

Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Pós-Graduação em Engenharia Elétrica



**Ambientes distribuídos em Realidade
Virtual como suporte à Aprendizagem
Cooperativa para a Resolução de
Problemas**

Orientador: Alexandre Cardoso, Dr
Co-Orientadora: Elise B. Mendes, Dra
Orientando: Luciano Ferreira Silva

Agosto
2009

Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Pós-Graduação em Engenharia Elétrica

**Ambientes distribuídos em Realidade Virtual como
suporte à Aprendizagem Cooperativa para a
Resolução de Problemas**

*Tese apresentada por Luciano Ferreira Silva à
Universidade Federal de Uberlândia, como parte dos
requisitos para obtenção do título de Doutor em Ciências,
aprovada em 28/08/2009 pela Banca:*

Alexandre Cardoso, Dr (UFU) - Orientador
Elise B. Mendes, Dra (UFU) - Co-Orientadora
Edgard Lamounier Júnior, PhD (UFU)
Eduardo Kojy Takahashi, Dr (UFU)
Liliane dos Santos Machado, Dra (UFPB)
Lucia Maria Martins Giraffa, Dra (PUC-RS)

Dados Internacionais de Catalogação na Publicação (CIP)

- S586a Silva, Luciano Ferreira, 1982-
Ambientes distribuídos em realidade virtual como suporte à aprendizagem cooperativa para a resolução de problemas / Luciano Ferreira Silva. - 2009.
184 f. : il.
- Orientador: Alexandre Cardoso.
Co-orientadora: Elise B. Mendes.
Tese (doutorado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.
1. Realidade virtual - Teses. 2. Ensino gerenciado por computador - Teses. 3. Física - Estudo e ensino - Teses. I. Cardoso, Alexandre, 1964- II. Mendes, Elise Barbosa. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.3:007.52

**Ambientes distribuídos em Realidade Virtual como
suporte à Aprendizagem Cooperativa para a
Resolução de Problemas**

*Tese apresentada por Luciano Ferreira Silva à Universidade
Federal de Uberlândia como parte dos requisitos para obtenção
do título de Doutor em Ciências.*

Alexandre Cardoso, Dr
Orientador e Coordenador

À minha esposa por seu companheirismo, apoio e dedicação. À minha mãe Lazara Maria Ferreira e ao meu pai Luziano Alves da Silva, pelos exemplos de respeito, honestidade e simplicidade, pelo carinho, dedicação e pelos fundamentais apoios. E a todos familiares e amigos que contribuíram para o desenvolvimento deste trabalho.

Agradecimentos

Agradeço primeiramente a Deus, por iluminar os meus caminhos nas minhas dificuldades e vitórias, fazendo-me ser sempre perseverante na busca dos meus ideais. E em seguida, a todos que contribuíram para a realização deste trabalho e, pela participação mais direta, sou especialmente grato:

Ao meu orientador, Professor Alexandre Cardoso, por sua amizade, dedicação, paciência e pela importante, competente e segura orientação prestada em todos os momentos da realização deste trabalho.

À minha co-orientadora, Professora Elise B. Mendes, por sua amizade, dedicação, valiosas sugestões e importantíssima exigência quanto à qualidade deste trabalho.

Ao Professor Edgard Lamounier Júnior, pela amizade, dedicação, apoio e valiosas sugestões.

Aos Professores Eduardo Kojy Takahaschi e Silvia Martins, pelas excelentes sugestões, contribuições, apoio e dedicação oferecidos durante a execução desse trabalho multidisciplinar.

Aos meus amigos e parceiros de trabalho Ezequiel e a Mônica, pelo apoio, companheirismo, amizade, e valiosas sugestões para esta pesquisa.

Aos meus amigos do laboratório de Computação Gráfica, Marlene, Leonardo, Kenedy e Regislaine pela troca de conhecimento e experiências, e pela amizade.

Ao Victor Gargiulo por contribuir significativamente com suas sugestões para o desenvolvimento pedagógico deste trabalho.

Ao meu primo Fabio pela amizade, apoio e valiosas sugestões.

Aos professores da Universidade Federal de Roraima Roberto Câmara de Araújo pela valiosíssima contribuição para a avaliação deste projeto.

Ao professor Leandro Nelinho Ballico pelo apoio e valiosas sugestões ao término deste trabalho.

À minha esposa Queila por sua dedicação, carinho e apoio durante o desenvolvimento deste trabalho.

Aos meus pais e minha irmã Laís, por acreditarem em meu potencial incentivando, sempre, meus estudos.

Aos demais, mestres, amigos ou simples conhecidos, que foram, em níveis diferentes, fundamentais para a minha formação e que me prestaram auxílio em muitas ocasiões.

Resumo

SILVA, Luciano F. *Ambientes distribuídos em Realidade Virtual como suporte a Aprendizagem Cooperativa para a Resolução de Problemas*, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2008.

Palavras-Chave: Realidade Virtual, Resolução de Problemas, Aprendizagem Cooperativa.

Um grande número de pesquisadores acredita que a resolução de problemas é inerente a grande parte dos processos de ensino e aprendizagem. Considera-se ainda, que esta metodologia apresenta uma maior eficácia quando aplicada cooperativamente, ou seja, um aprendiz melhora o seu desempenho, em resolver problemas, quando realiza esta atividade em cooperação com outros aprendizes. No contexto do desenvolvimento de sistemas computacionais aplicados ao ensino, a Realidade Virtual (RV) vem se destacando como uma tecnologia que motiva o aprendiz, uma vez que ela permite a navegação, visualização, manipulação e interação com objetos virtuais de seus cenários de aprendizagem. Neste contexto, este trabalho apresenta uma proposta que viabiliza o uso da RV para oferecer suporte à resolução de problemas por meio da aprendizagem cooperativa. Para tal, serão considerados os fundamentos da aprendizagem por experimentação e a efetividade de teorias e princípios pedagógicos construtivistas. Como estudo de casos optou-se por desenvolver o projeto no âmbito da Física. O presente trabalho aborda e fundamenta a sua proposta, por meio da descrição do estado da arte da pesquisa nesta área, efetua uma análise e fornece especificações do projeto, apresenta os detalhes de desenvolvimento do sistema, discute as avaliações do mesmo e por fim, apresenta as conclusões e propostas de trabalhos futuros para este projeto.

Abstract

SILVA, Luciano F. *Distributed environment in Virtual Reality as support the Cooperative Learning for the Problems Solving*, Uberlândia, University of Electric Engineering - UFU, 2005.

Key-words: Virtual reality, Problems Solving, Cooperative Learning.

A great number of researchers believe that problem resolution is inherent from the great part of the teaching and learning processes. It is considered that this methodology presents a better effectiveness when applied cooperatively, in other words, an apprentice improves your acting in solving problems when accomplishes his activity in cooperation with other apprentices. In the context of computational systems applied to learning, Virtual Reality (VR) is emerging as a technology that motivates the apprentice, since it allows the navigation, visualization, manipulation and interaction with virtual objects of its learning environment. In this context, this work presents a proposal that allows the use of VR to offers support to the resolution of problems through the cooperative learning. For this, the learning foundations will be considered by experimentation and the effectiveness of theories and constructivities pedagogic principles. As case study it was chosen for developing the project in the Physics ambit. The present work approaches and bases its proposal through the state of the art description of the research in this area. It makes an analysis, supplies specifications of the project, introduces details of system developing, discusses that evaluation and finally, presents the conclusions and proposals future work for this project.

Lista de Publicações

A seguir são apresentadas as publicações no período de realização deste trabalho:

1. SILVA, Luciano Ferreira; CARDOSO, Alexandre; MENDES, Elise Barbosa; LAMOUNIER, Edgard; TAKAHASHI, Eduardo Kojy; MARTINS, Silvia; ZORZAL, Ezequiel Roberto; OLIVEIRA, Mônica Rocha. Realidade Virtual e Ferramentas Cognitivas Usadas como Auxílio para o Ensino de Física. In: RENOTE – Revista de Novas Tecnologias na Educação, 2008.
2. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; KIRNER, Cláudio; CARDOSO, Alexandre; LAMOUNIER, Edgard; OLIVEIRA, Mônica Rocha. Aplicação de Jogos Educacionais com Realidade Aumentada. In: RENOTE – Revista de Novas Tecnologias na Educação, 2008.
3. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; KIRNER, Cláudio; CARDOSO, Alexandre; LAMOUNIER, Edgard; OLIVEIRA, Mônica Rocha. Ambientes Educacionais Colaborativos com Realidade Aumentada. In: RENOTE – Revista de Novas Tecnologias na Educação, 2008.
4. SILVA, Luciano Ferreira; CARDOSO, Alexandre; MENDES, Elise Barbosa. Ambientes distribuídos de Realidade Virtual como suporte a Aprendizagem Colaborativa e a Resolução de Problemas. In: SYMPOSIUM ON VIRTUAL AUGMENTED AND REALITY, 2007, Petrópolis. Workshop of Thesis and Dissertations. Petrópolis: SBC, 2007.
5. SILVA, Luciano Ferreira; CARVALHO, Regislaine de; CARDOSO, Alexandre; LAMOUNIER, Edgard. DESENVOLVIMENTO DE UMA ARQUITETURA DE DISTRIBUIÇÃO PARA AMBIENTES VIRTUAIS. In: WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA, 2007, Itumbiara. WRVA. Itumbiara: Sociedade Brasileira de Computação, 2007.
6. SILVA, Luciano Ferreira; CARDOSO, Alexandre; MENDES, Elise B; TAKAHASHI, Eduardo Kojy; LAMOUNIER, Edgard; MARTINS, Silvia. Pedagogical design for E-learning Virtual Reality Software. In: 2007 INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND PATTERN RECOGNITION, 2007, Orlando, Florida, USA. AIPR2007. Orlando, Florida, USA: AIPR2007, 2007.
7. SILVA, Luciano Ferreira; FREITAS, Marlene Roque de; CARDOSO, Alexandre; LAMOUNIER, Edgard. Proposta de Integração de Jogos e Realidade Virtual no Ensino/ Aprendizagem de Matemática – Projeto LUDOS TOP. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 2007, Santos. ICECE, 2007.
8. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; CARDOSO, Alexandre; LAMOUNIER, Edgard. Realidade Virtual e o Algoritmo de Busca Tabu Aplicados ao Problema de Carregamento de Veículos. In: SYMPOSIUM ON VIRTUAL AUGMENTED AND REALITY, 2007, Petrópolis. SVR 2007. Petrópolis: SBC, 2007.

9. SILVA, Luciano Ferreira; CARDOSO, Alexandre; SOUZA, Everton S de; FREITAS, Marlene Roque de; LAMOUNIER, Edgard. Uso do Middleware CORBA na Distribuição de Ambientes Distribuídos para suportar Jogos Multi-Jogadores. Estudo de Caso “Projeto LUDOS TOP”. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 2007, Belém. SBRC 2007. Belém: SBC, 2007.
10. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; KIRNER, Claudio; CARDOSO, Alexandre; LAMOUNIER, Edgard; BONIFÁCIO, João Paulo. VISUALIZAÇÃO DE INFORMAÇÕES EM AMBIENTES DE REALIDADE VIRTUAL E AUMENTADA. In: WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA, 2007, Itumbiara. WRVA. Itumbiara: Sociedade Brasileira de Computação, 2007.
11. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; CARDOSO, Alexandre; KIRNER, Claudio; LAMOUNIER, Edgard; YAMANAKA, Keiji. A Visualização de Dados por meio de Realidade Virtual e o Algoritmo de Busca Tabu aplicados ao Problema de Carregamento de Veículos. In: III SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 2006, Curitiba. SBSI, 2006.
12. SILVA, Luciano Ferreira; CARDOSO, Alexandre; MENDES, Elise Barbosa; TAKAHASHI, Eduardo Kojy; MARTINS, Silvia. Associating non-immersive virtual reality and cognitive tools for Physics Teaching. In: 2006 ICEE CONFERENCE, 2006, Pittsburgh. ICEE, 2006.
13. SILVA, Luciano Ferreira; MENDES, Elise Barbosa; KERINSKA, Nikoleta. Desenvolvimento Multidisciplinar de Software Educacional por meio de Realidade Virtual. In: CONGRESSO NACIONAL DE AMBIENTES HIPERMÍDIA PARA APRENDIZAGEM, 2006, Florianópolis. CONAHPA, 2006.
14. SILVA, Luciano Ferreira; CHAGAS, Keila de Fátima; CARDOSO, Alexandre; LAMOUNIER, Edgard; YAMANAKA, Keiji. O Desenvolvimento de Ambientes de Realidade Aumentada Controlados por Comandos de Voz por Meio de Redes Neurais. In: III WORKSHOP DE REALIDADE AUMENTADA, 2006, Rio de Janeiro. WRA, 2006.
15. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; CARDOSO, Alexandre; KIRNER, Claudio; LAMOUNIER, Edgard; YAMANAKA, Keiji. O Uso da Realidade Virtual na Visualização de dados e Logística. In: II WORKSHOP DE APLICAÇÕES DE REALIDADE VIRTUAL, 2006, Recife. WARV, 2006.
16. SILVA, Luciano Ferreira; FREITAS, Marlene Roque de; LAMOUNIER, Edgard; CARDOSO, Alexandre. Proposta de integração de jogos e Realidade Virtual na educação Matemática - projeto Ludos Top. In: III SIMPÓSIO DE EDUCAÇÃO DO TRIÂNGULO MINEIRO, 2006, Ituiutaba. Ituiutaba: Instituto Superior de Educação de Ituiutaba - ISEDI, 2006.
17. SILVA, Luciano Ferreira; CHAGAS, Keila de Fátima; YAMANAKA, Keiji. Realidade Aumentada controlada por comando de voz, usando técnicas de redes neurais artificiais. In: 10ª REUNIÃO ANUAL DE CIÊNCIA, 2006, Uberlândia. Uberlândia: Unitri, 2006.

18. SILVA, Luciano Ferreira; CHAGAS, Keila de Fátima; CARDOSO, Alexandre; LAMOUNIER, Edgard; YAMANAKA, Keiji. Redes Neurais Artificiais Aplicadas no Desenvolvimento de Ambientes de Realidade Aumentada Controlados por Comandos de Voz. In: III SEMANA ACADEMICA UFU, 2006, Uberlândia. UFU, 2006.
19. SILVA, Luciano Ferreira; ZORZAL, Ezequiel Roberto; KIRNER, Claudio; CARDOSO, Alexandre; LAMOUNIER, Edgard. Visualização de Informação com Realidade Virtual e Aumentada. In: KIRNER, Cláudio; SISCOOTTO, Robson. (Org.). REALIDADE VIRTUAL E AUMENTADA: Conceitos, Projeto e Aplicações. São Paulo, 2007, v. 1, p. 256-275.

Sumário

1	INTRODUÇÃO	1
1.1	UMA PERSPECTIVA SOCIOECONÔMICA DA EDUCAÇÃO NO MUNDO.....	1
1.2	MOTIVAÇÃO	3
1.3	OBJETIVOS.....	5
1.4	CONTRIBUIÇÕES DA TESE.....	6
1.5	ESTRUTURA DA TESE	7
2	FUNDAMENTOS TEÓRICOS.....	9
2.1	INTRODUÇÃO	9
2.2	REALIDADE VIRTUAL E EDUCAÇÃO	9
2.2.1	<i>Realidade virtual e cognição.....</i>	<i>10</i>
2.2.2	<i>Objetos virtuais de aprendizagem.....</i>	<i>12</i>
2.2.3	<i>Laboratórios virtuais e Realidade Virtual</i>	<i>13</i>
2.3	APRENDIZAGEM COOPERATIVA	15
2.4	RESOLUÇÃO DE PROBLEMAS.....	18
2.4.1	<i>Modelos teóricos de resolução de problemas</i>	<i>19</i>
2.4.1.1	O modelo de Pozo e Polya.....	19
2.4.1.2	O modelo de Bransford e Stein	20
2.4.1.3	O modelo Backer e Mayer.....	21
2.4.1.4	O modelo de Kapa	22
2.5	DESENHO PEDAGÓGICO DO LVCF	23
2.5.1	<i>Modelo de resolução de problemas proposto para o LVCF.....</i>	<i>24</i>
2.5.1.1	Contato Inicial	25
2.5.1.2	Solução	25
2.5.1.3	Avaliação.....	27

2.6	CONSIDERAÇÕES FINAIS	28
3	TRABALHOS CORRELATOS	30
3.1	INTRODUÇÃO	30
3.2	ANDES	30
3.3	EJORNAL.....	31
3.4	SCIENCESPACE	33
3.5	WEBTOP.....	34
3.6	SUPPORTING DISTANCE LEARNERS FOR COLLABORATIVE PROBLEM SOLVING	35
3.7	THE EUROPEAN CHALLENGE	37
3.8	ARCA.....	38
3.9	DEVRL.....	40
3.10	NICE.....	41
3.11	CONSTRUCT3D.....	42
3.12	CONSIDERAÇÕES FINAIS	43
4	ANÁLISE E ESPECIFICAÇÃO DO SISTEMA.....	45
4.1	INTRODUÇÃO	45
4.2	ESPECIFICAÇÕES DO SISTEMA	45
4.2.1	<i>Requisitos do sistema</i>	<i>45</i>
4.2.2	<i>Diagrama de casos de uso do sistema.....</i>	<i>47</i>
4.2.3	<i>A arquitetura geral do sistema.....</i>	<i>48</i>
4.3	ESPECIFICAÇÕES DO LABORATÓRIO VIRTUAL DISTRIBUÍDO	52
4.3.1	<i>O sistema local do LVD.....</i>	<i>53</i>
4.3.1.1	<i>Diagrama de casos de uso do sistema local do LVD.....</i>	<i>53</i>
4.3.1.2	<i>Diagrama de atividade do sistema local do LVD</i>	<i>54</i>
4.3.1.3	<i>Diagrama da arquitetura do sistema local do LVD</i>	<i>55</i>
4.3.2	<i>O sistema de distribuição do LVD.....</i>	<i>56</i>

4.3.2.1	Diagramas de atividade do sistema de distribuição do LVD.....	59
4.3.2.2	Diagrama da arquitetura de distribuição do LVD	62
4.4	CONSIDERAÇÕES FINAIS	63
5	DESENVOLVIMENTO DO PROTÓTIPO.....	64
5.1	INTRODUÇÃO	64
5.2	TECNOLOGIAS UTILIZADAS.....	64
5.2.1	<i>VRML e X3D</i>	64
5.2.2	<i>JavaScript</i>	67
5.2.3	<i>Ferramentas Java de distribuição</i>	68
5.3	IMPLEMENTAÇÃO DO SISTEMA LOCAL DO LVD	72
5.3.1	<i>Desenvolvimento dos objetos virtuais de aprendizagem</i>	73
5.3.1.1	Balança de precisão	77
5.3.1.2	Becker com líquido.....	78
5.3.1.3	Bico de Bunsen.....	82
5.3.1.4	Calculadora científica.....	84
5.3.1.5	Calorímetro.....	85
5.3.1.6	Cronômetros	86
5.3.1.7	Máquina de líquidos	87
5.3.1.8	Mola.....	88
5.3.1.9	Pesos	89
5.3.1.10	Pistão didático	90
5.3.1.11	Termômetro	91
5.3.1.12	Trena digital.....	92
5.3.1.13	Trilho de ar	93
5.3.2	<i>Centro de atualização</i>	95
5.3.3	<i>Interface do LVD</i>	96

5.4	IMPLEMENTAÇÃO DO SISTEMA DE DISTRIBUIÇÃO DO LVD	99
5.4.1	<i>Cliente e Servidor Java TCP</i>	99
5.4.2	<i>Transmissor</i>	104
5.5	CONSIDERAÇÕES FINAIS	113
6	AVALIAÇÃO E VALIDAÇÃO DO SISTEMA	114
6.1	INTRODUÇÃO	114
6.2	METODOLOGIA DE AVALIAÇÃO	114
6.3	A CRIAÇÃO DOS GRUPOS DE TESTE NA AVALIAÇÃO DO LVCF	117
6.4	EXEMPLIFICAÇÃO DA ANÁLISE COMPARATIVA EM UM PROBLEMA	123
6.5	RESULTADOS POR NÍVEIS DE PROBLEMAS E GERAL DA AVALIAÇÃO DO LVCF	131
6.6	CONSIDERAÇÕES FINAIS	140
7	CONCLUSÕES E TRABALHOS FUTUROS.....	141
7.1	INTRODUÇÃO	141
7.2	CONCLUSÕES	141
7.3	TRABALHOS FUTUROS	145
	REFERÊNCIAS BIBLIOGRÁFICAS	147
	ANEXO 1.....	154
	ANEXO 2.....	155
	ANEXO 3.....	156
	ANEXO 4.....	159

Lista de Figuras

Figura 1.1: Área de pesquisa e contribuições do LVCF.....	6
Figura 1.2: Contribuições para as áreas multidisciplinares do trabalho.....	7
Figura 2.1: Modelo mental, Interação e Representação.	12
Figura 2.2: Cronômetro virtual.....	13
Figura 2.3. A localização do LVCF perante as três áreas do conhecimento.	24
Figura 2.4: Esquema ilustrativo para o modelo de resolução de problemas do LVCF.	29
Figura 3.1: Interface do Andes.	31
Figura 3.2: Interface do eJornal.....	32
Figura 3.3: Interface Cockpit.....	33
Figura 3.4: Ambientes gráficos do ScienceSpace.	33
Figura 3.5: Interação no ScienceSpace.....	34
Figura 3.6: Uma experiência do <i>software</i> WebTop.....	35
Figura 3.7: Interface do Supporting Distance Learners For Collaborative Problem Solving. .	37
Figura 3.8: Interface do ambiente colaborativo RV.	39
Figura 3.9: Ambiente de experimentação do projeto ARCA.	39
Figura 3.10: Interfaces do DEVRL.	40
Figura 3.11: Crianças interagindo com o NICE.	41
Figura 3.12: O uso do Construct3D.....	42
Figura 4.1: Diagrama de casos de uso do sistema.	48
Figura 4.2: Arquitetura do sistema.	49
Figura 4.3: Interface da plataforma de ensino a distância Moodle.....	51
Figura 4.4: Arquitetura do sistema com a plataforma Moodle.....	52
Figura 4.5: Diagrama de caso de uso do sistema local do LVD.....	54
Figura 4.6: Diagrama de atividade do sistema local do LVD.	55

Figura 4.7: Arquitetura do sistema local do LVD.	56
Figura 4.8: Modelo centralizado de sistema de RV multi-usuários.	57
Figura 4.9: Modelo distribuído de sistema de RV multi-usuários.....	58
Figura 4.10: Modelo distribuído replicado.	58
Figura 4.11: Modelo distribuído particionado.....	59
Figura 4.12: Diagrama de atividade do sistema de distribuição do LVD - enviar dados.....	60
Figura 4.13: Diagrama de atividade do sistema de distribuição do LVD - receber dados.	61
Figura 4.14: Arquitetura de distribuição do LVD.	62
Figura 5.1: Comparação entre <i>Sockets</i> e RMI.	71
Figura 5.2: Esquema de implementação dos objetos virtuais de aprendizagem.	73
Figura 5.3: Diagrama de classes dos objetos.....	74
Figura 5.4. Diagramas da classe <i>Bico_Bunsen</i> e <i>Peso_Trilho</i>	75
Figura 5.5: Modelo de interação dos objetos virtuais.....	77
Figura 5.6: Catálogo de ilustrações da balança de precisão.	78
Figura 5.7: Catálogo de ilustrações do becker com líquido.	79
Figura 5.8: Mistura de coloração de líquidos.	80
Figura 5.9: Círculo cromático.....	81
Figura 5.10: Código do algoritmo de mistura de coloração.	82
Figura 5.11: Catálogo de ilustrações do bico de Bunsen.....	83
Figura 5.12: Aumento da passagem de gás no bico de Bunsen.....	83
Figura 5.13: Aumento da passagem de ar no bico de Bunsen.....	84
Figura 5.14: Mensagens informativas do bico de Bunsen.....	84
Figura 5.15: Catálogo de ilustrações da calculadora virtual.....	85
Figura 5.16: Catálogo de ilustrações do calorímetro virtual.	86
Figura 5.17: Catálogo de ilustrações do cronômetro.....	86

Figura 5.18: Catálogo de ilustrações do cronômetro digital controlado por sensores fotoelétricos.	87
Figura 5.19: Catálogo de ilustrações da máquina de líquidos virtual.....	87
Figura 5.20: Catálogo de ilustrações da mola virtual.	88
Figura 5.21: Código do algoritmo de deformação da mola virtual.	89
Figura 5.22: Conjunto de pesos virtuais.	90
Figura 5.23: Catálogo de ilustrações do pistão virtual.	91
Figura 5.24: Adição de pesos no pistão virtual.	91
Figura 5.25: Catálogo de ilustrações do termômetro virtual.	92
Figura 5.26: Catálogo de ilustrações da trena digital virtual.....	92
Figura 5.27: Catálogo de ilustrações do trilho de ar virtual.	94
Figura 5.28: Componentes do trilho de ar virtual.....	95
Figura 5.29: Centro de atualização.	96
Figura 5.30: Interface do LVD.	97
Figura 5.31: Câmeras do LVD.	98
Figura 5.32: <i>Sockets</i> cliente e Servidor.	100
Figura 5.33: Interface de estabelecimento e encerramento de conexão do servidor.	100
Figura 5.34: Trecho de código da classe <i>Server</i>	101
Figura 5.35: Aplicação do LVD chamando a classe <i>Server</i>	102
Figura 5.36: Interface de estabelecimento e encerramento de conexão do cliente.....	102
Figura 5.37: Trecho de código da classe <i>Client</i>	103
Figura 5.38: Algoritmo para enviar dados do Transmissor.	106
Figura 5.39: Algoritmo para receber dados do Transmissor.	108
Figura 5.40: Ambientes LVD conectados.	110
Figura 5.41: O uso de câmeras em LVDs conectados.....	110
Figura 5.42: Ambientes virtuais do LVD conectados remotamente.	111

Figura 5.43: Tela dos computadores conectados remotamente apresentando a conexão do LVD.....	112
Figura 6.1: Notas da amostra λ	117
Figura 6.2: Histograma da amostra λ	118
Figura 6.3: Curva de densidade de probabilidade λ	119
Figura 6.4: Notas dos grupos α , β e γ	120
Figura 6.5: Histogramas e curvas de densidade dos grupos α , β e γ	121
Figura 6.6: Sobreposição das curvas de densidade de probabilidade de α , β , γ e λ	122
Figura 6.7: Ilustração do problema exemplo.....	124
Figura 6.8: Ilustração da experimentação de um problema.....	125
Figura 6.9: Gráfico sobre o desempenho dos alunos em montar os experimentos.....	125
Figura 6.10: Gráfico da etapa de identificação do problema referente ao problema exemplo.	126
Figura 6.11: Gráfico da etapa de representação mental do problema referente ao problema exemplo.	127
Figura 6.12: Gráfico da etapa de geração de hipóteses referente ao problema exemplo.....	127
Figura 6.13: Gráfico da etapa de busca e exploração de estratégias do problema exemplo. .	128
Figura 6.14:Gráfico da etapa de projeto e solução referente ao problema exemplo.....	128
Figura 6.15: Gráfico da etapa de previsão de resultados referente ao problema exemplo.....	129
Figura 6.16: Gráfico da etapa de aplicação referente ao problema exemplo.....	129
Figura 6.17: Gráfico da etapa de gerenciamento da aplicação referente ao problema exemplo.	130
Figura 6.18: Gráfico da etapa de crítica da solução referente ao problema exemplo.....	130
Figura 6.19: Gráfico da etapa de entrega da solução referente ao problema exemplo.....	131
Figura 6.20: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 1.	133
Figura 6.21: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 2.	135

Figura 6.22: Ilustração do problema do nível 3.....	136
Figura 6.23: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 3.	137
Figura 6.24: Gráfico referente ao desempenho geral dos alunos na resolução de problemas.	139

Lista de Abreviaturas

ANSI – American National Standards Institute

API – Application Programming Interface

ARCA – Ambiente de Realidade Virtual Cooperativo de Aprendizagem

AVC – Ambiente Virtual Cooperativo

CAD – Computer-Aided Design

CALVIN – Collaborative Architectural Layout Via Immersive Navigation

CEBE – Center for Education in the Built Environment

CEDUC – Centro de Educação

CORBA – Common Object Request Broker Architecture

DEVRL – Distributed Extensible Virtual Reality Laboratory

EAI – External Authoring Interface

EFA – Objetivos de Educação para Todos

EPFL – École Polytechnique Fédérale de Lausanne

GUI – Graphics User Interface

IAO – Information Awareness Office

ISD – Instructional System Design

JRMP – Java Remote Method Protocol

LDB – Lei de Diretrizes e Bases da Educação Nacional

LVCF – Laboratório Virtual Cooperativo de Física

LVD – Laboratório Virtual Distribuído

Moodle – Modular Object-Oriented Dynamic Learning Environment

NICE – Narrative Immersive Constructionist/Collaborative Environments

OECD – Organisation for Economic Co-operation and Development

OMG – Object Management Group

PISA – Programme for International Student Assessment

RMI – Remote Method Invocation

RA – Realidade Aumentada

RV – Realidade Virtual

SAI – Scene Access Interface

SCORM – Sharable Content Object Reference Model

SDLFCPS – Supporting Distance Learners For Collaborative Problem Solving

TCP – Transport Control Protocol

TEC – The European Challenge

UDP – User Datagram Protocol

UFRR – Universidade Federal de Roraima

UNED – Universidad Nacional de Educación a Distancia

UPC – Universitat Politècnica de Catalunya

VRML – Virtual Reality Modeling Language

XML – Extensible Markup Language

Capítulo 1

Introdução

1.1 Uma perspectiva socioeconômica da Educação no mundo

Em 2007, a UNESCO, Organização das Nações Unidas para Educação, a Ciência e a Cultura, ponderou um breve preâmbulo sobre o panorama e as perspectivas do sistema de ensino no mundo. De acordo com estes estudos a situação da educação mundial é preocupante. Atualmente, são necessários de 15 a 35 milhões de professores para atingir os Objetivos de Educação para Todos (EFA) até 2015. O movimento de educação para todos é um compromisso mundial para prover uma educação básica de qualidade a todas as crianças e a todos os jovens e adultos. Este compromisso assumido no Fórum Mundial de Educação em Dakar (Senegal), em 2000, visa reduzir pela metade o número de analfabetos até 2015 [Unesco, 2001].

No Hemisfério Norte, assim como no Sul, a falta de professores está alcançando níveis alarmantes. Não somente em termos quantitativos, visto que a necessidade de professores está em torno de 30 milhões, mas também em relação à qualificação desses. No esforço de cortar despesas públicas, alguns países estão solicitando professores-voluntários que são mal treinados e igualmente mal pagos. Além disso, em muitos países, os salários dos professores são muito baixos para atraírem profissionais, sobretudo, nas áreas rurais. Na África, por exemplo, de acordo com o Relatório de Monitoramento Global de Educação para Todos de 2005, o salário de professores em 2000 foi menor do que em 1970. Em algumas escolas, o índice de matrícula tem crescido mais rapidamente que o número de professores, de modo que a proporção estudante/professor chega a ser de 60 (sessenta) alunos para um professor [Unesco, 2007].

No Brasil, apesar da quantidade de licenciados estar aumentando, o número de ingressos nos cursos de graduação que oferecem licenciatura, mais que dobrou, passando de 166 mil em 1991 para 362 mil em 2002. Estima-se que a carência de professores ainda perpetuará por algum tempo. Este problema deverá ser maior para as séries finais do ensino

fundamental e para o ensino médio, principalmente, nas áreas de Física e Química [Inep, 2004].

Em termos de qualificação, a formação de professores melhorou em todos os níveis de ensino e houve redução de professores leigos na última década. Contudo, apenas 57% dos docentes que atuavam na pré-escola, ensino fundamental e ensino médio possuíam formação em nível superior, que estavam totalmente de acordo com critérios exigidos pela Lei de Diretrizes e Bases da Educação Nacional – LDB. Por exemplo, no ensino médio, das 469 mil funções docentes, 89% têm a graduação completa e 79% deles a licenciatura. Neste nível de ensino, no entanto, quase 11% dos docentes ainda possuem somente o ensino médio completo, o que indica que ainda existe uma forte necessidade de maiores investimentos em formação de professores [Inep, 2004].

Outro fator agravante é a qualidade do processo de ensino e aprendizagem no Brasil devido às más condições de trabalho oferecidas ao professor. Nas escolas públicas brasileiras, 45% dos professores atuam em escolas sem biblioteca. A falta de laboratório de Ciências, para aulas práticas, configura-se no pior indicador em relação a infra-estrutura. No País, 80% dos docentes trabalham em escolas que não contam com esse suporte pedagógico experimental. Nas Regiões Norte e Nordeste, esta situação atinge 94% dos profissionais [Inep, 2004].

Diante destes dados, o governo federal pretende diminuir uma parte destes problemas investindo na informática como ferramenta de ensino. Por exemplo, em 2007 foi prevista a liberação de R\$ 128 milhões para investimentos na parte de infra-estrutura do Programa Nacional de Informática na Educação (Proinfo) - responsável pela informatização das escolas desde 1997, na aquisição de computadores e outros equipamentos. Outros R\$ 75 milhões serão gastos com a criação de conteúdos digitais. Até 2010, o governo pretende instalar laboratórios de informática em todas as 130 mil instituições de ensino público do País [Terra, 2007].

Em 2003, o programa PISA – *Programme for International Student Assessment*, desenvolvido pela OECD – *Organisation for Economic Co-operation and Development*, divulgou outro dado estatístico preocupante. De acordo com o PISA, dentre os 40 países que fazem parte do programa, o Brasil encontra-se em antepenúltimo lugar no nível de proficiência em resolução de problemas, estando melhor apenas que a Indonésia e a Tunísia.

A Finlândia se destacou em primeiro lugar, seguida respectivamente pela Coréia e Hong Kong [Oecd, 2004].

Com base nestes dados estatísticos e outros fatores científicos é fundamentada a motivação deste trabalho, a qual é descrita na próxima seção.

1.2 Motivação

Nos últimos anos, ampliaram-se as pesquisas que têm como intuito promover a aprendizagem cooperativa e colaborativa em ambientes computacionais [Behrens, 2002], bem como o desenvolvimento de competências cognitivas, tais como a resolução de problemas [Aaas, 2001]. Inúmeros autores consideram que essas novas pesquisas devem contrapor-se aos estudos que utilizam as novas tecnologias apenas para transmitir e tornar disponível as informações aos estudantes. Dessa forma, as novas propostas de ensino buscam proporcionar cenários de aprendizagem que estimulem o desenvolvimento da autonomia intelectual e isto é possível por meio de processos de aprendizagem que promovem o raciocínio e a resolução de problemas de forma coletiva [Barros, 1994]. Algumas estratégias de ensino associadas à mídia moderna podem oferecer suporte a essa aprendizagem, uma vez que transformam as formas de transmissão e assimilação do conhecimento, influenciando a maneira de conceber e realizar a formação de professores e alunos [Silveira e Arruda, 1998]. Dentre elas, a Realidade Virtual (RV) apresenta-se como uma tecnologia que oferece prósperas contribuições ao desenvolvimento de aplicações voltadas para o ensino [Burdea e Coiffet, 2003], [Sherman e Craig, 2003], [Vince, 2004]. De acordo com Barros e Kelner (2003) as razões para utilizar RV no ensino são diversas, destacando-se, por exemplo, a maior motivação do usuário [Barros e Kelner, 2003]; pois RV é uma forma com a qual as pessoas visualizam, manipulam e interagem com computadores e dados extremamente complexos [Kirner, 2004]. Além disso, a RV possibilita a criação de ambientes virtuais interativos e intuitivos, proporcionando a manipulação direta de seus objetos que respondem às ações do usuário, estimulando sua participação ativa.

Os avanços das pesquisas em RV e a oferta de produtos nessa área, envolvendo *hardware*, *software* e dispositivos não convencionais, têm viabilizado a existência de diversas plataformas para suas aplicações. Essas plataformas variam desde sistemas baseados em microcomputadores, passando por estações de trabalho e máquinas paralelas, até sistemas distribuídos; podendo ser manipuladas de forma individual ou coletiva. O desenvolvimento de sistemas de RV multi-usuários em ambientes distribuídos vem crescendo e estes apresentam-

se como uma tecnologia de elevado potencial de aplicação para diversas áreas do conhecimento.

Ao encerrar as considerações atuais sobre a relevância da resolução de problema e da aprendizagem cooperativa, e após ressaltar as contribuições da tecnologia de RV para o processo de ensino e aprendizagem, verifica-se que a motivação desta pesquisa relaciona-se ainda com os seguintes fatores:

i. Científicos:

- os processos criativos de aprendizagem iniciam-se por meio de problemas [Bono, 1973];
- o desenvolvimento do pensamento formal e complexo requer a comunicação e cooperação entre vários aprendizes [Jones e Jacobs, 2000];
- a RV contribui com a aprendizagem por descoberta e a identificação das variáveis de um problema, visto que é possível a experimentação virtual por meio da imersão, manipulação e observação dos resultados das suas ações, de acordo com situações reais [Bullinger, Müller-Spahn e Rößler, 1996];
- sistema distribuídos de RV possibilitam que vários usuários interajam em um mesmo ambiente virtual, o que pode viabilizar a cooperação entre os aprendizes;

ii. Socioeconômicos:

- os estudantes apresentam, em média, um baixo índice em proficiência em resolução de problemas [Oecd, 2004];
- a carência de laboratórios para o ensino de ciências da natureza e professores qualificados para estimular a aprendizagem por experimentação [Inep, 2004];
- o investimento do governo na informática para a Educação, sendo destinada uma parte da renda para a construção de laboratórios de computação nas escolas;
- o processo de inclusão digital;

Por fim, neste contexto pode-se observar que a resolução de problemas é uma importante estratégia de ensino, que está envolvida em uma grande parte das situações de aprendizagem. Percebe-se, também, que o sucesso desta metodologia está profundamente

relacionado com a sua ligação a modelos pedagógicos que contemplam a aprendizagem cooperativa [Barros, 1994]. Verifica-se ainda, que a RV oferece vantagens significativas para a sua aplicação em processos de ensino e aprendizagem, e, além disso, ela pode oferecer suporte a aprendizagem cooperativa por meio de ambientes virtuais distribuídos.

1.3 Objetivos

Considerando os fatores e dados supracitados, que norteiam as motivações desta pesquisa, o objetivo deste trabalho é propor e validar estratégias computacionais e metodologias pedagógicas associadas ao desenvolvimento de ambientes de RV, em sistemas distribuídos, para suporte à resolução de problemas por meio da aprendizagem cooperativa.

Para tanto, deve-se considerar as teorias e os princípios pedagógicos construtivistas e os fundamentos de experimentação por meio de laboratórios virtuais. Como estudo de caso, optou-se por desenvolver o projeto no âmbito da Física. Para atingir o objetivo proposto são traçadas as seguintes metas:

- pesquisar sobre a associação de teorias da aprendizagem cooperativa e resolução de problemas, e inquirir sobre as possibilidades e vantagens de aplicar RV na Educação;
- analisar ferramentas computacionais de ensino que abordem, de forma integrada ou separada, as competências de resolução de problemas, a aprendizagem cooperativa e a RV;
- pesquisar sobre fundamentos de experimentação por meio de laboratórios virtuais e sobre conceitos de objetos de aprendizagem;
- investigar sobre os diferentes tipos de sistemas multi-usuários para RV existentes e suas respectivas técnicas de desenvolvimento;
- investigar técnicas e ferramentas de RV com o objetivo de criar uma interface intuitiva para o Laboratório Virtual Cooperativo de Física (LVCF), a ser desenvolvido nesta pesquisa. Este laboratório deve possibilitar aos alunos a construção coletiva de cenários virtuais de aprendizagem, a manipulação intencional de objetos virtuais e experimentação de fenômenos físicos em sistemas distribuídos. Para tanto deve-se desenvolver o desenho pedagógico do LVCF para aprendizagem de Física, considerando as teorias de aprendizagem cooperativa, resolução de problemas e conceitos de experimentação em laboratórios virtuais por meio RV;

- elaborar a arquitetura computacional do LVCF, assim como gerar a sua documentação de engenharia de *software*;
- elaborar a arquitetura de distribuição do sistema, e pesquisar sobre as ferramentas e técnicas a serem aplicadas;
- desenvolver o LVCF de acordo com as especificações do padrão SCORM – Sharable Content Object Reference Model;
- aplicar e avaliar o LVCF segundo os parâmetros e modelos contemporâneos de avaliação de *softwares* educacionais. Avaliar, ainda, os modelos pedagógicos, baseados na aprendizagem cooperativa e resolução de problemas, desenvolvidos para o LVCF, verificando neste contexto o nível de aprendizagem obtida pelos usuários em decorrência do uso do *software*.

1.4 Contribuições da tese

O desenvolvimento deste trabalho tem como principais contribuições os seguintes resultados:

- elaboração de estratégias computacionais e aplicação de metodologias pedagógicas que permitam criar ambientes de RV como suporte para a aprendizagem cooperativa de resolução de problemas. Em geral, as contribuições deste trabalho encontram-se mediante a interseção das três áreas de pesquisa e conhecimento: Realidade Virtual, Aprendizagem Cooperativa e Resolução de Problemas, como ilustra a Figura 1.1;

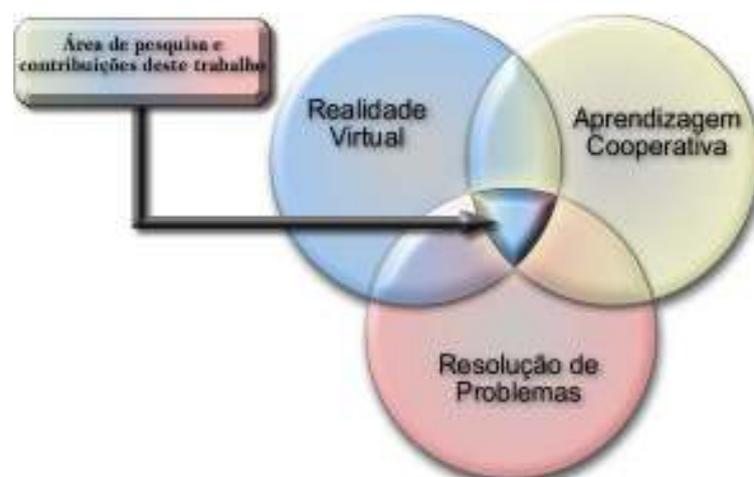


Figura 1.1: Área de pesquisa e contribuições do LVCF.

O presente trabalho visa contribuir para as áreas multidisciplinares envolvidas no desenvolvimento deste projeto, Computação, Física e Educação, conforme a Figura 1.2.



Figura 1.2: Contribuições para as áreas multidisciplinares do trabalho.

- construção de um laboratório virtual para o ensino de Física, com base em técnicas de RV, teorias da aprendizagem cooperativa e resolução de problemas;
- demonstração de que a experimentação por meio de laboratórios virtuais, baseada em conceitos de objetos de aprendizagem, pode contribuir para o aprendizado;
- criação de ambientes virtuais distribuídos com interfaces ergonômicas e desenvolvimento de estratégias computacionais que permitem a geração de ambientes virtuais de baixo custo de transmissão e armazenamento;
- desenvolvimento de uma ferramenta computacional, baseada em modelos pedagógicos contemporâneos, que permita verificar a validade da RV para a resolução de problemas em sistemas distribuídos;
- verificação de que a aplicação das teorias pedagógicas de aprendizagem cooperativa e resolução de problemas ligadas à RV pode contribuir positivamente para o processo de ensino e aprendizagem;
- validação das estratégias computacionais e metodologias pedagógicas decorrente de uma avaliação sistemática de alunos potenciais, por meio de grupos de controle, em amostras estatisticamente válidas;
- criação de um *software* que contribui para o ensino de Física, com baixo custo de utilização.

1.5 Estrutura da tese

Este volume foi dividido em sete capítulos, a saber:

No Capítulo 1, Introdução, registra-se as considerações iniciais sobre as condições e possibilidades para a Educação no mundo, e são discutidas a motivação, os objetivos, as contribuições dessa pesquisa e a organização da tese.

No Capítulo 2, aborda-se as fundamentações cognitivas da aplicação da RV na Educação, as metodologias e conceitos intrínsecos a objetos de aprendizagem, aprendizagem cooperativa e resolução de problemas, e, ao finalizar, discute-se a fundamentação pedagógica do sistema construído.

No Capítulo 3, discute-se o estado atual do *software* educacional em relação às competências de resolução de problemas, à aprendizagem cooperativa e à RV, analisando suas vantagens e limitações junto ao processo de ensino e de aprendizagem.

No Capítulo 4, apresenta-se a documentação referente à engenharia de *software* usada para arquitetar o LVCF, descreve-se a arquitetura geral do sistema e as demais arquiteturas derivadas desta, que são as comportamentais e a de distribuição.

No Capítulo 5, abordam-se as ferramentas e técnicas utilizadas para o desenvolvimento do sistema, e apresenta-se a documentação referente à engenharia de *software* usada na implementação do sistema.

O Capítulo 6 apresenta a metodologia de avaliação do sistema e os resultados obtidos com a sua aplicação.

O Capítulo 7 apresenta as conclusões alcançadas ao longo da pesquisa e as sugestões para trabalhos futuros. Finalmente, são apresentadas as referências bibliográficas utilizadas para redação deste volume.

Capítulo 2

Fundamentos teóricos

2.1 Introdução

O projeto LVCF é fruto de uma pesquisa de caráter multidisciplinar, visando o uso da RV como suporte à aprendizagem cooperativa para apoiar a resolução de problemas. Com intuito de facilitar a compreensão dos fundamentos, procedimentos e efeitos desta aplicação, este capítulo apresenta os subsídios pedagógicos e cognitivos que formam o aporte teórico do presente trabalho. Para tanto, são aduzidos conceitos e teorias envolvendo RV, cognição, objetos de aprendizagem, ambientes cooperativos, resolução de problemas, entre outros.

2.2 Realidade virtual e Educação

Existem diversas definições sobre RV, as quais envolvem aspectos científicos variados. Uma abordagem que apresenta uma visão geral e que sintetiza as várias discussões de RV é apresentada por Kirner e Siscoutto (2007): “Realidade Virtual é uma interface avançada para aplicações computacionais, que permite ao usuário navegar, manipular e interagir, em tempo real, com um ambiente tridimensional gerado por computador” [Kirner e Siscoutto, 2007].

É importante complementar, que a interação do usuário com o ambiente virtual é um dos aspectos mais importantes da interface baseada em RV, visto que ela está relacionada com a capacidade do computador detectar e reagir às ações do usuário, promovendo alterações na aplicação [Bowman *et al.*, 2005].

A navegação é um tipo de interação mais simples, que decorre da movimentação do usuário no espaço tridimensional, por meio de algum dispositivo, como o mouse 3D, comandos de voz ou de gestos detectados por algum dispositivo de captura, resultando na visualização de novos pontos de vista do cenário. Nesse caso, não há mudanças no ambiente virtual, mas somente um passeio exploratório. Interações, propriamente ditas, com alterações no ambiente virtual, ocorrem quando o usuário entra no espaço virtual das aplicações e visualiza, explora, manipula e aciona ou altera os objetos virtuais, usando seus sentidos,

incluindo os movimentos tridimensionais de translação e rotação naturais do corpo humano [Kirner e Siscoutto, 2007].

A interface baseada em RV permite que habilidades e conhecimentos intuitivos do usuário possam ser utilizados para a manipulação dos objetos virtuais. Esse tipo de interação é realizado através de dispositivos não convencionais, como capacete de visualização ou luvas, o próprio corpo, como gestos e comandos de voz, ou até mesmo dispositivos convencionais como mouse, teclado e monitor de vídeo. O usuário deve ter a impressão de estar atuando no ambiente virtual, apontando, pegando, manipulando e executando outras ações sobre os objetos virtuais, em tempo-real. Esta experiência no virtual permite que o usuário possa aprender a partir da experimentação e descoberta.

Segundo Freire (1997) “aprender é uma descoberta do novo, com abertura ao risco, à aventura e a novas experiências, pois ensinando se aprende e aprendendo se ensina” [Freire, 1997]. Portanto, a educação pode ser vista como um processo de descoberta, exploração e de observação, além de eterna construção do conhecimento [Ficheman *et al.*, 2006].

De acordo com Mantoan (1999) é preciso criar cada vez mais motivos para que a tecnologia e a educação se encontrem e integrem seus propósitos e conhecimentos, buscando complementos, uma na outra [Mantoan, 1999]. Neste contexto, a RV predispõe-se como um instrumento valioso no auxílio à educação, pois suas características específicas contemplam significativamente os processos de ensino e aprendizagem, uma vez que estes podem ser vistos como um conjunto de procedimentos de descoberta, exploração, observação, e construção do conhecimento. Por exemplo, a RV possibilita a aprendizagem de lugares não-visitados, estudar sobre um tema inserido no seu contexto, e oferecer um *feedback* das atividades do usuário a cada ação que este fizer no ambiente de aprendizagem [Braga, 2001].

Estudos mostram ainda, que a RV influencia a estrutura cognitiva e mental do usuário de maneira positiva para o aprendizado. Tais estudos e seus resultados são apresentados na próxima seção.

2.2.1 Realidade virtual e cognição

De acordo com os estudos descritos por Bullinger, Muller-Spahn e Rößler (1996) a RV apresenta-se como uma ferramenta que contribui significativamente para a aquisição de conhecimento de forma implícita, pois incentiva a aprendizagem processual [Bullinger, Müller-Spahn e Rößler, 1996].

Em suma, o ser humano armazena e aprende informações por meio de dois tipos de memórias: a declarativa e a processual. O processo de aprender e armazenar a informação na memória declarativa ou processual ocorre, respectivamente, através da aprendizagem declarativa ou processual. A memória declarativa normalmente é usada para fatos e eventos, como por exemplo, a lembrança de acontecimentos e números de telefone; ela possui uma forma explícita, neste contexto é possível declarar o seu conteúdo. Já a memória processual é usada para procedimentos e habilidades, como dirigir, jogar bola ou dar o nó da gravata; ela apresenta uma forma implícita, ou seja, o seu conteúdo torna-se aparente através de atos comportamentais. De acordo com Silva (1994), a memória declarativa usada para datas ou fatos históricos é formada com a mesma facilidade com que é esquecida, enquanto a memória processual, usada para a aprendizagem de habilidades, caracteriza pelo fluxo inconsciente de artefatos mentais e de repetições freqüentes e regulares. Por isso, as pessoas podem esquecer facilmente, por exemplo, uma data de aniversário, mas dificilmente elas ficarão em dúvida se devem pisar na embreagem ou no freio na hora de parar um carro no sinal de trânsito [Silva, 1994].

É importante ressaltar que a RV apóia a aprendizagem processual de forma inconsciente, o que contrasta com os métodos pedagógicos convencionais, que apóiam, principalmente, a aquisição explícita de conhecimento de forma controlável e consciente [Hofmann, Hock e Müller-Spahn, 1995].

Por acreditar no potencial e nos benefícios da RV para os processos mentais de criatividade e aprendizagem, os pesquisadores da *University Clinic of Psychiatry of Basle* e da *Fraunhofer Institute for Industrial Engineering* investiram esforços notórios para o desenvolvimento de um projeto de pesquisa cooperativo para analisar os efeitos de experiências virtuais sobre a mente humana. Este projeto visa contribuir na concepção de um laboratório de conhecimentos interativos, desenvolvido pela *Information Awareness Office (IAO)* in mid-1996 nos Estados Unidos [Bullinger, Müller-Spahn e Rößler, 1996].

A RV facilita e torna mais intuitivos os processos de abstração e transmissão de informações, pois ela mantém coerente ao modelo mental humano os seus modos de interação e representação [Bullinger, Müller-Spahn e Rößler, 1996]. As pessoas ao interagirem no ambiente real por meio de translação, rotação e visualização em 3D criam um modelo mental, isso significa que elas abstraem conceitos e informações do ambiente real e as representam mentalmente em 3D. Considerando estas ponderações, conclui-se que o produto computacional que apresenta maior compatibilidade ao modelo mental do usuário e viabiliza

uma melhor visualização, abstração, concretização da informação em modelos mentais é um protótipo que possua uma representação tridimensional e permita interações em 3D. Como mostra a Figura 2.1, a RV e CAD¹ contemplam de forma tridimensional a interação e a visualização, ao contrário de protótipos 2D que operam apenas com recursos em 2D.

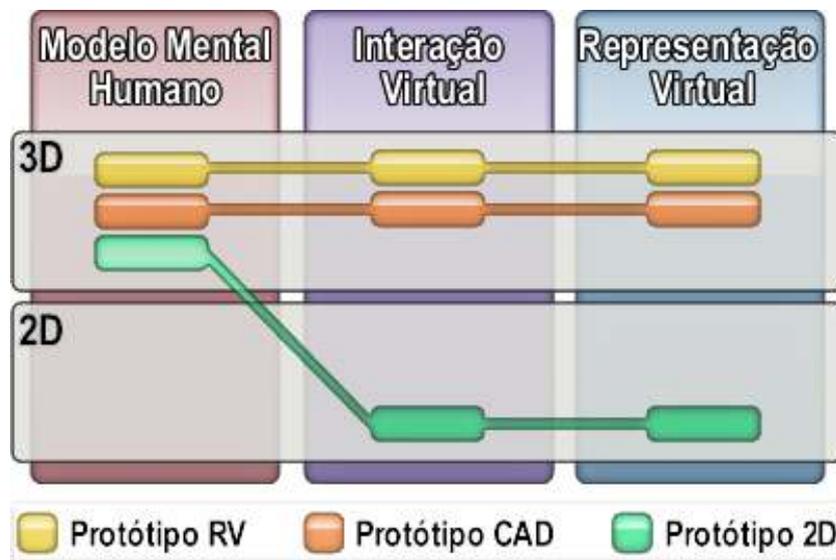


Figura 2.1: Modelo mental, Interação e Representação. Modificado de: [Bullinger, Müller-Spahn e Rößler, 1996]

Um ambiente em RV é composto por objetos virtuais que oferecem respostas comportamentais quando o usuário interage no cenário. Estas respostas caracterizam um dos principais diferenciais da tecnologia de RV em relação a outras, como a CAD, por exemplo. Caso a aplicação em RV tenha fins de ensino, estes objetos virtuais para serem considerados objetos de aprendizagem devem ser construídos de acordo com algumas diretrizes pedagógicas. Um breve resumo sobre os conceitos principais relacionados à teoria de objetos de aprendizagem encontra-se na próxima seção.

2.2.2 Objetos virtuais de aprendizagem

Segundo Miranda (2004) objetos virtuais de aprendizagem são elementos do ensino baseados no computador e na Internet, sendo que, a sua origem na educação surgiu a partir do paradigma de orientação a objetos da Ciência da Computação [Miranda, 2004]. Existem várias interpretações deste conceito, no entanto, optou-se por esta de Miranda (2004) por

¹ As tecnologias CAD – *Computer-Aided Design* são sistemas computacionais usados pela engenharia, geologia, arquitetura, e design para facilitar o desenvolvimento de projetos e desenhos técnicos. Os *softwares* CAD possibilitam ao usuário desenhar apenas em 2 dimensões ou criar modelos tridimensionais por meio de um modelador 3D. Nestes *softwares* pode haver o intercâmbio entre o modelo 3D e o desenho 2D (por exemplo, o modelo 3D pode ser gerado automaticamente a partir do desenho 2D ou vice-versa).

³ Pode-se encontrar, em algumas literaturas, a designação Ambiente Virtual Colaborativo sinonimamente no lugar do termo Ambiente Virtual Cooperativo.

acreditar que ela melhor se adequou aos propósitos desta pesquisa. A orientação a objetos valoriza a criação de componentes (objetos) que podem ser reutilizados em diversos contextos; esta é a idéia principal por trás de objetos de aprendizagem: educadores têm a possibilidade de criar componentes educacionais que podem ser reutilizados várias vezes em diferentes contextos de aprendizagem [Wiley, 2000]. Sendo assim, objetos de aprendizagem são definidos como qualquer entidade, digital ou não-digital, que pode ser utilizada, reutilizada ou referenciada durante o processo de aprendizagem que tenha uma tecnologia como suporte [Ieee, 2002].

No presente projeto LVCF os objetos virtuais, que compõem o laboratório virtual de experimentação, são construídos com base nessa teoria. Por exemplo, o cronômetro virtual do LVCF, Figura 2.2, pode ser usado tanto em experiências de termodinâmica, quanto em experiências de mecânica.



Figura 2.2: Cronômetro virtual.

Como exemplo de aprendizagem por meio de objetos citam-se os sistemas de treinamento por computador, ambientes de aprendizagem interativos, sistemas de ensino a distância e ambientes de aprendizagem colaborativa e cooperativa [Miranda, 2004].

2.2.3 Laboratórios virtuais e Realidade Virtual

A informática oferece um conjunto com possibilidades muito diversificadas para o ensino da Física e outras ciências, podendo ser usada para medições, gráficos, avaliações, apresentações, modelagens, animações e simulações [Rogers, 1996]. No entanto, diversos pesquisadores admitem que há equívocos em relação à confecção de *softwares* educacionais, devido a falta de cuidados no seu desenvolvimento e implementação, ou ainda, devido a uma falta de conhecimento em Física que pode ocorrer e conduzir os alunos a pensarem de modo incorreto[Vrankar, 1996].

É preciso assinalar ainda que a simples utilização da informática não garante que os estudantes tenham uma boa aprendizagem. De acordo com Medeiros e Medeiros (2002) toda experiência proposta aos estudantes tem, intencionalmente ou não, uma abordagem que corresponde a esquemas conceituais, epistemológicos, pedagógicos e psicológicos [Medeiros e Medeiros, 2002].

Críticos mais severos do uso irrefletido da informática na educação têm chamado a atenção para o fato de que as simulações computacionais parecem limitar a possibilidade de os estudantes serem confrontados com a riqueza heurística da experiência dos erros experimentais e, assim, da tentativa de resolverem problemas da vida real [Kimbrough, 2000].

Neste contexto, Medeiros e Medeiros (2002) destacam que ao construir simulações virtuais para o ensino, é necessário considerar criticamente quais as características do sistema modelado que poderiam ser negligenciadas e quais aquelas que deveriam ser incluídas no modelo [Medeiros e Medeiros, 2002]. O valor educacional de uma simulação dependerá diretamente do fato de ela poder vir a representar para o estudante um papel de auxiliar heurístico e não apenas cumprir um papel algorítmico ou meramente ilustrativo. Estas características são totalmente pertinentes à resolução de problemas, na qual a aprendizagem pela heurística é um de seus fundamentos básicos.

Deste modo, pode-se concluir que simulações computacionais voltadas para o ensino devem incentivar a aprendizagem por meio da heurística, ou seja, elas devem apoiar fortemente que o aluno aprenda através da experimentação. Um ambiente condizente com esta realidade é um laboratório virtual, que permite o acompanhamento e o controle de experimentos e ensaios remotamente [Arpaia *et al.*, 1997], [Waller e Foster, 2000].

Segundo [Bottentuit e Coutinho, 2006] a criação dos laboratórios virtuais surgiu principalmente da necessidade do uso dos laboratórios em tempo real, com acesso a qualquer hora do dia e por um grande número de pessoas, uma vez que um único experimento pode ser compartilhado por dezenas de pessoas, estando elas na mesma cidade ou geograficamente dispersas.

Assim, a RV é uma tecnologia que pode auxiliar consideravelmente no desenvolvimento de laboratórios virtuais, devido as suas características básicas, como, por exemplo, permitir ao usuário visualizar, explorar, manipular e acionar ou alterar objetos virtuais, e observar os efeitos destas alterações.

2.3 Aprendizagem cooperativa

Existem diversas teorias que abordam a aprendizagem coletiva e destacam a importância do estudo e do trabalho em grupo no processo de ensino e aprendizagem, e no desenvolvimento sociocultural dos aprendizes. Em levantamentos realizados em relação às recordações de experiências de aprendizagem, constatou-se que a maioria das pessoas lembra de situações envolvendo a aprendizagem que requer alguma ação coletiva [Larocque e Faucon, 1997].

Estes estudos sobre a cooperação e colaboração entre aprendizes partem da ideia de que o conhecimento é resultante de um consenso entre membros de uma comunidade de conhecimento, algo que as pessoas constroem pela comunicação, trabalhando juntas direta ou indiretamente e chegando a coordenações de pontos de vista [Torres, Alcantara e Irala, 2004].

De acordo com Jones e Jacobs (2000) a cooperação ocorre quando dois ou mais aprendizes trabalham juntos com um objetivo comum, para alcançar, efetivamente, o máximo de seus próprios objetivos [Jones e Jacobs, 2000]. A cooperação deve correr simplesmente porque o estudante não pode aprender sozinho, pois a execução de tarefas requer a integração de especializações baseadas em técnicas diferentes, sendo assim, demanda debates entre perspectivas múltiplas. Em um grupo podem ocorrer a complementação de capacidades, de conhecimentos e de esforços individuais, e a interação entre pessoas com entendimentos, pontos de vista e habilidades complementares [Gerosa, Fuks e Lucena, 2002].

Frequentemente encontram-se obras nas quais os termos cooperação e colaboração são usados como sinônimos, porém, cada um destes termos, ao longo dos anos, desenvolveu distinções próprias e diferentes aplicações práticas [Stahl, Koschmann e Suthers, 2006]. Alguns pesquisadores acreditam que o termo cooperação é mais abrangente, com distinções hierárquicas de ajuda mútua, ao passo que na colaboração existe um objetivo comum entre as pessoas que trabalham em conjunto sem uma hierarquia [Nitzke *et al.*, 1999].

Muitos autores na literatura atual definem a aprendizagem cooperativa como uma aprendizagem mais estruturada, com técnicas de sala de aula mais prescritivas e com regras mais definidas de como deve se processar a interação entre os alunos, se comparada com a aprendizagem colaborativa [Oxford, 1997]. Por exemplo, Olsen e Kagan (1992) define a aprendizagem cooperativa como uma atividade de aprendizagem em grupo organizada de tal maneira que a aprendizagem seja dependente da troca de informações socialmente estruturada entre os alunos em grupos e na qual cada aluno é responsável por sua própria aprendizagem e

é motivado a contribuir com a aprendizagem dos outros [Olsen e Kagan, 1992]. Harasim (1989) define a aprendizagem colaborativa como qualquer atividade na qual duas ou mais pessoas trabalham juntas para criar significado, explorar um tópico ou melhorar habilidades, ou seja, pode-se generalizar a idéia de que qualquer atividade desenvolvida em conjunto animada por um objetivo final que leve a aquisições determinadas é uma situação de aprendizagem colaborativa [Harasim, 1989].

De acordo com Torres, Alcântara e Irala (2004) a cooperação apresenta-se como um conjunto de técnicas e processos que grupos de indivíduos aplicam para a concretização de um objetivo final ou a realização de uma tarefa específica, e este processo é mais direcionado do que o processo de colaboração e mais controlado pelo professor [Torres, Alcântara e Irala, 2004]. Gerosa, Fuks e Lucena (2003) ressaltam que sem coordenação e mediação pedagógica, boa parte dos esforços de comunicação não será aproveitada na cooperação, isto é, para que o grupo possa operar em conjunto de forma satisfatória, é necessário que os compromissos assumidos nas conversações entre os participantes sejam realizados durante a cooperação [Gerosa, Fuks e Lucena, 2003].

Neste trabalho adota-se a expressão “aprendizagem cooperativa” segundo os estudos de Mendes (2008:9) [Mendes, 2008]:

Segundo a epistemologia genética, o conceito de cooperação enfatiza que a vida social transforma a inteligência pela mediação da linguagem (signos), do conteúdo dos intercâmbios (valores intelectuais) e das regras impostas ao pensamento (normas coletivas, lógicas e pré-lógicas) (Piaget, 1977; Grize, 1994). A cooperação, ou seja, a coordenação de pontos de vista entre um grupo de indivíduos, é o ponto de partida de uma série de condutas importantes para a constituição e o desenvolvimento do pensamento formal (Piaget, 1977). Sem intercâmbio do pensamento e sem cooperação com os outros, o indivíduo não chegaria a agrupar suas operações em um todo coerente; como afirma Piaget (1977: 164), “seria muito difícil compreender como o indivíduo conseguiria agrupar suas operações de representações intuitivas em operações transitivas reversíveis, idênticas e associativas sem o intercâmbio do pensamento lógico”. Essa coordenação de pontos de vista se processa por colaborações no trabalho, pela troca de idéias e pelo controle mútuo.

Nesse sentido, o conceito de cooperar é mais abrangente que colaborar por que envolve as trocas de idéias, o trabalho colaborativo e os exercícios de pensamento formal, como as operações transitivas reversíveis, idênticas e associativas (Piaget, 1977). Esses processos cognitivos complexos são estimulados por situações de aprendizagem que requerem transferência e não somente retenção (Mayer, 2002), essas ações de aprendizagem são classificadas no ensino em processos cognitivos de entender, aplicar, analisar, avaliar, criar e seus respectivos subprocessos (Anderson *et al.*, 2001). As operações cognitivas de soluções de problemas requerem processos mais complexos de transferência visto que estão intrinsicamente relacionadas ao conhecimento metacognitivo, conceitual, específico e procedimental (Mayer, 2002); portanto são reconhecidas como um dos principais objetivos educacionais¹ (Mayer, 2002).

Mendes (2002) considera que no contexto da aprendizagem cooperativa apoiada por computador existem dois grandes desenhos pedagógicos contemporâneos que estimulam a aprendizagem cooperativa e distribuída [Mendes, 2002]. Uma dessas novas propostas de ensino é o modelo *Computer-Supported Collaborative Learning Design*, um desenho pedagógico que combina teorias da aprendizagem cooperativa com tecnologias da comunicação. Nesse modelo, os alunos de forma cooperativa constroem o conhecimento e projetam o desenho de seus ambientes de aprendizagem. Dessa forma, a aprendizagem não é um processo individual, linear, cíclica ou repetitiva; começa com um protótipo rudimentar que é gradualmente refinado pelos usuários.

Outra proposta é o modelo *Distributed Knowledge Design*. A expansão da Internet originou a maior mudança nos sistemas de ISD (*Instructional System Design*). A combinação entre ambiente de conhecimento distribuído, informação e tecnologias revoluciona a aprendizagem. Atualmente não é somente o indivíduo que oferece significado ao ambiente, mas sim os aprendizes, de forma cooperativa em fluxo contínuo de informação e comunicação, constroem o conhecimento por meio da Internet. O modelo *Distributed Knowledge Design* atrai a atenção de pesquisadores e professores porque contrasta com a noção aceita de que o conhecimento e a cognição residem “na cabeça de cada indivíduo”. O conceito de cognição distribuída significa que a cognição é distribuída entre indivíduos e seus ambientes. O conhecimento distribuído torna-se predominante em ambientes de multimeios, em que as informações são distribuídas por diferentes tipos de recursos. Dessa forma, o sistema oferece ampla oportunidade para os aprendizes cooperarem com seus diferentes pares, de forma assíncrona ou síncrona, com professores, especialistas, comunidades, etc.

Existe uma grande expectativa em relação à aplicação da RV em ambientes cooperativos. De acordo com Oravec (1996) e Barnatt (1995) esta aplicação pode ocorrer por meio de um sistema computacional denominado Ambiente Virtual Cooperativo³ (AVC), [Oravec, 1996], [Barnatt, 1995].

Um AVC usa a tecnologia de RV distribuída para oferecer suporte ao trabalho em grupo. De acordo com Benford e Greenhalgh (1997) o AVC deve possibilitar o acesso simultâneo de multi-usuários a um sistema de RV que permite realizar um trabalho cooperativo [Benford e Greenhalgh, 1997]. Trefftz (2006) ressalta que um aspecto importante neste tipo de sistemas é a existência de um espaço virtual, um mundo não material, que permite interação a distância para múltiplos usuários através de computadores ligados em rede [Trefftz *et al.*, 2006].

Por fim, Gerosa, Fuks e Lucena (2003) destacam que em atividades educacionais realizadas de forma cooperativa, os membros do grupo têm retorno para identificar precocemente inconsistências e falhas em seus raciocínios e, juntos, podem buscar novas idéias, informações e referências, o que auxilia, de modo geral, trabalhos envolvendo a resolução de problemas [Gerosa, Fuks e Lucena, 2003]. Conceitos e metodologias de resolução de problemas são apresentados na próxima seção.

2.4 Resolução de problemas

A humanidade confronta-se com a palavra “problema” há milhares de anos. No entanto, a sua definição propicia a inferência de diversas interpretações. Por exemplo, problema pode ser considerado como um estado subjetivo da mente inerente a cada indivíduo, um desafio, uma situação não resolvida, cuja resposta não é imediata, que resulta em reflexão e uso de estratégias conceituais e procedimentais, provocando uma mudança nas estruturas mentais [Costa e Moreira, 1995]. Hayes (1980) definiu problema como a fenda que separa um estado presente de um estado almejado [Hayes, 1980]. Pérez, Terrogrosa e Senent (1988) considera problema como uma situação para a qual não há soluções evidentes [Pérez, Torregrosa e Senent, 1988], já Perales (1993) o define como uma situação qualquer que produz, de um lado, certo grau de incerteza e, de outro, uma conduta em busca de uma solução [Perales, 1993].

Os processos cognitivos realizados pela mente humana no tratamento e na resolução de problemas despertaram interesses no campo da Psicologia, que iniciou intensos estudos sobre tais processos. Estes estudos demonstraram que o indivíduo adquire um aprendizado decorrente destes processos cognitivos, [Bono, 1973] acredita que processos criativos e de aprendizagem devem iniciar-se efetivamente por meio de um problema. Com isto, esta área de conhecimento tornou-se atrativa e aplicável ao ensino e serviu como suporte para a criação de metodologias de ensino denominadas resolução de problemas. Essas metodologias têm despertado o interesse de pesquisadores de varias áreas do ensino como, por exemplo, Biologia, Matemática, Física, etc. Este interesse pode ser atribuído ao fato de que a resolução de problemas contribui significativamente para as atividades de raciocínio e pensamento exigidos nestas disciplinas [Costa e Moreira, 1995].

Alguns pesquisadores consideram a resolução de problemas como “um processo pelo qual o aprendiz descobre uma combinação de regras anteriormente aprendidas, que ele pode aplicar para atingir uma solução para uma situação problemática nova” [Gagné, 1965]. Este

processo deve favorecer a aprendizagem significativa na medida em que propicia uma reorganização da informação e do conhecimento armazenado na estrutura cognitiva do sujeito [Novak e Tyler, 1986].

De acordo com Turoff, Rao e Hiltz (1991) a resolução de problemas em sistemas cooperativos é distribuída entre os aprendizes que cooperam entre si, pois eles dividem e compartilham os conhecimentos sobre o processo de obtenção da solução [Turoff, Rao e Hiltz, 1991].

Na próxima seção são apresentados alguns modelos teóricos de resolução de problemas consagrados historicamente.

2.4.1 Modelos teóricos de resolução de problemas

Nesta seção são extraídos na íntegra os estudos de Victor Gargiulo e da professora Dra. Elise Barbosa Mendes da Universidade Federal de Uberlândia sobre resolução de problema a partir dos modelos teóricos de Pozo (uma revisão da obra de Polya), Bransford e Stein, Backer e Mayer, e Kapa. Estes autores destacam-se no desenvolvimento de teorias de resolução de problemas, assim, nas subseções abaixo serão apresentados os modelos teóricos propostos por eles.

2.4.1.1 O modelo de Pozo e Polya

Pozo (1998) define a atividade de resolução de problemas como sendo subjetiva, exigindo motivação e uma postura de investigação e de reflexão constantes [Pozo *et al.*, 1998]. Caso não haja disposição, o problema se transformará em um simples exercício, executado mecanicamente e destituído de significado. Para tanto, é fundamental

*(...) planejar situações suficientemente abertas para induzir nos alunos uma busca e apropriação de estratégias adequadas não somente para darem resposta a perguntas escolares como também às da realidade cotidiana. Sem procedimentos eficazes — sejam habilidades ou estratégias — o aluno não poderá resolver problemas [Pozo *et al.*, 1998].*

O modelo de resolução de problemas usado por Pozo é de George Polya, famoso matemático que tratou em diversos livros sobre os processos em solução de problemas. Polya define a resolução de problemas como a busca de um meio para se alcançar um objetivo que está cercado de dificuldades e obstáculos. Tais obstáculos são delimitados no modelo proposto por Polya e revisado por Pozo, considerando que a resolução de problemas ocorre de

acordo com quatro fases: a *compreensão do problema*, a *concepção de um plano*, a *execução do plano* e a *retrospecção*.

Na fase de *compreensão* do problema é preciso distinguir quais dados são oferecidos, qual a linguagem usada e em que contexto e condições o problema está definido, sendo necessário perceber que se está à frente de uma nova situação e que não se tem todos os elementos necessários para explicá-la.

Na *concepção* de um plano, o solucionador deve se perguntar qual a distância entre a situação da qual partimos e a meta à qual pretendemos chegar, e quais são os procedimentos mais úteis para diminuir essa distância. Para tanto, são essenciais dois tipos de conhecimentos: heurísticos e algorítmicos. O conhecimento heurístico envolve métodos e meios gerais de solução (como o uso de variações do problema em relação a outros conhecidos, com generalização, particularização e analogia), enquanto que o conhecimento algorítmico é voltado à transformação direta da informação fornecida, sendo também conhecido como conhecimento de regras ou operações.

Na *execução* desenvolve-se o plano que havia sido previamente elaborado e transforma o problema por meio de regras algorítmicas conhecidas. O problema pode ser transformado em outro, a partir da variação dos elementos que são conhecidos ou não.

Na fase de *retrospecção*, o estudante precisa rever os procedimentos de solução, relacionando-os a outros problemas, identificando semelhanças e diferenças.

De acordo com Pozo (1998) é essencial procurar outros contextos em que os elementos de determinado problema podem ser transferidos com sucesso. O processo de solução não pode terminar na execução do plano, é essencial que se continue explorando o problema em novos contextos.

2.4.1.2 O modelo de Bransford e Stein

Bransford e Stein (1993) estudam as principais teorias científicas em resolução de problemas e exemplificam aplicações e referências acessíveis ao público em geral, tornando-as compreensíveis e úteis [Bransford e Stein, 1993].

Para esse autores, há uma distinção entre problemas rotineiros e problemas não-rotineiros. A preocupação de suas pesquisas direciona-se aos não-rotineiros — situações problemas que apresentam discrepância entre um estado inicial e uma meta, não havendo uma

solução evidente, o que requer um esforço maior na busca, entendimento, aplicação e transformação de informações [Bransford e Stein, 1993].

O modelo proposto por Bransford e Stein (1993), usualmente chamado de Modelo Ideal, possui como componentes principais de resolução de problemas: *identificar problemas e oportunidade; definir metas; explorar possíveis estratégias; antecipar resultado e agir; e observar os efeitos e aprender* [Bransford e Stein, 1993].

O componente *identificar problemas* e oportunidades envolve a análise de uma determinada situação e a capacidade de identificar nela possíveis oportunidades de se fazer algo novo.

Definir metas refere-se como o problema foi compreendido e envolve a necessidade de se estabelecer os objetivos a serem alcançados. Nessa fase, o problema toma forma e os possíveis caminhos a se percorrer começam a aparecer. Muitas pessoas passam direto da identificação de problemas para a exploração de estratégias sem definir metas, sendo que este processo é que orientará a escolha das estratégias.

O componente *explorar possíveis estratégias* refere-se à maior parte do processo de solução proposto pelo modelo ideal. Neste componente, o solucionador precisa reconhecer os conhecimentos relevantes, determinar quais as melhores formas de solucionar o problema, definir os critérios de uso dos procedimentos conhecidos, representar o problema de outras formas, de modo a tornar seu enunciado mais claro, determinar a adequação das hipóteses traçadas durante a definição do problema, e criar um plano de ação.

Antecipar resultados e agir é um componente importante, pois antes de executar o plano traçado, é preciso que se antecipem possíveis conseqüências das estratégias escolhidas, sempre se perguntando se são suficientes ou adequadas. Desta forma, a solução se fará de forma mais precisa e com menos imprevistos.

O componente *observar e aprender* envolve a verificação da estratégia escolhida e a aprendizagem a partir da experiência.

2.4.1.3 O modelo Backer e Mayer

Baker e Mayer (1999) tratam dos componentes necessários à avaliação da solução de problemas em ambientes virtuais [Baker e Mayer, 1999]. De acordo com estes autores, a resolução de problemas é um processamento cognitivo direcionado à transformação de uma

situação dada em uma situação desejada, quando nenhum método de solução óbvio está disponível para o solucionador.

Baker e Mayer (1999) definem os seguintes componentes como principais para a resolução de problemas: a *translação*, *integração*, *planejamento* e *execução do problema*. De acordo com esses autores, a *translação* do problema ocorre quando um solucionador converte uma sentença de um enunciado para uma representação interna, este processo requer conhecimento *factual e conhecimento lingüístico* [Baker e Mayer, 1999]. A *integração* do problema ocorre quando se constrói um modelo mental de uma situação descrita no enunciado do problema, este processo requer *conhecimento esquemático*, ou seja, conhecimento de diversas formas de problemas. O *planejamento do problema* envolve criar e monitorar um plano de solução, isto é, determinar os passos a serem tomados; exigindo-se conhecimento estratégico de como quebrar o problema em partes. A *execução* do problema ocorre quando o plano é colocado em ação, baseando-se portanto no *conhecimento procedimental*.

2.4.1.4 O modelo de Kapa

Muitos autores indiretamente associam a metacognição aos processos de solução de problemas como parte fundamental de elaboração e de desenvolvimento das habilidades que lhes são inerentes. Kapa (2002) também refere-se à metacognição de forma explícita ao descrever o funcionamento de um *software* voltado à resolução de problemas em matemática. Este *software* oferece um suporte metacognitivo específico para cada processo de solução, permitindo aos estudantes receberem um *feedback* imediato para cada etapa. [Kapa, 2002].

Kapa (2002) delimita os seguintes processos de resolução de problemas: *identificar o problema*, *representá-lo mentalmente*, *planejar como solucioná-lo*, *executar a solução*, *avaliar*, e *reagir ao feedback* [Kapa, 2002].

No processo de *identificação do problema*, o estudante ao ler o enunciado, aos poucos ativa os conhecimentos prévios acerca dos conceitos do problema, e codifica os elementos importantes.

A *representação mental* do problema envolve o processo de relacionar a informação recentemente adquirida com a informação assimilada no passado. É fundamental saber como combinar e integrar os diversos dados em um todo conectado. Essa informação é organizada por meio de inferência, imaginação e pensamento lógico.

O *planejamento da solução* é um processo em que se decide quais estratégias serão utilizadas e em que ordem elas irão aparecer na solução. Segundo Kapa (2002), há três formas

de raciocínio empregado no planejamento: ensaio-e-erro, passo-a-passo e holístico [Kapa, 2002]. Solucionadores mal-sucedidos criam um plano a partir de números e palavras-chave retiradas do problema (e acabam solucionando por ensaio-e-erro), enquanto os bem-sucedidos criam um modelo para a situação descrita (planejamento holístico) e se baseiam neste modelo ao planejar a solução.

Na *execução da solução*, os aspectos metacognitivos de um plano de solução são o monitoramento e o controle, e podem servir como guias para seleção de estratégias de solução ou como meios de organização de descobertas.

A resolução de um problema é vista como uma função do conhecimento matemático e do pensamento operacional, como distinção entre informação relevante e irrelevante, como aplicação de conhecimento metacognitivo e obtenção de controle dos procedimentos empregados no processo de solução. As habilidades em que diferem os estudantes são relacionadas ao conhecimento declarativo e procedimental: translação de informação, uso de heurísticas, generalização, uso de avaliação da solução e rapidez de cálculo mental [Kapa, 2002].

Na *avaliação* ocorre a interação entre uma pessoa, uma solução e uma estratégia. Ao solucionar um problema é preciso ter consciência do que tem sido feito, do que se está fazendo no momento e do que resta a fazer.

A *reação ao feedback* refere-se a três elementos: aumentar o rigor do julgamento, diminuir confiança excessiva e desenvolver a capacidade de retrospecção. Espera-se dos solucionadores uma reação positiva ao *feedback*, de modo a usá-lo para continuar o processo de solução. Sem um *feedback* de correção, os estudantes tornam-se propensos a repetir os mesmos erros anteriormente cometidos.

2.5 Desenho pedagógico do LVCF

A integração da teoria de aprendizagem cooperativa e da resolução de problemas requer o desenvolvimento de um ambiente computacional de aprendizagem que estimule os conhecimentos conceitual, procedimental e metacognitivo. A metacognição é considerada a espinha dorsal para a aprendizagem com sucesso. Refere-se à ação de planejar, monitorar e controlar o próprio aprendizado e transferir e generalizar o conhecimento adquirido para outros contextos, como também orientar a auto-reflexão, a responsabilidade e a iniciativa do aprendiz [Mendes, 2008]. Para tanto, estes conhecimentos podem ser estimulados por meio

da RV em sistemas distribuídos, já que existe a possibilidade da experimentação com objetos virtuais, da interação e da comunicação para solucionar problemas abertos pela coordenação de pontos de vistas (cooperação). Assim, conclui-se que o desenho pedagógico⁴ do LVCF deve estar no âmbito de três áreas de conhecimento: a RV (com ambientes computacionais em RV), a aprendizagem cooperativa (suportada por rede de computadores) e a resolução de problemas (aplicados à Física); como ilustra a Figura 2.3.



Figura 2.3. A localização do LVCF perante as três áreas do conhecimento.

2.5.1 Modelo de resolução de problemas proposto para o LVCF

A metodologia de ensino e o sistema de avaliação do desenho pedagógico, proposto para o LVCF, serão fundamentados nos estudos de resolução de problemas desenvolvidos pela equipe multidisciplinar da educação⁵.

Este estudo resulta na integração das teorias de Polya, Pozo, Bransford e Stein, Backer e Mayer, e Kapa em um modelo geral, que tem como matriz de compreensão a taxonomia revisada dos objetivos cognitivos. Nesta metodologia, os processos cognitivos e conhecimentos necessários a cada estágio da resolução dos problemas foram explicitamente definidos, diferentemente dos modelos de solução revisados, que focam mais nos passos de solução do que nos detalhes das tarefas. Para tanto, foram delimitadas três operações não-lineares inerentes à resolução de problemas, que são: *contato inicial*, *solução* e *avaliação*. Estas operações são apresentadas nas seções abaixo.

⁴ Desenho pedagógico é uma ferramenta que permite a descrição e o desenvolvimento de um projeto pedagógico para um ambiente de ensino.

⁵ Equipe composta por Victor Gargiulo e a professora Dra. Elise Barbosa Mendes da Universidade Federal de Uberlândia.

2.5.1.1 Contato Inicial

A operação *contato inicial* constitui-se de três processos fundamentais: a *identificação do problema*, a *representação mental do problema* e a *geração de hipóteses*. O processo de Identificação do Problema, da mesma forma como propõem Bransford e Stein (1993), é uma etapa de perceber a existência de um problema e de tratá-lo como uma oportunidade de criar algo novo [Bransford e Stein, 1993]. Os objetivos cognitivos envolvidos na *identificação do problema* são: interpretação e inferência de conhecimento factual (elementos e detalhes específicos), conceitual (princípios e generalizações) e metacognitivo (contexto e condições).

Para o processo de *representação mental do problema* Pozo (1998) definiu as seguintes prioridades: enunciar a questão, e identificar relações, informações fornecidas e omitidas [Pozo *et al.*, 1998]. Kapa (2002) ressalta a diferenciação entre informação relevante e irrelevante, bem como a ativação de conhecimento prévio relacionado às novas informações fornecidas pelo problema [Kapa, 2002].

Para Baker e Mayer (1999), este processo requer translação (ação que envolve conhecimento factual e lingüístico) e integração do problema (construção de um modelo mental que faz uso de conhecimento esquemático, ou seja, de tipos de problemas) [Baker e Mayer, 1999]. Para este modelo proposto, considera-se que além destes aspectos, a *representação mental* de um problema requer os seguintes conhecimentos e processos cognitivos: recordação, inferência, interpretação, explicação, organização de conhecimento factual e conceitual, comparação e diferenciação de elementos e detalhes específicos.

No processo de *geração de hipóteses*, o aluno precisa pensar em diversas alternativas de solução, afinal, quanto mais variadas forem, maiores as chances de se criar algo novo. Assim, entende-se que a *geração de hipóteses* envolve também explicação e geração de conhecimento conceitual.

2.5.1.2 Solução

A *solução* constitui-se de cinco processos fundamentais: a *busca e exploração de estratégias*, o *projeto de solução*, a *previsão de resultados*, a *aplicação* e o *gerenciamento da aplicação*.

Para o Modelo Ideal, o aprendizado de ferramentas conceituais é fundamental ao desenvolvimento dos processos de *exploração de estratégias*, afinal a experiência prévia em determinado campo de conhecimento é determinante do sucesso na escolha e experimentação de estratégias [Pozo *et al.*, 1998]. Sobre isso, Bransford e Stein (1993) lembram que

especialistas sabem melhor do que aprendizes “quando, como e porque aplicar um conjunto de estratégias” [Bransford e Stein, 1993].

Sendo assim, a *busca e exploração de estratégias* inicia-se pela a ativação de conhecimento prévio relacionado ao problema, bem como pela explicação e exemplificação de estratégias escolhidas, que são relacionadas ao contexto e condições do problema. Neste aspecto, Pozo (1998) recomenda que sejam procurados outros problemas mais simples ou alternativos [Pozo *et al.*, 1998]. Isto, de fato, pode ser útil para a ativação do chamado conhecimento inerte e para o desenvolvimento de estratégias alternativas mais complexas; relacionando-as à verificação de hipóteses, revisão das metas estabelecidas no primeiro contato com o problema e à seleção de estratégias e procedimentos que sejam de fato adequados a elas.

Todas essas situações são permeadas de aprendizado, implicando na geração de conhecimento conceitual e procedimental sobre a resolução de problemas. Daí a importância de conhecimento específico numa determinada área ser tão relevante para o sucesso na resolução de problemas.

Os principais objetivos cognitivos que podem ser enunciados neste processo são: reconhecimento, lembrança, explicação e exemplificação de elementos, de detalhes específicos; de teorias, de modelos, de estruturas; de procedimentos e seus critérios de uso; de estratégias e do contexto e condições do problema; verificação de categorias, teorias, modelos e estruturas; diferenciação de procedimentos e de estratégias; e geração de classificações e categorias, de princípios e generalizações, de teorias, modelos e estruturas e de critérios de uso de procedimentos.

Para o processo de *projeto de solução*, Pozo (1998) trata da Concepção de um Plano como sendo a percepção da distância entre a situação atual e a desejada [Pozo *et al.*, 1998]. É uma fase de integração das idéias geradas sobre critérios de uso de procedimentos e estratégias num todo consistente. Kapa (2002) trata essa integração como envolvendo as seguintes funções metacognitivas: conceitualização, integração, formulação e escolha heurística [Kapa, 2002].

A partir do estudo destes modelos, entende-se o *projeto de solução* como ações que envolvem organização, classificação, explicação e planejamento de procedimentos e estratégias. É preciso, portanto, compreender bem quais as estratégias escolhidas, saber como planejar sua aplicação e ordenação em relação aos procedimentos e contexto do problema.

O processo *previsão de resultados* foi retirado do Modelo Ideal de Bransford e Stein (1993), segundo o qual é preciso saber quais são as possíveis conseqüências das escolhas feitas durante o *plano de solução* [Bransford e Stein, 1993]. Precisa-se ter certeza de que as estratégias que a serem selecionadas não criarão problemas desnecessários depois de aplicadas. Quando isso é feito, possibilita-se a verificação e a comparação dos procedimentos, contextos, condições e estratégias e gerando, deste modo, um conhecimento conceitual e procedimental.

A *aplicação*, também tratada como *execução* ou *ação*, é a fase de colocar o plano em prática, uma fase de transformação do problema (o que pode gerar novos problemas). Kapa (2002) ressalta a importância de se ter consciência dos procedimentos empregados para aprendizagem e controle dos mesmos [Kapa, 2002]. Neste momento é necessário conhecimento procedimental (algorítmico) e estratégico (heurístico), e a aplicação (execução e implementação) da solução delineada.

Para o processo *gerenciamento da aplicação* Pozo (1998) acredita que “o plano proporciona apenas um roteiro geral”, afinal é preciso revisá-lo enquanto o problema é solucionado [Pozo *et al.*, 1998] e, aos poucos refinado-o até que se transforme em uma solução, decompondo suas tarefas e submetas, e sempre que necessário, é preciso abrir mão de algumas escolhas e procurar por outras mais eficientes [Bransford e Stein, 1993].

Tudo isso permite que o solucionador desenvolva a translação de informação, o uso de heurísticas, a generalização, o uso de avaliação e rapidez de solução [Kapa, 2002].

Sendo assim, entende-se que o *gerenciamento da aplicação* envolve verificação de procedimentos e de estratégias, e geração de hipóteses acerca de conhecimento conceitual e procedimental.

2.5.1.3 Avaliação

A operação *avaliação* constitui-se de dois processos fundamentais: a *crítica da solução* e a *entrega da solução*.

Pozo (1998) chama o processo de *crítica da solução* de Retrospecção, e diz que deve-se refletir sobre o sucesso do plano traçado, se foi possível ou não atingir os objetivos [Pozo *et al.*, 1998]. Além disso, ele aconselha que o processo de solução do problema esteja relacionado com o de outros, procurando diferenciar entre aproximações e divergências, sendo essencial que se continue a explorá-lo, buscando aperfeiçoar a compreensão da solução. No modelo ideal, esta fase se intitula Observar e Aprender, e é idêntica ao que Polya propõe.

Kapa (2002) se diferencia dos demais autores ao subdividir esse processo em dois (com ênfase na relação orientação-estudante): a Avaliação, em que se requer consciência do que está sendo feito e do que resta a fazer; e a Reação ao Feedback, em que é essencial saber como aproveitar a avaliação feita das ações (por um professor, computador ou leitura) de modo a aprimorar a solução e a evitar que se repitam os mesmos erros [Kapa, 2002].

Tais recomendações levam aos seguintes objetivos cognitivos: crítica de procedimentos e estratégias, interpretação, inferência, explicação, sumarização, organização e geração de conhecimento conceitual e procedimental.

O processo de *entrega da solução* foi retirado do modelo comum de Deek, Turoff e Mchugh (1999), sendo o único modelo que tratou especificamente da entrega da solução. Segundo eles depois de solucionado o problema é preciso que se apresente a solução, descrevendo sua estrutura e os processos aplicados para se alcançá-la [Deek, Turoff e Mchugh, 1999]. Acredita-se que, mesmo sendo voltado especificamente para o desenvolvimento da solução de problemas em trabalhos com programação, este processo pode ser generalizado para outras situações, sendo um exercício muito útil ao desenvolvimento da metacognição, pois envolve produção, organização, explicação e exemplificação de conhecimento conceitual e metacognitivo. Descrever o processo de solução, com os diferentes estágios envolvidos é muito importante para o solucionador ter consciência de todas as habilidades trabalhadas e das suas relações internas e externas, relativas aos conhecimentos de outros problemas.

2.6 Considerações finais

Neste capítulo verificou-se a viabilidade da RV para o desenvolvimento de ambientes de aprendizagem que estimulam a experimentação, pois ela possibilita a interação e manipulação e respostas em tempo real às ações do usuário. Esta característica é condizente com a resolução de problemas, que possui como uma de suas bases à aprendizagem pela heurística. Neste contexto optou-se pelo desenvolvimento de laboratórios virtuais (que por definição apóiam a experimentação virtual) em RV. Estes laboratórios devem ser construídos com base na teoria de objetos de aprendizagem.

Os laboratórios desenvolvidos devem fornecer apoio à aprendizagem cooperativa, pois ela é um fator essencial em processos de aprendizagem e principalmente na resolução de problemas.

A análise de diversos modelos de resolução de problemas levou ao desenvolvimento de um novo estudo desta teoria, o qual pode ser observado de modo esquemático na Figura 2.4. Este estudo permitirá a avaliação de cada processo envolvido na resolução de problemas.

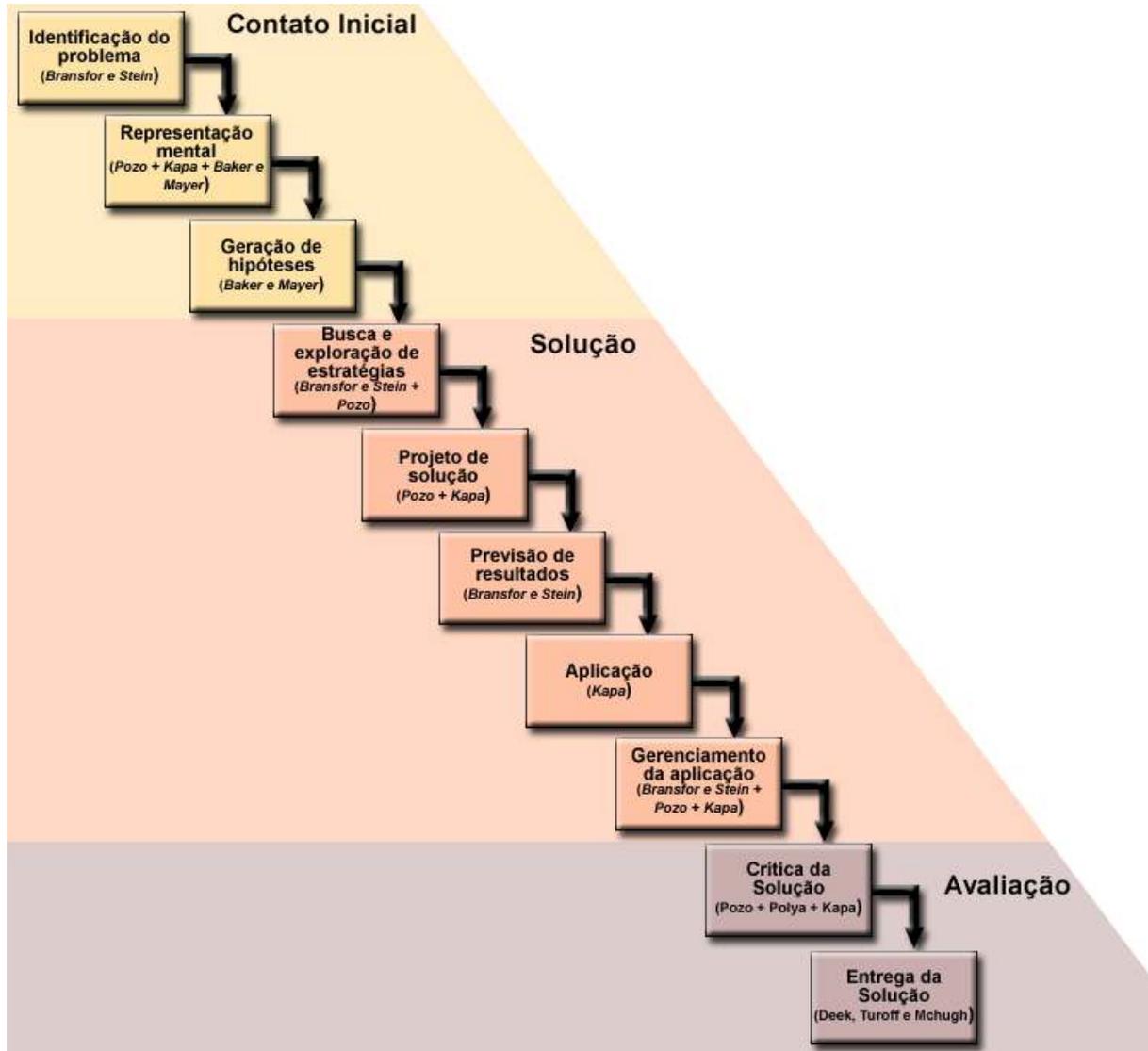


Figura 2.4: Esquema ilustrativo para o modelo de resolução de problemas do LVCF.

No próximo capítulo são apresentados os trabalhos correlatos que envolvem as áreas do conhecimento abordadas neste trabalho.

Capítulo 3

Trabalhos correlatos

3.1 Introdução

Com intuito de avaliar o estado das pesquisas concernentes ao presente trabalho, foram analisadas ferramentas computacionais voltadas para o ensino, que se baseiam na resolução de problemas, e/ou aprendizagem cooperativa e/ou RV, sendo verificadas suas vantagens e limitações no processo de ensino e aprendizagem.

3.2 Andes

O projeto Andes foi desenvolvido pela *University of Pittsburgh* em parceria com a *United States Naval Academy*, com apoio do *Office of Naval Research* por meio do Programa de Ciência Cognitiva (*Cognitive Science Program*). Este projeto estuda o uso de metodologias de resolução de problemas aplicadas ao ensino de Física. Para tanto, uma de suas principais metas é modificar o modo com que os alunos efetuam suas tarefas de casa; pretende-se substituir o método tradicional de resolver exercícios usando “lápiz e papel” pelo método no qual o aluno utiliza o computador para tal fim. Cujo objetivo era verificar se esta mudança aumenta o aprendizado do estudante [Vanlehn *et al.*, 2005].

Inicialmente a interface do Andes, ilustrada na Figura 3.1, fornece ao usuário um problema de Física e o usuário deve: construir um modelo matemático, usando vetores, para representar a situação problema; definir corretamente as variáveis; e inserir as equações adequadas; para chegar a sua solução. Estas atividades podem ser realizadas através de uma caixa de ferramentas, composta de um conjunto de botões com diferentes funcionalidades, e campos de texto.

The screenshot shows the ANDES Physics Workbench interface. The main window displays a physics problem: "A 2000-kg car in neutral at the top of a 20.0 deg inclined driveway 20.0 m long slips its parking brake and rolls down. If we ignore friction and drag, what would the magnitude of the velocity of the car be when it hits the garage door?" Below the text is a diagram of a car on a 20-degree incline and a free-body diagram of the car with forces labeled 'c' and 'Fw'. An "Answer:" field is present but empty.

To the right, a "Variables" table lists the following:

Name	Definition
T0	car starts rolling
T1	car hits garage door
x	axis
mc	mass of car
d	magnitude of the Displacement of car at time T0 to T1
Fw	magnitude of the Weight Force on car at time T0 to T1 du

Below the variables table is a list of equations:

- $mc = 2000 \text{ kg}$
- $d = 20.0 \text{ m}$
- $Fw_y = mc * g$
-
-
-
-
-
-
-
-

At the bottom, a text box contains a hint: "T: Now that you have stated all of the given information, you should start on the major principles. What quantity is the problem seeking? S: The magnitude of the instantaneous Velocity of car at time T1. T: Yep. What is the first principle application that you would like to work on? Hint: this principle application will usually be one that mentions the sought quantity explicitly. Therefore it's equation may contain the sought quantity that the problem seeks."

Figura 3.1: Interface do Andes.
Fonte: [Vanlehn et al., 2005]

O sistema oferece alguns tipos de ajuda ao usuário, como, por exemplo: uma calculadora interna; mensagens de alerta, acionadas quando algum campo de texto é esquecido em branco; o recurso de mudança da cor de textos para vermelho, ativado quando são inseridos dados errôneos; e sugestões e dicas sobre passos posteriores da sua atividade.

3.3 Ejournal

O eJournal (The Electronic Laboratory Journal) é um projeto, norteado pela *Manchester Metropolitan University*, que visa o desenvolvimento de um ambiente colaborativo e cooperativo voltado para o ensino por meio da experimentação baseada em Web. De acordo com o projeto eJornal, foram desenvolvidas varias ferramentas para apoiar a colaboração entre estudantes. Estas ferramentas incluem principalmente artifícios para compartilhar documentos e habilitar discussões.

O eJournal propõe-se a aumentar o diário de laboratório tradicional, proporcionando a um grupo de estudantes ferramentas baseadas em Web que permitam coleccionar, anotar,

organizar e compartilhar dados necessários para completar suas tarefas de experimentação. Os objetos inseridos podem ser compostos de fragmentos de qualquer formato, como texto, imagens, gráficos, manuscritos, medida anota ou resultados experimentais. O eJournal também controla a submissão de resultados à pedagogos o que facilita a supervisão remota. Atualmente o eJournal está sendo avaliado *pela School of Engineering, da École Polytechnique Fédérale de Lausanne (EPFL)*, no âmbito das atividades de experimentação que focalizam em manipulação remota e simulações baseadas em *Web* [Fakas, Nguyen e Gillet, 2005]. A Figura 3.2 apresenta a interface do eJornal.



Figura 3.2: Interface do eJornal.
Fonte: [Fakas, Nguyen e Gillet, 2005]

O eJornal permite a experimentação remota, de conteúdos como Automação de Controles, Biomecânica e Mecânica de Fluidos, por meio de um ambiente gráfico denominado *Cockpit*, sua interface encontra-se ilustrada na Figura 3.3. Após realizar experiências neste ambiente o usuário pode armazenar os resultados do seu aprendizado e submetê-los aos pedagogos, conforme descrito acima.



Figura 3.3: Interface Cockpit.
Fonte: [Fakas, Nguyen e Gillet, 2005]

3.4 ScienceSpace

O projeto *ScienceSpace*, desenvolvido pela *George Mason University*, juntamente com a *University of Houston*, com o apoio da *NASA's Johnson Space Center*, propõe-se a explorar as contribuições da RV para o processo de ensino e aprendizagem na área de ciências. Este sistema é composto por uma coletânea de ambientes virtuais imersivos, que foram projetados para auxiliar os estudantes na aprendizagem de conceitos científicos, essencialmente relacionados à Física e Química. O projeto possui três ambientes gráficos principais: o *NewtonWorld*, um ambiente que permite investigar a cinemática e dinâmica de movimentos unidimensionais, este ambiente encontra-se ilustrado na Figura 3.4 (a); o *MaxwellWorld*, que possibilita ao usuário explorar a eletrostática e conceitos envolvidos na Lei de Gauss, ilustrado na Figura 3.4 (b); e o *PaulingWorld*, que habilita o estudo de estruturas moleculares por meio de uma variedade de representações virtuais das mesmas, como elucida a (c) [Dede *et al.*, 1999].

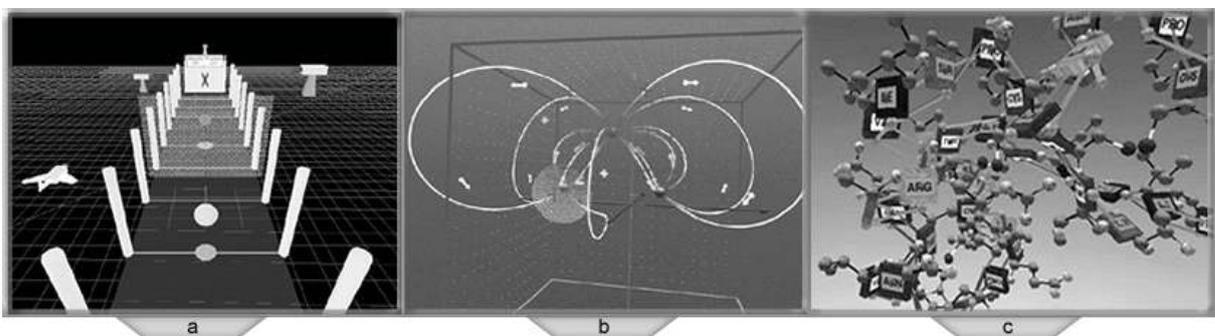


Figura 3.4: Ambientes gráficos do ScienceSpace.
Fonte: [Dede *et al.*, 1999]

A modelagem do sistema é composta, fundamentalmente, por objetos tridimensionais e representações qualitativas, construídos a partir de geometrias poligonais, combinando formas básicas, texturas e cores. O sistema permite atribuir comportamentos a estes objetos,

por meio do uso de um *software* desenvolvido pela *Nasa* que, possibilita definir ambientes virtuais e conectá-los a algumas simulações físicas. A interação com o ambiente virtual ocorre, essencialmente, por meio de uma luva, que permite acessar menus virtuais, como exemplifica a Figura 3.5.

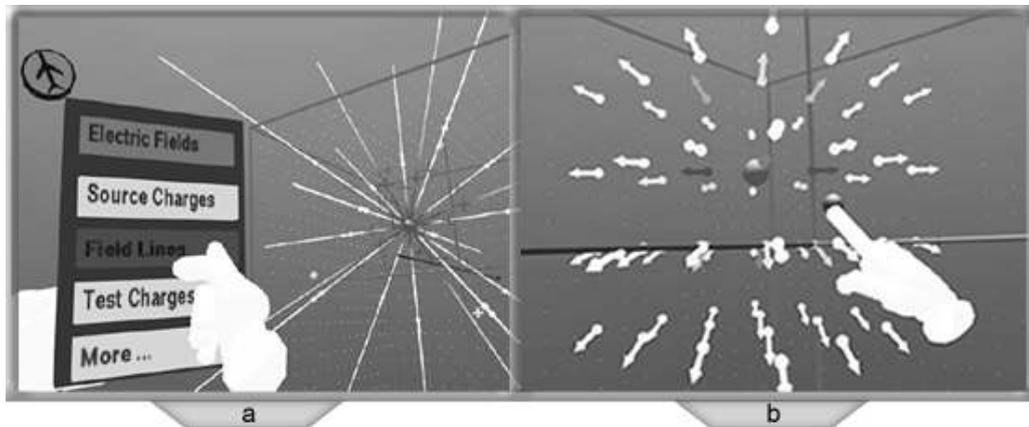


Figura 3.5: Interação no ScienceSpace.
Fonte: [Dede *et al.*, 1999]

O *ScienceSpace* não permite que diferentes estudantes, em pontos remotos ou em grupo, possam, simultaneamente, partilhar os mesmos ambientes virtuais, deste modo, o sistema não enfatiza a aprendizagem cooperativa em um grupo aprendizes, uma vez que trabalha apenas localmente [Cardoso, 2000]. No entanto, em função da larga experiência no desenvolvimento e nas diversas realimentações obtidas pelos autores ao longo dos anos, um grande conjunto de melhorias tem tornado o aprendizado, através dos ambientes virtuais, mais efetivo, permitindo melhorias na motivação dos estudantes com respeito ao interesse pelos tópicos abordados [Dede *et al.*, 1999].

3.5 WebTop

Com o intuito de ensinar Óptica, a *University of State Mississippi* criou a ferramenta WebTOP, totalmente desenvolvida em VRML e Java. O sistema é composto por dezesseis módulos diferentes, envolvendo ondas, ótica geométrica, reflexão e refração, polarização, interferência, difração, distribuídos em oito áreas da Óptica. Cada módulo possui simulações computacionais interativas, animadas e em 3D para apresentação de fenômenos físicos, sendo utilizados *applets* Java para controlar interações do usuário com os ambientes de experimento [Mzoughi *et al.*, 2005].

Juntamente com cada simulação, o WebTop propõe uma seção sobre a teoria envolvida no fenômeno em estudo, uma seção sobre como operar a simulação, e uma seção com exercícios sugeridos.

O WebTop foi desenvolvido para auxiliar professores e alunos, sendo seus experimentos voltados para estudantes universitários nivelados do curso de Física ou áreas afins. Durante quatro anos, a universidade utilizou o WebTop para ensinar Óptica em suas instalações, atualmente ele está sendo usado nacional e internacionalmente em inúmeras outras universidades. A Figura 3.6 apresenta uma tela típica de um experimento do *software* WebTop.

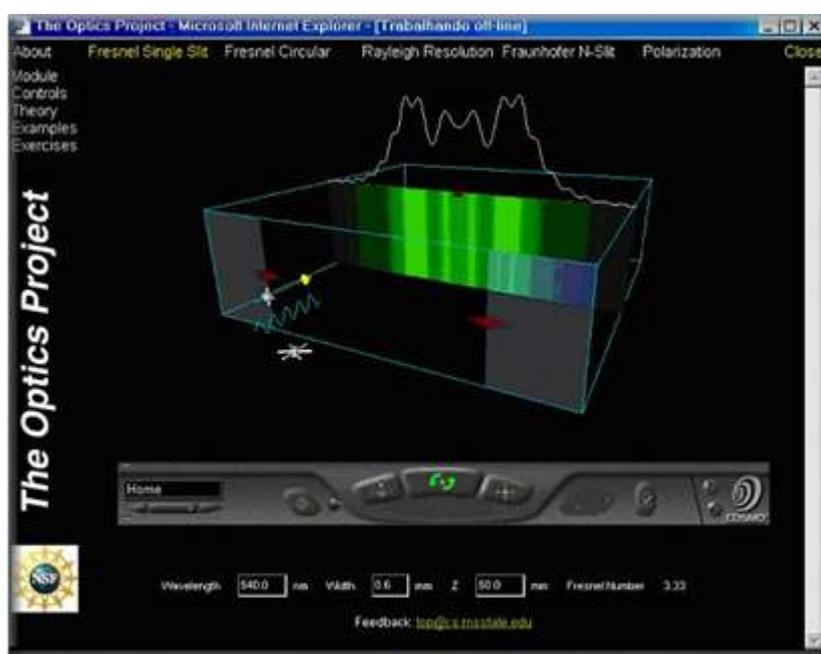


Figura 3.6: Uma experiência do *software* WebTop.
Fonte: [Webtop, 2009]

3.6 Supporting Distance Learners for Collaborative Problem Solving

Este projeto, *Supporting Distance Learners For Collaborative Problem Solving* (SDLFCPS), foi desenvolvido pela *Universidad Nacional de Educación a Distancia* – UNED em parceria com a *Universitat Politècnica de Catalunya* – UPC. A UNED trabalha com ensino superior a distância por toda a Espanha e já possui instalações físicas em quase todas as suas capitais, a UPC está situada em Barcelona capital da Catalunha – Espanha.

Esta pesquisa focaliza, essencialmente, o estudo de métodos de resolução de problemas por meio de sistemas colaborativos, em particular, aplicados ao desenvolvimento

de *software*. Para tanto, a arquitetura e construção do sistema foram baseadas em observações e discussões de estudantes da UNED e UPC, assim como, em suas anotações usadas para descrever o domínio de objetos de aprendizagem, e descrições de problemas e instâncias necessárias para a sua solução [Verdejo, Barros e Abad, 1998].

O sistema é, fundamentalmente, elaborado com base em três módulos, o *activity model*, o *dialogue model* e o *domain model*. O *activity model* tem como objetivo informar características sobre a atividade proposta, como, por exemplo, o seu nome, a sua descrição, se ela será resolvida de forma individual ou colaborativa, e os participantes desta atividade. Este módulo possibilita, ainda, a análise de um histórico, que organiza os dados armazenados por datas. Estes dados podem ser, por exemplo, as ferramentas e suas devidas descrições, usadas pelo aluno na resolução da atividade.

O *dialogue model* visa promover o debate entre os estudantes em tarefas e situações específicas, sendo que, a forma e a estrutura deste debate, assim como, a sua relação com os demais módulos podem variar de acordo com a tarefa.

E por fim, o *domain model* propõe-se a prover uma estrutura que contemple a resolução de problemas e possibilitar diferentes tipos de abstrações para os mais variados usuários. Para tanto, o sistema possui uma área de edição que permite a escrita da solução do problema. Esta solução ocorre por meio de um conjunto de objetos primitivos aos quais podem ser aplicadas algumas operações pré-definidas, e uma linguagem pré-estipulada que visa eliminar a ambigüidade e a padronização dos símbolos utilizados no sistema [Verdejo, Barros e Abad, 1998]. A interface do sistema, que contempla estes módulos, encontra-se ilustrada na Figura 3.7.

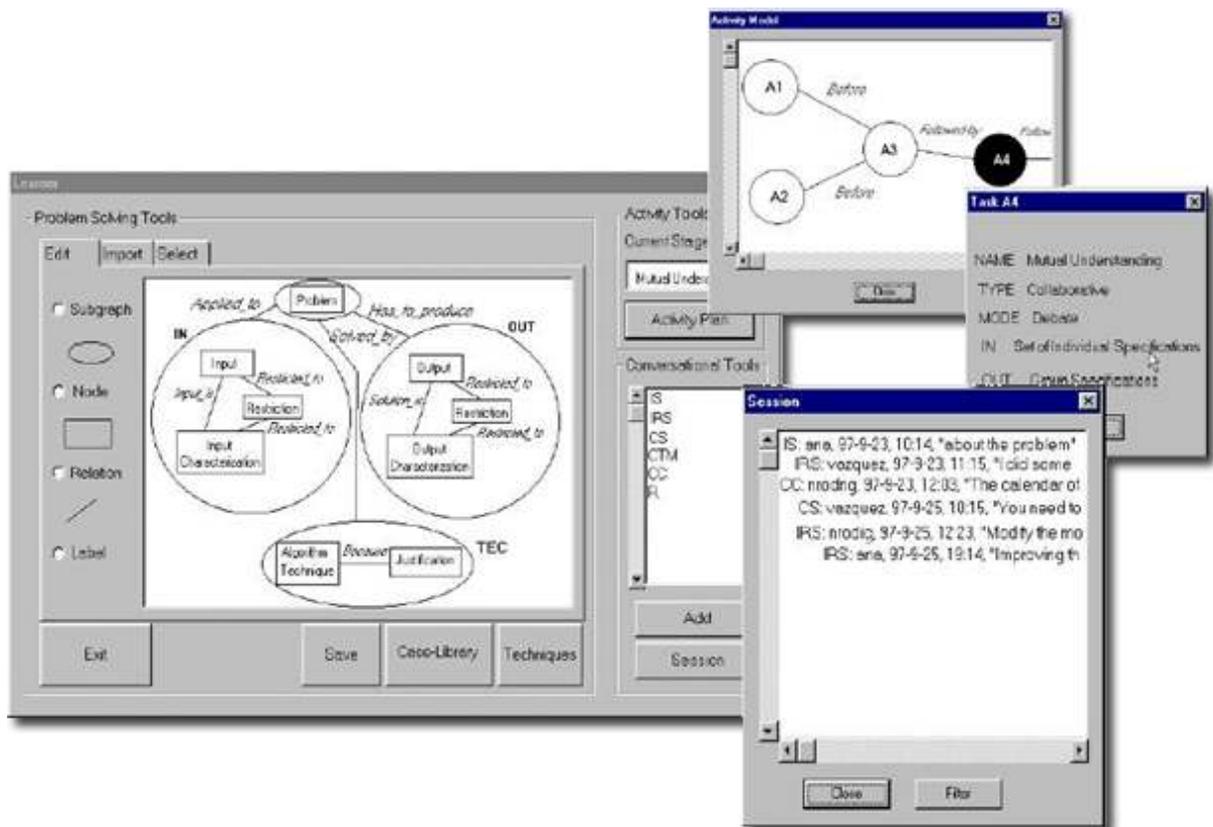


Figura 3.7: Interface do Supporting Distance Learners For Collaborative Problem Solving.
Fonte: [Verdejo, Barros e Abad, 1998]

De acordo com [Verdejo, Barros e Abad, 1998] este é um sistema protótipo que passa por melhorias e aperfeiçoamentos, com base em observações, análises e discussões sobre as atividades dos alunos. Este sistema foi aplicado por dois anos consecutivos no curso de Algoritmos na universidade *Technical University of Spain* especificamente para tal fim.

3.7 The European Challenge

O *The European Challenge* (TEC) é um projeto acadêmico desenvolvido pela *Kingston University* que visa à aprendizagem colaborativa por meio de uma situação problema. Este projeto envolve um conjunto de instituições, como a RICS, FIGO, FIABCI e NAR, e um conjunto de empresas comerciais, como a *Estates Gazette* e CBRE, e reúne ainda tutores e estudantes de diversas universidades da Europa. Este trabalho destina-se a alunos de graduação e pós-graduação, que investigam tomadas de decisão sobre a alocação estratégica de bens imóveis, neste âmbito, o projeto aborda um problema contemporâneo, a realocação de um edifício sede na Europa de uma multinacional dos EUA. Esta simulação possui como objetivo estimular e conscientizar os estudantes sobre a importância do local de designio da propriedade, considerando orçamentos, planejamentos, especificações, a interação com a

imprensa, relacionamento com cliente, etc. O projeto utiliza um ambiente de aprendizagem virtual para administrar e estimular grupos de estudantes que trabalham em equipes, focalizando a pesquisa e a publicação de suas experiências. Este ambiente, denominado *Blackboard VLE* e oferecido pela CEBE – *Center for Education in the Built Environment*, [Cebe, 2008], age como um canal formal de comunicação e repositório de informações.

Por meio desta ferramenta cada grupo de estudante possui autonomia para projetar novos instrumentos de comunicação ou outras ferramentas de aprendizagem que satisfaçam as necessidades da equipe [Nunnington, Lecturer e Eilander, 2005].

3.8 ARCA

O projeto ARCA, acrônimo de Ambiente de Realidade Virtual Cooperativo de Aprendizagem, está sendo desenvolvido por três áreas de ensino da Universidade Federal do Rio Grande do Sul – UFRGS: Psicologia, Educação e Informática, coordenado pelo Programa de Pós-Graduação em Informática na Educação [Ufrgs, 2008]. Este projeto propõe-se a desenvolver um ambiente de ensino e aprendizagem que, apoiado pela Internet, possa atuar como instrumento no auxílio a uma prática pedagógica diferenciada. O seu ambiente visa criar condições para uma aprendizagem significativa por meio de um ambiente desenvolvido em RV que permite a cooperação. Os estudantes podem, ainda, utilizar telepresença via avatar, esta ferramenta possui como objetivo permitir aos estudantes e professores realizar uma comunicação por meio de um personagem, o avatar.

De acordo com Tarouco (2000) outro aspecto abordado neste projeto é a chamada Lógica Operatória, um modelo desenvolvido por Jean Piaget para o estudo das relações entre os níveis de explicação que o sujeito, em sua vivência cognitiva, estabelece sobre os fenômenos do mundo físico e social em que vive. Este projeto espera como resultados entender como e por que estender um modelo de análise operatória advindo da teoria piagetiana para analisar um ambiente de RV, e identificar como aparecem as relações de egocentrismo, conformismo, coação e cooperação (mecanismos sócio-cognitivos) quando o sujeito interage num ambiente de RV [Tarouco *et al.*, 2000]. A Figura 3.8 apresenta exemplos do ambiente virtual no qual ocorre à cooperação e comunicação entre os usuários por meio de avatares.



Figura 3.8: Interface do ambiente colaborativo RV.
 Fonte: [Ufrgs, 2008]

Por meio desta interface o usuário pode, ainda, acessar ambientes de experimentação desenvolvidos em *Flash*, por meio da linguagem *Action Scripts*, e realizar experiências de diferentes conteúdos, como conservação da energia, eletrização, eletrodinâmica, eletromagnetismo e movimentos. A Figura 3.9 ilustra um exemplo destas experiências.

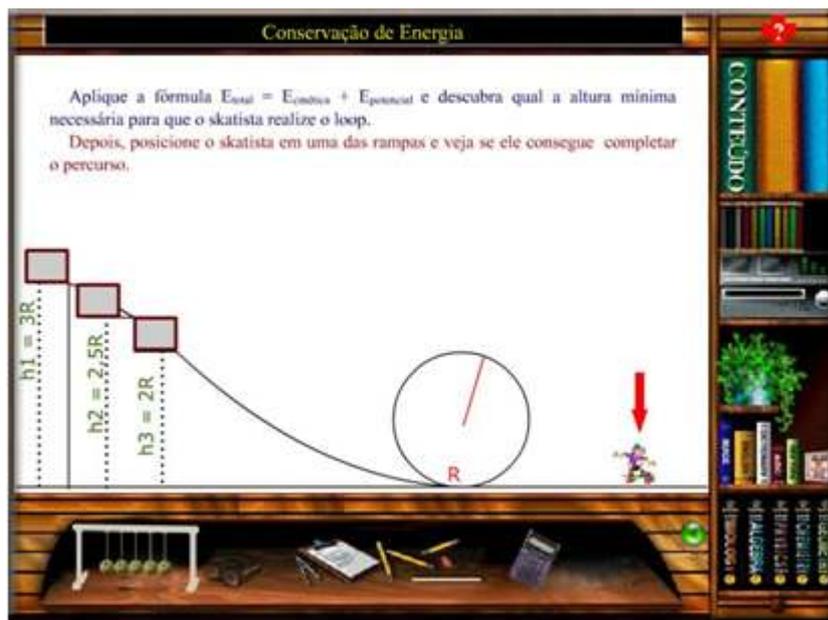


Figura 3.9: Ambiente de experimentação do projeto ARCA.
 Fonte: [Ufrgs, 2008]

No entanto, é importante ressaltar que este ambiente para experimentos de Física não é cooperativo, pois pode ser usado apenas de maneira local. Porém, este ambiente não é o foco de estudo do projeto ARCA, e sim as discussões resultantes dele, realizadas no ambiente colaborativo ilustrado na Figura 3.8.

3.9 DEVRL

A sigla DEVRL é o acrônimo de *Distributed Extensible Virtual Reality Laboratory*. Este projeto, desenvolvido pela *University of Lancaster*, visa possibilitar à alunos de diferentes localidades do Reino Unido trabalhar em um mesmo espaço virtual e realizar tarefas comuns [Lancaster_University, 2008]. Duas aplicações teste foram desenvolvidas: o *Nottingham*, que permite ao usuário navegar localmente por representações 3D das informações contidas no espaço virtual, assim como exemplifica a Figura 3.10 (a); e o *Lancaster*, que possibilita a criação de uma sala de jogos virtual. Esta sala tem como objetivo fornecer aos usuários um sistema distribuído que possibilita aplicações em RV baseadas em colaboração, e que permitam a experimentação das propriedades físicas por meio de simulações de jogos [Mel *et al.*, 1996], como ilustra a Figura 3.10 (b). Este projeto recebe patrocínio de diversas empresas como a BT, ARRL, BAe e DIVISION.

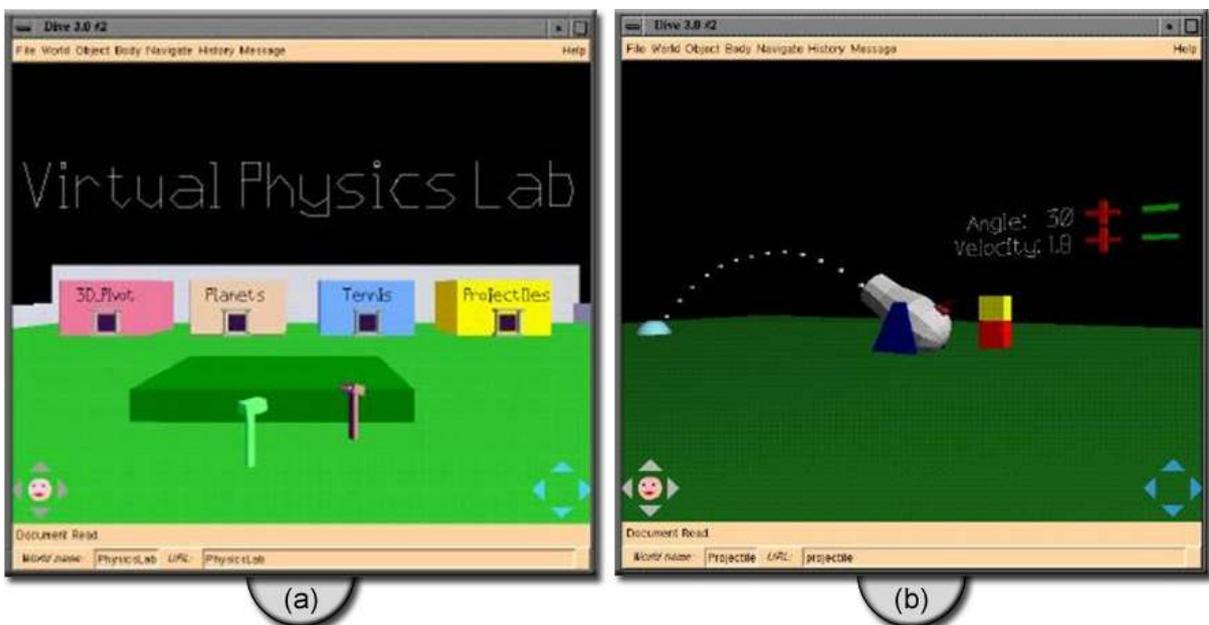


Figura 3.10: Interfaces do DEVRL.
Fonte: [Lancaster_University, 2008]

É importante destacar que o foco da pesquisa do projeto DEVRL é o estudo de experiências físicas por meio de jogos virtuais colaborativos. Neste sentido, não faz parte do

seu contexto usar e analisar explicitamente a resolução de problemas, seus modelos e etapas. Apesar de que seja possível o uso de um problema para motivar o experimento físico.

3.10 NICE

O projeto NICE, acrônimo de *Narrative Immersive Constructionist/Collaborative Environments*, foi desenvolvido pela *University of Illinois at Chicago* e tem como intuito aplicar RV para a criação de ambientes educacionais para crianças. Ele foi projetado com base em teorias construtivistas, proporcionando aos seus usuários uma motivação, por meio de uma narrativa, para construírem mundos virtuais colaborativamente [Roussos *et al.*, 1997]. Este processo encontra-se ilustrado na Figura 3.11.



Figura 3.11: Crianças interagindo com o NICE.
Fonte: [Roussos *et al.*, 1997]

A concepção do NICE baseia-se diretamente em dois sistemas o CALVIN e o *Graphical StoryWriter* [Roussos *et al.*, 1997]. O CALVIN (*Collaborative Architectural Layout Via Immersive Navigation*) é uma rede de ambientes colaborativos para a concepção de espaços arquitetônicos. O *Graphical Storywriter*, por sua vez, permite que

crianças compartilhem trabalhos com o intuito de criarem histórias estruturalmente completas. O *software* do NICE engloba as características destes dois sistemas, sendo executado em *Caves* para RV e suportando multi-usuários.

3.11 Construct3D

O Construct3D foi desenvolvido pelo *Institute of Software Technology and Interactive Systems* na *Vienna University of Technology* e pelo *Institute for Computer Graphics and Vision* na *Graz University of Technology*. Este projeto visa à criação de uma ferramenta que possibilite a construção geométrica tridimensional, sendo que, ela está direcionada especificamente para o ensino de matemática e geometria [Kaufmann e Schmalstieg, 2006]. O seu desenvolvimento baseia-se no sistema móvel de Realidade Aumentada (RA) colaborativa denominado *Studierstube*. A Figura 3.12 apresenta usuários utilizando o Construct3D.

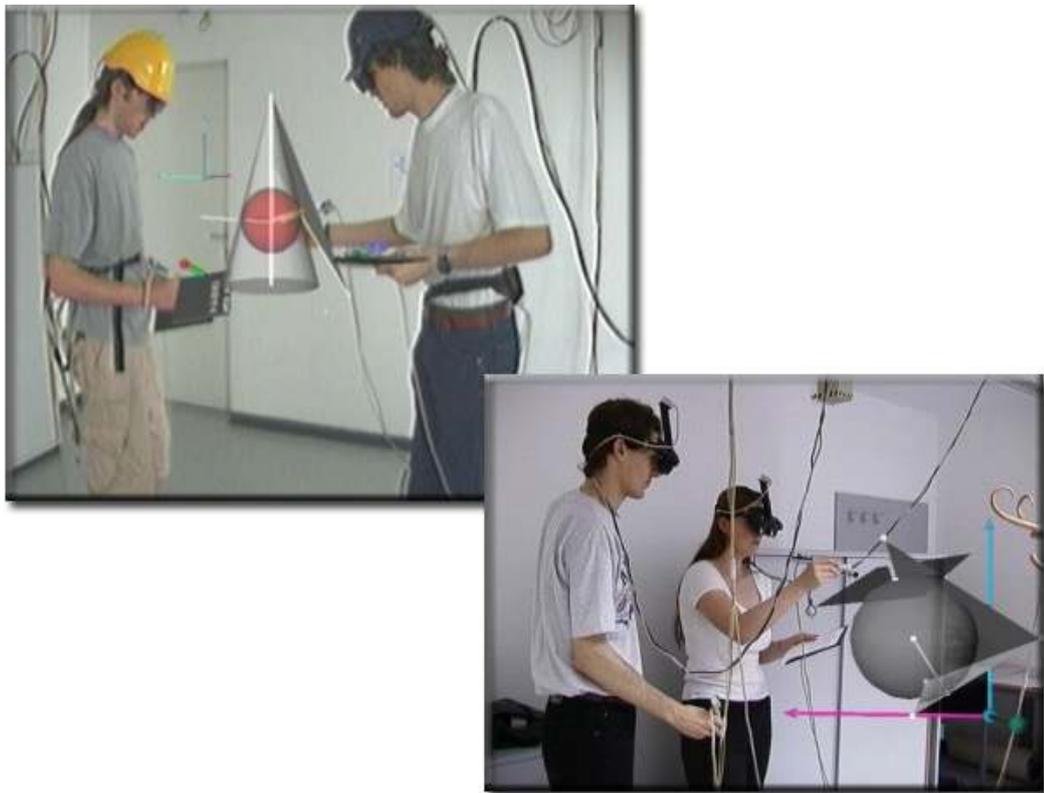


Figura 3.12: O uso do Construct3D.
Fonte: [Construct3d, 2009]

Atualmente o Construct3D está sendo usado em salas de aula no ensino superior. Os seus pesquisadores utilizam métodos de aprendizagem flexíveis, que visam se adaptar ao contexto do usuário, e *hardwares* híbridos e equipamentos especializados de RA, como por exemplo, capacetes HMDs, que permitem a sua aplicação junto aos alunos. Todo este

processo tem como intuito melhorar o desempenho do Construct3D em avaliações futuras sobre o processo de ensino e aprendizagem [Kaufmann e Schmalstieg, 2006].

3.12 Considerações finais

Os projetos descritos nas seções anteriores possuem um reconhecimento significativo junto ao meio acadêmico e comercial. Deste modo, o estudo destes projetos sugere uma representação simplificada do estado em que se encontram as pesquisas, em termos tecnológicos, nas áreas de resolução de problemas, aprendizagem cooperativa e RV. É importante ressaltar que estas áreas de pesquisa compõem o foco de estudo deste trabalho. A Tabela 3.1 apresenta, resumidamente, a distribuição destes projetos em suas devidas linhas de pesquisa.

Tabela 3.1: Projetos e seus focos de pesquisa.

		FOCOS DE PESQUISA		
		Resolução de Problemas	Aprendizagem Cooperativa	Realidade Virtual
P R O J E T O S	Andes	✓		
	eJornal		✓	
	ScienceSpace			✓
	WebTop			✓
	SDLFCPS	✓	✓	
	TEC	✓	✓	
	ARCA		✓	✓
	DEVRL		✓	✓
	NICE		✓	✓
	Construct3D		✓	✓

Por meio da análise da Tabela 3.1, pode-se observar que existem trabalhos advindos de pesquisas focalizadas, especificamente, em apenas uma das áreas de conhecimento tratadas neste projeto ou em conjuntos resultantes da interseção das mesmas. Por exemplo, o Andes está direcionado, principalmente, para a resolução de problemas; já o eJornal, pesquisa primordialmente a aprendizagem cooperativa; e o ScienceSpace e WebTop visam,

essencialmente, examinar as contribuições da RV para os processos de ensino e aprendizagem. Existem, ainda, pesquisas que se focalizam, principalmente, na interseção de duas destas áreas, como, por exemplo, o *Supporting Distance Learners For Collaborative Problem Solving*, o *The European Challenge*, que se baseiam na resolução de problemas junto à aprendizagem cooperativa; ou os projetos ARCA, DEVRL, NICE e Construct3D que se situam na interseção das áreas de aprendizagem cooperativa e RV.

É importante ressaltar que entende-se neste trabalho que um projeto focaliza a resolução de problemas quando ele pesquisa explicitamente os conceitos e os modelos desta teoria, como é o caso, por exemplo, do projeto Andes. Porém, deve-se admitir a existência de projetos que utilizam a resolução de problemas apenas para motivar alguma parte do ensino, como por exemplo, pode-se aplicar um problema em que um aluno deve usar um ambiente virtual para resolvê-lo, no entanto, não necessariamente deve-se estar interessado em estudar os conceitos, os modelos e os processos realizados pelo aluno para encontrar a solução.

Pode-se perceber ainda na Tabela 3.1 que as áreas de resolução de problema, aprendizagem cooperativa e RV estão sendo estudadas atualmente, seja de maneira individual ou interseccional. No entanto, um fator saliente, neste contexto, é carência de projetos que direcionem suas pesquisas para a interseção das áreas de resolução de problemas e RV. Observa-se ainda, a inexistência de pesquisas voltadas para a interseção entre as três áreas discutidas neste trabalho.

No entanto, é imprescindível destacar que todos os trabalhos estudados neste capítulo são projetos de qualidade, e o fato deles não pesquisarem a interseção das três áreas não diminui de forma alguma os seus valores para a comunidade científica. Os projetos analisados focalizaram em estudos que serviram de base e motivação para o presente trabalho. Entende-se ainda que um trabalho de qualidade, independentemente do seu foco, contribui para o avanço da ciência, e o fato de pesquisas tomarem rumos diferentes com certeza aumenta o campo de conhecimento alcançado e a velocidade do crescimento científico.

Com base no estudo do estado da arte, na fundamentação pedagógica, na motivação e nos objetivos desta pesquisa, foi projetado um sistema computacional, denominado LVCF. O projeto desenvolvido encontra-se documentado neste trabalho no próximo capítulo.

Capítulo 4

Análise e especificação do sistema

4.1 Introdução

A partir do estudo descrito nos capítulos anteriores e com base em fundamentos da Engenharia de Software elaborou-se o projeto do LVCF. Foram documentados alguns pontos que são essência para a compreensão do funcionamento do sistema, como, por exemplo, a análise de requisitos e o desenvolvimento de alguns diagramas comportamentais UML – *Unified Modeling Language* e das arquiteturas do sistema.

4.2 Especificações do sistema

Nesta seção é apresentada uma visão do sistema, mostrando de modo geral seus principais componentes, suas funcionalidades e como são conectados. Para tanto, são utilizados alguns diagramas UML que permitem a melhor compreensão desta perspectiva. Posteriormente, são abordados os componentes com maior complexidade e de maior importância para este projeto.

4.2.1 Requisitos do sistema

Requisito pode ser definido, simplesmente, como "algo que um cliente necessita", entretanto, do ponto de vista de um desenvolvedor, requisito pode também ser definido como "algo que necessita ser projetado" [Macaulay, 1996].

Os requisitos caracterizam-se sob duas formas fundamentais, os funcionais e os não-funcionais [Sommerville e Sawyer, 1997]:

- por requisitos funcionais, entendem-se os fenômenos ambientais referentes ao negócio e domínio da aplicação que se queira conhecer e estudar, ou seja, o que fazer. São escritos do ponto de vista do cliente, sendo assim, normalmente expressos em linguagem natural, diagrama informal ou usando alguma notação que é apropriada para o entendimento do problema;

- por requisitos não-funcionais, entendem-se as especificações que sugerem como melhor adequar a solução do problema para responder ao cliente, como por exemplo, confiabilidade, tempo de resposta, espaço de armazenamento, redução de custo, velocidade da máquina, etc. Elas são especificações que podem ser expressas como um modelo abstrato do sistema. Pode ser um modelo matemático ou baseado em notações gráficas, tais como diagrama de fluxo de dados, hierarquia de classes de objetos e, de preferência em linguagens de representação formal.

Neste contexto, os principais requisitos do LVCF são:

- requisitos funcionais:
 - serão necessários login e senha para acessar o sistema;
 - os usuários poderão criar cadastros e definir o tipo de usuário (aluno ou professor);
 - o administrador poderá ativar contas de professores;
 - os professores poderão ativar contas de alunos;
 - os professores poderão excluir contas de alunos;
 - os usuários poderão modificar um cadastro já ativado;
 - o professor poderá cadastrar, alterar, e apagar problemas para alunos;
 - os alunos poderão acessar problemas pré-cadastrados;
 - os alunos poderão cadastrar, alterar, e apagar as soluções, e suas devidas considerações sobre os problemas pré-cadastrados;
 - os professores poderão acessar as soluções e considerações dos alunos sobre os problemas cadastrados;
 - o software deve permitir aos alunos carregar, juntamente com um problema, um ambiente virtual com objetos que permitam a resolução deste;
 - os ambientes virtuais devem permitir aos alunos navegar, interagir e manipular intencionalmente os objetos;

- os alunos poderão resolver os problemas propostos, por meio dos ambientes virtuais, cooperativamente e sincronamente;
- os alunos poderão trocar informações entre eles por meio de um Chat;
- o usuário deve ter acesso a ajuda relativa ao sistema;
- requisitos não-funcionais:
 - deverá operar em alta velocidade de processamento;
 - o sistema deverá usar uma rede de computadores local para efetuar troca de dado;
 - a rede de computadores a qual o sistema operará deve ser de alta velocidade.

4.2.2 Diagrama de casos de uso do sistema

Formalmente, um caso de uso pode ser definido como um tipo de classificador representando uma unidade funcional coerente provida pelo sistema, subsistema, ou classe manifestada por seqüências de mensagens intercambiáveis entre os sistemas e um ou mais atores. Um ator é um humano ou entidade máquina que interage com o sistema para executar um significativo trabalho. Ou seja, em suma, o caso de uso representa uma unidade discreta da interação entre um usuário (humano ou máquina) e o sistema.

Neste contexto, foram desenvolvidos diagramas de casos de uso do LVCF. A Figura 4.1 mostra o diagrama principal do sistema.

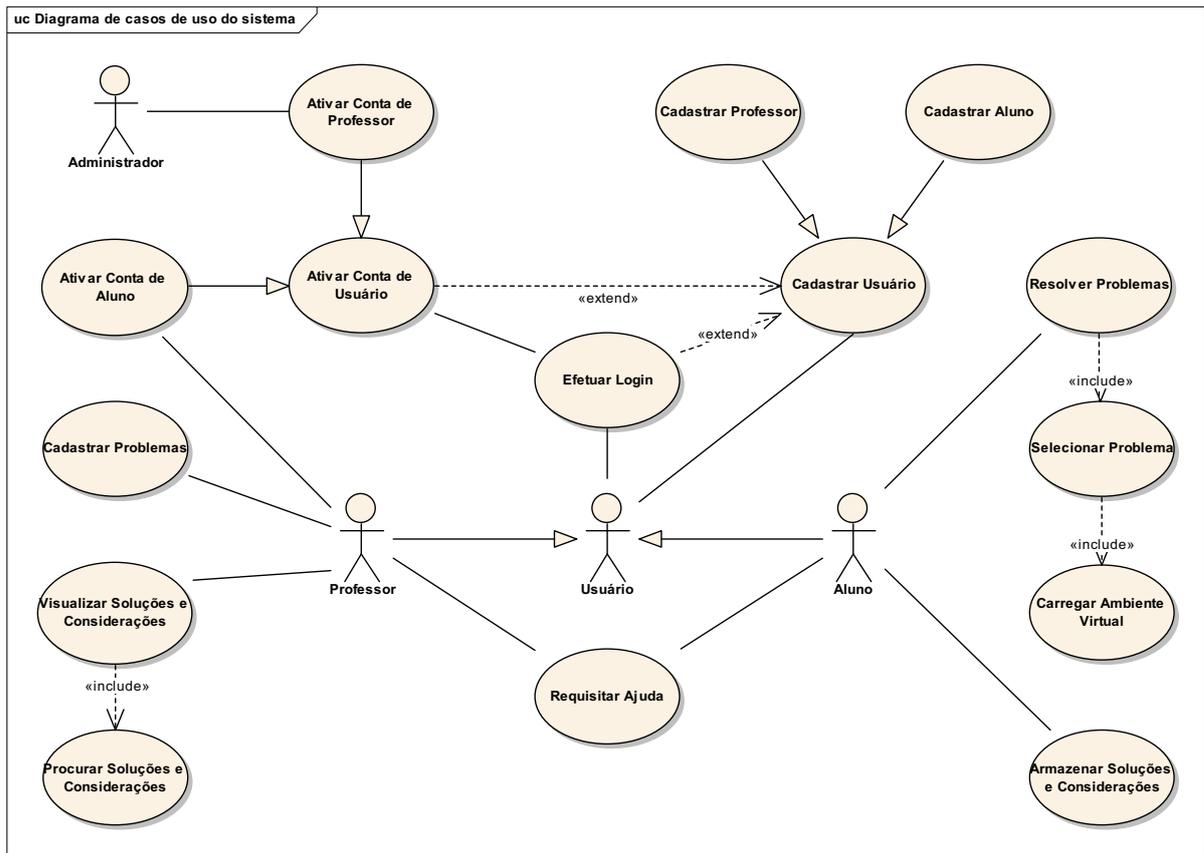


Figura 4.1: Diagrama de casos de uso do sistema.

4.2.3 A arquitetura geral do sistema

Bass, Clements e Kazman (2003) definem a arquitetura de um programa ou sistema computacional como a estrutura ou estruturas do sistema que abrangem os componentes de *software*, as propriedades externamente visíveis desses componentes e as relações entre eles [Bass, Clements e Kazman, 2003]. Um componente de *software* pode ser algo tão simples quanto um módulo de programa ou uma classe orientada a objetos, mas também pode ser ampliado para incluir bancos de dados e *middleware*, que permite a configuração de uma rede de clientes e servidores.

A arquitetura do sistema desenvolvida para o LVCF visou atender os requisitos propostos para o sistema. Ela tem como objetivo central oferecer suporte para professores e alunos no processo de ensino e aprendizagem, o qual se baseia em teorias da aprendizagem cooperativa e resolução de problemas, aplicadas a sistemas distribuídos de RV.

Para tanto, deve-se possibilitar ao professor gerenciar e acompanhar as atividades do aluno, permitindo que ele proponha atividades experimentais (problemas) e tenha retornos quanto à efetividade das mesmas. Este processo permite ao professor verificar como o aluno

está trabalhando com as atividades sugeridas e como está ocorrendo sua aprendizagem mediante elas. Deste modo, o professor terá uma noção avaliativa do processo de aprendizagem do aluno, o que facilitará a sugestão de novas atividades e o direcionamento do estudo do aluno.

O aluno, por sua vez, deve ter acesso a laboratórios virtuais, desenvolvidos por meio de RV, nos quais poderá resolver problemas com apoio de experimentos virtuais. O aluno deve, ainda, enviar a solução de sua atividade e suas dúvidas no decorrer do processo de resolução, para análise dos professores.

A Figura 4.2 ilustra o diagrama da arquitetura do sistema.

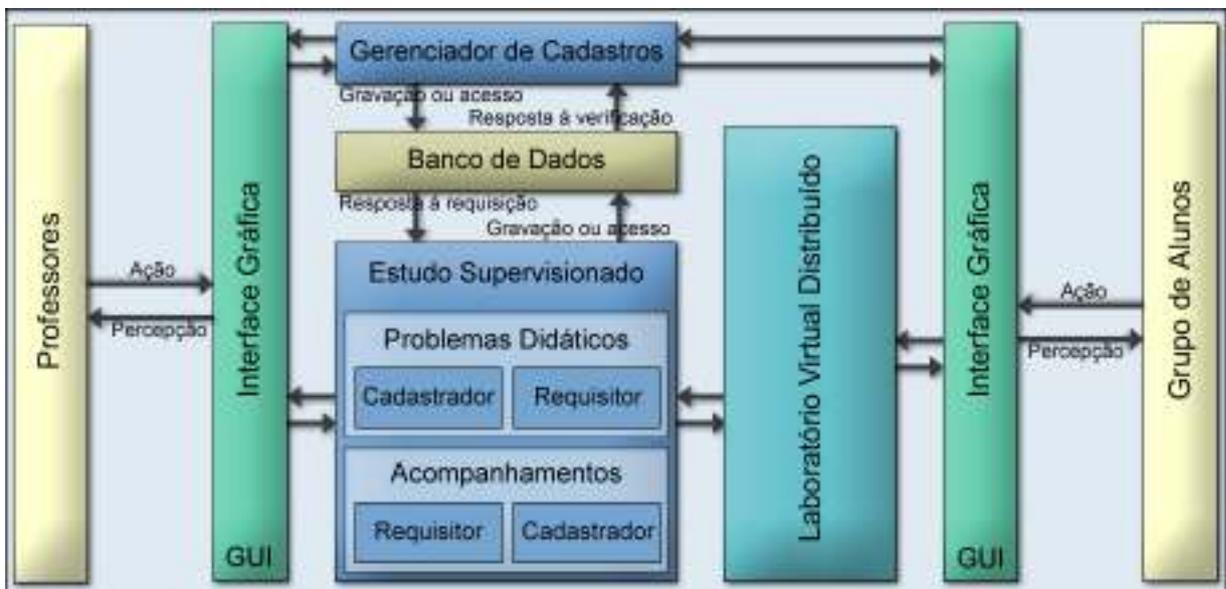


Figura 4.2: Arquitetura do sistema.

O sistema possui duas interfaces gráficas (GUI – *Graphics User Interface*) distintas, uma para professores outra para alunos. Ambas as interfaces acessam o Gerenciador de Cadastros, cuja função efetiva é restringir a entrada de usuários em áreas indevidas do sistema. Basicamente, ele coleta informações como *login*, senha e tipo (professor ou aluno) e verifica a sua existência no Banco de Dados, caso ele encontre o usuário é liberado para a área referente ao seu tipo de usuário, caso contrário ele bloqueia o acesso ao sistema. O Gerenciador de Cadastros pode ainda incluir novos cadastros no Banco de Dados. Porém, estes cadastros serão susceptíveis a uma ativação, que será realizada por usuários responsáveis desta ação. Por exemplo, cadastros de alunos serão ativados por professores, cadastros de professores serão ativados por um administrador ou professor/administrador.

O Gerenciador de Cadastros permite o acesso de Professores ao Estudo Supervisionado. Este bloco permite que o professor armazene problemas no Banco de Dados,

os quais ele carrega no Laboratório Virtual Distribuído para os alunos. É importante destacar que o professor pode carregar um problema para cada grupo de alunos, sendo ele diferente ou não.

Os alunos ao serem liberados pelo Gerenciador Cadastros podem acessar o Laboratório Virtual Distribuído, que consiste em um ambiente em RV com um cenário de um laboratório de Física. Ele permite aos alunos realizar experimentos e a troca de informações por meio de um *Chat*. Ao ser executado pelos alunos, o laboratório já contém o problema a ser resolvido, este problema foi carregado previamente do Banco de Dados pelo bloco Estudo Supervisionado.

Ao finalizar suas atividades os alunos devem postar suas soluções e suas respectivas considerações no sistema. Para tanto, o bloco Estudo Supervisionado as grava no Banco de Dados, as quais ficam disponíveis para a requisição e análise de professores.

É importante ressaltar que os blocos Gerenciador de Cadastros e Estudo Supervisionado foram construídos, fundamentalmente, para oferecer suporte à aplicação das metodologias de resolução de problemas e permitir o acompanhamento e avaliação quanto a sua eficácia para a aprendizagem no ambientes virtual desenvolvido, o Laboratório Virtual Distribuído.

Após analisar efetivamente as funcionalidades dos blocos Gerenciador de Cadastros e Estudo Supervisionado percebeu-se que elas poderiam ser atendidas completamente pela plataforma de ensino a distância Moodle.

A plataforma Moodle, acrônimo de *Modular Object-Oriented Dynamic Learning Environment*, é um *software* livre de apoio à aprendizagem, e foi criado pelo educador e cientista computacional Martin Dougiamas em 2001. Voltado para programadores e acadêmicos da educação, constitui-se em um sistema de administração de atividades educacionais destinado à criação de comunidades on-line, em ambientes virtuais voltados para a aprendizagem colaborativa [Moodle-Community, 2009]. Permite, de maneira simplificada, a um estudante ou professor integrar-se, estudando ou lecionando, num curso on-line à sua escolha.

O Moodle é gratuito e pode ser instalado em diversos ambientes (Unix, Linux, Windows, Mac OS) desde que os mesmos consigam executar a linguagem PHP. Como base de dados podem ser utilizados MySQL, PostgreSQL, Oracle, Access, Interbase ou ODBC. É desenvolvido colaborativamente por uma comunidade virtual, que reúne programadores e

desenvolvedores de *software* livre, administradores de sistemas, professores, designers e usuários de todo o mundo [Moodle-Community, 2009].

Muitas instituições de ensino (básico e superior) e centros de formação estão adaptando a plataforma aos próprios conteúdos, com sucesso, não apenas para cursos totalmente virtuais, mas também como apoio aos cursos presenciais. A plataforma também vem sendo utilizada para outros tipos de atividades que envolvem formação de grupos de estudo, treinamento de professores e até desenvolvimento de projetos.

A plataforma Moodle disponibiliza diversos recursos e permite a realização de várias atividades, como por exemplo: gestão de conteúdos; avaliação do curso; blog; chat; diálogo; diário; fórum; glossário; lição; pesquisa de opinião; questionário (em diversos formatos); scorm; tarefas; trabalho com revisão; wiki; geração e gestão de base de dados; suporte multi-idioma (são mais de sessenta idiomas suportados pela interface atual); etc. A Figura 4.3 ilustra a interface da plataforma Moodle.



Figura 4.3: Interface da plataforma de ensino a distância Moodle.

Outro fator preponderante que influenciou na escolha da plataforma Moodle para complementar este trabalho, foi o fato que o público alvo para a avaliação do sistema já utilizavam esta plataforma há dois anos aproximadamente para realizar suas atividades acadêmicas. Neste contexto, os usuários a serem avaliados, ambientados com o Moodle há dois anos, apresentariam menores dificuldades em trabalhar com esta plataforma, uma vez comparada com um sistema totalmente novo para eles. Esta ambientação é de grande importância para a avaliação do sistema, pois pode influenciar diretamente os resultados finais obtidos por meio deste trabalho. Sendo assim, a arquitetura geral do sistema pode ser reestruturada conforme ilustra a Figura 4.4.

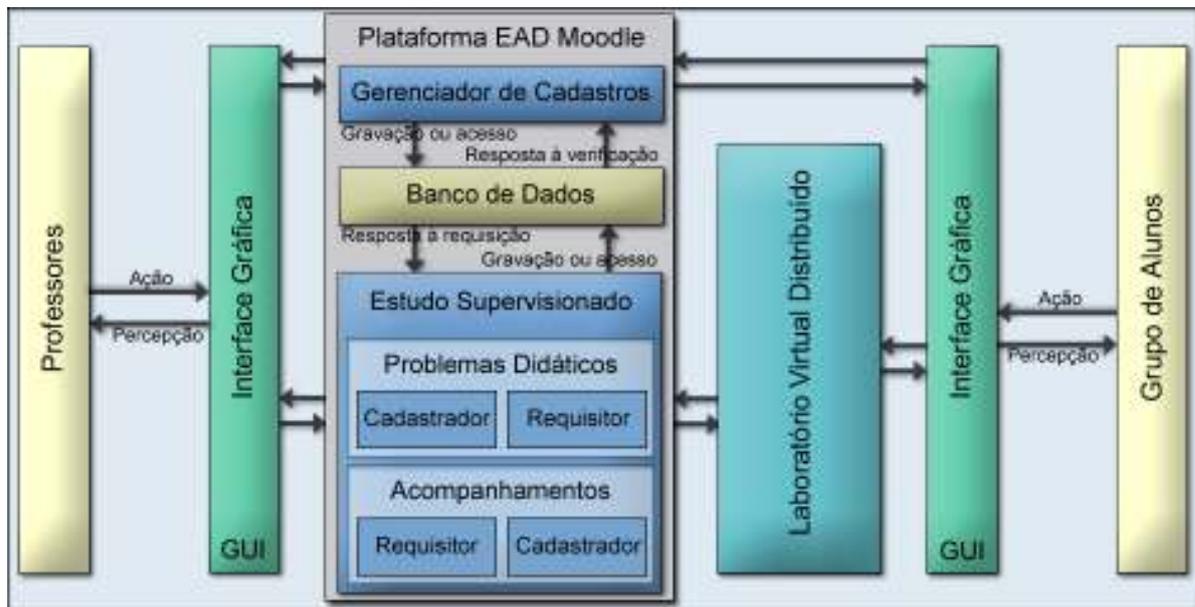


Figura 4.4: Arquitetura do sistema com a plataforma Moodle.

A ligação entre a plataforma Moodle e o Laboratório Virtual Distribuído (LVD) será realizada por meio de um link na plataforma endereçado ao ambiente em RV do laboratório. A próxima seção apresenta com maiores detalhes o projeto do ambiente virtual do LVD.

4.3 Especificações do Laboratório Virtual Distribuído

Nesta seção, são apresentadas algumas documentações referentes ao LVD. Para facilitar a compreensão sobre esta parte do sistema, primeiramente realizou-se uma abordagem ao LVD que propicia a sua visualização e percepção como um sistema local. Posteriormente foi abordado o seu sistema de distribuição e descrita a sua conectividade com o sistema local, transparecendo, deste modo, a completude de ambos.

4.3.1 O sistema local do LVD

O conjunto de componentes considerado como local no LVD é a parte do sistema que independe dos componentes de distribuição, ou seja, o sistema local funcionaria mesmo em um computador desconectado da rede. Todo o laboratório virtual, considerado de maneira local, faz parte deste sistema. Um usuário, mesmo fora da rede de computadores, pode usar o laboratório virtual normalmente.

O laboratório virtual é composto por diferentes objetos virtuais de aprendizagem. O sistema foi projetado de modo que cada objeto virtual independa dos demais. Sendo que, qualquer influência que um objeto possa efetuar sobre outro é realizada por intermédio de um Centro de Atualização, responsável pela transferência de informações entre os objetos virtuais. Deste modo, os objetos virtuais não efetuam trocas de dados diretamente uns com os outros. Qualquer modificação realizada em um objeto, pelo usuário, é automaticamente enviada para o Centro de Atualização que as transfere para o restante dos objetos. Um objeto virtual possui a propriedade de se auto-atualizar com referência em informações externas, sejam elas advindas de uma intervenção direta do usuário ou provenientes do Centro de Atualização. O modo com que cada objeto se auto-atualiza depende da sua natureza física, ou seja, este processo está relacionado com suas características físicas reais e as leis físicas que as regem.

Este projeto visa garantir a eficiência do sistema e manter uma relação coerente entre os objetos virtuais criados e a teoria de objetos de aprendizagem. O projeto ainda tem como intuito proporcionar certo nível de robustez ao sistema, de modo que, seja facilitada a inserção de novos objetos virtuais no laboratório, com poucas alterações necessárias no sistema.

As próximas seções abordam, como maiores detalhes, o projeto referente ao sistema local do LVD, sendo apresentadas algumas de suas documentações UML, as quais têm como intuito facilitar a compreensão sobre o funcionamento e a organização do LVD.

4.3.1.1 Diagrama de casos de uso do sistema local do LVD

Ao usar o ambiente virtual do LVD, o aluno realiza seus experimentos por meio dos objetos virtuais de aprendizagem. As suas ações no interior do laboratório se resumem em manipular intencionalmente estes objetos, ou seja, os alunos devem alterá-los para alcançar um determinado objetivo, que normalmente relaciona-se com atingir a resolução do problema proposto. Esta alteração pode ser de qualquer espécie, como por exemplo, trocar o objeto de

lugar, modificar o seu tamanho, intervir no seu funcionamento de alguma forma, etc. A Figura 4.5 apresenta o caso de uso referente a esta ação do usuário.

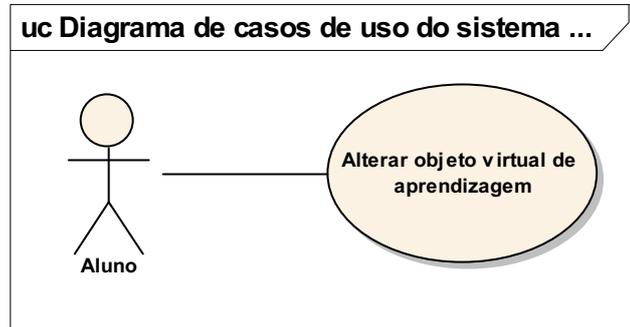


Figura 4.5: Diagrama de caso de uso do sistema local do LVD.

4.3.1.2 Diagrama de atividade do sistema local do LVD

O diagrama de atividades é uma notação UML, similar a um fluxograma, que é usada para representar o que acontece quando o sistema executa suas funções, ou seja, ele indica os fluxos conduzidos por processamentos. Os diagramas de atividades apresentam uma simbologia própria, como, por exemplo: os retângulos de cantos arredondados, que representam uma função específica do sistema; as setas, que definem o fluxo através do sistema; os losangos, que representam uma decisão de desvio, barras horizontais e verticais para indicar a existência de atividades em paralelo, etc [Pressman, 2006].

Quando um objeto virtual é alterado pelo usuário, imediatamente o sistema efetua uma correspondência entre a alteração e um conjunto de dados do objeto denominados atributos de entrada, que são o único meio pelo qual o objeto virtual recebe informações externas. Após receber estes atributos de entrada, são calculados os atributos de saída, que são o único meio pelo qual o objeto virtual pode enviar informações para o centro de atualizações. Este cálculo é realizado por meio das equações de leis físicas que regem o comportamento do objeto no mundo real. Com base nestes atributos de saída o objeto se auto-atualiza, e envia os seus valores para o Centro de Atualização. O Centro de Atualização por sua vez reenvia estes valores para o restante dos objetos virtuais. Ao receber estes dados, os objetos são atualizados com o mesmo processo do objeto de origem. A Figura 4.6 ilustra o diagrama de atividades que apresenta o fluxo de processos descritos acima.

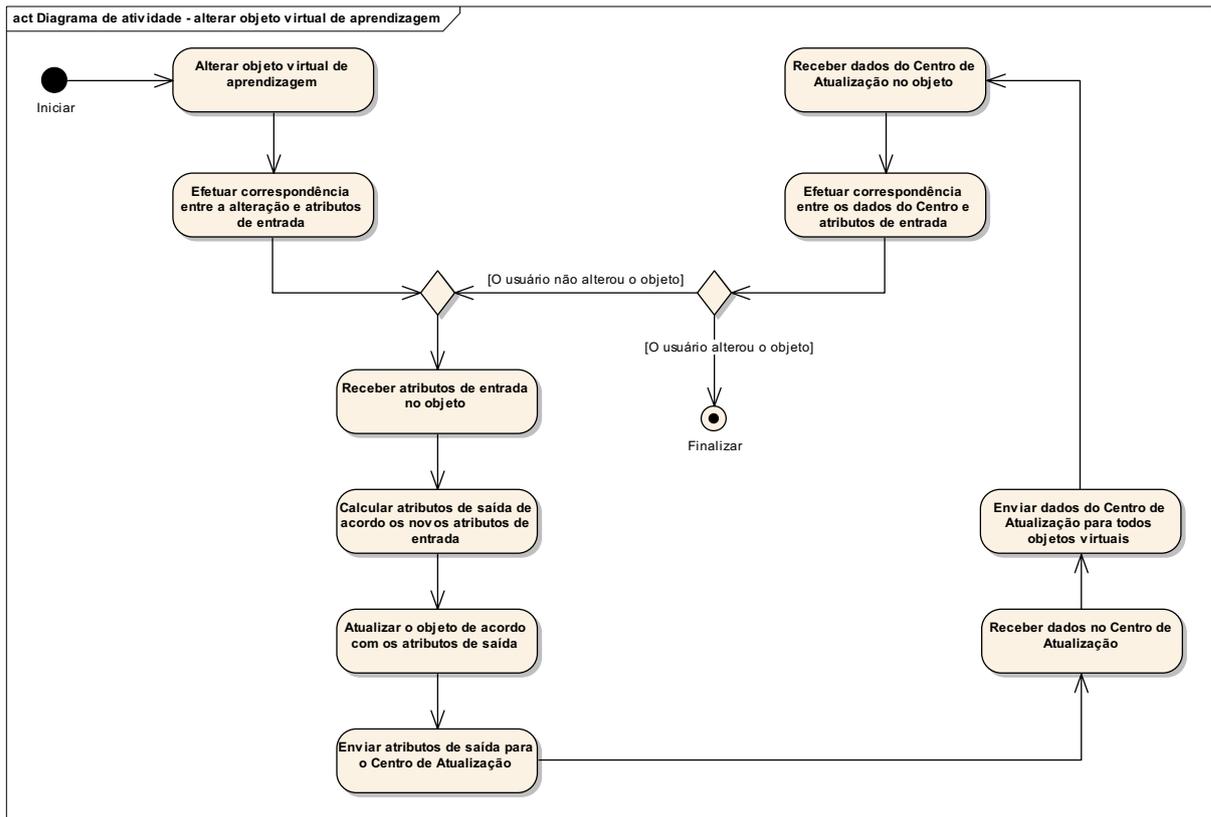


Figura 4.6: Diagrama de atividade do sistema local do LVD.

Outra documentação criada para o sistema local do LVD foi a sua arquitetura, a qual é abordada na próxima seção.

4.3.1.3 Diagrama da arquitetura do sistema local do LVD

Para possibilitar a construção efetiva do sistema local do LVD de acordo com a sua forma proposta utiliza-se um padrão para o desenvolvimento dos objetos virtuais de aprendizagem. Cada objeto é composto de uma modelagem tridimensional e um modelo comportamental. O modelo comportamental é responsável por receber novos atributos de entrada, calcular os atributos de saída por meio das equações físicas do objeto, e atualizar a modelagem tridimensional.

Construído nestes moldes o objeto pode ser ligado ao Centro de Atualização, sendo que, o objeto recebe e envia informações para o Centro de Atualização através, respectivamente, dos atributos de entrada e saída.

Com base nestas considerações foi definida a arquitetura do sistema local do LVD, a qual é ilustrada pela Figura 4.7.

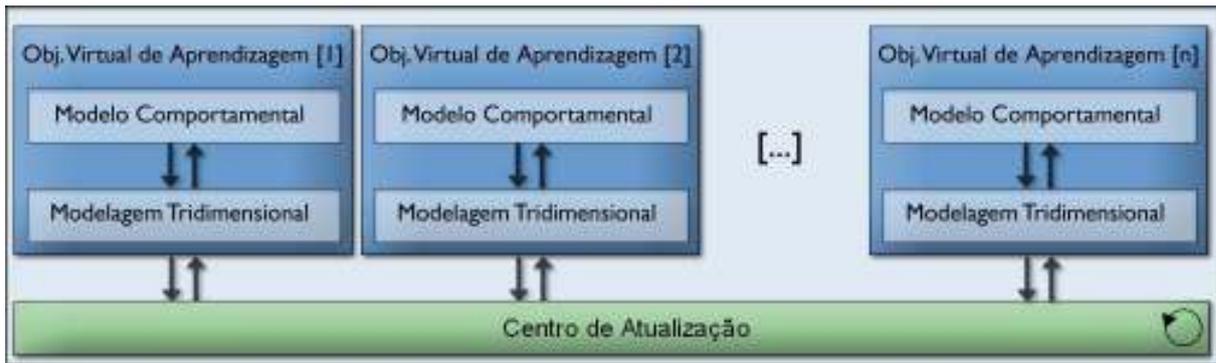


Figura 4.7: Arquitetura do sistema local do LVD.

O projeto proposto nesta seção descreve um sistema que trabalha eficientemente de forma local, porém, é preciso que o LVD trabalhe de forma distribuída, para que seja possível viabilizar a aprendizagem cooperativa para os alunos. Neste sentido, fez-se necessária a construção de um sistema de distribuição para o LVD, o qual deve considerar as características do sistema local, e trabalhe com ele sem alterar suas propriedades e funcionalidades. Este sistema de distribuição é abordado na próxima seção.

4.3.2 O sistema de distribuição do LVD

Os sistemas de RV multi-usuários é um campo crescente e próspero da pesquisa computacional. Estes sistemas permitem que os usuários geograficamente dispersos atuem em ambientes virtuais compartilhados por meio da troca de informações.

Sistemas de RV multi-usuários podem ser centralizados ou distribuídos. No modelo centralizado um computador recebe todas as informações provenientes de usuários que estão conectados ao ambiente virtual e que estejam em diferentes máquinas, armazena as mudanças ocorridas e as aplica no ambiente virtual do computador central, devolvendo para cada um dos usuários conectados os resultados das alterações feitas sobre o ambiente virtual, atualizando-os. A Figura 4.8 ilustra a estrutura do modelo centralizado.



Figura 4.8: Modelo centralizado de sistema de RV multi-usuários.

O modelo centralizado apresenta facilidades em relação à implementação das estruturas de controle da comunicação entre usuários. Porém, ele possui restrições quanto à escalabilidade de distribuição. A escalabilidade refere-se ao fato do ambiente virtual crescer e suportar uma grande quantidade de usuários sem perdas na qualidade de comunicação. Isso se deve ao fato de que quanto maior o número de usuários em um ambiente virtual, maior será o tráfego de mensagens com o computador central e dessa maneira a velocidade de acesso cai, pois o computador terá que receber, processar e devolver as mensagens a todos os usuários.

O modelo distribuído melhora a escalabilidade em relação ao modelo centralizado. Neste modelo cada usuário possui uma cópia ou parte do ambiente virtual em sua máquina, realizando, ele mesmo, as tarefas de renderização, computação e animação dos objetos do mundo virtual. Assim, quando um usuário realiza qualquer alteração sobre o ambiente virtual, ele mesmo se encarrega de comunicar a todos os outros usuários sobre a atualização feita. A Figura 4.9 ilustra a estrutura de um modelo distribuído.

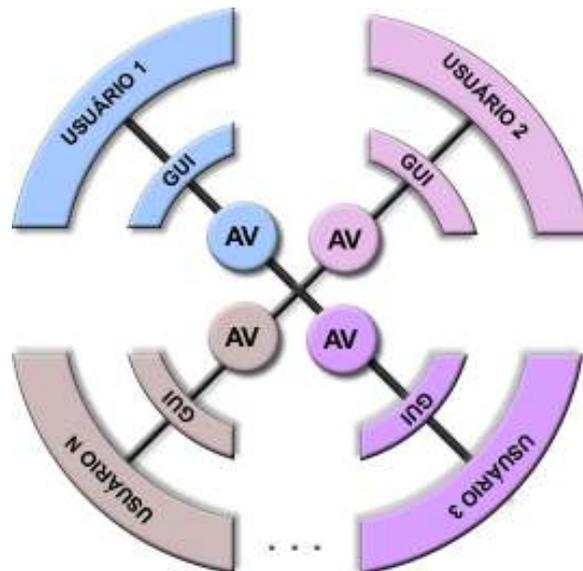


Figura 4.9: Modelo distribuído de sistema de RV multi-usuários.

No modelo distribuído, o ambiente virtual pode ser replicado (para mundos pequenos) ou particionado (para mundos virtuais de grande porte). Em um sistema replicado com “n” usuários, qualquer alteração no ambiente virtual é comunicada para todas as suas (n-1) versões, onde estão os outros usuários, constituindo, dessa forma, a difusão chamada de *broadcast*, na qual uma única mensagem é enviada para todos os usuários que estão conectados na rede. A Figura 4.10 ilustra este modelo.

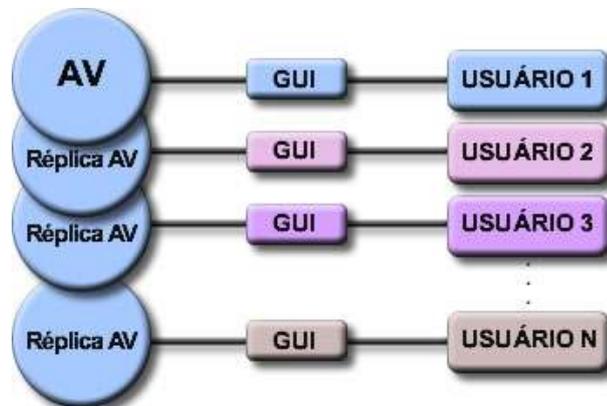


Figura 4.10: Modelo distribuído replicado.

Já em um sistema particionado com “n” usuários, a situação é mais complexa, uma vez que o ambiente virtual é dividido em várias partes e cada máquina ficará encarregada de uma delas. Como o usuário pode navegar no ambiente virtual, ele poderá penetrar em outras regiões, de forma que sua máquina ou servidor deverá receber uma réplica da região, onde ele se encontra. Assim cada máquina estará cuidando de uma região fora da sua parcela. Se existirem vários usuários em uma mesma região do ambiente virtual, esse grupo de usuários receberá uma cópia dessa região. Qualquer alteração no ambiente virtual, feita por um

membro do grupo, será retransmitida para o restante do grupo, constituindo a retransmissão por grupo conhecida como *multicast*. Da mesma forma que ocorre no *broadcast*, onde uma única mensagem é enviada para todos que estão conectados ao ambiente, ocorre com a transmissão *multicast*, com a diferença que só os usuários interessados nas mensagens irão recebê-las. A Figura 4.11 ilustra o modelo distribuído com sistema particionado.

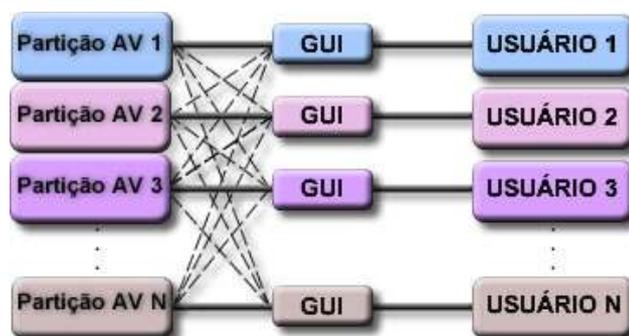


Figura 4.11: Modelo distribuído particionado.

O projeto deste sistema baseia-se no modelo distribuído replicado, pois considerou-se que o LVD é um ambiente virtual de pequeno porte, sendo assim, desnecessária a aplicação do modelo particionado. As próximas seções apresentam maiores detalhes sobre os métodos usados para realizar a distribuição do LVD.

4.3.2.1 Diagramas de atividade do sistema de distribuição do LVD

A comunicação entre os usuários é realizada com base no modelo distribuído replicado, sendo que a sua conexão baseia-se no modelo cliente/servidor. Deste modo entende-se cada réplica do ambiente virtual como um cliente. O servidor é um componente que possui a única função de viabilizar a troca de dados entre estes clientes. Em suma, em relação à distribuição, cada cliente pode realizar duas ações: enviar e receber dados.

Para enviar dados de um cliente para outro, inicialmente, o sistema os envia do Centro de Atualização para um componente denominado Transmissor. Em seguida, este componente compara os dados recebidos com os dados de outro componente chamado Controle E/S. Somente os dados que diferem, nesta comparação, são enviados para o servidor pelo Transmissor. Deste modo, é minimizada a transferência de dados na rede. Caso todos os dados do Transmissor sejam iguais aos dados do Controle E/S, significa que não ocorreu nenhuma alteração no ambiente virtual, portanto, não devem ser transmitidas informações. O servidor, por sua vez, reenvia os dados recebidos para o restante dos clientes. Ao enviar os dados para o servidor o sistema atualiza o Controle E/S com estes valores. A Figura 4.12 apresenta o diagrama de atividades relativo à seqüência de processos descrita acima.

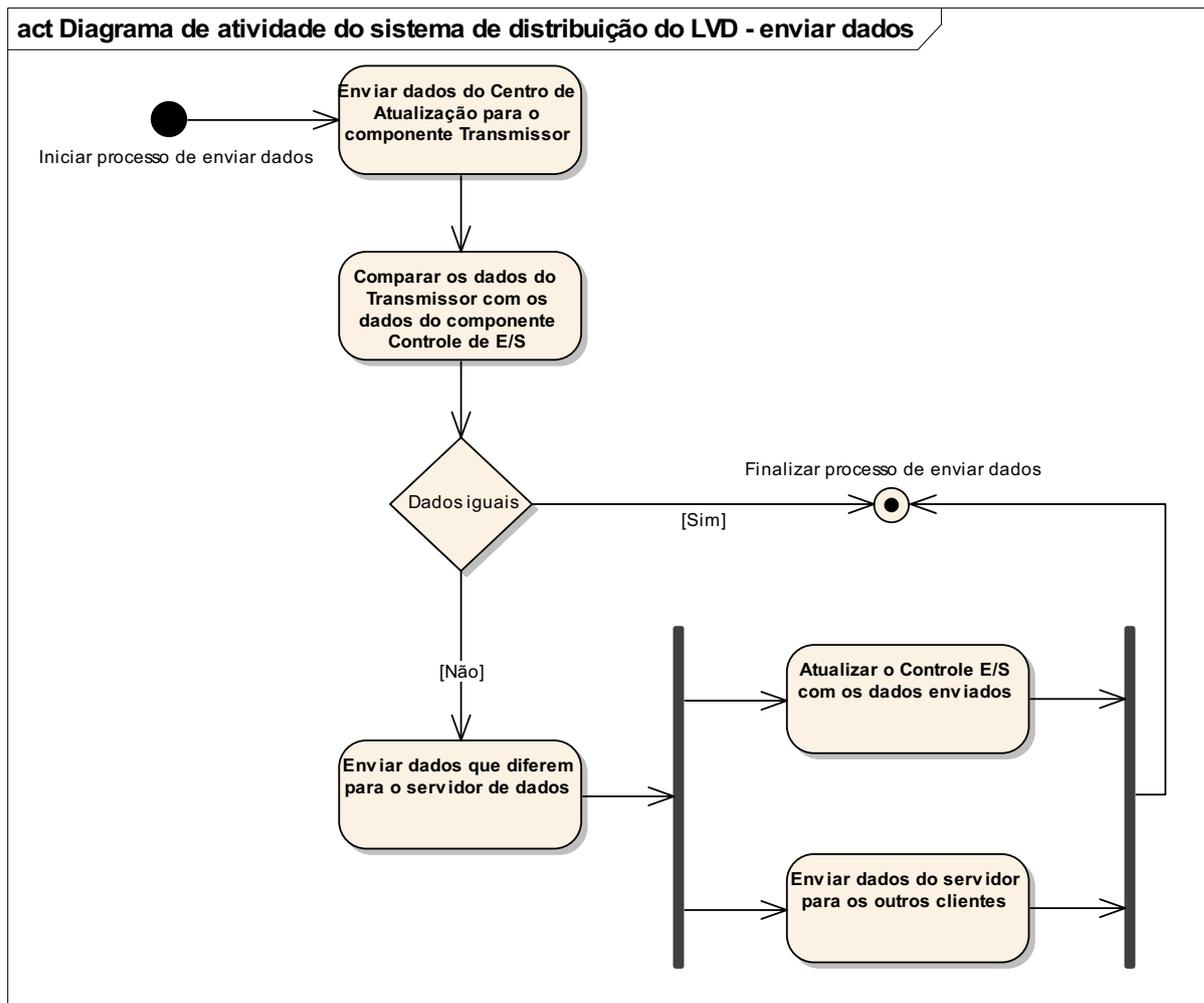


Figura 4.12: Diagrama de atividade do sistema de distribuição do LVD - enviar dados.

O sistema recebe os dados diretamente do servidor por meio do Transmissor. Ele reenvia os dados recebidos para o Centro de Atualização e atualiza o Controle E/S. Uma vez no Centro de Atualização estes dados são tratados como se estivessem em um sistema puramente local, visto na seção 4.3.1. O centro automaticamente envia todos os dados para cada objeto virtual interessado, e cada objeto se auto-atualizará com base nestes dados. Os processos computacionais envolvidos na ação de receber dados são apresentados no diagrama de atividades da Figura 4.13.

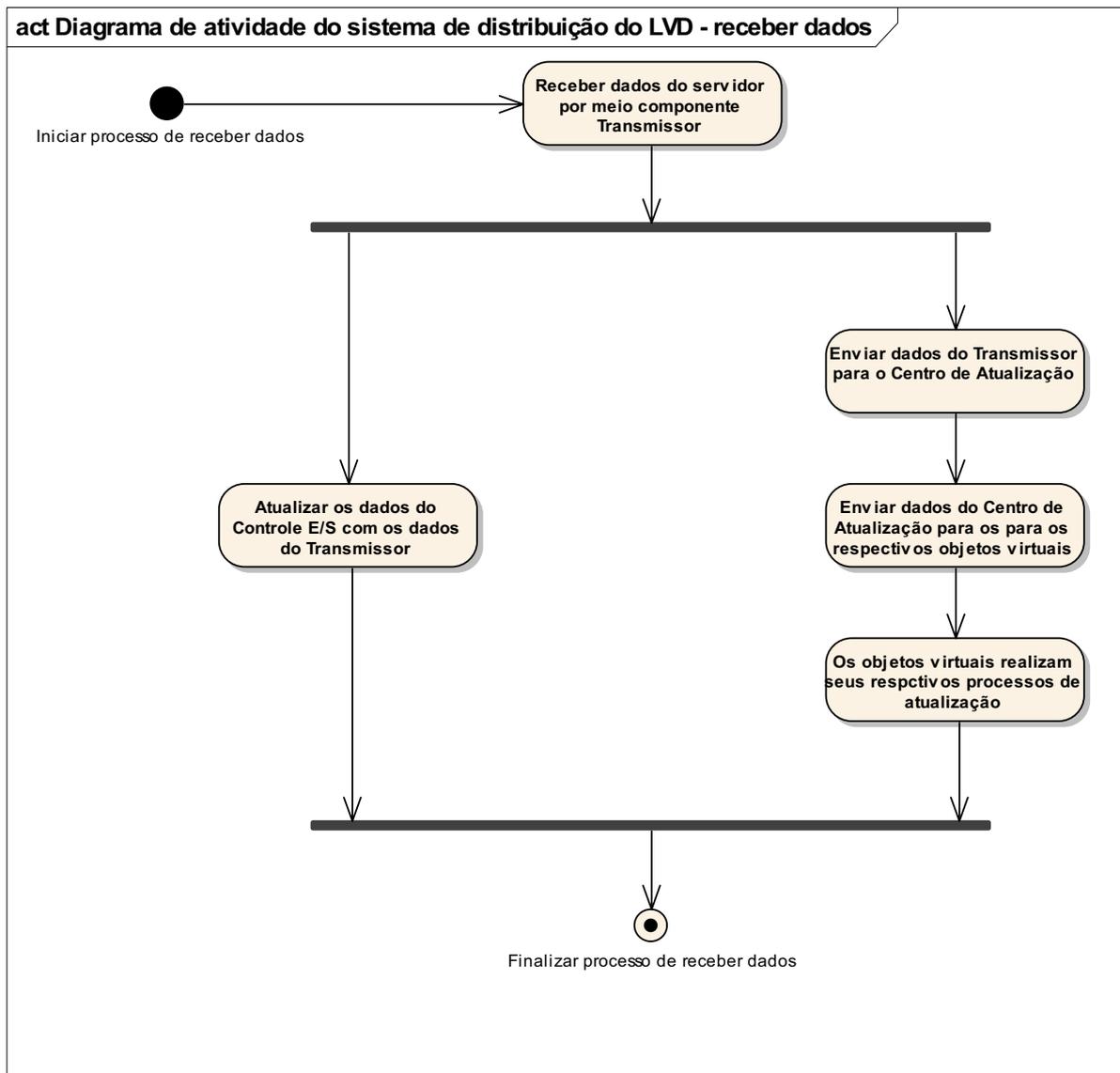


Figura 4.13: Diagrama de atividade do sistema de distribuição do LVD - receber dados.

É importante ressaltar que o Centro de Atualização realiza seus processos ininterruptamente, ou seja, ele recebe e envia dados incessantemente. Neste contexto, toda informação advinda de um objeto virtual ou do Transmissor é automaticamente retransmitida para os outros objetos e devolvida para o próprio Transmissor. Porém, pode-se perceber que o fato de se realizar uma comparação entre os dados do Transmissor e os do Controle E/S, e transmitir somente os dados diferentes para os outros clientes, evita o reenvio de dados que acabaram de ser recebidos, o que poderia causar conflitos e congestionamentos na rede de conexão dos computadores. Estas considerações são observadas com maiores detalhes na arquitetura de distribuição do LVD, abordada na próxima seção.

4.3.2.2 Diagrama da arquitetura de distribuição do LVD

A arquitetura de distribuição do LVD mantém a estrutura projetada para sistema local, sendo acoplados a esta estrutura somente dois componentes de distribuição, o Transmissor e o Controle E/S, como ilustra a Figura 4.14.

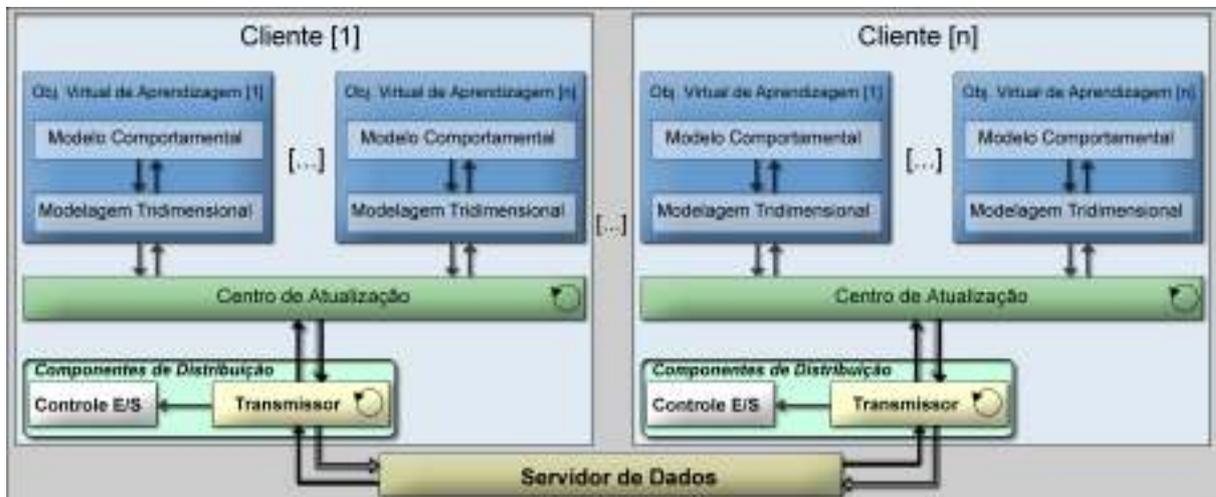


Figura 4.14: Arquitetura de distribuição do LVD.

O Transmissor é conectado ao Centro de Atualização e servidor de dados, a sua principal função é viabilizar a troca de dados entre eles. Para evitar transferências de dados desnecessárias e possíveis conflitos o Transmissor tem o auxílio do Controle E/S. Este componente tem a função de memorizar quais dados foram recebidos e enviados pelo Transmissor.

O conflito de dados ainda é possível com esta arquitetura quando dois ou mais clientes tentarem modificar o mesmo objeto em seus respectivos ambientes virtuais. Neste caso o sistema tentará enviar e receber dados por meio de uma mesma variável ao mesmo tempo. Porém, para estes casos existem técnicas consagradas para aplicações distribuídas, como, por exemplo, técnicas de *callback* ou *locks*, que podem ser usadas para resolver este problema. A técnica usada será apresentada ao serem abordados os detalhes de implementação deste sistema, no capítulo 5.

O servidor, nesta estrutura, trabalha apenas como um mediador entre os clientes, ou seja, ele tem, unicamente, a função de repassar informações advindas de um cliente para o restante dos clientes. Todos os cálculos necessários para a atualização dos ambientes virtuais são realizados localmente.

4.4 Considerações finais

A documentação apresentada neste capítulo mostra como o projeto modelado a partir dos requisitos identificados. E são apresentadas técnicas e arquiteturas que podem atribuir melhores qualidades ao sistema projetado, como por exemplo, maior robustez e escalabilidade de distribuição. O próximo capítulo aborda os detalhes de desenvolvimento que foram usados para implementar esta proposta.

Capítulo 5

Desenvolvimento do protótipo

5.1 Introdução

Neste capítulo são abordados os detalhes de implementação do LVD com maior relevância para o desenvolvimento deste trabalho. São apresentados, quando necessário, trechos do seu código e discutidas as suas respectivas funcionalidades no sistema.

5.2 Tecnologias utilizadas

Antes de apresentar efetivamente os detalhes de desenvolvimento deste sistema, será realizada uma breve exposição sobre as tecnologias necessárias para o seu funcionamento e concepção, bem como, será destacada a importância do uso de cada uma delas.

5.2.1 VRML e X3D

Existem várias abordagens sobre a RV, sendo que algumas requerem o uso de tecnologias avançadas como plataformas de *hardware* e *software* sofisticadas. No entanto, a linguagem VRML, acrônimo de *Virtual Reality Modeling Language*, possibilita a criação de mundos virtuais tridimensionais, com alta qualidade, aplicáveis à Internet, utilizando apenas um *browser* e um *plug-in* para realizar a visualização. A VRML permite a descrição de um espaço tridimensional, tornando possível construir mundos correspondentes a uma dada realidade ou completamente imaginários. Fundamentalmente, o *browser* gera em tempo real a representação visual equivalente à descrição textual da cena.

O X3D (Extensible 3D) é uma linguagem de descrição de cenas 3D de padrão aberto, capaz de representar e comunicar cenas tridimensionais e objetos, desenvolvida com a sintaxe XML – *Extensible Markup Language*. Essa sintaxe foi escolhida porque permite uma melhor interoperação com a *web* e permite incorporar novas tecnologias de forma padronizada. *Web Services* e SVG são bons exemplos de tecnologias que podem operar perfeitamente integradas com o X3D [Cardoso, Paiva e Soares, 2007].

Construído a partir da VRML, o X3D apresenta um conjunto de novas funcionalidades, tais como interface avançada de programação de aplicações, formatos de codificação de dados adicionais, conformação mais estrita e arquitetura dividida em componentes, o que permite o suporte modular ao padrão, e ainda ocorreram trocas de alguns nomes de campos para uma maior consistência. Porém ele mantém os recursos básicos da VRML como, por exemplo: oferecer suporte para gráficos 3D, transformações de geometria, iluminação, materiais e texturas; permite animação com temporizadores e interpoladores de condução contínua; permitir interação com o mouse e entradas de teclado; a navegação no ambiente acontece por meio do uso de câmeras, com características de colisão, proximidade e visibilidade, detecção e vários tipos de iluminação; por meio de um nó *script* é possível mudar dinamicamente a cena através de linguagens de programação como o JavaScript e o Java.

Pode-se considerar que a VRML e o X3D estabelecem duas classes de entidades: os objetos sensoriais, que envolvem características visuais, sonoras, etc., e objetos comportamentais que incluem cliques de mouse, temporizadores, etc. As ligações entre esses objetos, denominadas rotas, representam o fluxo de propagação de eventos. Esses sistemas de entidades assemelham-se a um grafo direcionado acíclico, sendo chamados de grafo de cena (o conjunto de objetos) e grafo de comportamentos (o conjunto de rotas) [X3d, 2008].

O grafo de cena do X3D é quase idêntico ao do VRML97, porém, ele sofreu algumas modificações com o intuito de: incorporar os avanços nos dispositivos gráficos comerciais, por meio da introdução de novas funcionalidades; melhorar a precisão do sistema de iluminação e do modelo de eventos; e dar acesso aos valores de transparência nos campos de cores [X3d, 2008].

O X3D emprega uma arquitetura modular para promover uma maior expansibilidade e flexibilidade. Ele introduz o conceito de perfis, uma coleção predefinida de componentes comumente encontrados em certos domínios de aplicações, plataformas, entre outros. O VRML97 requer um suporte completo das suas funcionalidades para estar em conformidade. Já o X3D permite vários graus de suporte do padrão para suprir uma variedade de necessidades [X3d, 2008].

Assim como na VRML, para que a visualização de arquivos X3D seja possível em um *browser* se faz necessária à instalação de um *plug-in* específico. O Xj3D é uma ferramenta Java, com o código aberto, criada para a visualização e a manipulação de conteúdos escritos em VRML e X3D. Elaborada pelo grupo de trabalho da Web3D que idealizou o X3D, ela foi

desenvolvida inicialmente como carregador de arquivos X3D para a API Java 3D [X3d, 2008]. Contudo, o seu grau de desenvolvimento tornou-a principal ferramenta usada pelos grupos de teste da especificação X3D. A sua adoção pela comunidade X3D se deve ao fato do Xj3D ter ganhado algumas funcionalidades importantes, como, por exemplo, permitir o uso de diferentes vias para a renderização de cenas, tais como Java 3D, DirectX e OpenGL; e possibilitar o uso da interface de programação externa do X3D, a SAI - *Scene Access Interface*, e da interface de programação externa do VRML, a EAI - *External Authoring Interface*. Fundamentalmente, a SAI e a EAI permitem o uso de toda a estrutura da linguagem Java. Tal estrutura possui como grande trunfo um interpretador (*Java Virtual Machine - JVM*) que é instalado por padrão em muitos sistemas operacionais, ou pode ser implantado nos sistemas que não o possuem automaticamente, sem que para isso seja necessário qualquer esforço adicional do usuário [Brutzman e Daly, 2007].

Para possibilitar o acesso às funcionalidades da linguagem Java, existe um nó denominado *script* dentro da estrutura do X3D e VRML, que possibilita efetuar uma ligação das cenas do ambiente virtual com as classes Java. Em suma, a própria cena X3D (ou VRML) ao ser carregada em um *browser* X3D (ou VRML) invoca as classes Java, nas quais existe o código que efetuará as alterações pretendidas na cena.

Devido às características funcionais e possibilidades de desenvolvimento e aplicação de sistemas VRML e X3D e por: apresentarem portabilidade, ou seja, serem multiplataforma; contarem com a distribuição gratuita de sistemas que permitem executá-los; e disponibilizarem seus códigos abertos; optou-se por usar o padrão destas linguagens para realizar descrição dos cenários virtuais do LVCF. Optou-se ainda por utilizar a ferramenta Xj3D para a manipulação e visualização dos objetos virtuais destes cenários. O Xj3D foi escolhido neste projeto principalmente por fornecer o suporte dos recursos Java para a estrutura VRML e X3D.

No decorrer do desenvolvimento deste trabalho foram construídas duas versões dos cenários virtuais, uma em VRML, outra em X3D. Os grafos de cenas de ambas não apresentam diferenças sobressalentes. Porém, a versão em X3D não apresenta todas as suas funcionalidades ativas no Xj3D, pois, o *browser* não consegue transferir as atualizações de dados de objetos advindos de protótipos para os *scripts*. É importante destacar que a linguagem X3D é relativamente nova e ainda passa por diversas revisões, concorrentemente, as ferramentas que a suportam também estão sendo aperfeiçoadas e seus erros constantemente corrigidos referentes ao X3D. A VRML, por sua vez, é uma linguagem consolidada, já

foi revisada e seus erros foram minimizados. Neste contexto, a versão com todas as suas funcionalidade ativas é a desenvolvida em VRML, sendo ela a que passou por testes e foi avaliada pedagogicamente junto aos alunos.

5.2.2 JavaScript

Um componente estritamente importante, tanto para a VRML quanto o X3D, é o nó *script*. Ele provê meios de incluir formas mais complexas de interação, e pode ser usado para realizar uma grande variedade de tarefas, porém, ele é tipicamente usado para desempenhar as funções de:

- interagir como o estágio da "máquina de execução" para controlar uma animação;
- processar entradas para agir como o estado de Lógica em uma animação.

Como outros nós, um *script* recebe eventos de entrada e gera eventos de saída. Ao contrário de outros nós, entretanto, ele pode executar algum tipo de processamento antes de produzir esses eventos de saída. A parte do nó que realiza o processamento é um programa escrito em uma linguagem de programação suportada pelo *browser*, e as linguagens geralmente utilizadas são Java e JavaScript.

Cada vez que um nó *script* recebe um conjunto de eventos de entrada, o *browser* entrega-os para o programa especificado. Através de vários métodos (dependendo da linguagem utilizada para escrever o programa), cada evento é tratado. A abordagem mais comum para o tratamento de eventos de entrada é a utilizada em JavaScript: definir uma função no programa para cada evento de entrada listado no nó *script*. Quando um evento de entrada chega ao nó, o *browser* chama a função de mesmo nome. A função recebe como parâmetros o valor do evento e um *timestamp* que indica, geralmente, quando o evento foi enviado.

Para cada campo e evento de saída definido em um nó *script*, o *browser* cria uma variável global com o mesmo nome e o mesmo tipo, para ser usada pelo programa. Para inserir o valor de um evento de saída, basta atribuir um valor à variável correspondente.

A linguagem JavaScript é aplicada neste sistema, essencialmente, para o desenvolvimento do sistema local do LVD. O modelo comportamental de cada objeto virtual de aprendizagem e o Centro de Atualização são construídos por meio desta linguagem. O

JavaScript foi escolhido para implementar esta parte do projeto, fundamentalmente, por oferecer uma certa facilidade para programar animações em VRML ou X3D.

5.2.3 Ferramentas Java de distribuição

O JavaScript é uma linguagem desenvolvida especificamente para a programação de *scripts*, diferentemente do Java. Ele não oferece todos os recursos computacionais da linguagem Java e, principalmente, não inclui nenhum suporte às funções de acesso à rede. Por esta razão, para criar ambientes virtuais multi-usuários, em VRML ou X3D, os desenvolvedores geralmente optam pela linguagem Java.

É importante ressaltar que a VRML e o X3D não incluem mecanismos que permitam a interoperabilidade entre aplicações de rede. Os ambientes virtuais multi-usuários que desejem utilizar arquivos VRML ou X3D para representar suas cenas devem criar sistemas de compartilhamento de informações personalizados.

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 pelo programador James Gosling, na empresa *Sun Microsystems*. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "*bytecode*" que é interpretado por uma máquina virtual (JVM – *Java Virtual Machine*) [Deitel, 2005]. Esta estratégia de compilação seguida por interpretação é que garante a Java a sua portabilidade. A linguagem Java possui recursos consagrados para distribuição de dados via rede, como, por exemplo, o RMI e o *Socket*.

O RMI – *Remote Method Invocation* (Invocação de Métodos Remotos) é uma tecnologia suportada pela plataforma Java que possui como intuito facilitar o desenvolvimento de aplicações distribuídas. Como o próprio nome indica, o RMI permite ao programador invocar métodos de objetos remotos, ou seja, que estão alojados em JVMs distintas, de uma forma muito semelhante às invocações a objetos locais.

O protocolo utilizado para a comunicação entre os objetos pelo RMI é o JRMP – *Java Remote Method Protocol* [Trinta, 2000]. Este protocolo baseia-se na serialização de objetos Java, isto é, os objetos utilizados como parâmetros podem ser transportados como um fluxo de dados, por meio de uma sessão TCP – *Transport Control Protocol*.

O processo de comunicação envolve, primeiramente, a localização do objeto, que é fornecida através de uma *string* que contém o nome do objeto e o endereço da máquina que o abriga. O uso de um protocolo padrão facilita a implementação de RMI, porém pode provocar

uma perda de desempenho, devido às mensagens de controle enviadas pelo TCP para garantir a entrega das mensagens.

Os *Sockets* são abstrações criadas com o intuito de facilitar o desenvolvimento de aplicações que envolvem a comunicação entre dois ou mais computadores interligados por uma rede. Um *Socket* representa uma conexão entre duas máquinas onde tal conexão funciona como um duto de dados, isto é, um canal que permite a transmissão de dados de uma máquina para a outra de forma bidirecional possibilitando o envio e recebimento simultâneo de dados. Isto significa que cada *Socket* possui um canal de entrada e outro de saída, sendo que, o que é enviado pelo canal de saída de uma máquina é recebido pelo canal de entrada da outra máquina e vice-versa.

Os *Sockets* têm dois modos principais de operação: o modo baseado em conexões e o modo sem conexão. O modo a ser utilizado é determinado pelas necessidades de um aplicativo. Se a confiabilidade é importante, então a operação baseada em conexões é a melhor opção. Os servidores de arquivos precisam fazer todos os seus dados chegarem corretamente e em seqüência. Garantir a seqüência e a correção dos dados exige processamento extra e utilização de mais memória; esse *overhead* adicional pode reduzir o tempo de resposta de um servidor.

A operação baseada em conexões emprega o protocolo TCP. Um *Socket* nesse modo de operação precisa se conectar ao destino antes de transmitir os dados. Uma vez conectados, os *Sockets* são acessados pelo uso de uma interface de fluxos: abertura-leitura-escrita-fechamento. Tudo que é enviado por um *Socket* é recebido pela outra extremidade da conexão, exatamente na mesma ordem em que foi transmitido. A operação baseada em conexões é menos eficiente do que a operação sem conexão, mas é garantida [Hopson e Ingram, 1997].

A operação sem conexão utiliza o UDP – *User Datagram Protocol*. Um *datagrama* é uma unidade autônoma que tem todas as informações necessárias para tentar fazer sua entrega. Similar a um envelope, o *datagrama* tem um endereço do destinatário e do remetente e contém em seu interior os dados a serem enviados. Um *Socket* nesse modo de operação não precisa se conectar a um *Socket* de destino; ele simplesmente envia o *datagrama*. O protocolo UDP só promete fazer o melhor esforço possível para tentar entregar o *datagrama*. A operação sem conexão é rápida e eficiente, mas não é garantida [Hopson e Ingram, 1997].

Existe ainda a CORBA (*Common Object Request Broker Architecture*), criada pela OMG (*Object Management Group*), que é uma especificação-padrão de arquitetura de sistemas distribuídos que conquistou ampla aceitação. CORBA é um padrão aberto elaborado para habilitar a interoperação entre programas em sistemas heterogêneos, bem como homogêneos. Similar à RMI, CORBA suporta objetos como parâmetros ou valores de retorno em procedimentos remotos durante a comunicação interprocessos. Entretanto, diferentemente da RMI (que é baseada em Java), CORBA é independente de linguagem e de sistema, o que significa que aplicações escritas em linguagens de programação diferentes e em sistemas operacionais diferentes operam entre si por meio de acesso a um núcleo comum da arquitetura CORBA [Kurose e Ross, 2006].

Em termos de comparação de desempenhos entre as tecnologias *Socket* (TCP), RMI e CORBA, Munoz e Zalewshi (2001) realizaram testes avaliativos e obtiveram os resultados apresentados na Tabela 5.1. Foram analisadas aplicações cliente/servidor, com o uso do processador Sun UltraSparcs e sistema operacional *Solaris 2.5.1*, e *Pentium II* com *Windows NT 4.0*, sendo que, os principais aspectos abordados foram os tempos de resposta dos clientes para o servidor, considerando as variações do número de clientes e número de parâmetros dos métodos.

Tabela 5.1: Tempos médios de resposta dos clientes em segundos das tecnologias *Socket*, RMI e CORBA.
Fonte: [Munoz e Zalewshi, 2001]

Execution Nº	Local	Remote	Local	Remote	Local	Remote	Local	Remote	Local	Remote
	Sockets	Sockets	CorbaSII(*)	CorbaSII(*)	CorbaSII(**)	CorbaSII(**)	CorbaDII	CorbaDII	RMI	RMI
1	0,340	0,521	1,021	1,031	1,022	1,071	1,061	1,071	0,941	1,122
2	0,350	0,520	1,021	0,982	1,111	1,101	1,512	1,101	0,931	1,111
3	0,351	0,511	1,021	1,022	1,051	1,081	1,142	1,082	0,931	1,152
4	0,370	0,520	1,061	1,011	1,382	1,031	1,513	1,152	0,931	1,152
5	0,351	0,551	1,312	1,011	1,101	1,122	1,181	1,102	0,941	1,152
6	0,351	0,530	1,332	1,022	1,382	1,112	1,061	1,071	0,931	1,101
7	0,350	0,510	1,212	1,001	1,102	1,142	1,462	1,202	0,921	1,132
8	0,371	0,521	1,182	1,021	1,042	1,081	1,162	1,102	0,931	1,102
9	0,351	0,531	1,082	1,061	1,052	1,161	1,132	1,072	0,932	1,151
10	0,350	0,541	1,062	1,021	1,382	1,141	1,131	1,152	0,941	1,102
	Local	Remote	Local	Remote	Local	Remote	Local	Remote	Local	Remote
Statistics	Sockets	Sockets	CorbaSII(*)	CorbaSII(*)	CorbaSII(**)	CorbaSII(**)	CorbaDII	CorbaDII	RMI	RMI
average	0,354	0,526	1,131	1,018	1,163	1,104	1,236	1,111	0,933	1,128
max	0,371	0,551	1,332	1,061	1,382	1,161	1,513	1,202	0,941	1,152
min	0,340	0,510	1,021	0,982	1,022	1,031	1,061	1,071	0,921	1,101
stdev	0,010	0,013	0,121	0,020	0,154	0,039	0,184	0,044	0,006	0,023

Por meio dos resultados apresentados pode-se observar que Munoz e Zalewshi (2001) obtiveram um melhor desempenho com *Sockets*, seguido respectivamente por, localmente:

RMI, CORBASII(*), CORBASII(**) e CORBADII; e remotamente: CORBASII(*), CORBASII(**), CORBADII e RMI.

Em um estudo similar, Ahuja e Quintão (2000) realizaram a comparação entre *Sockets* e RMI. Os testes foram feitos no sistema operacional *Solaris* em uma rede de alta velocidade, sendo que, os resultados deste trabalho também indicaram um melhor desempenho da ferramenta *Socket*. A Figura 5.1 apresenta os gráficos das leituras feitas em servidores de dados *Socket* e RMI. Foram realizados dois tipos de observações, na primeira o servidor apenas lê os dados recebidos dos clientes, na segunda o servidor lê os dados e atualiza o restante dos clientes.

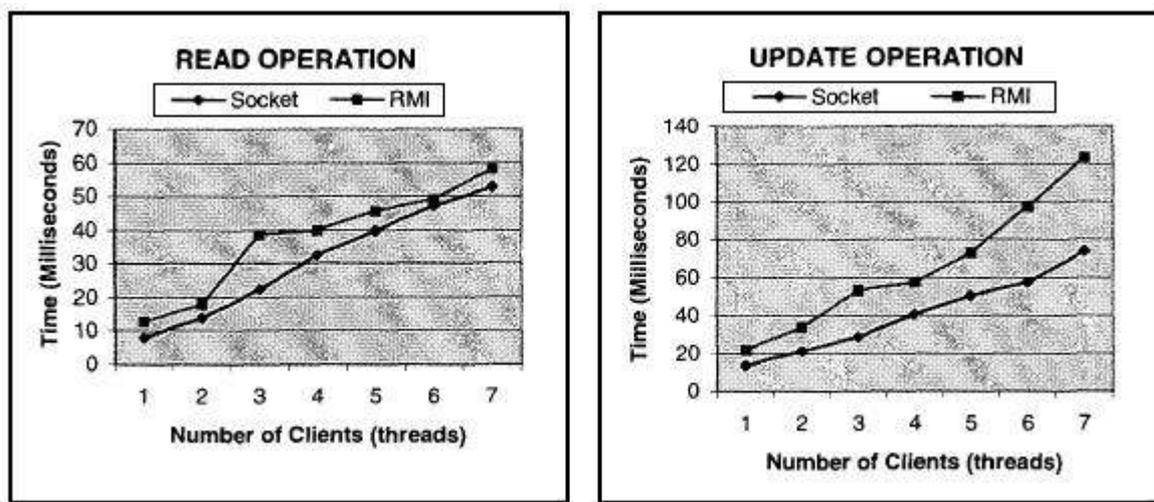


Figura 5.1: Comparação entre *Sockets* e RMI.
Fonte: [Ahuja e Quintao, 2000]

É importante ressaltar que, neste contexto, o melhor desempenho está diretamente relacionado ao menor tempo de transmissão de dados, porém, dependendo da aplicação, ele pode relacionar-se com outros fatores como, por exemplo, com a segurança, a confiabilidade, etc.

A confiabilidade em um ambiente virtual distribuído é a garantia de que os dados enviados de um computador que esteja conectado a este ambiente cheguem aos demais computadores de forma correta, evitando assim a necessidade de reenviar pacotes periodicamente.

Se um ambiente virtual utilizasse protocolos de comunicação orientados à conexão, como o TCP que envia a confirmação de recebimento ou não de mensagens, ele asseguraria a sua confiabilidade, porém, dependendo do modo com que é aplicado, ele nem sempre é a opção que garantirá a melhor qualidade do sistema distribuído. Segundo [Macedonia e Zyda, 1997] o atraso causado pelo envio da confirmação de recebimento dos dados prejudica o

desempenho de simulações em tempo real, e dependendo da estratégia de desenvolvimento do sistema seria melhor usar o protocolo não orientado a conexão UDP. Ele é um protocolo onde a troca de mensagens não é confiável, pois não garante a entrega dos *datagramas*, a eliminação de duplicados, e a ordem de entrega. Contudo, como não há um controle de recebimento e de erro, a entrega das mensagens é agilizada, sendo esse um fator essencial em ambientes com interações em tempo real.

Porém, outros fatores podem influenciar na escolha do protocolo a ser usado. Por exemplo, o *Socket* TCP normalmente é utilizado em sistemas síncronos, e o UDP para assíncronos. Este fato ocorre devido exatamente às estruturas de trabalho dos *Sockets*, o TCP garante a entrega dos dados de forma ordenada e isso garante que os sistemas comunicantes funcionem de maneira compassada. O *Socket* UDP por sua vez, em virtude da possível perda de informações propicia fortemente o assincronismo dos sistemas. Uma estratégia comumente utilizada para sincronizar sistemas UDP é a aplicação de *Threads*. Deste modo, pode-se concluir que programadores que optaram pelo uso de *Sockets* para desenvolver sistemas que devem trabalhar sincronicamente, o ideal é utilizar *Sockets* TCP ou UDP sincronizados por *Threads*.

A arquitetura de distribuição projetada para o LVD exige que os sistemas comunicantes trabalhem sincronicamente. Neste contexto, optou-se por utilizar o recurso *Socket* TCP da linguagem Java para desenvolver o sistema de distribuição do LVD.

Para melhorar o desempenho do sistema realiza-se a conexão e o fechamento dos *Sockets* apenas uma vez, quando se inicia e termina a comunicação dos sistemas, respectivamente. Para estabelecer a conexão, inicialmente, o servidor começa a “escutar” uma determinada porta, o cliente tenta a conexão nesta porta, o servidor aceita a conexão e envia uma mensagem de sucesso para o cliente. Após este processo a conexão estará estabelecida e o fluxo de dados pode iniciar. O processo de conexão é relativamente lento, por isso faz-se necessário que seja feito uma única vez. Estabelecida a conexão, o fluxo de dados é liberado, sendo ele um processo rápido e seguro.

5.3 Implementação do sistema local do LVD

Nesta seção são abordados os detalhes de desenvolvimento do sistema local do LVD. Primeiramente são apresentados os objetos virtuais que constituem o laboratório virtual e discutidos os detalhes de implementação de cada um deles. Posteriormente é discutida a

implementação do centro de atualização e, por fim, o laboratório virtual, composto dos objetos virtuais de aprendizagem, será apresentado.

5.3.1 Desenvolvimento dos objetos virtuais de aprendizagem

Os objetos virtuais, que compõem o laboratório virtual, por padrão, são constituídos basicamente por um componente *script*, que permite a programação em JavaScript do modelo comportamental do objeto, e um componente Proto, que possibilita a definição de uma classe VRML (ou X3D). Por meio deste Proto são criadas as modelagens tridimensionais dos objetos e inseridas suas propriedades físicas, como, por exemplo, o valor de sua massa, a sua temperatura, calor específico densidade, taxa de transmissão de calor etc. É importante ressaltar que alguns destes parâmetros são baseados em propriedades de objetos reais, como, por exemplo, a massa de um objeto, outros são baseados em valores obtidos em experimentos reais, como, por exemplo, a taxa de transmissão de calor de uma fonte de aquecimento. O *script* do modelo comportamental é executado infinitamente de forma cíclica, este fato decorre da necessidade de realizar atualizações na modelagem tridimensional independentemente do disparo de eventos pelo usuário. A Figura 5.2 ilustra, esquematicamente, os componentes de um objeto virtual.

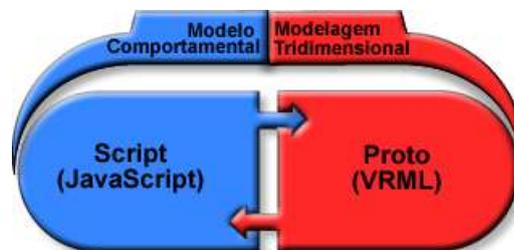


Figura 5.2: Esquema de implementação dos objetos virtuais de aprendizagem.

Fundamentalmente, cada objeto virtual de aprendizagem possui características particulares, porém, são todos construídos com base em um padrão, considerando este fato, o diagrama de classes apresentado na Figura 5.3 propõe uma generalização para o conjunto de classes a ser criado para cada objeto de aprendizagem.

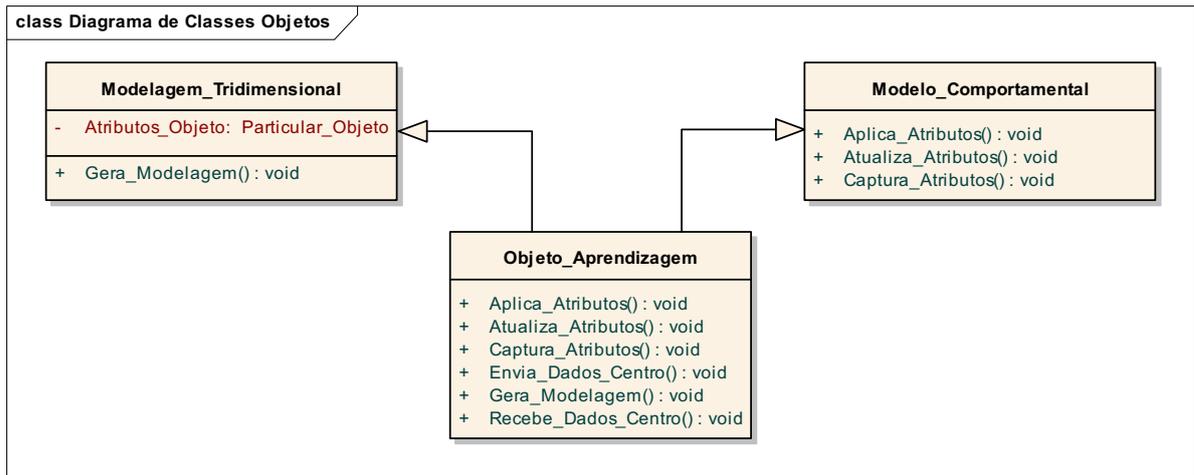


Figura 5.3: Diagrama de classes dos objetos.

A classe *Modelagem_Tridimensional* gera a forma do objeto de acordo com os atributos desejados. A classe *Modelo_Comportamental* é composta essencialmente de métodos, os quais resumem-se a: *Captura_Atributos()*, que tem como função capturar os atributos da modelagem tridimensional; *Atualiza_Atributos()*, permite redefinir os atributos de acordo com parâmetros físicos; *Aplica_Atributos()*, ela possibilita aplicar valores a modelagem tridimensional.

A classe *Objeto_Aprendizagem* herda as classes *Modelagem_Tridimensional* e *Modelo_Comportamental*, e possui dois novos métodos o *Envia_Dados_Centro()* e o *Recebe_Dados_Centro*, eles permitem, respectivamente, enviar e receber dados do centro de atualização.

De acordo com a necessidade de desenvolvimento, os objetos virtuais podem ser declarados de uma mesma classe ou não. Por exemplo, o bico de Bunsen é objeto da *Bico_Bunsen*, já os oito pesos usados nos experimentos com trilho de ar são todos objetos da classe *Peso_Trilho*. Mas, é importante ressaltar que, apesar das classes *Bico_Bunsen* e *Peso_Trilho* serem diferentes e suas declarações resultarem em objetos distintos, elas obedecem o padrão da classe *Objeto_Aprendizagem* (o mesmo modelo de composição, mesmos métodos, etc.), como ilustra a Figura 5.4.

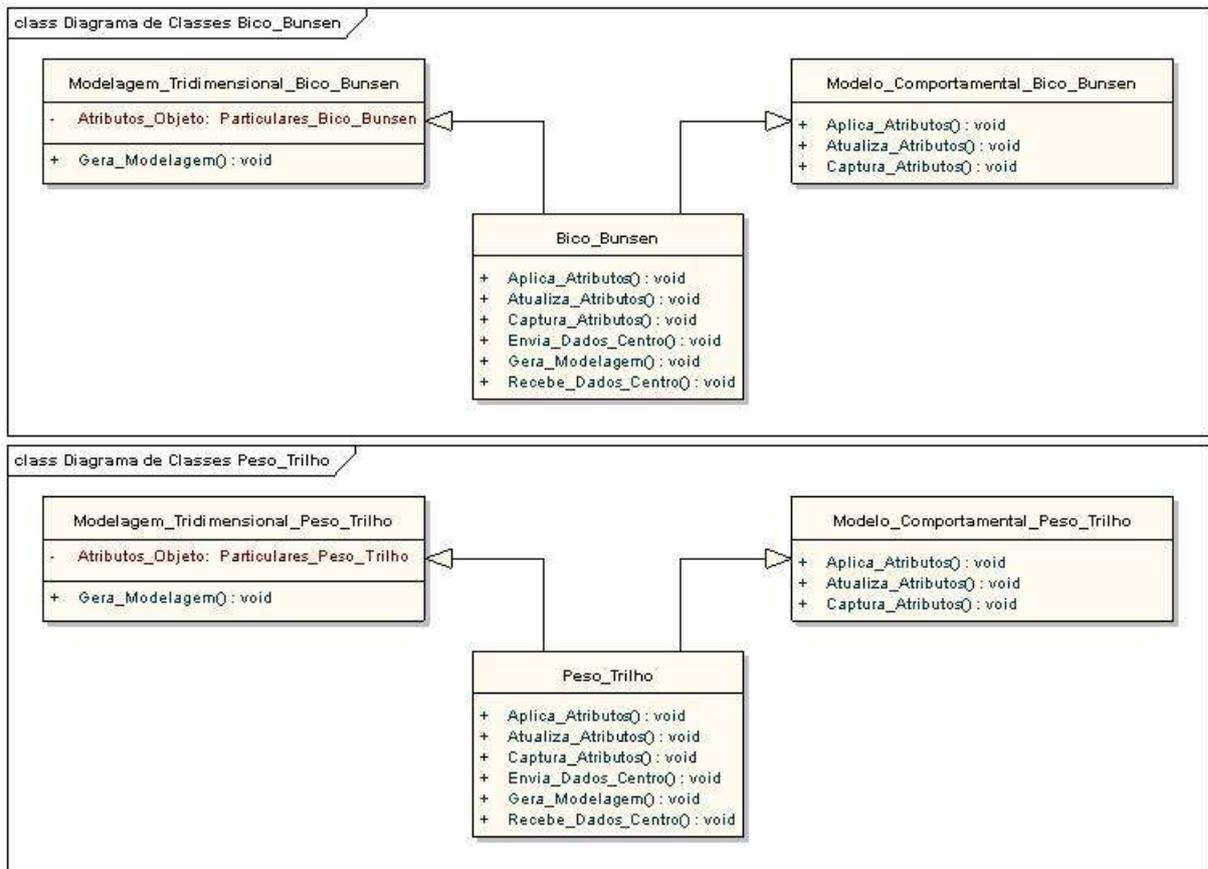


Figura 5.4. Diagramas da classe *Bico_Bunsen* e *Peso_Trilho*.

O fato de ambas as classes *Bico_Bunsen* e *Peso_Trilho* seguirem o padrão da classe *Objeto_Aprendizagem* e resultarem objetos virtuais completamente diferentes, ocorre, principalmente, devido as particularidades dos atributos de suas modelagens tridimensionais e a distinção de seus comportamentos. Esta diferença de comportamentos decorre computacionalmente do desenvolvimento da classe *Modelo_Comportamental*, em especial do método *Atualiza_Atributos()*, pois é ele que permiti a inserção da equações físicas e particularização do modo de agir de cada objeto virtual.

Foram construídos diversos objetos virtuais de aprendizagem de tipos e funcionalidades diferentes. Estes objetos podem, por exemplo, ter a função de realizar medições, como é caso da balança de precisão, do cronômetro, da trena digital e do termômetro, que medem respectivamente, peso, tempo, distância e temperatura. Por meio destas grandezas é possível realizar cálculos, de acordo com equações físicas, e deduzir novas grandezas físicas. Para facilitar estes cálculos propõe-se uma calculadora virtual.

De acordo com a funcionalidade e o desempenho de cada objeto de aprendizagem no cenário virtual foi possível separá-los em grupos. Este agrupamento tem como intuito estabelecer um padrão desenvolvimento dos modelos comportamentais dos objetos de cada

grupo. Esta padronização tem como intuito possibilitar a criação de um modelo de interação computacional entre os objetos virtuais de aprendizagem. As principais vantagens desta estratégia são tornar modular o desenvolvimento dos modelos comportamentais e, após a compreensão do modelo interação, minimizar o tráfego na rede utilizando ao máximo o processamento do sistema local.

Os objetos que possuem a função de realizar medições (balança de precisão, cronômetros, trena digital e termômetro) foram colocados em um grupo denominado *Objetos de Medição*. Computacionalmente, estes objetos obedecem ao padrão de receber um determinado dado e mostrar o seu valor para o usuário de alguma maneira. Os objetos que possuem apenas a função de auxiliar o usuário na sua atividade de resolver problemas, como é o caso da calculadora virtual, fazem parte do grupo chamado *Objetos Auxiliares*. Estes objetos possuem comportamentos isolados de outros objetos, ou seja, não realizam trocas de informação com eles.

Existe ainda o grupo dos *Objetos Processadores*, que contém os objetos de aprendizagem tais como: o becker com líquido, as molas, um tubo transparente dotado com um pistão preenchido com gás (que será chamado neste trabalho apenas por pistão) e o trilho de ar. O becker consiste em um recipiente que pode portar algum tipo de líquido cujas propriedades, como cor, massa, volume, densidade e temperatura podem ser alteradas. As molas podem sofrer deformações mediante a adição de pesos. O pistão a alteração da temperatura, volume e pressão do gás, de acordo com os pesos que são adicionados a ele. O trilho de ar permite o movimento de um carrinho praticamente sem atrito. Também de acordo com a adição de pesos o carrinho pode mover-se com maior ou menor velocidade dependendo da situação. Os objetos deste grupo são caracterizados por receber um determinado conjunto de dados, e realizar processamentos computacionais, de acordo com equações físicas e os dados recebidos, para atualizar e enviar alguns dados predefinidos.

Outro grupo objetos que foi definido foram os *Objetos Transmissores*, este grupo é composto pelo bico de Bunsen, calorímetro, máquina de líquidos e pesos. O bico de Bunsen fornece uma fonte de calor que realiza alterações na temperatura de outros objetos, o calorímetro, por sua vez, evita que trocas de calor com o ambiente ocorram ou sejam minimizadas. A máquina de líquidos fornece uma quantidade, escolhida pelo usuário, de um líquido, pré-selecionado por ele. Os pesos realizam diferentes tipos de alterações em diversos objetos, como, por exemplo, a deformação em molas, a modificação na velocidade do

carrinho do trilho de ar, etc. O padrão deste grupo de objetos é sofrer uma alteração direta do usuário e enviar dados de acordo com esta alteração.

Neste contexto, o modelo de interação dos objetos virtuais de aprendizagem de acordo com os grupos que os contém (*Objetos de Medição, Auxiliares, Processadores e Transmissores*), encontra-se ilustrado na Figura 5.5. É importante ressaltar que toda troca de informações entre os objetos é feita por meio do centro de atualização.

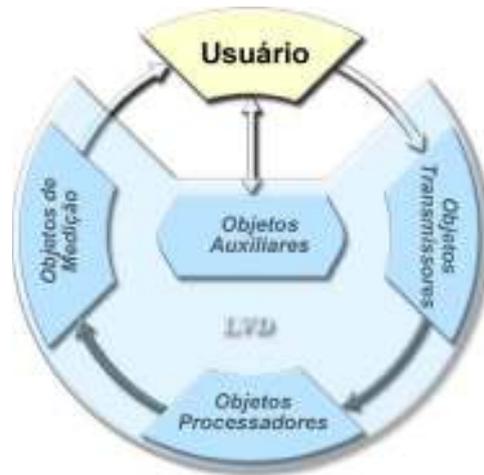


Figura 5.5: Modelo de interação dos objetos virtuais.

Em suma, o usuário manipula os *Objetos Transmissores* para causar alterações físicas nos *Objetos Processadores*. Estes, por sua vez, realizam os cálculos necessários e deixam determinados dados prontos para serem lidos pelo usuário por meio dos *Objetos de Medição*. Os *Objetos Auxiliares* são acessados somente pelo usuário.

Pode-se perceber que em uma transmissão em rede, seria uma melhor opção transmitir apenas os dados dos *Objetos Transmissores*, pois com estes dados os *Objetos Processadores* poderia realizar os cálculos físicos localmente, assim como os *Objetos de Medição* poderiam realizar as leituras de dados já atualizados também de maneira local. Tal estratégia reduz o tráfego de dados na rede, pois se torna necessário apenas o envio de dados dos *Objetos Transmissores*. E uma vez que estes dados sejam recebidos, a atualização do ambiente virtual é toda local, com o processamento dos cálculos físicos e de possíveis medições todo feito no computador que os recebeu.

Cada objeto construído será apresentado com maiores detalhes nas próximas seções.

5.3.1.1 Balança de precisão

A balança de precisão é um equipamento eletrônico que possibilita medir o peso de objetos físicos com alta precisão. A balança construída permite verificar medições até

miligramas. Porém, assim como em uma balança real, ela pode cometer um erro na terceira casa decimal, denominado erro mecânico. Este erro ocorre de acordo com a posição que o objeto foi colocado no prato da balança. É importante ressaltar que este erro foi programado na balança virtual, para proporcionar ao usuário um ambiente virtual com maior fidelidade ao ambiente real. Deste modo, foram mantidos os mesmos erros de experimentação que ocorrem em um laboratório real. Esta estratégia é importante para demonstrar para o aluno que existem diferenças entre a solução de um problema físico por meio das equações físicas estudadas na teoria, que são desenvolvidas para ambientes em condições ideais, e a solução do problema por meio de testes no meio físico real, que nem sempre está em condições ideais. A Figura 5.6 apresenta um catálogo de ilustrações da balança de precisão virtual.



Figura 5.6: Catálogo de ilustrações da balança de precisão.

Uma função importante na balança de precisão é a tara, que armazena o valor de uma medição e o desconta em medições posteriores. Esta tara pode ser zerada a qualquer momento bastando o usuário acionar o seu botão quando a balança estiver vazia.

Em termos técnicos a balança apenas recebe o campo massa (M) do objeto a ser pesado, atribui um erro (E) devido a sua localização, e verifica existência de uma tara (T). Por meio de uma *string*, um campo de saída, ela exhibe o cálculo $M + E - T$ para o usuário. Este campo de saída não sofre alterações nem por parte do usuário e nem de outros objetos.

5.3.1.2 Becker com líquido

O becker é um tipo de recipiente utilizado nos laboratórios de Física ou Química. Feito de vidro pyrex refratário ou de polímeros como o polietileno ou o polipropileno, ele pode ser utilizado em uma ampla faixa de temperatura. Suas capacidades volumétricas mais comuns

são 80, 125, 250 e 400 ml. O becker virtual desenvolvido possui a capacidade de 250 mL, como ilustra a Figura 5.7.



Figura 5.7: Catálogo de ilustrações do becker com líquido.

O becker é usado no laboratório virtual, fundamentalmente, para transportar líquidos. Ele pode ser aquecido, permitindo, desta forma, variações na temperatura do líquido transportado. Ele possui ainda uma escala de graduação que permite analisar o volume deste líquido.

Tecnicamente, cada propriedade do líquido, como massa, coloração, volume, densidade, temperatura, etc., é um campo do nó Proto que define este líquido. O becker recebe dados de outros objetos, como o bico de Bunsen, o calorímetro e a máquina de líquidos, e atualiza suas propriedades de acordo com equações físicas. Por exemplo, uma vez colocado no bico de Bunsen ele recebe um valor correspondente a uma taxa de aquecimento (quantidade de calor por segundo), e então, com base neste valor, ele recalcula a temperatura, volume e densidade do líquido. Todas as alterações realizadas nestes dados podem ser sentidas pelo usuário por meio das atualizações na modelagem tridimensional. Um procedimento similar ocorre quando o becker é colocado no interior do calorímetro.

Ao ser colocado na máquina de líquidos o becker recebe as propriedades do líquido fornecido por ela e uma taxa de inserção ou remoção de volume por segundo. O becker calcula então as novas propriedades físicas do composto final, como, por exemplo, a sua nova densidade, os seus novos pontos de ebulição e fusão, o seu novo calor específico, etc. Este cálculo é realizado com base nas propriedades dos componentes primários da mistura e de suas respectivas proporções volumétricas.

Uma importante propriedade da mistura final é a sua coloração, pois está diretamente ligada à visão do usuário. Neste contexto foram pesquisadas técnicas de mistura de texturas de acordo com o ambiente real. A Figura 5.8 ilustra como seria a reação da coloração quando são

misturados 125 ml de um líquido verde com a mesma quantidade de um líquido vermelho. Percebe-se que assim como no ambiente real o resultado da mistura seria 250 ml de um líquido amarelo.

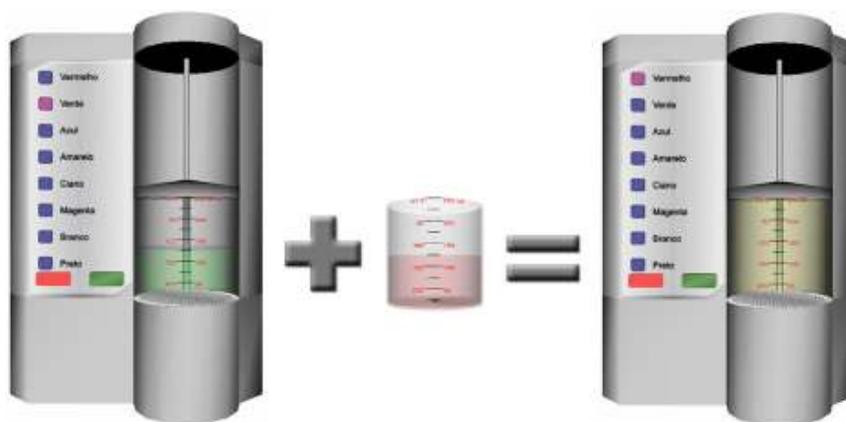


Figura 5.8: Mistura de coloração de líquidos.

Para possibilitar que esta mistura de coloração de líquidos ocorra como no ambiente real, considerou-se que cada líquido possui cor devido a uma pigmentação. Pigmentos coloridos atuam como seletores ou filtros de luz. Este processo compõe o sistema de cores chamado CMY (*ciano, magenta, yellow*).

A mistura de dois pigmentos ocorre por meio de uma síntese subtrativa, ou seja, cada tipo de pigmento tem seu próprio poder seletor, que absorve (subtrai) um ou mais tipos de radiações de luzes. A cada sobreposição de um pigmento, diminui o número de radiações refletidas, até conseguir a ausência absoluta de toda radiação, isto é, a sensação de preto, fim da mistura subtrativa.

As cores básicas da mescla subtrativa são o amarelo, o ciano e o magenta, também chamadas de cores primárias. Essa escolha se deve ao fato de que o pigmento de cada uma das três cores não é o resultado da combinação de outros. Pelo contrário, da mistura desses pigmentos, de dois em dois ou de três em três, em porções oportunas, pode-se obter uma vastíssima gama de outras tonalidades.

O resultado da mistura de duas cores primárias é uma nova cor denominada secundária. Por exemplo, mesclando o ciano e o amarelo em ponto de saturação certa, visualiza-se o verde, uma cor secundária. E a mistura do amarelo e do magenta dá o vermelho e do magenta com o ciano dá o azul-violeta, ambas são consideradas como cores secundárias.

A Figura 5.9 apresenta um círculo cromático, usado para visualizar a mistura de cores.

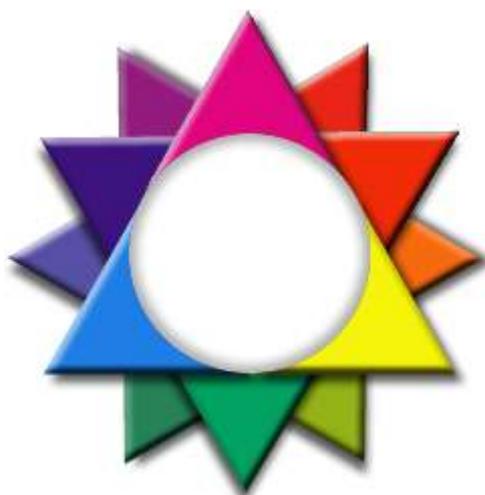


Figura 5.9: Círculo cromático.

A mistura de uma cor primária e uma secundária, vizinhas no círculo cromático, produz uma cor terciária. São seis: amarelo-esverdeado, verde-azulado, azul-violeta, vermelho-violeta, vermelho-magenta, magenta-laranja.

Porém, normalmente os programas computacionais trabalham internamente com um sistema de cores denominado RGB (*Red, Green, Blue*), e não com o CMY. Este sistema é usado para tratar luzes coloridas, nas quais monitores de computadores baseiam o seu funcionamento, e não matérias coloridas. Projetando-se em uma tela luzes com comprimentos de ondas adequadas, de modo que suas cores sejam vermelha, verde e violeta, aparecerá uma luz branca no espaço onde houver a superposição dessas três cores. Deste modo, pode-se perceber que o sistema RGB trabalha com uma síntese aditiva, ao contrário da composição de cores de pigmentos do sistema CMY, que utiliza uma síntese subtrativa.

Neste contexto foi necessária a criação de um algoritmo que identifique a coloração de dois líquidos virtuais no sistema RGB, com suas respectivas cores no sistema CMY. Posteriormente, o algoritmo deve misturar estas cores, proporcionalmente à quantidade de cada líquido, por meio da síntese subtrativa do sistema CMY. E por fim, deve aplicar a cor resultante ao líquido no padrão RGB, para que o usuário possa visualizá-la em seu monitor. O trecho de código apresentada na Figura 5.10 ilustra as bases do algoritmo desenvolvido.

```

:
:
// Abaixo é feito o cálculo proporção de líquido inserida, com
// base na quantidade de líquido final (H) e a inicial (HInicial).
Proporcao = (H - HInicial)/(HInicial)

// Computacionalmente uma cor no sistema RGB corresponde a um
// vetor de três posições, preenchidas com números reais que
// variam de 0 a 1, identificando a intensidade das cores
// vermelha, verde e azul. Abaixo estes valores são calculados
// separadamente.

// Primeiramente é classificado o tipo de mistura a ser feita,
// mistura de duas cores primárias, secundárias, primária com
// secundária, etc. O tipo da mistura de determina o valor das
// variáveis operador (podendo ser +1 ou -1) e CMY (podendo ser 0
// ou 1), para que esta mistura seja feita de forma correta.

// Abaixo é feito o cálculo da intensidade da cor vermelha, ou
// seja, a primeira coordenada do vetor RGB.
Cor_Final[0] = CMY + operador * (CMY + operador * Cor_Inicial[0]
+ (CMY + Operador * Cor_Enchimento[0]) * Proporcao)
if(Cor_Final[0]<=0){Cor_Final[0]=0}
if(Cor_Final[0]>=1){Cor_Final[0]=1}

// Abaixo é feito o cálculo da intensidade da cor verde, ou seja, a
// segunda coordenada do vetor RGB.
Cor_Final[1] = CMY + Operador * (CMY + Operador * Cor_Inicial[1]
+ (CMY + Operador * Cor_Enchimento[1]) * Proporcao)
if(Cor_Final[1]<=0){Cor_Final[1]=0}
if(Cor_Final[1]>=1){Cor_Final[1]=1}

// Abaixo é feito o cálculo da intensidade da cor azul, ou seja, a
// terceira coordenada do vetor RGB.
Cor_Final[2] = CMY + operador * (CMY + operador * Cor_Inicial[2]
+ (CMY + Operador * Cor_Enchimento[2]) * Proporcao)
if(Cor_Final[2]<=0){Cor_Final[2]=0}
if(Cor_Final[2]>=1){Cor_Final[2]=1}

// E por fim, a cor encontrada (Cor_Final) é aplicada no objeto
// virtual através do comando abaixo.
CorLiq = Cor_Final

:
:

```

Figura 5.10: Código do algoritmo de mistura de coloração.

5.3.1.3 Bico de Bunsen

O aquecimento em laboratório geralmente é feito por meio de queimadores de gases combustíveis, sendo o mais simples deles o bico de Bunsen. Normalmente o bico de Bunsen é usado em laboratórios para: aquecer soluções aquosas não inflamáveis; polir a fogo vidros quebrados; secar sais hidratados; fundir amostras; aquecer sais para observar espectros emitidos.

Para aquecer recipientes de vidro, como o becker, utiliza-se uma tela de amianto sustentada por um tripé de ferro. A tela de amianto é uma tela metálica com a região central coberta de amianto. Ela é usada para produzir uma distribuição uniforme de calor durante o aquecimento com um bico de gás.

A Figura 5.11 apresenta um catálogo de ilustrações do bico de Bunsen. Nestas ilustrações pode-se observar, além do bico de Bunsen, a tela de amianto e o tripé de ferro.



Figura 5.11: Catálogo de ilustrações do bico de Bunsen.

De uma maneira geral, o gás entra no bico de Bunsen pela sua base e seu fluxo é regulado por uma torneira externa na parte inferior do bico. A quantidade de gás influencia diretamente no tamanho da chama, como ilustra a Figura 5.12.

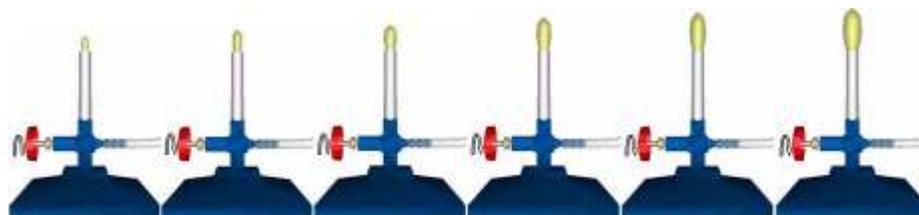


Figura 5.12: Aumento da passagem de gás no bico de Bunsen.

À medida que o gás sobe pelo tubo do bico de Bunsen é injetado ar através de orifícios situados um pouco acima da base. A quantidade de ar pode ser controlada girando-se um anel que fica sobre os orifícios. A etapa inicial para se acender o bico é fechar a entrada de ar. A seguir, deve-se abrir o gás e acende-lo posteriormente. A chama obtida apresenta uma cor amarela brilhante, ela é denominada chama "fria", resultante de uma mistura gasosa com pouco oxigênio, e configura-se como inadequada ao uso por ser pouco oxidante. Esta chama não tem uma temperatura suficiente para o aquecimento de substância alguma, para conseguir uma chama mais "quente", a entrada de ar deve ser aberta até que se consiga uma chama azulada; isto ocorre porque o oxigênio mistura-se com o gás, tornando a queima deste mais eficiente. A Figura 5.13 ilustra o aumento da passagem de ar no bico de Bunsen virtual e a sua influência na coloração da chama.

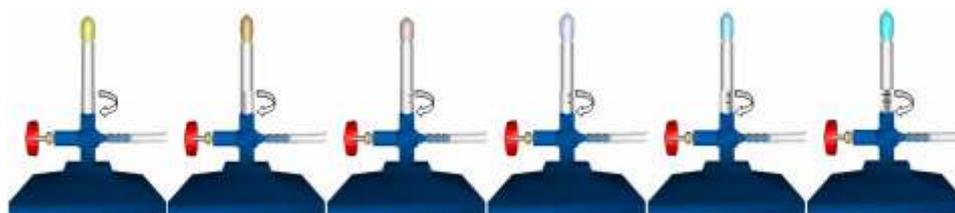


Figura 5.13: Aumento da passagem de ar no bico de Bunsen.

Ao manipular o bico de Bunsen o usuário pode receber algumas mensagens informativas relacionadas ao modo com que se deve manipular o bico. Como, por exemplo, ao retirar a tela de amianto do tripé de ferro é emitida a mensagem “Use a tela de amianto para levar o becker ao fogo”, como ilustra a Figura 5.14(a), prevenindo-o de possíveis acidentes. Também pode ser emitida a mensagem “Interrompa a passagem de ar antes de interromper ou liberar a passagem de gás”, como ilustra a Figura 5.14(b), que explica o funcionamento correto e seguro do bico.



Figura 5.14: Mensagens informativas do bico de Bunsen.

Em termos técnicos, o bico de Bunsen virtual apenas fornece como dado de saída uma taxa de aquecimento de transmissão de calor por segundo para os demais objetos, como o becker com líquido. Esta taxa é calculada de acordo com a quantidade de gás e ar liberada, ela foi obtida experimentalmente.

5.3.1.4 Calculadora científica

A calculadora científica virtual é um objeto que possui exclusivamente a função de auxiliar o usuário em seus cálculos. Ela permite cálculos de operações básicas como soma, subtração, divisão, multiplicação, potenciação e radiciação, e ainda possibilita a resolução de funções como seno, cosseno, tangente, e suas inversas, e função logarítmicas. Este objeto não se comunica com os demais objetos, ou seja, ele não recebe nem envia dados para outros objetos. Em suma, ela recebe dados do usuário, realiza os cálculos necessários com os mesmos, e por fim, apresenta os resultados por meio de uma *string*. A calculadora virtual encontra-se ilustrada na Figura 5.15.



Figura 5.15: Catálogo de ilustrações da calculadora virtual.

5.3.1.5 Calorímetro

O calorímetro é um instrumento que auxilia a realizar medições na calorimetria. Basicamente, ele é um recipiente isolante que reduz a troca de calor entre o corpo no seu interior e o ambiente externo. Deste modo é possível realizar medições sem grandes influências do ambiente. Ele possui uma abertura na sua tampa que permite a inserção de um termômetro para medir a temperatura do corpo no seu interior.

O calorímetro participa das trocas de calor no experimento, recebendo ou fornecendo calor, isto influencia diretamente nos resultados finais da experiência. Sendo assim, é necessário determinar experimentalmente a sua capacidade térmica, que permitirá o equacionamento do experimento.

Capacidade térmica é a grandeza física que determina a variação térmica de um corpo ao receber ou perder determinada quantidade de calor. O valor da capacidade térmica é correspondente à quantidade de calor necessária para elevar ou diminuir a temperatura do corpo em uma unidade de variação de temperatura.

Calorímetros podem ser considerados como ideais, supondo-se que, depois de calibrados, não ocorrem variações de temperatura. Estes calorímetros são chamados isotérmicos. Porém, no ambiente real é praticamente impossível à existência de um calorímetro que isole completamente um corpo do ambiente externo. Neste contexto pode-se considerar que a troca de calor com o ambiente ocorre de maneira linear, este tipo de calorímetro é denominado de varredura exploratória.

O calorímetro virtual construído baseia-se no calorímetro de varredura exploratória, visando uma maior aproximação do seu comportamento com um modelo real. A Figura 5.16 ilustra o calorímetro virtual desenvolvido.



Figura 5.16: Catálogo de ilustrações do calorímetro virtual.

Tecnicamente, o calorímetro fornece como dado de saída, para os objetos no seu interior, uma taxa de redução de transferência de calor por segundo. Esta taxa varia linearmente de acordo com o calorímetro real de varredura exploratória. Por meio desta taxa o objeto no seu interior calcula a sua temperatura instantaneamente e ciclicamente.

5.3.1.6 Cronômetros

Cronômetro é simplesmente um instrumento de precisão, eletrônico ou mecânico, para medir intervalos de tempo. Existem diversos tipos de cronômetros aplicados a diferentes situações. Para o laboratório virtual foram desenvolvidos dois tipos de cronômetros digitais, um de uso manual e o outro que é associado a sensores fotoelétricos.

O cronômetro manual pode ser utilizado para medir intervalos de tempos de situações aleatórias, sendo que, o próprio usuário inicia, para e reinicia a contagem de tempo no cronômetro. Este tipo de cronômetro é importante para que o usuário perceba a existência de erros em experimentos reais. Como, por exemplo, o erro de medição do tempo de um experimento. Neste tempo está embutido o tempo de reação humano de visualizar o início de uma ação e disparar o cronômetro. Este tempo de reação apesar de pequeno é contabilizado como tempo de experimento, o que provoca um determinado erro ao serem calculadas outras grandezas físicas derivadas deste tempo. Na Figura 5.17 são apresentadas imagens do cronômetro virtual, de uso manual, desenvolvido.



Figura 5.17: Catálogo de ilustrações do cronômetro.

Outro tipo de cronômetro desenvolvido foi o controlado por sensores fotoelétricos. Neste cronômetro podem ser conectados pares de sensores que enviam sinais para ele quando os seus feixes de fotoelétrons são interrompidos. Quando o primeiro sensor detecta algo o cronômetro é zerado e disparado, quando o segundo sensor realizar uma detecção o contador

de tempo será parado. O cronômetro digital com estas funções desenvolvido neste projeto encontra-se ilustrado na Figura 5.18.



Figura 5.18: Catálogo de ilustrações do cronômetro digital controlado por sensores fotoelétricos.

Ambos os cronômetros, tecnicamente, apenas disparam ou param uma variável de contagem de tempo, de acordo com algum sinal externo advindo de um usuário ou de um sensor. Por fim, esta variável é exibida na tela por meio de uma *string*.

5.3.1.7 Máquina de líquidos

A máquina de líquidos é o único objeto virtual construído que não é comumente encontrado em laboratórios de Física reais. Ela possui a capacidade de simular a inserção de líquidos pré-cadastrados em um frasco, podendo inclusive, efetuar a mistura dos mesmos. Optou-se por construir este objeto não-convencional com o intuito de evitar as simulações de transições de líquidos. Estas simulações possuem algoritmos que exigem um alto custo de processamento computacional, o que possivelmente sobrecarregaria o sistema. A máquina de líquidos virtual encontra-se ilustrada na Figura 5.19.



Figura 5.19: Catálogo de ilustrações da máquina de líquidos virtual.

Em suma, a máquina oferece como dados de saída às propriedades físicas do líquido que está sendo fornecido, como a sua coloração, a sua densidade, o seu coeficiente de dilatação, os seus pontos de ebulição e fusão, o seu calor específico e os seus calores latentes de fusão e ebulição. Ela fornece ainda, uma taxa de inserção ou remoção do volume de líquido por segundo. Todos estes dados de saída são usados pelo objeto que está no seu interior, normalmente o becker, que realiza os devidos cálculos para definir a sua nova condição física.

5.3.1.8 Mola

Em experimentos com mola, normalmente, são realizados testes quanto à sua deformação e comportamento, mediante a aplicação de forças com valores diferentes. Comumente, são usados pesos, que permitem aplicar forças com valores conhecidos, e réguas milimetradas para medir as deformações da mola. A Figura 5.20 ilustra a mola virtual desenvolvida e seus respectivos componentes auxiliares.



Figura 5.20: Catálogo de ilustrações da mola virtual.

Todo sistema mecânico possui algum grau inerente de atrito, que age como um consumidor de energia mecânica. Deste modo, um corpo que vibra, oscila com uma amplitude que gradualmente decresce e, posteriormente, pára. Neste contexto, a mola comporta-se de acordo com um oscilador subamortecido. Ele é regido pela equação diferencial $m x''(t) + b x'(t) + k x(t) = 0$, na qual m é uma massa que deforma a mola, b é o coeficiente de amortecimento, k é constante de elasticidade da mola, e x fornece a sua deformação.

Computacionalmente, o valor de m é atribuído pelos pesos, e logo após o que se faz é calcular a deformação da mola por meio da resolução desta equação. Por fim aplica-se o valor desta deformação à geometria da mola. O trecho de código ilustrado na Figura 5.21 apresenta este procedimento.

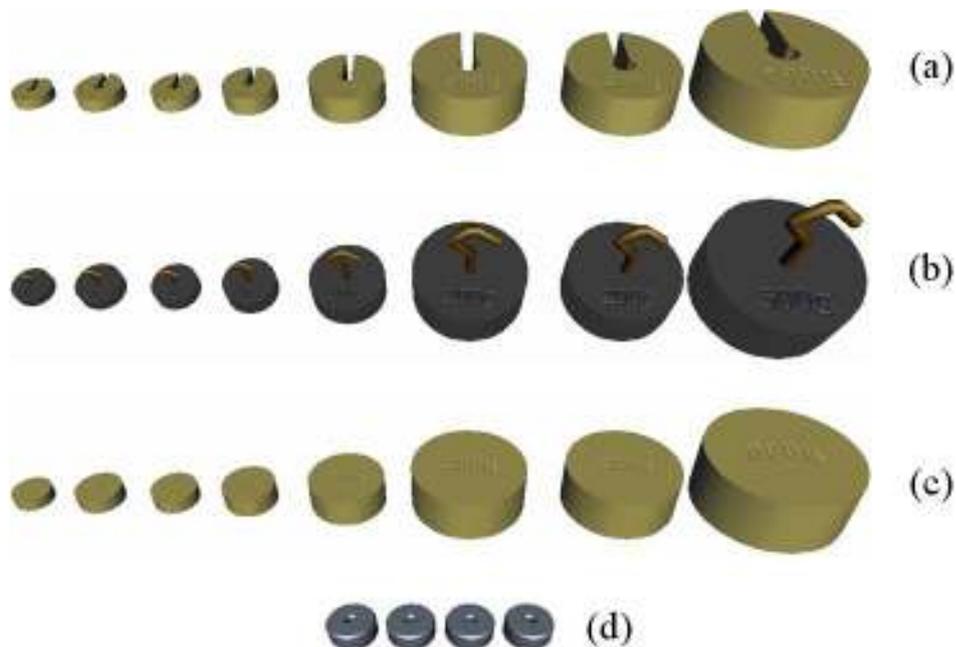


Figura 5.22: Conjunto de pesos virtuais.

Tecnicamente, os pesos possuem apenas um dado de saída: o valor de sua massa. Eles transmitem este dado quando encontram uma região de reconhecimento no laboratório virtual. Por exemplo, quando um peso encontra a superfície de compressão superior do pistão ele transmite sua massa. O pistão receberá este dado e realizará os seus devidos cálculos e respectivas atualizações.

5.3.1.10 Pistão didático

O pistão didático é um instrumento usado na Física para realizar experimentos envolvendo a compressão de gases. Estes experimentos são usados para estudar o comportamento de gases, em termos de sua pressão, volume e temperatura.

Normalmente, um pistão didático é composto de um cilindro milimetrado, que permite medir o volume de um gás no seu interior, e um êmbolo que possibilita a sua compressão ou distensão. O pistão é ligado a sensores que captam a sua variação de temperatura e a sua pressão. Estes sensores, por sua vez, são ligados a um computador, que possibilita a visualização dos valores captados. A Figura 5.23 apresenta um conjunto de ilustrações do pistão virtual desenvolvido.



Figura 5.23: Catálogo de ilustrações do pistão virtual.

O pistão sofre alterações, de volume, temperatura e pressão, por meio da aplicação de pesos, como ilustra a Figura 5.24.



Figura 5.24: Adição de pesos no pistão virtual.

Em suma, o pistão virtual recebe como dado de entrada o valor do peso que está sendo aplicado a ele. Logo após, ele realiza os devidos cálculos, de acordo com as equações físicas do comportamento de gases, e por fim, ele é atualizado de acordo com estes cálculos.

5.3.1.11 Termômetro

O termômetro é um aparelho usado para medir temperaturas ou as variações de temperatura. Existem diversos tipos de termômetros como, por exemplo, os metálicos, os digitais e os de mercúrio.

O termômetro de mercúrio é o mais comum. Ele consiste basicamente de um tubo capilar de vidro, fechado a vácuo, e um bulbo (uma espécie de bolha arredondada) em uma extremidade contendo mercúrio.

O mercúrio, como todos os materiais, dilata quando aumenta a temperatura. Por ser extremamente sensível, ele aumenta de volume à menor variação de temperatura. O volume do mercúrio aquecido se expande no tubo capilar do termômetro. E essa expansão é medida pela variação do comprimento, numa escala graduada que pode ter uma precisão de $0,05^{\circ}\text{C}$. É dessa forma, pela expansão do líquido, que a variação da temperatura pode ser observada.

Para o laboratório virtual desenvolveu-se o termômetro de mercúrio, como ilustra a Figura 5.25. O usuário pode escolher entre as escalas Celsius, Fahrenheit, Kelvin para realizar suas medições.

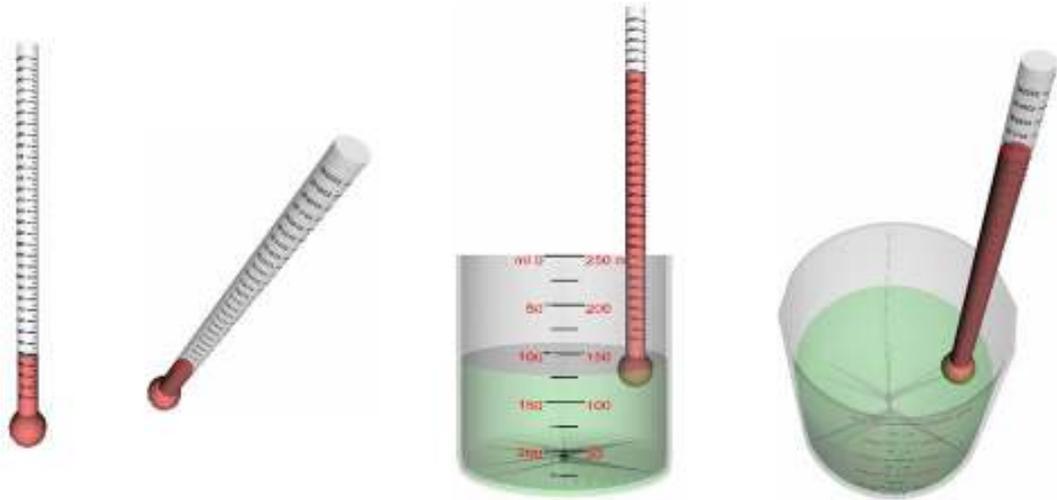


Figura 5.25: Catálogo de ilustrações do termômetro virtual.

Tecnicamente o termômetro virtual, ao reconhecer um objeto com propriedades térmicas, recebe como dado de entrada o valor da temperatura deste objeto e realiza suas devidas atualizações, que se resume a dilatar a barra de mercúrio proporcionalmente a temperatura recebida.

5.3.1.12 Trena digital

Trena é um instrumento usado para medir distâncias entre dois pontos quaisquer. Existem dois tipos de trena a mecânica e a digital. Para o laboratório virtual optou-se por desenvolver a trena digital, como ilustra a Figura 5.26.



Figura 5.26: Catálogo de ilustrações da trena digital virtual.

De maneira técnica, a trena virtual permite que o usuário escolha dois pontos distintos do espaço, como, por exemplo, ponto $A(x_A, y_A, z_A)$ e ponto $B(x_B, y_B, z_B)$. Então o sistema

calcula o valor da distância (d) entre estes pontos, por meio da equação $d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$. E logo após, este valor d é exibido por meio de uma *string*.

5.3.1.13 Trilho de ar

O trilho de ar é um sistema que permite estudar movimentos unidimensionais reduzindo drasticamente as forças de atrito habitualmente presentes. Ele é composto de chapas metálicas de perfil reto, com pequenos orifícios regularmente espaçados nas faces laterais. Injeta-se ar comprimido dentro do trilho. Este ar sai através dos orifícios, gerando um colchão de ar entre o trilho e o carrinho (objeto colocado sobre o trilho para fazer as experiências) de cerca de meio milímetro de espessura. Este colchão de ar faz com que o carrinho “flutue”, provocando uma grande redução do atrito. O único atrito existente no experimento, denominado residual, é devido basicamente à fricção com o meio gasoso.

O trilho de ar permite a execução de uma grande variedade de experimentos, como, por exemplo, experiências envolvendo as leis de Newton. Normalmente estes experimentos ocorrem por meio da adição de pesos. Podem-se adicionar pesos ao carrinho e a um suporte que é ligado a ele através de um fio.

O trilho de ar virtual desenvolvido e seus complementos encontram-se ilustrados na Figura 5.27.

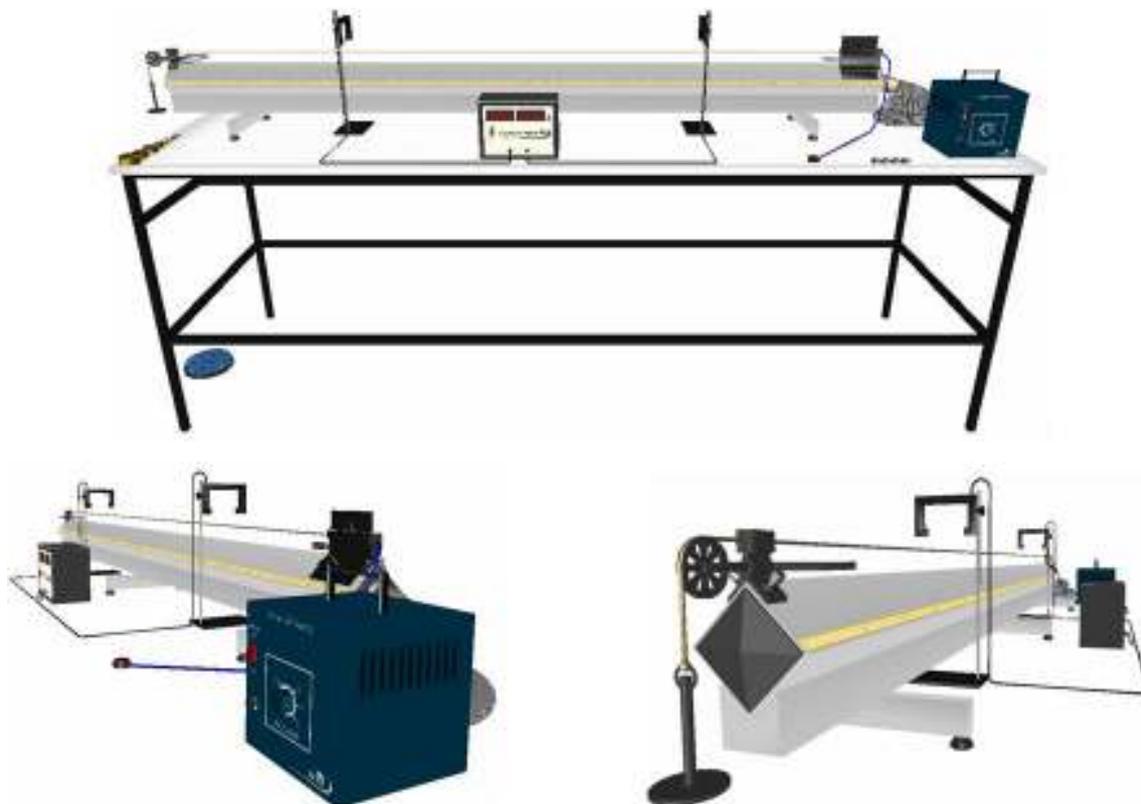


Figura 5.27: Catálogo de ilustrações do trilho de ar virtual.

O trilho de ar necessita de um conjunto de componentes e acessórios para o seu funcionamento correto em alguns experimentos como, por exemplo:

- o gerador de fluxo de ar, ilustrado na Figura 5.28(a), ele é ligado a uma das extremidades do trilho por uma mangueira, e possui a função de criar o fluxo de ar dentro do tubo e permitir que o ar, ao sair pelos orifícios do trilho, crie o "colchão de ar";
- o carrinho, apresentado na Figura 5.28(b), é o elemento a ser lançado sobre o trilho de ar para análise de seus respectivos movimentos. Basicamente, o carrinho consiste da união de três placas plásticas por um lado comum, de modo que sua base pudesse ser perfeitamente encaixada na base do trilho;
- o disparador magnético, ilustrado na Figura 5.28(c), consiste de uma peça metálica que, por meio de um fluxo magnético, pode segurar o carrinho enquanto são adicionados os pesos. Ao término da adição de pesos, o fluxo magnético pode ser interrompido, por intermédio de um botão ligado a peça metálica através de um fio, esta interrupção provoca a liberação do carrinho;
- o amortecedor de choques e roldana, apresentados na Figura 5.28(d), o amortecedor consiste de uma junta acoplada a um elástico que possui a função

de reduzir o impacto do carrinho, a roldana possui a função de acomodar o fio que liga o carrinho ao suporte de pesos;

- o sensor fotoelétrico, ilustrado na Figura 5.28(e), possui a função de emitir um sinal quando o seu feixe de fotoelétrons é interrompido pelo carrinho;
- o cronômetro digital eletrônico, apresentado na Figura 5.28(f), como já mencionado na seção 5.3.1.6, possui a função de marcar o tempo entre a interrupção do primeiro sensor fotoelétrico e o segundo.

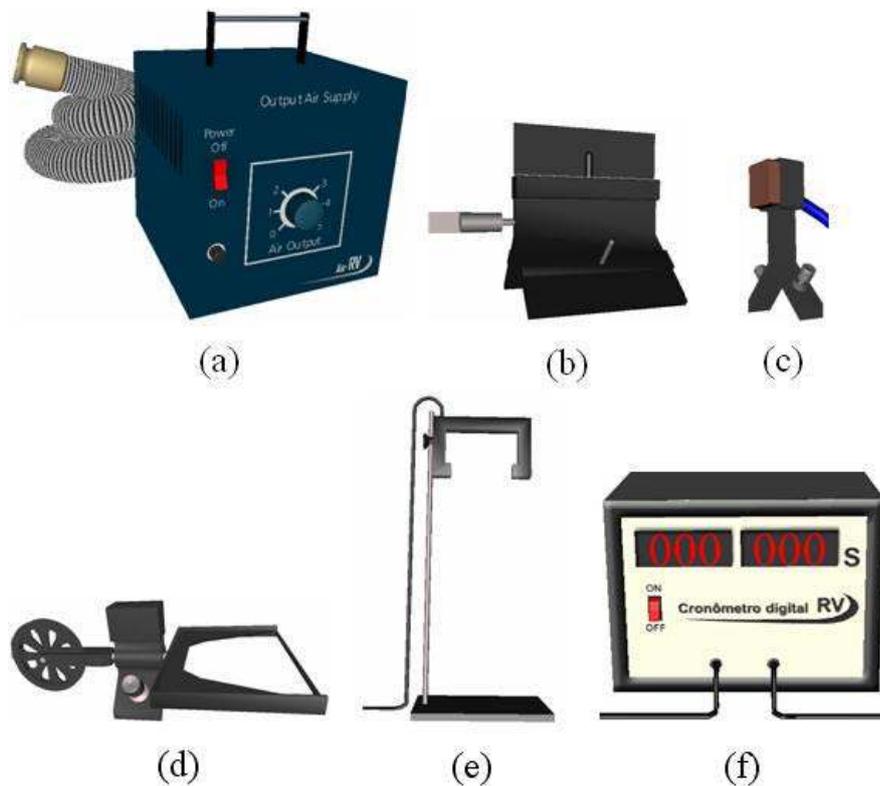


Figura 5.28: Componentes do trilho de ar virtual.

Existe ainda uma fita métrica acoplada ao longo do trilho, que realiza o papel de sistema de referência do movimento, de maneira que a posição do carrinho em qualquer instante pode ser marcada pela fita.

Tecnicamente, o trilho de ar recebe como dados de entrada os valores das massas dos pesos atribuídos ao carrinho e ao suporte de pesos. Após receber estes dados o sistema do trilho virtual desempenha um comportamento de acordo com equações físicas.

5.3.2 Centro de atualização

O centro de atualização é um conjunto de métodos, como ilustra a Figura 5.29, que possibilita a troca de informações entre as entidades internas do ambiente computacional

LVD. Por exemplo, ele pode realizar troca de dados entre objetos virtuais de aprendizagem, e entre um objeto e o Transmissor. Assim como nos modelos comportamentais dos objetos, o centro de atualização também é uma estrutura executada ciclicamente de modo infinito. Este fato decorre da necessidade de realizar trocas de informações independentemente do disparo de eventos pelo usuário.

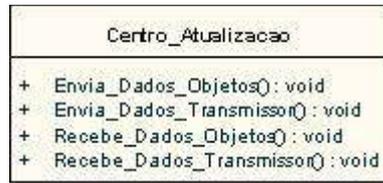


Figura 5.29: Centro de atualização.

Esta estrutura tem como intuito propiciar uma maior modularidade e escalabilidade no uso de objetos virtuais no LVD. Utilizando o centro de atualização e os objetos de aprendizagem desenvolvidos criou-se uma interface para o laboratório, que é apresentada na próxima seção.

5.3.3 Interface do LVD

A interface do laboratório virtual foi desenvolvida visando facilitar e tornar intuitivo o uso dos objetos virtuais de aprendizagem. Neste contexto, buscou preconizar um aspecto organizacional ergonômico e harmônico para estes objetos. Elaborou-se ainda uma estrutura físico-arquitetônica (paredes, teto, lâmpadas, bancos, etc.) com o intuito de aproximar a interface virtual do LVD a um laboratório real. A Figura 5.30 apresenta a interface criada.



Figura 5.30: Interface do LVD.

Com intuito de proporcionar uma melhor navegação no ambiente virtual foram pré-fixadas diversas câmeras, como ilustra a Figura 5.31.

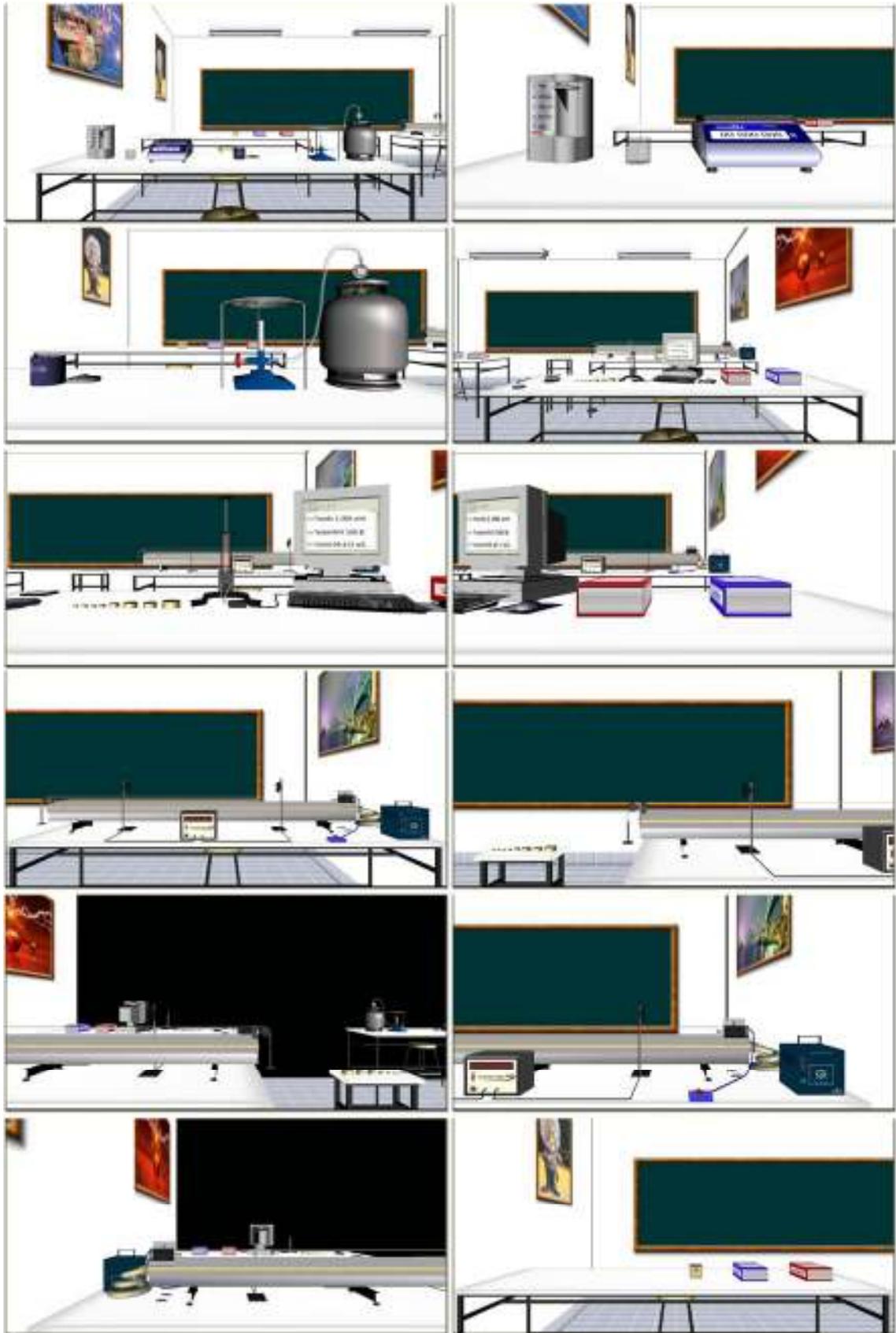


Figura 5.31: Câmeras do LVD.

O sistema do LVD discutido até o presente momento trabalha localmente. Para possibilitar a sua distribuição foram elaboradas estratégias computacionais, que preservam a

estrutura local desenvolvida e permitem a comunicação entre ambientes distintos e em máquinas diferentes. Estas estratégias e técnicas de implementação usadas para desenvolver o sistema de distribuição do LVD são apresentadas na próxima seção.

5.4 Implementação do sistema de distribuição do LVD

Nesta seção são abordados os detalhes de desenvolvimento do sistema de distribuição do LVD. Em suma, são apresentados os componentes que possibilitam a distribuição e discutidas as suas respectivas funcionalidades junto ao sistema. Apresentam-se ainda os algoritmos criados para atingir uma maior eficácia na transmissão de dados.

5.4.1 Cliente e Servidor Java TCP

Os *sockets*, da linguagem Java, são criados por um conjunto de classes do pacote *java.net*. Na conexão TCP, o cliente e o servidor são oriundos de classes diferentes. O servidor usa a classe *ServerSocket* para “escutar” uma porta de rede da máquina a espera de requisições de conexão, sendo assim, o seu construtor possui como um de seus parâmetros o número desta porta.

O cliente, por sua vez, deve conhecer em qual máquina o servidor está sendo executado e a porta a qual ele está escutando. Neste contexto, o cliente utiliza a classe *Socket* para requisitar conexão a um servidor específico e então transmitir dados. O seu construtor tem como parâmetros o endereço IP do servidor e a porta escutada.

O servidor usa o método *accept()* para bloquear a execução e aguardar até que um cliente requisição conexão. Uma vez recebida a conexão o servidor responde ao cliente, e então é criado um fluxo de conexão.

No modelo de fluxos (*streams*) dos *sockets* utilizam-se dois fluxos: um de entrada e outro de saída, criados a partir das classes *InputStream* e *OutputStream*, respectivamente. Um processo envia dados a outro quando escreve em um fluxo de saída associado a um *socket*. E um processo recebe dados escritos por outro processo lendo um fluxo de entrada associado a um *socket*.

Para encerrar a conexão tanto cliente quanto servidor utilizam o método *close()*. A Figura 5.32 ilustra os processos de estabelecimento de conexão, troca de informação e encerramento de conexão em *sockets* cliente e servidor.



Figura 5.32: Sockets cliente e Servidor.

Considerando o funcionamento dos *sockets* TCP implementou-se os serviços do cliente e servidor para o LVD. Para tanto, foram desenvolvidas duas classes denominadas *Client* e *Server*, que são responsáveis pelas etapas de estabelecimento e encerramento da conexão do cliente e o servidor, respectivamente. Estas classes também possuem a função de gerar uma interface gráfica para que o usuário realize estas tarefas.

A Figura 5.33 apresenta a interface gerada pela classe *Server*.



Figura 5.33: Interface de estabelecimento e encerramento de conexão do servidor.

Por meio da interface *Server* o usuário pode executar o servidor em uma determinada máquina, clicando no botão “*Start*”. Esta ação coloca o servidor em espera por uma requisição de conexão de um cliente. Pode-se observar que o servidor deve ser executado antes de do cliente para aguardar a esta requisição.

A interface apresenta ainda uma barra de status, que oferece as informações: “*Waiting...*”, que indica a espera de uma ação do usuário ou de uma requisição de conexão de um cliente; “*Conected*”, aponta que o servidor está conectado a algum cliente; “*Disconnected*”, indica que a conexão foi rompida pelo cliente.

Caso o usuário queira encerrar o servidor basta fechar a interface *Server* clicando no botão de fechamento da janela. A Figura 5.34 apresenta o trecho de código que implementa a classe *Server*.

```

class Server extends JFrame {

    public static ObjectOutputStream output;
    public static ObjectInputStream input;
    public static ServerSocket server;
    public static Socket connection;
    private JButton start;
    private JLabel label1;
    public static JLabel label;

    public Server(){

        //Criando a interface
        super("Server");
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        label1 = new JLabel("Execute the Server.", SwingConstants.LEFT); c.add(label1);
        start = new JButton("Start"); c.add(start);
        label = new JLabel("Status: Waiting...", SwingConstants.CENTER); c.add(label);

        //Estabelecimento da conexão
        start.addActionListener(
            new ActionListener(){
                public void actionPerformed (ActionEvent str){
                    start.setEnabled(false);

                    try{server = new ServerSocket(3000);
                       connection = server.accept();
                       input = new ObjectInputStream(connection.getInputStream());
                       output = new ObjectOutputStream(connection.getOutputStream());
                       output.flush();}
                    catch(IOException ioException){ioException.printStackTrace();}
                    label.setText("Status: Connected");
                }
            }
        );

        //Encerramento da conexão
        Server.addWindowListener(
            new WindowAdapter(){
                public void windowClosing(WindowEvent e){
                    try{input.close();
                       output.close();
                       connection.close();
                       server.close();}
                    catch(IOException ioException){ioException.printStackTrace();}
                    label.setText("Status: Disconnected");
                    System.gc();
                }
            }
        );

        setSize(150, 110);
        show();
    }
}

```

Figura 5.34: Trecho de código da classe *Server*.

A classe *Server* deve ser a primeira executada em um programa, garantido a possibilidade de conexão ao usuário e todas as funcionalidades do sistema. Em termos de implementação, o desenvolvedor pode optar por chamá-la em uma aplicação isolada ou em uma aplicação similar a um cliente, no caso deste trabalho uma aplicação do LVD. Sendo que, logo após a sua chamada, em ambos os casos, deve-se implementar os respectivos recursos para promover a troca de informações.

No desenvolvimento deste projeto optou-se por chamar a classe *Server* dentro de uma aplicação do LVD, com intuito de minimizar a quantidade de janelas que devem ser manipuladas e as ações realizadas pelo usuário para colocar o sistema em funcionamento.

Para garantir que a classe *Server* seja chamada antes de qualquer execução, ela foi declarada dentro da função *public void initialize()* da classe principal do Transmissor Java conectado ao respectivo LVD. Esta função executa todo o procedimento em seu interior no momento em que o ambiente está sendo carregado. O resultado deste processo é ilustrado na Figura 5.35.

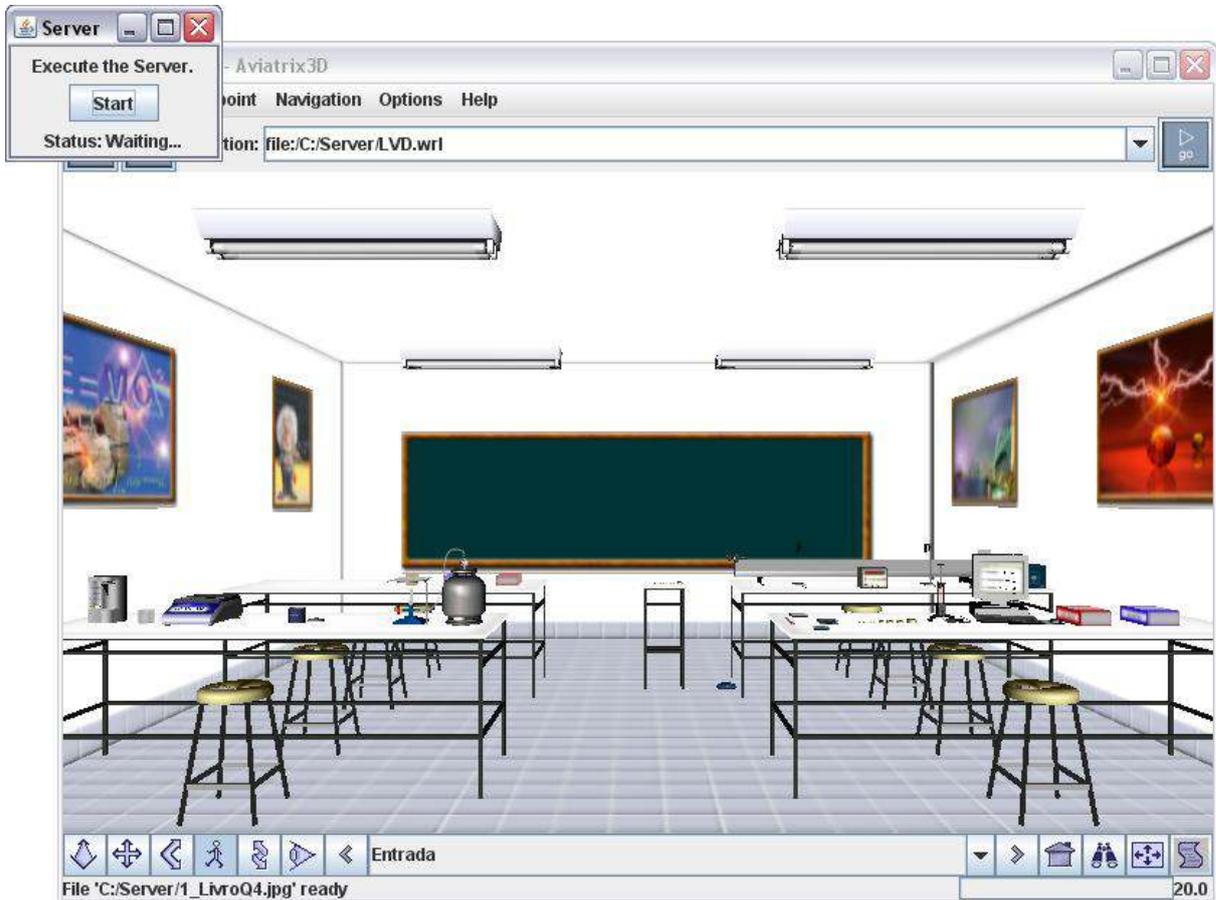


Figura 5.35: Aplicação do LVD chamando a classe *Server*.

Caso o usuário feche a interface *Server*, o ambiente LVD associado a ela continua sendo executado normalmente, mas de maneira local, assim como os outros ambientes conectados a ele. É importante ressaltar que adotando ou não esta estratégia o sistema continua trabalhando de acordo com a arquitetura desenvolvida.

A interface referente à classe *Client* encontra-se ilustrada na Figura 5.36.

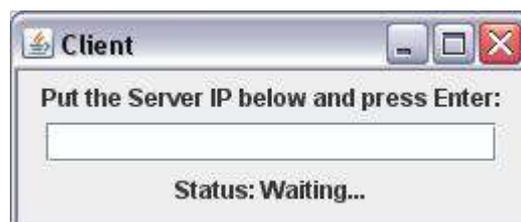


Figura 5.36: Interface de estabelecimento e encerramento de conexão do cliente.

A interface do cliente possui um campo no qual o usuário pode digitar o endereço IP da máquina onde o servidor está sendo executado. Ao pressionar a tecla “*Enter*” do teclado o cliente requisita conexão ao servidor, que responde a requisição, e deste modo a conexão é estabelecida e a troca de informações pode ser iniciada. A interface apresenta ainda uma barra de status que oferece as mesmas informações que a barra de status da interface *Server*: “*Waiting...*”, “*Conected*” e “*Disconnected*”. Caso o usuário queira desconectar o cliente, basta fechar a interface *Client* clicando no botão de fechamento da janela. Após esta ação o ambiente LVD associado a esta interface passa a trabalhar localmente.

Diferentemente da classe *Server* a classe *Client* obrigatoriamente deve ser associada a uma aplicação do LVD e neste projeto optou-se por declará-la dentro da função *public void initialize()* da classe principal do Transmissor Java conectado ao respectivo LVD. Neste contexto, o resultado de execução de um cliente é similar ao apresentado na Figura 5.35, porém, o usuário visualiza a interface *Client* ao invés da *Server*. O código da classe *Client* é análogo ao da classe *Server*, mudando apenas alguns componentes Java de interface e os componentes de conexão apresentados na Figura 5.37.

```

        :
        :
        :
enter.addActionListener(
    new ActionListener(){
        public void actionPerformed (ActionEvent str){
            IP = Str.getActionCommand();
            enter.setEnabled(false);

            //Realizando a conexão
            try{client = new Socket(InetAddress.getByName(IP), 3000);
                input = new ObjectInputStream(client.getInputStream());
                output = new ObjectOutputStream(client.getOutputStream());
                output.flush();}
            catch(IOException ioException){ioException.printStackTrace();}

            label.setText("status: Connected");
        }
    }
);
        :
        :
        :
```

Figura 5.37: Trecho de código da classe *Client*.

Os objetos *input* e *output* declarados nos códigos das classes *Server* e *Client* referem-se às classes *InputStream* e *OutputStream* e são usados na classe Transmissor para a troca de informações. A classe transmissor e os algoritmos criados para otimizar esta troca de informações são apresentados na próxima seção.

5.4.2 Transmissor

O Transmissor é um *script* VRML que utiliza a linguagem Java para processar e distribuir um conjunto de dados. Em suma, quando o Xj3D está interpretando o arquivo LVD.wrl e encontra o *script* Transmissor, com o campo url direcionado para o endereço do arquivo Transmissor.class, ele coloca em execução na máquina virtual o conteúdo da classe *public class Transmissor extends Script* deste arquivo. Esta execução ocorre da seguinte forma:

- enquanto o *browser* está carregando o arquivo LVD.wrl, a máquina virtual executa primeiramente o conteúdo do método *public void initialize()*, independentemente de eventos de entrada;
- após, de acordo com o evento de entrada interno ao *script Transmissor*, denominado *Clock*, ele executa o método *public void processEvent(Event e)*. Como o *Clock* está ligado a um tempo em *loop* infinito, este método é executado de forma cíclica ininterruptamente.

A classe *Transmissor* possui como principal tarefa enviar e receber dados dos ambientes replicados. Ou seja, ele trabalha diretamente na fase de troca de informações dos *sockets* e como já mencionado, as classes *Client* e *Server* são responsáveis pelo estabelecimento e encerramento da conexão.

Para enviar e receber dados neste contexto computacional, de forma eficaz e otimizada, fez-se necessário o desenvolvimento de algoritmos tal fim. Estes algoritmos visam enviar apenas os dados que sofreram modificações pelo ambiente virtual e baseiam-se no envio de *flags* para controlar e evitar o conflito de informações.

As *flags* possuem ainda a função de comunicar aos outros ambientes conectados se ocorrerá o envio (quando o seu valor é 1) ou não (quando o seu valor é 0) de uma nova informação, permitido, deste modo, que eles se preparem para recebê-la. Pode-se perceber então que, o envio das *flags* faz-se necessário a cada execução e como cada variável distribuída necessita de uma *flag* para controle, torna-se inviável enviar cada uma delas em separado. Sendo assim, adotou-se a estratégia de enviar todas concatenadas de uma única vez.

Por exemplo, considere o ambiente (1) com as variáveis distribuídas *A1*, *B1*, *C1* e *D1*, com as respectivas *flags* *F_{A1}*, *F_{B1}*, *F_{C1}* e *F_{D1}*. O ambiente (1) comunica-se com o ambiente (2), que possui as variáveis distribuídas *A2*, *B2*, *C2* e *D2*, com as respectivas *flags* *F_{A2}*, *F_{B2}*, *F_{C2}* e

F_{D2} . Estas variáveis possuem uma correspondência na distribuição, ou seja, $A1$ recebe e envia dados a $A2$, $B1$ a $B2$, $C1$ a $C2$ e $D1$ a $D2$.

Suponha agora que (1) irá enviar somente o valor de $B1$ para (2), logo ele deve enviar $F_{B1} = 1$ para F_{B2} , possibilitando que (2) prepare $B2$ para receber o valor de $B1$, o restante das *flags* serão enviadas com o valor zero. O que se faz é enviar a seqüência binária 0100 de (1) para (2), que associará os valores 0 1 0 0 às *flags* F_{A2} , F_{B2} , F_{C2} e F_{D2} , entendendo deste modo que apenas $B2$ receberá informação.

Esta estratégia reduz significativamente a quantidade de dados na rede, observe, se um ambiente possuir 100 variáveis e o envio das *flags* ocorresse em separado, mesmo sem alterações no ambiente, seriam enviados 100 dados a cada execução, referente aos valores das *flags*. Com a concatenação envia-se apenas a seqüência binária, ou seja, um único dado por execução.

É importante ressaltar ainda que os algoritmos desenvolvidos consideram apenas o envio de valores inteiros ou flutuantes. Neste contexto, caso seja necessário enviar a posição de um objeto virtual, por exemplo, que é um vetor de três dimensões (X, Y, Z), enviam-se separadamente os valores reais de X, Y e Z e não o vetor completo.

A principal vantagem deste procedimento é que caso apenas uma das coordenadas do vetor posição for alterada pelo ambiente, apenas ela será transmitida, evitando deste modo, o envio desnecessário das outras duas coordenadas. Este fato minimiza o tráfego de dados na rede tornando-o mais fluente e rápido.

O algoritmo de envio criado é apresentado na Figura 5.38. Ele exemplifica como enviar um valor advindo do ambiente virtual.

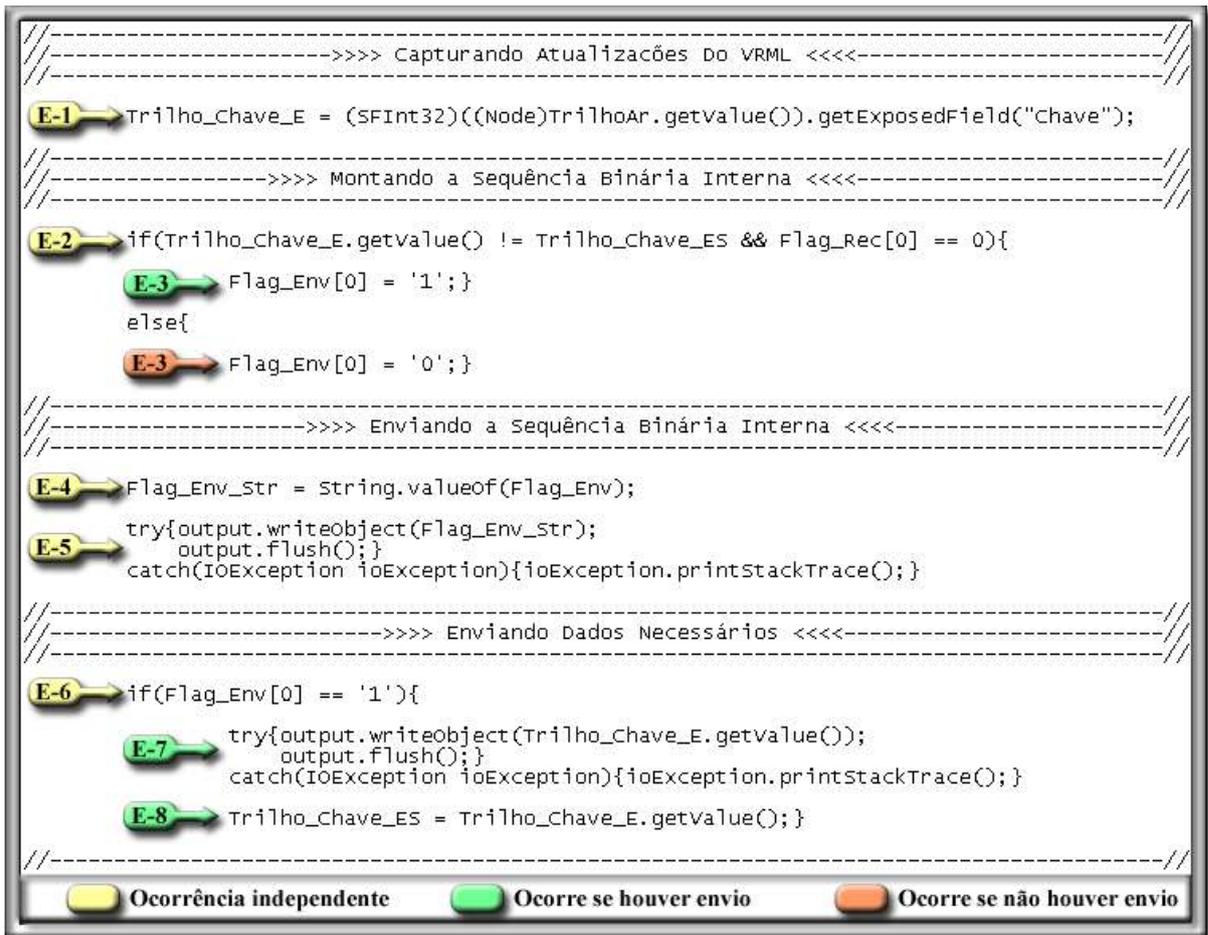


Figura 5.38: Algoritmo para enviar dados do Transmissor.

O primeiro passo (E-1) do algoritmo de envio é capturar o dado candidato a ser transmitido do ambiente VRML e armazená-lo em uma variável Java do mesmo tipo. Por exemplo, no código de E-1 captura-se o campo “Chave”, do tipo SFInt32, que pertence ao nó VMRL “TrilhoAr” e armazena-se o seu valor na variável Java “Trilho_Chave_E”, que também é do tipo SFInt32. A função *getValue()* retorna este valor no formato int da linguagem Java.

Outro tipo de variável importante é o determinado pelas variáveis de Entrada/Saída, no caso do algoritmo de envio da Figura 5.38 tem-se a “Trilho_Chave_ES”. Elas armazenam os valores transmitidos ou recebidos na execução anterior, com isto é possível verificar se ocorreram mudanças no ambiente. Por exemplo, se o valor da variável “Trilho_Chave_E” for diferente de “Trilho_Chave_ES” indica que o campo VRML em questão foi alterado, pois o seu valor atual não é igual ao da execução anterior, neste contexto sugere-se a sua transmissão. Caso contrário, “Trilho_Chave_E” possui o mesmo valor de “Trilho_Chave_ES”, então este campo VRML não foi modificado, e sua transmissão seria redundante e desnecessária.

Existem ainda as variáveis do tipo *flag*, que são as componentes dos vetores “*Flag_Rec*” e “*Flag_Env*”. As componentes de “*Flag_Rec*” podem assumir os valores 0 ou 1, e servem para identificar se o campo analisado está recebendo dados de outros clientes no momento daquela execução. Se o seu valor for 0 indica que não está ocorrendo nenhuma transmissão referente a este campo, caso contrário o seu valor será 1. Se o campo está recebendo dados não deve-se permitir que o sistema envie dados referentes a ele, isto acarretaria em um conflito de informações, o que prejudicaria significativamente a operacionalidade do programa. Já “*Flag_Env*” armazena informações sobre as modificações nas variáveis distribuídas internas do sistema a serem enviadas aos demais ambientes conectados.

O próximo passo (E-2) do algoritmo é verificar se acontecerá a transmissão do dado. Para tanto, primeiramente é verificado se ocorreram modificações no campo VRML analisado (“*Trilho_Chave_E != Trilho_Chave_ES*”); em seguida, se o campo não está recebendo dados no momento da execução (“*Flag_Rec[0] == 0*”). Caso estas duas condições sejam confirmadas o envio do dado em questão deverá ocorrer, caso contrário entende-se que a transmissão deste dado não deverá ser realizada.

Supondo que o envio do dado em questão ocorrerá o próximo passo será (E-3) verde, no qual se atribui o valor 1 a “*Flag_Env[0]*”, para avisar aos demais ambientes que acontecerá uma transmissão relativa a variável que ela representa. Caso o envio não ocorra o próximo passo será (E-3) vermelho, no qual se atribui o valor 0 a “*Flag_Env[0]*” para avisar ao restante dos ambientes que não haverá uma transmissão relativa a variável que ela representa.

No passo (E-4) concatena-se todas *flags* do tipo *Flag_Env* e no passo (E-5) envia-se a seqüência binária resultante desta concatenação para os demais ambientes conectados

Em (E-6) verifica-se qual a ordem interna para a variável em questão, transmitir o seu valor ou não, analisando o valor de “*Flag_Env[0]*”. Caso o seu valor seja 1, a transmissão ocorrerá e os passos (E-7) e (E-8) são executados. Em (E-7), envia-se o novo valor do campo na execução corrente (“*Trilho_Chave_E.getValue()*”). E por fim, em (E-8), o novo valor deve ser gravado na variável do controle de E/S (“*Trilho_Chave_ES*”), preparando o sistema para a próxima execução.

Para receber os dados enviados de modo adequado e garantir o funcionamento do sistema, fez-se necessário o desenvolvimento de outro algoritmo, o qual é apresentado na Figura 5.39.

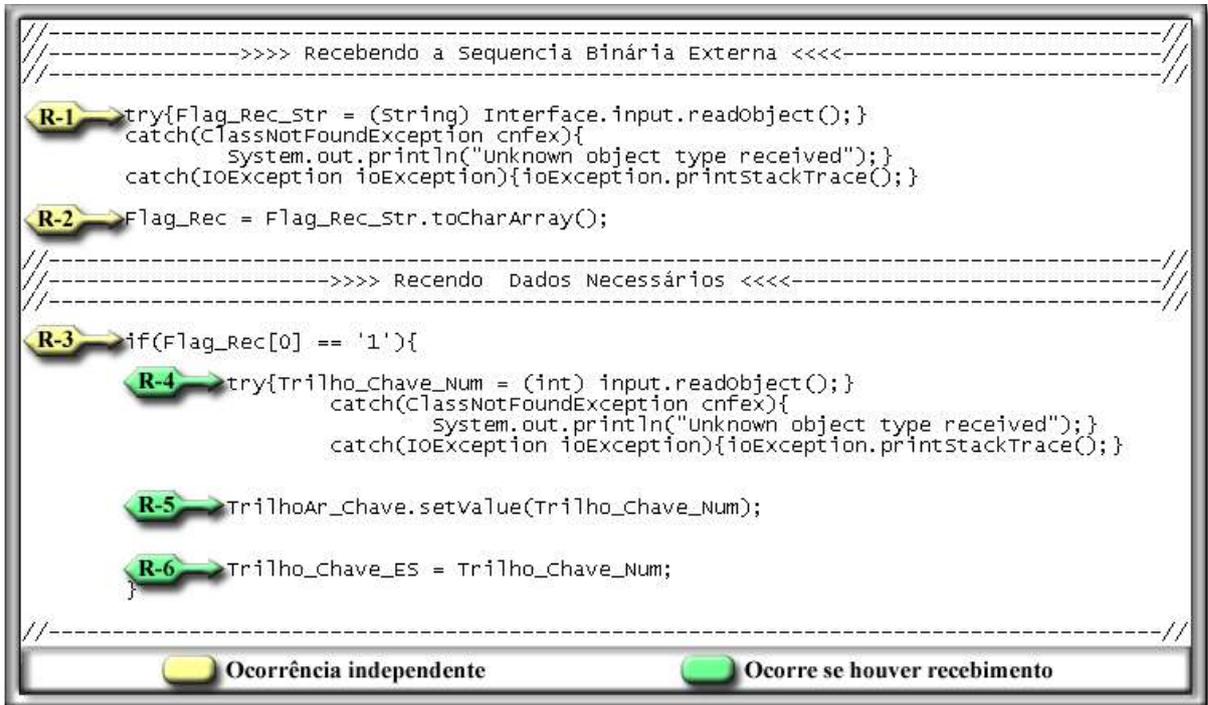


Figura 5.39: Algoritmo para receber dados do Transmissor.

O primeiro passo, (R-1), do algoritmo é receber a seqüência binária, no caso utilizou-se a variável “*Flag_Rec_Str*” para armazenar o seu valor. Logo após, em (R-2) se desfaz a concatenação realizado algoritmo de envio e armazena-se o resultado no vetor “*Flag_Rec*”.

No próximo passo, (R-3), é verificado o valor da *flag* da variável em análise, de acordo com o exemplificado na Figura 5.39 tem-se “*Flag_Rec[0]*”. Caso este valor seja 0 entende-se que o dado não será transmitido e sendo assim, nenhuma ação faz-se necessária. Caso o valor da variável *flag* seja igual a 1, deve-se iniciar um seqüência de passos para receber o novo valor para o campo VRML em questão. Esta seqüência compõe-se dos passos (R-4), (R-5) e (R-6).

No passo (R-4) recebe-se o valor do dado transmitido e o armazena em uma variável de tipo adequado. No caso do algoritmo da Figura 5.39 a variável “*Trilho_Chave_Num*” é responsável por esta tarefa e o seu tipo é int.

Em seguida, no passo (R-5), utiliza-se uma função *setValue* para aplicar o valor recebido a um evento de saída apropriado. No caso do algoritmo de recebimento da Figura 5.39, deve-se aplicar o valor de “*Trilho_Chave_Num*” à “*Trilho_Chave*”, o qual é um

eventOut SFInt32. Este evento de saída é ligado, por meio de uma rota, ao objeto virtual que possui o campo VRML em análise, neste contexto, é possível atualizá-lo de acordo com a transmissão realizada. Por fim, no passo (R-6), armazena-se o valor de “*Trilho_Chave_Num*” na variável “*Trilho_Chave_ES*”, preparando o sistema para a próxima execução.

Ao analisar os algoritmos de envio e recebimento é possível perceber que eles se completam, sendo que um depende do outro. Por exemplo, no momento de enviar um dado, o algoritmo de envio transmite o valor 1 por meio dos passos (E-3) e (E-5), e o de recebimento o recebe por meio de (R-1) e (R-2); envia-se o valor do dado em (E-7) e ele é recebido em (R-4). Neste contexto, em nenhum momento existe a possibilidade de um sistema enviar um dado e outro sistema comunicante não recebê-lo, ou um sistema tentar receber um dado que não foi enviado.

Outro artifício utilizado, e o qual se aconselha, é o uso de tratamento de exceções para os comandos referentes aos *sockets*, na linguagem Java ele é feito por meio das funções *try* e *catch*. Este de tratamento previne que possíveis erros afetem o restante do sistema. Estes erros podem se enquadrar em problemas no envio e recebimento, na abertura e fechamento dos *sockets*, etc.

Normalmente, o tratamento de exceções é usado em situações excepcionais em que um método é incapaz de completar sua tarefa por razões que ele não pode controlar. Em suma, dentro de um bloco *try* inclui-se o código que pode gerar uma exceção e no bloco *catch* especifica-se o tipo de exceção que se pode capturar e um tratador de exceção.

Por meio da criação das classes *Server*, *Client* e da classe *Transmissor*, desenvolvida com base nos algoritmos de envio e recebimento apresentados, foi possível construir o sistema de distribuição do LVD. A Figura 5.40 ilustra dois ambientes LVD conectados através deste sistema. Neste momento eles estão conectados através do IP *localhost* (127.0.0.1), ou seja, os dois ambientes virtuais estão se comunicando por meio de uma mesma máquina virtual Java.

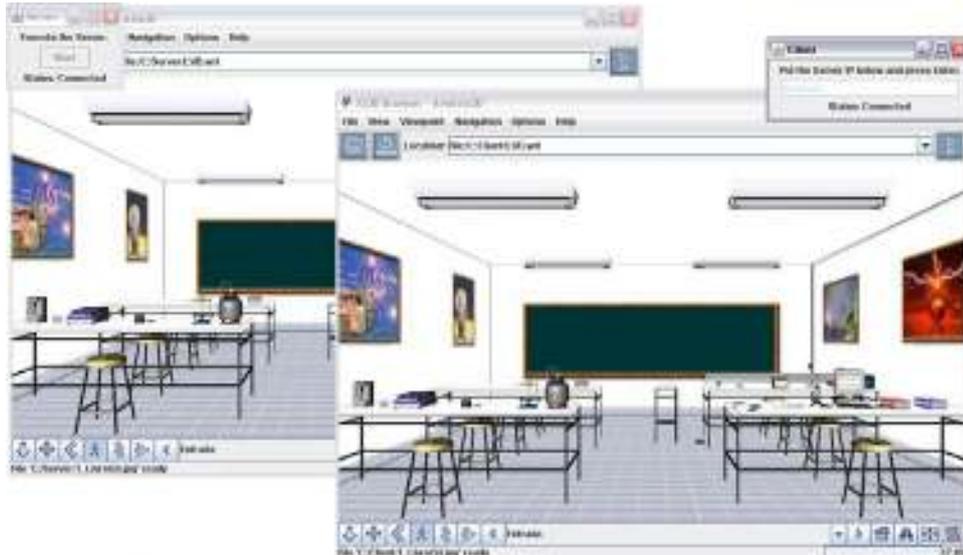


Figura 5.40: Ambientes LVD conectados.

Como já mencionado, apenas os dados dos objetos virtuais são distribuídos, deste modo, as mudanças de câmeras e as alterações de suas posições, realizadas em um dado ambiente, não influenciam as câmeras dos demais ambientes conectados a ele. Neste contexto, cada usuário de um ambiente LVD, que esteja em distribuição, pode escolher o ponto de visão que melhor lhe adéque no seu trabalho e mesmo assim, visualizar a mesma experiência que os demais participantes, como ilustra a Figura 5.41.

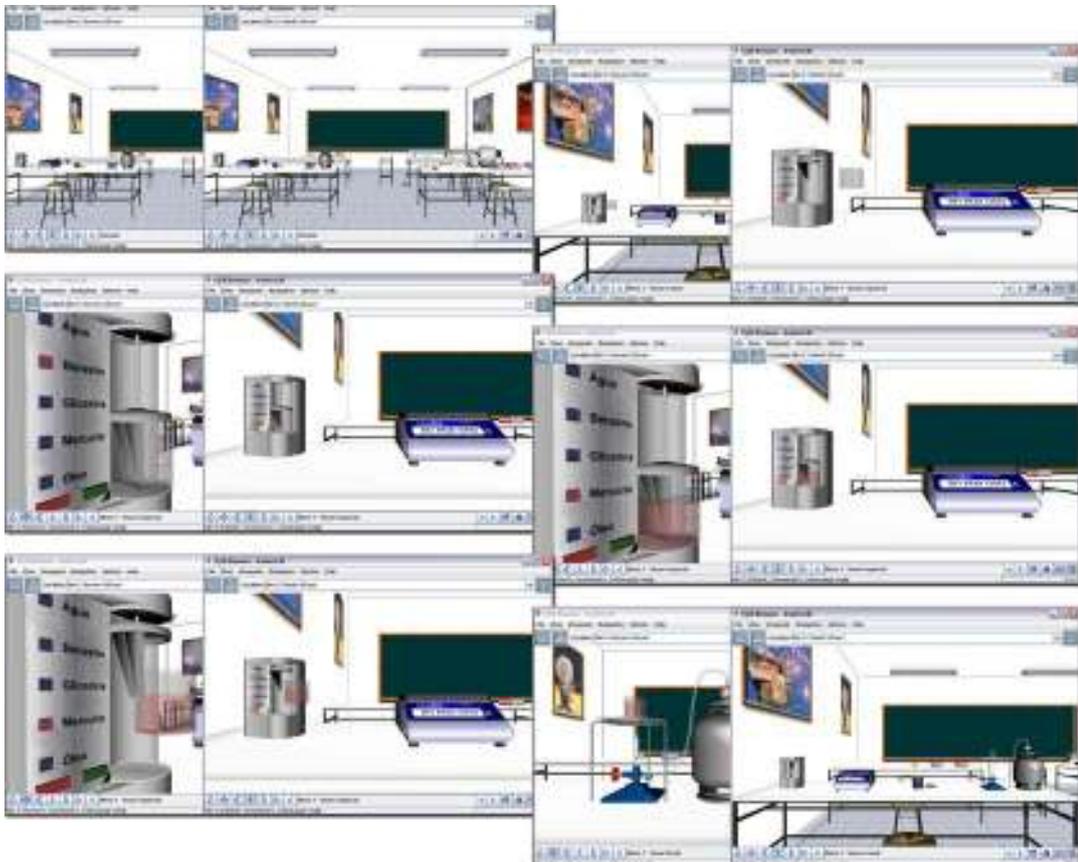


Figura 5.41: O uso de câmeras em LVDs conectados.

A Figura 5.42 apresenta dois computadores portando cada um deles o ambiente virtual do LVD, estes ambientes estão conectados remotamente via uma rede local. É importante ressaltar que estes computadores possuem desempenhos aceitáveis e compatíveis para a execução dos ambientes. Neste sistema todos os computadores conectados trabalham de acordo com a velocidade de processamento do computador de menor velocidade, uma vez que o ambiente LVD baseia-se no sincronismo para realizar a distribuição. A rede de conexão entre eles permitiu o alcance de uma alta velocidade na transmissão de dados. Estes fatores viabilizaram as condições para que estes ambientes apresentassem uma alta qualidade no momento da execução, permitindo uma comunicação próxima à realizada em tempo real. É plausível de destaque, que a qualidade do *hardware* e o desempenho da rede são fatores que influenciam diretamente a performance de execução e distribuição do LVD.



Figura 5.42: Ambientes virtuais do LVD conectados remotamente.

Estes dois computadores apresentado na figura usam sistemas operacionais diferentes, sendo que um deles utiliza o Windows XP e o outro o Windows Vista. A Figura 5.43 ilustra, com maiores detalhes as telas destes computadores, simultaneamente, durante a execução da conexão do LVD.

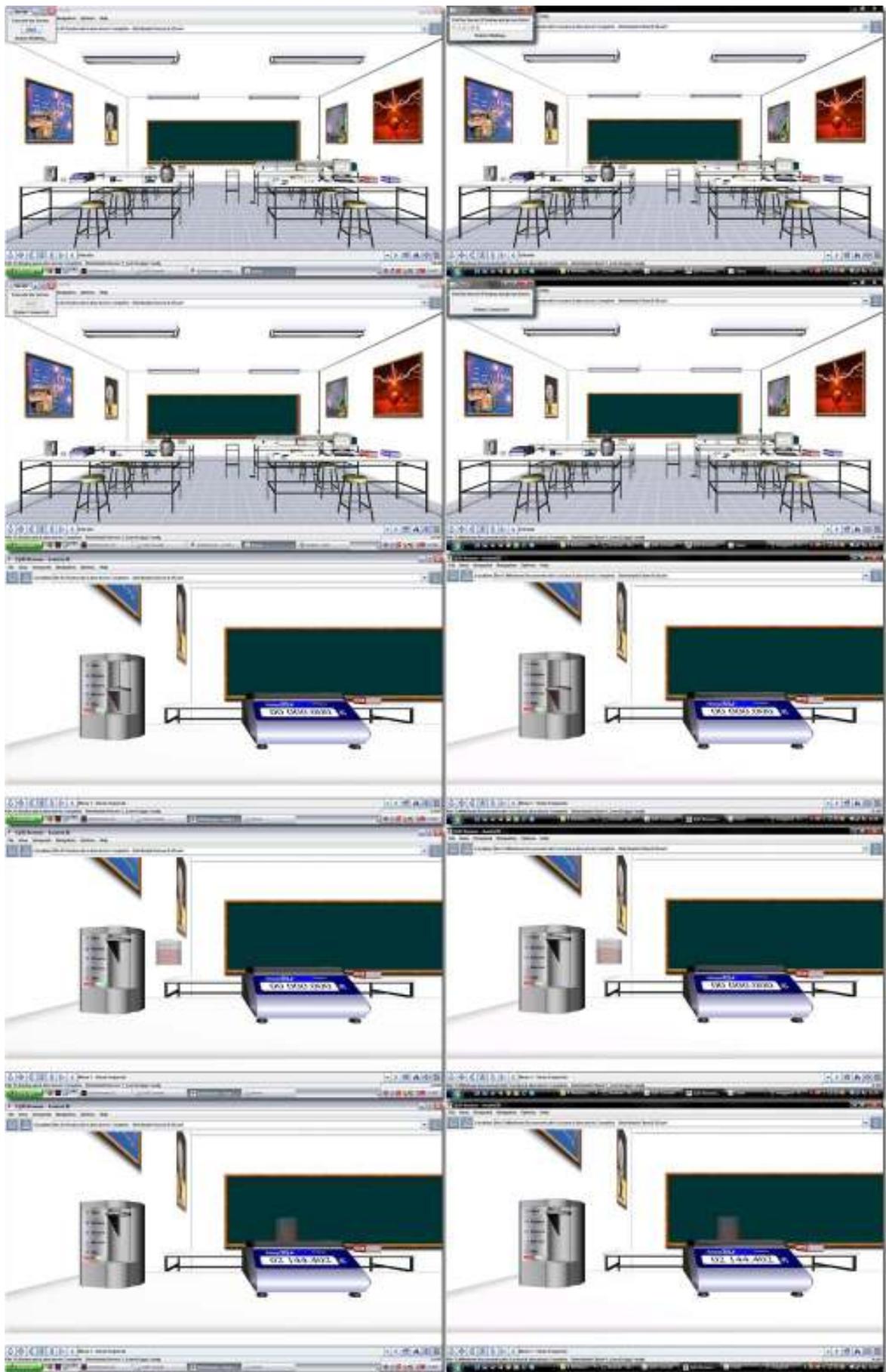


Figura 5.43: Tela dos computadores conectados remotamente apresentando a conexão do LVD.

5.5 Considerações finais

Este capítulo apresentou alguns dos detalhes de desenvolvimento do LVD, como as linguagens adotadas para a criação do sistema, a estruturação e implementação dos objetos de aprendizagem que compõem o laboratório virtual e por fim, foi abordada a construção do seu sistema de distribuição. Destaca-se que em laboratórios virtuais permitem ao aluno entrar em contato com equipamentos que dificilmente são encontrados nas escolas e vivenciar experiências educacionais de grande importância para o seu desenvolvimento cognitivo.

Optou-se pelo uso das linguagens VRML/X3D e Java, por serem distribuídas gratuitamente e apresentarem um padrão multiplataforma. No entanto, as linguagens VRML e X3D não possuem componentes internos que permitam a distribuição em rede e são linguagens estáticas, ou seja, baseiam-se em disparos de eventos para realizar alguma ação ou atualização. Neste contexto, foi necessário o desenvolvimento de uma arquitetura de distribuição que tratasse deste fator adequadamente, a solução encontrada foi colocar a estrutura responsável pela distribuição trabalhar em um ciclo infinito de execução. Deste modo, os algoritmos desenvolvidos para a transmissão de dados foram implementados para trabalhar nestas condições e oferecer um melhor desempenho para os sistemas comunicantes.

Após apresentar e finalizar as discussões sobre a implementação do sistema, aborda-se, no próximo capítulo, a avaliação pedagógica deste trabalho junto a usuário potenciais.

Capítulo 6

Avaliação e validação do sistema

6.1 Introdução

Este capítulo aborda a avaliação do presente trabalho, para tanto, apresenta-se a metodologia de avaliação utilizada, demonstra-se um caso de uso da mesma e são discutidos os resultados obtidos deste processo avaliativo. Por fim, realiza-se ainda uma correspondência entre os resultados da avaliação e o modelo pedagógico proposto nesta pesquisa. Todo o procedimento avaliativo, inferência, análise e discussão dos resultados fundamentam em teorias estatísticas consagradas e coerentes ao estudo almejado.

6.2 Metodologia de avaliação

A metodologia de avaliação utilizada no projeto LVCF consiste em uma seqüência de passos que visa o alcance de resultados condizentes, veridicamente, com a situação real avaliada. Para tanto, faz-se necessária à adoção de técnicas e procedimentos avaliativos adequados para minimizar os fatores que influenciam, negativamente ou positivamente, o processo de validação do sistema.

O primeiro passo da metodologia de avaliação é separar uma amostra λ de uma população Ω de alunos aptos aos testes, sendo que, conceitualmente considera-se como população um conjunto de elementos ou indivíduos que possuem certas propriedades comuns e amostra como um subconjunto representativo da população.

Caso a população Ω seja maior que a amostra λ , sugere-se o uso do processo de amostragem probabilístico aleatório simples sem reposição para extração de λ . Neste processo garante-se a cada um dos elementos da população Ω a mesma probabilidade de ser incluído na amostra λ . O intuito desta estratégia é evitar a tendenciosidade na amostra e aumentar a confiabilidade dos resultados da avaliação.

É importante observar que quanto maior o número de elementos que compõem a amostra λ , menores serão os erros estatísticos envolvidos no processo de inferência. Todavia, fatores como a quantidade de alunos disponíveis para participarem da avaliação, o local e os

equipamentos que podem ser utilizados na mesma, influenciam diretamente na determinação do número de elementos de λ .

No processo de avaliação do LVCF a amostra λ é igual à população Ω , ou seja, optou-se por tomar todos os alunos aptos e disponíveis para a realização dos testes. Neste contexto, λ é composta por 84 alunos, todos eles estudantes da Escola de Aplicação da Universidade Federal de Roraima – UFRR e cursam o segundo ou o terceiro colegial. A Escola de Aplicação pertence ao CEDUC – Centro de Educação da UFRR, ela destina-se a execução de atividade práticas dos cursos superiores de Pedagogia e Psicologia, por intermédio do contato direto com a Educação Básica (ensino infantil, fundamental e médio).

O segundo passo da metodologia consiste em apresentar aulas expositivas do assunto a ser avaliado para a amostra λ , no caso do projeto LVCF foram ministras aulas sobre mecânica, em específico o estudo de movimentos retilíneos uniformes ou variados e leis de Newton. Neste contexto, comenta-se ainda sobre alguns instrumentos de laboratórios, aplicados nestas áreas, e suas funções em uma experiência física. As aulas expositivas têm como intuito transmitir e reforçar os conhecimentos básicos necessários para a avaliação e diminuir a diferencia erudita entre os alunos, visando aumentar o nível de homogeneidade de λ .

No terceiro passo ocorre a aplicação de um teste básico sobre os conceitos aprendidos nas aulas expositivas do segundo passo. Em relação à avaliação do LVCF, este teste ocorreu em duas etapas seqüenciais, em um primeiro dia foi ministrada a aula sobre movimentos uniformes e uniformemente variados e em seguida aplicado teste relativo a este assunto, o qual se encontra no Anexo 1. No segundo dia foi exposta a aula de leis de Newton e em seguida aplicado o seu respectivo teste, o qual é apresentado no Anexo 2. Esta etapa visa avaliar os conhecimentos aprendidos e de certa forma quantificá-los, para uso no próximo passo.

No quarto passo a amostra λ é dividida em três grupos, α , β e γ , com quantidades iguais de elementos. Na avaliação do LVCF estes grupos possuem 28 indivíduos cada. Este procedimento precisa ser realizado de modo que α , β e γ preservem as características de λ . Para tanto, utiliza-se como parâmetro, característica numérica estabelecida para toda a população, a nota de cada indivíduo no teste do terceiro passo.

Calcula-se então a média das notas do teste na amostra λ (μ_λ) e o seu desvio padrão (σ_λ). Estes valores são, respectivamente, a medida de tendência central dos dados e a medida

de dispersão ou variação em relação a esse centro. Como os dados trabalhados não são constantes, faz-se necessário estabelecer um valor ao redor do qual se agrupa a amostra e o valor de flutuação dos dados em torno do mesmo.

Após, considerando μ_α , μ_β , e μ_γ como sendo, respectivamente, a média de α , β e γ , e conhecendo-se os seus respectivos desvios padrões σ_α , σ_β e σ_γ , devem-se então ocorrer que μ_α , μ_β e μ_γ são estatisticamente iguais a μ_λ . Esta igualdade estatística pode ser verificada por meio de um teste de hipóteses de diferença de médias.

Para que esta divisão ocorra como o esperado considera-se a amostra λ como uma população e amostra-se α , β e γ por um processo intencional de agrupamento por meio de dois estágios. O primeiro estágio consiste em desenvolver categorias ou cotas de controle de elementos da população. No segundo estágio selecionam-se os elementos da amostra com base na conveniência ou no julgamento. É importante ressaltar que a intencionalidade nesta etapa é fundamental, uma vez que se deseja forçar a construção de três novos grupos com as mesmas características de λ .

No quinto passo os grupos α , β e γ são separados fisicamente e cada um deles passa a trabalhar de forma específica. As atividades para os três grupos são aplicadas, de modo que, os indivíduos do grupo α resolverão estas atividades de forma individual e sem utilizarem o ambiente virtual do LVD; os indivíduos do grupo β resolverão suas atividades também de maneira individual, porém, com o auxílio do ambiente LVD; os indivíduos do grupo γ , por sua vez, realizarão as suas atividades usando o LVD de maneira cooperativa, em grupos de duas pessoas escolhidas aleatoriamente por meio de sorteio.

As atividades aplicadas são as mesmas para os três grupos e são compostas efetivamente por problemas. Estes problemas propõem inicialmente uma situação a qual deriva novos tópicos problemas, sendo que, cada um destes tópicos é desenvolvido em correspondência com uma das fases do modelo de resolução de problemas proposto neste projeto.

As estratégias supracitadas visam permitir a avaliação de cada etapa da resolução de problemas em relação às tecnologias e metodologias pedagógicas propostas neste trabalho. Para tanto, entende-se que deve ser realizada uma avaliação de maneira comparativa, na qual uma primeira amostra é testada sem a influência da RV, uma segunda utiliza a RV de forma individual e uma terceira utilizada a RV cooperativamente. Como os grupos avaliados possuem as mesmas características, a comparação dos resultados torna-se mais confiável,

sendo possível verificar em quais etapas da resolução de problemas a RV contribuiu e ainda, em quais de seus aspectos o uso cooperativo da RV influencia.

No início de testes com amostras que utilizam a RV é plausível separar um período para explicar e permitir que o usuário se adapte com a interface do sistema. No caso LVCF foram necessárias duas horas para tal procedimento. Foram transmitidas ainda, a amostra γ , informações básicas sobre o funcionamento da plataforma Moodle como, o procedimento para acessar o sistema com uma senha, visualizar as tarefas, postar documentos, usar o chat, etc. O desconforto e a inabilidade do usuário dentro de um ambiente virtual podem comprometer os resultados de uma avaliação.

Por fim, no sexto passo são catalogados os resultados das avaliações referentes a cada etapa do modelo de resolução de problemas, colocando-os em gráficos para uma melhor visualização, análises e conclusões.

A próxima seção aborda a subdivisão da amostra λ na avaliação do LVCF, estruturalmente descrita no quarto passo da metodologia de avaliação. Nesta abordagem são apresentados e detalhados, de maneira numérica e gráfica, os procedimentos estatísticos adotados.

6.3 A criação dos grupos de teste na avaliação do LVCF

Como já mencionado a amostra λ possui como parâmetro a nota de cada aluno no teste aplicado no terceiro passo. Todos os procedimentos de análise, inferência e de subdivisão da amostra λ são realizados com base neste parâmetro. O conjunto de notas dos alunos, orçadas de maneira crescente, é apresentado na Figura 6.1.

2.5	3.0	3.5	3.5	4.0	4.5	4.5
4.5	4.5	4.5	5.0	5.0	5.0	5.0
5.0	5.0	5.0	5.5	5.5	5.5	5.5
5.5	5.5	5.5	5.5	5.5	5.5	6.0
6.0	6.0	6.0	6.0	6.0	6.0	6.0
6.0	6.0	6.0	6.0	6.0	6.0	6.0
6.5	6.5	6.5	6.5	6.5	6.5	6.5
6.5	6.5	7.0	7.0	7.0	7.0	7.0
7.0	7.0	7.0	7.0	7.5	7.5	7.5
7.5	7.5	7.5	8.0	8.0	8.0	8.0
8.0	8.0	8.5	8.5	8.5	8.5	8.5
9.0	9.5	9.5	9.5	10.0	10.0	10.0

Figura 6.1: Notas da amostra λ .

De posse destes dados é possível realizar o cálculo dos valores da média e do desvio padrão da amostra λ , os quais são, respectivamente, $\mu_\lambda = 6.4702$ e $\sigma_\lambda = 1.6045$. A Figura 6.2 apresenta o histograma (cor azul) dos dados supracitados. Ela contém ainda uma curva, na cor vermelha, que consiste em uma aproximação deste conjunto de dados para uma função contínua gaussiana. Perceba que o histograma tende ao formato da curva, é de caráter comum notas de avaliações acompanhem este padrão, sendo que, quanto maior a amostra melhor é a percepção de tal fato.

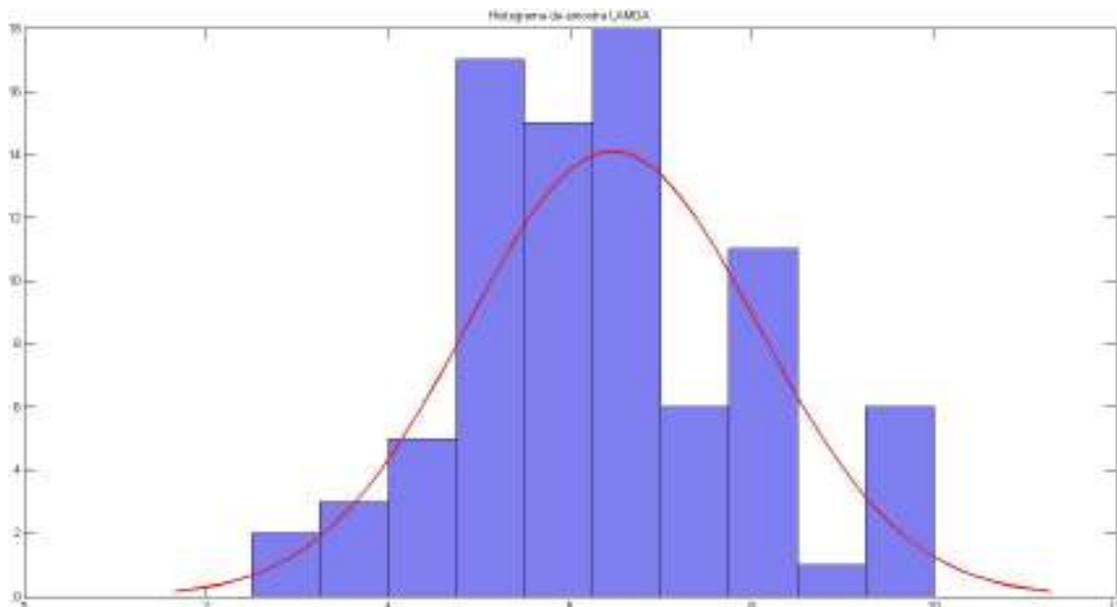


Figura 6.2: Histograma da amostra λ .

Neste contexto, torna-se plausível verificar se existe uma função de densidade de probabilidade gaussiana que represente estes dados de uma forma aceitável. O teste estatístico que se realiza nesta situação é denominado teste de aderência. Ao efetivar esta verificação encontrou-se a curva de densidade ilustrada na Figura 6.3, a qual possui um nível de confiança de 97.94%, ou seja, a probabilidade destes dados se reportarem a esta curva de densidade é de 0.9794. Observe que este valor é expressivo e caracteriza uma aproximação de qualidade, uma vez que, é consensual na comunidade estatística que o valor de 90% ainda é aceitável.

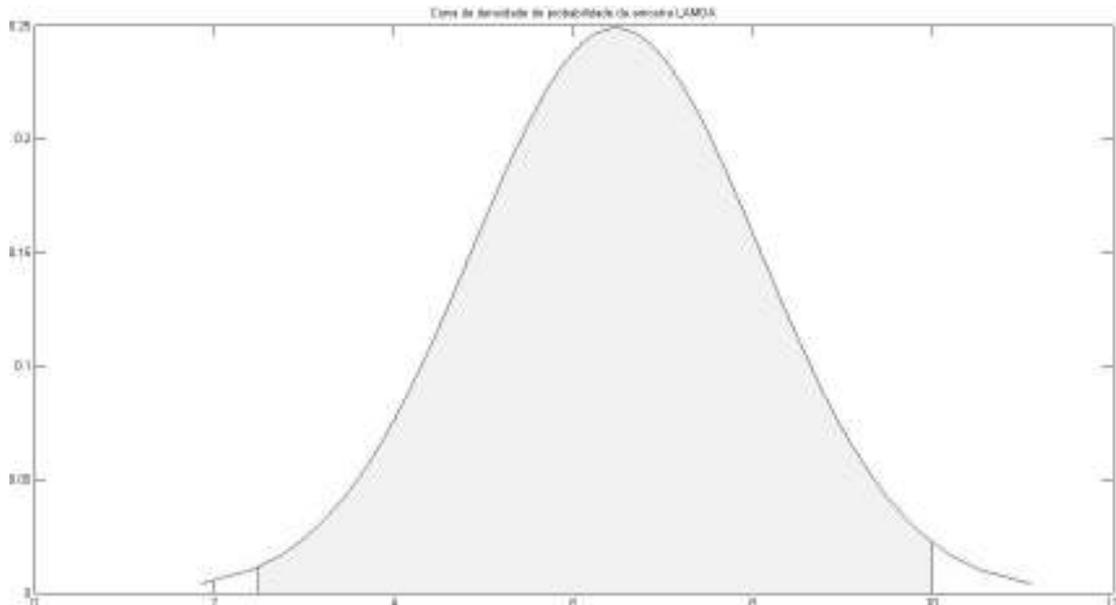


Figura 6.3: Curva de densidade de probabilidade λ .

Uma distribuição de probabilidade descreve a chance que uma variável pode assumir ao longo de um espaço de valores. Esta distribuição pode ser discreta ou contínua dependendo de sua variável. A função densidade de probabilidade é uma função utilizada para representar a distribuição de probabilidade caso a sua variável seja contínua. A função de densidade de

probabilidade gaussiana é dada por $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ [Triola, 2005]. Na qual, x é a variável, $f(x)$ retorna a probabilidade de certo valor assumido por x acontecer em um grupo de dados, σ e μ são respectivamente a o desvio padrão e a média deste grupo, e por fim, π e e são os números irracionais *pi* (3.14159...) e *neper* (2.71828...).

A partição da amostra λ ocorreu de acordo com o descrito no quarto passo, originando-se, deste modo, os grupos α , β e γ com 28 elementos cada como mencionado, sendo que, $\alpha \cup \beta \cup \gamma = \lambda$. A subdivisão realizada intuiu que α , β e γ herdassem as características de λ , desta forma, a partição que apresentou melhor resultado foi a ilustrada na Figura 6.4.

ALPHA	2.5	4.5	4.5	4.5	5.0	5.5	5.5
	5.5	5.5	6.0	6.0	6.0	6.0	6.0
	6.5	6.5	6.5	7.0	7.0	7.0	7.5
	7.5	8.0	8.0	8.5	8.5	9.5	10.0
BETA	3.0	4.0	4.5	5.0	5.0	5.0	5.5
	5.5	5.5	6.0	6.0	6.0	6.0	6.0
	6.5	6.5	6.5	7.0	7.0	7.0	7.5
	7.5	8.0	8.0	8.5	8.5	9.5	10.0
GAMMA	3.5	3.5	4.5	5.0	5.0	5.0	5.5
	5.5	5.5	6.0	6.0	6.0	6.0	6.0
	6.5	6.5	6.5	7.0	7.0	7.0	7.5
	7.5	8.0	8.0	8.5	9.0	9.5	10.0

Figura 6.4: Notas dos grupos α , β e γ .

Para atingir estes resultados foram estabelecidas três cotas de 28 elementos cada, número referente à quantidade indivíduos desejados em cada grupo após a subdivisão. As retiradas de λ foram feitas de três em três elementos seqüenciais, como por exemplo, na primeira retirada obtêm-se os elementos 2.5, 3.0 e 3.5, na segunda retirada têm-se os elementos 3.5, 4.0 e 4.5, e assim por diante. Após cada retirada distribui-se um elemento para cada grupo, almejando sempre manter a somatória dos elementos em cada grupo, o mais próxima possível entre α , β e γ , como segue: depois da primeira retirada têm-se que $\alpha = \{2.5\}$, $\beta = \{3.0\}$ e $\gamma = \{3.5\}$; após a segunda $\alpha = \{2.5, 4.5\}$, $\beta = \{3.0, 4.0\}$ e $\gamma = \{3.5, 3.5\}$. Perceba que na segunda retirada a somatória dos elementos, em todos os grupos, é igual a sete. Este processo deve ser realizado até que as cotas estabelecidas sejam cumpridas, chegando à formação final dos grupos.

Por meio deste procedimento de partição os novos grupos tendem a uma mesma medida central (média) com dispersões similares (desvio padrão). Calculada a média e o desvio padrão nos grupos α , β e γ obtêm-se, respectivamente, que $\mu_\alpha = 6.4643$, $\mu_\beta = 6.4643$ e $\mu_\gamma = 6.4821$, e $\sigma_\alpha = 1.6326$, $\sigma_\beta = 1.6097$ e $\sigma_\gamma = 1.6301$. Pode-se observar que as médias μ_α , μ_β , μ_γ e μ_λ apresentam valores muito próximos, assim como os valores de σ_α , σ_β , σ_γ e σ_λ . Porém, deve-se avaliar ainda se estes valores garantem a igualdade estatística da amostras. A Figura 6.5 apresenta os histogramas e as curvas de densidade de probabilidade dos grupos α , β e γ .

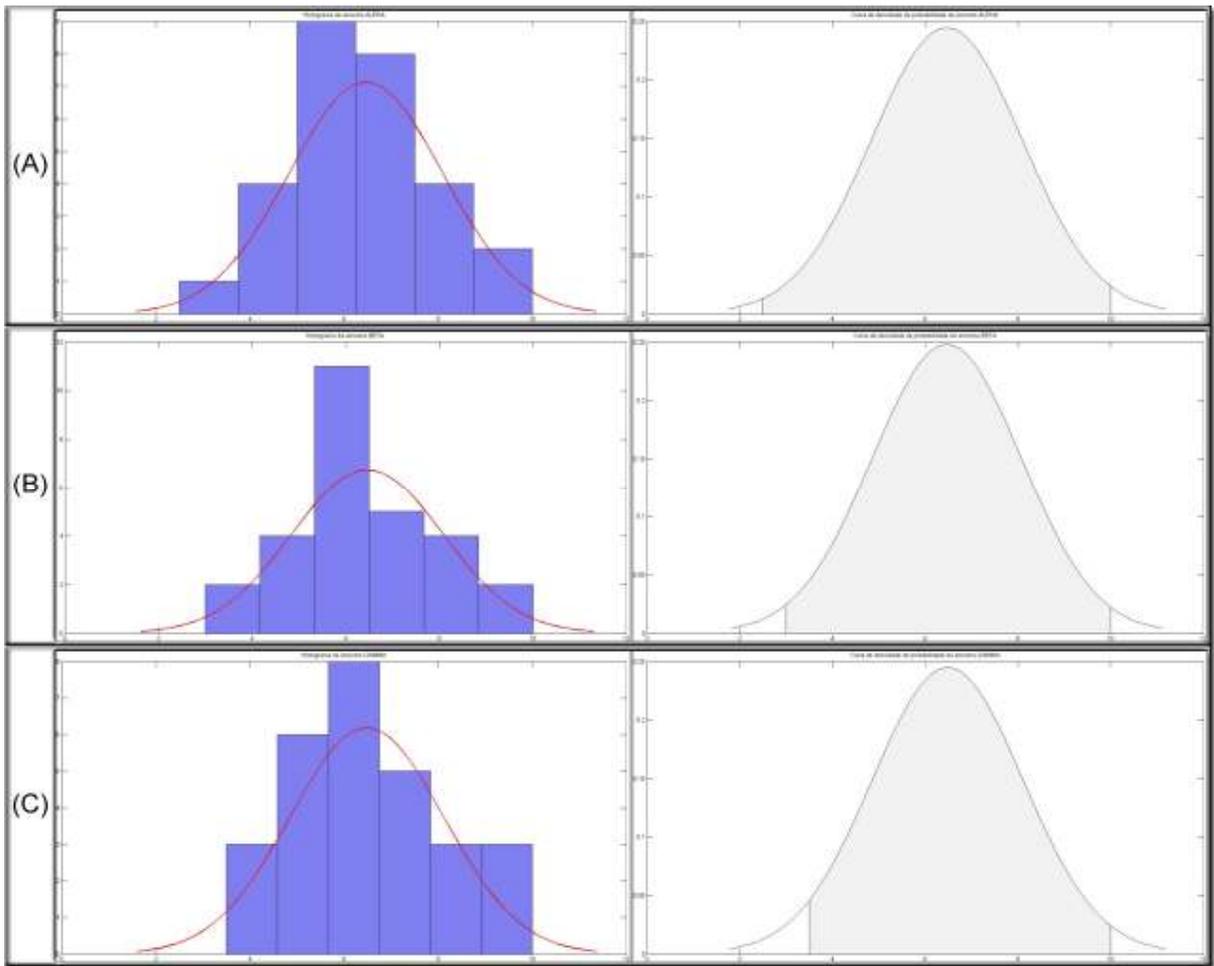


Figura 6.5: Histogramas e curvas de densidade dos grupos α , β e γ .

Observe que os histogramas de α , β e γ (Figura 6.5 (a), (b) e (c) respectivamente) obviamente possuem formatos diferentes, pois são advindos de grupos de dados diferentes. No entanto, percebe-se que eles tendem a curvas gaussianas similares. Para uma análise de maior consistência faz-se necessária a verificação de suas curvas de densidade de probabilidade, as quais também são ilustradas na Figura 6.5. Estas curvas foram obtidas com confiança de 97.72%, 97.03% e 95.09%, para, respectivamente, os grupos α , β e γ . A Figura 6.6 apresenta uma sobreposição destas curvas.

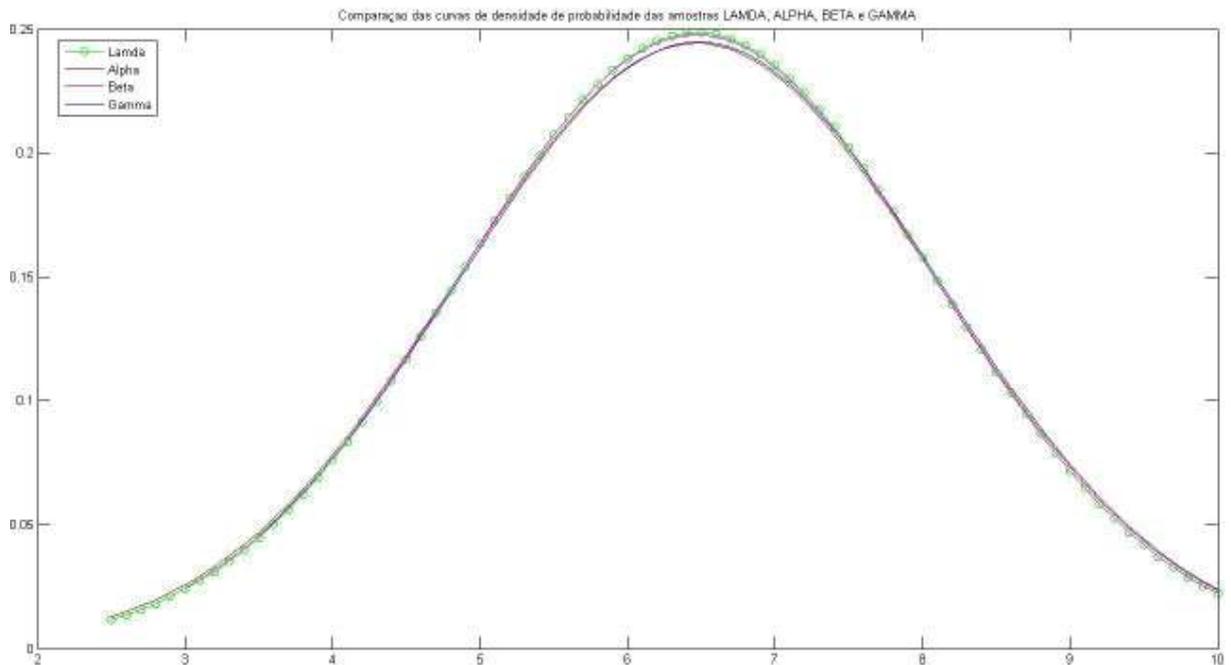


Figura 6.6: Sobreposição das curvas de densidade de probabilidade de α , β , γ e λ .

Pode-se observar que as curvas de densidade de probabilidade de α , β , γ e λ apresentam fortes similaridades, em alguns momentos elas se sobrepõem totalmente. Este fato deve-se à grande proximidade estatística das amostras. O próximo passo consiste em verificar se esta proximidade é suficiente para considerar as amostras estatisticamente iguais.

A teoria estatística denominada teste de hipóteses é a convencionalmente utilizada para realizar este tipo de análise. Para tanto, deve-se utilizar o teste de hipóteses adequando ao tipo de amostra em estudo. Neste trabalho foi utilizado o teste t de *Student* para diferença entre médias, ele permite verificar se existe igualdade entre médias de duas amostras [Morettin e Bussab, 2003].

Primeiramente foi testado se os grupos α , β e γ são estatisticamente iguais a λ , duas a duas, ou seja, verificou se α é igual a λ , β é igual a λ , e γ é igual a λ . Obteve-se para estes três casos a conclusão da existência da igualdade, com significância do teste de 98.65% ($\alpha = \lambda$), 98.65% ($\beta = \lambda$) e 97.30% ($\gamma = \lambda$). Deste modo, pode-se verificar com um nível de confiança considerável, que α , β e γ herdaram as características de λ .

Obviamente como α , β e γ são iguais a λ , então estes grupos também serão iguais entre eles. Apenas para confirmar tal fato, foi aplicado o teste t de *Student* nos grupos α , β e γ , obtendo-se significâncias de 100% para $\alpha = \beta$, 96.75% para $\alpha = \gamma$ e 96.73% para $\beta = \gamma$.

Observe que com estes resultados o estudo avaliativo tem uma maior confiabilidade no que se refere a uma comparação entre testes diferentes aplicados aos grupos α , β e γ . Pois caso

os testes fossem realizados em uma mesma amostra a metodologia de um teste influenciaria drasticamente nos resultados dos outros. E caso os testes fossem aplicados em amostras estatisticamente diferentes não existiriam padrões para comparações.

A próxima seção exemplifica os estudos comparativos realizados, apresentando os desempenhos dos alunos em um problema, o qual é associado às diversas etapas do modelo de resolução de problemas proposto.

6.4 Exemplificação da análise comparativa em um problema

No processo avaliativo do LVCF os problemas aplicados aos grupos α , β e γ , mencionados no quinto passo da metodologia de avaliação, focalizaram essencialmente os conteúdos da Física de movimentos retilíneos uniformes e uniformemente variados e as leis de Newton. Tais conteúdos foram apresentados nas aulas expositivas do segundo passo da metodologia.

Estes assuntos são abordados e analisados em experimentos reais nos laboratórios de Física por meio do trilho de ar. Neste contexto, os problemas desenvolvidos nesta etapa são correlatos às aplicações do trilho de ar virtual do LVD, ou seja, os alunos poderão utilizar este objeto virtual para auxiliar na resolução de problemas. No entanto, deve-se ressaltar que os problemas criados viabilizam a amostra α , que não utiliza tecnologia no processo de resolução, a trabalhar com os mesmos com naturalidade.

As amostras β e γ foram motivadas a usar o ambiente virtual por meio da apresentação da existência de uma relação entre os experimentos que podem ser realizados com o trilho de ar e os conteúdos em questão, sendo fornecido, complementarmente, a estes alunos um manual explicativo do trilho virtual, o qual é apresentado no Anexo 3. A realização destes esclarecimentos quanto ao objeto de aprendizagem trilho de ar é de suma importância, uma vez que a grande maioria destes alunos nunca entrou em contato, nem mesmo visual, com um deles. Deve-se ressaltar que o trilho não é o objeto de estudo nestas situações de aprendizagem, mas sim o experimento que ele possibilita realizar e os conceitos e deduções que podem ser compreendidos a partir deste experimento.

Foram aplicadas ainda para as amostras β e γ questões específicas, que visam direcionar os alunos a entrarem em contato com a experimentação e aprendizagem pela heurística, acarretando assim uma maior aproximação dos mesmos com o ambiente virtual. Para a amostra γ foi explicado também o funcionamento da cooperação no sistema, visando

proporcionar aos seus usuários o conhecimento necessário sobre o ambiente para evitar desconfortos e desentendimentos entre os participantes imersos e uma maior possibilidade do uso da cooperação de uma forma benéfica para a aprendizagem.

Para avaliar o desempenho dos grupos α , β e γ na resolução de problemas elaborou-se um conjunto de situações problema, as quais servem de fomento para análises do aluno em relação ao modelo de resolução de problemas proposto. Por exemplo, um dos problemas aplicados foi: “Um bloco A de 40 g está conectado a um bloco B de 50 g por meio de um fio, como ilustra a Figura 6.7. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine o tempo que o bloco A gasta para percorrer 60 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.

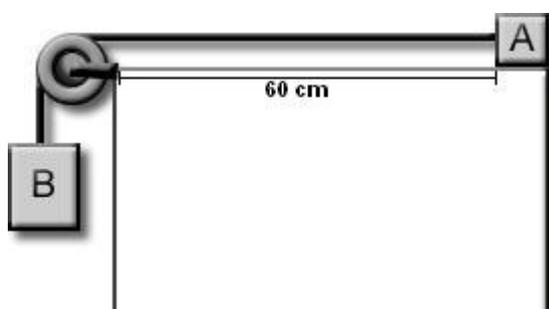


Figura 6.7: Ilustração do problema exemplo.

A solução numérica deste problema pode-se ser obtida por intermédio dos conceitos da segunda lei de Newton e de movimentos uniformemente variados. Uma possibilidade de solução seria:

- (i) Encontrar o peso do bloco B: $P = m \cdot g \Rightarrow P_B = 0,05 \cdot 9,8 \Rightarrow P_B = 0,49 \text{ N}$;
- (ii) Calcular a aceleração do sistema: $F_R = m \cdot a \Rightarrow 0,49 = 0,09 \cdot a \Rightarrow a = 5,44 \text{ m/s}^2$
- (iii) Encontrar o tempo pedido: $d = v_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2 \Rightarrow 0,6 = 0 \cdot t + \frac{1}{2} \cdot 5,44 \cdot t^2 \Rightarrow t^2 = 0,6 / 2,72 \Rightarrow t^2 = 0,22 \Rightarrow t = \pm 0,46 \Rightarrow t = 0,46 \text{ s}$

A observação de forma experimental no ambiente virtual poderia ocorrer por meio dos seguintes passos:

- (I) Colocar o carrinho disparador magnético ligado;
- (II) Posicionar um primeiro sensor em situação de eminência ao carrinho, de modo que quando iniciado o movimento ele dispare o tempo instantaneamente;
- (III) Posicionar um segundo sensor distando 60 cm do primeiro;
- (IV) Liberar o carrinho para movimento desligando o disparador magnético;

- (V) Por fim, observar o tempo gasto pelo carrinho no percurso no display do cronômetro digital.

Este procedimento é ilustrado na Figura 6.8.

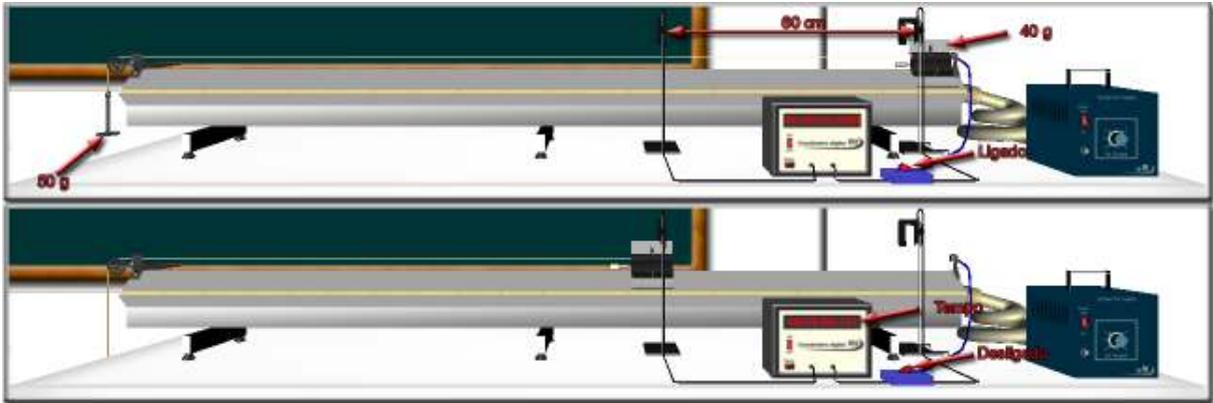


Figura 6.8: Ilustração da experimentação de um problema.

Ao realizar o experimento o aluno pode observar uma pequena diferença entre o resultado obtido pelas equações matemáticas e resultado experimental. Esta diferença pode advir de pequenos erros de medição ao posicionar os sensores, de pequenos atrasos na parada ou acionamento do tempo no cronômetro, etc. Realizando o mesmo experimento mais de uma vez ele pode verificar que os seus resultados podem ser diferentes mesmo em situações parecidas de experimentação. Tais vivências são importantes para o estudante, levando-o a compreender melhor o funcionamento da teoria no mundo real, tornando suas abstrações um pouco mais próximas do concreto.

Quanto à dificuldade dos alunos em associar o problema à montagem experimental do trilho virtual pôde-se observar que os estudantes das amostras β e γ apresentaram um desempenho médio acima de 60%, como mostra o gráfico da Figura 6.9.

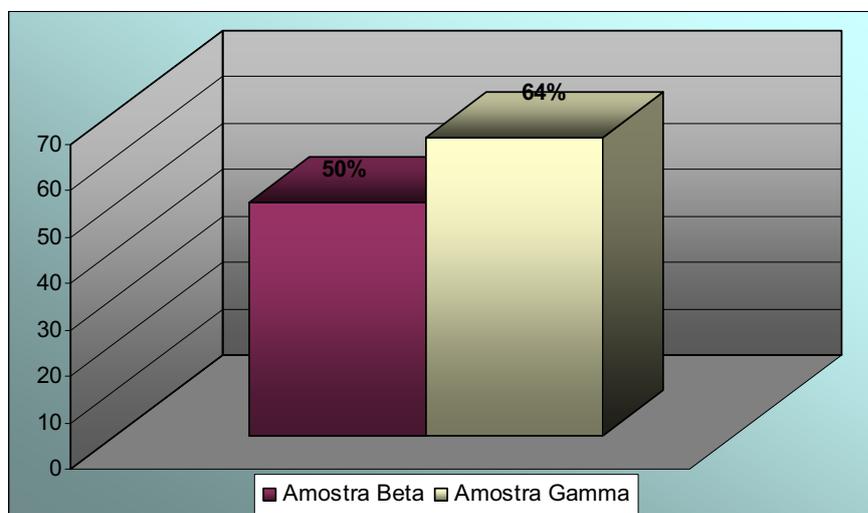


Figura 6.9: Gráfico sobre o desempenho dos alunos em montar os experimentos.

Estes valores apresentados no gráfico foram inferidos com base em uma descrição pedida ao aluno, para cada problema, do processo de montagem do experimento no trilho virtual que ele realizara no momento de resolução do problema, como se observa no último item do Anexo 4. Neste contexto, pode-se observar que a amostra β obteve uma média de acertos nesta descrição de 50% e a amostra γ obteve um percentual um pouco mais alto de 64%.

No capítulo 2 foi apresentado o modelo de resolução de problemas para o LVCF, o qual delimita três operações não-lineares, que são: *contato inicial*, *solução* e *avaliação*. O *contato inicial*, por sua vez, engloba a *identificação do problema*, a *representação mental do problema* e a *geração de hipóteses*, a *solução* inclui a *busca e exploração de estratégias*, o *projeto de solução*, a *previsão de resultados*, a *aplicação* e o *gerenciamento da aplicação*, por fim, *avaliação* engloba a *crítica da solução* e a *entrega da solução*. Realizou-se então uma investigação junto ao aluno em cada um dos problemas aplicados em relação a estas etapas do modelo.

Na *identificação do problema* foram requisitados ao aluno quais os dados fornecidos pelo problema e o que o problema esperava como resposta. Estes questionamentos visam verificar se o aluno conseguiu interpretar corretamente o problema e absorver detalhes específicos em relação ao seu enunciado. O desempenho dos alunos, referente ao número de acertos destes questionamentos, no problema apresentado nesta seção pode ser observado no gráfico da Figura 6.10.

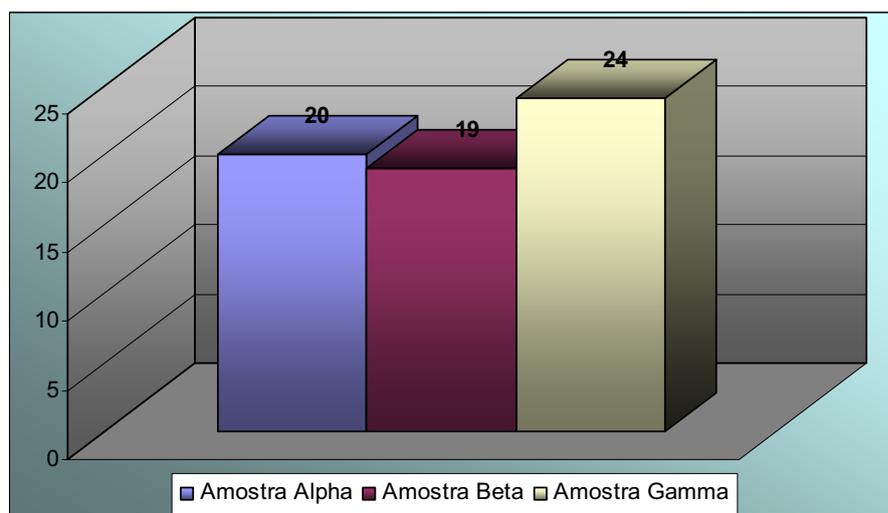


Figura 6.10: Gráfico da etapa de identificação do problema referente ao problema exemplo.

Ao analisar o gráfico pode-se perceber que as amostras α e β obtiveram desempenhos próximos e a amostra γ conseguiu um nível de aproveitamento mais elevado que ambas.

Na etapa de *representação mental do problema* analisou-se os conhecimentos prévios do aluno sobre os dados retirados na etapa anterior, requisitando a ele que descrevesse as definições, conceitos, unidades de medida e correlação entre os mesmos. Os resultados obtidos encontram-se no gráfico ilustrado na Figura 6.11.

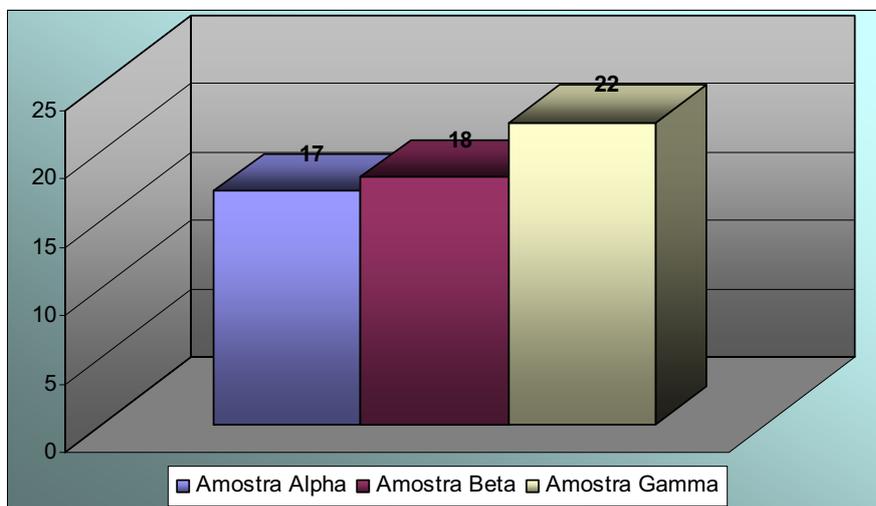


Figura 6.11: Gráfico da etapa de representação mental do problema referente ao problema exemplo.

Pode-se observar no gráfico que as amostras α e β apresentaram desempenhos similares e a amostra γ obteve melhores resultados nesta etapa.

Na etapa de *geração de hipóteses* o aluno foi indagado sobre o que ele precisaria encontrar para chegar até a solução final do problema, visando analisar a sua capacidade de reconhecimento de hipóteses no problema. Os números de acertos por amostra, no problema exemplo, podem ser observados no gráfico ilustrado na Figura 6.12.

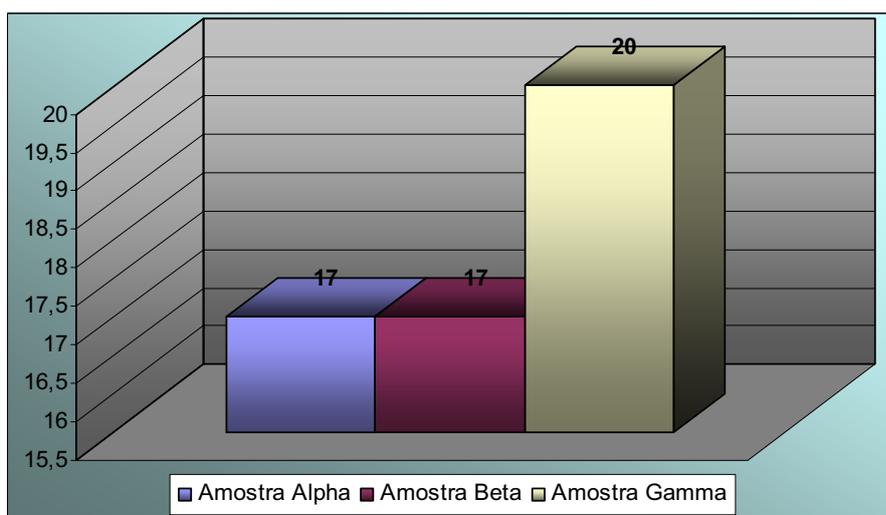


Figura 6.12: Gráfico da etapa de geração de hipóteses referente ao problema exemplo.

Neste gráfico pode-se perceber que as amostras α e β obtiveram o mesmo resultado e a amostra γ atingiu um resultado melhor que ambas.

Na etapa de *busca e exploração de estratégias*, o aluno foi questionado sobre quais teorias e equações ele deveria utilizar para encontrar as hipóteses anteriores. Este questionamento visa avaliar os conhecimentos prévios do aluno relacionados às metas que ele deverá cumprir. A Figura 6.13 ilustra o gráfico com os resultados referentes a esta etapa.

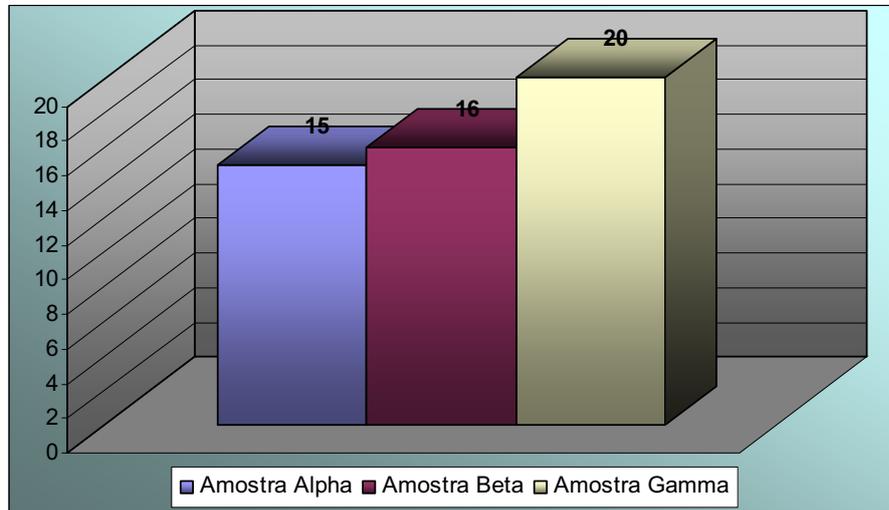


Figura 6.13: Gráfico da etapa de busca e exploração de estratégias do problema exemplo.

Observa-se no gráfico que as amostras α e β obtiveram resultados compatíveis e a amostra γ atingiu um resultado um pouco melhor que ambas.

Na etapa de *projeto de solução*, foi pedido ao aluno que descrevesse um plano de solução para o problema por meio de uma seqüência de passos, visando analisar a sua capacidade de organização, classificação e planejamento. A Figura 6.14 apresenta o gráfico referente a estes resultados.

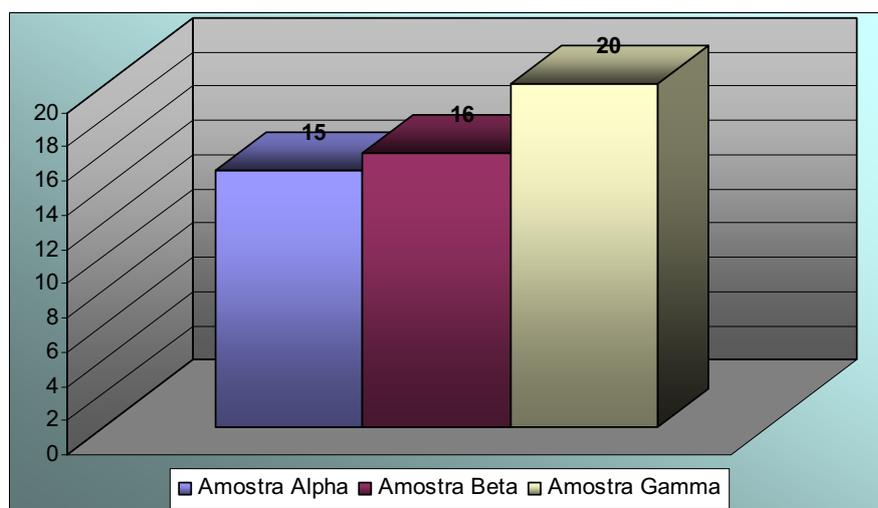


Figura 6.14: Gráfico da etapa de projeto e solução referente ao problema exemplo.

Pode-se observar no gráfico que as amostras α e β obtiveram resultados similares e a amostra γ conseguiu um desempenho melhor que ambas.

Na etapa de *previsão de resultados*, questionou-se ao aluno o que ele prevê encontrar em cada passo estabelecido no projeto de solução. Este questionamento visou identificar se o aluno estava consciente das possíveis conseqüências das escolhas feitas por ele durante o projeto. Os resultados desta etapa são apresentados no gráfico da Figura 6.15.

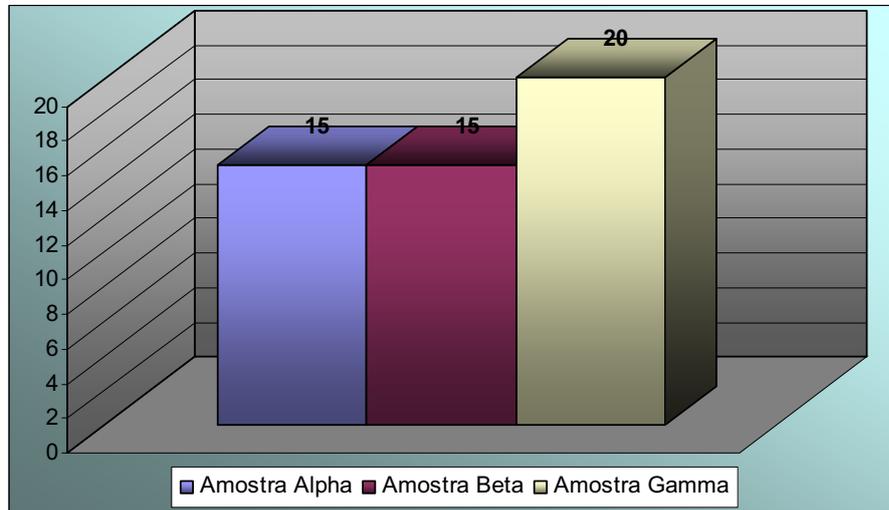


Figura 6.15: Gráfico da etapa de previsão de resultados referente ao problema exemplo.

Neste gráfico pode-se perceber que as amostras α e β obtiveram o mesmo resultado e a amostra γ atingiu um resultado melhor que ambas.

Na etapa de *aplicação* foi solicitado ao aluno que ele resolvesse o problema, colocando em prática o plano de solução traçado. A Figura 6.16 apresenta o gráfico relativo ao desempenho dos alunos nesta etapa.

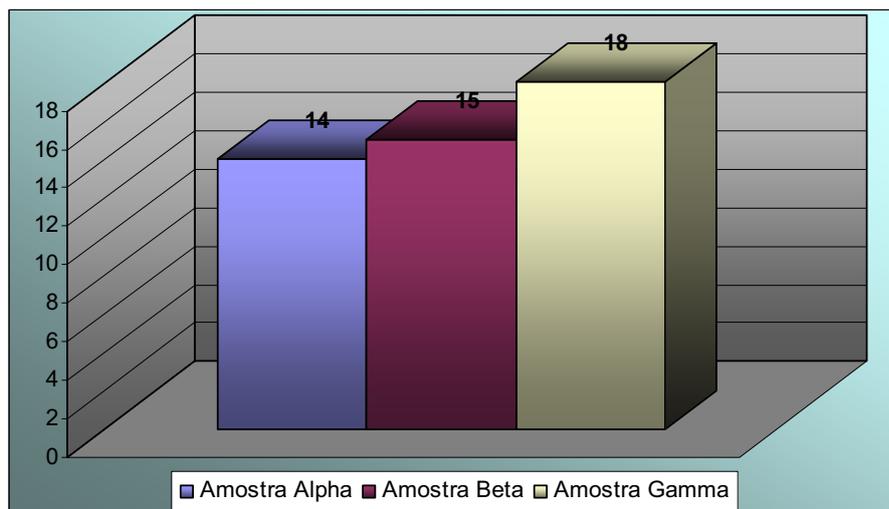


Figura 6.16: Gráfico da etapa de aplicação referente ao problema exemplo.

Observa-se no gráfico que as amostras α e β obtiveram resultados próximos e a amostra γ um desempenho um pouco acima de ambas.

Na etapa de *gerenciamento da aplicação* o aluno foi questionado sobre a existência de outras formas de resolver o problema, nas quais se alteram alguns dos passos do plano de solução e chega-se ao resultado com maior rapidez e eficiência. Os resultados desta etapa são ilustrados no gráfico da Figura 6.17.

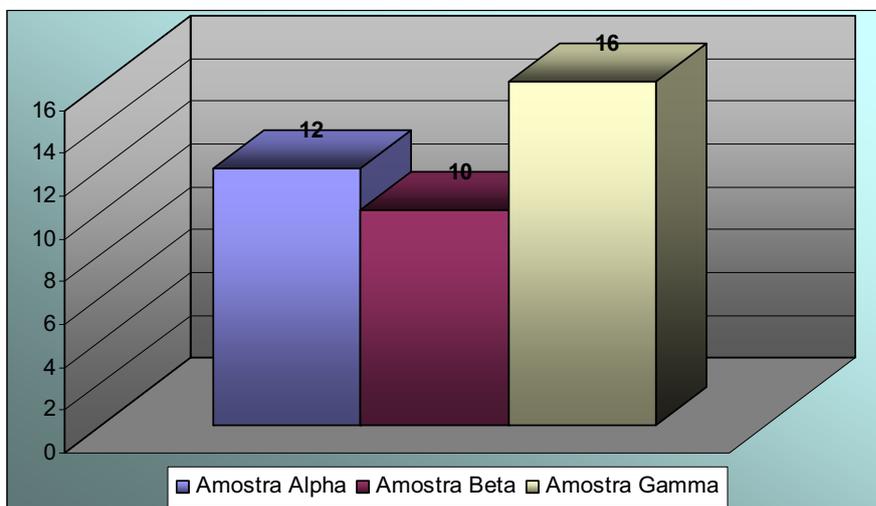


Figura 6.17: Gráfico da etapa de gerenciamento da aplicação referente ao problema exemplo.

No gráfico pode-se observar que a amostra β obteve o menor resultado, a qual é seguida proximamente da amostra α . A amostra γ obteve o melhor resultado.

Na etapa de *crítica da solução* perguntou-se ao aluno se ele conseguiu chegar ao seu objetivo. Em caso de resposta afirmativa, questionou quais ações ele tomou que poderiam ser consideradas erradas e que deveriam ser evitadas em uma próxima resolução para melhorar a qualidade da solução. Em caso de resposta negativa, questionou-se a ele quais foram os erros cometidos que o impediram de chegar aos objetivos do plano de solução. A Figura 6.18 apresentam o gráfico com os resultados desta etapa.

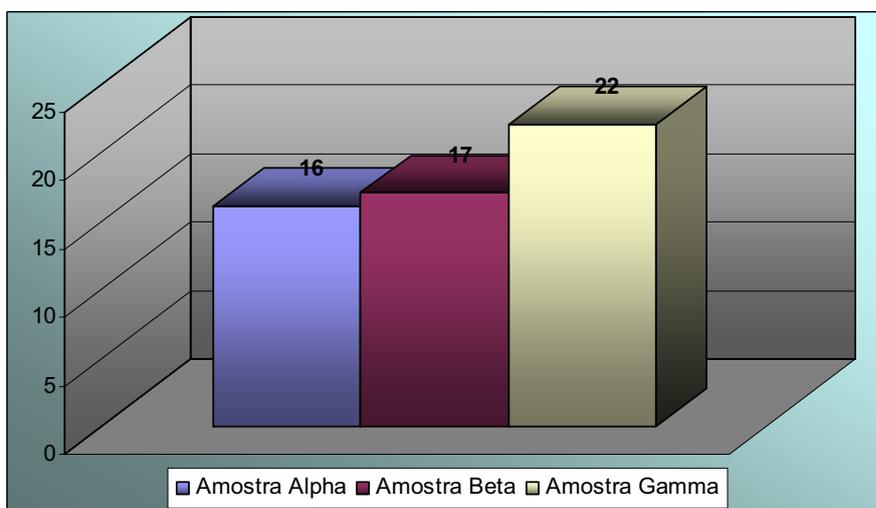


Figura 6.18: Gráfico da etapa de crítica da solução referente ao problema exemplo.

Observa-se no gráfico que as amostras α e β obtiveram resultados próximos e a amostra γ um desempenho um pouco acima de ambas.

Por fim, na etapa de *entrega da solução* pediu-se ao aluno que escrevesse um resumo explicativo da solução em tópicos. Os resultados obtidos nesta etapa são apresentados no gráfico ilustrado na Figura 6.19.

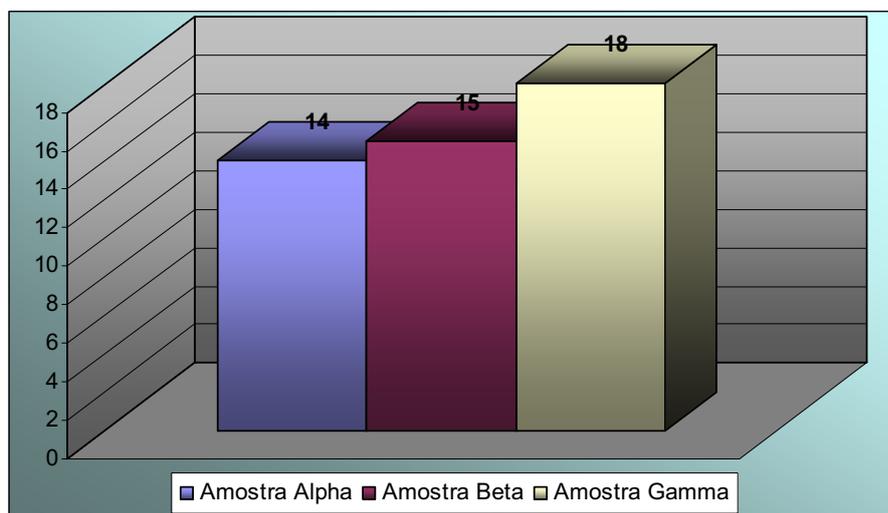


Figura 6.19: Gráfico da etapa de entrega da solução referente ao problema exemplo.

Observa-se no gráfico que as amostras α e β obtiveram resultados próximos e a amostra γ um desempenho um pouco acima de ambas.

O procedimento de questionamento para avaliar as etapas do modelo de resolução proposto em cada problema aplicado foi seguido, de uma maneira padrão, como o apresentado nesta seção, sendo que, as arguições foram organizadas de acordo a ficha, referente ao problema exemplo, apresentada no Anexo 4. A próxima seção aborda os resultados da avaliação observando os níveis de problemas aplicados.

6.5 Resultados por níveis de problemas e geral da avaliação do LVCF

Para possibilitar a avaliação da evolução do aluno em resolver problemas no decorrer do processo avaliativo foi realizada uma classificação dos problemas desenvolvidos em termos de níveis de dificuldade de solução. Quanto maior o número de conceitos necessários para a resolução e elevada a complexidade do plano de solução maior será considerado o seu nível de dificuldade do problema.

A análise dos resultados da avaliação de acordo com os níveis de dificuldade dos problemas permite verificar comparativamente o quanto cada amostra aproveitou, de maneira benéfica, o processo de resolução para aprimorar os seus conhecimentos e a sua capacidade de generalização e de aplicação de estratégias e teorias. Pozo (1998) acredita que a partir de problemas mais simples o aluno pode ativar o seu conhecimento para desenvolver estratégias mais elaboradas e resolver problemas de maior complexidade [Pozo *et al.*, 1998]. Sendo assim, intui-se observar como estaria ocorrendo este processo de ativação durante o processo de avaliação do LVCF. Consequentemente visa-se observar as contribuições, para o aprimoramento do aluno na resolução de problemas, de cada técnica educacional aplicada às amostras, as quais são: o modo tradicional de resolver problemas, sem apoio tecnológico e sem cooperação; o uso da tecnologia de RV; e a utilização de RV cooperativamente.

Os problemas desenvolvidos foram classificados em três níveis de dificuldade, o *nível 1*, *2* e *3*, que caracterizam crescentemente a complexidade dos problemas utilizados no processo de avaliação. O *nível 1* compõe-se dos seguintes problemas:

- (1º) “Um bloco A de 40 g está conectado a um bloco B de 50 g por meio de um fio, como ilustra a Figura 6.7. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine o tempo que o bloco A gasta para percorrer 60 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.
- (2º) “Um bloco A de 60 g está conectado a um bloco B de 470 g por meio de um fio, ilustração similar a Figura 6.7. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine o tempo que o bloco A gasta para percorrer 80 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.
- (3º) “Um bloco A de 80 g está conectado a um bloco B de 400 g por meio de um fio, ilustração similar a Figura 6.7. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine a distância que o bloco A percorre em 0,41 s. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.

Os resultados percentuais referentes ao desempenho dos alunos para o *nível 1*, em todas as etapas do modelo de resolução de problemas proposto, encontram-se no gráfico ilustrado na Figura 6.20.

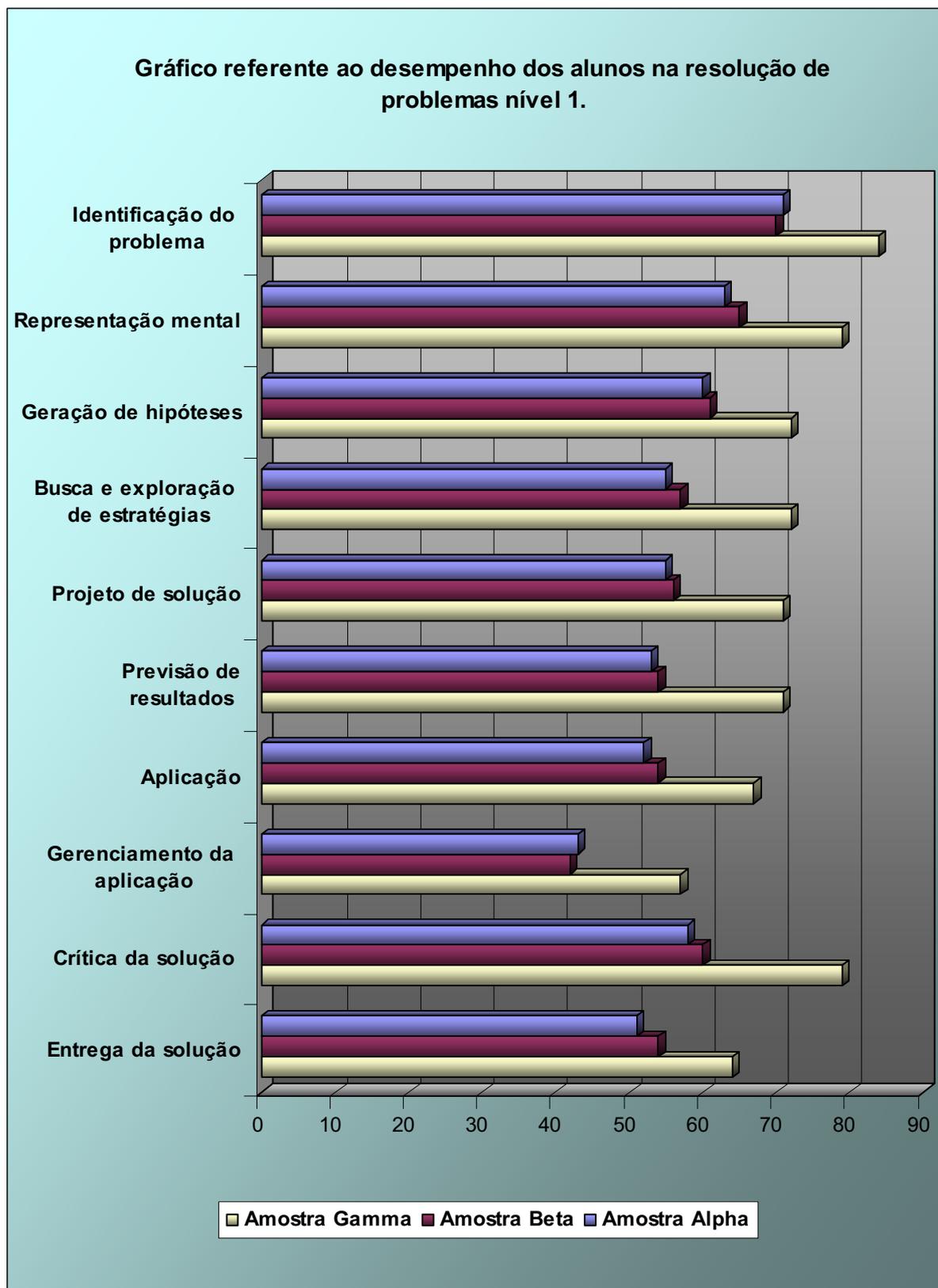


Figura 6.20: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 1.

Pode-se observar no gráfico que a amostra γ apresenta melhores resultados em todas as etapas do modelo, já as amostras α e β obtiveram desempenhos próximos, sendo que em algumas etapas α conseguiu um resultado um pouco melhor que β e vice-versa.

Os problemas do *nível 2* possuem uma complexidade um pouco maior que os problemas do *nível 1*. Estes problemas são:

- (4º) “Um bloco A de 40 g está conectado a um bloco B de 50 g por meio de um fio, como ilustra a Figura 6.7. O sistema possui uma velocidade inicial de 1,25 m/s. Determine o tempo que o bloco A gasta para percorrer 65 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.
- (5º) “Um bloco A de 60 g está conectado a um bloco B de 500 g por meio de um fio, como ilustra a Figura 6.7. O sistema possui uma velocidade inicial de 1,75 m/s. Determine o tempo que o bloco A gasta para percorrer 52,5 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.
- (6º) “Um bloco A de 80 g está conectado a um bloco B de mesma massa por meio de um fio, ilustração similar a Figura 6.7. O sistema possui uma velocidade inicial de 1,715 m/s. Determine a distância que o bloco A percorre em 0,2 s. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.

Os resultados percentuais referentes ao desempenho dos alunos para o *nível 2*, em todas as etapas do modelo de resolução de problemas proposto, encontram-se no gráfico ilustrado na Figura 6.21.

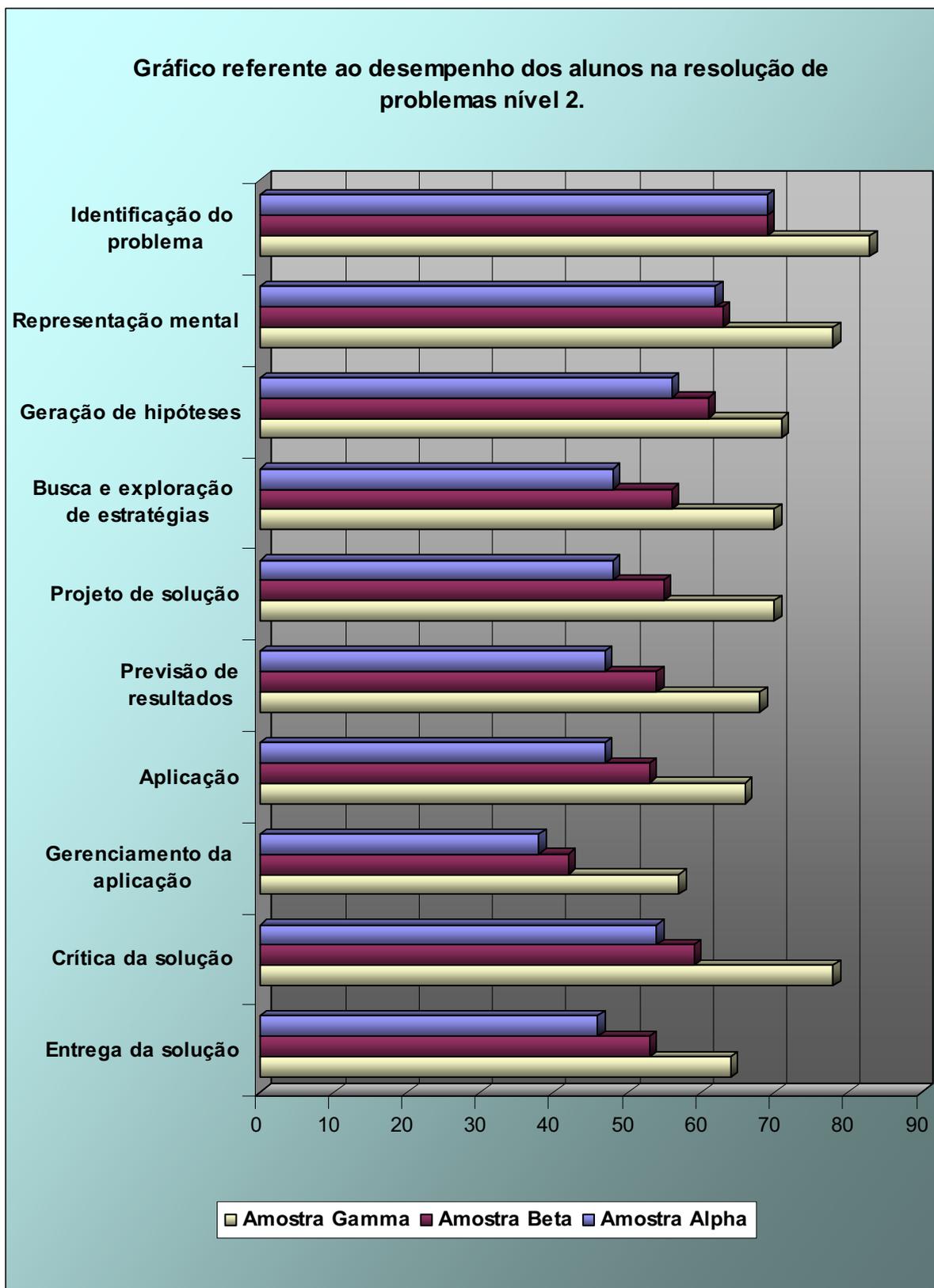


Figura 6.21: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 2.

Observa-se no gráfico que a amostra γ apresenta melhores resultados que α e β em todas as etapas do modelo. A amostra β obteve um desempenho um pouco acima que a amostra α também em todas as etapas.

Por fim, os problemas do *nível 3* possuem uma complexidade um pouco maior que os problemas do *nível 2*. Estes problemas são:

- (7°) “Um bloco A de 40 g está conectado a um bloco B de 50 g por meio de um fio, sendo que o bloco B encontra-se a 82 cm do solo, como ilustra a Figura 6.22. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine o tempo que o bloco A gasta para percorrer 1,60 m. Lembre-se que quando o bloco B tocar o solo a força que ele exercia em A cessará e deste modo A passa a se mover com uma velocidade constante. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.

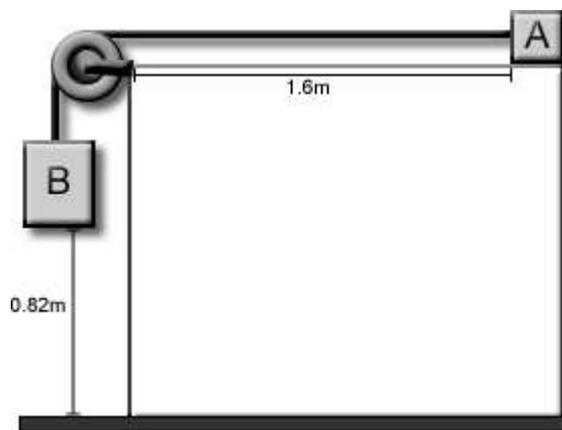


Figura 6.22: Ilustração do problema do nível 3.

- (8°) “Um bloco A de 60 g está conectado a um bloco B de 500 g por meio de um fio, sendo que o bloco B encontra-se a 64,5 cm do solo, como ilustra a Figura 6.22. O sistema possui uma velocidade inicial de 1,75 m/s. Determine o tempo que o bloco A gasta para percorrer 1,50 m. Lembre-se que quando o bloco B tocar o solo a força que ele exercia em A cessará e deste modo A passa a se mover com uma velocidade constante. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.
- (9°) “Um bloco A de 80 g está conectado a um bloco B de 570 g por meio de um fio, sendo que o bloco B encontra-se a 30 cm do solo, como ilustra a Figura 6.22. O sistema possui uma velocidade inicial de 2,27 m/s. Determine a distância que o bloco A percorre em 0,4 s. Lembre-se que quando o bloco B tocar o solo a força que ele exercia em A cessará e deste modo A passa a se mover com uma velocidade constante. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.”.

Os resultados percentuais referentes ao desempenho dos alunos para o *nível 3*, em todas as etapas do modelo de resolução de problemas proposto, encontram-se no gráfico ilustrado na Figura 6.23.

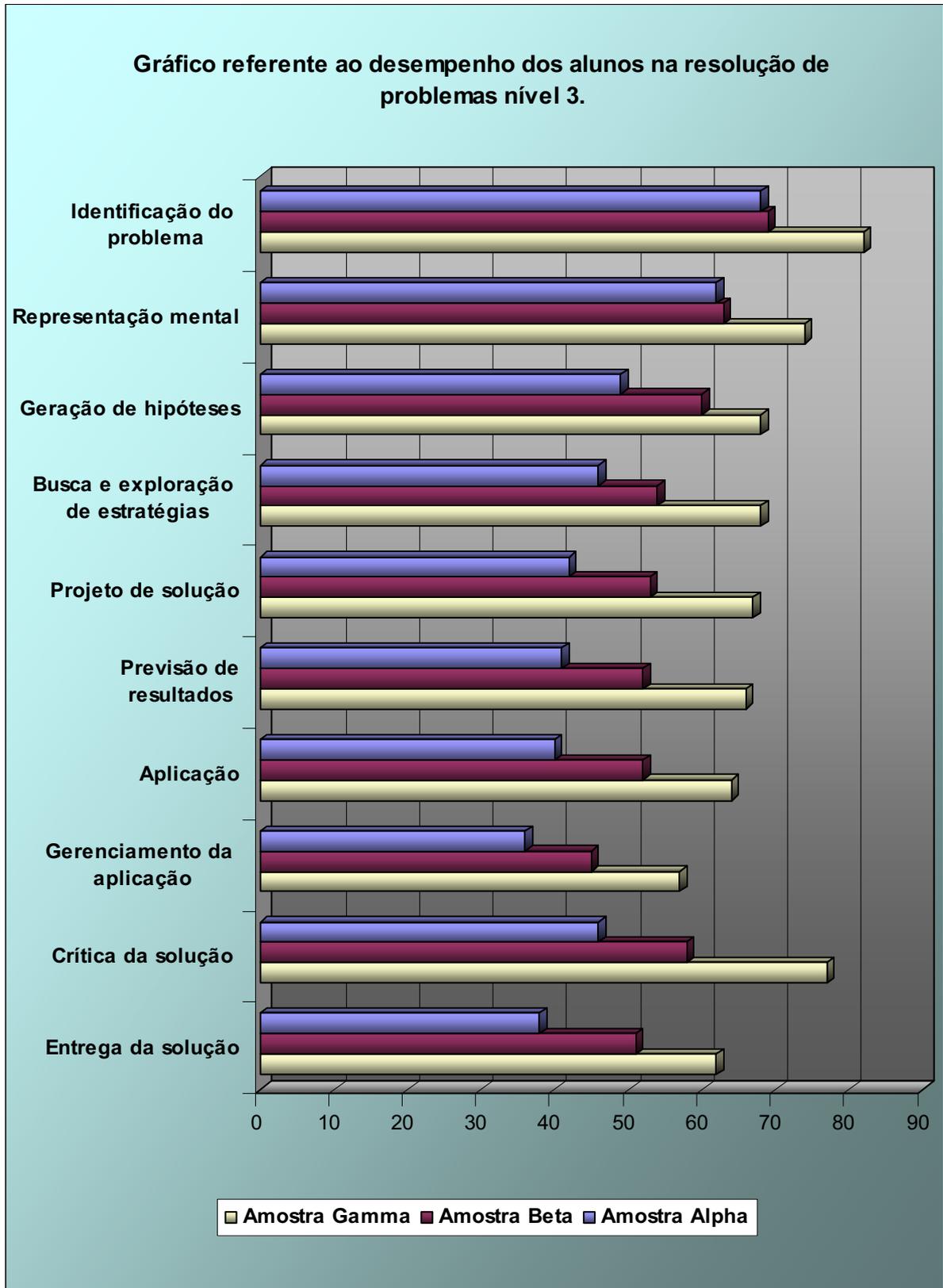


Figura 6.23: Gráfico referente ao desempenho dos alunos na resolução de problemas nível 3.

Pode-se perceber no gráfico que a amostra γ apresenta melhores resultados que α e β em todas as etapas do modelo. A amostra β obteve um desempenho acima que a amostra α também em todas as etapas.

Analisando os três gráficos dos níveis 1, 2 e 3 observa-se que a amostra α teve uma perda de rendimento maior de acordo com o aumento da complexidade dos problemas, comparada as amostra β e γ , na maioria das etapas do modelo de resolução, com exceção das etapas de *identificação do problema* e de *representação mental*. A amostra β atingiu perdas menores que α , mantendo-se um pouco mais estável o seu desempenho. A amostra γ também obteve perdas de desempenho relativamente pequenas e não foi superada em nenhuma das etapas pelas outras amostras.

Durante a aplicação dos testes os alunos das amostras β e γ demoram um tempo considerável maior que α para finalizar as questões. Para cada problema foi estabelecido um prazo de duas horas como tempo limite para a sua resolução, as amostras β e γ normalmente gastavam quase todo este tempo, já a amostra α , em média, consumia a metade do tempo limite.

Deve-se ressaltar que os problemas desenvolvidos tiveram como objetivo servirem de suporte para a avaliação do LVCF, neste contexto eles visaram facilitar a avaliação das etapas e tornar visível como estava ocorrendo à evolução do aluno durante o processo avaliativo. O formato dos problemas teve como intuito ainda permitir e instigar uma associação do problema com o experimento virtual e deste modo motivar o aluno a usar o ambiente em RV na sua aprendizagem.

O gráfico ilustrado na Figura 6.24 apresenta o percentual de desempenho dos alunos de maneira geral, considerando agora o conjunto de resultados dos nove problemas aplicados.

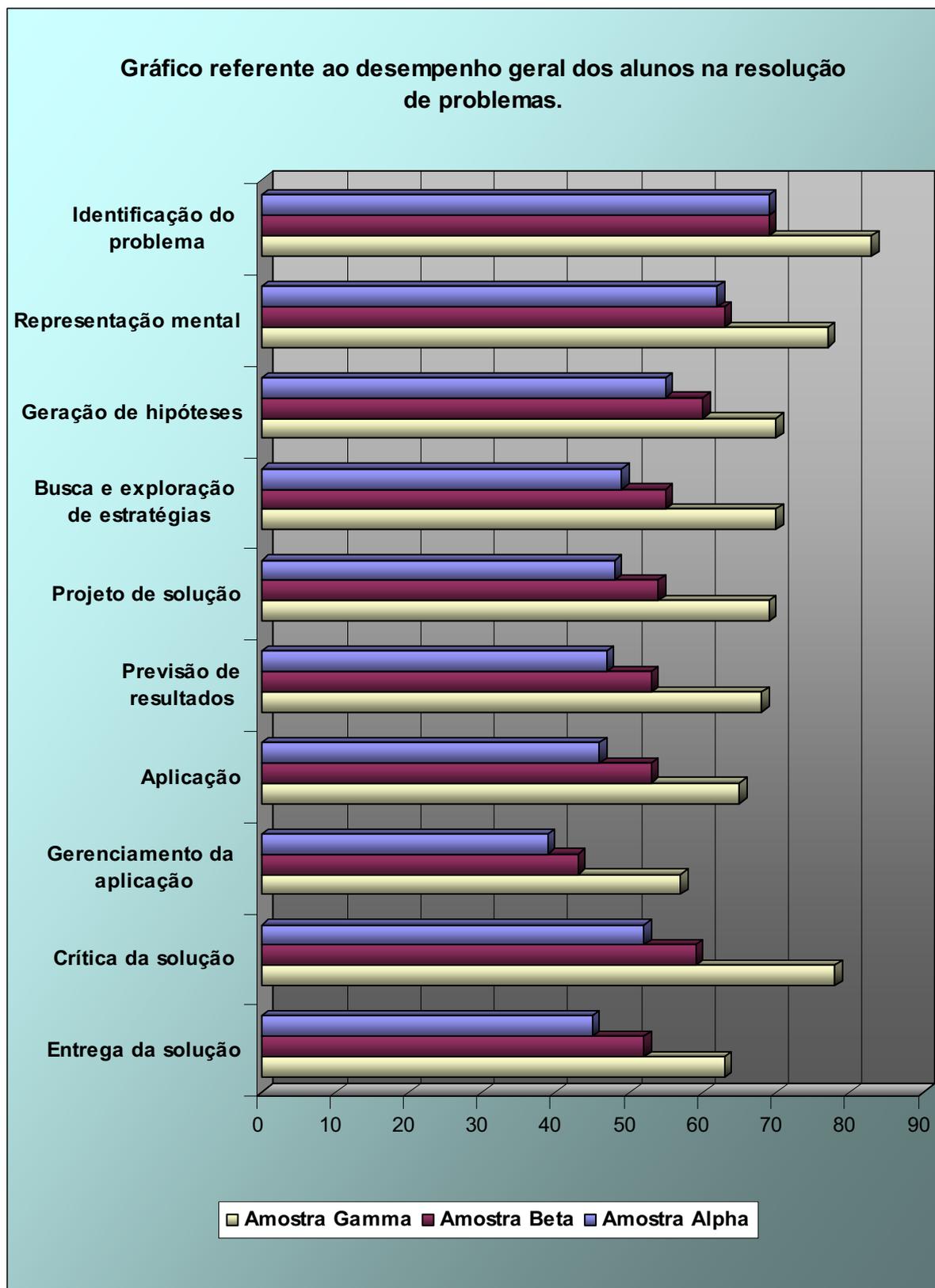


Figura 6.24: Gráfico referente ao desempenho geral dos alunos na resolução de problemas.

Pode-se observar no gráfico que a amostra γ manteve o seu desempenho consideravelmente acima das outras duas amostras em todas as etapas do modelo resolução de problemas. A amostra β , por sua vez, obteve resultados sempre acima de α , no entanto, a

diferença entre os seus valores é pouco expressiva em algumas etapas. A seção a seguir apresenta as considerações finais deste capítulo.

6.6 Considerações Finais

Neste capítulo foi apresentada a metodologia de avaliação do LVCF, a qual visou produzir um processo avaliativo com resultados mais confiáveis e representativos em relação às metodologias pedagógicas aplicadas.

Como resultado obteve-se que a amostra γ que trabalhou com resolução de problemas por intermédio da aprendizagem cooperativa no ambiente de RV conseguiu um melhor desempenho que as amostras α e β , que trabalharam de forma individual sem o auxílio tecnológico e com apoio do ambiente LVD em RV para a resolução de problemas, respectivamente. As amostras que utilizaram os ambientes em RV conseguiram melhores resultados durante resolução de problemas mais complexos.

A próxima seção apresenta as conclusões deste trabalho e algumas sugestões para trabalhos futuros em relação a esta pesquisa.

Capítulo 7

Conclusões e trabalhos futuros

7.1 Introdução

Esta tese descreveu uma pesquisa realizada de forma multidisciplinar (Engenharia Elétrica, Educação e Física) que engloba estudos sobre técnicas de Realidade Virtual e processos pedagógicos em ambientes computacionais. O presente capítulo traz algumas conclusões a respeito deste trabalho e sugestões para possíveis trabalhos futuros.

7.2 Conclusões

No decorrer desta pesquisa, em um contexto geral, observou-se que a educação está longe de ser considerada ideal na sociedade contemporânea e, sendo assim, conclui-se o quanto é necessário o surgimento de novas metodologias de ensino e tecnologias educacionais que contribuam para a qualificação do ensino. Em específico, na educação brasileira existe um déficit considerável na proficiência dos alunos em resolver problemas [Oecd, 2004].

Ao analisar projetos nas áreas de pesquisa da resolução de problemas, aprendizagem cooperativa e RV pode-se perceber certa carência de trabalhos com focos nas interseções destas áreas. Neste contexto, a presente pesquisa se dispôs a gerar estudos de cunho teórico e tecnológico para criar alternativas de qualidade que contribuam para o processo ensino e aprendizagem em grande extensão nas escolas.

Dessa forma, a integração dos estudos teóricos sobre desenhos pedagógicos relacionados à cooperação e solução de problemas aliados à realidade virtual, possivelmente possam contribuir com os objetivos educacionais mais amplos de estimular o pensamento formal e a capacidade de resolução de problemas de forma cooperativa.

Assim, criou-se um *software* educacional que utiliza técnicas computacionais de ambientes em RV distribuída para dar suporte à experimentação em laboratórios virtuais para o ensino de Física, a área que se compõe como o estudo de caso desta pesquisa. Este sistema foi fundamentado em desenhos pedagógicos contemporâneos para ambientes computacionais, que prevêem cenários virtuais compostos por objetos de aprendizagem, os quais objetivam estimular o conhecimento conceitual e o procedimental, bem como a competência

metacognitiva de solucionar problemas de forma cooperativa. Ressalta-se que a arquitetura geral proposta neste trabalho foi de suma importância para que as diferentes áreas de pesquisa tratadas no projeto (RV, aprendizagem cooperativa e resolução de problemas) coexistissem e interagissem de maneira eficiente em um *software* educacional.

Afirma-se que o produto final do sistema obteve melhoras significativas devido às contribuições da equipe multidisciplinar. A equipe da Educação contribuiu com a arquitetura e os desenhos pedagógicos do cenário e dos objetos de aprendizagem que o compõe, e com o modelo de resolução de problemas utilizado para avaliar o projeto. A equipe da Física ofereceu um suporte conceitual para o sistema e estratégias para a criação dos objetos virtuais e simulações, e ainda contribuíram para o desenvolvimento dos problemas avaliados.

Apesar de o projeto proposto focalizar o uso do sistema desenvolvido como ferramenta para a aprendizagem cooperativa, percebe-se na sua arquitetura e suas disposições finais, que ele pode ser utilizado como apoio tanto para a aprendizagem cooperativa quanto para a colaborativa, visto que a sua aplicação pode ocorrer de forma síncrona e assíncrona. Por exemplo, os aprendizes podem utilizar o sistema sincronicamente conectando os seus ambientes virtuais distribuídos, que funcionam em tempo real, e trocar seus conhecimentos de forma imediata por meio do *chat*. Outra possibilidade existente seria os estudantes executarem os experimentos virtuais de modo assíncrono, com o uso atemporal e individual dos mesmos, e utilizarem posteriormente o *fórum* para postarem suas dúvidas, resultados, sugestões, etc., realizando, desta forma, o compartilhamento de informações entre os aprendizes do grupo colaborativo.

Deve-se ressaltar que um sistema é considerado síncrono quando um grupo de usuários tem a sensação que as atualizações em seus ambientes computacionais, mesmo sendo executados em computadores diferentes, ocorrem ao mesmo tempo, ou seja, em tempo real. Porém, destaca-se que mesmo em sistemas síncronos garante-se apenas a sensação da ocorrência em tempo real, visto que é fisicamente impossível garantir que um evento disparado em um computador seja percebido por ele e por outros computadores no mesmo espaço cronológico de tempo. Isto se deve ao tempo gasto para transmitir uma informação de um computador para outro, que acarreta conseqüentemente um atraso nas atualizações dos computadores comunicantes, mesmo que seja mínimo. No entanto, quanto menor este atraso melhor será a qualidade do sincronismo dos ambientes cooperativos. Os principais fatores que preponderam esta qualidade são as ferramentas de comunicação utilizadas, a arquitetura de distribuição do sistema e os algoritmos de envio e recebimento de informações.

Neste sentido, utilizou-se a ferramenta Java sockets para realizar a comunicação entre os ambientes virtuais distribuídos, devido a sua performance em relação às demais ferramentas. A arquitetura do sistema e os algoritmos de envio e recebimentos desenvolvidos garantiram que a quantidade de informações transmitida fosse minimizada, tornando o tráfego na rede mais rápido e fluente.

Estes aspectos tecnológicos permitiram estabelecer um conjunto de diretrizes para o desenvolvimento de aplicações distribuídas em VRML e X3D de qualidade. Estas linguagens são pouco pesquisadas no campo da RV distribuída devido ao fato de serem estáticas e baseadas em eventos, o que dificulta significativamente a transferência e atualização de parâmetros entre ambientes virtuais comunicantes.

Os ambientes distribuídos desenvolvidos (LVD) apresentaram um alto desempenho e mostraram-se adequados para a efetivação dos testes em relação à aprendizagem cooperativa. Porém, para obter informações mais completas, concretas e contundentes sobre a qualidade da arquitetura e dos algoritmos de distribuição propostos neste projeto, faz-se necessária a realização de uma avaliação específica quanto aos parâmetros qualitativos de uma distribuição, como por exemplo, desempenho, escalabilidade, tolerância a falhas, segurança, transparência, etc.

A linguagem VRML foi apropriada para modelar os mundos virtuais, porém para incluir interações, animações e realizar cálculos foi preciso utilizar uma linguagem de programação específica que contemplasse os comandos (JavaScript). A linguagem Java foi utilizada para viabilizar distribuição dos ambientes. O uso deste conjunto de linguagens garantiu a possibilidade de desenvolver e utilizar um sistema portátil sem custos de aquisição de *softwares* e *hardwares* adicionais.

Quanto à avaliação pedagógica do sistema observou-se que, as técnicas estatísticas aplicadas para a subdivisão em amostras, antes da aplicação efetiva dos testes, garantiram uma maior confiabilidade nos resultados e possibilitaram a comparação dos mesmos. A estratégia de aplicar testes para três amostras igualitárias estatisticamente, focalizando uma avaliação sem acesso a tecnologias, outra com acesso individual ao ambiente em RV e outra utilizando o ambiente em RV cooperativamente, permitiu analisar o quanto a tecnologia e a cooperação contribuíram no processo de resolução de problemas.

Como resultado obteve-se que a amostra que trabalhou com resolução de problemas cooperativamente no ambiente de RV conseguiu um melhor desempenho que as outras duas

amostras, que trabalharam de forma individual sem o auxílio tecnológico e com apoio do ambiente LVD em RV para a resolução de problemas. As amostras que utilizaram os ambientes em RV conseguiram melhores desempenhos durante o aumento da complexidade dos problemas, o que pode indicar que a RV contribuiu para estas amostras ativarem os seus conhecimentos para desenvolverem estratégias mais elaboradas e resolverem problemas de maior complexidade.

Ressalta-se que o ambiente virtual deste projeto foi desenvolvido com base em princípios construtivistas. Em consonância com os resultados da avaliação pode-se inferir que estes princípios contribuíram significativamente para o bom desempenho dos aprendizes que utilizaram a tecnologia RV. Acredita-se que isto ocorre devido ao fato de o aluno ter que realizar conexões cognitivas e raciocínios lógicos aprofundados, em relação ao conteúdo estudado, para conseguir construir, executar e compreender o experimento virtual. Tais conexões permanecem em sua estrutura mental facilitando a busca de estratégias e elaboração de planos de solução, e contribuindo conseqüentemente, para o processo de resolução de problemas.

Destaca-se também que o ambiente virtual possibilitou aos estudantes entrarem em contato com objetos virtuais com um alto nível de fidelidade a instrumentos laboratoriais reais, que não foram nem mesmo vistos pela maioria deles. Estes alunos tiveram a oportunidade de realizarem experiências e observarem como a teoria pode ser aplicada na prática, viabilizando um aprendizado mais rico e significativo.

Quanto ao modelo de resolução de problemas proposto, por ser mais completo, permitiu analisar a resolução de uma maneira mais ampla. Devido a este fato, os resultados desta avaliação possibilitam a inferência do desempenho dos alunos usando o foco de outros modelos menos gerais.

E finalmente, é possível verificar que esta pesquisa contribuiu para a melhoria do entendimento sobre o uso das técnicas de Realidade Virtual em ambientes distribuídos para prover princípios da aprendizagem cooperativa e auxiliar na resolução de problemas e, conseqüentemente, nos processos de ensino e aprendizagem. A próxima seção apresenta os trabalhos futuros do presente projeto.

7.3 Trabalhos Futuros

Como trabalhos futuros, que compõem possíveis propostas para a continuação deste trabalho, pode-se sugerir:

- O desenvolvimento da estrutura de comunicação do LVD por meio do protocolo UDP, com o sincronismo realizado por *threads*, para avaliação de performance;
- A implementação do sistema de distribuição do LVD utilizando *multithreads* com intuito de viabilizar a conexão de vários clientes;
- A implementação da estrutura de distribuição do LVD por meio de outras tecnologias de comunicação, como por exemplo, Java RMI, CORBA, etc.
- A realização de testes no sistema visando análises específicas quanto a parâmetros de distribuição, como: o desempenho, a escalabilidade, a conectividade, a segurança, a confiabilidade, a tolerância à falhas e a transparência. Apenas através destes testes que poderão ser tomadas conclusões efetivas em relação à eficiência da arquitetura de distribuição proposta. Vale ressaltar ainda que tais testes não foram realizados nesta pesquisa por que este trabalho focaliza diretamente os fatores pedagógicos e cognitivos que os ambientes distribuídos podem influenciar em um aluno, e não os seus fatores tecnológicos. Porém, ressalta-se que a arquitetura de distribuição do LVD pode contribuir e servir como ponto de partida para tais estudos;
- O desenvolvimento de novos objetos virtuais de aprendizagem e novos laboratórios visando à disponibilidade de uma nova gama de experimentos, aumentando conseqüentemente a quantidade de problemas abrangida pelos ambientes virtuais;
- A criação de um sistema computacional de histórico para os laboratórios virtuais, o qual permita armazenar as ações do usuário assim como as respostas do ambiente a estas ações. Este histórico terá o objetivo de servir como suporte mais completo para as análises pedagógicas do professor;
- O desenvolvimento de um sistema computacional de avaliação e acompanhamento do aluno (tutor/assistente inteligente), que utilize técnicas de

inteligência artificial para realizar análises sobre o seu aprendizado e lhe fornecer dicas, por meio de um agente inteligente, de acordo com suas ações no ambiente virtual, visando facilitar a sua aprendizagem;

- A criação de uma base de problemas categorizados por complexidade e utilizar um tutor/assistente inteligente e técnicas de adaptabilidade para selecionar os exercícios a partir do perfil do aluno e seu desempenho;
- A criação de avatares individuais para representar os alunos dentro de ambientes virtuais distribuídos;
- A avaliação deste projeto junto a uma amostra maior de indivíduos, com intuito de aumentar a confiança estatística nos resultados;
- A realização da avaliação com duas ou mais repetições dos testes em momentos cronologicamente separados por certo período de tempo, visando analisar o quanto à teoria proposta neste trabalho contribui para a aprendizagem significativa e memória de conceitos a longo prazo;
- A criação e avaliação de um número maior de problemas, com o intuito de testar uma maior quantidade de possibilidades de variação de problemas, aumentando, deste modo, a consistência das conclusões para a teoria de resolução de problemas;
- A realização de uma avaliação que focaliza a aprendizagem cooperativa, na qual se aplicam questões aos aprendizes em cooperação exija a comunicação entre eles, como por exemplo, exercícios subdivididos em partes distintas em que cada aluno responsabiliza-se por uma delas e ainda, construídos de modo que um participante precise da resolução da parte de outro para conseguir resolver a sua. Neste contexto ter-se-ia uma maior garantia da existência da troca de conhecimentos entre os estudantes;
- O desenvolvimento deste projeto focalizando-o em outra área do conhecimento para o estudo de caso, como Química, Biologia, Matemática, etc., visando analisar os efeitos das teorias propostas neste projeto para disciplinas diferentes da Física;
- A adição no projeto pedagógico deste trabalho da teoria de jogos e posterior avaliação dos efeitos cognitivos desta adição.

Referências Bibliográficas

Aaas. project 2061. Atlas of Science Literacy. Washington: American Association for the Advancement of Science 2001.

Ahuja, S. P. e R. Quintao. Performance Evaluation of Java RMI: A Distributed Object Architecture for Internet Based Applications. Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. Washington, DC, USA: IEEE Computer Society 2000. 565 - 569 p.

Arpaia, P., *et al.* A remote measurement laboratory for educational experiments. Measurement - Journal of the International Measurement Confederation v.21, n.4, p.157-169. 1997.

Baker, E. L. e R. E. Mayer. Computer-based assessment of problem solving. Computers in Human Behavior, v.15, p.269-282. 1999.

Barnatt, C. Cyber Business: Mindsets for a Wired Age. Spring: John Wiley & Sons. 1995

Barros, L. A. Suporte a Ambientes Distribuídos para Aprendizagem Cooperativa (Doutorado). COPPE, Universidade do Rio de Janeiro, Rio de Janeiro, 1994.

Barros, P. G. e J. Kelner. Simulação de tráfego – uma experiência com realidade virtual. VI Symposium on Virtual Reality - SVR03. Ribeirão Preto - SP: SBC, 2003. 140–151 p.

Bass, L., P. Clements e R. Kazman. Software Architecture in Practice. Boston: Addison-Wesley. 2003

Behrens, M. A. A prática pedagógica e o desafio do paradigma emergente. Revista Brasileira de Estudos Pedagógicos. Brasília. 80: 383-403 p. 2002.

Benford, S. e C. Greenhalgh. Collaborative Virtual Environments. 5th European conference on computer supported cooperative work. Lancaster - UK: ECSCW'97 1997.

Bono, E. D. Lateral Thinking: Creativity Step by Step. H. R. Publishers. New York 1973.

Bottentuit, J. B. e C. Coutinho. Laboratories based on Internet: comparative analyses of current experiences and development of a virtual laboratory. IV International Conference On Multimedia And Information And Communication Technologies In Education. Seville, 2006. 1284-1289 p.

Bowman, D. A., *et al.* 3D User Interfaces: Theory and Practice. Boston: MA: Addison-Wesley. 2005

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

- Bransford, J. D. e B. S. Stein. The ideal problem solver. New York: W. H. Freeman and Company. 1993
- Brutzman, D. e L. Daly. X3D: Extensible 3D Graphics for Web Author. Oxford: Morgan Kaufmann. 2007
- Bullinger, H. J., F. Müller-Spahn e A. Rößler. Encouraging Creativity - Support of Mental Processes by Virtual Experience. Virtual Reality World 1996, IDG conferences & seminars 1996.
- Burdea, G. C. e P. Coiffet. Virtual Reality Technology. New York: John Wiley and Sons, v.2. 2003
- Cardoso, A. Uma Arquitetura para Elaboração de Experimentos Virtuais Interativos suportados por Realidade Virtual Não imersiva. (Tese). Engenharia de Computação e Sistemas Digitais, Universidade São Paulo, São Paulo, 2000.
- Cardoso, A., J. G. Paiva e L. P. Soares. VRML e X3D. In: A. Cardoso, C. Kirner, *et al* (Ed.). Tecnologias para o desenvolvimento de sistemas de realidade virtual e aumentada. Recife: Universitária da UFPE, 2007. VRML e X3D, p.51-74
- Cebe.CEBE - The Centre for Education in the Built Environment. Disponível em: <http://www.cebe.heacademy.ac.uk/>. Acesso em: 02 abr. 2008.
- Construct3d.Construct3D. Disponível em: <http://www.ims.tuwien.ac.at/research/construct3d/>. Acesso em: 20 mar. 2009.
- Costa, S. S. C. D. e M. A. Moreira. Resolução de problemas I: diferença entre novatos e especialistas. Comissão Organizadora da IX Reunião de Educação em Física da Associação Argentina de Professores de Física. Salta - Argentina 1995.
- Dede, C., *et al*. Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts. Springer-Verlag, p.1-29. 1999.
- Deek, F. P., M. Turoff e J. A. Mchugh. A Common Model for Problem Solving and Program Development. Ieee Transactions on Education, v.4, n.42, p.331-336. 1999.
- Deitel. Java como programar: Pearson Education. 2005. 1152 p.
- Fakas, G. J., A. V. Nguyen e D. Gillet. The Electronic Laboratory Journal: A Collaborative and Cooperative Learning Environment for Web-Based Experimentation. Computer Supported Cooperative Work - Springer. 14: 189–216 p. 2005.
- Ficheman, I. K., *et al*. Gruta Digital: um Ambiente de Realidade Virtual Imersivo Itinerante para Aplicações Educacionais. XVII Simpósio Brasileiro de Informática na Educação Brasília: SBC, 2006. p.
- Freire, P. Saberes Necessários à Prática Educativa. São Paulo: Paz e Terra. 1997
- Gagné, R. M. The Conditions of Learning New York: Holt, Rinehart and Winston. 1965

Gerosa, M. A., H. Fuks e C. J. P. D. Lucena. The Development and Application of Distance Learning on the Internet. The Journal of Open and Distance Learning, v.17, n.1, p.23-38. 2002.

Gerosa, M. A., H. Fuks e C. J. P. D. Lucena. Suporte à Percepção em Ambientes de Aprendizagem Colaborativa. Revista Brasileira de Informática na Educação, v.11, n.2. 2003.

Harasim, L. On-Line Education: A New Domain. In: R. Mason e A. Kaye (Ed.). Mindweave: Communication, Computers and Distance instruction. Oxford: Pergamon Press, 1989. On-Line Education: A New Domain

Hayes, J. R. The complete problem solving. Philadelphia: Franklin Institute. 1980

Hofmann, M., C. Hock e F. Müller-Spahn. Computer-based cognitive training in Alzheimer's disease patients. Annals New York Academy of Science: J .H. Gordon, R. Nitsch, J. Wurtmann 1995.

Hopson, K. C. e S. E. Ingram. Desenvolvendo Applets com java. Rio de Janeiro: Editora Campus. 1997. 335 – 351 p.

Ieee.IEEE Standard for Learning Object Metadata. Disponível em: <http://www.ieeeltsc.org/standards/1484-12-1-2002>. Acesso em: 17 set. 2007.

Inep. Estatísticas do Professor no Brasil, v.1. 2004

Jones, P. M. e J. L. Jacobs. Cooperative Problem Solving in Human–Machine Systems: Theory, Models, and Intelligent Associate Systems. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews. 30: 397-407 p. 2000.

Kapa, E. A Metacognitive Support During the Process of Problem Solving in a Computerized Environment. Educational Studies in Mathematics, v.47, p.317–336. 2002.

Kaufmann, H. e D. Schmalstieg. Designing Immersive Virtual Reality for Geometry Education. IEEE Virtual Reality. Alexandria, Virginia, USA: IEEE, 2006. 52-58 p.

Kimbrough, D. R. Can we go the distance in chemistry? Reflections on distance learning laboratories. Strategies for Success. 2000.

Kirner, C. Realidade virtual: Dispositivos e aplicações. UFSCar 2004.

Kirner, C. e R. A. Siscoutto. Fundamentos de Realidade Virtual e Aumentada. In: C. Kirner e R. A. Siscoutto (Ed.). Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações: SBC, 2007. Fundamentos de Realidade Virtual e Aumentada

Kurose, J. F. e K. W. Ross. REDES DE COMPUTADORES E A INTERNET - Uma Abordagem Top-Down: Pearson Education. 2006

Lancaster_University.DEVRL - The Distributed Extensible Virtual Reality Laboratory. Disponível em: <http://www.comp.lancs.ac.uk/computing/research/cseg/projects/devrl/>. Acesso em: 02 abr. 2008.

Larocque, D. e N. Faucon. Me, myself and ... you? Collaborative learning: why bother? Teaching in the Community Colleges Online Conference - Trends and Issues in Online Instruction. Toronto, Ontario 1997.

Macaulay, L. A. Requirements Engineering. Great Britain: Springer-Verlag London. 1996. 202 p.

Macedonia, M. R. e M. J. Zyda. A Taxonomy for Networked Virtual Environments. Multimedia, IEEE, v.4, n.1, p.48-56. 1997.

Mantoan, M. T. Teachers' education for inclusive teaching: refinement of institutional actions. In: (Ed.). Revue Francophone de la Déficience Intellectuelle. Montréal/Québec, Canadá: Colloque Recherche Défi 1999, v.10, 1999. Teachers' education for inclusive teaching: refinement of institutional actions, p.52-54

Medeiros, A. e C. F. D. Medeiros. Possibilidades e Limitações das Simulações Computacionais no Ensino da Física. Revista Brasileira de Ensino de Física. 24: 77-86 p. 2002.

Mel, S., *et al.* Distributed extensible virtual reality laboratory (DEVRL): a project for co-operation in multi-participant environments. Proceedings of the Eurographics workshop on Virtual environments and scientific visualization '96. Monte Carlo, Monaco: Springer-Verlag 1996.

Mendes, E. Revisão Crítica do Currículo Integrado às Tecnologias Computacionais. In: Ufrj (Ed.). Ciências & Cognição Rio de Janeiro, Brasil, v.13, 2008. Revisão Crítica do Currículo Integrado às Tecnologias Computacionais, p.263-279

Mendes, E. B. Tendências futuras para Planejamento Curricular Construtivista. (Doutorado). Engenharia de Produção, Faculdade de Universidade Federal de Santa Catarina, Santa Catarina, 2002.

Miranda, R. M. D. GROA: Um Sistema de Gerência de Repositórios de Objetos de Aprendizagem. (Dissertação de Mestrado). Universidade Federal do Rio Grande do Sul Porto Alegre - Rio Grande do Sul, 2004.

Moodle-Community. A plataforma Moodle. Disponível em: www.moodle.org. Acesso em: 12 fev. 2009.

Morettin, P. A. e W. D. O. Bussab. Estatística Básica: Saraiva, v.1. 2003. 426 p.

Munoz, C. e J. Zalewski. Architecture and Performance of Java-Based Distributed Object Models: CORBA vs RMI. Real-Time Systems - SpringerLink, v.21, n.1-2, p.43-75. 2001.

Mzoughi, T., *et al.* WebTOP: Web-based interactive 3D optics and waves simulations. International Journal of Continuing Engineering Education and Life-Long Learning, v.15, n.2, p.79-94. 2005.

Nitzke, J. A., *et al.* Avaliando aplicações para criação de ambientes de aprendizagem colaborativa. SBIE - Simpósio Brasileiro de Informática na Educação. Curitiba, 1999. 303-310 p.

Novak, J. D. e R. W. Tyler. A Theory of Education: Cornell University Press. 1986

Nunnington, N., S. Lecturer e H. Eilander. The European Challenge. CEBE Transactions. 2: 44-63 p. 2005.

Oecd. Problem Solving for Tomorrow's World – First Measures of Cross-Curricular Competencies from PISA 2003. 2004

Olsen, B. R. E. W. e S. Kagan. About cooperative learning. In: C. Kessler (Ed.). Cooperative language learning: A teacher's resource book. Englewood Cliffs - NJ:: Prentice Hall, 1992. About cooperative learning, p.1-30

Oravec, J. A. Virtual Individuals, Virtual Groups - Dimensions of Groupware and Computer Networking. New York: Cambridge University Press, v.11. 1996 (Cambridge Series on Human-Computer Interaction Human)

Oxford, R. L. Cooperative Learning, Collaborative Learning, and Interaction: Three Communicative Strands in the Language Classroom The Modern Language Journal., v.81, n.4, p.443-456. 1997.

Perales, F. J. La resolución de problemas: una revisión estructurada. Enseñanza de las Ciencias : revista de investigación y experiencias didácticas. 11: 170-178 p. 1993.

Pérez, D. G., J. M. Torregrosa e F. P. Senent. El fracaso en la resolución de problemas de Física: una investigación orientada por nuevos supuestos. Enseñanza de las Ciencias : revista de investigación y experiencias didácticas. 6: 131-146 p. 1988.

Pozo, J. I., *et al.* A solução de Problemas: Aprender a resolver, resolver para aprender. (Tradução de Beatriz Affonso Neves). Porto Alegre: Artmed. 1998

Pressman, R. S. Engenharia de software. São Paulo: McGraw-Hill. 2006

Rogers, L. The Use of Software to Explore Experimental Data. GIREP-ICPE-ICTP International Conference: New Ways of Teaching Physics. Ljubjana, Slovenia 1996.

Roussos, M., *et al.* NICE: combining constructionism, narrative and collaboration in a virtual learning environment. ACM SIGGRAPH Computer Graphics, v.31, n.3, p.62 - 63. 1997.

Sherman, W. R. e A. B. Craig. Understanding Virtual Reality: Morgan Kaufmann. 2003

Silva, F. H. L. D. A memória: dos neurónios às redes neuronais. Revista de cultura científica, v.15, n.4. 1994.

Silveira, M. e L. Arruda. Desenvolvimento de um Sistema Multimídia para a Autoria e Apresentação de Cursos de Controle. 3rd Symposium Investigation and Development of Educational Software. Évora - Portugal, 1998. p.

Sommerville, I. e P. Sawyer. Requirements engineering - A good practice guide. England J. Wiley & Sons. 1997. 391 p.

Stahl, G., T. Koschmann e D. Suthers. Computer-supported collaborative learning: An historical perspective. In: R. K. Sawyer (Ed.). Cambridge handbook of the learning sciences Cambridge, UK: Cambridge University Press., 2006. Computer-supported collaborative learning: An historical perspective., p.409-426

Tarouco, L. M., *et al.* Aprendizagem e avaliação em um Ambiente de Realidade Virtual Cooperativo de Aprendizagem (Projeto ARCA). Revista Brasileira de Informática na Educação, v.6. 2000.

Terra.Governo investirá R\$ 203 milhões no Proinfo em 2007. Disponível em: <http://tecnologia.terra.com.br/interna/0,,OI1582233-EI4801,00.html>. Acesso em: 11 set. 2007.

Torres, P. L., P. R. Alcantara e E. A. F. Irala. Grupos de Consenso: uma proposta de aprendizagem colaborativa para o processo de ensino-aprendizagem. Revista Diálogo Educacional, v.4, n.13, p.129-145. 2004.

Trefftz, H., *et al.* Supporting the learning process in the classroom using 3d collaboration on mobile devices. IADIS International Journal on WWW/Internet, v.4, n.2, p.1-15. 2006.

Trinta, F. A. M. Arquiteturas Distribuídas para Co-autoria Cooperativa de Aulas na Internet. (Mestrado). Ciências da Computação, Universidade Federal de Pernambuco, Recife, 2000.

Triola, M. F. Introdução à estatística: LTC - Livros Técnicos e Científicos. 2005. 656 p.

Turoff, M., U. Rao e S. R. Hiltz. Collaborative hypertext in computer mediated communications. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on. Hawaii IEEE. 8-11 Jan, 1991. 357-366 p.

Ufrgs.ARCA. Disponível em: <http://penta.ufrgs.br/pgie/arca/arca.htm>. Acesso em:

Unesco. World Education Forum - Educação para todos: o Compromisso de Dakar 2001.

Unesco.Semana de Educação para Todos 2006 - O mundo precisa de professores mais bem qualificados. Disponível em: http://www.unesco.org.br/areas/educacao/efa2006/professoresqualificados/mostra_documento. Acesso em: 11 set. 2007.

Vanlehn, K., *et al.* The Andes Physics Tutoring System: Lessons Learned. International Journal of Artificial Intelligence in Education, v.15, n.3, p.1-47. 2005.

Verdejo, M. F., B. Barros e M. T. Abad. Supporting Distance Learners for Collaborative Problem Solving. ED-MEDIA & E-TELECOM, v.1, p.1407-1412. 1998.

Vince, J. Introduction to Virtual Reality: SpringerVerlag. 2004

Vrankar, L. Computer Games and Physics. GIREP-ICPE-ICTP International Conference: New Ways of Teaching Physics
Ljubjana, Slovenia 1996.

Waller, J. C. e N. Foster. Training via the web: a virtual instrument. Computers & Education, v.35, n.161-167. 2000.

Webtop.WebTop. Disponível em: <http://webtop.msstate.edu>. Acesso em: 20 mar. 2009.

Wiley, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In: D. A. Wiley (Ed.). The Instructional Use of Learning Objects, 2000. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy

X3d.X3D. Disponível em: <http://www.web3d.org/x3d/>. Acesso em: 08 abr. 2008.

Anexo 1

Teste Avaliativo do Projeto LVCF		Parte 1 – Movimentos uniformes e uniformemente variados	
Nome:		Série/Turma:	
Data:	/	/	/
Exercícios			
1	Uma pessoa lhe informa que um corpo está em movimento retilíneo uniforme. (a) O que está indicado pelo termo “retilíneo”? (b) E pelo termo “uniforme”?		
2	Quando um corpo está em movimento uniforme, com velocidade v , qual é a expressão matemática que nos permite calcular a distância d que ele percorre depois de decorrido um tempo t ?		
3	Usando a expressão solicitada no exercício anterior, calcule: (a) A distância percorrida por um carro, que se movimenta com velocidade constante $v = 54$ Km/h, durante um tempo $t = 0,50$ h. (b) A velocidade, suposta constante, de um nadador que percorre uma distância $d = 100$ m, em nado livre, em um tempo $t = 50$ s. (c) O tempo que a luz gasta para vir do Sol a Terra ($d = 1,5 \times 10^{11}$ m) sabendo-se que sua velocidade é constante e vale $v = 3,0 \times 10^8$ m/s.		
4	Um corpo, partindo do repouso, desloca-se em linha reta com aceleração constante. Neste caso que tipo de relação existe entre d e t ?		
5	Um corpo em movimento uniformemente variado, com velocidade inicial v_0 e aceleração a , percorre uma distancia d . Qual equação que nos permite calcular a velocidade no fim do percurso em função destes dados? (Observe que o tempo t não é um dado do problema.)		
6	Um corpo em movimento retilíneo uniforme acelerado possui, no instante $t = 0$, uma velocidade inicial $v_0 = 5,0$ m/s e sua aceleração é $a = 1,5$ m/s ² . Calcule a velocidade do corpo no instante $t = 8,0$ s.		
7	Um carro está se movendo com uma velocidade de 12 m/s. Em certo instante ($t = 0$) o motorista pisa no freio, fazendo com que o carro adquira um movimento uniformemente retardado, com uma aceleração cujo valor numérico é 10 m/s ² . Calcule a velocidade deste automóvel após percorrer uma distância de 40 m a partir do início da freiada.		

Anexo 2

Teste Avaliativo do Projeto LVCF		Parte 2 – Leis de Newton	
Nome:		Série/Turma:	Data: / /
Exercícios			
1	Responda: <p>(a) Se um corpo está se movendo, que tipo de movimento ele tende a ter, em virtude da inércia?</p> <p>(b) A força é uma grandeza escalar ou vetorial? Justifique a sua resposta.</p> <p>(c) O que é o peso de um corpo? Qual são a direção e o sentido do vetor que representa o peso de um corpo?</p> <p>(d) Uma partícula em repouso está em equilíbrio? Uma partícula em equilíbrio pode estar em movimento? Justifique a sua resposta.</p>		
2	Um carro tromba com um grande caminhão carregado. Nesta interação, a força que o carro exerce no caminhão é maior, menor ou igual à força que o caminhão exerce no carro? Normalmente, neste tipo de batida, o carro estraga mais que o caminhão, por quê?		
3	Um bloco está se movendo em linha reta com uma velocidade v constante sobre uma superfície horizontal lisa. Em certo instante uma força F constante é aplicada no bloco. Que tipo de movimento que o bloco passa a descrever supondo que: <p>(d) F tem mesma direção e o mesmo sentido de v.</p> <p>(e) F tem a mesma direção e sentido contrario a v.</p>		
4	A resultante das forças que atuam em um corpo, cuja massa é $m = 4,0$ kg, vale $R = 20$ N. Qual é o valor da aceleração que este corpo possui?		
5	Um bloco, sob a ação de uma força resultante $R = 2,0$ kgf, adquire uma aceleração $a = 400$ cm/s ² . <p>(a) Para se calcular, em kg, a massa do bloco, em que unidades devem estar expressos os valores de R e a?</p> <p>(b) Calcule a massa do bloco em kg.</p>		

Anexo 3

Manual explicativo LVCF	O trilho de ar
------------------------------------	----------------

O trilho de ar é um sistema que permite estudar movimentos unidimensionais reduzindo drasticamente as forças de atrito habitualmente presentes. Ele é composto de chapas metálicas de perfil reto, com pequenos orifícios regularmente espaçados nas faces laterais. Injeta-se ar comprimido dentro do trilho. Este ar sai através dos orifícios, gerando um colchão de ar entre o trilho e o carrinho (objeto colocado sobre o trilho para fazer as experiências) de cerca de meio milímetro de espessura. Este colchão de ar faz com que o carrinho “flutue”, provocando uma grande redução do atrito. O único atrito existente no experimento, denominado residual, é devido basicamente à fricção com o meio gasoso.

O trilho de ar permite a execução de uma grande variedade de experimentos, como, por exemplo, experiências envolvendo as leis de Newton. Normalmente estes experimentos ocorrem por meio da adição de pesos. Podem-se adicionar pesos ao carrinho e a um suporte que é ligado a ele através de um fio.

No programa de computador apresentado tem-se o trilho de ar virtual, o qual é ilustrado na figura abaixo.

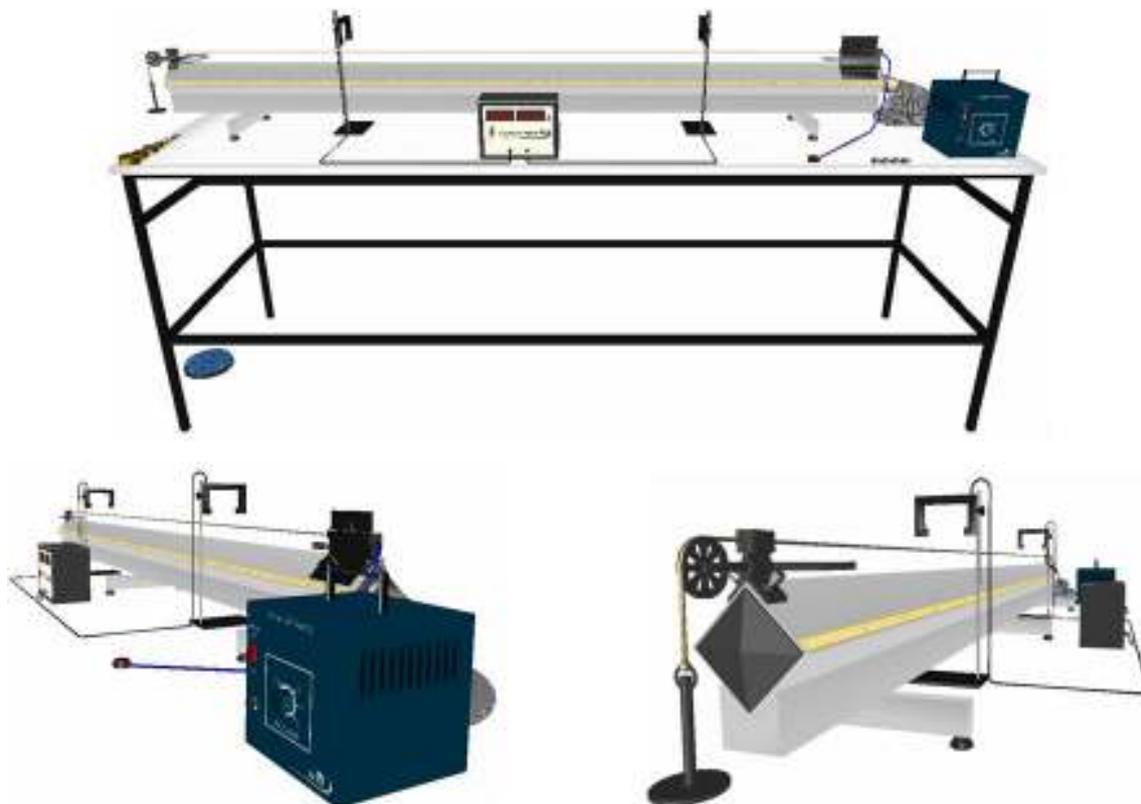


Figura 1: Catálogo de ilustrações do trilho de ar virtual.

O trilho de ar necessita de um conjunto de componentes e acessórios para o seu funcionamento correto em alguns experimentos como:

Componente	Ilustração	Descrição
gerador de fluxo de ar		é ligado a uma das extremidades do trilho por uma mangueira, e possui a função de criar o fluxo de ar dentro do tubo e permitir que o ar, ao sair pelos orifícios do trilho, crie o "colchão de ar", reduzindo o atrito;
carrinho		é o elemento a ser lançado sobre o trilho de ar para análise de seus respectivos movimentos. Basicamente, o carrinho consiste da união de três placas plásticas por um lado comum, de modo que sua base pudesse ser perfeitamente encaixada na base do trilho;
disparador magnético		consiste de uma peça metálica que, por meio de um fluxo magnético, pode segurar o carrinho enquanto são adicionados os pesos. Ao término da adição de

		pesos, o fluxo magnético pode ser interrompido, por intermédio de um botão ligado a peça metálica através de um fio, esta interrupção provoca a liberação do carrinho;
amortecedor de choques e roldana		o amortecedor consiste de uma junta acoplada a um elástico que possui a função de reduzir o impacto do carrinho, a roldana possui a função de acomodar o fio que liga o carrinho ao suporte de pesos;
sensor fotoelétrico		possui a função de emitir um sinal para o cronômetro quando o seu feixe de fotoelétrons é interrompido pelo carrinho;
cronômetro digital eletrônico		ele possui a função de marcar o tempo entre a interrupção do primeiro sensor fotoelétrico e segundo.

Existe ainda uma fita métrica acoplada ao longo do trilho, que realiza o papel de sistema de referência do movimento, de maneira que a posição do carrinho em qualquer instante pode ser marcada pela fita.

Anexo 4

**Teste Avaliativo do Projeto
LVCF**

Primeiro problema

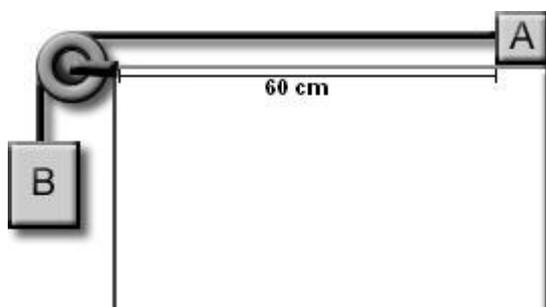
Nome: _____

Série/Turma: _____

Data: ____ / ____ / ____

PROBLEMA

Um bloco A de 40 g está conectado a um bloco B de 50 g por meio de um fio, como ilustra a figura abaixo. O sistema encontra-se inicialmente em repouso, uma vez em movimento, determine o tempo que o bloco A gasta para percorrer 60 cm. Considere o atrito da superfície que o bloco A se apóia desprezível e o valor da gravidade igual a $9,8 \text{ m/s}^2$.



Quais os dados fornecidos pelo problema? O que o problema espera como resposta?

Descreva as definições, conceitos, unidades de medida e relação entre os dados retirados.

Quais dados ou valores precisam ser encontrados para chegar à solução do problema?

Quais teorias e equações você usará para encontrar os dados anteriores?

Descreva um plano de solução para o problema por meio de uma seqüência de passos.

O que você prevê encontrar em cada passo acima?

Resolva o problema.

Existe de outras formas melhores de resolver o problema? Quais?

Você conseguiu chegar ao resultado do problema? Se sim, quais ações você tomou que poderiam ser consideradas erradas e que deveriam ser evitadas na resolução de um próximo problema. Se não, quais foram às dificuldades que o impediram de chegar à solução?

Escreva um resumo da solução em tópicos.

Descreva como você realizou o experimento no ambiente computacional. (Item aplicado somente para as amostra β e γ .)