído após a última verificação dos pedagogos e dos professores, que analisaram os itens, apontando críticas e sugestões. Assim, estas críticas e sugestões consolidaram o trabalho de elaboração e cadastramento dos itens.

Nesse caso de uso, faz-se necessário entender o cadastro de itens que são todas as ações que estão relacionadas à formulação dos itens até o seu armazenamento. Na Tabela 2 a seguir, estão descritos os atores e casos de usos que fazem parte da Figura 15 exposta anteriormente.



Tabela 2 – Documentação do Caso de Uso Cadastrar Itens

|  |   |
|--|---|
| <b>Nome do Caso de Uso</b>   | <b>Cadastrar Itens</b>  |
| <b>Ator Principal</b>  | Professor   |
| <b>Ator Secundário</b>   | Sistema Virtual-TANEB   |
| <b>Resumo</b>  | Este caso de uso descreve as etapas percorridas por um professor para formular e armazenar itens.   |
| <b>Ações do Ator</b>   | <b>Ações do Sistema</b>   |
| 1. Elaborar Itens  |   |
| 2. Verificar os parâmetros dos descritores estabelecidos pelo SAEB |   |
|  | 3. Se for necessário, gravar ou atualizar o cadastro do Item.                                       |
| <b>Restrições/Validações</b>                                       | 1. Para elaborar um item é preciso seguir os parâmetros estabelecidos pelo SAEB para os descritores |
|  | 2. O item só pode ser formulado pelo professor da área.   |

Na Figura 16, a seguir, é exibido os Casos de Uso pertencentes ao Módulo II do software Virtual-TANEB, que traz a atividade a ser executada pelo Agente Seletor.

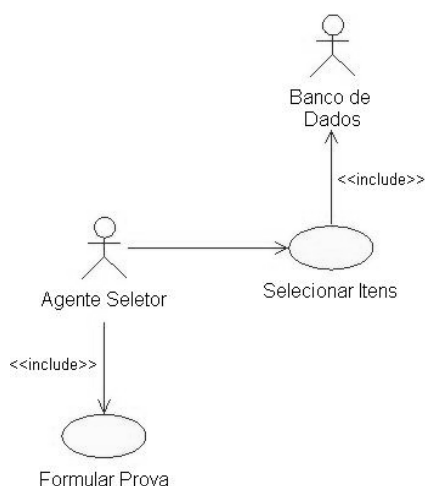


Figura 16 – Diagrama de Casos de Uso “Selecionar Itens”.

O Agente Seletor inicia o processo de formulação da prova selecionando os itens armazenados no banco de dados de acordo com os descritores estabelecidos pelo Sistema Nacional de Avaliação da Educação Básica (SAEB) para a matemática da quarta série do ensino fundamental, respectivamente sobre a área de geometria (espaço e forma). Este verifica se as solicitações seguem o padrão dos

descritores, registra-os, faz a seleção e repassa os itens para as respectivas provas a serem realizadas pelos alunos.

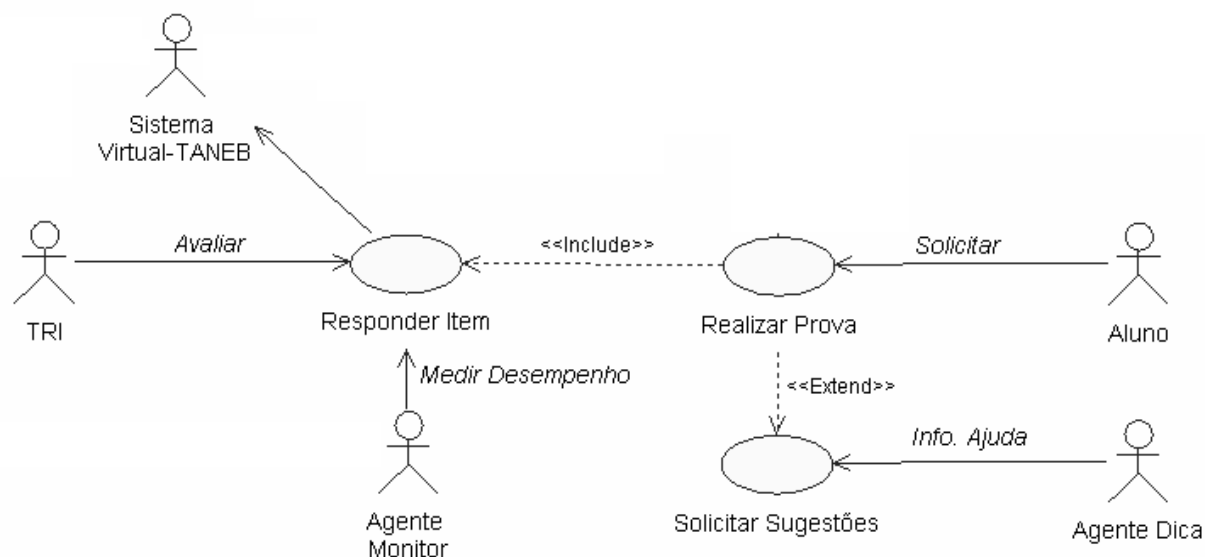
Na Tabela 3 a seguir, estão descritos os atores e casos de usos que fazem parte da Figura 16 exposta anteriormente.

**Tabela 3 – Documentação do Caso de Uso Selecionar Itens**

| <b>Nome do Caso de Uso</b>  | <b>Cadastrar Itens</b>   |
|---|--|
| <b>Ator Principal</b>   | Agente Seletor   |
| <b>Ator Secundário</b>  | Banco de Dados   |
| <b>Resumo</b>   | Este caso de uso descreve as etapas a serem executadas pelo agente seletor no processo de selecionar itens que estão armazenados em um repositório de dados, para formular uma prova |
| <b>Pré-Condições</b>  | O pedido de seleção precisa ser ativado  |
| <b>Pós-Condição</b>   | É necessário realizar uma análise inicial  |
| <b>Ações do Ator</b>  | <b>Ações do Sistema</b>  |
| 1. Solicitar a seleção dos Itens  |  |
| 2. Verificar os parâmetros estabelecidos pelo SAEB para os descritores no processo de realização da seleção |  |
|   | 3. Avaliar o pedido de seleção   |
|   | 4. Aprovar o pedido de seleção   |
| 5. Escolher o item no repositório   |  |
|   | 6. Definir Item selecionado  |
| 7. Fornecer o item selecionado a prova a ser aplicada.  |  |
| <b>Restrições/Validações</b>  | 1. Para selecionar um item é preciso seguir os parâmetros do SAEB  |

A partir da Visão Geral do Virtual-TANEB apresentada, identificou-se os Casos de Uso do Módulo III, realizar prova demonstrada na Figura 17. Neste módulo, o aluno será analisado com o objetivo de identificar as suas aptidões (habilidades) e deficiências sobre o tema abordado nos itens a serem respondidos. Na análise serão utilizados dois mecanismos que possibilitará a execução correta do

objetivo proposto aqui, são eles: o primeiro é o modelo matemático de função logística de três parâmetros da Teoria da Resposta ao Item, que caracteriza cada questão em 03 parâmetros, conforme pode ser visto no capítulo 03 e o segundo, um agente de software responsável pelo monitoramento na resolução dos itens, como por exemplo, a verificação do nível de desempenho por itens acertados.



**Figura 17 – Diagrama de Casos de Uso “Realizar Prova”.**

Na Tabela 4 a seguir, estão expostos os atores e casos de usos que fazem parte da Figura 17 exibida anteriormente.

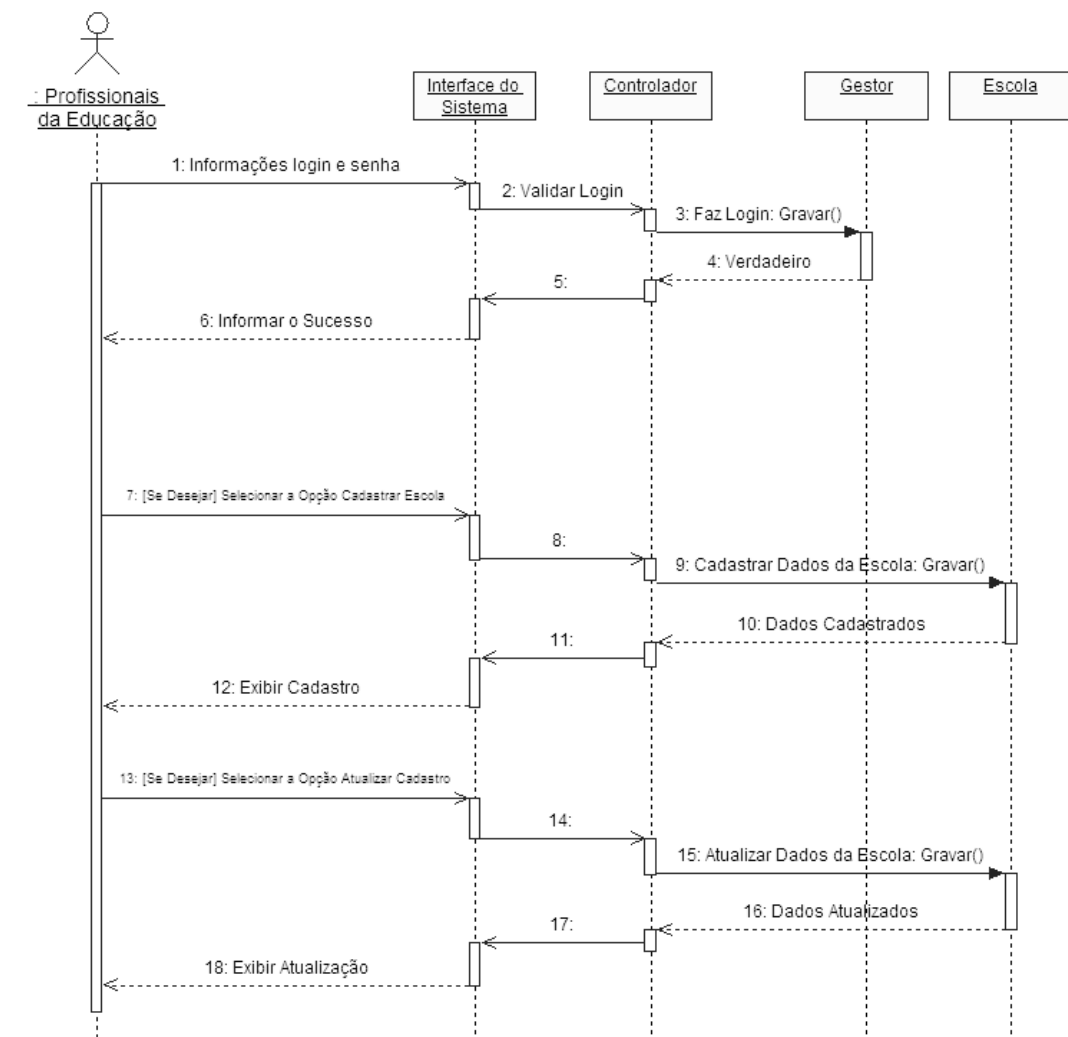
**Tabela 4 – Documentação do Caso de Uso Realizar Prova**

|   |  |
|---|--|
| <b>Nome do Caso de Uso</b>                            | <b>Cadastrar Itens</b>   |
| <b>Ator Principal</b>                                 | Agente Monitor   |
| <b>Atores Secundários</b>                             | Agente Dica, TRI, Aluno, Sistema Virtual-TANEB   |
| <b>Resumo</b>   | Este caso de uso informa como será executado o processo de realização da prova pelo aluno                  |
| <b>Pré-Condições</b>                                  | A prova precisa ser formulada através da seleção dos itens pelo agente seletor                             |
| <b>Ações do Ator</b>                                  | <b>Ações do Sistema</b>  |
| 1. Solicitar a realização da prova                    |  |
|   | 2. Aplicar a prova   |
| 3. Auxiliar os alunos com sugestões sobre as questões |  |
| 4. Avaliar a habilidade do aluno                      |  |
|   | 5. Gerar relatórios sobre o desempenho do aluno  |
| <b>Restrições/Validações</b>                          | 1. Para responder a prova, faz-se necessário o aluno, a escola e a disciplina estão cadastradas no sistema |

### 5.3.2 Diagrama de Seqüência

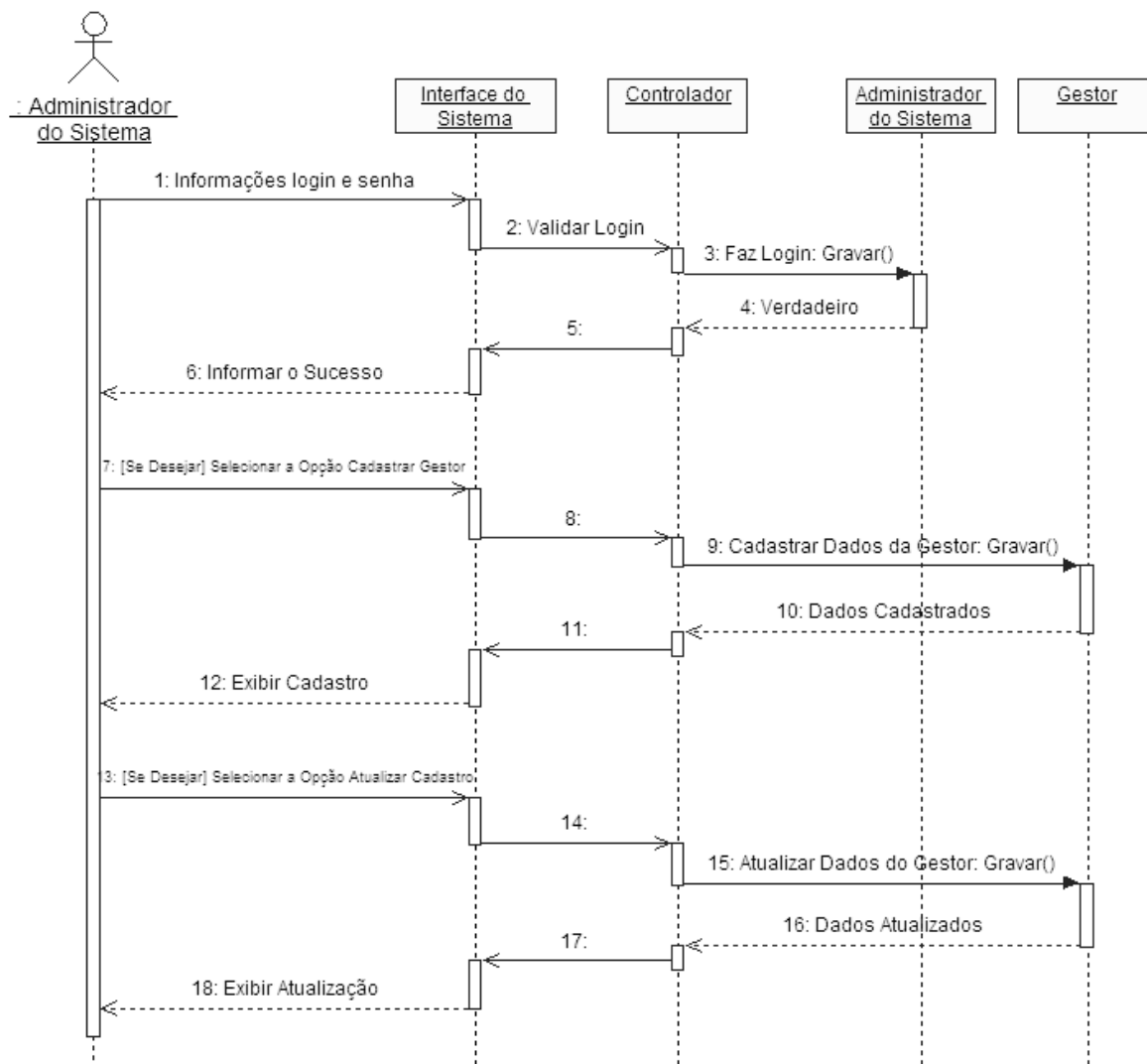
Na seção anterior, foram descritos os casos de usos do Virtual-TANEB dividido em módulos. Já nesta seção, serão apresentados os diagramas de seqüência, cujos mesmos, procuram determinar a seqüência de eventos e quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem ocorreu no processo de cadastro efetuado pelo Virtual-TANEB.

A Figura 18 adiante ilustra o diagrama de seqüência do Virtual-TANEB, cuja principal utilidade no processo de desenvolvimento do software é servir como uma etapa para se armazenar as informações relativas ao processo de cadastramento da escola sob o funcionamento do VT.



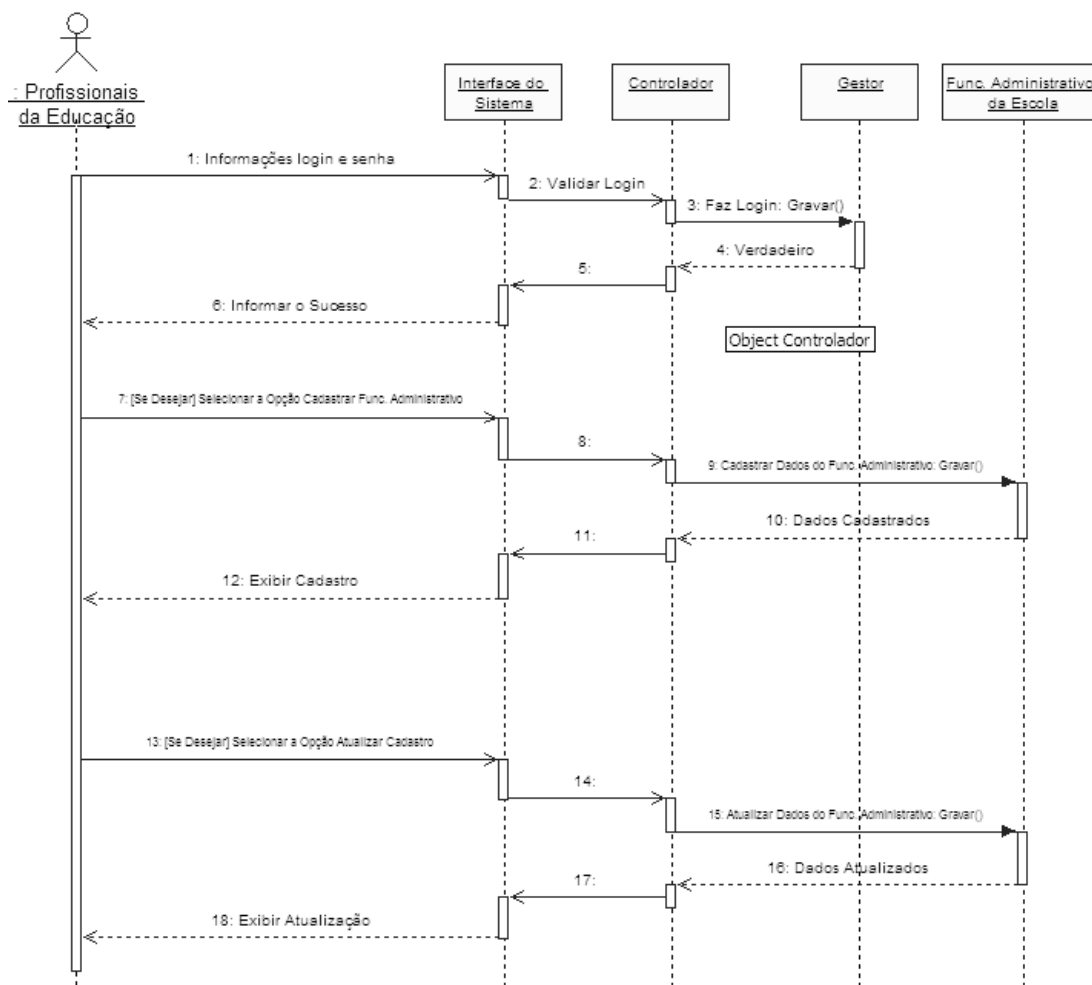
**Figura 18 – Diagrama de Seqüência “Cadastrar Escola”.**

O diagrama de seqüência mostrado na Figura 19 ilustra o processo para cadastrar o gestor escolar, focalizando a imagem do administrador do sistema responsável por realizar este cadastro e, a imagem do gestor, habilitado a realizar suas atividades diante do software Virtual-TANEB.



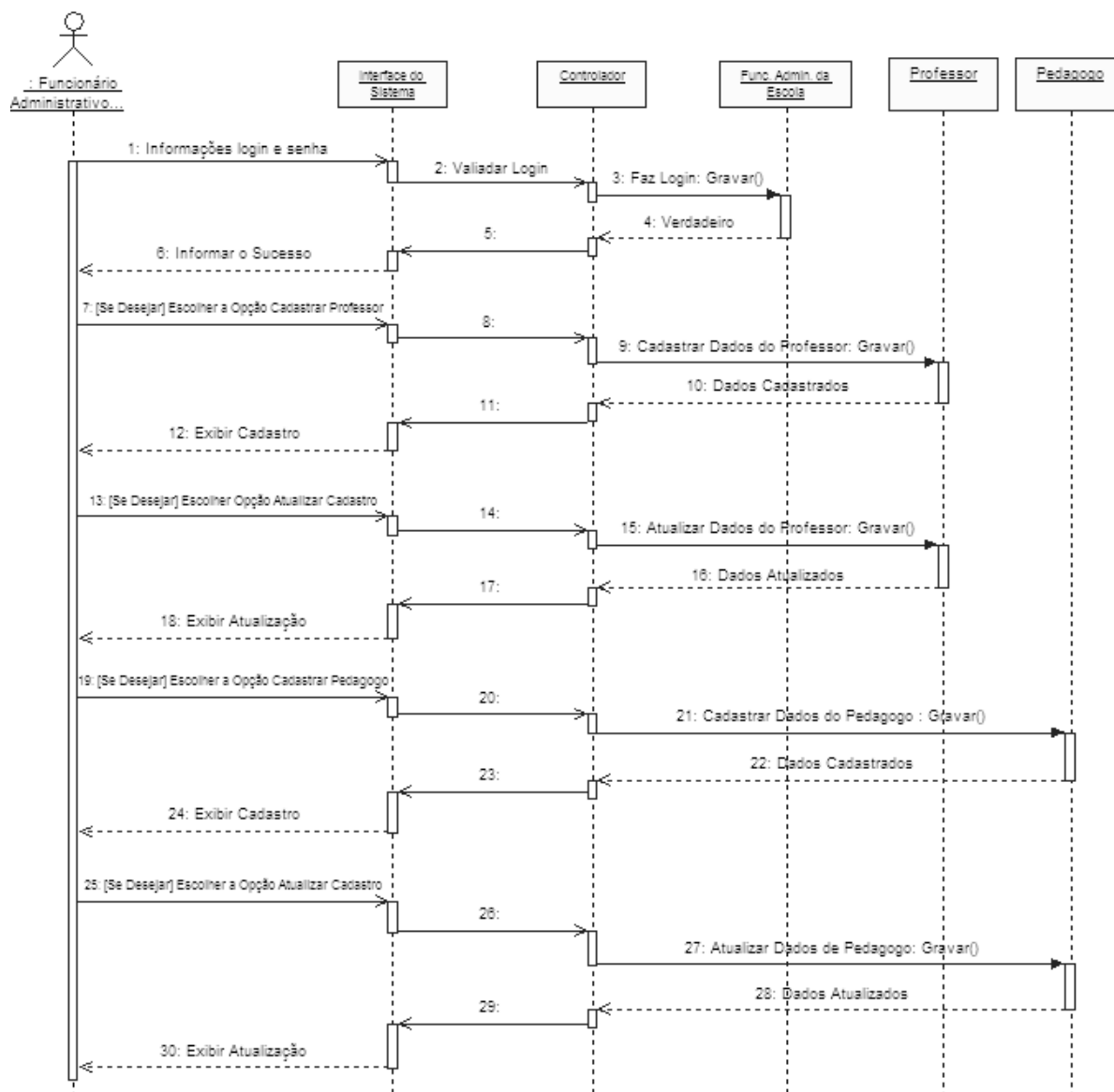
**Figura 19 – Diagrama de Seqüência “Cadastrar Gestor”.**

A Figura 20 a seguir, apresenta o diagrama de seqüência que mostra os passos necessários para realizar o cadastramento dos funcionários administrativos da escola. A responsabilidade de efetuar este cadastro é do administrador do sistema.



**Figura 20 – Diagrama de Seqüência “Cadastrar Funcionário Administrativo”.**

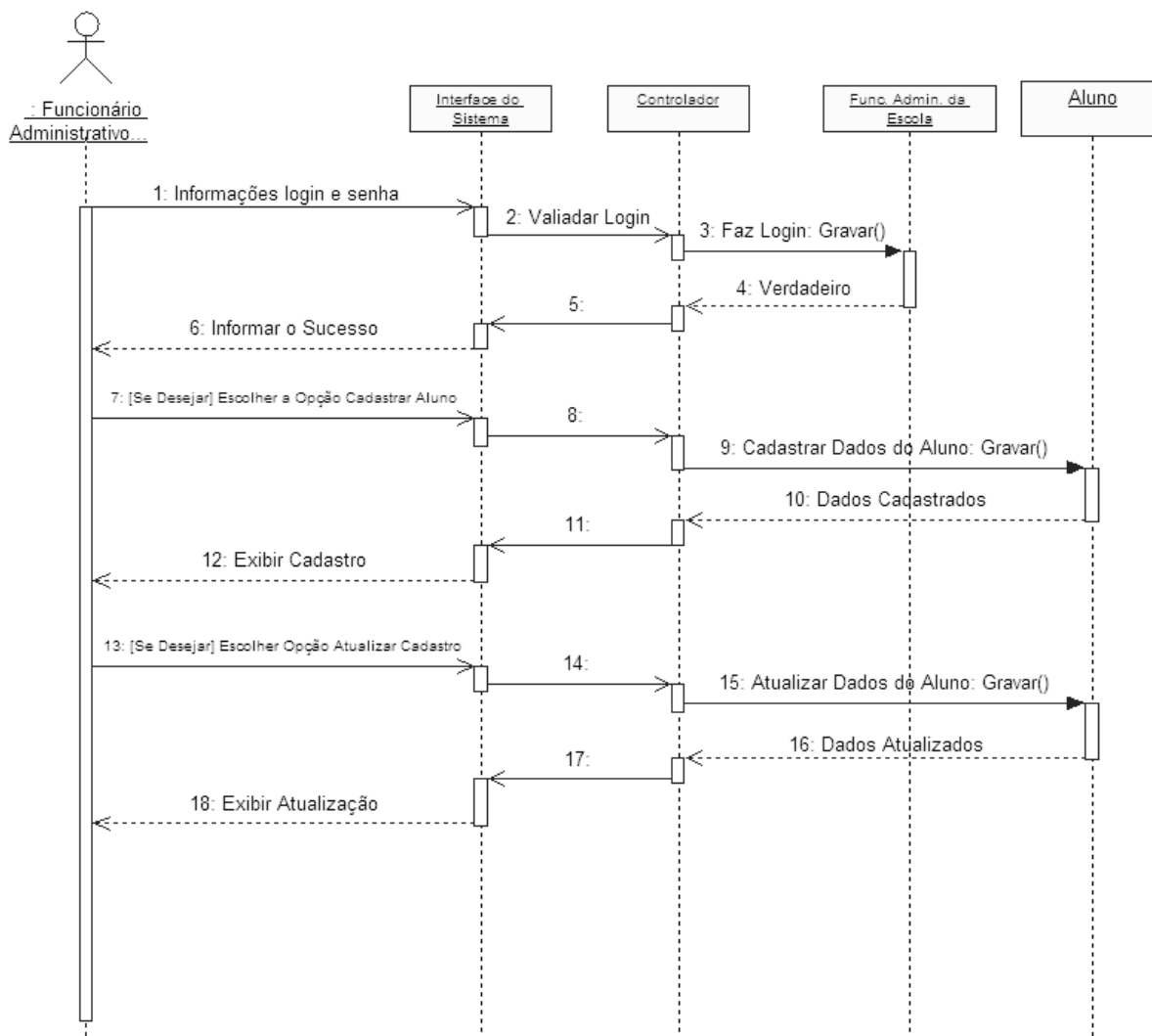
No diagrama apresentado na Figura 21, definimos o processo para cadastrar os professores e os pedagogos que fazem parte da escola cadastrada no sistema Virtual-TANEB. Esta tarefa é de responsabilidade dos funcionários administrativos, cadastrados anteriormente.



**Figura 21 – Diagrama de Seqüência “Cadastrar Professor e Pedagogo”.**

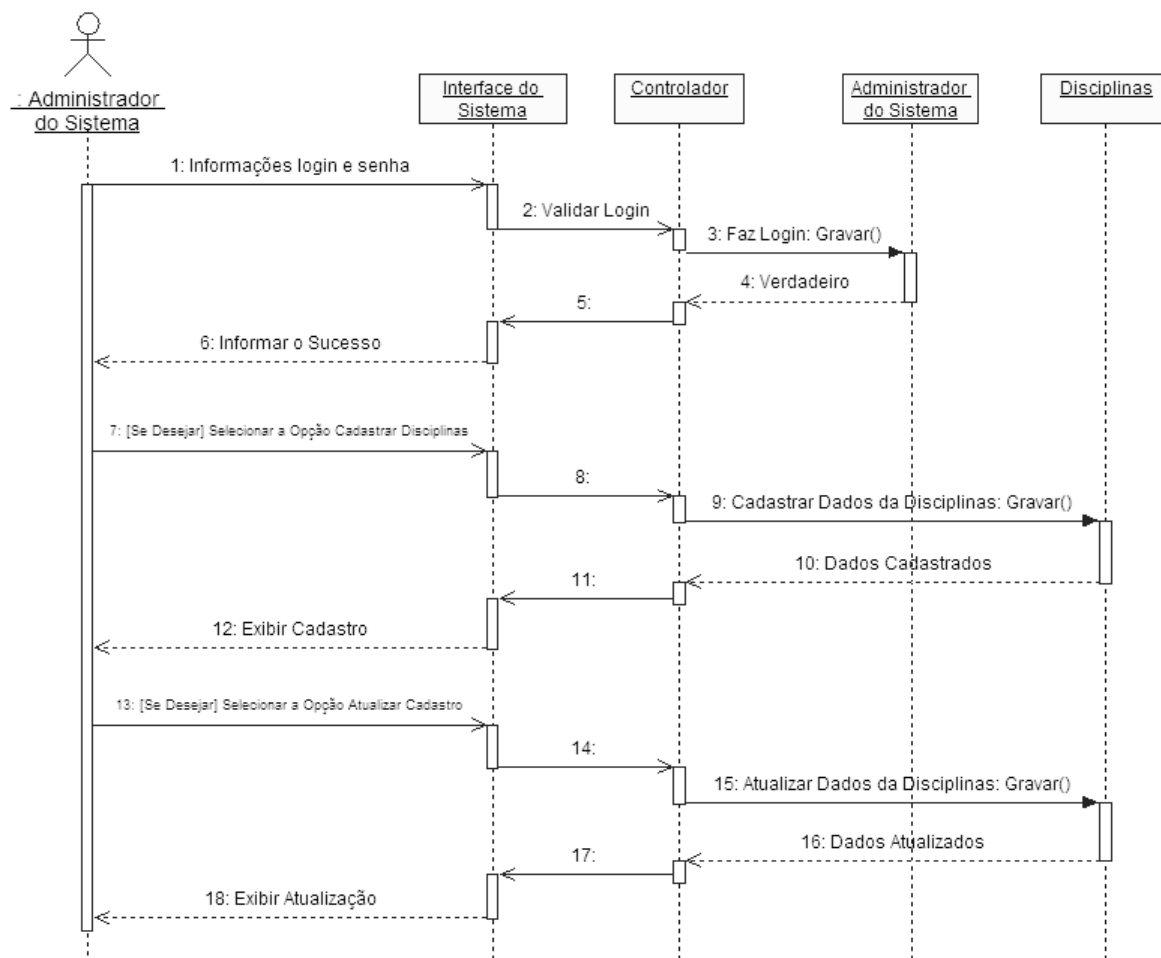
No diagrama de seqüência apresentado na Figura 22 a seguir, é exibido o processo de cadastramento dos alunos. Este cadastro é de responsabilidade dos funcionários administrativos da escola já cadastrada.





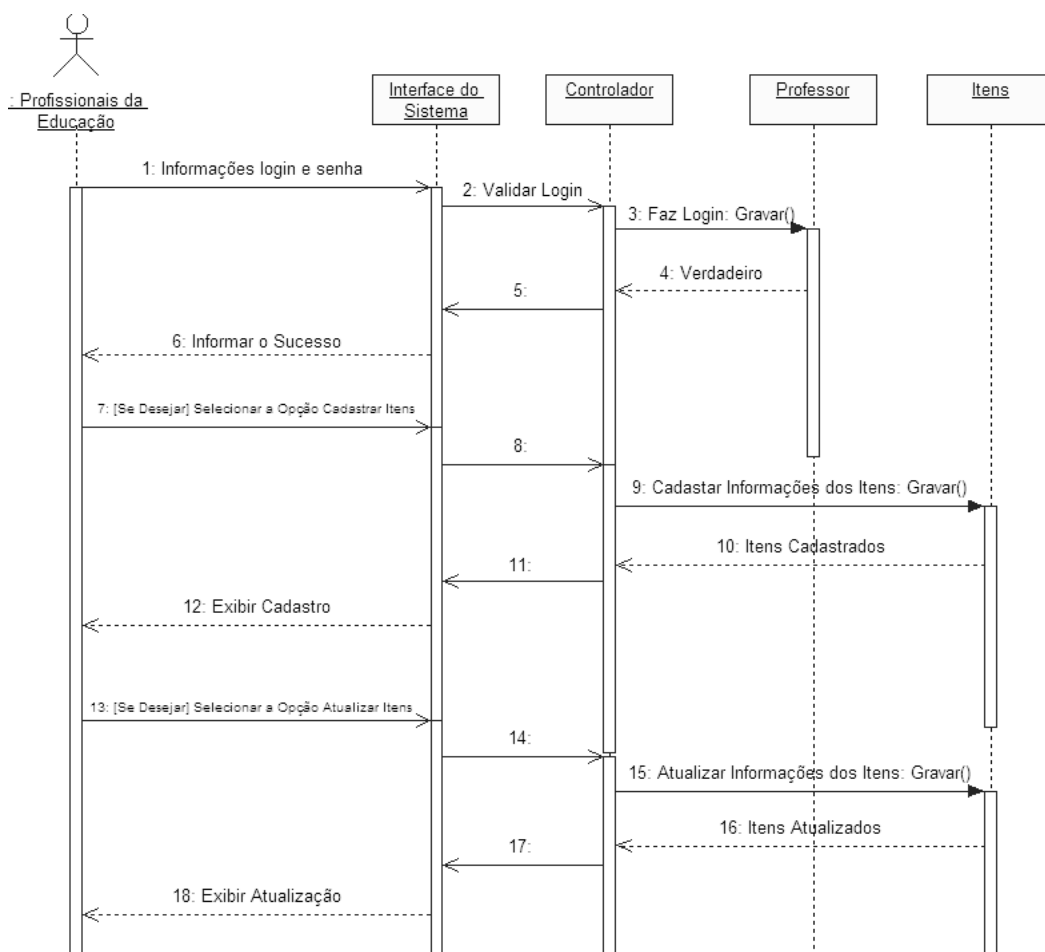
**Figura 22 – Diagrama de Seqüência “Cadastrar Aluno”.**

O diagrama de seqüência mostrado na Figura 23 ilustra o processo para cadastrar disciplinas, focalizando a imagem do administrador do sistema responsável por realizar este cadastro e, a imagem da disciplina diante do software Virtual-TANEB.



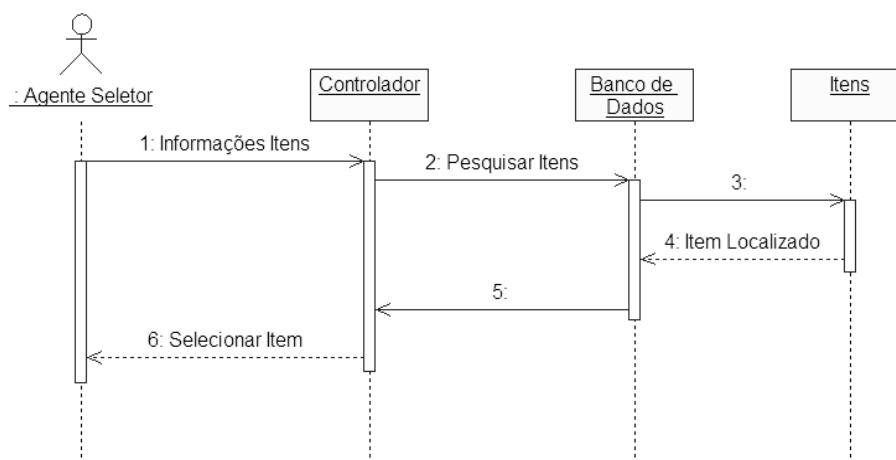
**Figura 23 – Diagrama de Seqüência “Cadastrar Disciplinas”.**

A Figura 24 a seguir, ilustra o processo para cadastrar os itens, focalizando a imagem do professor responsável pela elaboração e correção dos itens a serem armazenados no repositório de dados do Virtual-TANEB.



**Figura 24 – Diagrama de Seqüência “Cadastrar Itens”.**

E por último, a Figura 25, exibe o diagrama de seqüência que traz o funcionamento do agente seletor que é buscar os itens armazenados obedecendo aos parâmetros estabelecidos para os descritores conforme o SAEB.



**Figura 25 – Diagrama de Seqüência “Selecionar Itens”.**

Diante dessas modelagens dos módulos do Virtual-TANEB apresentadas anteriormente, o VT visa à utilização de agentes de software e modelos matemáticos relacionados à Teoria da Resposta ao Item para melhorar o processo de avaliação da educação básica nas escolas, tanto de domínio público quanto de particular. Esta teoria, a TRI, no Virtual-TANEB será um módulo fracamente acoplado e fortemente coeso, possibilitando a sua interação no desenvolvimento de outras aplicações similares ou distintas do VT.

Portanto, o VT dará um suporte amigável e interativo para que alunos professores, pedagogos e gestores possam usufruir do software, visando identificar inicialmente os problemas existentes com o ensino da geometria da quarta série do ensino fundamental, ou seja, servirá para apontar falhas e soluções a serem implantadas relacionadas a este ensino.

#### **5.4 Considerações Finais**

Neste capítulo foi descrita a estrutura do VT com o objetivo de identificar algumas funcionalidades primordiais do Virtual-TANEB, bem como as contribuições referentes à arquitetura geral do sistema, o desenvolvimento dos diagramas de casos de uso e de seqüência, explanando as funcionalidades que estarão em foco no próximo capítulo deste trabalho em forma de implementação do software VT.

## **6 IMPLEMENTAÇÃO DO SOFTWARE VIRTUAL-TANEB PARA A AVALIAÇÃO DA APRENDIZAGEM**

### **6.1 Considerações Iniciais**

É notório o crescimento cada vez mais acelerado da qualidade do ensino brasileiro. E devido a isso, que atualmente vários software tem surgido, visando analisar e solucionar problemas relacionados à educação. Por outro lado, como o leque de sistemas se torna cada vez mais extenso, as instituições de ensino se tornaram mais exigentes, forçando os grupos de pesquisas e empresas voltadas a área de desenvolvimento de software educacional a oferecer um sistema personalizado e eficiente na atividade, buscando uma condição satisfatória ao ensino público e privado.

Dessa maneira, este capítulo busca especificar um software, chamado Virtual-TANEB, que satisfaça à necessidade das instituições de ensino público e privado no quesito de avaliação do nível de aprendizagem dos alunos na disciplina matemática da quarta série do ensino fundamental. Assim, é previsto que o software proposto disponha de interfaces distintas: uma para os cadastros com as informações pessoais dos usuários e outra para cadastrar os itens em um banco de dados que serão usados na elaboração da prova de forma automática pelo sistema.

O objetivo deste capítulo é apresentar uma implementação que comprove a efetividade da modelagem proposta no Capítulo 5 do Virtual-TANEB, observando conceitos das tecnologias usadas. Além desta seção com as considerações iniciais, temos a seção 6.2 que relata como cadastrar informações gerais (tanto do usuário como dos itens). A seção 6.3 apresenta as considerações finais do capítulo.

### **6.2 Cadastrar no Virtual-TANEB**

Esta seção apresenta algumas interfaces do Virtual-TANEB. Para a implementação do protótipo, foi utilizado o JSP (*JavaServer Pages*) que é uma tecnologia baseada na linguagem Java, que simplifica o processo de desenvolvimento dinâmico de aplicativos voltados a Web. Foram usadas, ainda, as plataformas JADE (*Java Agent Development Framework*) e JESS (*Java Expert System Shell*) que através de formalização foram usadas para criar regras de



“Gestor Escolar”, através do fornecimento de seus dados pessoais. Este processo é feito por meio da Interface de Cadastro de Usuários visualizados na Figura 27.

**Figura 27 – Interface de Cadastro de Usuários no Virtual-TANEB.**

Na etapa subsequente, após o cadastramento do “Gestor Escolar”, executa-se o *login* (informando o nome de usuário cadastrado e senha) por meio da Interface de *Login* do Virtual-TANEB (Figura 28) e, em seguida, ele realiza o cadastro dos “Funcionários Administrativos” daquela escola já cadastrada pelo sistema seguindo os passos estabelecidos pela Interface de Cadastro (Figura 27). Por último, os “Funcionários Administrativos” são responsáveis pelo cumprimento de algumas atividades, dentre elas está o cadastramento de outros membros que compõem a escola já efetivada pelo sistema, são eles: os “Professores”, os “Pedagogos” e os “Alunos”. Este último sendo habilitado para a resolução da prova que medirá seu desempenho sobre o assunto da geometria da quarta série do ensino fundamental.

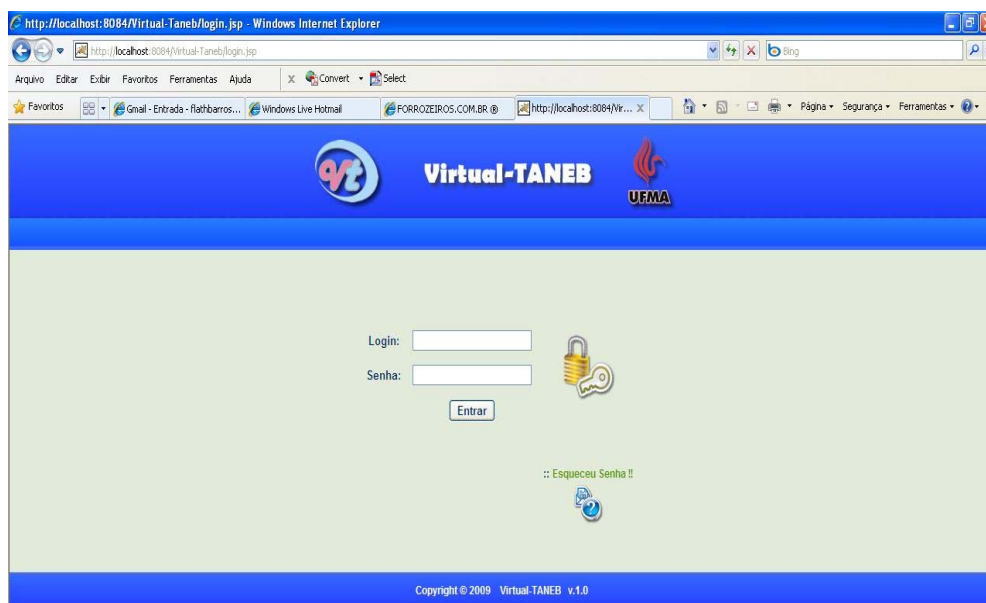


Figura 28 – Interface do **Login** no Virtual-TANEB.

Depois da realização de todos os cadastros referentes aos usuários, faz-se necessário uma interface para a visualização de todas as pessoas inseridas pelo sistema Virtual-TANEB. Esta interface se encontra exibida na Figura 29 que traz todas as pessoas cadastradas no VT, ou seja, ilustra os usuários aptos a manusear o software implementado aqui.

| OPERAÇÃO           | NOME            | EMAIL            | DATA / NASCIMENTO | SEXO | MÃE               | IDENTIDADE | ORG / EMISSOR | DATA / EXPEDIÇÃO | LOGIN   | ENDEREÇO                       | TIPO / USUARIO           | INST / CNPJ | PAÍS / NAC.         | COR / PELE | FUNÇÃO  | CARGO  |
|--------------------|-----------------|------------------|-------------------|------|-------------------|------------|---------------|------------------|---------|--------------------------------|--------------------------|-------------|---------------------|------------|---------|--------|
| Alterar    Excluir | Itena           | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | n11     | n, 9 - 87 n                    | Administrador do Sistema | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Jackson         | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | jackson | n, 9 - 87 n                    | Administrador do Sistema | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Kaka            | ad               | 26/11/1985        | M    | adljfakdjfl       | 784        | SSP           | 03/03/2003       | kaka    | adfafef, 7 - 65120-000 adfaj   | Administrador do Sistema | 123         | Brasil / Brasileiro | Amarela    | Sem     | Sem    |
| Alterar    Excluir | Maria           | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | maria   | n, 9 - 87 n                    | Aluno                    | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Fábio           | aktj             | 23/01/1987        | M    | d                 | 34         | ddd           | 25/12/1999       | n22     | efr, 345 - 34 fefef            | Gestor                   | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Francisco       | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | n2      | n, 9 - 87 n                    | Gestor                   | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Carlos José     | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | n5      | n, 9 - 87 n                    | Pedagogo                 | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Dudu dos Santos | santos@gmail.com | 13/10/1974        | M    | Lurdes dos Santos | 125487     | SSM-MA        | 03/05/2001       | dudu    | blablable, 78 - 65000-000 Anil | Professor                | 123         | Brasil / Brasileiro | Parda      | Nenhuma | Nenhum |
| Alterar    Excluir | José Flávio     | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | flavio  | n, 9 - 87 n                    | Professor                | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Luiz            | aktj             | 23/01/1987        | M    | d                 | 34         | ddd           | 25/12/1999       | n33     | efr, 345 - 34 fefef            | Tecnico Administrativo   | 123         | Brasil / Brasileiro | Amarela    | n       | n      |
| Alterar    Excluir | Gugu            | n                | 12/10/1984        | M    | n                 | 98         | m             | 30/12/2001       | n3      | n, 9 - 87 n                    | Tecnico Administrativo   | 123         | Brasil / Brasileiro | Amarela    | n       | n      |

Figura 29 – Interface dos **Usuários Cadastrados** no Virtual-TANEB.

O Virtual-TANEB possui um procedimento que permite ao usuário recuperar sua senha, caso ele a esqueça, viabilizado através da interface exibida na



Figura 30 (abaixo). Ela ilustra a funcionalidade de recuperação da senha e *login* do usuário do VT via email.



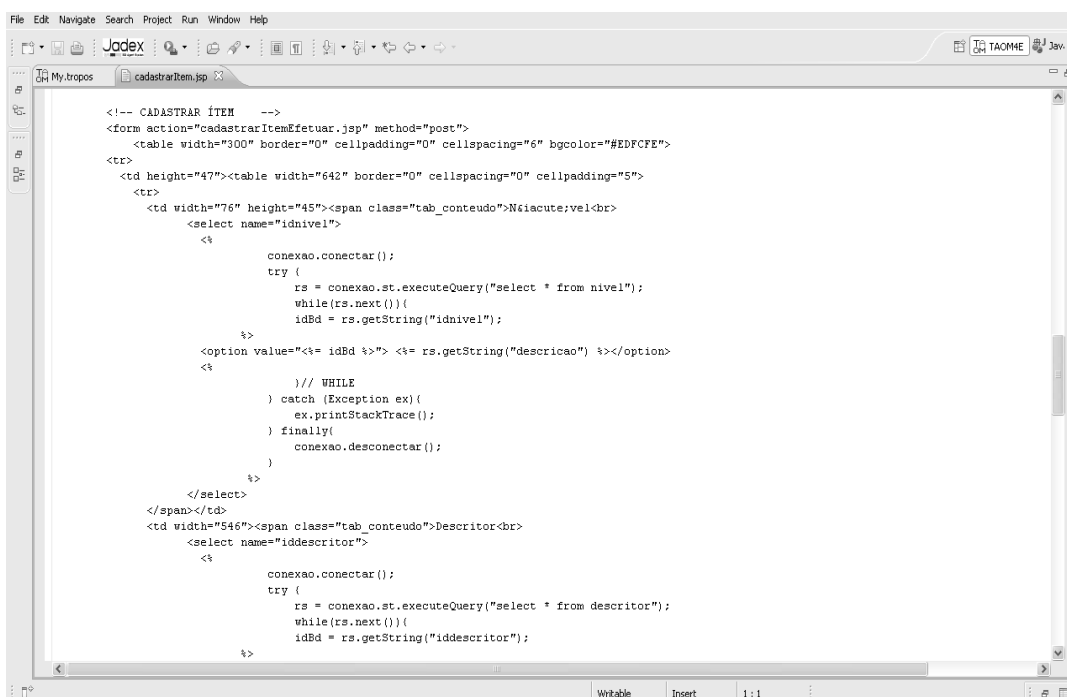
**Figura 30 – Interface para Recuperar Senha no Virtual-TANEB.**

Outra atividade importante no Virtual-TANEB é o procedimento de cadastrar os itens seguindo os parâmetros estabelecidos pelo SAEB, que serão aproveitados na elaboração da prova de matemática, a ser respondida pelos alunos da quarta série do ensino fundamental.

Os itens antes do efetivo cadastro passarão por uma análise e, em consequência disso, por possíveis correções, caso haja problemas. Esta verificação é feita por meio de observações dadas pelos pedagogos baseados nos descritores estabelecidos pelo SAEB para a matemática da quarta série do ensino fundamental. Após todo este procedimento, os itens são finalmente cadastrados no Banco de Dados.

Os itens ficarão armazenados no Banco de Dados (BD) até o momento que de realização da prova (via *WEB*). Assim, ao ser informado o início desta avaliação, um agente chamado de “Agente Seletor” é disparado, fazendo a seleção aleatória dos itens, proporcionando a montagem da prova.

A Figura 31 a seguir, mostra um trecho do código desenvolvido na concepção da Interface de Cadastro dos Itens. E a Figura 32 ilustra a Interface responsável pelo cadastramento desses itens.



```

<!-- CADASTRAR ÍTEM -->
<form action="cadastrarItemEfeetuar.jsp" method="post">
  <table width="300" border="0" cellpadding="0" cellspacing="6" bgcolor="#EEDFCFE">
  <tr>
  <td height="47"><table width="642" border="0" cellpadding="0" cellspacing="5">
  <tr>
  <td width="76" height="45"><span class="tab_conteudo">Nível<br>
  <select name="idnivel">
  <%
  conexao.conectar();
  try {
  rs = conexao.st.executeQuery("select * from nivel");
  while(rs.next()){
  idBd = rs.getString("idnivel");
  <%
  <option value="<%= idBd %%"> <%= rs.getString("descricao") %></option>
  <%
  // WHILE
  } catch (Exception ex){
  ex.printStackTrace();
  } finally{
  conexao.desconectar();
  }
  <%
  </select>
  </span></td>
  <td width="546"><span class="tab_conteudo">Descriptor<br>
  <select name="iddescriptor">
  <%
  conexao.conectar();
  try {
  rs = conexao.st.executeQuery("select * from descriptor");
  while(rs.next()){
  idBd = rs.getString("iddescriptor");
  <%
  </select>
  </td>
  </tr>
  </table>
  </tr>
  </table>
  </form>
  
```

Figura 31 – Trecho do Código para Cadastrar Itens.



Figura 32 – Interface para Cadastrar Itens no Virtual-TANEB.

A Figura 33 mostra um trecho do código do desenvolvimento da Interface que permite a visualização dos itens cadastrados no Virtual-TANEB.

```

File Edit Navigate Search Project Run Window Help
Jadex TAOMME Jav.
My.tropos cadastrarLayout.jsp visualizarItem.jsp
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Virtual-TANEB :: Visualizar Usuários</title>
<link rel="stylesheet" type="text/css" href="style/css.css"/>
<style type="text/css">
<!--
body (background-image: url(imagens/fundo.jpg));
-->
</style>
</head>
<body leftmargin="0" topmargin="0" bottommargin="0">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="175" height="130" valign="top" background="imagens/topo.jpg"><div align="center"></div></td>
<td width="968" valign="top" background="imagens/topo.jpg"><div align="center">
</div>
<div align="center"></div></td>
<td width="217" valign="top" background="imagens/topo.jpg"><div align="center">
<a href="sair.jsp"></a><br>
<div class="topo"><%= UsuSession %><br><b><%= PrivSession %></b></div></td>
</tr>
<tr>
<td height="516" rowspan="2" valign="top"><table width="157" height="253" border="0" cellpadding="0" cellspacing="0">
<tr>
<td colspan="3"><div align="left"></div></td>
</tr>
<tr>
<td colspan="3"></td>
</tr>
<tr>
<td width="8" height="176" background="imagens/menu_esq.gif">&nbsp;</td>

```

Figura 33 – Trecho do Código para Visualizar os Itens.

| OPERAÇÃO          | NÍVEL    | DESCRIPTOR | PERGUNTA | IMAGEM PERGUNTA | ALTERNATIVAS   | ALTERNATIVA CORRETA |
|-------------------|----------|------------|----------|-----------------|--|---------------------|
| Alterar   Excluir | Básico   | D1         | P1       | img             | 1) p1_A01<br>imagem p1_A01_img1<br>2) p1_A02<br>imagem p1_A01_img2<br>3) p1_A03<br>imagem p1_A01_img3<br>4) p1_A04<br>imagem p1_A01_img4 | 1                   |
| Alterar   Excluir | Médio    | D1         | P2       | 2img            | 1) p2_A01<br>imagem p2_A01_img1<br>2) p2_A02<br>imagem p2_A01_img2<br>3) p2_A03<br>imagem p2_A01_img3<br>4) p2_A04<br>imagem p2_A01_img4 | 2                   |
| Alterar   Excluir | Avançado | D1         | P3       | P3_img          | 1) A01_P3<br>imagem img_A01<br>2) A02_P3<br>imagem img_A02<br>3) A03_P3<br>imagem img_A03<br>4) A04_P3<br>imagem img_A04                 | 3                   |
| Alterar   Excluir | Básico   | D1         | P4       | img_p4          | 1) A01<br>imagem<br>2) A02<br>imagem<br>3) A03<br>imagem<br>4) A04<br>imagem   | 1                   |

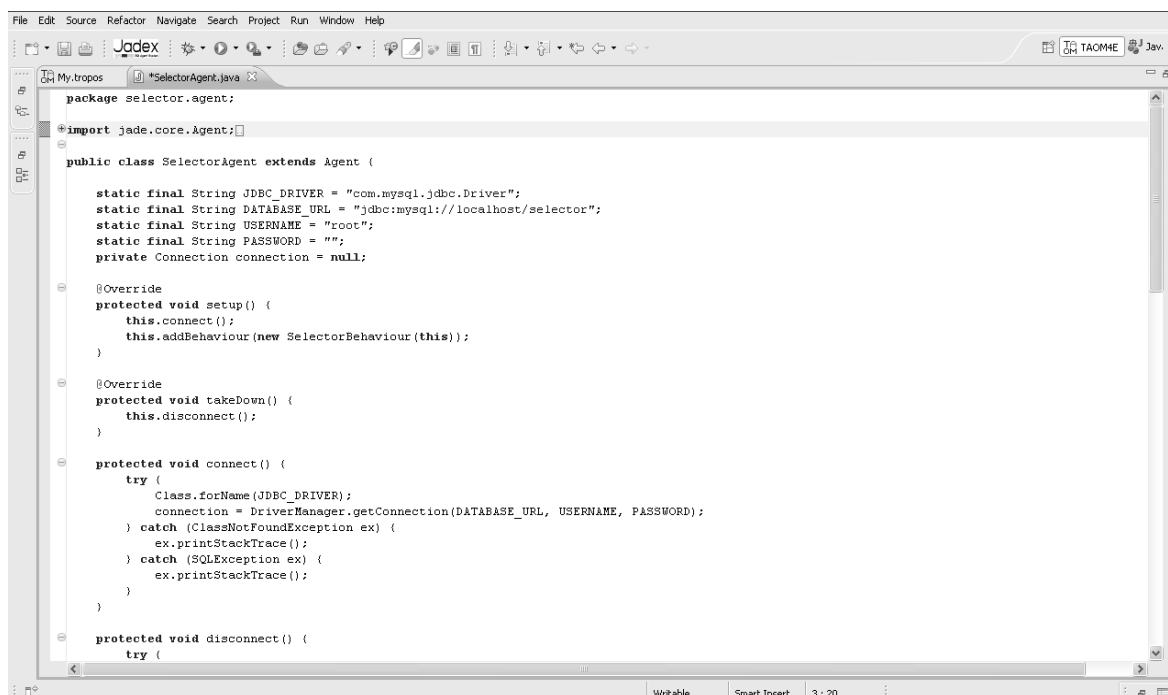
Figura 34 – Interface dos Itens Cadastrados no Virtual-TANEB.

A Figura 34 (acima) traz a Interface que exibe todos os itens efetivamente cadastrados pelo “Professor” seguindo as observações efetuadas pelos pedagogos

conforme os descritores estabelecidos através do Sistema Nacional de Avaliação da Educação Básica (SAEB).

Na etapa da realização da prova, alguns procedimentos são usados: (1) – o Gestor Escolar solicita ao Administrador do Sistema um agendamento determinando o dia e a hora que será realizada a prova; (2) – o Administrador faz este agendamento no sistema VT; (3) – no dia e na hora estabelecida o sistema dispara o Agente Seletor, ficando responsável pela escolha dos itens que farão parte da prova a ser respondida naquele momento; (4) – o aluno responde a prova, cuja avaliação foi formada a partir da seleção de itens por meio de um agente de software.

A Figura 35 ilustra um trecho do código da *Class SelectorAgent*, cuja classe é responsável pela estrutura organizacional do Agente Seletor. Como exemplo desta estrutura, temos a conexão a ser utilizada pelo agente para acessar o Banco de Dados que contém os itens armazenados.



```
package selector.agent;

import jade.core.Agent;

public class SelectorAgent extends Agent {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DATABASE_URL = "jdbc:mysql://localhost/selector";
    static final String USERNAME = "root";
    static final String PASSWORD = "";
    private Connection connection = null;

    @Override
    protected void setup() {
        this.connect();
        this.addBehaviour(new SelectorBehaviour(this));
    }

    @Override
    protected void takeDown() {
        this.disconnect();
    }

    protected void connect() {
        try {
            Class.forName(JDBC_DRIVER);
            connection = DriverManager.getConnection(DATABASE_URL, USERNAME, PASSWORD);
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    protected void disconnect() {
        try {
```

Figura 35 – Trecho do Código do Agente Seletor.

A Figura 36 mostra um trecho do código da *Class SelectorBehaviour*, cuja classe foi desenvolvida para determinar como o agente irá se comportar diante da funcionalidade empregada, ou seja, determinar quando ele deve disparar (funcionar)

seguindo as configurações estabelecidas pelo Administrador do Sistema (agendamento do dia e da hora da realização da prova).

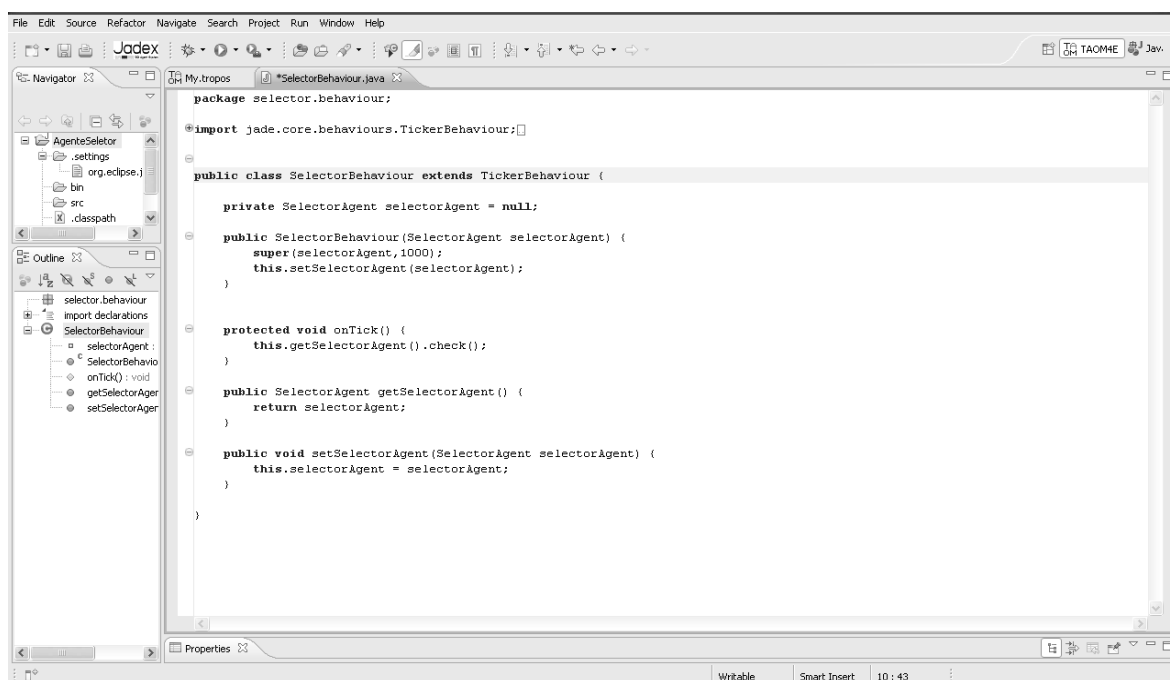


Figura 36 – Trecho do Código sobre o Comportamento do Agente Seletor.

### 6.3 Considerações Finais

Neste capítulo foi descrito a implementação elaborada com o objetivo de concretizar a modelagem proposta no Capítulo 5.

Assim, delineou-se o funcionamento do Virtual-TANEB, tendo como passo inicial o processo de cadastramento dos dados pessoais dos usuários e os itens a serem utilizados. Por fim, foi explanado o processo de elaboração da prova a partir do uso de um agente de software, chamado de “Agente Seletor”.

Em geral, o desenvolvimento fluiu de maneira normal, com pequenos problemas relacionados à familiaridade e prática em unir o JSP e o agente implementado, para que este sistema seja empregado como uma peça fundamental no suporte a avaliação da aprendizagem no ensino fundamental.

## 7 CONCLUSÃO

Nas últimas décadas, é crescente a demanda pelo desenvolvimento de software conciliando fatores relacionados à educação e a Inteligência Artificial (IA). Nesse sentido, é de fundamental importância implementar sistemas que trabalhem com a identificação de problemas no ensino. Assim, as instituições de ensino, tanto no âmbito público como privado, trarão soluções que refletirá na qualidade do ensino brasileiro.

Nesse contexto, os software que trazem recursos da IA, como os agentes de software, tornam-se bastantes eficientes. Isso porque os sistemas multiagentes são excelentes para solucionar problemas complexos, ou seja, os agentes sendo uma abstração de software se tornaram peça fundamental na compreensão da complexidade dos problemas atuais.

No presente trabalho, ao se propor a modelagem e o desenvolvimento do agente seletor, buscou-se pegar as vantagens da IA (características dos agentes de software), revertendo-as em favor da avaliação da aprendizagem sobre os alunos da quarta série do ensino fundamental.

O trabalho aqui apresentado foi objeto de publicações no XX Simpósio Brasileiro de Informática na Educação (SBIE 2009) realizado na cidade de Florianópolis-SC, e, também foi aceito para publicação na XI International Conference on Engineering and Technology Education (INTERTECH 2010) a ser realizada entre os dias 7 a 10 de março de 2010 na cidade de Ilhéus-BA.

### 7.1 Resultados Alcançados

O presente trabalho, além de ter permitido, um plausível entendimento sobre os temas que lhe embasam (Capítulos 2, 3 e 4), apresentou como resultados especificamente relacionados com sua proposta os seguintes:

- Revisão dos sistemas que fazem a avaliação da educação básica brasileira;
- Revisão das teorias de medidas (TRI e TCM) usadas no processo de avaliação de um determinado indivíduo;
- Revisão da Engenharia dos Sistemas Multiagentes, que se baseia em agentes para o desenvolvimento de software educacional;

- Elaboração da modelagem do Virtual-TANEB (Capítulo 5), que se baseia na UML e no Processo Unificado para o desenvolvimento dos diagramas que refletem as funcionalidades do sistema VT;
- Implementação de uma parte inicial do Projeto Virtual-TANEB (VT) para avaliar o nível de conhecimento dos alunos sobre um determinado assunto, usando conceitos da IA (agentes de software) para solucionar esta abordagem sugerida (Capítulo 6).

## **7.2 Principais Contribuições**

As principais características distintas do Projeto Virtual-TANEB em relação à maneira como é feita a prova atualmente pelo SAEB são:

- Estabelecer mecanismos Computacionais utilizando Inteligência Artificial para melhorar e aperfeiçoar a avaliação do nível de conhecimento dos alunos em relação à matemática da quarta série do ensino fundamental;
- Automatizar o processo de avaliação da educação básica;
- Reduzir o tempo de resposta atual para a execução, análise e resultado do diagnóstico da prova do SAEB;
- Permitir de forma paralela e distribuída a aplicação da prova a várias escolas, através de um portal na Internet.

## **7.3 Perspectivas Futuras**

No presente trabalho, empregou-se somente a abordagem dos agentes de software no processo de selecionar itens para formular uma prova. Nesse contexto, como proposta para trabalhos futuros, sugerimos:

- Expansão da modelagem do Virtual-TANEB acerca do processo de “Realizar a Prova”;
- Expansão da Implementação do Virtual-TANEB permitindo que o pedagogo faça a avaliação do item a ser cadastrado de forma automática, cuja avaliação atualmente é feita manualmente;
- Implementação de um Agente que faça o monitoramento de todas as ações dos alunos no momento da execução da prova;

- Implementação de um software que automatize a Teoria da Resposta ao Item (TRI);
- Implementação de um Agente que tenha como responsabilidade a entrega de sugestões (dicas) aos alunos no momento de dificuldade na resolução de um item;
- Implementar a interação os agentes “Monitor” e “Seletor”, permitindo o Virtual-TANEB se adaptar ao nível de conhecimento dos alunos.



## REFERÊNCIAS

- [1] ALEXANDRE, J. W. C., ANDRADE, D. F., VASCONCELOS, A. P. e ARAUJO, A. M. S. **Uma Proposta de Análise de um Construto para Medição dos Fatores Críticos da Gestão pela Qualidade por Intermédio da Teoria da Resposta ao Item**. Fortaleza – CE, 2002.
- [2] ALMEIDA, Gustavo O. **Utilização da Abordagem de Sistemas Multiagentes no Problema do Controle do Espaço Aéreo**. Universidade Federal de Pelotas – RS, 2007.
- [3] ANDRADE, D. F., TAVARES, H. R. e VALLE, R. C. **Teoria da Resposta ao Item: Conceitos e Aplicações**. In SINAPE, São Paulo – SP, 2000.
- [4] ANDRADE, D. F., JUSTINO, G. **Software para Avaliação de Aprendizagem Utilizando a Teoria da Resposta ao Item**. In XIII Workshop sobre Informática na Escola. Rio de Janeiro – RJ, 2007
- [5] BOOCH, G., RUMBAUGH, J. e JACOBSON, I. **UML: Guia do Usuário**. Editora Campus. Rio de Janeiro - RJ, 2000.
- [6] BRASIL. Ministério da Educação. **PDE: Plano de Desenvolvimento da Educação: Prova Brasil: ensino fundamental: matrizes de referência, tópicos e descritores**. Brasília: MEC, SEB; Inep, 2008.193 p.: il.
- [7] BRASIL, Ministério da Educação. **Primeiros Resultados: Médias de Desempenho do SAEB/2005 em Perspectiva Comparada, 2007**. Disponível em:[http://www.inep.gov.br/download/saeb/2005/SAEB1995\\_2005.pdf](http://www.inep.gov.br/download/saeb/2005/SAEB1995_2005.pdf). Acessado em: 15 ago. 2009.

- [8] BRENNER, W., RÜDIGERR, Z., WITTIG, H. **Intelligent Software Agents – Foundations and Applications**. Berlin: Springer-Verlag, 1998.
- [9] CASTRO, Maria Helena Guimarães de. **Avaliação do Sistema Educacional Brasileiro: Tendências e Perspectivas**. Brasília: Instituto Nacional de Estudos e Pesquisas Educacionais, 1998.
- [10] COELHO, Maria Inês de Matos. **Vinte Anos de Avaliação da Educação Básica no Brasil: Aprendizagens e Desafios**. Avaliações Políticas Públicas Educacionais, Rio de Janeiro, v. 16, n. 59, p. 229-258, abr./jun. 2008.
- [11] CONDÉ, Frederico Neves. **O Efeito dos Modelos de Testes na Estimativa da Habilidade dos Estudantes: Comparação entre Prova Brasil e SAEB 2005**. Disponível em: <http://www.psi-ambiental.net/pdfjornada/D04FredericoEfeito.pdf>, Brasília-DF, 2007. Acessado em: 27 ago. 2009.
- [12] COSENTINO, M., POTTS, C.. **A CASE tool supported methodology for the design of multi-agent systems**. The 2002 International Conference on Software Engineering Research and Practice (SERP'02). Las Vegas, NV, USA. June 24 - 27, 2002.
- [13] DELOACH, S. **Analysis and Design using MaSE and AgentTool**, Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001). Miami University, Oxford, Ohio, March 31 - April 1, 2001.
- [14] DELMANTO, Dileta. et al. **Prova Brasil na Escola: Material para Professores, Coordenadores Pedagógicos e Diretores de Escolas de Ensino Fundamental**. Disponível em: [http://avaliacoes.educacao.sp.gov.br/prova%20brasil\\_FINAL.pdf](http://avaliacoes.educacao.sp.gov.br/prova%20brasil_FINAL.pdf), São Paulo-SP, 2007. Acessado em: 27 ago. 2009.

- [15] FERNANDES, F. M. O., FERREIRA, E. B. e OLIVEIRA, M. S. **Análise da Prova de Português para o Curso de Agronomia Através da Teoria da Resposta ao Item.** In: terceira RBRAS. Lavras – MG, 2008
- [16] FIALHO, Hugo Sampaio. **WORKFLAGE: Uma Ferramenta de Definição e Criação de Workflow Baseado em Agentes de Software para Weicot.** Dissertação (Mestrado em Engenharia de Eletricidade – Área de Ciência da Computação), Universidade Federal do Maranhão (CPGEE/UFMA). São Luís-MA. 2003.
- [17] FROZZA, Rejane., KONZEN, Andréa., MAINIERI, Alessandra., SCHREIBER, Jacques., MOLZ, Kurt., TAUTZ, Jorge., PEDÓ, Ricardo., DRESCH, Jonas. **Agentes Tutor e Companheiro em um Ambiente Educacional baseado em Estilos Cognitivos.** WESAAC 2007 – Workshop – Escola de Sistemas de Agentes para Ambientes Colaborativos. Pelotas-RS, 2007.
- [18] GUEDES, Gilleanes T. A. **UML: Uma Abordagem Prática.** terceira ed. Novatec Editora. São Paulo – SP, 2008.
- [19] GIUNCHIGLIA, Fausto, MYLOPOULOS, John, PERINI, Anna. **The Tropos Software Development Methodology: Processes, Models and Diagrams.** Proceedings of Third International Workshop on Agent-oriented Software Engineering – AOSE, 2002.
- [20] HAYES, C. C. **Agents in A Nutshell – A Very Brief Introduction.** IEEE Transactions on Knowledge and Data Engineering, vol 11, no. 1, Jan./Fev., 1999.

- [21] INEP. Ministério da Educação. **PISA 2000 – Relatório Nacional**. Disponível em: <http://www.inep.gov.br/download/internacional/PISA/PISA2000.pdf>, Brasília-DF, 2001. Acessado em: 25 ago. 2009.
- [22] INEP. Ministério da Educação. **Exame Nacional do Ensino Médio (Enem): Fundamentação Teórico – Metodológica**. Disponível em: [http://www.publicacoes.inep.gov.br/arquivos/%7B9330753A-7FCB-43EC-B99F-2FB657407BE7%7D\\_ENEM\\_para%20uma%20discuss%C3%A3o%20te%C3%B3rica.pdf](http://www.publicacoes.inep.gov.br/arquivos/%7B9330753A-7FCB-43EC-B99F-2FB657407BE7%7D_ENEM_para%20uma%20discuss%C3%A3o%20te%C3%B3rica.pdf), Brasília-DF, 2005. Acessado em: 25 ago. 2009.
- [23] INEP. Ministério da Educação. **ENCCEJA**. Disponível em: [http://www.cereja.org.br/arquivos\\_upload/encceja2008.pdf](http://www.cereja.org.br/arquivos_upload/encceja2008.pdf), Brasília-DF, 2008. Acessado em: 27 ago. 2009.
- [24] LINDOSO, Alisson Neres. **Uma Metodologia Baseada em Ontologias para a Engenharia de Aplicações Multiagente**. Dissertação (Mestrado em Engenharia de Eletricidade – Área de Ciência da Computação), Universidade Federal do Maranhão (CPGEE/UFMA). São Luís-MA, 2006.
- [25] MAYNART, Marlon Cavalcante. **As Mudanças do Exame Nacional do Ensino Médio (Enem) 2009 e Um Novo Sistema de Seleção Unificada**. Disponível em: [http://www.deleste3.com.br/comunicado/Enem\\_2009.pdf](http://www.deleste3.com.br/comunicado/Enem_2009.pdf), São Paulo-SP, 2009. Acessado em: 25 ago. 2009.
- [26] MASSAE, L. **Propostas e Programas de Avaliação da Educação Básica**. In: Fórum Permanente de Desafios do Magistério. Campinas – SP, 2006.
- [27] MELLO, Julio., FROZZA, Rejane. **Fred – Um Agente Pedagógico Mediador na Construção do Conhecimento**. WESAAC 2007 – Workshop – Escola de Sistemas de Agentes para Ambientes Colaborativos. Pelotas-RS, 2007.

- [28] MYLOPOULOS J., Castro, J., Kolp M.. **Tropos: Towards Agent-Oriented Information Systems Engineering**. Position Paper at the Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS 2000). Stockolm, Sweden. June 5-6, 2000.
- [29] MISLEVY, R. J. and BOCK, R. D., **BILOG 3: Item Analysis and TestScoring with Binary Logistic Models**. Chicago: Scientific Software, 1990.
- [30] OMG – THE OBJECT MANAGEMENT GROUP – **Agent Platform Special Interest Group**. Agent Technology – Green Paper. Version 1.0. Disponível em: [www.omg.org](http://www.omg.org), 2000. Acessado em jan. 2007.
- [31] PASQUALI, L. e PRIMI, R. **Fundamentos da Teoria da Resposta ao Item: TRI**, Disponível em: [http://pepsic.bvspsi.org.br/scielo.php?script=sci\\_arttext&pid=S16770471200300200002&lng=es&nrm=iso](http://pepsic.bvspsi.org.br/scielo.php?script=sci_arttext&pid=S16770471200300200002&lng=es&nrm=iso), Brasília-DF, 2003. Acessado em: 27 ago. 2009.
- [32] PARANDEKAR, S., AMORIM, E. e WELSH, A., **Prova Brasil – Construindo um Arcabouço para Mensurar o Progresso nos Resultados da Aprendizagem**, 2008. Disponível em: [http://siteresources.worldbank.org/INTENBREVE/Newsletters/21809380/Mar08\\_121\\_BR\\_PROVA\\_PR.pdf](http://siteresources.worldbank.org/INTENBREVE/Newsletters/21809380/Mar08_121_BR_PROVA_PR.pdf). Acessado em: 15 ago. 2009.
- [33] PEROTTO, Marcelo Antonio. **Estudo de uma Metodologia Orientada a Agentes: Um Protótipo para um Ambiente Virtual**. Dissertação (Mestrado em Engenharia de Produção), Universidade Federal de Santa Catarina (UFSC). Florianópolis - SC. 2002.
- [34] RABELLO, Roberto dos Santos., REATEGUI, Eliseo., COSTI, Júlio César G. **Um Agente de Interface Aplicado ao Ensino de Equações de Primeiro**

**Grau.** X Ciclo de Palestras sobre Novas Tecnologias na Educação. Porto Alegre-RS, 2007.

- [35] RUSSELL, Stuart. e NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. Prentice-Hall, Saddle River, NJ, 1995.
- [36] SILVA, Leonardo Ayres de Moraes. **Estudo e Desenvolvimento de Sistemas Multiagentes usando JADE: Java Agent Development framework**. Fortaleza, 2003.
- [37] SILVA, Mauro Lopes Carvalho. **Modelo de IDS Remoto Baseado na Tecnologia de Agentes, Web Services e MDA**. Dissertação (Mestrado em Engenharia de Eletricidade – Área de Ciência da Computação), Universidade Federal do Maranhão (CPGEE/UFMA). São Luís-MA. 2006.
- [38] SOARES, José Francisco e ALVES, Maria Teresa Gonzaga. **Desigualdades Raciais no Sistema Brasileiro de Educação Básica**. Educ. Pesqui. (online). 2003, vol.29, n.1, pp. 147-165. ISSN 1517-9702. doi: 10.1590/S1517-97022003000100011.
- [39] SOUZA, F. M. e SAMPAIO, E. M. **SAEB: Inclusão pela Qualidade, 2003**. Disponível em: <http://www.tvebrasil.com.br/salto/boletins2003/saeb/tetxt2.htm>. Acessado em: 15 ago. 2009.
- [40] TEZZA, Rafael e BORNIA, Antonio Cesar. **Teoria da Resposta ao Item: Vantagens e Oportunidades para a Engenharia de Produção**. In. XXIX ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. Salvador-BA, 2009.
- [41] TIVES, Heloise Acco., SOUZA, Igor Flávio., SILVA, Kátia Gomes., OLIVEIRA, Fernando Luiz. **Proposta de Desenvolvimento de um Agente Inteligente**

**Móvel a ser Utilizado na Rede de Computadores do CEULP/ULBRA.**  
Palmas – To, 2007.

- [42] THURSTONE, L. L. **A law of comparative judgments.** Psychological Review, v.34, p.278–286, 1928.
- [43] WILSON, D. T., WOOD, R., DOWNS, P. K. and GIBBONS, R. **TESTFACT: Test Scoring, Item Statistics and Item Factor Analysis.** Chicago: Scientific Software, 1991.
- [44] WOOLDRIDGE, M., JENNINGS, N.. **Intelligent Agents: Theory and Practice.** The Knowledge Engineering Review, v. 10 (2), p. 115-152, 1995.
- [45] ZAMBONELLI, Franco, et al. **Developing Multiagent systems: The Gaia Methodology.** ACM Transactions on Software Engineering and Methodology, 12(3), pages 417-470. 2003.
- [46] ZIMOWSKI, M. F., MURAKI, E., MISLEVY, R. J. and BOCK, R.D. **BILOG-MG: Multiple-Group IRT Analysis and Test Maintenance for Binary Items.** Chicago: Scientific Software, 1996, Inc.

## APÉNDICE



#####  
TABELAS DO BANCO DE DADOS DO VIRTUAL-TANEB  
#####



```
#####
CLASSE CADASTRAR NO VIRTUAL-TANEB
#####
```

```
package virtualTaneb.usuario;
import virtualTaneb.*;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class cadastrarDAO {

    public void cadastrarFuncao(funcao func){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO funcao "
                + "VALUES(" + func.getIdfuncao() + ", '"
                + func.getDescricao() + "')");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
                + "de dados!!!\nNÃo foi possÃvel cadastrar funcao!",
            "ERRO",
                JOptionPane.ERROR_MESSAGE);
        } finally {
            conexao.desconectar();
        }
    } // FIM CADASTRAR DADOS

    public void cadastrarCargo(cargo cgo){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO cargo "
                + "VALUES(" + cgo.getIdcargo() + ", '"
                + cgo.getDescricao() + "')");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
                + "de dados!!!\nNÃo foi possÃvel cadastrar cargo!",
            "ERRO",
                JOptionPane.ERROR_MESSAGE);
        } finally {
            conexao.desconectar();
        }
    } // FIM CADASTRAR DADOS

    public void cadastrarEndereco(endereco end){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO endereco "
                + "VALUES(" + end.getIdendereco() + ", "
                + end.getIdmunicipio() + ", '"
                + end.getRua() + "', '"
                + end.getNumero() + "', '"
                + end.getCep() + "', '"
                + end.getBairro() + "')");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
                + "de dados!!!\nNÃo foi possÃvel cadastrar endereco!",
            "ERRO",
                JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarMunicipio(municipio mcp) {
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO municipio "
            + "VALUES(" + mcp.getIdmunicipio() + ", "
            + mcp.getIduf() + ", '"
            + mcp.getNome() + "'"");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃO foi possível cadastrar
        municipio!", "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarUsuarios(usuarios usr) {
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO usuario "
            + "VALUES(" + usr.getIdusuario() + ", "
            + usr.getIdendereco() + ", "
            + usr.getIdtipo() + ", "
            + usr.getIdinstituicao() + ", "
            + usr.getIdpais() + ", "
            + usr.getIdtipo_nacionalidade() + ", "
            + usr.getIdcor() + ", "
            + usr.getIdfuncao() + ", "
            + usr.getIdcargo() + ", '"
            + usr.getNome() + "', '"
            + usr.getEmail() + "', '"
            + usr.getData_Nasc() + "', '"
            + usr.getSexo() + "', '"
            + usr.getNome_Mae() + "', '"
            + usr.getIdentidade() + "', '"
            + usr.getOrgao_Emissor_Ident() + "', '"
            + usr.getData_Expedicao_Ident() + "', '"
            + usr.getLogin() + "', '"
            + usr.getSenha() + "'"");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃO foi possível cadastrar usuá;rio!",
        "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS
}

```

```
#####  
CLASSE USUÁRIOS NO VIRTUAL-TANEB  
#####
```

```
package virtualTaneb.usuario;  
import java.util.Date;  
  
public class usuarios {  
  
    private int idusuario;  
    private int idendereco;  
    private int idtipo;  
    private int idinstituicao;  
    private int idpais;  
    private int idtipo_nacionalidade;  
    private int idcor;  
    private int idfuncao;  
    private int idcargo;  
    private String nome;  
    private String email;  
    private Date data_nasc ;  
    private String sexo;  
    private String nome_mae;  
    private String identidade;  
    private String orgao_emissor_ident;  
    private Date data_expedicao_ident;  
    private String login;  
    private String senha;  
  
    //MÉTODOS GETS  
    public int getIdusuario() {  
        return idusuario;  
    }  
    public int getIdendereco() {  
        return idendereco;  
    }  
    public int getIdtipo() {  
        return idtipo;  
    }  
    public int getIdinstituicao() {  
        return idinstituicao;  
    }  
    public int getIdpais() {  
        return idpais;  
    }  
    public int getIdtipo_nacionalidade() {  
        return idtipo_nacionalidade;  
    }  
    public int getIdcor() {  
        return idcor;  
    }  
    public int getIdfuncao() {  
        return idfuncao;  
    }  
    public int getIdcargo() {  
        return idcargo;  
    }  
    public String getNome() {  
        return nome;  
    }  
    public String getEmail() {
```

```
        return email;
    }
    public Date getData_Nasc() {
        return data_nasc;
    }
    public String getSexo() {
        return sexo;
    }
    public String getNome_Mae() {
        return nome_mae;
    }
    public String getIdentidade() {
        return identidade;
    }
    public String getOrgao_Emissor_Ident() {
        return orgao_emissor_ident;
    }
    public Date getData_Expedicao_Ident() {
        return data_expedicao_ident;
    }
    public String getLogin() {
        return login;
    }
    public String getSenha() {
        return senha;
    }
}

//MÉTODOS SETS
public void setIdusuario(int idusuario) {
    this.idusuario = idusuario;
}
public void setIdendereco(int idendereco) {
    this.idendereco = idendereco;
}
public void setIdtipo(int idtipo) {
    this.idtipo = idtipo;
}
public void setIdinstituicao(int idinstituicao) {
    this.idinstituicao = idinstituicao;
}
public void setIdpais(int idpais) {
    this.idpais = idpais;
}
public void setIdtipo_nacionalidade(int idtipo_nacionalidade) {
    this.idtipo_nacionalidade = idtipo_nacionalidade;
}
public void setIdcor(int idcor) {
    this.idcor = idcor;
}
public void setIdfuncao(int idfuncao) {
    this.idfuncao = idfuncao;
}
public void setIdcargo(int idcargo) {
    this.idcargo = idcargo;
}
public void setNome(String nome) {
    this.nome = nome;
}
public void setEmail(String email) {
    this.email = email;
}
}
```

```

public void setData_nasc(Date data_nasc) {
    this.data_nasc = data_nasc;
}
public void setSexo(String sexo) {
    this.sexo = sexo;
}
public void setNome_mae(String nome_mae) {
    this.nome_mae = nome_mae;
}
public void setIdentidade(String identidade) {
    this.identidade = identidade;
}
public void setOrgao_emissor_ident(String orgao_emissor_ident) {
    this.orgao_emissor_ident = orgao_emissor_ident;
}
public void setData_expedicao_ident(Date data_expedicao_ident) {
    this.data_expedicao_ident = data_expedicao_ident;
}
public void setLogin(String login) {
    this.login = login;
}
public void setSenha(String senha) {
    this.senha = senha;
}

public usuarios(){
}

public usuarios(int idusuario){
    this.idusuario = idusuario;
}

public usuarios(int idendereco, int idtipo, int idinstituicao, int
idpais, int idtipo_nacionalidade, int idcor, int idfuncao, int idcargo,
String nome, String email, Date data_nasc, String sexo, String nome_mae,
String identidade, String orgao_emissor_ident, Date data_expedicao_ident,
String login, String senha) {
    this.idendereco = idendereco;
    this.idtipo = idtipo;
    this.idinstituicao = idinstituicao;
    this.idpais = idpais;
    this.idtipo_nacionalidade = idtipo_nacionalidade;
    this.idcor = idcor;
    this.idfuncao = idfuncao;
    this.idcargo = idcargo;
    this.nome = nome;
    this.email = email;
    this.data_nasc = data_nasc;
    this.sexo = sexo;
    this.nome_mae = nome_mae;
    this.identidade = identidade;
    this.orgao_emissor_ident = orgao_emissor_ident;
    this.data_expedicao_ident = data_expedicao_ident;
    this.login = login;
    this.senha = senha;
}
}

```

```
#####
CLASSE CADASTRAR ITENS NO VIRTUAL-TANEB
#####
```

```
package virtualTaneb.item;
import virtualTaneb.conexao;
import java.sql.SQLException;
import javax.swing.JOptionPane;

// CADASTRAR ITEM
public class cadastrarDAO {

    public void cadastrarDisciplina(disciplina disc){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO disciplina "
                + "VALUES(" + disc.getIddisciplina() + ", "
                + disc.getNome() + "'");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
            + "de dados!!!\nNÃfo foi possÃvel cadastrar
            disciplina!", "ERRO",
            JOptionPane.ERROR_MESSAGE);
        } finally {
            conexao.desconectar();
        }
    } // FIM CADASTRAR DADOS

    public void cadastrarAssunto(assunto ass){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO assunto "
                + "VALUES(" + ass.getIdassunto() + ", "
                + ass.getIddisciplina() + ", "
                + ass.getDescricao() + "'");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
            + "de dados!!!\nNÃfo foi possÃvel cadastrar assunto!",
            "ERRO",
            JOptionPane.ERROR_MESSAGE);
        } finally {
            conexao.desconectar();
        }
    } // FIM CADASTRAR DADOS

    public void cadastrarDescritor(descritor desc){
        try {
            conexao.conectar();
            conexao.st.execute("INSERT INTO descritor "
                + "VALUES(" + desc.getIddescritor() + ", "
                + desc.getIdtema() + ", "
                + desc.getNome() + "', "'
                + desc.getSigla() + "'");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
            acesso ao banco "
            + "de dados!!!\nNÃfo foi possÃvel cadastrar
            descritor!", "ERRO",
```



```

        JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarTema(tema tem){
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO tema "
            + "VALUES(" + tem.getIdtema() + ", "
            + tem.getIdassunto() + ", '"
            + tem.getDescricao() + "'"");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃo foi possÃvel cadastrar tema!",
        "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarAlternativas(alternativas alt){
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO alternativas "
            + "VALUES(" + alt.getIdalternativas() + ",
            "
            + alt.getAlternativa1() + "', '"
            + alt.getImagem1() + "', '"
            + alt.getAlternativa2() + "', '"
            + alt.getImagem2() + "', '"
            + alt.getAlternativa3() + "', '"
            + alt.getImagem3() + "', '"
            + alt.getAlternativa4() + "', '"
            + alt.getImagem4() + "', "
            + alt.getAlternativa_correta() + "'"");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃo foi possÃvel cadastrar
        alternativas!", "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarNivel(nivel niv){
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO nivel "
            + "VALUES(" + niv.getIdnivel() + ", '"
            + niv.getDescricao() + "'"");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃo foi possÃvel cadastrar nivel!",
        "ERRO",

```

```

        JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarPergunta( pergunta perg ){
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO pergunta "
            + "VALUES(" + perg.getIdpergunta() + ", '"
            + perg.getDescricao() + "', '"
            + perg.getImagem() + "'" );
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃo foi possÃvel cadastrar pergunta!",
        "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS

public void cadastrarItem(item item){
    try {
        conexao.conectar();
        conexao.st.execute("INSERT INTO item "
            + "VALUES(" + item.getIditem() + ", "
            + item.getIdnivel() + ", "
            + item.getIddescriptor() + ", "
            + item.getIdpergunta() + ", "
            + item.getIdalternativas() + ")" );
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e.getMessage() + "\nErro no
        acesso ao banco "
            + "de dados!!!\nNÃo foi possÃvel cadastrar item!",
        "ERRO",
            JOptionPane.ERROR_MESSAGE);
    } finally {
        conexao.desconectar();
    }
} // FIM CADASTRAR DADOS
}

```

```

#####
CLASSE PRINCIPAL DO AGENTE SELETOR
#####

```

```

/*
 * Main.java
 *
 * Created on 15 de Outubro de 2009, 18:16
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

```

```

package selector;

```

```

public class Main {

    /** Creates a new instance of Main */
    public Main() {
    } /**
    * @param args the command line arguments
    */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}

#####
CLASSE DO AGENTE SELETOR
#####

package selector.agent;

import jade.core.Agent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Date;
import java.util.Calendar;
import java.util.GregorianCalendar;
import selector.behaviour.SelectorBehaviour;

public class SelectorAgent extends Agent {

    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DATABASE_URL = "jdbc:mysql://localhost/selector";
    static final String USERNAME = "root";
    static final String PASSWORD = "";
    private Connection connection = null;

    @Override
    protected void setup() {
        this.connect();
        this.addBehaviour(new SelectorBehaviour(this));
    }

    @Override
    protected void takeDown() {
        this.disconnect();
    }

    protected void connect() {
        try {
            Class.forName(JDBC_DRIVER);
            connection = DriverManager.getConnection(DATABASE_URL, USERNAME, PASSWORD);
        }
    }
}

```

```

    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}
protected void disconnect() {
    try {
        connection.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public void check() {
    try {
        PreparedStatement statement = connection.prepareStatement("SELECT
`idprova`,`data`,`hora_inicio` FROM prova");
        ResultSet resultSet = statement.executeQuery();
        String currentDate = this.getDate(this.getCurrentDateTime());
        String currentTime = this.getTime(this.getCurrentDateTime());
        while (resultSet.next()) {
            int id = resultSet.getInt("idprova");
            String date = this.getDate(new Date(resultSet.getDate("data").getTime()));
            String time = this.getTime(new Date(resultSet.getTime("hora_inicio").getTime()));
            if (date.equals(currentDate) && time.equals(currentTime)) {
                System.out.println("Criando nova prova...");
                this.create(id);
            }
        }
        System.out.println(currentDate + " " + currentTime);
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public int create(int idprova) {
    try {
        PreparedStatement statement = connection.prepareStatement("SELECT `iditem` FROM
`selector`.`item` ORDER BY RAND() LIMIT 5");
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            int iditem = resultSet.getInt("iditem");
            System.out.println(iditem);
            statement = connection.prepareStatement("INSERT INTO `selector`.`item_prova`
(`iditem_prova`,`iditem`,`idprova`) VALUES (NULL,?,?)");
            statement.setInt(1, iditem);
            statement.setInt(2, idprova);
            statement.execute();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

```

```
    }  
    return 0;}  
    protected Date getCurrentDateTime() {  
        return new GregorianCalendar().getTime();  
    }  
  
    protected String getTime(Date date) {  
        GregorianCalendar calendar = new GregorianCalendar();  
        calendar.setTime(date);  
        int hour = calendar.get(Calendar.HOUR_OF_DAY);  
        int minute = calendar.get(Calendar.MINUTE);  
        int second = calendar.get(Calendar.SECOND);  
        return String.format("%02d:%02d:%02d", hour, minute, second);  
    }  
  
    protected String getDate(Date date) {  
        GregorianCalendar calendar = new GregorianCalendar();  
        calendar.setTime(date);  
        int day = calendar.get(Calendar.DAY_OF_MONTH);  
        int month = calendar.get(Calendar.MONTH);  
        int year = calendar.get(Calendar.YEAR);  
        return String.format("%02d-%02d-%04d", day, month, year);  
    }  
}
```

```
#####  
CLASSE DO COMPORTAMENTO DO AGENTE SELETOR  
#####
```

```
package selector.behaviour;  
  
import jade.core.behaviours.TickerBehaviour;  
import java.util.Calendar;  
import java.util.GregorianCalendar;  
import javax.crypto.NullCipher;  
import selector.agent.SelectorAgent;  
  
public class SelectorBehaviour extends TickerBehaviour {  
  
    private SelectorAgent selectorAgent = null;  
  
    public SelectorBehaviour(SelectorAgent selectorAgent) {  
        super(selectorAgent,1000);  
        this.setSelectorAgent(selectorAgent);  
    }  
  
    protected void onTick() {  
        this.getSelectorAgent().check();  
    }  
  
    public SelectorAgent getSelectorAgent() {  
        return selectorAgent;  
    }  
  
    public void setSelectorAgent(SelectorAgent selectorAgent) {  
        this.selectorAgent = selectorAgent;  
    }  
}
```