



**UNIVERSIDADE FEDERAL RURAL DO SEMIÁRIDO  
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE  
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**



**Igor Saraiva Brasil**

**ATEP: Um Jogo 3D Persistente Baseado em Navegador para  
Apoiar a Aprendizagem Baseada em Projetos de Operadores de  
Sondas de Perfuração de Poços de Petróleo Terrestres**

**MOSSORÓ – RN**

**2012**

**Igor Saraiva Brasil**

**ATEP: Um Jogo 3D Persistente Baseado em Navegador para  
Apoiar a Aprendizagem Baseada em Projetos de Operadores de  
Sondas de Perfuração de Poços de Petróleo Terrestres**

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

Orientador: Prof. Dr. Francisco Milton Mendes Neto – UFRSA.

**MOSSORÓ – RN**

**2012**

**Igor Saraiva Brasil**

**Um Jogo 3D Persistente Baseado em Navegador para Apoiar a  
Aprendizagem Baseada em Projetos de Operadores de Sondas de  
Perfuração de Poços de Petróleo Terrestres**

Dissertação apresentada ao Mestrado em Ciência  
da Computação para a obtenção do título de Mestre  
em Ciência da Computação.

APROVADA EM: 28/03/2012

BANCA EXAMINADORA

  
\_\_\_\_\_  
Prof. Dr. Francisco Milton Mendes Neto – UFERSA

Presidente

  
\_\_\_\_\_  
Prof. Dr. Pedro Fernandes Ribeiro Neto - UERN

Primeiro Membro

  
\_\_\_\_\_  
Prof. Dr. Jefferson Teixeira de Souza – UECE

Segundo Membro

## **DEDICATÓRIA**

Dedico esse trabalho a minha esposa Janaina e aos meus pais Netinha e Hugo Brasil.

## **AGRADECIMENTOS**

À Deus por permitir que eu tenha saúde e paz para lutar pelos meus objetivos.

Agradeço ao apoio de minha família, aos meus pais Netinha e Hugo Brasil, pelo carinho, dedicação e incentivo dados a mim em todos os momentos de minha vida. Em especial, agradeço a minha esposa Janaina pela paciência e suporte nesse período de mestrado.

Ao meu Orientador, Milton Mendes, meus agradecimentos pela orientação na realização deste trabalho.

À CAPES pelo apoio financeiro através de bolsa no período que precisei e à UFRSA, na pessoa de Kleber Jacinto, por apoiar a qualificação e formação dos servidores que lidera.

Aos meus colegas da SUTIC, que estiveram junto comigo nesta jornada, pela compreensão por algumas ausências, a eles meus sinceros agradecimentos.

E por fim, a todos aqueles que direta ou indiretamente colaboraram para que este projeto fosse concluído.

## RESUMO

O processo de extração de petróleo apresenta muitos perigos que podem ameaçar o meio ambiente e a vida dos trabalhadores. A fim de reduzir as chances de acidentes e falhas, é altamente recomendada uma formação continuada e eficaz para os trabalhadores, que ajude a mitigar erros durante a operação de uma sonda. Devido, principalmente, à distribuição geográfica do público alvo a ser treinado, à falta tempo disponível e aos riscos de participar de um treinamento prático presencial, as ferramentas de Educação a Distância (EaD) entram nesse contexto como uma ótima opção. No entanto, ferramentas de EaD convencionais não possuem os atributos suficientes para atender ao objetivo proposto, principalmente devido à necessidade de se exercitar o trabalho prático. Este trabalho apresenta um jogo que simula uma sonda de perfuração. Neste jogo, os jogadores são expostos a várias circunstâncias em que podem exercitar seus conhecimentos sobre seu trabalho e aprender a reagir a situações adversas. A fim de realizar essas funções serão aplicadas técnicas de inteligência artificial. Este jogo sério também inclui comunicação instantânea entre os jogadores, simulando o que acontece no ambiente real, em um ambiente virtual 3D multijogador. O jogo proposto é um ambiente baseado na Web, o que permite aos jogadores executá-lo remotamente em diferentes plataformas.

Palavras-Chave: Aprendizagem Baseada em Projetos, Agentes, Aprendizagem Colaborativa, Jogos Educativos.

## **ABSTRACT**

The oil extraction is not an easy process. It poses many hazards which may threaten the environment and the lives of workers at an oil well. In order to reduce the chances of accidents and failures, a continued and effective training to the workers is highly recommended, which can assist in mitigate errors during probe operation. Mainly due to the geographical distribution of audience to be trained, the lack of available time and the risks of participating in a practical training, the Distance Education tools emerge in this context as a great option. However, these conventional tools do not possess the sufficient attributes reach the proposed goal, mainly due to the need of exercise the practical work.. This work presents a game which simulates a drilling rig. In this game, the players are exposed to various circumstances in which they can exercise their knowledge about their work and learn what to do in problematic events. Artificial Intelligence techniques will be applied to the game in order to accomplish that feature. This serious game also includes instant interactions among several players in a virtual 3D environment. The proposed game is a Web-based environment, which allows players to remotely execute it on different platforms.

**Keywords:** Project-Based Learning, Agents, Collaborative Learning, Educative Games.

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Avaliação das metodologias baseada nos conceitos. Adaptado de (OLIVEIRA, 2010). ..... | 41 |
| Tabela 2 – Tabelas relacionadas à persistência do jogo.....                                      | 67 |
| Tabela 3 – Descrição das tabelas do ambiente Web do ATEP.....                                    | 69 |
| Tabela 4 – Modelo textual do agente Sondador. ....   | 84 |
| Tabela 5 – Modelo textual do agente Plataformista.....   | 85 |
| Tabela 6 – Modelo textual do agente Torrista.....  | 86 |
| Tabela 7 – Modelo textual do agente Coordenador.....   | 87 |
| Tabela 8 – Modelo textual do agente Cliente.....   | 88 |
| Tabela 9 – Modelo textual do agente Plataforma. ....   | 91 |
| Tabela 10 – Modelo textual do agente Poço. ....  | 92 |

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 – DrillSystem 3D (DRILLING, 2011).....   | 14 |
| Figura 2 – 3I. Adaptado de (LI, YUE e JAUREGUI, 2009).....                                | 18 |
| Figura 3 – Agente Reativo Simples. ....   | 44 |
| Figura 4 – Estrutura do Agente Reativo Baseado em Modelo. ....                            | 44 |
| Figura 5 – Funções de Pertinência. ....   | 45 |
| Figura 6 – Funcionamento da representação <i>Fuzzy</i> . ....                             | 47 |
| Figura 7 – Sonda de Perfuração Terrestre (ALBERTO, ALBERTO e VEROTTI, 2004). ....         | 48 |
| Figura 8 – Imagem da Plataforma construída no jPCT.....                                   | 53 |
| Figura 9 – Área de trabalho do Blender. ....  | 54 |
| Figura 10 – Detalhes da modelagem da sonda. ....  | 55 |
| Figura 11 – Redenrização da plataforma de Perfuração. ....                                | 56 |
| Figura 12 – Interface de desenvolvimento JADE.....  | 57 |
| Figura 13 – Ciclo de Vida do Agente. Adaptado de (Bellifemine, <i>et al.</i> , 2007)..... | 58 |
| Figura 14 – Tela Inicial do Ambiente do Jogo. ....  | 59 |
| Figura 15 – Arquitetura Geral do Jogo. ....   | 60 |
| Figura 16 – Arquitetura do H-N2N.....   | 62 |
| Figura 17 - Trecho de código-fonte do envio de uma mensagem no ambiente multijogador..    | 63 |
| Figura 18 – Eixo Z no plano cartesiano. ....  | 64 |
| Figura 19 – Trecho do código-fonte para persistir as posições dos objetos.....            | 66 |
| Figura 20 – Trecho do código-fonte para posicionar os objetos persistidos. ....           | 66 |
| Figura 21 – Modelo E/R da persistência do jogo. ....                                      | 67 |
| Figura 22 – Modelo E/R dos dados do ambiente Web do ATEP. ....                            | 68 |
| Figura 23 – Formulário do Relatório Final de Avaliação. ....                              | 73 |
| Figura 24 – Área para comunicação instantânea no ATEP.....                                | 74 |
| Figura 25 – Escolha de tubos para a operação.....   | 76 |
| Figura 26 – Cunha e Chaves Flutuantes em destaque na plataforma. ....                     | 77 |
| Figura 27 – Tela de resolução de situação adversa.....                                    | 78 |
| Figura 28 – Detalhes da jobabilidade do ATEP. ....  | 80 |
| Figura 29 – Diagrama de Tarefas. ....   | 82 |
| Figura 30– Modelo de Recursos.....  | 82 |
| Figura 31– Diagrama de Classes Agentes.....   | 83 |

|  |    |
|--|----|
| Figura 32 – Modelo de Papéis. ....   | 83 |
| Figura 33 – Modelo de Organização dos Agentes. ....                              | 84 |
| Figura 34 – Modelo do Agente Sondador. ....                                      | 87 |
| Figura 35 – Modelo do Agente Cliente. ....                                       | 89 |
| Figura 36 – Trecho de código do registro dos agentes. ....                       | 89 |
| Figura 37 – Trecho de código fonte para consulta ao DF. ....                     | 90 |
| Figura 38 – Trecho de código do registro no H-N2N. ....                          | 91 |
| Figura 39 – Modelo do Agente Plataforma. ....                                    | 93 |
| Figura 40 – Modelo de interação dos agentes utilitários. ....                    | 96 |
| Figura 41 – Modelo de interação entre os agentes cooperativos e de suporte. .... | 97 |

## LISTA DE SIGLAS

|            |  |
|------------|--|
| AML        | <i>Agent Modeling Language</i>                   |
| ATEP       | Ambiente de Treinamento de Equipes de Perfuração |
| EaD        | Educação a Distância                             |
| E-learning | <i>Electronic learning</i>                       |
| IDE        | <i>Integrated Development Environment</i>        |
| PBL        | <i>Project-Based Learning</i>                    |
| UML        | <i>Unified Modeling Language</i>                 |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>1. INTRODUÇÃO.....</b>  | <b>13</b> |
| 1.1. CONTEXTUALIZAÇÃO .....  | 13        |
| 1.2. PROBLEMÁTICA .....  | 14        |
| 1.3. OBJETIVOS.....  | 15        |
| 1.4. RESULTADOS ESPERADOS.....   | 16        |
| 1.5. ORGANIZAÇÃO DA DISSERTAÇÃO.....   | 16        |
| <b>2. APRENDIZAGEM COLABORATIVA BASEADA EM JOGOS .....</b>                               | <b>17</b> |
| 2.1. REALIDADE VIRTUAL NA EDUCAÇÃO.....  | 17        |
| 2.2. JOGOS PERSISTENTES BASEADOS EM NAVEGADOR.....                                       | 19        |
| 2.3. JOGOS EDUCATIVOS .....  | 20        |
| <b>3. TEORIAS DE APRENDIZAGEM .....</b>  | <b>22</b> |
| 3.1. TEORIA DA CARGA COGNITIVA.....  | 22        |
| 3.2. TEORIA SOCIOCULTURAL DE VYGOTSKY .....  | 23        |
| 3.3. TEORIA DA COGNIÇÃO DISTRIBUÍDA.....   | 24        |
| 3.4. TEORIA DA FLEXIBILIDADE COGNITIVA .....   | 24        |
| 3.5. TEORIA DA APRENDIZAGEM BASEADA EM PROBLEMA.....                                     | 25        |
| 3.6. TEORIA DA APRENDIZAGEM BASEADA EM PROJETO.....                                      | 26        |
| 3.7. APRENDIZAGEM BASEADA EM PROJETO <i>VERSUS</i> APRENDIZAGEM BASEADA EM PROBLEMA..... | 28        |
| <b>4. AGENTES E SISTEMAS MULTIAGENTE.....</b>  | <b>30</b> |
| 4.1. AGENTES .....   | 30        |
| 4.2. METODOLOGIAS PARA MODELAGEM DE SISTEMAS MULTIAGENTE.....                            | 32        |
| 4.2.1. TROPOS .....  | 32        |
| 4.2.2. PASSI.....  | 33        |
| 4.2.3. Prometheus .....  | 35        |
| 4.2.4. MESSAGE .....   | 36        |
| 4.2.5. MaSE - <i>Multi-agent Systems Engineering</i> .....                               | 38        |
| 4.2.6. MAS-CommonKADS+.....  | 39        |
| 4.2.7. Análise comparativa entre as metodologias .....                                   | 41        |
| 4.2.8. Linguagens de Modelagem .....   | 43        |
| 4.3. AGENTE REATIVO BASEADO EM MODELO COM LÓGICA <i>FUZZY</i> .....                      | 43        |
| 4.3.1. Conjuntos e Lógica <i>Fuzzy</i> .....   | 44        |
| <b>5. PERFURAÇÃO DE POÇOS DE PETRÓLEO TERRESTRE .....</b>                                | <b>48</b> |
| 5.1. SISTEMAS DE UMA SONDA DE PERFURAÇÃO .....   | 49        |
| <b>6. AMBIENTE DE TREINAMENTO DE EQUIPES DE PERFURAÇÃO.....</b>                          | <b>51</b> |
| 6.1. ESPECIFICAÇÃO DO JOGO.....  | 51        |

|        |  |           |
|--------|--|-----------|
| 6.2.   | FERRAMENTAS UTILIZADAS.....                          | 52        |
| 6.2.1. | Motor Gráfico.....                                   | 53        |
| 6.2.2. | Ferramenta de Modelagem 3D.....                      | 54        |
| 6.2.3. | Plataforma de Execução de Agentes.....               | 56        |
| 6.2.4. | Plataforma de Desenvolvimento <i>Web</i> .....       | 58        |
| 6.2.5. | StarUML.....   | 59        |
| 6.3.   | ARQUITETURA.....                                     | 60        |
| 6.3.1. | Arquitetura do Jogo.....                             | 60        |
| 6.3.2. | Arquitetura do Ambiente Mutijogador.....             | 62        |
| 6.3.3. | Persistência.....                                    | 63        |
| 6.4.   | PORTAL WEB.....                                      | 69        |
| 6.4.1. | Modulo do Instrutor.....                             | 70        |
| 6.4.2. | Módulo do Treinando.....                             | 71        |
| 6.4.3. | Modulo de Administração.....                         | 75        |
| 6.5.   | CENÁRIO E JOGABILIDADE.....                          | 75        |
| 6.5.1. | Pontuação.....                                       | 77        |
| 6.5.2. | Jogabilidade.....                                    | 79        |
| 6.6.   | AGENTES E SISTEMA MULTIAGENTE.....                   | 80        |
| 6.6.1. | Detalhes dos Agentes Cooperativos.....               | 84        |
| 6.6.2. | Detalhes dos Agentes Utilitários.....                | 87        |
| 6.6.1. | Detalhes dos Agentes de Suporte.....                 | 91        |
| 6.6.2. | Modelo de Interação dos Agentes.....                 | 96        |
| 7.     | <b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b> | <b>98</b> |

## 1. INTRODUÇÃO

### 1.1. CONTEXTUALIZAÇÃO

O registro da participação do petróleo na vida do homem remonta a tempos bíblicos. Na antiga Babilônia, os tijolos eram assentados com asfalto e o betume era largamente utilizado pelos fenícios na calafetação<sup>1</sup> de embarcações. Na primeira década do século 21, o petróleo se firmou como a principal fonte de combustível e, com o advento da indústria petroquímica, centenas de novos produtos derivados são produzidos. Portanto, o petróleo passou a ser imprescindível às facilidades e comodidades da vida moderna. Contudo, junto aos benefícios, temos os problemas ocasionados pela exploração de petróleo. Esses, geralmente vinculados a acidentes, têm como resultado vítimas fatais ou danos ao meio ambiente, que, em sua maioria, acarreta graves consequências ambientais (ALBERTO, ALBERTO e VEROTTI, 2004).

A produção de petróleo no Brasil cresceu de cinco mil, na época da criação da Petrobras, até o pico de dois milhões de barris de petróleo por dia (PETROBRAS, 2011). Hoje, apesar das grandes reservas de petróleo serem marítimas e estarem numa camada profunda, chamada pré-sal, ainda há muita atividade de exploração e produção em terra. Para manter esse crescimento, são mobilizadas consideradas somas de recursos. Partes desses recursos são investidos em treinamento de pessoal, devido ao alto risco inerente às inúmeras atividades envolvidas. Por isso a importância de um treinamento contínuo e eficaz, que ajude a mitigar erros e acidentes. Devido, principalmente, à distribuição geográfica do público alvo a ser treinado, à falta tempo disponível e aos riscos de participar de um treinamento prático presencial, as ferramentas de Educação a Distância (EaD) entram nesse contexto como uma ótima opção (QUALHARINI, 2007).

A EaD é uma modalidade de ensino e aprendizagem que tem crescido e apresentado bons resultados. Apesar de seu crescimento, um dos grandes desafios atuais é a evolução do suporte informatizado a esta atividade. No contexto da indústria de petróleo, onde o treinamento necessário exige, constantemente, a verificação, na prática, dos conhecimentos adquiridos e a avaliação da capacidade de reação a situações de riscos não previstas, as

---

<sup>1</sup> Vedar as juntas, buracos ou fendas de uma embarcação, para impedir penetração da água.

ferramentas de EaD convencionais devem ser usadas em conjunto com outras tecnologias (ZUO e XIAO-MING, 2010).

## 1.2. PROBLEMÁTICA

A exploração de petróleo é uma atividade cheia de riscos. Ela requer tarefas perigosas, como perfurar rochas em regiões profundas, enfrentar altas pressões e manipular grandes volumes de gás. Uma das soluções adotadas para reduzir tais riscos é o treinamento dos operadores de sondas de perfuração através de modelos virtuais das sondas (simuladores), nos quais é possível realizar atividades relacionadas à operação de perfuração. Um exemplo desse tipo de modelo virtual é o simulador em 3 dimensões (3D) de uma sonda de perfuração marítima, denominado DrillSystem 3D (DRILLING, 2011), mostrado na Figura 1.



Figura 1 – DrillSystem 3D (DRILLING, 2011).

Através deste simulador é possível aprender sobre operação de perfuração, manutenção, segurança, localização de defeitos, além de outros aspectos.

No entanto, esse tipo de treinamento torna-se oneroso devido à quantidade de pessoas a serem treinadas e pelo treinamento, na maioria das vezes, sequer poder ser realizado no Brasil. Aliado a esse fator, simuladores deste tipo não são específicos para treinamento de operadores de sondas terrestres, o que acaba por tornar o treinamento distante da realidade destas sondas. Outro quesito a ser considerado é que este tipo de simulador não contempla o treinamento colaborativo em equipe (DRILLING, 2011).

Diante disso, operadores de sonda permanecem, durante longo período de tempo, em treinamento até exercerem, de fato, as suas verdadeiras funções. Logo, percebe-se a necessidade de uma solução que contemple os requisitos exigidos nesse tipo de treinamento, que motive o profissional durante a realização do mesmo e que seja economicamente viável. A EaD se apresenta como um boa solução, entretanto, devido a características particulares da exploração de petróleo, as ferramentas de EaD convencionais não atendem os requisitos, pois não contemplam atividades que irão exercitar habilidades como tomada rápida de decisões e reação a situações de riscos.

Outro aspecto fundamental, não considerado em muitos ambientes de aprendizagem, é a fundamentação do processo de aprendizagem em teorias de aprendizagem existentes, não sendo possível comprovar a eficácia do processo, ou seja, não é possível validar o método para aprendizagem (LIU, 2005).

O desenvolvimento de um sistema de EaD para atender os requisitos das atividades de perfuração de poços de petróleo não é uma tarefa trivial, pois o mesmo precisa avaliar na prática o conhecimento do treinando e necessita ter características e situações que simulem uma operação de perfuração real. Além disso, existe a necessidade de treinamento simultâneo de pessoas geograficamente distribuídas e com diferentes papéis na atividade de perfuração. Um outro requisito importante é que a perfuração de petróleo é feita seguindo um processo definido em um projeto, então, simular o aspecto temporal do processo e a colaboração da equipe para atingir os objetivos estipulados no projeto é essencial para o aprendizado do treinando (AHMAD, *et al.*, 2008).

### 1.3. OBJETIVOS

Tendo em vista a problemática apresentada, este trabalho tem como objetivo propor um ambiente colaborativo de aprendizagem para apoiar a Aprendizagem Baseada em Projeto (*Project-Based Learning* - PBL), para treinamento dos operadores de sondas de perfuração de poços de petróleo. O mesmo permitirá, entre outras coisas, que estes profissionais utilizem uma sonda virtualmente, testem habilidades específicas e adquiram os conhecimentos necessários para realização de suas atividades de um modo eficaz. Este tipo de ambiente virtual de ensino-aprendizagem também apresenta um caráter motivador e desafiador para os participantes (instrutores e treinandos), visto que o sistema apresenta características de um

jogo, onde cada participante possui objetivos bem definidos a serem alcançados de acordo com as tarefas reais executadas por eles em uma sonda de perfuração de petróleo terrestre.

#### 1.4. RESULTADOS ESPERADOS

Como resultados deste trabalho, teremos o desenvolvimento de um ambiente de aprendizado colaborativo, baseado em agentes, para apoiar a PBL, seguindo assim os requisitos dessa abordagem. Para validar o trabalho, os conceitos apresentados foram implementados em um jogo para treinamento de operadores de sondas de perfuração de poços de petróleo terrestres. O jogo consiste em: i) um ambiente 3D não imersivo, rodando em um navegador Web; e ii) um portal que dispõe, além de acesso, ferramentas para acompanhamento dos treinandos e emissão de relatórios para os treinandos e instrutores.

#### 1.5. ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho está organizado da seguinte forma: No Capítulo 2 são apresentados os aspectos relacionados à aprendizagem colaborativa baseada em jogos educativos. No Capítulo 3 são apresentadas as principais teorias de aprendizagem. No Capítulo 4 são descritos os conceitos de agentes e sistemas multiagente e apresentadas metodologias de modelagem de sistemas multiagente. No Capítulo 5 são apresentados conceitos da engenharia de petróleo e os sistemas de uma sonda de perfuração. No Capítulo 6 é detalhado o ATEP – Ambiente de Treinamento de Equipes de Perfuração, mostrando a arquitetura e módulos do sistema. Por fim, o Capítulo 7 apresenta as considerações finais e trabalhos futuros.

## 2. APRENDIZAGEM COLABORATIVA BASEADA EM JOGOS

Este capítulo apresenta os aspectos envolvidos na aprendizagem colaborativa baseada em jogos. A Seção 2.1 trata do uso de realidade virtual na educação e suas principais aplicações. A Seção 2.2 define jogos persistentes baseados em navegadores. A Seção 2.3 delinea os aspectos que definem um jogo educativo.

### 2.1. REALIDADE VIRTUAL NA EDUCAÇÃO

Várias são as definições sobre a Realidade Virtual (RV), mas, em geral, refere-se a uma experiência interativa baseada em imagens gráficas 3D geradas em tempo real por computador, ou seja, é uma simulação gerada por computador de um mundo real ou apenas imaginário (BRAGA, 2001).

Latta e Oberg (1994) conceituam a RV como uma interface homem-máquina avançada que simula um ambiente realístico, permitindo que os participantes interajam com ele. Essa interface é considerada como sendo a mais avançada disponível até agora, pois busca levar ao usuário sensações que lhe dão informações sobre o mundo virtual como se ele realmente existisse.

Pimentel e Teixeira (1995) definem RV como o uso de alta tecnologia para convencer o usuário de que ele está em outra realidade, promovendo completamente o seu envolvimento.

Para Li, Yue e Jauregui (2009), RV é uma técnica integrada que envolve computação gráfica, interação humano-computador e Inteligência Artificial (IA). Seu objetivo inclui 3I - Imersão, Interação e Imaginação, como pode ser visto na Figura 2. A RV usa o computador para criar o mundo virtual em 3D.

O avanço tecnológico na área de comunicação e informação ampliou a utilização da RV, possibilitando que outras áreas do conhecimento também se beneficiassem de sua utilização, como, por exemplo (LI, YUE e JAUREGUI, 2009; YANG e WU, 2010):

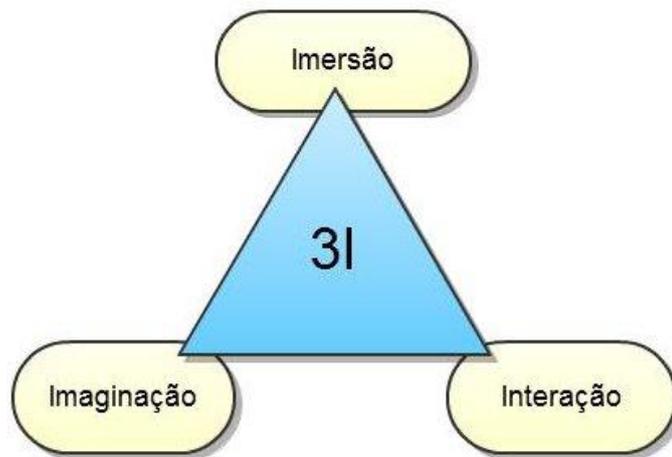


Figura 2 – 3I. Adaptado de (LI, YUE e JAUREGUI, 2009).

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

Li, Yue e Jauregui (2009) defendem que uma das tarefas mais importante de uma ferramenta de EaD é a forma como ela pode atrair a atenção do público. Isto está relacionado diretamente com o efeito de aprendizagem. A fim de atrair mais audiência e aumentar o interesse na aprendizagem, recursos de multimídia, incluindo texto, imagem, vídeo e áudio, são amplamente utilizados neste tipo de ferramenta. No entanto, a tecnologia de EaD, por si só, não pode resolver todos os problemas existentes na educação tradicional, apesar de ter melhorado as técnicas tradicionais de ensino. Alguns conceitos abstratos são ainda muito difíceis para os treinandos compreenderem porque não existem entidades reais para demonstrá-los. Eles apenas compreendem esses conceitos através do significado literal dos conceitos ou da imaginação. Especialmente em alguns temas experimentais da física e da química, os treinandos raramente têm oportunidade de se envolver pessoalmente no experimento pela escassez de recursos, pelo custo ou pelo alto risco que a atividade oferece (YANG e WU, 2010).

As tecnologias de RV compensam algumas deficiências do EaD, podendo simular equipamentos e oferecer um ambiente virtual experimental para os treinandos. Os mesmos podem interagir com o ambiente virtual utilizando apenas dispositivos de entrada simples, como mouse e teclado. Os instrutores, por sua vez, podem usar os objetos virtuais construídos

pela tecnologia de RV para oferecer uma forma estimulante e interessante de aprendizagem para os treinandos (YANG e WU, 2010)

## 2.2. JOGOS PERSISTENTES BASEADOS EM NAVEGADOR

Com a popularização da internet, foram criados os jogos persistentes baseados em navegador (do inglês *Persistent Browser Based Games* - PBBG). Eles podem ser definidos como jogos eletrônicos que podem ser jogados e acessados em um navegador da Web e que apresentam um ambiente virtual compartilhado persistente, onde os eventos continuam a acontecer mesmo na ausência do usuário, podendo este recuperar sua sessão posteriormente (PROJECT, 2011). Eles usam um navegador Web padrão como interface com o jogador e para a exibição do mundo do jogo (KLIMMT, SCHMID e ORTHM, 2009). Este tipo de jogo foca principalmente no raciocínio lógico e, portanto, se comparado com jogos tradicionais, os jogos persistentes objetivam, além do entretenimento, testar habilidades. PBBG são utilizados não apenas como um meio de diversão, mas também para permitir um aprendizado de uma forma consistente, contribuindo para que os jogadores tornem-se usuários assíduos da aplicação (KLIMMT, SCHMID e ORTHM, 2009). Os mesmos podem ser aplicados a uma grande variedade de áreas de aplicação, como, por exemplo, governamental, educacional, militar, de saúde, entre outras (BACKLUND, ENGSTROM e HAMMA, 2007).

Em Project (2011), PBBG são definidos como jogos de computador que satisfaçam os seguintes critérios:

- O jogo deve ser completamente executado na internet usando apenas o navegador;
- Nas múltiplas sessões do jogador, as suas progressões no jogo devem ser completamente armazenadas.

Desse modo, podemos considerar que a definição de um PBBG é bastante ampla, na qual jogos de categorias como *single-player* (jogos de um único participante) e *multi-player* (múltiplos participantes), e de diferentes gêneros - jogos de estratégias (POTISARTRA e KOTRAJARAS, 2010), tiro (YOON, JANG e CHO, 2010), sociais (GUO, TRTTEBERG e WANG, 2010), entre outros - também podem ser englobados (PROJECT, 2011). Porém, não podemos dizer que um PBBG engloba também jogos do tipo MMOG (*Massively Multi-player Online Game*) (PANG, CHEN e YANG, 2009), já que estes podem ser também executados

como uma aplicação instalada localmente, eliminando assim o conceito de portabilidade através do uso do navegador que um PBBG propõe.

Este tipo de jogo apresenta desafios e provê um retorno a respeito do desempenho dos jogadores para motivá-los e informá-los sobre como estão se comportando (NERBRATEN e ROSTAD, 2009). Mais importante ainda, a estrutura de tempo do PBBG se diferencia de outros tipos de jogos, uma vez que usuários podem acessar o jogo várias vezes no mesmo dia e tomar decisões que apenas produzirão algum efeito depois de certo tempo, como acontece com frequência em ambientes reais (KLIMMT, SCHMID e ORTHM, 2009). Dessa forma, o usuário se compromete a jogar com certa frequência, caso queira avançar no jogo, e, conseqüentemente, irá adquirir mais conhecimentos acerca daquele domínio específico. Essa característica permite que aqueles jogadores que não dispõem de tempo suficiente progridam tanto quanto usuários que jogam durante um tempo maior, desde que os primeiros tomem atitudes mais acertadas no jogo.

### 2.3. JOGOS EDUCATIVOS

Os chamados jogos educativos surgiram como um meio mais atrativo e motivador de ensino. O aprendizado baseado em jogos educativos ou jogos sérios está se tornando uma tendência importante na área de pesquisa em EaD, pois trata vários problemas típicos da EaD, como altas taxas de abandono, devido à frustração e à falta de motivação para continuar a estudar, e a sobrecarga cognitiva do treinando (YESSAD, LABAT e FRANÇOIS, 2010).

Essa nova abordagem de jogos educativos eletrônicos a distância tem emergido como um meio mais atrativo e motivador de EaD e é uma das áreas mais promissoras para a melhoria da educação (JUNQI, ZONGKAI e YUMEI, 2009). Apesar desse crescimento, o uso de jogos para a aprendizagem é discutível, pois são geralmente vistos como instrumentos para diversão (HODGSON, MAN e LEUNG, 2010). No entanto, como o jogo on-line requer dos jogadores concentração e dedicação para superar os desafios, os instrutores podem usar essa motivação para permitir aquisição de conhecimento e habilidades por parte dos treinandos. Estudos indicam que jogadores melhoram sua concentração, seus reflexos e sua capacidade de tomada de decisões rápidas (MORSI e JACKSON, 2007).

De fato, tem havido uma tendência crescente para a utilização de jogos educativos e simulações para ajudar os treinandos a dominarem os desafios inerentes à aprendizagem de

conceitos e refletir sobre suas experiências de aprendizagem através dessas interações (FREITAS e OLIVER, 2006). Um dos princípios defendidos por Junqi *et al.*, (2009) é que educação e entretenimento possuem a mesma importância em jogos de computador educacionais. Isso significa dizer que, quanto à educação, os jogos utilizados para educação devem incluir o conhecimento necessário e, quanto ao entretenimento, fatores que motivem os treinandos a explorarem o jogo devem ser inclusos, como, por exemplo, desafios, ilusões, respostas imediatas, objetivos e competição (JUNQI, ZONGKAI e YUMEI, 2009).

Reis *et al.* (2008) descrevem um jogo lúdico educacional sobre a revolução da cabanagem que apresenta a revolução através de imagens, textos, animações, narrativas e através do próprio ambiente virtual do jogo. O jogo educativo sobre a cabanagem foi desenvolvido baseado na abordagem construtivista, onde o treinando, ao jogar, constrói o conhecimento abordado pelo jogo, ficando o instrutor responsável por guiá-lo na construção do seu conhecimento. O jogo da cabanagem inspira-se nos jogos comerciais para permitir ao treinando aprendizado através de um ambiente motivador.

Focando mais nos conceitos de um jogo educativo para a indústria de petróleo, Perez, Marin e Perez (2007) apresentam o processo de concepção de um protótipo de RV neste contexto. Neste trabalho, os cenários interativos foram desenvolvidos para o treinamento de segurança em uma plataforma de petróleo marítima. A metodologia adotada no projeto incluiu a seleção de um conjunto de ferramentas de RV, a implementação de cenários, a otimização de elementos interativos e o desenvolvimento de ferramentas adicionais.

O ambiente virtual proposto em (PEREZ, MARIN e PEREZ, 2007) contempla os seguintes aspectos: (i) interface de usuário, que controla o acesso a outras aplicações; (ii) explicação passo a passo na forma de animação; (iii) livre exploração; (iv) contingência; (v) resposta a uma contingência; (vi) consultas a um banco de dados; e (vii) manutenção. No entanto, este ambiente virtual não inclui pontuação, ambiente multijogador, aprendizado coletivo ou agentes para criar situações adversas.

### 3. TEORIAS DE APRENDIZAGEM

Muitas teorias apoiam a aprendizagem colaborativa. Esse capítulo tem por objetivo apresentar as principais teorias que oferecem apoio a este tipo de aprendizagem.

#### 3.1. TEORIA DA CARGA COGNITIVA

O processo de aquisição de conhecimento e entendimento através do pensamento é denominado processo cognitivo. A teoria da carga cognitiva baseia-se na impossibilidade intrínseca do ser humano em processar muitas informações a cada momento na memória. O excesso de informação no processo cognitivo acarreta em um esforço, criando assim uma sobrecarga cognitiva. Essa sobrecarga impõe uma restrição fundamental no desempenho e na capacidade de aprendizado (VOGEL-WALCUTT, *et al.*, 2010).

A teoria da carga cognitiva defende que a elaboração de materiais didáticos, principalmente os que utilizam de recursos multimídia, deve seguir alguns princípios para diminuir a sobrecarga cognitiva do treinando e, assim, potencializar seu aprendizado. Existem três tipos de cargas cognitivas (VOGEL-WALCUTT, *et al.*, 2010).

- Carga Intrínseca: relacionada com a quantidade ou a complexidade da informação que está sendo transmitida. Ela leva em consideração a natureza do material didático e o nível de perícia do treinando;
- Carga Irrelevante: relacionada com a presença de informações irrelevantes para o aprendizado, o que gera processamento desnecessário na memória de trabalho (*Work Memory – WM*)<sup>2</sup> do treinando. É vista como resultado de um projeto instrucional mal elaborado; e
- Carga Relevante: também conhecida como Carga Cognitiva Pertinente ou *Germane Cognitive Load*, que está relacionada com o esforço cognitivo necessário à aquisição,

---

<sup>2</sup> Memória de trabalho ou memória de curto prazo entende-se como um componente cognitivo ligado à memória, que permite o armazenamento temporário de informação com capacidade limitada.

construção e automatização de esquemas. Ela é fortemente influenciada pela forma como a informação é apresentada. Não é inerente ao conteúdo ensinado, mas sim, ao projeto instrucional utilizado.

### 3.2. TEORIA SOCIOCULTURAL DE VYGOTSKY

A teoria sociocultural de Vygotsky (VIGOTSKI, 1998; RIBAS e MOURA, 2006) enfatiza que a inteligência do indivíduo se origina da sociedade e da cultura, ou seja, de uma interação entre o indivíduo e o ambiente no qual ele está inserido. Esta teoria de aprendizagem enfatiza que o conhecimento humano advém primeiramente do meio, através de relações interpessoais, e não de aspectos internos (intrapessoal) do indivíduo. Esta perspectiva remete à importância da sociabilidade e interação das pessoas no desenvolvimento mental dos indivíduos. Este é o elemento fundamental da teoria de Vygotsky sobre interação social, ou seja, que esta possui um papel formador e construtor. Isto significa que muitas funções mentais do indivíduo não são possíveis de serem construídas sem as interações sociais (VIGOTSKI, 1998).

Em (VIGOTSKI, 1998), é apresentado um dos principais conceitos desta teoria de aprendizagem, a Zona de Desenvolvimento Proximal - ZDP. Segundo este conceito, os indivíduos possuem uma capacidade de resolução de problemas elementar e outra capacidade que só pode ser alcançada com a ajuda de um instrutor (facilitador). Este limite corresponde à ZDP. Assim, cada um é capaz de resolver alguns problemas sem ajuda e isto representa seu conhecimento em uma dada idade. Contudo, o indivíduo pode atingir um limite (ZDP) caso obtenha ajuda do facilitador no processo de aprendizagem, onde este limite seria sua idade mental real. Vygotsky analisou em sua obra exemplos de crianças em seu processo de aprendizagem inicial.

É importante salientar que neste aspecto o treinando possui uma grande importância no processo de aprendizagem. O instrutor exerce um papel de facilitador da aprendizagem, guiando o treinando na resolução de problemas mais complexos, que ele por si só não poderia solucionar. Esta teoria pode ser facilmente aplicada em ambientes de aprendizagem colaborativa, pois, além das interações com os demais colegas no processo de aprendizagem,

o instrutor desempenha o papel do facilitador neste processo, dinamizando o processo de aprendizagem (VIGOTSKI, 1998).

### 3.3. TEORIA DA COGNIÇÃO DISTRIBUÍDA

A Teoria da Cognição Distribuída foi desenvolvida por Hutchins (1995) citado por (ROGERS, 2006), sendo considerada uma mudança radical na forma de pensar sobre o fenômeno da cognição. A visão tradicional enfoca que o fenômeno de cognição é intrínseco ao indivíduo, enquanto Hutchins afirma que este processo se dá de forma distribuída no ambiente, englobando fatores externos, sobretudo artefatos do meio (ROGERS, 2006).

Assim, a cognição distribuída está relacionada com a interação do indivíduo com os artefatos do meio e não só com uma perspectiva interna. Em Hutchins (1995), são mostrados alguns exemplos do uso desta teoria em problemas reais, como, por exemplo, na análise do processo de navegação de um navio e na análise de um piloto de avião em sua cabine (ROGERS, 2006). Em outro exemplo, Rogers (1997) mostra um estudo de como controladores aéreos interagem com um sistema de radar ao controlar o tráfego. Estes exemplos mostram que a cognição está relacionada com o indivíduo e sua relação com o meio.

### 3.4. TEORIA DA FLEXIBILIDADE COGNITIVA

Esta teoria de aprendizagem foi proposta por Spiro na década de 80 (SPIRO, COLLINS e THOTA, 2003). Ela afirma que as pessoas adquirem conhecimento em ambientes não-estruturados (complexos) pela associação entre unidades conhecidas. Este conceito pode ser melhor ilustrado (implementado) em ambientes que utilizam documentos hipermídia, onde os documentos estão inter-relacionados, ou seja, os conceitos similares estão dispostos através de referências cruzadas.

A teoria da flexibilidade cognitiva afirma que o aprendizado é dependente do contexto do treinando. Ela está embasada em teorias construtivistas e também enfatiza a construção do conhecimento pelo indivíduo. Assim, esta teoria é bem aplicada em ambientes interativos de

aprendizagem, onde o usuário aprende pela interação com o ambiente de aprendizagem (SPIRO, COLLINS e THOTA, 2003).

Em (GREENE e ROGERS, 2006), foi descrito um framework para avaliação de treinandos da área de saúde, baseado na teoria da flexibilidade cognitiva, com o objetivo de verificar se realmente os treinandos atingiram as habilidades necessárias.

### 3.5. TEORIA DA APRENDIZAGEM BASEADA EM PROBLEMA

De acordo com Hmelo-Silver (2004), a Aprendizagem Baseada em Problema, do inglês *Problem-Based Learning*, é um método no qual os treinandos aprendem através da resolução de um problema. A aprendizagem é centrada no treinando e auto-dirigida. Os treinandos trabalham em pequenos grupos colaborativos na resolução de um problema. O instrutor atua como facilitador do processo de aprendizagem ao invés de apenas passar conhecimentos (PONTES, 2010; HMELO-SILVER e BARROS, 2006).

O método está estruturado em um ciclo de desenvolvimento contendo as seguintes etapas (RIBEIRO, 2008):

- É apresentado um cenário do problema aos treinandos;
- Os treinandos formulam e analisam o problema e extraem fatos relevantes ao problema em questão;
- Os treinandos têm um melhor entendimento do problema e formulam hipóteses para uma possível solução;
- O conhecimento adquirido é avaliado e são identificadas deficiências neste conhecimento;
- Um novo conhecimento é gerado a partir desta avaliação anterior;
- Ao final de cada problema, os treinandos irão refletir sobre os conhecimentos adquiridos.

O instrutor tem o papel de guiar os treinandos neste processo, identificando possíveis deficiências de conhecimento e habilidades necessários à solução do problema proposto. Assim, nesta teoria de aprendizagem, no lugar do instrutor apenas repassar os conhecimentos e depois testá-los através de avaliações, ele faz com que os treinandos apliquem o seu

conhecimento em situações novas. Os treinandos se deparam com problemas muitas vezes mal estruturados e tentam descobrir, através da investigação e pesquisa, soluções úteis.

### 3.6. TEORIA DA APRENDIZAGEM BASEADA EM PROJETO

Os sistemas educacionais vêm sofrendo transformações com o intuito de adaptar os currículos para desenvolver capacidades que são requisitos dos profissionais nos dias de hoje, tais como: conhecimentos em informática, administração, custos, controle pessoal, psicologia, entre outras. Além disso, estas capacidades devem estar aliadas a características como criatividade, liderança, iniciativa, capacidade de negociação e cultura geral. Espera-se ainda que o profissional seja inovador, questionador e que busque alternativas e soluções para os problemas apresentados, encorajando assim a construção de novas competências (Intel Educar, 2011).

Das principais indicações disponíveis na literatura acadêmica, a proposta teórico-metodológica de reorganização do currículo por projetos, do educador espanhol Fernando Hernández, parece ser a mais adequada para utilização em Aprendizagem Colaborativa Baseada em Jogos (NITTA, TAKEMURA e IZURU, 2009). Ela se baseia nas idéias de John Dewey (1859-1952), filósofo, psicólogo e pedagogo norte-americano que defendia a relação da vida com a sociedade, dos meios com os fins e da teoria com a prática (RAILSBACK, 2002).

Segundo Thomas (2000), a Aprendizagem Baseada em Projeto (PBL, do inglês *Project-Based Learning*), é um modelo que organiza o processo de aprendizagem em torno de projetos, os quais podem ser definidos como tarefas complexas baseadas em desafios ou problemas que envolvem os treinandos no desenvolvimento, na solução de problemas, na tomada de decisões e em atividades investigativas e de pesquisa. Estas atividades, realizadas pelos treinandos com certa autonomia, resultam em produtos ou apresentações. Conforme Wijnen (2000), a PBL pode ser caracterizada como: (i) integradora, indo além das disciplinas convencionais; (ii) cooperativa, estimulando o trabalho em equipe; (iii) orientada à prática (*hands on*) e às competências; (v) multidisciplinar; (vi) criativa; e (vii) motivadora.

Para Hernández (1998), pode-se afirmar que PBL é uma estratégia de ensino-aprendizagem que tem por finalidade vincular teoria e prática, por meio da investigação de um tema ou problema. No contexto de jogos de simulação, esta teoria de aprendizagem pode

proporcionar aprendizagem diversificada, inserida em uma situação onde o treinando é sujeito ativo no processo de produção do conhecimento. Essa teoria de aprendizagem rompe com a imposição de conteúdos de forma rígida e pré-estabelecida, na medida em que estes se constituem como parte fundamental para o desenvolvimento do projeto.

A PBL apresenta muitos benefícios para os treinandos e instrutores, sendo os principais apresentados a seguir:

- **Aumento da Motivação:** A participação dos treinandos na PBL propicia maior interesse por parte dos treinandos bem como condutas motivadas pela vontade de aprender. Os instrutores que utilizam essa abordagem frequentemente relatam que os treinandos voluntariamente dedicam tempo e esforço extra para um projeto (THOMAS, 2000).
- **Desenvolvimento de Novas Habilidades:** Algumas das habilidades desenvolvidas são: i) **Autonomia:** Os treinandos que trabalham com projetos têm mais oportunidades de desenvolverem habilidades relacionadas à construção da autonomia, tanto em relação à responsabilidade para com as tarefas como em relação à própria aprendizagem (BOALER, 1999); ii) **Resolução de problemas e tomada de decisões:** A PBL oportuniza aos treinandos situações nas quais podem desenvolver habilidades relacionadas à resolução de problemas e à tomada de decisões. Em um projeto, os treinandos trabalham na apresentação e na resolução de problemas complexos e na tomada de decisões importantes (THOMAS, 1998); iii) **Colaboração:** Muitos projetos dependem de que os treinandos trabalhem em grupos ou equipes. Partindo do pressuposto de que a aprendizagem se dá na interação, os treinandos aprendem mais em um ambiente colaborativo (WIBURG, 1994); iv) **Organização, gerenciamento de tempo e autocontrole:** O trabalho em projetos envolve a realização de tarefas complexas, as quais proporcionam o desenvolvimento da organização, do gerenciamento de tempo e do autocontrole (THOMAS, 2000).
- **Maior Engajamento dos Treinandos:** Os treinandos que mais se beneficiam de PBL tendem a ser aqueles para os quais os métodos e abordagens tradicionais não são eficazes. Na PBL, os treinandos que anteriormente eram "difíceis de atingir" começam a participar mais das atividades. O acesso a uma ampla variação de oportunidades de

aprendizagem fornece uma estratégia para o engajamento de treinandos culturalmente diferentes (RAILSBACK, 2002).

A PBL oferece aos treinandos e instrutores a oportunidade de aprenderem a trabalhar em grupo e realizar tarefas comuns. Ela exige que os treinandos monitorem seu próprio desempenho e suas contribuições ao grupo. A PBL força os treinandos a confrontar problemas inesperados e descobrir como resolvê-los, além de oferecer aos treinandos tempo para se aprofundar em um assunto e ensinar aos outros os que aprenderam (MARKHAM, LARMER e RAVITZ, 2008).

É uma estratégia de ensino-aprendizagem que exige mais dos treinandos e instrutores para que possa funcionar. Para os instrutores, significa uma maior reflexão sobre a atividade de ensino e uma mudança de postura tradicional de especialista de conteúdo para um facilitador de aprendizagem.

### 3.7. APRENDIZAGEM BASEADA EM PROJETO *VERSUS* APRENDIZAGEM BASEADA EM PROBLEMA

Neste trabalho, o termo PBL refere-se à Aprendizagem Baseada em Projeto. Contudo, outros utilizam PBL quando se referem à Aprendizagem Baseada em Problema. Para reforçar o entendimento e deixar claro a diferença entre as duas abordagens, será feito um comparativo entre elas. Em primeiro lugar, não há um acordo quanto ao significado exato para esses termos e, muitas vezes, eles são utilizados como sinônimos. Ambas apresentam o processo de utilizar problemas “mal-estruturados”, deliberadamente formulados, para que os treinandos adquiram conhecimento de conteúdo específico e habilidades de resolução de problemas enquanto procuram soluções para questões significativas (MARKHAM, LARMER e RAVITZ, 2008).

A Aprendizagem Baseada em Projeto é um termo geral que descreve um método de ensino que utiliza projetos como foco central de ensino em uma diversidade de disciplinas. Muitas vezes, os projetos surgem a partir de um contexto autêntico, abordando questões controversas ou importantes na comunidade, e se desenvolvem de modos imprevistos. Em contraste, a Aprendizagem Baseada em Problema utiliza o desempenho de papéis e cenários realistas para conduzir os treinandos por um caminho de resultados. Independente de como

seja rotulado, um projeto deve estar vinculado a padrões de conteúdo e permitir uma investigação de uma questão significativa centrada no treinando (MARKHAM, LARMER e RAVITZ, 2008).

## 4. AGENTES E SISTEMAS MULTIAGENTE

Este capítulo tem por objetivo apresentar os conceitos de sistemas multiagente. A Seção 4.1 apresenta os conceitos de agentes e sistemas multiagente. A Seção 4.2 mostra um breve estudo das metodologias de modelagem de sistemas multiagente e um comparativo entre elas. A Seção 4.3 detalha o agente reativo baseado em modelo e o uso da lógica *fuzzy*.

### 4.1. AGENTES

Embora não exista um consenso acerca da definição do termo agente, pode-se entendê-lo a partir de algumas definições de pesquisadores da área. Segundo Henderson-Sellers e Giorgini (2005), agentes são entidades de software ou não que são caracterizadas por serem: autônomas, proativas, situadas em um ambiente e direcionadas a objetivos. Um agente ser autônomo significa que ele pode agir de forma independente e definir seu próprio comportamento. Um agente proativo é aquele que não necessita de pedido externo para agir. Todo agente também está situado em algum ambiente específico. Um agente é orientado a objetivos quando ele tem alguma meta no ambiente no qual está inserido (PONTES, 2010).

Segundo Russell e Norvig (2009), um agente está situado em determinado ambiente, percebendo-o através dos seus sensores e agindo sobre ele através de seus atuadores. Os autores também introduzem o conceito de agente racional, que é aquele que realiza de forma correta suas tarefas dada uma medida estipulada de desempenho, que irá avaliar o grau de sucesso dos comportamentos do agente no ambiente em questão.

Em (RUSSELL e NORVIG, 2009), um agente racional é definido como:

Um agente racional é aquele que, para cada possível sequência de percepções, deve selecionar a ação que maximize sua medida de desempenho, baseado na sua sequência de percepções e no seu conhecimento do ambiente.

Outra definição de agentes com boa aceitação é a definição de Wooldridg (2002), segundo a qual um agente é um sistema computacional que está situado em algum ambiente e

que é capaz de efetuar ações autônomas neste ambiente a fim de cumprir os objetivos para os quais ele foi feito.

De acordo com Maes (1995), agentes inteligentes são sistemas computacionais residentes em ambientes dinâmicos complexos, os quais percebem e atuam autonomamente neste ambiente e, ao fazê-lo, realizam um conjunto de objetivos e tarefas para os quais foram designados.

Quanto à estrutura, Almir (2008) classifica os tipos de agentes em:

- **Agente tabela:** É a estrutura mais simples que um agente pode assumir. Neste modelo, todas as percepções e ações possíveis estão relacionadas em uma tabela.
- **Agente reativo simples:** Possui uma estrutura simples, substituindo a tabela de percepção-ação por um conjunto de regras do tipo condição-ação.
- **Agente reativo baseado em modelo:** Este tipo de agente mantém internamente um modelo do seu ambiente e, para selecionar uma ação, leva em consideração o histórico das percepções.
- **Agente baseado em objetivos:** Este tipo de agente pondera suas ações sempre levando em consideração a tentativa de alcançar seus objetivos, além de possuir também um conjunto de planos internos que são usados para cumprir estes objetivos.
- **Agente baseado em utilidade:** Este tipo de agente pondera suas ações sempre tentando maximizar uma função de utilidade que avalia seu grau de “felicidade” na escolha de uma determinada ação.
- **Agente com aprendizagem:** Este agente tem uma grande adaptabilidade às mudanças do ambiente, pois vai aprendendo com ele. Ele é composto por quatro componentes: elemento de aprendizado, elemento de desempenho, elemento crítico e elemento gerador de problemas.

Portanto, dada a definição do que seria um agente isolado, podemos introduzir o conceito de Sistema Multiagente (SMA) como aquele composto de agentes cooperativos ou competitivos que interagem entre si para atingir um objetivo individual ou comum (HENDERSON-SELLERS e GIORGINI, 2005). Este conceito de SMA nos leva à percepção

de algumas características novas, tais como comunicação e sociabilidade, ou seja, os agentes necessitam de alguma forma de comunicação para atingir suas metas. Os agentes deverão trabalhar juntos e isso exige um comportamento social dos mesmos (RUSSELL e NORVIG, 2009).

## 4.2. METODOLOGIAS PARA MODELAGEM DE SISTEMAS MULTIAGENTE

Esta seção tem por objetivo apresentar diversas metodologias de modelagem de sistemas multiagente.

### 4.2.1. TROPOS

Tropos (CASTRO, GIORGIN e KETHERS, 2005) é uma metodologia orientada a agentes dirigida a requisitos. Esta metodologia usa conceitos de um framework de modelagem chamado *i\** (YU, 1995).

Tropos possui quatro fases (CASTRO, GIORGIN e KETHERS, 2005): análise de requisitos inicial, análise de requisitos posterior, projeto da arquitetura e projeto detalhado.

A fase de análise de requisito divide-se em: análise de requisitos inicial e análise de requisitos posterior. A fase da análise de requisitos inicial consiste no entendimento da organização em termos de requisitos, ou seja, o que o sistema deve fazer para resolução das necessidades da organização. Na análise de requisitos posterior, o modelo conceitual desenvolvido inicialmente é expandido para modelar os requisitos funcionais e não-funcionais do sistema.

Em ambas as fases são usados dois tipos de diagramas: (i) diagrama de atores, que é um diagrama que envolve as dependências entre os atores do sistema no que tange ao cumprimento de metas; e (ii) diagrama de raciocínio, que descreve o processo de raciocínio de cada ator no sistema.

No projeto da arquitetura é definida a arquitetura global do sistema em termos de atores e suas dependências. Esta fase é efetuada em três passos: i) no primeiro passo toda a arquitetura do sistema é definida. Novos atores podem ser adicionados; ii) o segundo passo é

caracterizado pela definição das capacidades de cada ator no cumprimento de suas metas; e iii) o último passo consiste no agrupamento de atores de acordo com suas habilidades. Neste passo, os agentes são associados em grupos conforme suas capacidades.

Por fim, o projeto detalhado consiste na fase em que é feita uma especificação detalhada de cada agente. Metas, crenças e capacidades, bem como comunicação entre os agentes, são especificadas em detalhes. Esta fase se caracteriza pelas escolhas de implementação. Nesta fase são usados diagramas da UML (*Unified Modeling Language*) (SOMMERVILLE, 2010), bem como AUML (*Agent UML*) (RONALD, STERLING e KIRLEY, 2006), para definir protocolos específicos aos agentes. Estes diagramas são: (i) Diagramas de Capacidade, que especificam a capacidade de um determinado agente; (ii) Diagramas de Planos, que modelam os planos a serem feitos; e (iii) Diagramas de Interação entre agentes, que usam os diagramas AUML para modelagem da comunicação entre os agentes.

#### 4.2.2. PASSI

PASSI (*Process for Agent Societies Specification and Implementation*) (CHELLA, COSENTINO e VALERIA, 2009) é uma metodologia que atua no desenvolvimento de SMA usando conceitos extraídos da orientação a objetos. O uso da UML se estende da fase de requisitos até a codificação.

A metodologia PASSI possui cinco modelos: modelo de requisitos de sistema, modelo de sociedade de agentes, modelo de implementação de agentes, modelo de código e modelo de implantação (COSENTINO, 2005).

O primeiro modelo a ser gerado trata-se do modelo de requisitos do sistema que tem por objetivo obter os requisitos do sistema. Ele está dividido em quatro fases (COSENTINO, 2005):

- Descrição de Requisitos de Domínio, contemplando uma descrição funcional do sistema usando casos de uso convencionais;
- Identificação de Agentes, onde as funções do sistema são agrupadas e distribuídas para cada agente;
- Identificação de Papéis, descrevendo uma sequência de diagramas relacionados a uma funcionalidade específica;

- Especificação de Tarefas, especificando as capacidades de cada agente.

Já o modelo de sociedade de agentes é um modelo de interações e dependências entre os agentes envolvidos no sistema. Este modelo tem três fases (CHELLA, COSSENTINO e VALERIA, 2009):

- Descrição Ontológica, que usa diagramas de classes para descrever o conhecimento atribuído a cada agente e suas comunicações;
- Descrição de Papéis, onde diagramas de classes são usados para mostrar os papéis desempenhados por cada agente, as tarefas envolvidas, as capacidades de comunicação e as dependências entre agentes;
- Descrição de Protocolos, que usa diagramas de sequência de mensagens para especificar a gramática de cada protocolo de comunicação.

O modelo de implementação dos agentes é um modelo clássico de solução arquitetural baseado em classes e métodos. Ele se diferencia do método clássico por existir dois diferentes níveis de abstração, a saber: o nível social (múltiplos agentes) e o modelo de um único agente, que possui duas etapas: Definição de Estrutura de Agentes, onde diagramas de classe convencionais descrevem as classes dos agentes participantes da solução; e Descrição de Comportamento dos Agentes, onde diagramas de atividade descrevem o comportamento individual de cada agente.

O modelo de implementação possui dois passos (definição da estrutura e definição de comportamentos) que visam descrever a arquitetura interna dos agentes e seus comportamentos. Também são representadas a invocação de métodos e as trocas de mensagens entre agentes. No modelo de codificação, o sistema é codificado (passo de produção de código) sempre buscando reutilizar padrões predefinidos de agentes e tarefas. Neste último modelo são definidas restrições para mobilidade de agentes e como serão alocadas as unidades de agentes para uma posterior distribuição (CHELLA, COSSENTINO e VALERIA, 2009).

A metodologia PASSI possui uma ferramenta que auxilia no processo de desenvolvimento da metodologia, o PASSI ToolKit (PTK) (COSSENTINO, LOMBARDO e SABATUCCI, 2011).

### 4.2.3. Prometheus

Prometheus (PADGHAM e WINIKOFF, 2004; WINIKOFF e PADGHAM, 2005) é uma metodologia de desenvolvimento que proporciona mecanismos para a análise e projeto de SMAs baseados em arquiteturas BDI (*Belief Desire Intention*) (PADGHAM e WINIKOFF, 2003). Prometheus possui mecanismos de estruturação hierárquica que o permite ser executado em diferentes níveis de abstração. Ele utiliza o conceito de capacidade, a qual pode ser composta por planos, eventos, crenças e outras capacidades que dão habilidades específicas ao agente. Prometheus é uma metodologia usada em aplicações industriais e para ensino de graduação. A metodologia se propõe a ser prática e detalhada.

A motivação para construção de Prometheus deve-se a diversos fatores, dentre os quais destacam-se (PADGHAM e WINIKOFF, 2003): i) desenvolver uma metodologia prática, precisa, completa e detalhada; ii) desenvolver agentes orientados a metas e planos; iii) facilitar a construção de ferramentas de modelagem que aplicam a metodologia; e iv) ser utilizada em diversas áreas, seja ela comercial ou de ensino.

Prometheus pode ser dividida de forma macro em três atividades: especificação de sistema, projeto da arquitetura e projeto detalhado (PADGHAM e WINIKOFF, 2004).

A etapa de especificação de sistema parte da execução de algumas atividades: Metas do Sistema, onde é descrito o sistema em termos de metas em alto nível, gerando um artefato chamado *diagrama de metas*; Funcionalidades, que definem as capacidades do sistema, agrupando um conjunto de funcionalidades; Cenários, que são usados para descrever os passos a serem efetuados dado um evento; e Interface do Sistema, que descreve percepções e ações a serem efetuadas pelos agentes do sistema.

A etapa de projeto de arquitetura realiza três atividades: i) decidir os tipos de agentes que comporão o sistema, onde os agentes são identificados com base nos grupos de funcionalidades; ii) descrever as interações entre os agentes no sistema através de diagramas de interação e protocolos de interação; e iii) desenvolver a arquitetura global do sistema, onde o sistema é descrito através de um diagrama de sistema panorâmico, que captura os tipos de agentes do sistema e os limites do sistema em termos de percepções e ações.

A etapa de projeto detalhado consiste em: i) desenvolver o interior dos agentes em termos de capacidades; ii) desenvolver diagramas de processo dos protocolos de interação; e iii) desenvolver outras capacidades que podem ser originadas de capacidades desenvolvidas

anteriormente, bem como em termos de eventos, planos e dados. Capacidade é um mecanismo estruturado semelhante aos módulos, podendo conter: planos, dados ou eventos.

PDT (*Prometheus Design Tool*) (SUN, THANGARAJAH e PADGHAM, 2010) é uma ferramenta desenvolvida para auxiliar os projetistas no desenvolvimento dos modelos da metodologia Prometheus. Em sua versão atual, PDT foi desenvolvida como um *plugin* para o ambiente Eclipse. PDT inclui várias características para auxiliar na modelagem, como:

- Editor gráfico: A ferramenta guia o projetista por todas as fases da metodologia, oferecendo um grande número de diagramas para modelar o projeto;
- Descritores de entidade: Através de abas, podem ser editadas e visualizadas várias propriedades das entidades;
- Propagação da informação: As informações são propagadas entre as fases da metodologia, oferecendo uma manutenção de consistência quando ocorrem alterações no projeto, através de verificação cruzada entre os vários diagramas;
- Geração de código: PDT cria estruturas que podem ser usadas na plataforma JACK (RONALD, STERLING e KIRLEY, 2006) para a geração dos agentes;
- Geração de relatórios: A geração de relatórios é feita através de um documento HTML, que pode conter todo o projeto ou apenas alguns diagramas específicos.

#### 4.2.4. MESSAGE

MESSAGE (GARIJO, GOMEZ-SANZ e MASSONET, 2005) é uma metodologia para modelagem de SMA que estende a UML com novos conceitos relacionados a agentes e notações gráficas. Ela enriquece e adapta o modelo de processo RUP (SOMMERVILLE, 2010) para a análise e projeto de sistemas multiagente (GARIJO, GOMEZ-SANZ e MASSONET, 2005). MESSAGE abrange as fases de análise e projeto no ciclo de vida do desenvolvimento de software.

O processo de desenvolvimento de MESSAGE consiste nas fases análise e projeto. A fase de análise tem como objetivo gerar uma especificação do sistema que descreve o

problema a ser solucionado, gerando um modelo abstrato que auxilia na validação e no desenvolvimento do projeto. Essa fase gera cinco modelos:

- Modelo de organização;
- Modelo de agentes/papéis;
- Modelo de metas/tarefas;
- Modelo de domínio; e
- Modelo de interação.

A etapa de análise serve para produzir uma especificação do sistema ou modelo de análise. O processo de análise é feito por um refinamento por etapas. No topo da análise, ou no nível 0, é feita uma definição global do sistema. No próximo nível, o comportamento das entidades, tais como agentes, organização e objetivos, é definido. Outros refinamentos podem tratar de requisitos funcionais e não-funcionais.

O propósito do projeto é definir as entidades computacionais que representam o sistema multiagente representado na fase de análise. Estes artefatos são transformados para possibilitar a implementação. Esta fase é dividida em dois processos: projeto de alto nível e projeto detalhado.

No projeto de alto nível, o modelo de análise é refinado produzindo uma versão inicial da arquitetura do sistema.

Essa atividade é realizada em quatro etapas: a) atribuir papéis aos agentes e definir o diagrama de agentes; b) prover serviços às tarefas, detalhando-as através de um *workflow* (diagrama de atividades da UML); c) refinar os protocolos de interação através do uso de protocolos de interação da AUML (ODELL, PARUNAK e BAUER, 2000) e de diagramas de estado da UML; e d) especificar o comportamento da interação do papel em diagramas de estado da UML.

No projeto de alto nível, o modelo de análise é refinado produzindo uma versão inicial da arquitetura do sistema.

O projeto detalhado pressupõe a definição das estruturas de implementação possíveis para o projeto dos agentes, sendo que o MESSAGE define duas abordagens.

A primeira é direcionada por uma arquitetura de agente e por uma organização multiagente, considerando que o agente é uma entidade mais abrangente que uma classe. Ele é encarado como um subsistema com uma arquitetura interna que define os relacionamentos dos seus diferentes componentes.

A segunda abordagem é mais orientada à plataforma de agente, levando-se em conta que cada agente pode ser mapeado para uma classe. Este conceito é extraído dos modelos suportados pela maioria das ferramentas para a implementação de agentes, como a JADE (BELLIFEMINE, CAIRE e GREENWOOD, 2007), na qual há uma classe Agente a partir da qual tipos específicos de agentes são derivados.

#### **4.2.5. MaSE - *Multi-agent Systems Engineering***

MaSE (*Multi-agent Systems Engineering*) é uma metodologia para a análise e projeto de sistemas multiagente originalmente desenvolvida para projetar sistemas multiagente fechados, heterogêneos e de propósito geral (DELOACH e KUMAR, 2005). MaSE trata agentes como especializações dos modelos de objetos aplicando várias técnicas do paradigma orientado a objetos para especificação e projeto do sistema multiagente. Para auxílio no processo de construção de sistemas multiagente, na metodologia MaSE é usada uma ferramenta chamada *agentTool* (DELOACH, 2001).

A metodologia MaSE pode ser dividida em duas fases principais: Análise e Projeto. Apesar desta divisão, a metodologia tem um caráter iterativo, assim o projetista tem liberdade para alternar entre as fases sem seguir uma sequência rígida.

A fase de análise tem por objetivo definir um conjunto de modelos que possam ser usados para alcançar os objetivos globais do sistema. Tal fase é dividida em três passos: captura de metas, especificação de casos de uso e refinamento de papéis.

A captura de metas tem por objetivo identificar as metas e submetas do sistema, extraídas de um conjunto de requisitos, e organizá-las em um diagrama hierárquico de metas.

Em seguida é realizada a especificação de casos de uso, que traduz metas e submetas em casos de uso. Estes capturam os cenários previamente identificados com uma descrição detalhada e um conjunto de diagramas de sequência de mensagens. Estes casos de uso representam os comportamentos desejados do sistema e a sequência de eventos.

Por fim, é realizado o refinamento de papéis. Neste passo são determinados os papéis do sistema e a comunicação entre eles, associando com tarefas específicas que representam a comunicação e o conhecimento interno do agente. As tarefas são documentadas em um diagrama de tarefas concorrentes.

A fase de projeto tem por objetivo converter papéis e tarefas em uma forma mais sugestiva para implementação, especialmente os agentes e os diálogos. Tal fase é composta por quatro passos: criação das classes de agentes, construção dos diálogos, montagem das classes de agentes e projeto do sistema.

O primeiro passo dessa fase é a criação de classes de agentes, que tem o objetivo de identificar as classes de agentes e seus diálogos e, então, documentá-las em diagramas de classes de agentes. O diagrama de classes de agentes, que resulta deste passo, é semelhante ao diagrama de classes orientado a objetos, porém com duas diferenças: as classes de agentes são definidas por papéis ao invés de atributos e os métodos e os relacionamentos entre as classes de agentes são diálogos.

No segundo passo, a construção de diálogos cria um projeto de diálogo detalhado. Diálogos modelam comunicação entre duas classes de agentes usando um par de autômatos de estado finito, semelhante, na forma e na função, a tarefas concorrentes. Cada tarefa gera múltiplos diálogos, quando solicita comunicação com mais de uma classe de agente.

O passo seguinte envolve a definição da arquitetura interna de cada agente. MaSE não assume nenhuma arquitetura de agente em particular e permite que seja usada uma grande variedade de arquiteturas já existentes ou propostas novas arquiteturas. A arquitetura do sistema é definida usando componentes semelhantes aos definidos na UML.

O passo final da metodologia MaSE implica na definição do sistema a ser implementado, pois já foram definidos os diagramas para mostrar o número, os tipos e as localizações dos agentes no sistema. Nesta etapa, o framework de comunicação selecionado e a linguagem de programação escolhida são utilizados para o processo de desenvolvimento do sistema (SOUSA, DA CUNHA e MARTINS, 2009).

#### **4.2.6. MAS-CommonKADS+**

A metodologia MAS-CommonKADS (IGLESIAS, 1998; IGLESIAS, 2005) é uma extensão da metodologia CommonKADS (SCHREIBER, 2000) com abordagem voltada para modelagem de sistemas orientados a agentes.

A metodologia MAS-CommonKADS é dividida nas seguintes fases: conceituação, análise, projeto, codificação, integração, operação e manutenção. Assim como o

CommonKADS, o MAS-CommonKADS também possui um conjunto de modelos que são executados ao longo de suas fases. Seus modelos são descritos a seguir:

**Modelo de Agentes:** especifica as características dos agentes, capacidades de raciocínio, sensores e atuadores, habilidades dos agentes e os grupos aos quais pertence cada agente.

**Modelo de Tarefas:** descreve as tarefas de cada agente, metas, métodos de solução de problemas, etc.

**Modelo de Experiência ou Conhecimento:** descreve o conhecimento necessário para cada agente com vistas ao cumprimento dos seus objetivos.

**Modelo de Organização:** descreve a organização na qual os agentes estarão inseridos, ou seja, descreve a relação da organização com seu meio.

**Modelo de Coordenação:** descreve os métodos de comunicação entre os agentes e suas interações.

**Modelo de Comunicação:** modela as interações entre as pessoas e o sistema. Esta parte refere-se ao desenvolvimento de interfaces.

**Modelo de Projeto:** se embasa nos modelos anteriores e se subdivide em três modelos: i) projeto da rede, que desenvolve aspectos relevantes da infraestrutura de rede dos agentes; ii) projeto do agente, que seleciona a arquitetura mais apropriada para cada agente; e, por último, iii) projeto de plataforma, que seleciona a plataforma mais apropriada para cada arquitetura de agente previamente escolhida.

A primeira fase da metodologia MAS-CommonKADS, a conceitualização, consiste na identificação dos requisitos funcionais do sistema. Nesta fase nenhum modelo descrito acima é usado, contudo casos de uso são utilizados para identificar os possíveis usuários do sistema e possíveis metas dos usuários, descrevendo-se as maneiras de se atingir tais metas. Duas técnicas são usadas nesta fase: UER (*User-Environment-Responsibility*) (IGLESIAS, 2005), que analisa o sistema na perspectiva do usuário, do ambiente e em termos de requisitos que não necessitam da interação do usuário; e CRC (*Class-Responsibility-Collaboration*) (IGLESIAS, 2005), que consiste em tabelas que definem, para cada agente, suas metas e planos, bem como o conhecimento necessário para atingir estas metas.

A segunda fase, de análise, consiste na especificação dos requisitos do sistema. Nesta fase são utilizados os seguintes modelos: agentes, tarefas, coordenação, conhecimento ou experiência e organização.

A última fase, projeto, trata da seleção da plataforma mais apropriada para o desenvolvimento do sistema multiagente, usando os requisitos coletados nas demais fases para implementação do sistema.

Oliveira (2010) propôs MAS-CommonKADS+ para resolver alguns problemas na especificação do MAS-CommonKADS, contribuindo com a:

- Padronização dos modelos da metodologia através de uma linguagem de modelagem;
- Adição de novas características aos modelos de forma que possamos representar as técnicas de IA, como lógica *fuzzy*, sistemas baseados em regras, redes neurais, etc;
- Modelagem de agentes e objetos em uma única metodologia;
- Modelagem de SMA através de uma ferramenta de suporte.

#### 4.2.7. Análise comparativa entre as metodologias

Um estudo comparativo foi realizado por Oliveira (2010) baseado no framework proposto em (TRAN e LOW, 2005) e em suas avaliações. Este framework é composto por um método de avaliação que divide os conceitos relacionados aos SMAs em quatro grupos (Conceitos, Processos, Modelos e Notações, Aplicações e Suporte), permitindo assim uma melhor identificação de suas similaridades e diferenças, bem como os pontos fortes e fracos de cada metodologia. A Tabela 1 apresenta a avaliação das metodologias com base nestes conceitos.

Tabela 1 – Avaliação das metodologias baseada nos conceitos. Adaptado de (OLIVEIRA, 2010).

| Critérios        | Tropos             | PASSI                           | Prometheus                | MESSAGE                      | MaSE                                | Mas-Commonkads+                      |
|------------------|--------------------|---------------------------------|---------------------------|------------------------------|-------------------------------------|--------------------------------------|
| Metas do sistema | Diagrama de atores |                                 | Diagrama de objetivos     | Modelo de objetivos/tar efas | Diagrama de hierarquia de objetivos | Casos de Metas ( <i>Goal Cases</i> ) |
| Casos de uso     |                    | Modelo de requisitos do sistema | Descrição de casos de uso |                              | Diagrama de casos de uso            | Diagrama de casos de uso             |
| Papéis           |                    | Modelo de sociedade de agentes  |                           | Modelo de papéis             | Diagramas de papéis                 |                                      |
| Conceituação     |                    | Modelo de                       |                           | Modelo de                    |                                     | Modelo de                            |

|  |                               |                                   |                                  |                       |                                   |                        |
|--|-------------------------------|-----------------------------------|----------------------------------|-----------------------|-----------------------------------|------------------------|
| do domínio   |                               | sociedade de agentes              |                                  | Domínio               |                                   | Conhecimento           |
| Objetivos do agente                                      | Diagrama de atores            | Modelo de requisitos do sistema   | Descrição das classes de agentes |                       |                                   | Modelo do agente       |
| Crenças  | Diagrama de classe de agentes | Modelo de implementação do agente | Descritor de dados               |                       |                                   | Modelo de conhecimento |
| Serviços   |                               | Modelo de sociedade de agentes    |                                  |                       |                                   | Modelo do agente       |
| Percepções e ações                                       |                               |                                   | Descritor de percepções          |                       |                                   | Modelo de coordenação  |
| Arquiteturas do agente                                   | Arquitetura BDI               | Modelo de implementação do agente | Arquitetura BDI                  |                       | Diagrama de arquitetura do agente | Modelo de projeto      |
| Protocolos de interação                                  | Diagrama de seqüência         | Modelo da sociedade do agente     | Protocolo de interação           | Modelo de interação   | Diagrama de comunicação           | Modelo de Coordenação  |
| Ambiente   |                               |                                   | Diagrama de visão do sistema     | Modelo de organização |                                   |                        |
| Estrutura organizacional e relacionamentos entre agentes | Requisitos não funcionais     |                                   |                                  | Modelo de organização |                                   | Modelo de organização  |
| Implantação  |                               | Modelo de implantação             |                                  |                       | Diagrama de implantação           |                        |

Com objetivo de escolher uma metodologia que seja apropriada para realização dos objetivos de desenvolvimento do Ambiente de Treinamento de Equipes de Perfuração (ATEP), proposto neste trabalho, foi realizada uma revisão das metodologias apresentadas nessa seção. Foi verificado que algumas metodologias possuem aplicabilidades específicas enquanto outras são de propósito geral.

Em uma análise realizada por Pontes (2010) foi destacado para o uso em aplicações que possuem aprendizagem colaborativa as seguintes metodologias: MAS-CommonKADS, PASSI, Prometheus e MaSE. Para o contexto do ATEP a metodologia que se apresentou melhor, foi a MAS-CommonKADS+ principalmente por que possui dentre outras características a possibilidade de modelagem interna dos agentes, dessa forma, podendo representar as técnicas de IA, como lógica *fuzzy*, sistemas baseados em regras, redes neurais, etc.

#### 4.2.8. Linguagens de Modelagem

Devido às inúmeras propriedades de um agente de software, o desenvolvimento de SMAs torna-se uma tarefa de grande complexidade. As linguagens de modelagem são o meio pelo qual se expressam os modelos, tendo como principal objetivo a descrição dos sistemas. Estas linguagens buscam demonstrar, através de diagramas, todas as propriedades de um agente, como, por exemplo, organizações, papéis, ambientes e relacionamentos entre os agentes. Neste trabalho são utilizadas duas linguagens de modelagem: a UML, para descrever os sistemas de apoio do jogo, e a AML (do inglês, *Agent Modeling Language*) (CERVENKA e TRENCANSKY, 2007) que é uma extensão da UML 2.0 para modelagem de sistemas multiagente.

#### 4.3. AGENTE REATIVO BASEADO EM MODELO COM LÓGICA *FUZZY*

Os agentes Reativos Simples possuem a estrutura mais simples entre os agentes, tendo um conjunto de regras do tipo condição-ação. Portanto, com apenas duas regras, como R1 – “Se temperatura > 45° então acelerar ventoinha” e R2 – “Se temperatura < 45° então desacelerar ventoinha”, um agente deste tipo poderia decidir se deveria ou não acelerar a ventoinha de um motor. Da mesma forma, ele pode possuir uma tabela contendo todas as temperaturas possíveis e ações desejadas em cada caso. O agente reativo simples possui uma estrutura com quatro elementos (ARTERO, 2009):

1. **Sensores:** O agente percebe o ambiente com seus sensores;
2. **Percepções:** O agente mapeia cada estado do ambiente em uma percepção, definida em um conjunto de possíveis percepções;
3. **Ações:** O agente toma uma decisão processando a percepção e seleciona uma ação;
4. **Atuadores:** O agente executa a ação selecionada no ambiente através de atuadores.

A Figura 3 apresenta a estrutura de um agente reativo simples.

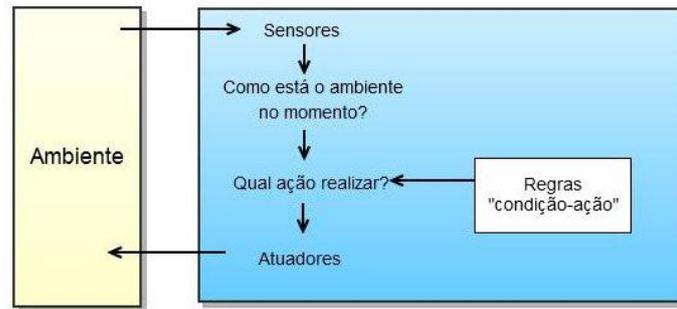


Figura 3 – Agente Reativo Simples.

O Agente Reativo Baseado em Modelo salva um estado interno com aspectos de controle do mundo, que não estão evidentes na percepção corrente, com o objetivo de ter um desempenho mais racional. Agente baseado em modelo é um agente que usa um modelo de mundo, então o estado do agente depende do histórico de percepções anteriores do ambiente, de como o ambiente evolui e de como as ações dele afetam o ambiente. A Figura 4 mostra a estrutura do agente baseado em modelo.

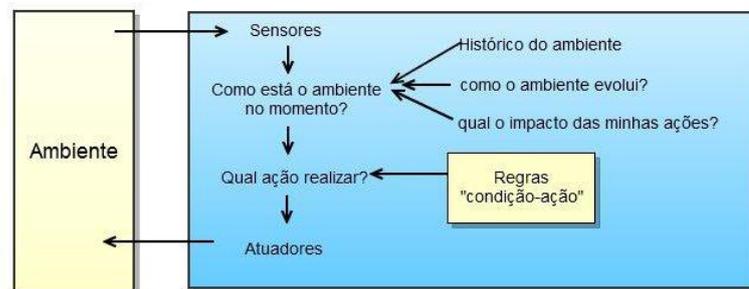


Figura 4 – Estrutura do Agente Reativo Baseado em Modelo.

#### 4.3.1. Conjuntos e Lógica *Fuzzy*

Para que os agentes não executem ações sempre “certas”, como seria o caso de uma agente apenas baseado em regras do tipo “condição-ação”, pode-se incluir um certo grau de incerteza na tomada de decisão dos agentes usando a Lógica *Fuzzy* (*Fuzzy Logic* - FL), ou Nebulosa (ZADEH, 1965), que vem sendo usada com bastante êxito em diversas aplicações, o que fez crescer bastante a sua disseminação na indústria e na comunidade acadêmica. Sua versatilidade faz dela uma excelente opção para aplicações que têm certo grau de incerteza

e/ou que precisam de flexibilidade e capacidade de adaptação. Sendo assim, jogos eletrônicos constituem um campo potencial e animador para sua aplicação (ARTERO, 2009). O campo de aplicação da FL é bastante amplo. Na engenharia química, ela pode ser usada para controle de processos. Já na ciência da computação é utilizada para reconhecimento de padrões e mineração de dados, o que é muito útil, tanto para aplicação em engenharia quanto em medicina, na identificação e diagnóstico por imagens. A FL também é amplamente usada em jogos eletrônicos (ARTERO, 2009).

Assim, a FL, introduzida por Zadeh (1965), propõe que no lugar de valores únicos sejam usados vários valores no intervalo de zero até um, representando uma variação entre o completo falso e a verdade absoluta. Um exemplo prático para entendimento é quando é medida a temperatura de algo, no caso, utilizando os extremos quente e frio, no entanto, pode-se admitir também os estados intermediários, como morno.

A FL pode ser usada para traduzir em termos matemáticos a informação imprecisa expressa por um conjunto de regras linguísticas. Se um operador humano for capaz de articular sua estratégia de ação como um conjunto de regras da forma *se ... então*, pode ser desenvolvido um algoritmo (ARTERO, 2009).

Para representação de conjuntos *fuzzy* e o seu grau de pertinência, a FL usa as funções de pertinência. As mais conhecidas são: (a) triangular, (b) trapezoidal e (c) gaussiana (AGUIAR e JUNIOR, 2007). A Figura 5 destaca graficamente essas funções.

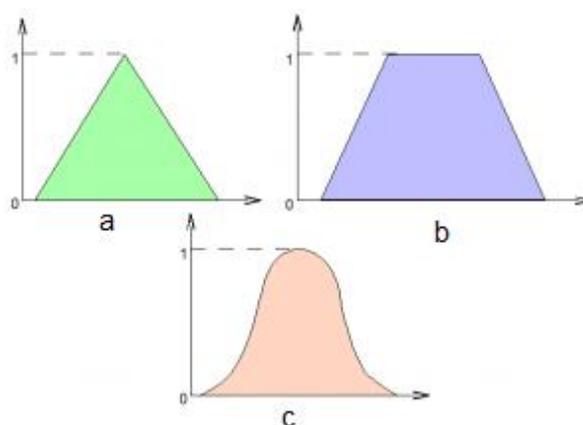


Figura 5 – Funções de Pertinência.

#### 4.3.1.1. Fuzzificação e Defuzzificação

Conforme Artero (2009), para que valores absolutos possam ser processados em um sistema usando FL, é preciso que eles sejam convertidos em valores nebulosos. Este processo de transformar um valor real em um valor nebuloso chama-se fuzzificação. Em outras palavras, a fuzzificação é usada para transformar conjuntos ordinários (*crisp sets*) em conjuntos *fuzzy*. Esta transformação é realizada por meio de funções de pertinência associadas aos elementos originais.

Já a defuzzificação é o processo de transformar um conjunto *fuzzy* em um elemento que em geral é um número real, ou seja, inverso da fuzzificação. Para o processo de defuzzificação é preciso utilizar um dos métodos apresentados a seguir (AGUIAR e JUNIOR, 2007):

- O **método do centróide** utiliza a soma dos valores de pertinência multiplicados pelos valores das variáveis, dividindo esse valor final pela área total do conjunto a ser defuzzificado;
- O **método bissetor** utiliza a soma dos valores de pertinência do conjunto;
- O **método da média dos máximos** utiliza a média dos valores máximos de pertinência do conjunto a ser defuzzificado;
- O **método do menos dos máximos** utiliza o menor valor de pertinência dentre os valores máximos de pertinência do conjunto *fuzzy* a ser defuzzificado;
- O **método do mais dos máximos** usa o maior valor de pertinência dentre os valores máximos de pertinência do conjunto *fuzzy*;
- O **método do centro de gravidade** se baseia no cálculo do centro de gravidade da função de associação. Nesse método, calcula-se a área da curva da variável linguística de saída produzida pela máquina de inferência, e acha-se o índice correspondente que divide esta área na metade.

O funcionamento do mecanismo de inferência *Fuzzy* é representado pela Figura 6. A primeira etapa é a definição das variáveis de entrada e saída e suas respectivas funções de inferência. Em seguida, inicia o processo de fuzzificação, que transforma os valores reais de entrada (no exemplo da Figura 6, 50 e 100) em valores linguísticos (na Figura 6, *baixo* e *alto*) a partir da sua função de inferência específica ( $f(x)$ , sendo  $x$  a variável). Na terceira etapa, o mecanismo de inferência deduz a variável de saída a partir das entradas linguísticas e das

regras de inferência definidas por um especialista. O último passo consiste em traduzir o valor linguístico de saída (na Figura 6, *médio*) para um valor real (na Figura 6, *estado 5*, a partir de um algoritmo de defuzzificação que consulta as funções de inferência.

Os sistemas computacionais têm o seu conhecimento geralmente representado por regras que ditam a maneira como ele deve agir. Com sistemas FL a situação não é diferente. Seus comportamentos também são ditados por regras. Entretanto, as regras utilizadas por esses sistemas são denominadas *fuzzy*.

As regras *fuzzy* caracterizam-se por uma expressão condicional da forma: SE <antecedente – expressão *fuzzy*> ENTÃO <conseqüente – expressão *fuzzy*>.

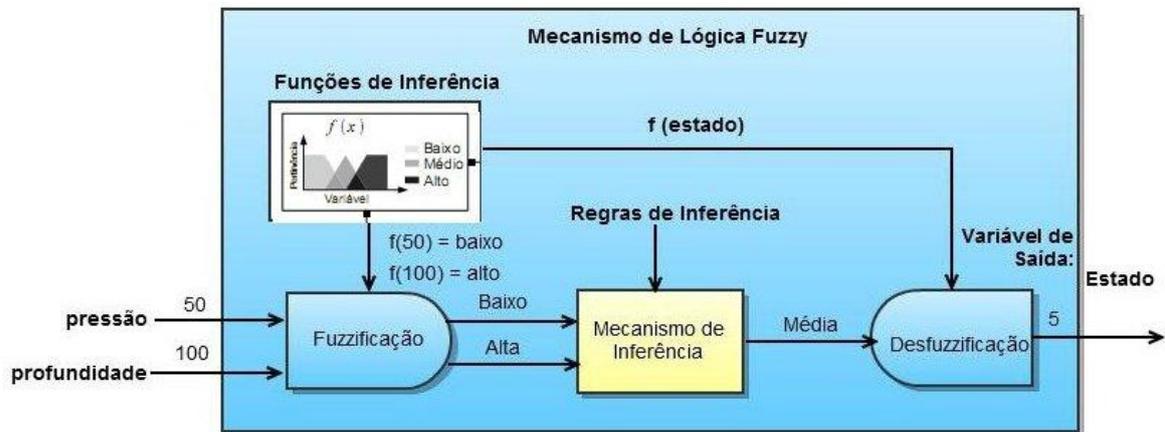


Figura 6 – Funcionamento da representação *Fuzzy*.

A lógica *fuzzy* pode traduzir termos nebulosos em valores que podem ser interpretados por computadores. Com base nos conceitos supracitados, o jogo para apoiar o treinamento de operadores de sonda de perfuração de poços, proposto neste trabalho, foi delineado para que os agentes usem um sistema *fuzzy* de dedução com base em regras pré-estabelecidas. Assim, o jogo proposto é capaz de promover desafios no ambiente cooperando de tal maneira que os agentes possam executar ações imperfeitas e gerar situações adversas.

## 5. PERFURAÇÃO DE POÇOS DE PETRÓLEO TERRESTRE

Para um melhor entendimento do trabalho proposto, esse capítulo apresenta alguns conceitos das operações de perfuração poços de petróleo terrestre. Durante a operação de perfuração existem diversos procedimentos que são realizados pelos operários, como: manobra, conexão, tratamento do fluido de perfuração e pescagem. Todos esses procedimentos e demais operações são realizadas pelos operários a partir de uma plataforma de perfuração semelhante à ilustrada na Figura 7 (ALBERTO, ALBERTO e VEROTTI, 2004).

A perfuração de um poço de petróleo é realizada através da quebra de rochas utilizando uma sonda de perfuração, que é uma estrutura desenvolvida com as especificações técnicas necessárias para dar suporte à operação. Na sonda, operários trabalham executando os processos de operação da coluna de perfuração, estrutura que compõe a sonda. A coluna de perfuração é composta de segmentos tubulares, que irão conduzir a broca, um componente que quebrará a rocha abaixo do solo até chegar às rochas onde está depositado o petróleo (ALBERTO, ALBERTO e VEROTTI, 2004).

Cada fase pode ser identificada pelo diâmetro dos segmentos utilizados. A cada etapa, uma camada de revestimento metálico é introduzida na parede do poço para dar segurança à operação. Cada camada possui um diâmetro diferente, do maior para o menor, assim como os segmentos. À medida que a perfuração vai avançando, são adicionados segmentos tubulares para prolongar a coluna de perfuração. Desta forma, prolonga-se o alcance da broca até atingir o objetivo, que são as rochas que guardam o petróleo (ALBERTO, ALBERTO e VEROTTI, 2004).



Figura 7 – Sonda de Perfuração Terrestre (ALBERTO, ALBERTO e VEROTTI, 2004).

### 5.1. Sistemas de uma Sonda de Perfuração

Todos os equipamentos de uma sonda rotativa, responsável por determinada função na perfuração de um poço, são agrupados nos chamados “sistemas” de uma sonda. Os principais sistemas são (ALBERTO, ALBERTO e VEROTTI, 2004):

- Sistema de sustentação de cargas:

O sistema de sustentação de cargas é constituído pelo mastro ou torre (*derrick*), pela subestrutura (*substructure*) e pela fundação. A torre é uma estrutura de aço especial, de forma piramidal, de modo a prover um espaçamento livre acima da plataforma de trabalho para permitir as operações. A subestrutura é constituída de vias de aço com espaçamento livre para execução de manobras.

- Sistema de geração e transmissão de energia:

Distribui a energia necessária para o acionamento dos equipamentos de uma sonda. Geralmente são usados geradores a diesel.

- Sistema de movimentação de cargas:

O sistema de movimentação de carga permite movimentar as colunas de perfuração, de revestimento e outros equipamentos.

Os principais componentes deste sistema são: guincho, bloco de coroamento (*crown block*), Catarina (*traveling block*), gancho e elevador.

- Sistema de Rotação:

Nas sondas convencionais, a coluna de perfuração é girada pela mesa rotatória localizada na plataforma da sonda. A rotação é transmitida a um tubo de parede poligonal (*Kelly*), que fica enroscado no topo da coluna de perfuração.

- Sistema de Circulação:

São os equipamentos que permitem a circulação e o tratamento do fluido de perfuração. Em uma circulação normal, o fluido de perfuração é bombeado através da coluna de perfuração até a broca, retornando pelo espaço anular até a superfície, trazendo consigo os cascalhos cortados pela broca. Na superfície, o fluido permanece dentro de tanques, após receber o tratamento adequado.

- Sistema de Segurança do Poço:

O sistema de segurança é constituído dos Equipamentos de Segurança de Cabeça de Poço (ESCP) e de equipamentos complementares que possibilitam o fechamento e o controle do poço. O mais importante deles é o BOP (*BlowOut Preventer*), que é um conjunto de válvulas que permite fechar o poço.

Os preventores são acionados sempre que houver ocorrência de um *kick*, fluxo indesejável do fluido contido numa formação para dentro do poço. Se este fluxo não for controlado eficientemente poderá se transformar em um *blowout*, ou seja, poço fluindo totalmente sem controle. Isto pode gerar sérias conseqüências, tais como danos aos equipamentos da sonda, acidentes pessoais, perda parcial ou total do reservatório, poluição e danos ao meio ambiente (ALBERTO, ALBERTO e VEROTTI, 2004).

## 6. AMBIENTE DE TREINAMENTO DE EQUIPES DE PERFURAÇÃO

Nos capítulos anteriores foram apresentados alguns conceitos relacionados à temática abordada, um estudo comparativo entre diversas metodologias de modelagem de sistemas multiagente e uma revisão das teorias que apóiam a aprendizagem. Este estudo teve por objetivo auxiliar na escolha das metodologias para dar suporte a este trabalho. Assim, para modelagem dos agentes foi escolhida a metodologia MAS-CommonKADS+ e para fundamentar o ambiente de aprendizagem foi escolhida a teoria de aprendizagem PBL. Este capítulo descreve o ambiente de treinamento, formado pelo jogo e pelo portal, o qual foi denominado de ATEP – Ambiente de Treinamento de Equipes de Perfuração. A Seção 6.1 apresenta a especificação do jogo. A Seção 6.2 mostra as ferramentas usadas no desenvolvimento deste projeto. A Seção 6.3 mostra a arquitetura completa do ambiente e apresenta uma descrição dos principais componentes do mesmo. A Seção 6.4 apresenta a interface Web e suas principais funcionalidades divididas em módulos. A Seção 6.5 descreve cenário e jogabilidade, explicando os controles do jogo e como funciona a pontuação. A Seção 6.6 mostra detalhes da arquitetura multiagente do ambiente sugerido neste trabalho, bem como a modelagem dos agentes, além de descrever como é realizada a comunicação entre eles.

### 6.1. ESPECIFICAÇÃO DO JOGO

O jogo proposto foi desenvolvido para um ambiente multijogador e baseado em agentes, com o intuito de gerar desafios semelhantes aos enfrentados pelos profissionais que trabalham na operação de perfuração de poços terrestres. Para isso, o jogo simula o trabalho de uma equipe de perfuração onde os jogadores formarão a equipe. Na equipe, cada um dos treinandos ocupa uma função específica relativa às operações da sonda de perfuração (plataformista, torrlista e sondador) e, caso a equipe não esteja completa, o sistema inclui agentes programados com os perfis dos cargos não contemplados para completar a equipe. Os agentes, com o intuito de simular situações reais de trabalho, executam operações de forma correta ou incorreta, gerando mais um desafio para a equipe.

O mundo virtual do jogo é adaptado para gerar desafios baseados em falhas, acidentes e outras situações decorrentes da operação de perfuração de poços de petróleo. Esses desafios são gerados pelos agentes que monitoram o ambiente, variando a quantidade e a complexidade das situações adversas, de modo que o jogo não se torne tedioso ou desmotivante.

Algumas questões requerem consideração ao projetar um jogo multijogador com fins educativos para avaliar o conhecimento e as habilidades dos treinandos. Em primeiro lugar, devido à atividade ser de alto risco, o mundo do jogo deve refletir bem o ambiente real. Pelo fato da Petrobras ser uma organização de grande porte e estar distribuída em todo o Brasil e em outros países, a facilidade de acesso e o caráter multiplataforma são características importantes. Para permitir o acesso distribuído, foi escolhido o desenvolvimento baseado na Web. Para permitir a simulação 3D em várias plataformas, foi escolhido um motor gráfico que possibilitasse a criação de uma aplicação 3D na Web. Para auxiliar a usabilidade, a tecnologia AJAX (OREILLY, 2005), que é amplamente utilizada em aplicações Web 2.0 (OREILLY, 2005), foi utilizada para colaboração e interação entre os jogadores, através de um painel contendo uma ferramenta de chat, a pontuação do jogo e mensagens relevantes.

Um aspecto fundamental, não considerado em muitos ambientes de aprendizagem, é a fundamentação do processo de aprendizagem em uma teoria de aprendizagem consolidada, o que permite comprovar a eficácia do processo, ou seja, validar o método para aprendizagem (HSU e LIU, 2005).

A escolha da teoria de aprendizagem PBL para apoiar este trabalho se deu pelo fato da perfuração de um poço de petróleo ser uma atividade que precisa ser realizada com base em um projeto (ALBERTO, ALBERTO e VEROTTI, 2004), fazendo com que a PBL se adequasse mais do que as outras teorias de aprendizagem. A PBL torna o treinando um sujeito ativo no processo de produção do conhecimento e oferece diversas ferramentas para a avaliação. Além disso, esta teoria de aprendizagem vem sendo usado em vários projetos de EaD apoiados por computador (NITTA, TAKEMURA e IZURU, 2009).

## 6.2. FERRAMENTAS UTILIZADAS

### 6.2.1. Motor Gráfico

O motor do jogo é o mecanismo que controla a reação do jogo em função das ações do usuário. A implementação do motor envolve diversos aspectos computacionais, tais como a escolha da linguagem de programação, o desenvolvimento de algoritmos específicos, o tipo de interface com o usuário, entre outros.

O motor gráfico jPCT<sup>3</sup> foi escolhido por oferecer bons recursos para criação de jogos 3D persistentes baseados em navegador na linguagem Java, tais como: a) É uma solução para jogos e aplicações baseados em navegadores usando um processador de imagens por *software*, bem como aceleração por OpenGL (OPENGL, 2011); b) Disponibiliza processamento por *software* (compatível Java 1.1 (DEITEL e DEITEL, 2010)) e por processador de *hardware* (Java 1.4 ou superior (DEITEL e DEITEL, 2010)); c) Se integra perfeitamente com o *Swing / AWT* (MECENAS, 2008), mas pode também ser uma opção para uso nativo do OpenGL. A Figura 8 apresenta a sonda renderizada pelo motor gráfico jPCT.

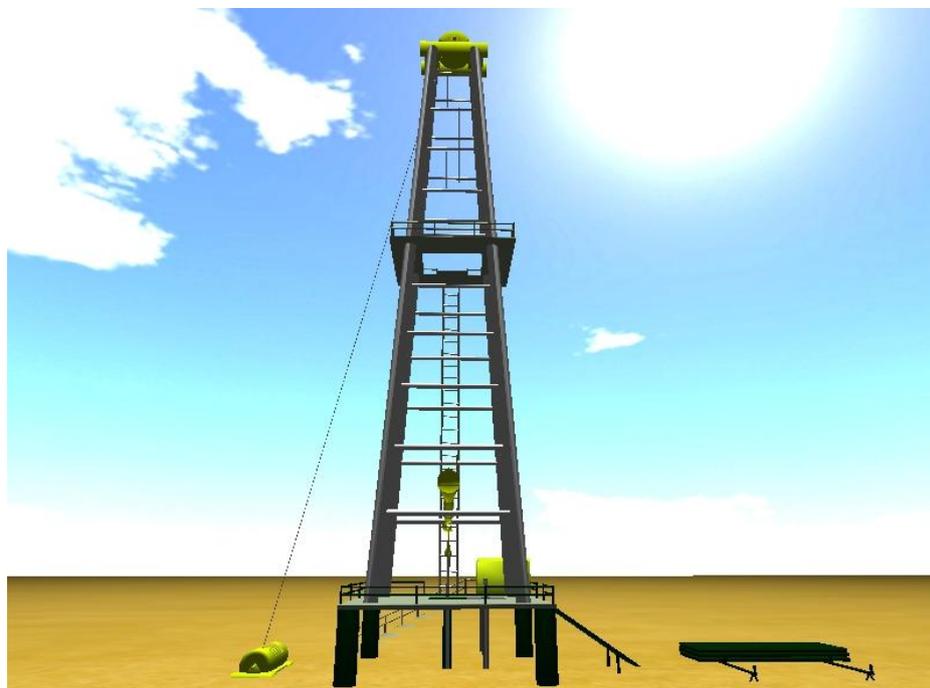


Figura 8 – Imagem da Plataforma construída no jPCT.

---

<sup>3</sup> <http://www.jpct.net/>

## 6.2.2. Ferramenta de Modelagem 3D

O Blender (BLENDER, 2011), também conhecido como *blender3d*, é um programa de computador de código aberto, desenvolvido pela *Blender Foundation* (FOUDATION, 2011), para modelagem, animação, texturização, composição, renderização, edição de vídeo e criação de aplicações interativas em 3D, tais como jogos, apresentações e outras, através de seu motor de jogo integrado, o *Blender Game Engine*. A Figura 9 apresenta a interface do Blender.

O Blender, software de modelagem 3D, foi escolhido por possibilitar o acesso a funcionalidades semelhantes às de outros softwares pagos, como, por exemplo: ferramentas de modelagem baseadas em modificadores, ferramentas de animação de personagens, dinâmica de corpo rígido, dinâmica de corpo macio e dinâmica de fluidos.

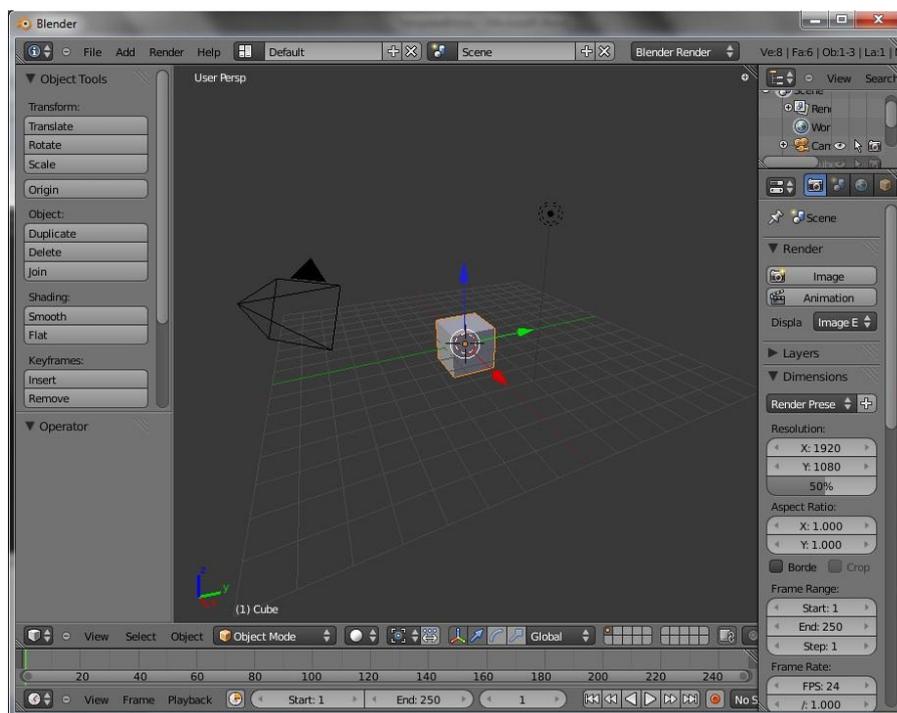


Figura 9 – Área de trabalho do Blender.

Os objetos do mundo virtual do jogo foram modelados a partir de imagens reais oriundas de visitas técnicas à Petrobras e em consultas a manuais técnicos. A Figura 10 apresenta uma visão geral da sonda com destaques para os seguintes componentes:

- A Figura 10a mostra uma visão aérea da sonda.

- A Figura 10b apresenta detalhes do bloco de coroamento que possui um conjunto de polias fixo que fica apoiado na parte superior da torre por onde passam os cabos de aço (cabos de perfuração).
- A Figura 10c destaca o componente chamado Catarina, que é composto por polias justapostas num pino central. É neste equipamento que ocorre a movimentação dos cabos passados com o bloco de coroamento, fazendo com que a Catarina se movimente ao longo da torre.

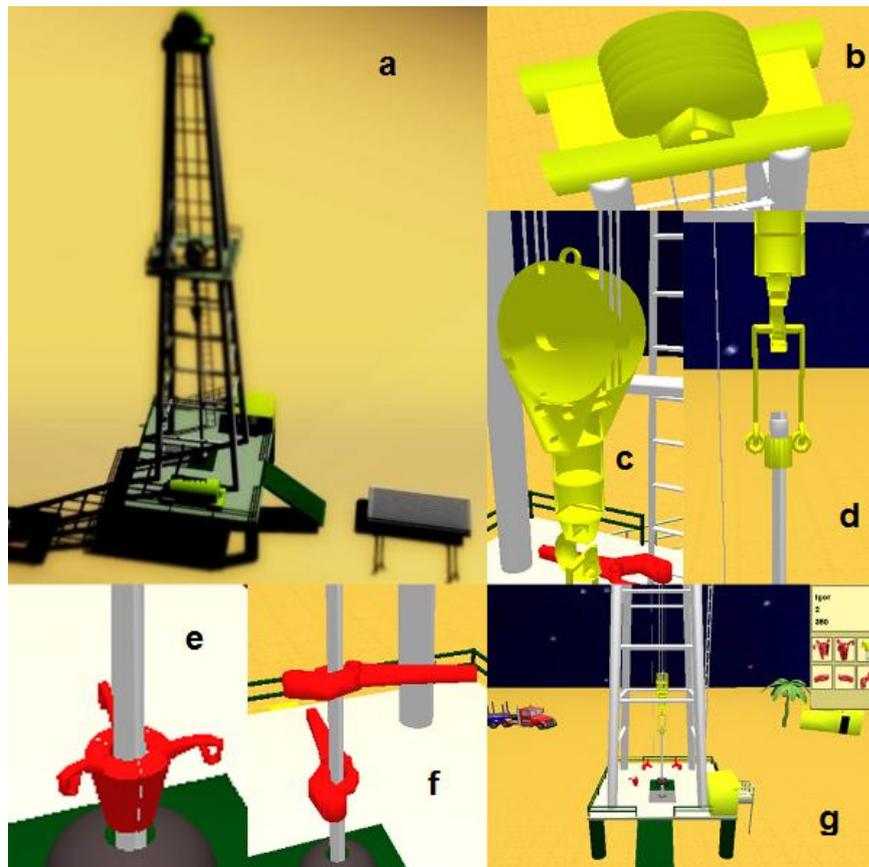


Figura 10 – Detalhes da modelagem da sonda.

- A Figura 10d mostra detalhes da coluna presa no guincho.
- A Figura 10e destaca a Cunha, que é um instrumento de ferro cortado em ângulo sólido que serve para suportar a coluna de tubos ou de revestimento na mesa rotativa.
- A Figura 10f mostra as chaves flutuantes que servem para fazer as conexões dos tubos.
- A Figura 10g destaca a visão lateral da sonda.

A Figura 11 apresenta o processo chamado de elaboração ou “redenrização”, do inglês *rendering*, de uma cena, que é a maneira pelo qual se pode obter o produto final a partir de um processamento digital. Portanto, a renderização é muito aplicada em objetos 3D, fazendo a conversão para uma representação em 2D, seja para obter uma imagem estática, seja para obter animação (DE PADUA, 2000).

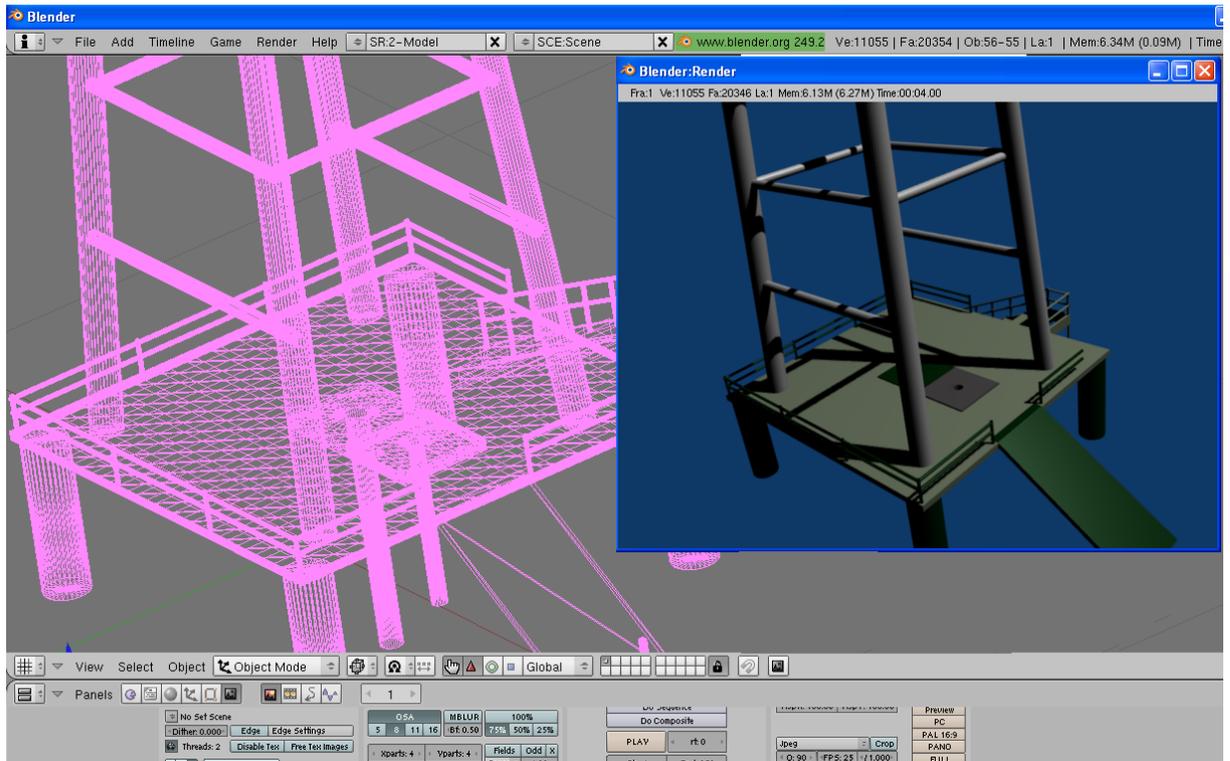


Figura 11 – Redenrização da plataforma de Perfuração.

### 6.2.3. Plataforma de Execução de Agentes

Para a implementação dos agentes inteligentes, foi utilizado o framework JADE (*Java Agent DEvelopment Framework*) (JADE, 2011). O JADE é um framework de software, independente de aplicação específica, capaz de prover funcionalidades de camada de *middleware*. Ele possui uma estrutura bastante flexível para o desenvolvimento de aplicações, onde o componente de abstração é um agente de software. Para isso, ele oferece suporte ao ciclo de vida e à lógica do núcleo de agentes em si, além de disponibilizar uma variedade de ferramentas gráficas que favorecem o desenvolvimento.

A tecnologia é totalmente escrita na linguagem de programação Java, o que traz benefícios por possuir um enorme conjunto de recursos e bibliotecas, oferecidos pela linguagem. Com isso, o programador pode economizar tempo com o trabalho inicial necessário para construir uma estrutura baseada em agentes, destinando seus esforços para o desenvolvimento da aplicação propriamente dita. A Figura 12 apresenta a interface gráfica para ativação dos agentes em um ambiente de desenvolvimento.

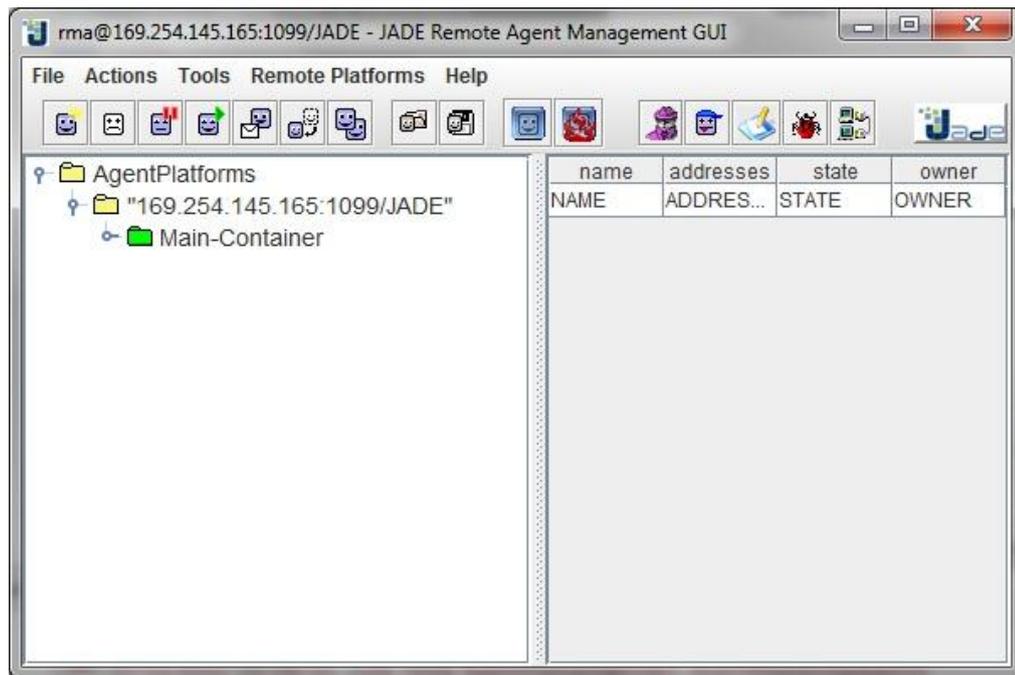


Figura 12 – Interface de desenvolvimento JADE.

O JADE inclui (BELLIFEMINE, CAIRE e GREENWOOD, 2007):

- Um ambiente de execução, onde os agentes JADE "vivem" e que deve estar ativo em um *host* antes que um agente possa ser executado;
- Uma biblioteca de classes, que programadores podem usar para desenvolver agentes;
- Um conjunto de ferramentas gráficas, que permite a administração e o monitoramento das atividades dos agentes em execução.

O Ciclo de Vida de um agente JADE é apresentado na Figura 13 (BELLIFEMINE, CAIRE e GREENWOOD, 2007).

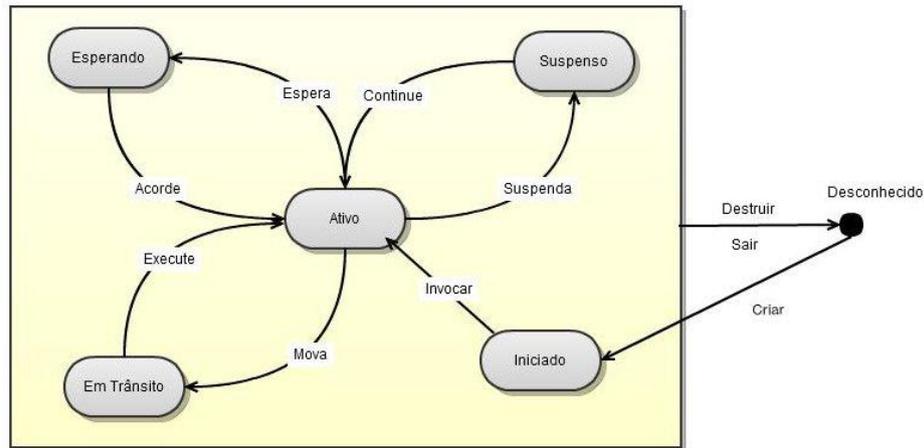


Figura 13 – Ciclo de Vida do Agente. Adaptado de (Bellifemine, *et al.*, 2007).

Como pode ser visto na Figura 13, os estados possíveis de um agente JADE são:

- **Iniciado:** O objeto agente é criado, mas ainda não se registrou no AMS, do inglês *Agent Management System*, assim, ainda não possui um identificador e não pode se comunicar com outros agentes.
- **Ativo:** O objeto agente está no AMS com um identificador e está habilitado a executar seus serviços.
- **Suspenso:** O objeto agente está parado, ou seja, sua *thread* interna está suspensa. Nesse estado o agente não está executando nenhum serviço.
- **Esperando:** O objeto agente está impedido, esperando por algum evento, ou seja, sua *thread* interna está “dormindo” e será “acordada” quando uma condição se tornar verdadeira.
- **Removido:** O objeto agente está terminado e removido do AMS.
- **Em Trânsito:** Um agente móvel entra neste estado quando está migrando para um novo contêiner. O sistema continua a armazenar as mensagens enviadas a este agente, que lhe serão passadas quando ele estiver no novo local.

#### 6.2.4. Plataforma de Desenvolvimento Web

Características do sistema, como o registro do usuário, impressão de relatórios e as ferramentas de avaliações propostas na PBL, foram implementadas usando o framework Java para Web JSF (*Java Server Faces*) (JSF, 2011), que é compatível com as outras tecnologias

empregadas. JSF é um *framework* MVC (DEITEL e DEITEL, 2010) para o desenvolvimento de aplicações Web, que permite o desenvolvimento de aplicações para a internet de forma visual, ou seja, arrastando e soltando os componentes na tela em uma IDE (*Integrated Development Environment*)<sup>4</sup>, definindo também as propriedades dos mesmos. O Netbeans (NETBEANS, 2012) foi a IDE utilizada no desenvolvimento de todos os módulos desse trabalho. A Figura 14 apresenta a tela inicial do ambiente desenvolvida com o *framework* JSF.



Figura 14 – Tela Inicial do Ambiente do Jogo.

### 6.2.5. StarUML

O StarUML (STARUML, 2011) é um projeto Open Source cujo intuito é disponibilizar uma ferramenta de modelagem de software e uma plataforma que possa substituir completamente ferramentas de modelagem UML comerciais.

Em (OLIVEIRA, 2010), foi proposta uma extensão para o StarUML que expande a AML para auxiliar na modelagem de projetos de sistemas multiagente utilizando a metodologia MAS-CommonKADS+. Toda a modelagem do sistema multiagente realizada no presente trabalho foi criada através dessa extensão.

<sup>4</sup> Como exemplo de IDEs temos Visual Studio (<http://www.microsoft.com/visualstudio>), Netbeans(<http://www.netbeans.org>) e Eclipse (<http://www.eclipse.org>).

### 6.3. ARQUITETURA

Nesta seção será detalhada toda a arquitetura do ambiente de treinamento onde será apresentada na primeira subseção a arquitetura do jogo. Em seguida, na segunda subseção, será enfatizado o ambiente multijogador, local onde será apresentado o framework H-N2N. No final, na terceira subseção, serão mostrados os detalhes da persistência dos dados do ATEP.

#### 6.3.1. Arquitetura do Jogo

Arquitetura do Ambiente de Treinamento de Equipes de Perfuração (ATEP) proposto neste trabalho é apresentada na Figura 15. A arquitetura é baseada em uma arquitetura cliente-servidor composta por vários elementos, inspirada no framework H-N2N (BURLAMAQUI, *et al.*, 2006).

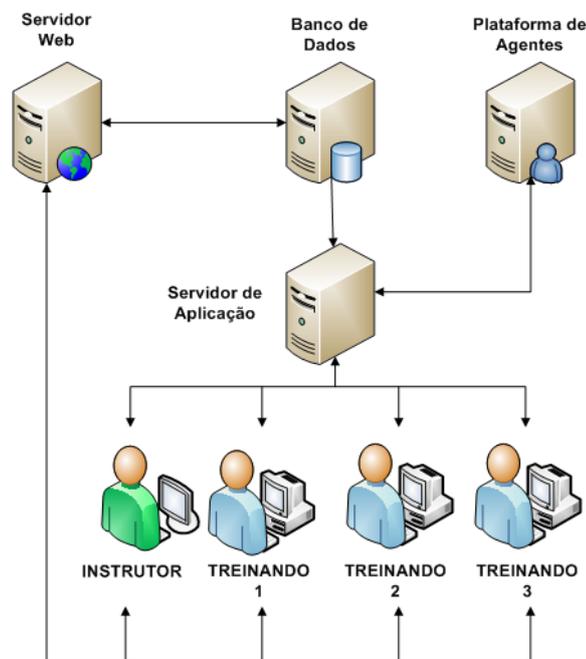


Figura 15 – Arquitetura Geral do Jogo.

Como apresentado na Figura 15, na arquitetura geral do ambiente, o lado do servidor é composto pelos seguintes componentes: servidor Web, servidor de banco de dados, plataforma de agentes e servidor de aplicação.

### **6.3.1.1. Servidor Web**

O servidor Web disponibiliza acesso ao jogo através de navegadores Web. O ATEP inclui informações sobre os projetos, sistema de autenticação (*login*), bate-papo e o jogo propriamente dito. Este ambiente será mais detalhado na Seção 6.4. No lado cliente existem dois tipos de usuários: treinando e instrutor. Existe uma interface específica na aplicação Web para cada tipo de usuário. A interface do instrutor permite o ajuste das configurações do jogo, enquanto que a interface do treinando permite gerenciar as opções relativas à operação da sonda. O servidor Web escolhido para este projeto foi o Apache Tomcat (TOMCAT, 2012).

### **6.3.1.2. Plataforma de Execução dos Agentes**

Ainda no lado do servidor, encontra-se o JADE, que é uma plataforma de execução de agentes que permite criar e iniciar os agentes no jogo. Esta plataforma se comunica diretamente com o servidor de aplicação.

### **6.3.1.3 Banco de Dados**

O servidor de banco de dados é usado para manter persistidos os dados do jogador e do ambiente. O servidor Web e o servidor de aplicação, quando necessário, comunicam-se com o banco de dados. O MySql (MYSQL, 2011) é o servidor de banco de dados escolhido para esse projeto.

### **6.3.1.3. Servidor de Aplicação**

Finalmente, o servidor de aplicação gerencia todo o jogo, compartilhando o ambiente entre vários jogadores, iniciando, parando e persistindo o estado e os dados do jogo. O

componente principal do servidor de aplicação é o *framework* H-N2N, que terá sua arquitetura detalha na próxima subseção.

### 6.3.2. Arquitetura do Ambiente Mutijogador

A Figura 16 mostra a arquitetura do suporte multijogador que foi implementado usando o *framework* H-N2N (BURLAMAQUI, *et al.*, 2006). Este *framework* tem como objetivo ajudar o desenvolvimento de sistemas distribuídos colaborativos, que são capazes de suportar um elevado número de usuários conectados ao mesmo tempo sem perdas na qualidade do serviço. A idéia principal consiste na separação de clientes individuais em grupos e na organização desses grupos em uma estrutura hierárquica.

No *framework* H-N2N, o servidor de aplicação é responsável pelas conexões dos clientes e atua como o servidor principal. Os servidores escravos mantêm a comunicação com os clientes e há um elemento deste tipo para cada cliente. Eles ajudam o servidor de grupo com a tarefa de enviar e receber mensagens dos clientes. Finalmente, o servidor de grupo é um componente que contém uma lista de servidores escravos representando clientes conectados. Ele é responsável por gerenciar a troca de mensagens entre os clientes de um grupo particular, respondendo pelo processamento das operações de filtragem e agrupamento. Além disso, ele pode se comunicar diretamente com servidores de outro grupo.

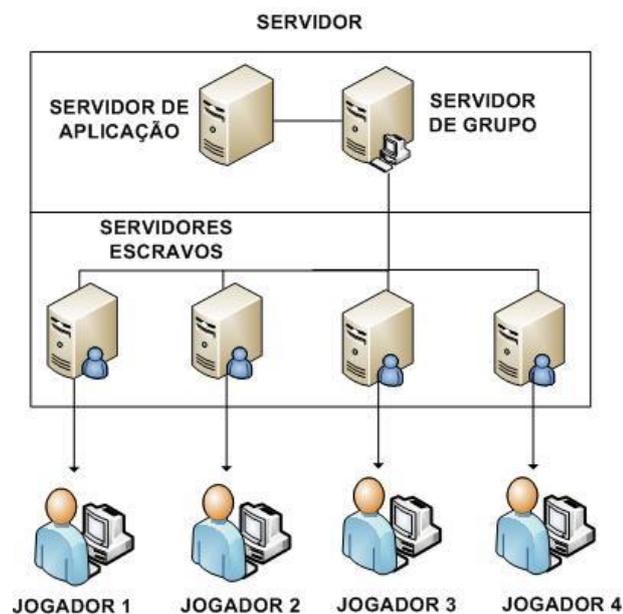


Figura 16 – Arquitetura do H-N2N.

A Figura 17 apresenta um trecho de código que mostra o envio de uma mensagem de um jogador para o ambiente multijogador, causando assim o efeito desejado em todos os jogadores.

```

1439         //DESCER ELEVADOR
1440         else if ((e.getX() > 668 && e.getX() < 790) &&
1441             (e.getY() > 99 && e.getY() < 124)) {
1442             mm.setCatarinaDir(0);
1443             mm.setAcao("descer_elevador");
1444             mm.setValor("1");
1445             enviaPacote();
1446             catarina_dir = 0;
1447         }
1448     }
1449 }
1450
2406
2407 public void enviaPacote() {
2408
2409
2410     mm.setSala(sala);
2411     mm.setUsuario(nome);
2412     mm.setId(sala + System.currentTimeMillis());
2413
2414     userClient.getCommunication().sendMessage(pacote);
2415     if (mm.getAcao().equals("parar_elevador")) {
2416
2417         util.persistirParadaElevador(coluna, catarina, elevador,
2418             cabos_catarina.size(), Integer.parseInt(sala));
2419     } else {
2420         util.persistir(mm.getAcao(), mm.getValor());
2421     }
2422 }
2423 }
2424

```

Figura 17 - Trecho de código-fonte do envio de uma mensagem no ambiente multijogador.

O primeiro trecho de código (linhas 1439 – 1447) faz parte de um método que trata os cliques do mouse no jogo, especificamente, o que trata da descida do elevador. Então, após o clique é atribuído uma ação e um valor para esta ação. Depois surge o método de enviar pacote (linhas 2407 - 2423), onde são atribuídos novos valores às propriedades existentes, tais como: a sala que está enviando o pacote e o nome do usuário. Em seguida, é atribuído um identificador para essa mensagem, que só então é enviada para os outros participantes.

### 6.3.3. Persistência

Na arquitetura mostrada na Figura 15, o elemento banco de dados é formado pelo Sistema de Gerenciamento de Banco de Dados (SGDB) MySQL (MYSQL, 2011). O MySQL é um banco de dados relacional que tem a função de persistência dos dados e utiliza uma linguagem de consulta estruturada (SQL, do inglês *Structured Query Language*) como

interface. No ATEP, a persistência refere-se à gravação de dados não voláteis para uso futuro e será descrita em duas partes. Na primeira serão apresentadas regras e tabelas relacionadas à persistência do jogo, mostrando como um objeto 3D é persistido e reposicionado. A segunda parte mostrará os detalhes referentes à persistência dos dados do portal.

### 6.3.3.1. Persistência do Jogo

Para que o jogo seja considerado persistente o mesmo deve armazenar as múltiplas seções dos usuários, ou seja, a cada ação o jogo deve armazenar os dados do ambiente em um determinado tempo para que este possa ser recuperado em um acesso futuro.

A persistência causada por uma ação no jogo acontece quando o jogador pressiona algum botão e todos os objetos envolvidos na ação são persistidos. Por exemplo, quando o sondador movimentar o elevador e, logo em seguida, o pára, a translação e a rotação dos objetos que mudaram de posição são persistidas no banco de dados. A translação de um objeto é a posição deste objeto no eixo de coordenadas em um plano cartesiano, acrescentando o eixo do Z. A Figura 18 destaca o eixo Z que posiciona o objeto em um ambiente 3D.

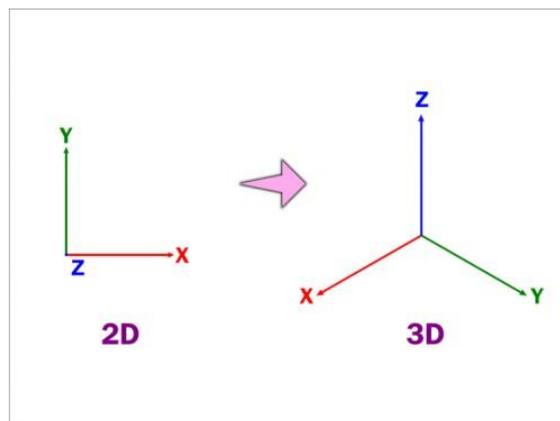


Figura 18 – Eixo Z no plano cartesiano.

Para a persistência desse valor usa-se a função `getTranslationMatrix.getDump` do jPCT, que tem como retorno uma matriz com a posição de um objeto dentro do mapa do jogo.

Já a rotação de um objeto é a maneira como o objeto está rotacionado no ambiente 3D, tendo como relação o mundo (*world*<sup>5</sup>) do jPCT, onde são adicionados todos os objetos da cena. Este valor é armazenado a partir da função *getRotationMatrix.getDump*, que retorna uma matriz com a rotação de um determinado objeto na cena. A Figura 19 mostra trecho do código para persistência de uma parte recorrente do jogo, que, neste caso, é a parada do elevador.

No final da cena de movimentação do elevador é chamado o método “persistirParadaElevador” (linha 344), que (i) primeiro atualiza as variáveis que definem a quantidade de cabos que no jogo ajustará a altura do guincho e a quantidade de tubo na coluna de perfuração (linhas 350 - 352); (ii) depois persiste os objetos catarina e elevador (linhas 360 - 364), guardando suas rotações e transações; (iii) nesse momento chama o método utilitário que transforma a matriz de rotação/translação em uma variável de texto (*string*) que será persistida no banco de dados (linha 46); (iv) finalmente, persiste a coluna, verificando todos os objetos na coluna de perfuração, excluindo as retiradas das colunas e atualizando as posições dos equipamentos que permaneceram (linha 366).

A persistência do jogo ocorre em cada ação, entretanto, o posicionamento dos objetos persistidos em um jogo acontece no início da seção, seja em uma nova partida ou em uma continuação. Durante o jogo, os objetos no ambiente multijogador são sincronizados através de mensagens, ou seja, a cada ação é enviada uma mensagem para os outros jogadores com a nova situação do ambiente. A Figura 20 mostra o trecho do código que refaz a coluna de perfuração no início de uma seção do jogo. O trecho de código inicia com a declaração de uma variável do tipo *Object3D*<sup>6</sup> (linha 804), em seguida, consulta os objetos de um projeto (linhas 810 - 813). Finaliza fazendo uma repetição de operação de acordo com a quantidade de registros retomados (linhas 814 – 822). A partir daí, (i) é criada uma instância do objeto 3D (*segmento3.3ds*<sup>7</sup>), em seguida, (ii) atribui suas rotações e transações para posicionamento no mundo e, finalmente, (iii) adiciona os elementos na coluna e no mundo.

---

<sup>5</sup> A classe *world* é a classe mais importante no jPCT. O *world* contém objetos e fontes de luz que definem o cenário para jPCT, em outras palavras, é a classe principal que mantém tudo junto.

<sup>6</sup> *Object3D* é uma classe de objetos de 3 dimensões localizada no pacote *com.threed.jpct*. Um objeto de 3 dimensões construído por esta classe é geralmente adicionado a uma instância do mundo para renderização (usando *World.addObject ()*).

<sup>7</sup> A extensão *3ds* é um dos formatos de arquivo usados pelo 3ds Max Autodesk para modelagem, animação e renderização 3D, formato que também pode ser criado e editado no Blender.

```

46 public String geraDump(float[] valores) {
47     String retorno = "";
48
49     for (int i = 0; i < valores.length; i++) {
50         retorno += String.valueOf(valores[i]) + "/";
51     }
52     return retorno;
53
54 }
55
56 public float[] pegaDump(String valores) {
57     float[] retorno = new float[16];
58     String[] array = valores.split("/");
59
60     for (int i = 0; i < array.length; i++) {
61         retorno[i] = Float.parseFloat(array[i]);
62     }
63     return retorno;
64
65 }
344 public void persistirParadaElevador(Coluna coluna, Catarina catarina, Elevador elevador,
345     int qtdCabos, int projeto_id) {
346     try {
347         Class.forName(driver);
348         Connection con = (Connection) DriverManager.getConnection(URL, USE, SENHA);
349         Statement stmt = (Statement) con.createStatement();
350         String str = "update persistencia_variaveis set ";
351         str += "quantidade_cabos = " + qtdCabos + ", tubos_adicionados = " + coluna.getSize() + " ";
352         str += " where projeto_id=" + projeto_id;
353         stmt.executeUpdate(str);
354     } catch (SQLException ex) {
355         System.out.println(ex.getMessage());
356     } catch (ClassNotFoundException ex) {
357         // Logger.getLogger(AgentePersistente.class.getName()).log(Level.SEVERE, null, ex);
358     }
359
360     persistirObjeto("catarina", geraDump(catarina.getTranslationMatrix()),
361         geraDump(catarina.getRotationMatrix()), projeto_id, 1, "ambiente");
362     persistirObjeto("elevador", geraDump(elevador.getTranslationMatrix()),
363         geraDump(elevador.getRotationMatrix()),
364         projeto_id, 1, "ambiente");
365
366     persistirColuna(coluna, projeto_id);
367 }

```

Figura 19 – Trecho do código-fonte para persistir as posições dos objetos.

```

803
804 Object3D obj_segimento_aux = null;
805 Class.forName(driver);
806 Connection con = (Connection) DriverManager.getConnection(URL, USE, SENHA);
807 //Verifica Id usuario
808 Statement stmtVerifica = (Statement) con.createStatement();
809
810 ResultSet resultSetVerifica = stmtVerifica.executeQuery("select * "
811     + "from persistencia_objetos where nomeObjeto = 'tubo' "
812     + "and projeto_id=" + sala + " and local = 'coluna' order by posicao asc");
813
814 while (resultSetVerifica.next()) {
815     obj_segimento_aux = new ComponentePlataforma("segmento3.3ds").getObject();
816
817     obj_segimento_aux.getRotationMatrix().setDump(util.pegaDump(resultSetVerifica.getString("rotacao")));
818     obj_segimento_aux.getTranslationMatrix().setDump(util.pegaDump(resultSetVerifica.getString("transacao")));
819
820     coluna.addComponente(obj_segimento_aux);
821     world.addObject(obj_segimento_aux);
822 }
823 resultSetVerifica.close();
824 stmtVerifica.close();
825
826

```

Figura 20 – Trecho do código-fonte para posicionar os objetos persistidos.

Existe um conjunto de tabelas que são usadas na persistência do jogo, que estão relacionadas conforme mostrado na Figura 21. Para efeito de organização, foram omitidas algumas tabelas e campos e o diagrama foi dividido em duas partes, uma contemplando as

tabelas relacionadas com a persistência das partidas de jogo e outro com os dados do ambiente Web do ATEP.

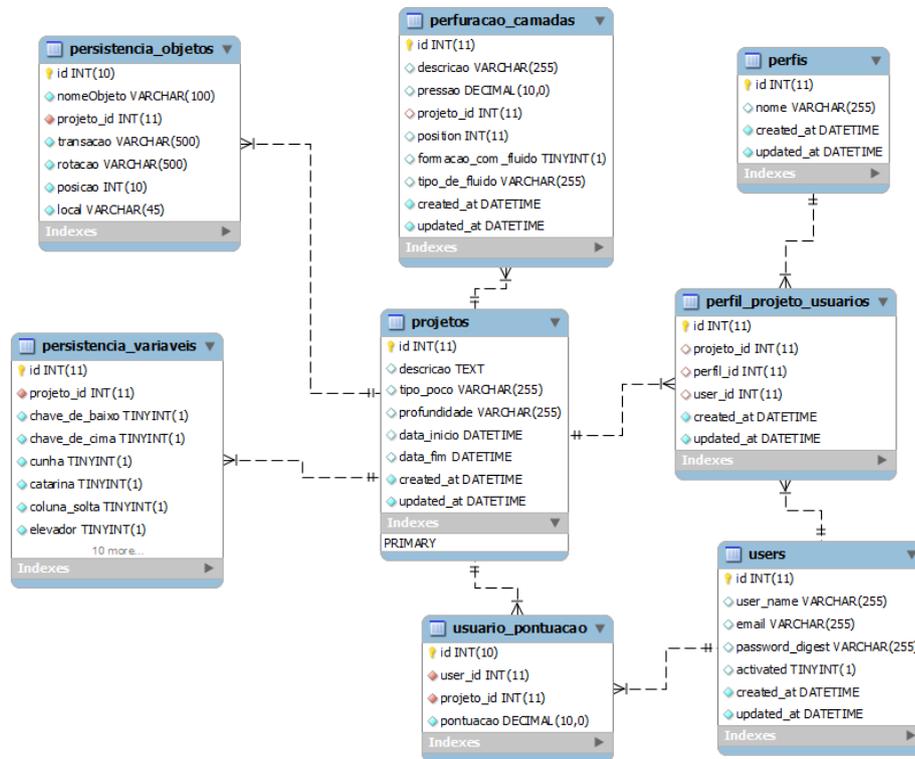


Figura 21 – Modelo E/R da persistência do jogo.

A Tabela 2 apresenta a descrição das tabelas apresentadas no diagrama da Figura 21.

Tabela 2 – Tabelas relacionadas à persistência do jogo.

| Nome da Tabela         | Descrição  |
|------------------------|--|
| projetos               | Contém os dados relacionados ao projeto.   |
| perfuracao_camada      | Tabela responsável por guardar as informações das camadas perfuradas de um determinado projeto.          |
| persistencia_objetos   | Tabela responsável por armazenar o posicionamento dos objetos em um determinado projeto.                 |
| persistencia_variaveis | Tabela responsável por armazenar dados sobre o ambiente e a persistência do estado de algumas variáveis. |
| users                  | Tabela de usuários.  |
| usuario_pontuacao      | Tabela que guarda a pontuação de um usuário em um determinado projeto.                                   |
| perfis                 | Tabela de perfis de usuário no jogo: Plataformista, Sondador e   |



Tabela 3 – Descrição das tabelas do ambiente Web do ATEP.

| <b>Nome da Tabela</b> | <b>Descrição</b>   |
|-----------------------|--|
| projetos              | Contém os dados relacionados ao projeto.   |
| tarefas               | Armazena as tarefas de um projeto.   |
| parcial_relatorio     | Tabela responsável por armazenar relatórios de tarefas de um projeto.                          |
| final_relatorio       | Tabela responsável por armazenar relatório final de um usuário em um projeto.                  |
| roles                 | Tabela com os papéis dos usuários no ATEP, que podem ser: Administrador, Instrutor, Treinando. |
| roles_users           | Tabela que relaciona usuários e seus papéis no ATEP.   |
| parametros            | Guarda informações de configurações do ATEP no formato código / valor.                         |
| questoes              | Cadastro de perguntas que serão usadas no decorrer do jogo para avaliar os treinandos.         |
| questoes_alternativas | Cadastro das alternativas de resposta de uma questão.  |

#### 6.4. PORTAL WEB

No servidor Web mostrado na arquitetura apresentada na Figura 15 se encontra o portal Web do ATEP. Esse portal concentra todas as funcionalidades do ambiente de treinamento. Para isso, o Portal Web deve abordar os aspectos relacionados à PBL, fornecendo espaço para avaliação dos treinandos. Esta avaliação é realizada através de preenchimento de formulários on-line e ferramentas de comunicação assíncronas. O Portal Web também provê uma interface para que o instrutor possa auxiliar, responder dúvidas dos treinandos e revisar os relatórios gerados a partir do preenchimento dos formulários de avaliação.

O portal possui três áreas, correspondentes aos três perfis de acesso:

1. Treinando: O usuário com este perfil é um treinando participando do treinamento. Ele tem acesso (i) às salas de jogos, (ii) à área com suas estatísticas e resultados, como também (iii) à área para preenchimento de

formulários de avaliação. Assim, o usuário com este perfil está apto a (a) participar de jogos de treinamento, realizando suas tarefas para conseguir avançar no jogo, (b) acessar conteúdos didáticos disponíveis no ambiente e (c) realizar as avaliações exigidas pela PBL.

2. Instrutor: O usuário com este perfil está apto a (i) criar treinamentos, (ii) criar tarefas específicas de acordo com a função executada por cada treinando em uma sonda de perfuração, (iii) monitorar as atividades dos treinandos, (iv) avaliar o desempenho dos treinandos, (v) ajustar parâmetros físicos do módulo de simulação do jogo e (vi) solicitar geração de relatórios de avaliação de desempenho, segundo a PBL.
3. Administrador: O usuário com esse perfil acessa todos os dados de configuração do sistema, podendo alterar as características do mesmo.

O sistema irá apresentar as interfaces de acordo com o perfil do usuário. Elas utilizam o mesmo padrão de desenho (*layout*), diferenciando apenas nas opções oferecidas. Enquanto o perfil de treinando tem acesso a uma área (módulo do treinando) onde pode visualizar suas informações e opções de jogo, os instrutores têm acesso às opções de gerenciamento de informações do projeto (módulo do instrutor). A seguir são apresentadas as principais características do ambiente de treinamento.

#### **6.4.1. Modulo do Instrutor**

##### **6.4.1.1. Cadastro do Projeto**

O instrutor é responsável por fazer o cadastro básico do projeto, escolhendo as características iniciais dos projetos e definindo o cronograma com suas tarefas e prazos. Após definir as características do projeto, o instrutor libera a sala de jogo para ser acessada pelos treinandos. Os treinandos podem fazer alterações e ajustes logo depois do cadastro do projeto e suas tarefas.

#### **6.4.1.2. Relatórios de Acompanhamento**

O instrutor tem acesso aos relatórios parciais e finais dos treinandos, podendo adicionar observações no espaço apropriado. Estes relatórios fornecem retorno sobre o desempenho dos treinandos, que ajuda o instrutor a (i) ajustar as características do projeto, alterando metas e prazos, e (ii) adequar os desafios do jogo, variando quantidade e complexidade.

#### **6.4.1.3. Cadastro de Perguntas e Respostas**

O Instrutor pode acessar o módulo de cadastro de questões relativas às atividades de perfuração de poços de petróleo. Este mecanismo funciona como um banco de questões que serão utilizadas no decorrer do jogo. O usuário inicialmente cadastra a pergunta fornecendo o enunciado da questão e a resposta correta. Em seguida, tem que cadastrar quatro alternativas de repostas que serão apresentadas ao jogador.

### **6.4.2. Módulo do Treinando**

#### **6.4.2.1. Acesso à área de projetos**

Logo que o usuário entra no sistema com o perfil de treinando, o sistema apresenta uma interface para seleção dos projetos que estão disponíveis para o usuário participar. Nesta interface são apresentados tanto os projetos já iniciados, onde o usuário irá continuar suas atividades, como os projetos já finalizados pelo treinando. O ambiente permite também o acesso a salas de jogos que possibilitam treinar habilidades específicas sem a necessidade de participar de um projeto completo.

Cada projeto criado possui uma sala de jogo associada. Os projetos contêm as características do poço, as atividades a serem desenvolvidas e seus respectivos prazos. É

possível, após o término das atividades, a alteração das características do projeto, estendendo prazos e reavaliando operações.

#### **6.4.2.2. Painel do Projeto**

Devido ao ambiente ser baseado em projetos, o portal do ATEP apresenta um painel para consolidar as informações do projeto. Este painel está disponível tanto para o treinando como também para o instrutor e contém as seguintes funcionalidades:

- Dados do Projeto (nome, descrição, data inicial, data final, etc);
- Visualização das tarefas com descrições e prazos, com botões de acesso para edição;
- Acesso para preenchimento dos relatórios das tarefas e do relatório final;
- Quadro de mensagens para comunicação assíncrona entre os participantes;
- Lista de participantes do projeto.

#### **6.4.2.3. Relatórios pós-tarefas para avaliação**

O ambiente proposto também oferece recursos destinados a ajudar os instrutores na tarefa de monitorar o progresso dos treinandos e interpretar seus conhecimentos, habilidades e ações, tendo em vista as mudanças esperadas no comportamento do treinando, propostas nos objetivos do projeto. Seguindo as recomendações da PBL, depois que o treinando terminar de executar algumas das tarefas no jogo, ele é convidado, através de uma interface específica do jogo, a preencher um relatório de progresso com os seguintes pontos:

- Eu pesquisei...;
- Eu realizei os seguintes passos...;
- Eu descobri que...;
- Eu aprendi a realizar os seguintes procedimentos... / adquiri as seguintes competências...;

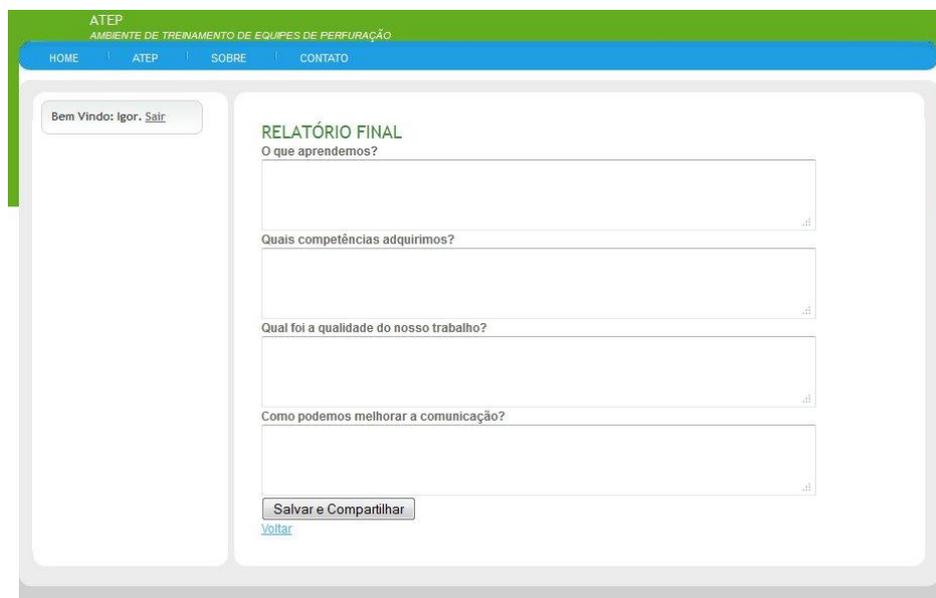
- Como resultado da minha pesquisa, eu acho que devemos fazer as seguintes alterações no projeto....

#### 6.4.2.4. Relatório final do Projeto

Cumprindo os requisitos de avaliação propostos pela PBL, no final do projeto todos os treinandos são convidados a um momento de reflexão, onde eles devem preencher um formulário *on-line* com perguntas focadas no conteúdo e nos resultados. As questões são:

- O que aprendemos?
- Quais as competências que adquirimos?
- Qual foi a qualidade do nosso trabalho?
- Como podemos melhorar a colaboração?

A fim de compartilhar o conhecimento, todos os resultados são disponibilizados para os outros treinandos através do painel do projeto. A Figura 23 apresenta a tela de preenchimento de relatório final.



A captura de tela mostra a interface web do sistema ATEP (Ambiente de Treinamento de Equipes de Perfuração). No topo, há um menu de navegação com as opções HOME, ATEP, SOBRE e CONTATO. À esquerda, uma barra lateral exibe uma mensagem de boas-vindas: "Bem Vindo: Igor, Sair". O conteúdo principal da página é o formulário "RELATÓRIO FINAL", que contém quatro perguntas abertas para serem respondidas em campos de texto:

- O que aprendemos?
- Quais competências adquirimos?
- Qual foi a qualidade do nosso trabalho?
- Como podemos melhorar a comunicação?

Na base do formulário, há um botão "Salvar e Compartilhar" e um link "Voltar".

Figura 23 – Formulário do Relatório Final de Avaliação.

#### 6.4.2.5. Ferramentas de Comunicação

A ferramenta de comunicação síncrona disponível no jogo permite aos jogadores conversarem entre si e com os instrutores em modo texto. Ela apresenta uma interface que exhibe, juntamente com o comentário, o nome de quem o inseriu. Para cada partida de jogo existente no ambiente, é criada, automaticamente, uma Sala de Conversação, onde só poderão participar de conversações os jogadores da atual partida e os instrutores. Estes últimos podem receber também mensagens dos agentes de software, comunicando alguma ação relevante realizada por um treinando. A Figura 24 apresenta a sala de conversação, que fica sempre visível do lado esquerdo da interface do ambiente virtual de treinamento, permitindo assim comunicação rápida e em tempo real.

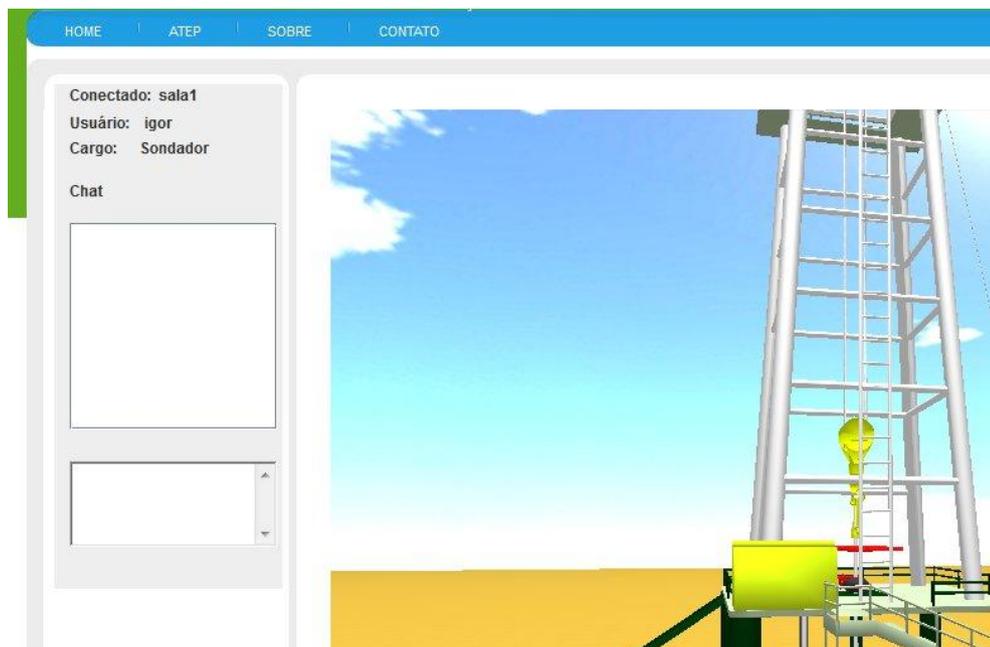


Figura 24 – Área para comunicação instantânea no ATEP.

O ambiente também disponibiliza uma ferramenta de comunicação assíncrona através de um quadro de mensagens localizado no painel do projeto. Qualquer participante pode deixar um recado, sugestões de alterações no projeto, lembretes, entre outras ações, e o sistema irá registrar o nome do usuário, o horário e a mensagem. Cada mensagem adicionada no quadro de mensagens será enviada por email para os participantes.

### **6.4.3. Modulo de Administração**

#### **6.4.3.1. Cadastro de Usuários**

É a área no portal responsável pelo cadastro dos participantes do ATEP. Na hora da realização do cadastro, o administrador pode optar pelo perfil que deseja atribuir ao usuário, que pode ser: Administrador, Instrutor ou Treinando.

#### **6.4.3.2. Configuração do Ambiente**

O administrador pode configurar o ambiente alterando vários parâmetros, como, por exemplo:

- **Endereço IP / URL:** Este campo indica em qual servidor o sistema está instalado. Essa informação vai para as aplicações clientes.
- **Portas:** Este campo informa as portas liberadas para a troca de mensagens no ambiente multijogador.
- **E-mail do sistema:** Este campo permite configurar o e-mail usado para enviar mensagens para os participantes do jogo.
- **Conexão do banco de dados:** Este campo guarda as informações da conexão do banco de dados.

### **6.5. CENÁRIO E JOGABILIDADE**

Um cenário do jogo pode ser descrito como segue. O instrutor acessa o ambiente fornecendo seu *login* e senha. O sistema então valida as informações, verificando o tipo de usuário (jogador ou instrutor), e redirecionando-o para sua respectiva interface.

No Cadastro do Porjeto, o instrutor fornece informações sobre o projeto, preenchendo um formulário com os parâmetros do projeto do poço de petróleo, como: tipo de poço, profundidade, tipo de solo e outras características. O instrutor também preenche a lista de tarefas do projeto, com os prazos, os objetivos e outros parâmetros. Em seguida, o instrutor cria uma sala de jogo com, no mínimo, dois plataformistas, um sondador e um torrlista.

No decorrer do jogo, o instrutor pode monitorar as atividades dos treinandos, usando uma interface que permite tanto agir como um observador como interagir com o jogo causando situações adversas.

O usuário, uma vez identificado pelo sistema como um treinando e redirecionado para a interface adequada, tem a opção de iniciar um novo jogo, previamente definido pelo instrutor, ou de retornar a um jogo em que o treinando já está participando.

A primeira fase do jogo contempla um passo importante antes da perfuração que é a escolha e a avaliação dos tubos, que são elementos que serão usados na coluna de perfuração. O jogador tem que escolher tubos perfeitos, porque, se houver algum com conexões riscadas e desgastadas inserido na coluna de perfuração, pode acarretar na quebra do tubo e consequente quebra da coluna de perfuração, prejudicando assim a operação. A

Figura 25 mostra a tela de seleção dos tubos de perfuração.

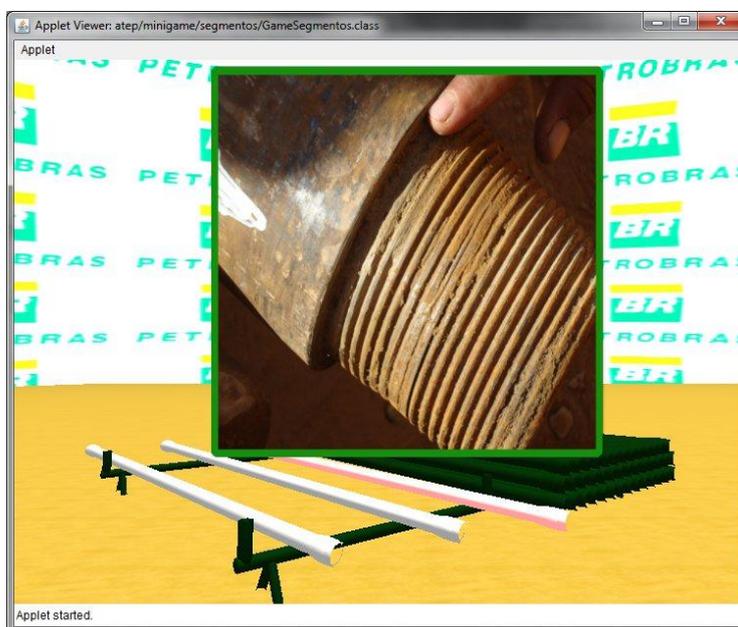


Figura 25 – Escolha de tubos para a operação.

O desafio mais comum é a conexão do tubo. Esta operação adiciona novos elementos tubulares na coluna de perfuração. Para realizar esta tarefa, o jogador seleciona uma ferramenta que evita a queda da coluna no poço, denominada *cunha*, e a coloca no lugar apropriado. Depois de encaixá-la, o treinando aguarda o guincho subir até ele estar pronto

para adicionar um novo segmento. Usando o painel de controle, o treinando adiciona um novo tubo de perfuração. Uma vez concluída esta etapa, o guincho descerá a coluna de perfuração até alcançar a mesa rotativa. Então o plataformista realiza a conexão.

Se houver algum problema na coluna, seja para trocar alguma peça, como, por exemplo, a broca, ou trocar algum tubo defeituoso, é realizada a operação de manobra, que consiste na retirada da coluna. A operação inversa é denominada descida da coluna.

A Figura 26 destaca a cunha e as chaves flutuantes, que são ferramentas de apoio à operação de conexão. A conexão ocorre quando o guincho move a extremidade inferior do tubo erguido até se juntar com a extremidade superior da coluna de perfuração. Em seguida, o jogador usa as chaves para conectar o novo tubo. Assim, a etapa final é retirar a cunha. Então, o guincho desce a coluna de perfuração com o novo elemento adicionando. Para evitar problemas, os jogadores devem estar atentos aos indicadores de pressão e para o dispositivo preventivo de *Blowout*. Problemas podem ser gerados pelos agentes para verificar o tempo de resposta e decisão do treinando.

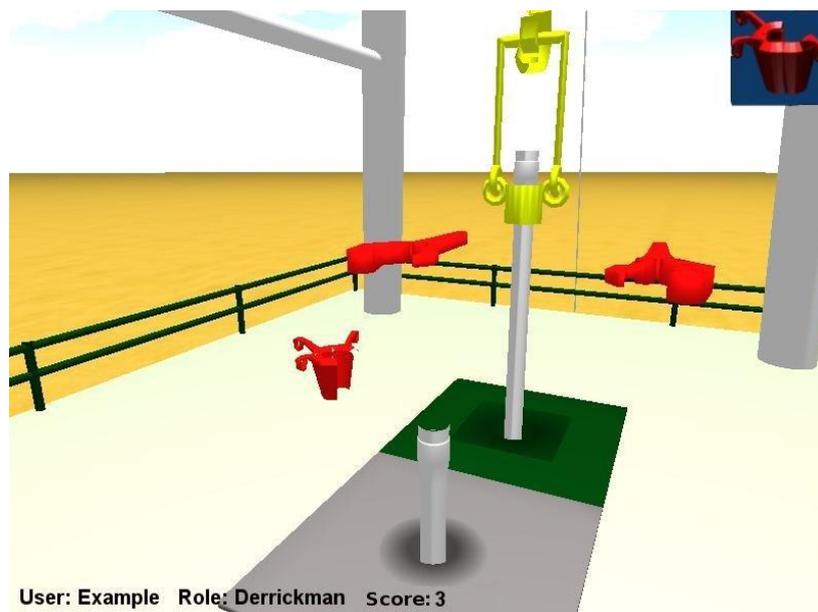


Figura 26 – Cunha e Chaves Flutuantes em destaque na plataforma.

### 6.5.1. Pontuação

O funcionamento da pontuação no ambiente de jogo funciona avaliando cada ação do jogador no ambiente, tais como, movimentos desnecessários, erro na operação e demais ações

que poderão acarretar na perda de pontos. Quando ocorre uma situação adversa, é iniciado um cronômetro e o treinando terá que responder questões no tempo correto, correndo o risco de perder pontos e agravar o problema caso não consiga.

A pontuação do acertos ocorre da seguinte forma:

- Realizar movimento correto na operação: Adição de 100 pontos;
- Responder questionário de resolução de situação adversa: Adição de 50 pontos por questão correta dentro do tempo.

Por outro lado, a penalidade na pontuação devido a erros ocorre do seguinte modo:

- Realizar movimento desnecessário durante a operação: Diminuição de 50 pontos por movimento desnecessário. Numa situação real, movimentar ferramentas sem necessidade pode por em risco a integridade de outro colaborador.
- Ultrapassar o tempo do cronômetro nas resoluções de questões de situações adversas: Perda dos pontos conquistados..
- Cometer erros de operação: Diminuição de 50 pontos por erro de operação. Por exemplo, caso o treinando desça a coluna com a cunha na posição fechada.



Figura 27 – Tela de resolução de situação adversa.

A Figura 27 apresenta uma tela de resolução de situação adversa. Quando ocorre algum problema, o treinando terá que resolver algumas questões em um determinado tempo. Esse tempo vai depender da gravidade da situação, sendo diminuído em uma situação grave onde os treinandos terão que responder rapidamente o questionário para superarem a situação.

### 6.5.2. Jogabilidade

A jogabilidade é um quesito muito importante em um jogo, pois é a maneira em que o usuário interage e obtém informações do jogo corrente. Para realizar as ações do jogo no ATEP, a interface oferece três maneiras de interação, são elas:

- Seleção de algum objeto na cena: Ao seleciona um objeto na cena, o mesmo pode realizar alguma ação ou apresentar uma descrição do item selecionado, auxiliando assim o aprendizado.
- Botões de ação: A interface fornece botões de ações através de um painel que, dependendo do papel do jogador na operação de perfuração, apresenta diferentes funções.
- Teclas: O teclado pode ser usado para executar determinadas ações.

A Figura 28 apresenta alguns painéis disponíveis na interface do jogo, dentre os quais se destaca na figura:

- a) Painel acionado pela seleção de alguma ferramenta no ambiente 3D, apresentando assim a descrição do objeto.
- b) Botões de ações do plataformista que, da esquerda para direita, servem para: (i) colocar a cunha; (ii) retirar a cunha; (iii) soltar a coluna do elevador; (iv) acionar a chave flutuante esquerda; (v) acionar a chave flutuante direita; e, por ultimo, (vi) retirar as chaves. Também possui informações com nome do usuário, papel deste na operação e pontuação.
- c) Além de painéis com teclas de ação, a interface do jogo mostra estatística e parâmetros do projeto, como, por exemplo, profundidade perfurada, pressão do poço, etc.

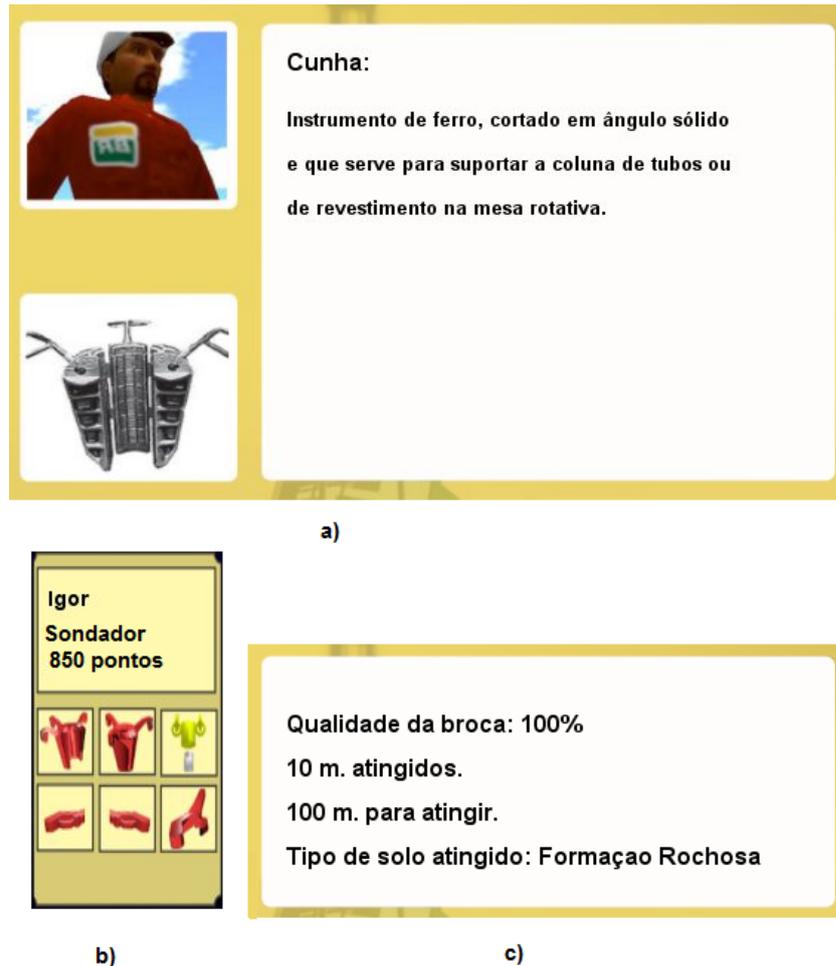


Figura 28 – Detalhes da jobabilidade do ATEP.

## 6.6. AGENTES E SISTEMA MULTIAGENTE

Neste trabalho, os agentes foram modelados para cooperar uns com os outros, exercendo uma das seguintes funções:

- Sondador (*driller*): Desenvolve as operações de perfuração, sondagem e recolhimento de amostras em poços de petróleo e gás, operando guinchos de sondas de perfuração e outros instrumentos específicos. Além disso, o sondador observa o programa estabelecido para possibilitar as análises do material recolhido.
- Torrlista (*derrickman*): Efetua o manuseio dos tubos a partir do topo da torre, executando trabalhos de apoio nas manobras de decida e retirada da coluna de

perfuração, auxiliando o plataformista a acionar e controlar as bombas de lama. O torrlista também monitora o fluxo do fluido de perfuração, entre outras atividades.

- Plataformista (*roughneck*): É responsável pela conexão de tubos à coluna de perfuração e sondagem; pela conexão de brocas na coluna, posicionando-as na mesa rotativa; pela preparação da coluna pronta para as operações de sondagem; pela lubrificação e abastecimento dos componentes de guincho e motores durante as operações de sondagem; pelo auxílio ao torrlista nas tarefas de adição de elementos químicos para o tratamento da lama, entre outras tarefas.

Essa cooperação acontece quando se inicia uma partida faltando jogadores em alguma dessas funções. Cada sala de jogo suporta quatro participantes, sendo dois plataformistas, um torrlista e um sondador, portanto os agentes assumem os papéis não contemplados na partida.

Além desses, existem agentes competitivos no ambiente que representam o comportamento das ferramentas, do ambiente e da estrutura da plataforma e do poço. Esses agentes estão sempre ativos.

Para criação desse ambiente multiagente, o passo inicial da modelagem foi a elaboração do modelo de tarefas por meio de diagramas e modelos textuais relacionados às tarefas do jogo. Esses artefatos, baseados em modelos MAS-CommonKADS+ (OLIVEIRA, 2010), descrevem os recursos que o jogo deve disponibilizar.

A Figura 29 apresenta o modelo de tarefas descrevendo todas as atividades que devem ser realizadas para alcançar uma determinada meta. Todas as atividades são demonstradas em diagramas, descrevendo suas respectivas subtarefas, que serão realizadas para completar o objetivo. Elas também são especificadas em *templates* textuais, que descrevem suas entradas, saídas, pré-condições, pós-condições e objetivos.

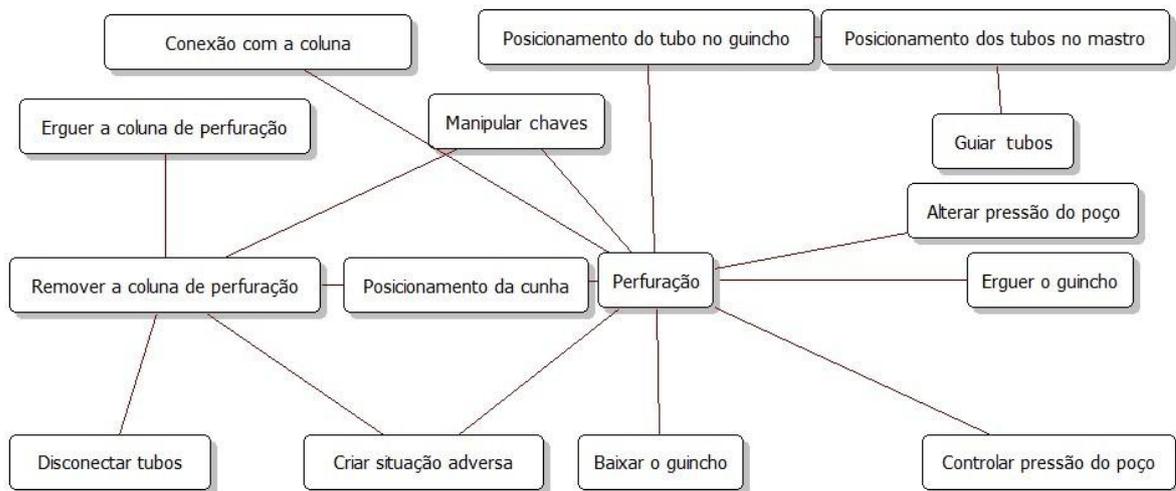


Figura 29 – Diagrama de Tarefas.

Seguindo os passos sugeridos pela metodologia MAS-CommonKADS+, deve-se realizar a captura dos objetos do sistema e representá-los no Modelo de Recursos. Esses objetos são descobertos através da análise dos requisitos e das tarefas do sistema. Em alguns casos, determinadas partes de um sistema não necessitam de autonomia, por exemplo, tarefas relativas à configuração do ambiente e de cadastro básico incluídas no diagrama de classes. Neste modelo são descritos também os recursos, como, por exemplo, o banco de dados. O diagrama de classe para o modelo de recursos é demonstrado na Figura 30.

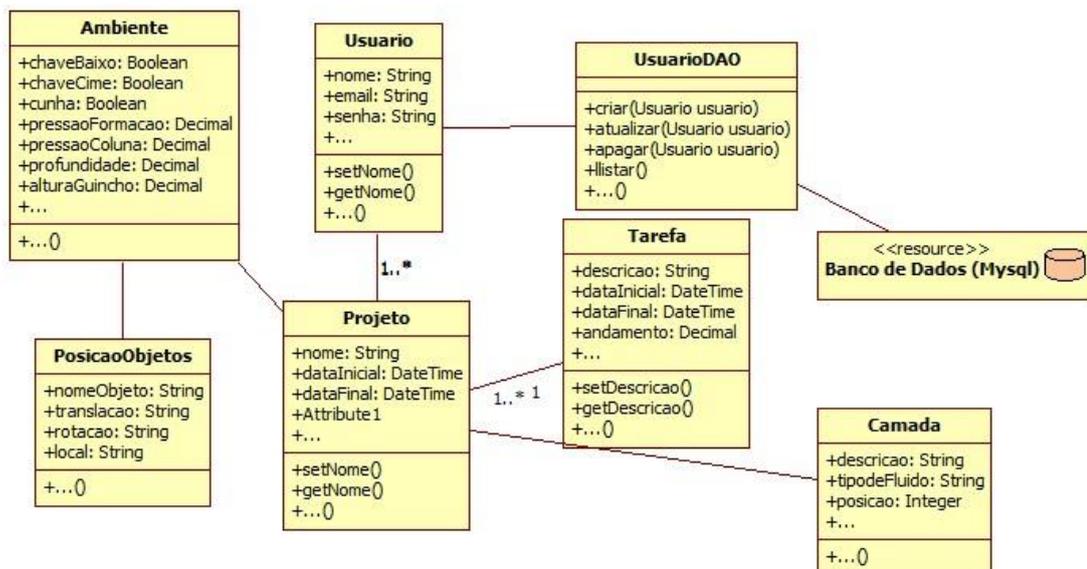


Figura 30– Modelo de Recursos.

O diagrama da Figura 31 apresenta a estrutura hierárquica dos agentes. Verifica-se que o agente Cliente e o agente DF herdam diretamente da classe *Agent* (do pacote *jade.core.Agent*). Já os outros agentes (Poço, Torrista, Plataformista, Plataforma e

Coordenador) herdam de uma classe abstrata chamada de *AgenteBase* que contém o código para acesso ao banco de dados e acesso ao ambiente multijogador.

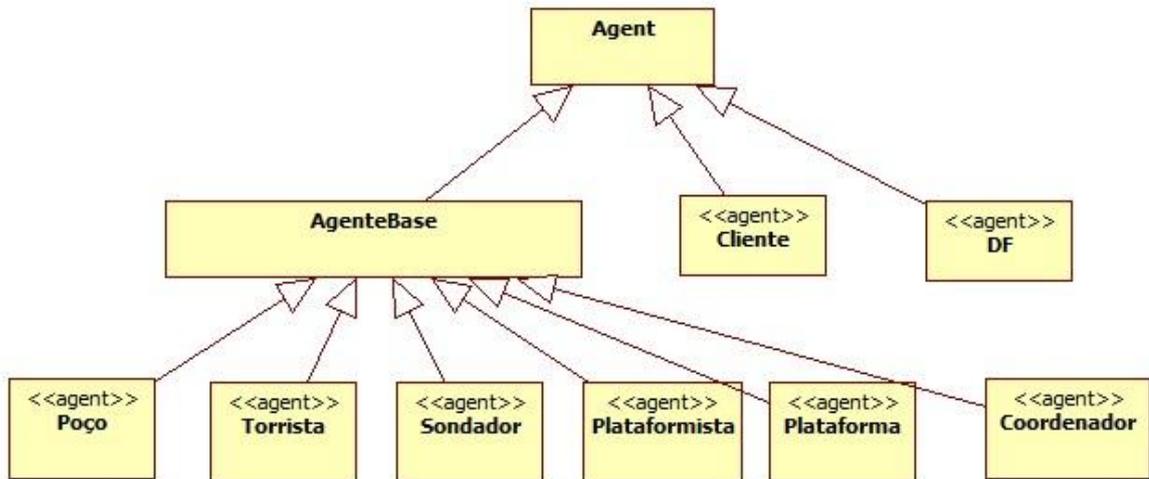


Figura 31– Diagrama de Classes Agentes.

Seguindo o modelo proposto pela metodologia, o passo seguinte é a descoberta dos papéis que cada agente pode exercer no ambiente. Esses papéis são responsáveis por desempenhar as tarefas existentes no modelo de tarefas. O modelo de Papéis, mostrado na Figura 32, é responsável por definir quais agentes serão responsáveis por cada tarefa apresentada. Este modelo define também a arquitetura de agentes, seus objetivos e características, tais como dados de entrada, condições de ativação do agente e tipos de informações disponíveis.

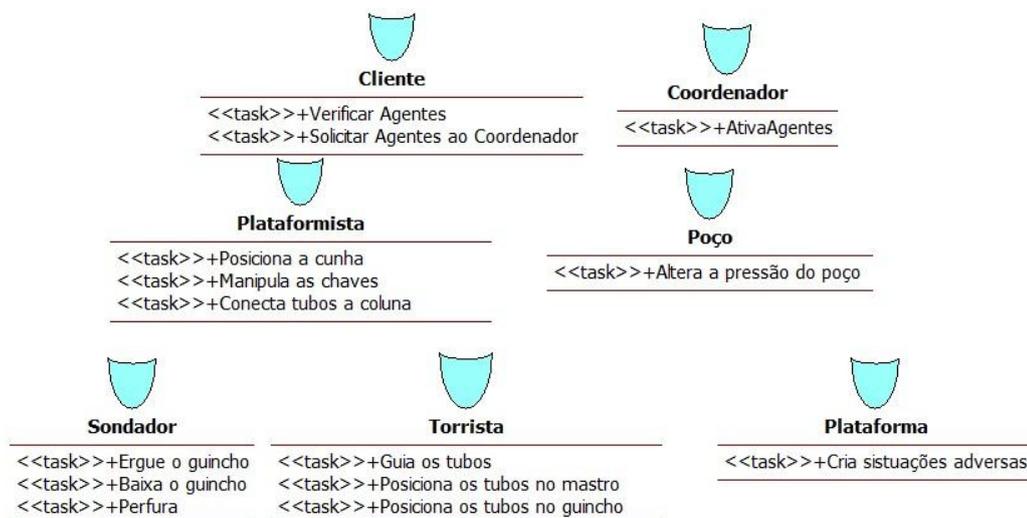


Figura 32 – Modelo de Papéis.

Em seguida, depois de definir os papéis que compõem o sistema, deve ser desenvolvido um diagrama contendo a organização dos agentes. Neste diagrama estarão contidas todas as hierarquias entre os papéis e os grupos de papéis. Este modelo deve ser dividido em organização interna e externa. Na organização interna serão apresentados os relacionamentos entre papéis do sistema, enquanto que na organização externa será demonstrado o relacionamento entre os grupos de agentes. O modelo de organização pode ser visualizado Figura 33.

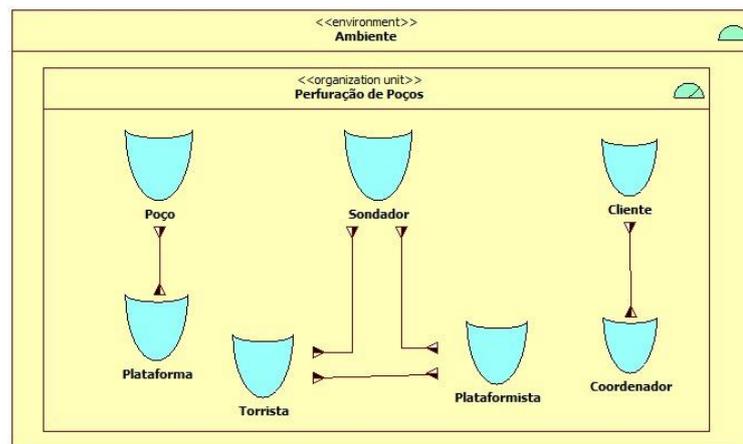


Figura 33 – Modelo de Organização dos Agentes.

Nas próximas subseções serão apresentados os modelos dos agentes e sua arquitetura, que neste trabalho foram divididos em três grupos: Agentes Cooperativos, Agentes de Suporte e Agentes Utilitários.

### 6.6.1. Detalhes dos Agentes Cooperativos

Os agentes cooperativos contemplam o grupo formado pelos agentes Sondador, Plataformista e Torrlista, que são ativados quando há a necessidade de completar uma equipe no projeto. As tabelas 4, 5 e 6 apresentam os modelos textuais dos agentes Sondador, Plataformista e Torrlista, respectivamente.

Tabela 4 – Modelo textual do agente Sondador.

|                          |
|--------------------------|
| <b>Agente: Sondador.</b> |
|--------------------------|

|   |
|---|
| <b>Tipo de Agente</b>   |
| Agente reativo baseado em modelo.   |
| <b>Objetivo</b>   |
| Realizar movimentação de guincho e ativar mesa rotativa.  |
| <b>Parâmetros de Entrada</b>  |
| Informações sobre o estado da plataforma (guincho, coluna, mesa rotativa, etc.).  |
| <b>Parâmetros de Saída</b>  |
| Conduzir o guincho.   |
| <b>Condição de Ativação</b>   |
| Quando o agente percebe o ambiente com a necessidade de movimentação do guincho e o posicionamento adequado da cunha.   |
| <b>Condição de Finalização</b>  |
| Quando o agente envia o comando para movimentar o guincho.  |
| <b>Informação Associada</b>   |
| Tem comportamento cíclico, percebendo e consultando o histórico do ambiente em intervalos preestabelecidos.   |
| <b>Descrição</b>  |
| O agente Sondador tem como objetivo mover a coluna através do guincho, visando fazer a conexão de $n$ segmentos. Para mover a coluna, é necessário o posicionamento correto da cunha. |

Tabela 5 – Modelo textual do agente Plataformista.

|  |
|--|
| <b>Agente: Plataformista.</b>  |
| <b>Tipo de Agente</b>  |
| Agente reativo baseado em modelo.  |
| <b>Objetivo</b>  |
| Realizar conexão de tubos.   |
| <b>Parâmetros de Entrada</b>   |
| Informações sobre o estado da plataforma (guincho, coluna, mesa rotativa, etc.). |
| <b>Parâmetros de Saída</b>   |
| Conduzir as chaves flutuantes e cunha.   |
| <b>Condição de Ativação</b>  |

|  |
|--|
| Quando o agente percebe o ambiente com um novo segmento a ser adicionado.  |
| <b>Condição de Finalização</b><br>Quando o agente envia o comando para movimentar as chaves flutuantes e os tubos são conectados.  |
| <b>Informação Associada</b><br>Tem comportamento cíclico, percebendo e consultando o histórico do ambiente em intervalos preestabelecidos.   |
| <b>Descrição:</b><br>O agente Plataformista tem como objetivo mover a cunha para apoiar a coluna de perfuração e manipular a chave flutuante, visando conectar ou desconectar tubos. |

Tabela 6 – Modelo textual do agente Torrlista.

|  |
|--|
| <b>Agente: Torrlista.</b>  |
| <b>Tipo de Agente</b><br>Agente reativo baseado em modelo.   |
| <b>Objetivo</b><br>Realizar movimentação de segmentos.   |
| <b>Parâmetros de Entrada</b><br>Informações sobre o estado da plataforma (guincho, coluna, mesa rotativa, etc.).                           |
| <b>Parâmetros de Saída</b><br>Conduzir os tubos nas manobras de decida e retirada da coluna.   |
| <b>Condição de Ativação</b><br>Quando o agente percebe que está sendo adicionado um novo tubo na torre.                                    |
| <b>Condição de Finalização</b><br>Posicionamento correto do tubo na torre.   |
| <b>Informação Associada</b><br>Tem comportamento cíclico, percebendo e consultando o histórico do ambiente em intervalos preestabelecidos. |
| <b>Descrição:</b><br>O agente Torrlista tem como objetivo mover os tubos através da torre.   |

Depois da confecção dos modelos textuais, é realizada a escolha da arquitetura interna dos agentes, demonstrando os comportamentos que o agente irá realizar quando receber as informações perceptivas.

Na Figura 34 podemos observar detalhes da modelagem do AgenteSondador.

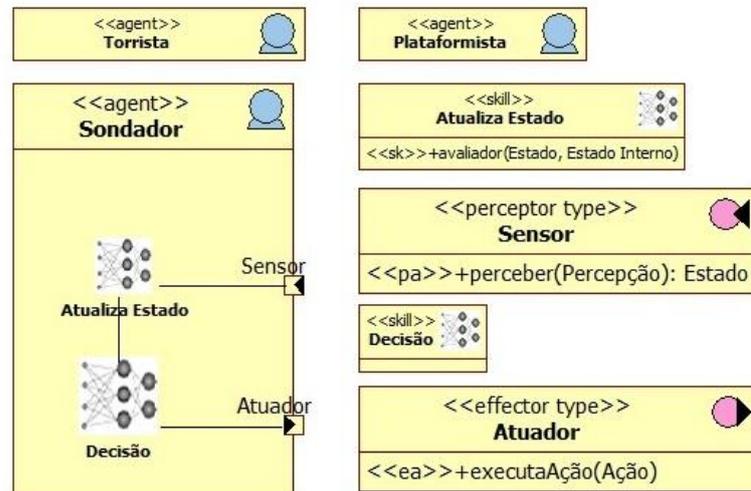


Figura 34 – Modelo do Agente Sondador.

### 6.6.2. Detalhes dos Agentes Utilitários

O grupo de agentes utilitários é formado pelo agente Coordenador e pelo agente Cliente. O primeiro está sempre ativo no *container* principal, enquanto que o segundo atua em um *container* remoto no navegador do usuário e só é ativado quando o jogo é iniciado. As tabelas 7 e 8 apresentam os modelos textuais dos agentes Coordenador e Cliente, respectivamente.

Tabela 7 – Modelo textual do agente Coordenador

| <b>Agente: Coordenador</b>   |  |
|------------------------------|--|
| <b>Tipo de Agente</b>        | Agente reativo.                            |
| <b>Objetivo</b>              | Iniciar agentes cooperativos e de suporte. |
| <b>Parâmetros de Entrada</b> |  |

|   |
|---|
| Código do Projeto e tipo de agente a ser iniciado.  |
| <b>Parâmetros de Saída</b><br>Mensagem com confirmação de inicialização do agente.            |
| <b>Informação Associada</b><br>Funcionamento cíclico sempre ativo esperando solicitações.     |
| <b>Descrição</b><br>Agente responsável por iniciar os agentes para participarem dos projetos. |

Tabela 8 – Modelo textual do agente Cliente.

|  |
|--|
| <b>Agente: Cliente</b>   |
| <b>Tipo de Agente</b><br>Agente reativo.   |
| <b>Objetivo</b><br>Modifica o estado da plataforma de perfuração.  |
| <b>Parâmetros de Entrada</b><br>Participantes da partida.  |
| <b>Parâmetros de Saída</b><br>Solicitação de ativação de agentes.  |
| <b>Condição de Ativação</b><br>Quando o agente percebe o ambiente e não encontra os agentes para participar da partida.  |
| <b>Condição de Finalização</b><br>Quando o agente recebe a resposta de ativação do agente.   |
| <b>Informação Associada</b><br>Este agente se ativa no início da seção para solicitar os agentes cooperativos e de suporte para participarem do projeto.   |
| <b>Descrição</b><br>Agente responsável por verificar o ambiente no início da seção e procurar pelos outros participantes, adicionando um agente cooperativo quando necessário. Os agentes de suporte são também adicionados. |

A Figura 35 apresenta os detalhes da modelagem do AgenteCliente.

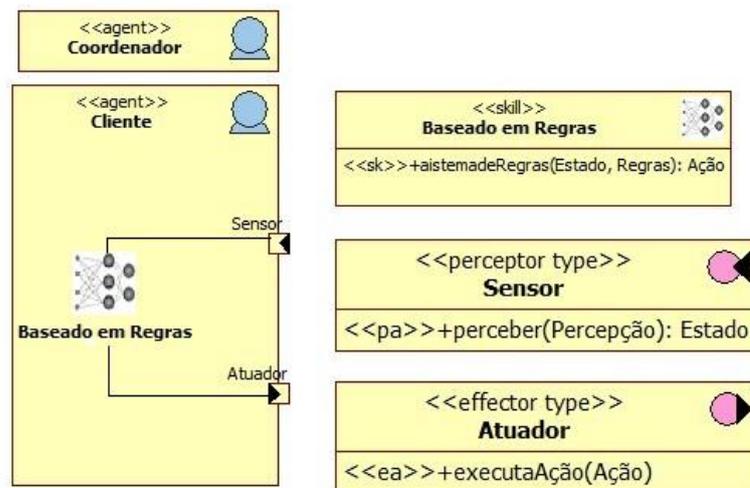


Figura 35 – Modelo do Agente Cliente.

Em um ambiente multiagente, a comunicação entre os agentes é um fator importante, pois os agentes precisam trocar informações para que sejam capazes de cooperar, colaborar e negociar. De uma forma geral, os agentes interagem através de algumas linguagens de comunicação. No ATEP, os agentes se comunicam de duas formas: (i) usando o serviço de páginas amarelas oferecido pelo Agente DF (*Directory Facilitator*); e (ii) usando o ambiente multijogador oferecido pelo H-N2N.

Para serem localizados, os agentes utilizam o serviço do Agente DF. Esse agente cadastra e procura por serviços oferecidos por outros agentes. No ATEP, todos os agentes precisam se identificar nesse serviço. O código mostrado na Figura 36 é aplicado para efetuar o registro de um agente no DF.

```

40
41 //Registra o serviço do tipo "agente-sondador" nas páginas amarelas
42 DFAgentDescription myDescriptionTemplate = new DFAgentDescription();
43 myDescriptionTemplate.setName(getAID());
44 ServiceDescription myServiceDescription = new ServiceDescription();
45 myServiceDescription.setType("agente-sondador");
46 myServiceDescription.setName("sondador-projeto");
47 myDescriptionTemplate.addServices(myServiceDescription);
48 try {
49     DFService.register(this, myDescriptionTemplate);
50 } catch (FIPAException fe) {
51 }
52

```

Figura 36 – Trecho de código do registro dos agentes.

Como pode ser visto no trecho de código apresentado na Figura 36, primeiramente é instanciado um objeto do tipo “*DFAgentDescription*<sup>8</sup>”, onde será preenchida a propriedade

<sup>8</sup> A classe *DFAgentDescription* pertence ao pacote “*jade.domain.FIPAAgentManagement*” do framework JADE.

*nome* com um identificador único criado pelo ambiente (*getAID()*). Em seguida, é criado um objeto do tipo “ServiceDescription”, onde é atribuído o tipo do agente e o nome. Finalizando, o agente é registrado no serviço do DF.

Quando há a necessidade de um agente se comunicar com outro agente, é necessário fazer uma consulta ao agente DF. A Figura 37 – Trecho de código fonte para consulta ao DF. Figura 37 apresenta o código do agente Cliente que, ao iniciar uma seção, verifica a existência de agentes para participar do projeto.

```

101
102    /// Verifica Agente Sondador
103    DFAgentDescription sondadorTemplate = new DFAgentDescription();
104    ServiceDescription sondadorSD = new ServiceDescription();
105    sondadorSD.setType("agente-sondador");
106    sondadorTemplate.addServices(sondadorSD);
107
108    try {
109        DFAgentDescription[] resultAgents = DFService.search(myAgent, sondadorTemplate);
110        //Se tiver algum serviço cadastrado no DF
111        if (resultAgents.length > 0) {
112
113            //Cria o array de serviços dos agentes retornado
114            agenteSondador = new AID[resultAgents.length];
115            for (int i = 0; i < resultAgents.length; i++) {
116                agenteSondador[i] = resultAgents[i].getName();
117            }
118
119        } else {
120
121            addBehaviour(new AtivaAgenteSondador());
122        }
123    } catch (FIPAException fe) {
124    }

```

Figura 37 – Trecho de código fonte para consulta ao DF.

Como pode ser visto no trecho de código apresentado na Figura 37, inicialmente é criado um *template* com a descrição do tipo de agente procurado (linhas 103-106). Em seguida, é realizada uma consulta ao serviço DF que retorna os agentes encontrados (linha 109). Finalmente, é verificado se a consulta retornou algum agente do tipo pesquisado e, caso não encontre nenhum agente, é adicionado um comportamento para solicitar ao agente Coordenador a ativação de um agente (linhas 114 – 122).

Para se conectar ao ambiente multijogador, o agente deve se registrar no H-N2N. A Figura 38 apresenta o código para essa operação.

```

52
53     //Porta do HN2N
54     porta = 6891;
55     //Endereço do HN2N
56     endereço = "192.168.2.108";
57     //Cliente do HN2N
58     userClient = new UserClient(endereço, porta);
59
60     cp = new ClientProcess();
61     //Aguarda a conexão
62     while (!userClient.isConnected());
63     try {
64         userClient.getCommunication().getConnectionOfControl().attach(cp);
65     } catch (Exception e) {
66     }
67

```

Figura 38 – Trecho de código do registro no H-N2N.

Como pode ser visto no trecho de código da Figura 38, os parâmetros utilizados para o registro do agente são o endereço e a porta do servidor. Em seguida, é instanciado um objeto do tipo *UserClient* do pacote H-N2N para, então, a conexão ser realizada.

### 6.6.1. Detalhes dos Agentes de Suporte

O grupo de agentes de suporte contempla o agente Plataforma e o agente Poço. Estes agentes permanecem ativados durante as seções do jogo. As tabelas 9 e 10 apresentam os modelos textuais dos agentes Plataforma e Poço, respectivamente.

Tabela 9 – Modelo textual do agente Plataforma.

| <b>Agente: Plataforma</b>    |   |
|------------------------------|---|
| <b>Tipo de Agente</b>        | Agente reativo baseado em modelo com lógica <i>fuzzy</i> .  |
| <b>Objetivo</b>              | Modifica o estado da plataforma de perfuração.  |
| <b>Parâmetros de Entrada</b> | Informações sobre o estado da plataforma (guincho, coluna, mesa rotativa, pontuação dos participantes, etc.). |
| <b>Parâmetros de Saída</b>   | Alterar estado das ferramentas.   |

|   |
|---|
| <b>Condição de Ativação</b>   |
| Quando o agente percebe o ambiente e o estado das ferramentas e objetos.  |
| <b>Condição de Finalização</b>  |
| Quando o agente modifica o estado de alguma ferramenta.   |
| <b>Informação Associada</b>   |
| Toma as decisões através de um sistema <i>fuzzy</i> .   |
| <b>Descrição</b>  |
| Agente responsável pelo estado da plataforma de perfuração (sonda). Ele representa todas as ferramentas, como cunha, chaves flutuantes, guincho, etc. Monitorando o ambiente, este agente pode simular uma quebra em uma das ferramentas, criando assim uma situação adversa. O agente Plataforma tem como objetivo alterar o estado dos componentes que compõem a mesma. |

Tabela 10 – Modelo textual do agente Poço.

|  |
|--|
| <b>Agente: Poço.</b>   |
| <b>Tipo de Agente</b>  |
| Agente reativo baseado em modelo com lógica <i>fuzzy</i> .   |
| <b>Objetivo</b>  |
| Monitorar e alterar parâmetros do poço.  |
| <b>Parâmetros de Entrada</b>   |
| Informações sobre o estado do poço e da plataforma (pressão do poço, profundidade, guincho, coluna, mesa rotativa, etc.).                                |
| <b>Parâmetros de Saída</b>   |
| Alteração da pressão do poço.  |
| <b>Condição de Ativação</b>  |
| Quando o agente percebe a perfuração do poço e altera a pressão do mesmo.  |
| <b>Condição de Finalização</b>   |
| Alteração da pressão do poço.  |
| <b>Informação Associada</b>  |
| Toma as decisões através de um sistema <i>fuzzy</i> .  |
| <b>Descrição</b>   |
| O agente Poço pode alterar a pressão do poço no decorrer da perfuração. Em caso de aumento repentino, gera uma situação adversa chamada de <i>kick</i> . |

A Figura 39 apresenta os detalhes da modelagem do AgentePlataforma.

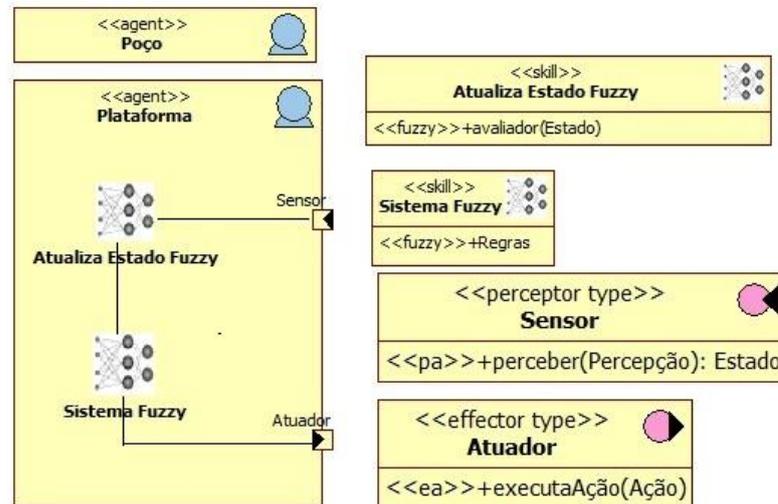


Figura 39 – Modelo do Agente Plataforma.

Os agentes de Suporte usam a Lógica *Fuzzy* na tomada de decisão. Para isso existem alguns pacotes de software disponíveis que facilitam o desenvolvimento de sistemas de inferência *fuzzy* (FIS, do inglês *Fuzzy Inference System*), como, por exemplo: o JFuzzyLogic (JFuzzyLogic, 2011), o SciLab (SciLab, 2011), o FFLL (FFLL, 2011) e o MatLab (MatLab, 2011). Dentre essas opções, verificou-se que: (i) o MatLab é um software proprietário e sem possibilidade de integrar ao ambiente; (ii) o FFLL não sofre atualização desde 2003; e (iii) o SciLab é um pacote muito extenso, cujas funcionalidades excedem as necessidades deste trabalho. O JFuzzyLogic se apresentou como melhor opção, pois é um software livre desenvolvido em Java compatível, assim, com a arquitetura baseada em Java deste trabalho.

A linguagem utilizada para implementar a Lógica *Fuzzy* em um sistema é a Linguagem de Controle *Fuzzy* (FCL), uma linguagem de domínio desenvolvida pela IEC (*International Electrotechnical Commission*) 61131-7 (COMMISSION, 2009). Este tipo de linguagem é utilizada para modelar o mecanismo de Lógica *Fuzzy*. Sua principal vantagem é a manutenção, ou seja, qualquer alteração que precise ser realizada na modelagem do controle pode ser efetivada com a alteração de um arquivo FCL. Um modelo de FCL com a representação de um *kick* é apresentado a seguir (o modelo teve partes ocultadas para melhoria da visualização):

---

**Modelo FCL**

---

```
// Definição do Bloco
FUNCTION_BLOCK poco
// Definição de variável de Entrada
VAR_INPUT
    pressao_coluna: REAL;
    profundidade : REAL;
    pressao_formacao: REAL;
END_VAR
// Definição de variável de Saída
VAR_OUTPUT
    broca : REAL;
    kick: REAL;
END_VAR
// Fuzzificação das variáveis de entrada para pressão
FUZZIFY pressao_formacao
    TERM baixa := (0, 1) (40, 0);
    TERM media := (10, 0) (40,1) (65,0);
    TERM alta := (55, 0) (70, 1) (85, 0);
    TERM muitoalta := (75, 0) (90, 1) ;
END_FUZZIFY
...
// Defuzzificação da variável Kick
DEFUZZIFY kick
    TERM fraco := (0,0) (5,1) (10,0);
    TERM medio := (10,0) (15,1) (20,0);
    TERM forte := (20,0) (25,1) (30,0);
    // Uso do método de desfuzzificação Centro de Gravidade
    METHOD : COG;
    // O valor padrão é 0 (se nenhuma regra ativar o defuzzificador)
    DEFAULT := 0;
END_DEFUZZIFY
...
```

```

//Bloco de regras
RULEBLOCK No1
  // Uso 'min' para 'and'
  AND : MIN;
  // Uso de método de ativação 'min'
  ACT : MIN;
  // Uso de método de ativação 'max'
  ACCU : MAX;

  RULE 1 : IF pressao_formacao IS baixa OR profundidade IS baixa
            THEN kick IS nulo;
  RULE 2 : IF pressao_formacao IS alta and pressao_coluna IS baixa THEN Kick IS
            medio;
  RULE 3 : IF profundidade IS grande AND pressao_formacao IS alta AND
            pressao_coluna IS baixa THEN Kick is forte;
  ...
END_RULEBLOCK
END_FUNCTION_BLOCK

```

---

O sistema *fuzzy* que controla o agente Poço é formado basicamente por três variáveis de entrada e duas de saída. As variáveis de entrada são "pressão da formação", "pressão da coluna" e "profundidade". As duas primeiras contêm quatro conjuntos de valores (baixa, media, alta, muito alta) e a terceira contém três conjuntos (pequena, média, grande). As variáveis de saída são "kick" e "desgaste da broca", contendo a primeira variável quatro conjuntos de valores (nulo, baixo, médio, alto) e a segunda os conjuntos (nenhum, baixo, médio, alto).

Foi utilizado um conjunto que variou de 10 a 15 regras para descrever o comportamento do agente Poço. Algumas das regras que foram usadas para gerar situações adversas são dadas a seguir:

- Se pressão da coluna é muito alta **E** pressão da formação é baixa **Então** kick é nulo;
- Se pressão da coluna é alta **E** pressão da formação é muito alta **Então** kick é médio;

- Se profundidade é pequena **E** pressão da formação é alta **Então** desgaste da broca é médio;
- Se pressão da coluna é baixa **E** pressão da formação é alta **Então** kick é forte;
- Se pressão da formação é alta **E** profundidade é grande **Então** desgaste da broca é alto.

### 6.6.2. Modelo de Interação dos Agentes

Nos modelos de interação entre os agentes apresentados nessa seção, é possível visualizar as interações que ocorrem entre os agentes e os recursos que são consultados por cada um dos tipos de agentes.

O modelo de interação da Figura 40 apresenta o comportamento dos agentes no momento em que o treinando inicia a seção do jogo. Nesse momento é criado o agente Cliente. O agente Cliente funciona em um ambiente remoto que roda no navegador Web do treinando.

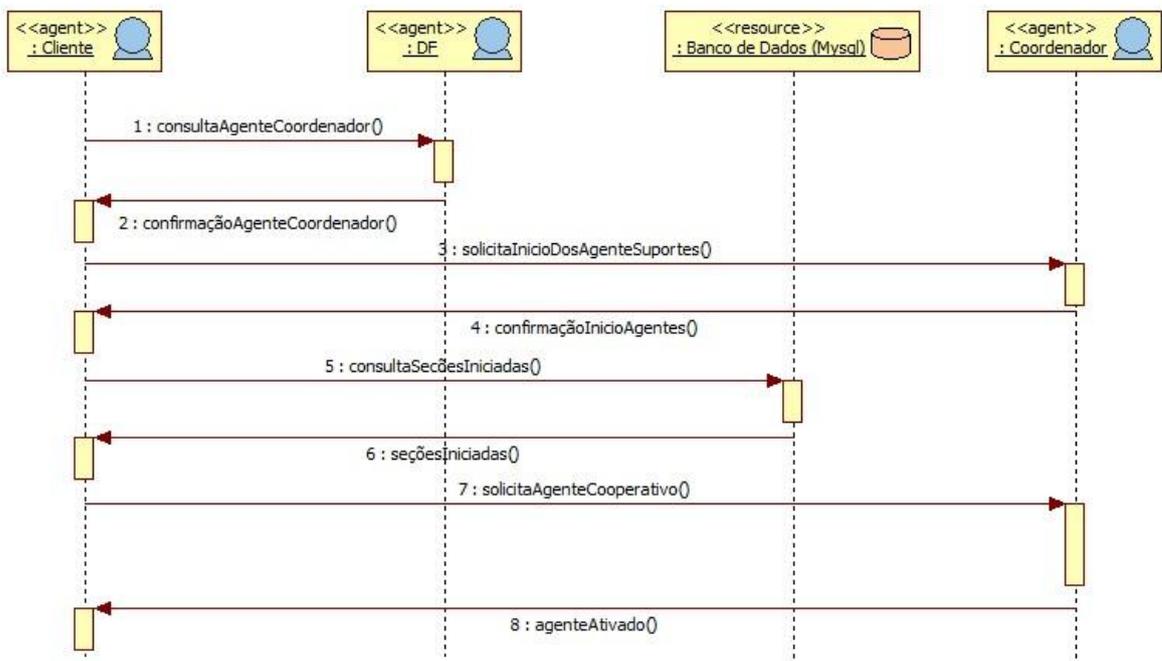


Figura 40 – Modelo de interação dos agentes utilitários.

Como pode ser visto no modelo de interação da Figura 40, ao iniciar o agente Cliente, as interações seguem o seguinte fluxo: (i) o agente Cliente consulta o serviço de páginas

amarelas oferecido pelo agente DF a procura do agente Coordenador; (ii) após receber a resposta de confirmação, o agente Cliente solicita ao agente Coordenador a inicialização dos agentes de suporte; (iii) após receber a confirmação, o agente consulta o banco de dados e recebe como retorno as seções iniciadas; (iv) concluindo, o agente Cliente solicita ao agente Coordenador a inicialização dos agentes cooperativos para completar a formação do jogo.

O modelo de interação da Figura 41 mostra o comportamento dos agentes logo após o início do agente cooperativo Sondador, desde o momento em que acessa a sala de jogo até quando efetua alguma ação no ambiente. É possível perceber que toda a interação entre jogadores ocorre através do framework H-N2N.

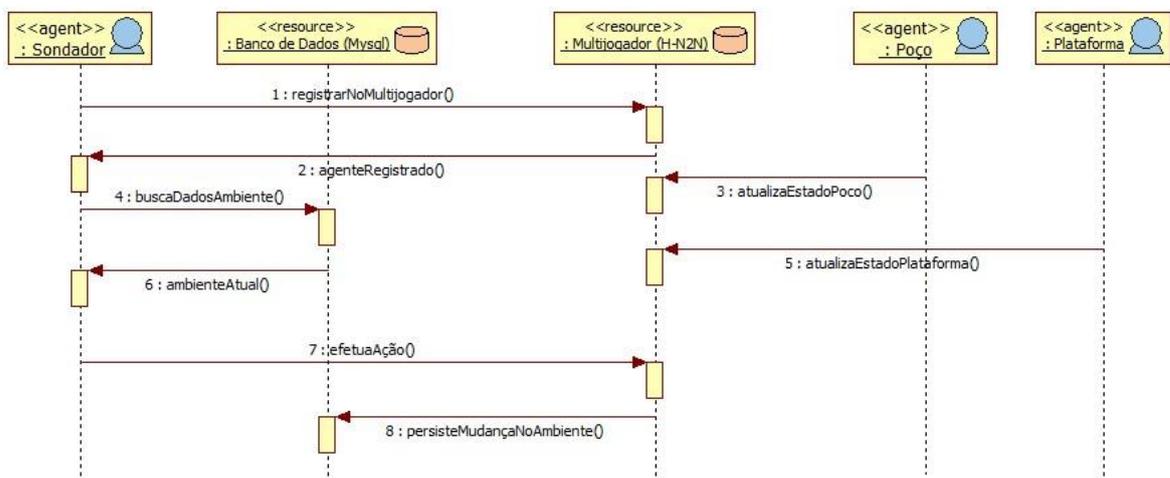


Figura 41 – Modelo de interação entre os agentes cooperativos e de suporte.

No modelo apresentado na Figura 41 é possível visualizar toda a interação que ocorre com o agente Sondador. Entretanto, esse fluxo é similar para todos os agentes que: (i) inicialmente se registram no ambiente multijogador (H-N2N); (ii) após isso recebem uma confirmação de registro e, a partir desse momento, passam a receber todas as mensagens do ambiente multijogador; (iii) após receber a confirmação do registro, os agentes passam a consultar o estado atual do ambiente persistido no banco de dados; (iv) em seguida, efetuam uma ação enviando uma mensagem para o ambiente multijogador; (v) finalmente, a ação é persistida no banco de dados e enviada para os outros participantes.

## 7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A EaD é uma modalidade de ensino e aprendizagem que tem crescido e apresentado bons resultados. Atualmente um dos grandes desafios é a evolução do suporte informatizado que possibilite novas facilidades e melhorias nesta modalidade de ensino. No contexto da indústria de petróleo, por ser uma atividade de alto risco, devido à exposição de fluidos com alta pressão e ferramentas pesadas, e necessitar sempre de mão de obra qualificada e atualizada, ferramentas de EaD convencionais cada vez mais devem ser usadas em conjunto com outras tecnologias. A incapacidade das ferramentas de EaD convencionais atenderem sozinhas aos requisitos de capacitação ocorre, principalmente, porque o treinamento necessário exige constantemente a verificação na prática dos conhecimentos adquiridos e a avaliação da capacidade de reação a situações de riscos não previstas.

Para tratar vários problemas típicos da EaD, como, por exemplo, altas taxas de abandono, os ambiente de aprendizagem podem fazer uso dos jogos educativos, que se tornaram importante área de pesquisa em EaD. Os jogos educativos eletrônicos se destacam como um meio motivador para o ensino e aprendizagem em ambientes de EaD.

Neste trabalho foi apresentado um ambiente de treinamento de equipes de perfuração (ATEP) que simula o processo de perfuração de poços de petróleo terrestres. Os principais elementos do jogo foram modelados em um cenário de plataforma de perfuração. Três perfis de trabalhadores e vários elementos de perfuração foram implementados neste jogo, que consiste em um ambiente 3D no qual o jogador pode navegar e interagir com os objetos, simulando tarefas reais.

O jogo persistente baseado em navegador para treinamento de operadores de sonda de perfuração de poços de petróleo, proposto neste trabalho, tem como principal objetivo testar os conhecimentos dos treinandos em situações adversas visando reduzir os riscos na atividade de perfuração de poços de petróleo. O jogo é apoiado pela teoria de aprendizagem PBL e procura abordar, além dos conceitos técnicos, os problemas relacionados ao trabalho em equipe.

A abordagem *fuzzy* para a modelagem dos agentes no jogo se mostrou versátil e flexível, aliada à facilidade de manipulação e à sua capacidade de adaptação, fazendo dela uma ferramenta de auxílio viável para o desenvolvimento de um sistema de IA bem elaborado e desafiador para o treinando. O objeto de estudo aqui apresentado pode ser facilmente

ampliado e adaptado com adições de novas regras e variáveis, as quais possibilitarão a mudança de comportamento dos agentes.

Diante do trabalho realizado, admite-se que há algumas melhorias que podem evoluir esse ambiente de aprendizagem. Como proposta de trabalhos futuros pode-se abordar:

- O aperfeiçoamento da simulação da perfuração no jogo, agregando novas funcionalidades e novas ferramentas à operação de perfuração de petróleo, que é uma tarefa complexa com muitas outras operações agregadas.
- O refinamento da modelagem 3D da sonda e seus objetos, bem como, a adição de novas ferramentas.
- A portabilidade do ambiente para outros dispositivos, como telefones e *tablets*. O jPCT possui um suporte para esse quesito que pode ser explorado em uma adaptação para plataforma Android<sup>9</sup>, fazendo com que o ATEP fique compatível com vários dispositivos móveis do mercado.
- A adição de novos cargos de trabalho em uma sonda(ex. encarregado de sonda, homem de área e outras) e implementação dos agentes para os novos cargos. Assim sendo, também deve-se evoluir o sistema de pontuação para esses novos cargos.
- Testar o ambiente desenvolvido em um contexto real, utilizando a Petrobras ou empresas de petróleo parceiras em treinamentos oficiais com instrutores. Isso permitirá uma melhor validação do ambiente em relação ao suporte à PBL.

Outras possibilidades de evolução da ferramenta estão no uso de controles de movimentos, como o Playstation Move<sup>10</sup> ou Kinect<sup>11</sup> e monitores 3D Estereoscópico<sup>12</sup>. Isso permitiria uma imersão e uma interação mais avançada com o ambiente, apresentando um ambiente mais envolvente e motivador para um jogo de treinamento.

---

<sup>9</sup> Sistema operacional para dispositivos moveis desenvolvido pela empresa Google (<http://www.android.com>).

<sup>10</sup> Controle sem fio baseado em bastão com botões e câmera desenvolvidos pela Sony (<http://www.sony.com>) para captação de movimentos.

<sup>11</sup> Controle avançado que capta movimentos apenas com o uso de uma câmera, desenvolvido pela Microsoft ([www.microsoft.com](http://www.microsoft.com)).

<sup>12</sup> A estereoscopia é a simulação de duas imagens da cena que são projetadas nos olhos em pontos de observação ligeiramente diferentes. O cérebro funde as duas imagens e, nesse processo, obtém informações quanto à profundidade, distância, posição e tamanho dos objetos, gerando uma sensação de visão 3D.

## REFERÊNCIAS

AGUIAR, H.; JUNIOR, O. **Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab**. São Paulo: Thmopson, 2007.

AHMAD, L. et al. Web-based e-learning in 3D large scale distributed interactive simulations using HLA/RTI. **Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on**, April 2008. 1 -4.

ALBERTO, A.; ALBERTO, C.; VEROTTI, C. **Fundamentos da Engenharia do Petróleo**. 2. ed. [S.l.]: "Editora Inteciência, 2004.

ARTERO, A. O. **Inteligência Artificial - Teórica e Prática**. 1. ed. [S.l.]: Livraria da Física, 2009.

BACKLUND, P.; ENGSTROM, H.; HAMMA, C. Sidh - a Game Based Firefighter Training Simulation. **Information Visualisation, International Conference on**, Los Alamitos, CA, USA, p. 899-907, 2007.

BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. Developing multi-agent systems with JADE, 2007.

BLENDER. Blender. **blender.org**, 2011. Disponível em: <<http://www.blender.org/>>. Acesso em: 30 Outubro 2011.

BOALER, J. **Project-based instruction: Creating excitement for learning**. Education Week Commentary, 29 (XVIII) March 31, pp 30 and 34. [S.l.]. 1999.

BRAGA, M. Realidade Virtual e Educação. **Revista de Biologia e Ciências da Terra**, v. 1, 2001.

BURLAMAQUI, A. et al. **H-n2n - uma soluçãoo escalável para ambientes virtuais colaborativos massivos**. XII Simpósio Brasileiro de Sistemas Multimídia e Web. [S.l.]: [s.n.]. 2006.

CASTRO, J.; GIORGIN, P.; KETHERS, S. Tropos: A Requirements-Driven Methodology for Agent-Oriented Software. In: HENDERSON-SELLERS; GIORGINI, P. **Agented-Oriented Methodologies**. [S.l.]: IDEA Group Publishing, 2005. p. 20-45.

CERVENKA, R.; TRENCANSKY, I. **AML, The Agent Modeling Language: A Comprehensive Approach to Modeling Multi-Agent Systems**. Basel, Suíça: Birkhauser Verlag, v. Whitestein Series in Software Agent Technologies and Automatic Computing, 2007.

CHELLA, A.; COSSENTINO, M.; VALERIA. Towards a Methodology for Designing Artificial Conscious Robotic Systems, 2009. Disponível em: <<http://www.aaai.org/ocs/index.php/FSS/FSS09/paper/view/882/1274>>.

- COMISSION, I. E. **Fuzzy control programming released**. IEC 61131-7. [S.l.]: [s.n.]. 2009.
- COSENTINO, M. From requirements to code with the PASSI methodology. **Agented-Oriented Methodologies**, 2005. 79--106.
- COSENTINO, M.; LOMBARDO, S.; SABATUCCI, L. M. Cossentino and S. Lombardo and L. Sabatucci. **PASSI Toolkit (PTK)**, 2011. Disponivel em: <<http://sourceforge.net/projects/ptk>>. Acesso em: 30 Outubro 2011.
- DE PADUA, W. **Multimídia: Conceitos e Aplicações**. 1. ed. [S.l.]: LTC, 2000. ISBN 8521612222.
- DEITEL, P.; DEITEL, H. **JAVA: COMO PROGRAMAR**. 8ª Edição. ed. [S.l.]: Pearson Education, 2010. ISBN 9788576055631.
- DELOACH, S. A. Analysis and Design using MaSE and agentTool, 2001.
- DELOACH, S.; KUMAR, M. Multiagent systems engineering: An overview and case study. In: HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-Oriented Methodologies**. Hershey, PA: IDEA Group Publishing, 2005. Cap. XI, p. 317-340.
- DRILLING, S. **Drill Systems Products**, 2011. Disponivel em: <<http://www.drillingsystems.com/products>>. Acesso em: 30 janeiro 2011.
- FFLL. **Free Fuzzy Logic Library**, 2011. Disponivel em: <<http://ffll.sourceforge.net/>>. Acesso em: 09 dez. 2011.
- FOUDATION, B. Blender Foudation, 2011. Disponivel em: <<http://www.blender.org/blenderorg/blender-foundation/about/>>. Acesso em: 30 out. 2011.
- FREITAS, S. D.; OLIVER, M. How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? **Comput. Educ.**, Oxford, UK, UK, April 2006.
- GARIJO, F. J.; GOMEZ-SANZ, J. J.; MASSONET, P. The MESSAGE Methodology for Agent-Oriented Analysis and Design. **Agented-Oriented Methodologies**, 2005. 203-234.
- GIORGINI, P. et al. Tropos: A requirements-driven methodology for agent-oriented software. In: HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-Oriented Methodologies**. Hershey, PA: IDEA Group Publishing, 2005. Cap. II, p. 20-45.
- GREENE, R.; ROGERS, G. Justifying core faculty assessment of students clinical performance using cognitive flexibility theory: A case example. **The Internet Journal of Allied Health Sciences and Practice**, n. 4, 2006.
- GUO, H.; TRTTEBERG, H.; WANG, A. I. TeMPS: A Conceptual Framework for Pervasive and Social Games. In: \_\_\_\_\_ **Digital Game and Intelligent Toy Enhanced Learning (DIGITEL)**, 2010 Third IEEE International Conference on. [S.l.]: [s.n.], 2010. p. 31-37.

HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-oriented Methodologies**. [S.l.]: Idea Group Inc (IGI), 2005. ISBN 1591405815.

HIBERNATE. hibernate. **hibernate**, 2012. Disponível em: <<http://www.hibernate.org/>>. Acesso em: 22 jan. 2012.

HMELO-SILVER, C. E. Problem-Based Learning: What and How Do Students Learn? **Educational Psychology Review**, 16, n. 3, 2004.

HMELO-SILVER, C. E.; BARROWS, H. S. Goals and strategies of a problem-based learning facilitator. **The Interdisciplinary Journal of Problem-based Learning**, 1, n. 1, 2006. 21-39.

HODGSON, P.; MAN, D.; LEUNG, J. Managing the Development of Digital Educational Games. **Proceedings of the 2010 Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning**, 2010. 191--193.

HSU, R. C.; LIU, W.-C. Project based learning as a pedagogical tool for embedded system education. **Information Technology: Research and Education, 2005. ITRE 2005. 3rd International Conference on**, p. 362 - 366, 2005.

IGLESIAS, C. A. **Definición de una Metodología para el Desarrollo de Sistemas Multiagente**. Departamento de Ingeniería de Sistemas Telemáticos, Universidade Politécnica de Madri. Madri. 1998.

IGLESIAS, C. A.; GARIJO, M. The Agent-Oriented Methodology MAS-CommonKADS. In: HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-Oriented Methodologies**. Hershey, PA: IDEA Group Publishing, 2005. Cap. III, p. 46-78.

IGLESIAS, G. M. The Agent Oriented Methodology MASCommonKADS. **Agented-Oriented Methodologies**, 2005. 48-78.

INTEL Educar. **Intel Educar - Aprendizagem Baseada em Projetos**, 2011. Disponível em: <<http://www.intel.com/education/la/pt/elementos/pba/content.htm>>. Acesso em: 16 dez. 2011.

JADE. Java Agent DEvelopment Framework. **Java**, 2011. Disponível em: <<http://jade.tilab.com/>>. Acesso em: 1 Novembro 2011.

JFUZZYLOGIC. **JFuzzyLogic**, 2011. Disponível em: <<http://jfuzzylogic.sourceforge.net/html/index.html>>. Acesso em: 10 dez. 2011.

JSF. Java Server Faces. **JSF**, 2011. Disponível em: <<http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>>. Acesso em: 09 dez. 2011.

JUNQI, W.; ZONGKAI, Y.; YUMEI, H. Z. Strategies for Designing Educational Computer Game, 3, 2009. 950--952.

KLIMMT, C.; SCHMID, H.; ORTHM, J. Exploring the Enjoyment of Playing Browser Games, 2009.

- LATTA, J. N.; OBERG, D. J. A Conceptual Virtual Reality Mode. **IEEE Comput. Graph. Appl**, Janeiro 1994. 23-29.
- LI, Z.; YUE, J.; JAUREGUI, D. A. G. A new virtual reality environment used for e-Learning, 2009. 445-449.
- LIU, R. C. H. A. W.-C. **Project based learning as a pedagogical tool for embedded system education**. [S.l.]: [s.n.], 2005. 362 - 366 p.
- MAES, P. Artificial life meets entertainment: lifelike autonomous agents. **Communications of the ACM**, New York, NY, USA, 38, 1995.
- MARKHAM, T.; LARMER, J.; RAVITZ, J. **Aprendizagem Baseada em Projetos - Guia para professores do ensino fundamental e médio**. 2. ed. Porto Alegre: Arned, 2008. ISBN 978-85-363-1108-1.
- MATLAB. **Mathworks, MatLab**, 2011. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Acesso em: 11 dez. 2011.
- MECENAS, I. **Java 6 Fundamentos, Swing, BlueJ e JDBC**. [S.l.]: Alta Books, 2008. ISBN 9788576082729.
- MORSI, R.; JACKSON, E. Playing and learning? Educational gaming for engineering education. **Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual**, October 2007. F2H--1--F2H--6.
- MYSQL. MySQL: The world's most popular open source database. **Site oficial do MySQL**, 2011. Disponível em: <<http://www.mysql.com/>>. Acesso em: 15 dez. 2011.
- NERBRATEN, O.; ROSTAD, L. ACMEgame: A Tool for Teaching Software Security. **"Availability, Reliability and Security, International Conference on**, 2009. 811-816.
- NETBEANS. Netbeans.org. **Netbeans**, 2012. Disponível em: <<http://www.netbeans.org/>>. Acesso em: 22 fev. 2012.
- NITTA, N.; TAKEMURA, Y.; IZURU, K. A practice of collaborative project-based learning for mutual edification between programming skill and artistic craftsmanship. **Proceedings of the 39th IEEE international conference on Frontiers in education conference**, San Antonio, Texas, USA, 2009. 1152--1156.
- ODELL, J.; PARUNAK, H. V. D.; BAUER, B. Extending UML for Agents. **Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligenc**, 2000. 3--17.
- OLIVEIRA, M. J. **MAS-CommonKADS+: Uma Extensão a Metodologia Mas-CommonKADS Para Suporte ao Projeto Detalhado de Sistemas Multiagentes Racionais**. Universidade Estadual do Ceara. [S.l.]. 2010.

OPENGL. OpenGL. **OpenGL - The Industry's Foundation for High Performance Graphics**, 2011. Disponível em: <[www.opengl.org](http://www.opengl.org)>. Acesso em: 12 dez. 2011.

OREILLY, T. What Is Web 2.0. **What Is Web 2.0**, 2005. Disponível em: <<http://oreilly.com/web2/archive/what-is-web-20.html>>. Acesso em: 30 Outubro 2011.

PADGHAM, L.; WINIKOFF, M. Prometheus: a methodology for developing intelligent agents. **Proceedings of the 3rd international conference on Agent-oriented software engineering III**, "Bologna, Italy, 2003. 174--185.

PADGHAM, L.; WINIKOFF, M. **Developing intelligent agent systems: a practical guide**. [S.l.]: John Wiley and Sons, 2004.

PADGHAM, L.; WINIKOFF, M. Prometheus: A practical agent-oriented methodology. In: HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-Oriented Methodologies**. Hershey, PA: IDEA Group Publishing, 2005. Cap. V, p. 107-135.

PANG, S.; CHEN, C.; YANG, Y. Classify players based on player relationship network in MMOG, 2009. 143-146.

PEREZ, Z. B.; MARIN, M.; PEREZ, E. **Developing a virtual environment for safety training**. CERMA 2007 - In Electronics, Robotics and Automotive Mechanics Conference. [S.l.]: [s.n.]. 2007. p. 545-550.

PETROBRAS. Produção de petróleo pela Petrobras superou dois milhões de barris por dia, 2011. Disponível em: <[www2.petrobras.com.br/ri/spic/bco/\\_arq/Release2milhõesPort.pdf](http://www2.petrobras.com.br/ri/spic/bco/_arq/Release2milhõesPort.pdf)>.

PIMENTEL, K.; TEIXEIRA, K. **Virtual reality - through the new looking glass**. New York: McGraw-Hill, 1995.

PONTES, A. A. **Uma Arquitetura de Agentes para Suporte a Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem**. Universidade Federal Rural do Semi-Árido - UFERSA. Mossoró, RN. 2010.

POTISARTRA, K.; KOTRAJARAS, V. An evenly matched opponent AI in Turn-based Strategy games, 2, 2010. 42-45.

PROJECT, P. Persistent Browser-Based Game - Defining a genre, 2011. Disponível em: <<http://www.pbbg.org>>. Acesso em: 29 Outubro 2011.

QUALHARINI, M. P. O. A. E. **Gestão de riscos na operação de plataformas de petróleo**. V Congresso Nacional de Excelência em Gestão. [S.l.]: [s.n.]. 2007.

RAILSBACK, J. Project-based instruction: Creating excitement for learning, 2002. Disponível em: <<http://educationnorthwest.org/webfmsend/460>>. San Jose, CA: Joint Venture, Silicon Valley Network.

REIS, F. et al. **Jogo educativo sobre a revolução da cabanagem fase pré-revolucionário.** em Anais do XXVIII Congresso da SBC - WIE - Workshop sobre Informática na Escola. [S.l.]: [s.n.]. 2008.

RIBAS, A. F. P.; MOURA, M. L. S. D. Abordagem sociocultural: algumas vertentes e autores. **Psicol. estud.**, 11, 2006.

RIBEIRO, L. R. C. Aprendizagem Baseada em Problemas (PBL) a Educação em Engenharia. **Revista de Ensino de Engenharia**, 27, n. 2, 2008. 23-32.

ROGERS, Y. Distributed cognition and communication. **The Encyclopedia of Language and Linguistics**, 2006. 181-202.

RONALD, N.; STERLING, L.; KIRLEY, M. Evaluating JACK Sim for Agent-Based Modelling of Pedestrians. **Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on**, December 2006. 81 -87.

RUSSELL, R.; NORVIG, P. **Artificial Intelligence, a Modern Approach 3th Edition.** [S.l.]: Prentice Hall, 2009.

SCHREIBER, G. **Knowledge engineering and management: the CommonKADS methodology.** [S.l.]: "MIT press, 2000.

SCILAB. **SciLab**, 2011. Disponível em: <<http://www.scilab.org/>>. Acesso em: 11 dez. 2011.

SOMMERVILLE, I. **Software Engineering.** 9ª Edição. ed. [S.l.]: Addison-Wesley, 2010. ISBN 978-0-13-703515-1.

SOUSA, R. C. S.; DA CUNHA, A. L. F.; MARTINS, R. Evaluating MaSE Methodology in the Requirements Identification. **Software Engineering Workshop (SEW), 2009 33rd Annual IEEE**, October 2009. 136 -143.

SPIRO, R. J.; COLLINS, B. P.; THOTA, J. J. Cognitive Flexibility Theory: Hypermedia for Complex Learning, Adaptive Knowledge Application, and Experience Acceleration. **Educational Technology**, 43, n. 5, 2003. 5-10.

STARUML. StarUML - The Open Source UML/MDA Platform. **Página Sourceforge do STARUML**, 2011. Disponível em: <<http://staruml.sourceforge.net/en/>>. Acesso em: 04 dez. 2011.

SUN, H.; THANGARAJAH, J.; PADGHAM, L. Hongyuan Sun and John Thangarajah and Lin Padgham. **Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems**, Richland, SC, 1, 2010. 1769--1770.

TEIXEIRA, F. G.; SILVA, R. P.; KOLTERMANN, T. L. **Software Engineering.** [S.l.]: Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, 2006. ISBN 85-7515-371-4.

THOMAS, J. W. **Project-based learning: Overview**. Buck Institute for Education. Novato, CA. 1998.

THOMAS, J. W. **A review of research on project-based learning**. Autodesk. San Rafael, CA. 2000.

TOMCAT. Apache Tomcat. **Apache Tomcat**, 2012. Disponível em: <<http://tomcat.apache.org/>>. Acesso em: 22 fev. 2012.

TRAN, Q. N.; LOW, G. M. Comparison of ten agent-oriented methodologies. **GIORGINI, B. H.S**, 2005.

VIGOTSKI, L. S. **A formação social da mente: o desenvolvimento dos processos psicológicos superiores**. 6. ed. São Paulo: Martins Fontes, 1998.

VOGEL-WALCUTT, J. J. et al. Cognitive load theory vs. constructivist approaches: which best leads to efficient, deep learning? **Journal of Computer Assisted Learning**, 2010.

WIBURG, K. **Thinking with computer - The Computing Teacher**. [S.l.]. 1994.

WINIKOFF, M.; PADGHAM, L. Prometheus: A Practical Agent-Oriented Methodology. **Agented-Oriented Methodologies**, 2005. 107--135.

WOOLDRIDGE, M. J. **An Introduction to Multiagent Systems**. New York, NY, USA: John Wiley & Sons, Inc., 2002.

YANG, F.; WU, W. "The Application of Virtual Reality in E-Learning, 2010. 5548-5551.

YESSAD, A.; LABAT, J.-M.; FRANÇOIS, K. SeGAE: A Serious Game Authoring Environment, 2010. 538-540.

YOON, J.-W.; JANG, S.-H.; CHO, S.-B. Enhanced user immersive experience with a virtual reality based FPS game interface, 2010. 69 -74.

YU, E. S. K. **Modelling strategic relationships for process reengineering**. Toronto, Ont., Canada, Canada: University of Toronto, 1995.

ZADEH, L. Fuzzy Sets. **Information and Control.**, p. 338-353, 1965.

ZAMBONELLI, F.; JENNINGS, N. R.; WOOLDRIDGE, M. Developing multiagent systems: The Gaia methodology. **ACM Transactions on software Engineering and Methodology**, 2003. 317--370.

ZAMBONELLI, F.; JENNINGS, N. R.; WOOLDRIDGE, M. Multi-Agent Systems as Computational Organizations: The Gaia Methodology. **Agented-Oriented Methodologies**, 2005. 136--171.

ZUO; XIAO-MING. An interactive multimedia web-based learning system for Pro/Engineer training, 2010. 152-154.