

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



VIRTUAL SUBSTATION
**Um sistema de Realidade Virtual para treinamento de
operadores de subestações elétricas**

Reginaldo Costa Silva

Orientando

Alexandre Cardoso Dr.

Orientador

Jose Rubens Macedo Jr Dr.

Coorientador

Uberlândia MG, 2012

REGINALDO COSTA SILVA

VIRTUAL SUBSTATION

Um sistema de Realidade Virtual para treinamento de operadores de subestações elétricas

Dissertação apresentada por Reginaldo Costa Silva à Universidade Federal de Uberlândia para obtenção do título de Mestre em Ciências, avaliada em 21 de setembro de 2012 pela banca examinadora.

Área de Concentração

Computação gráfica

Banca Examinadora

Alexandre Cardoso - UFU (orientador)

José Rubens Macedo, - UFU (Coorientador)

Alberto Raposo - PUC-Rio (convidado)

Uberlândia MG, 2012

VIRTUAL SUBSTATION

Um sistema de Realidade Virtual para treinamento de operadores de subestações elétricas

Dissertação apresentada por Reginaldo Costa Silva à Universidade Federal de Uberlândia para obtenção do título de Mestre em Ciências, avaliada em 21 de setembro de 2012 pela banca examinadora.

Área de Concentração

Computação gráfica

Alexandre Cardoso

Orientador

Jose Rubens Macedo Jr

Coorientador

Uberlândia MG, 2012

*Dedico este trabalho à minha mãe, pelo constante
amor e apoio que sempre me deu em todas as
minhas conquistas.*

Agradecimentos

Aos orientadores, pelo constante apoio e pelo suporte oferecido, sem a qual a realização deste trabalho seria impossível.

Ao professor Edgard Lamounier, meu terceiro orientador, por todo o apoio e consideração que dispensou ao meu trabalho.

Ao meu grande amigo Alexandre Ribeiro pelo apoio e por me fornecer um ombro para chorar nos momentos em que eu pensava em desistir.

Ao meu colega Abadio, pelos preciosos conselhos.

Ao Kenedy e Gerson Flávio cujas contribuições foram fundamentais para a construção do protótipo.

Ao professor Márcio Arruda e os operadores da Cemig da CEMIG por ceder seu tempo para responder meus questionários. Bem como aos alunos do laboratório de Qualidade da Energia Elétrica UFU pela calorosa acolhida ao projeto e as contribuições valiosas que incluíram em suas respostas.

A todos os professores do MINTER, pois cada um contribuiu á sua maneira para meu crescimento profissional e enquanto pessoa.

Aos dirigentes do IFTM e da UFU por viabilizarem a parceria que criou este programa de Mestrado Interinstitucional. Que ele possa ter prosseguimento que traga grandes frutos.

Ao professor Willian Alexandre Manzan, um dos principais responsáveis pelo sucesso deste programa de Mestrado Interinstitucional.

E finalmente, ao Projeto de Pesquisa e Desenvolvimento P&D - “GT 411 - PA - Desenvolvimento de ambientes virtuais, para centro de operação de sistemas representativos das subestações e usinas da Cemig, associados a tecnologias de projeção 3D” promovido pela parceria Cemig-UFU –Aneel.

“Grandes obras não são feitas com força, mas a perseverança.”

Samuel Johnson

Resumo

Para que a energia elétrica seja gerada e distribuída de forma segura, nosso país conta com uma enorme e complexa rede de distribuição. Dentro de tal rede, um componente importante são as subestações elétricas, responsáveis pelo controle, supervisão, medição e proteção dos fluxos desta energia. Devido aos enormes avanços tecnológicos dos últimos anos e a grande diversidade de projetos de subestação existentes, a operação delas só pode ser realizada por profissionais altamente qualificados. Tamanha é a exigência de qualificação por parte destes profissionais, que seu treinamento se torna demorado, complicada e caro. Neste contexto, a tecnologia de Realidade Virtual se mostra uma alternativa interessante, uma vez que permite a criação de objetos de aprendizagem altamente realísticos, com o qual o aspirante a operador pode interagir, sem colocar sua vida em risco e sem causas prejuízos financeiros. A proposta deste trabalho é investigar técnicas de realidade virtual para a construção de ambientes de treinamento adequados a estes profissionais. Para realizar a validação destas técnicas, foi construído um Sistema de Realidade Virtual denominado Virtual Substation no qual o estudante pode interagir com um transformador de força por meio de manipulação de objetos 3D ou do painel do sistema supervisão tradicional. Ambas as visualizações (2D e 3D) são mostradas de forma simultânea, para que o usuário do sistema possa verificar de que forma uma interfere na outra. Os passos realizados pelo trainee durante a sessão de treinamento são armazenados em um Sistema Gerenciador de Banco de Dados e podem ser recuperados por seu instrutor, através de um ambiente 3D próprio. Após a construção do protótipo, o mesmo foi apresentado a estudantes do curso de Engenharia Elétrica e profissionais da área que validaram sua contribuição como ferramenta pedagógica.

Palavras-chave: Realidade Virtual, subestações elétricas, treinamento

Abstract

For the electricity be generated and distributed safely, our country has a huge and complex distribution network. Within such a network, an important component of the electrical substations. They are responsible for the control, supervision, protection and metering the flow of this energy. Due to the enormous technological advances of recent years and the great diversity of existing substation projects, the operation of which can only be performed by highly skilled professionals. Such is the demand for qualified by these professionals, their training becomes time consuming, complicated and expensive. Thus, the Virtual Reality technology proves an interesting alternative, since it enables the creation of highly realistic learning objects, with which the aspiring trader can interact without putting your life at risk and without causes financial losses. The purpose of this study is to investigate virtual reality techniques for building training environments suitable for these professionals. To perform validation of these techniques, we built a system called Virtual Reality Virtual Substation in which students can interact with a transformer power through manipulation of 3D objects or panel traditional supervisory system. Both of these views (2D and 3D) are shown simultaneously, so that the user of the system can verify how one interferes with the other. The steps performed by the trainee during the training session are stored in a System Manager database and can be retrieved by their instructor, via a 3D environment itself. After the construction of the prototype, it was presented to students of Electrical Engineering and professionals who validated his contribution as a pedagogical tool.

Keywords: Virtual Reality, electrical substations, training

Lista de ilustrações

| | |
|--|----|
| Figura 1 - Os três pilares da RV..... | 27 |
| Figura 2 - Subestação elevadora de Juiz de Fora (MG) extraído de http://www.pracomprar.com/v2/abrir/site/blp-engenharia/servicos/1420 | 32 |
| Figura 3 - Subestação elétrica da usina de Igarapava – SP | 32 |
| Figura 4 - Transformador de Força..... | 33 |
| Figura 5 - interface gráfica com usuário (GUI) do sistema ArqElectrical..... | 36 |
| Figura 6 - Volante de navegação do sistema ArqElectrical..... | 37 |
| Figura 7 - Plataforma proposta pelo projeto RV-COPEL | 39 |
| Figura 8 - Seleção dos objetos na cena com o wiimote no sistema RV-COPEL..... | 40 |
| Figura 9 Vista do usuário próximo ao poste com os condutores cobertos no sistema RV-COPEL..... | 40 |
| Figura 10 - Tela do MineInside ®, mostrando a operação de mineração os diversos veículos no detalhe..... | 42 |
| Figura 11 Interface para programação do Simulador de Treinamento FURNAS..... | 44 |
| Figura 12 - Sala de treinamento do simulador FURNAS..... | 45 |
| Figura 13 - Interface gráfica para programação de cenários..... | 46 |
| Figura 14 - Controle wii – ou “wiimote” e os graus de liberdade proporcionados por ele Fonte: http://futurelab.com.br/site/laboratorio-inovacao/tour-3D/ | 47 |
| Figura 15 -Treinando controlando o cenário virtual por meio do "wiimote" | 48 |
| Figura 16 - Caso de uso do Virtual Substation..... | 52 |
| Figura 17 - Modelo MVC do gerenciador de login | 60 |
| Figura 18 - Interação das camadas MVC na área de treinamento | 61 |
| Figura 19 - esboço da tela do Virtual Substation..... | 62 |
| Figura 20 - Interação das camadas MVC do módulo de recuperação. | 63 |
| Figura 21 - Sistema desenvolvido em VRML –..... | 66 |
| Figura 22 - Ambiente virtual simulando o fundo do mar criado em Papervision..... | 67 |
| Figura 23 - Exemplo de ambiente virtual desenvolvido em Unity 3D para o celular Iphone. Fonte: http://unity3D.com/support/resources/example-projects/ | 68 |
| Figura 24 - Jogo desenvolvido com JavaFX. | 69 |
| Figura 25 - Pasta do Virtual Substation no gerenciador de arquivos..... | 75 |
| Figura 26 - Modelagem geométrica utilizando-se 3d Max..... | 77 |
| Figura 27 - Modelagem importada dentro do ambiente Unity 3d..... | 77 |
| Figura 28 - objeto First Person Controller do Unity 3d | 78 |
| Figura 29 - script criado para alternar entre as câmeras..... | 79 |
| Figura 30 - script criado para o desligamento da rotação de câmera..... | 80 |

| | |
|--|-----|
| Figura 31 - curva de carga no formato de planilha..... | 81 |
| Figura 32 - tabela vc_curva_carga criada com base na planilha de curva de carga..... | 81 |
| Figura 33 - Virtual Substation em execução no ambiente Windows..... | 82 |
| Figura 34 - virtual Substation em execução no navegador | 82 |
| Figura 35 - tela de Login | 83 |
| Figura 36 - Código Javascript responsável por fazer Login de usuários..... | 85 |
| Figura 37 - área do Operador Trainee..... | 86 |
| Figura 38 - alarme mostrando que os limites do comutador de tapes foram infringidos | 88 |
| Figura 39 - Menu de Trainees na área do instrutor..... | 88 |
| Figura 40 - Lista de sessões realizadas pelo Trainee..... | 89 |
| Figura 41 - Área de recuperação da sessão..... | 89 |
| Figura 42 - detalhe do log de eventos mostrando ação realizada pelo trainee..... | 90 |
| Figura 43 - Código fonte de GerenciadorDeLogin.php responsável pela autenticação dos usuários no sistema..... | 92 |
| Figura 44 - código responsável por obter as informações do usuário junto ao SGDB..... | 95 |
| Figura 45 - Diagrama físico do SGDB..... | 96 |
| Figura 46 - Respostas dos entrevistados à questão "1.1. O software auxilia no processo de aprendizagem na finalidade proposta?"..... | 100 |
| Figura 47 - Resposta dos entrevistados à pergunta "1.2. Como avalia globalmente a funcionalidade do software" | 101 |
| Figura 48 - Respostas dos entrevistados à pergunta "2.1. Em relação à facilidade de entender com funciona o programa" | 102 |
| Figura 49 - Respostas dos entrevistados à questão "2.2. Quanto à navegação na cena e manipulação dos objetos (facilidade de uso)" | 103 |
| Figura 50 - Respostas dos entrevistados à questão "3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados."..... | 104 |
| Figura 51 - Resposta dos entrevistados à questão "3.2. Capacidade de o software enviar mensagens de erro caso os mesmos ocorram "..... | 104 |
| Figura 52 - Respostas dos entrevistados à questão "4.1. Tempo de resposta às ações do usuário" | 105 |
| Figura 53 - Respostas dos entrevistados à questão "4.2. Fidelidade a uma subestação e um transformador de força reais" | 106 |
| Figura 54 - Respostas dos entrevistados à questão "4.3. Como avalia em termos globais a eficiência do software" | 106 |
| Figura 55 - Respostas dos entrevistados à questão "5.1. Facilidade de execução em diferentes ambientes (Executável Windows/Página web)" | 107 |

| | |
|---|-----|
| Figura 56 - Respostas dos entrevistados à questão "5.2. Como avalia em termos globais a Portabilidade do software" | 108 |
| Figura 57 - Respostas dos entrevistados à questão "6.1. A importância que atribui ao software para o aprendizado em geral." | 108 |
| Figura 58 - Resposta dos entrevistados à questão "6.2. A importância de treinar em Ambientes de Realidade Virtual" | 109 |
| Figura 59 - Resposta dos entrevistados à questão "6.3. A importância de treinar em um simulador de transformador de força" | 110 |
| Figura 60 - Respostas dos entrevistados à questão "6.4. A importância geral do sistema no treinamento de um operador do sistema " | 110 |
| Figura 61 - Comparação das respostas dadas por estudantes e operadores á pergunta "1.1. O software auxilia no processo de aprendizagem na finalidade proposta?" | 111 |
| Figura 62 - Comparação das respostas dadas por estudantes e operadores á pergunta "1.2. Como avalia globalmente a funcionalidade do software" | 112 |
| Figura 63 - Comparação das respostas dadas por estudantes e operadores á pergunta "2.1. Em relação à facilidade de entender com funciona o programa" | 113 |
| Figura 64 - Comparação das respostas dadas por estudantes e operadores á pergunta "2.2. Quanto à navegação na cena e manipulação dos objetos (facilidade de uso)" | 114 |
| Figura 65 - Comparação das respostas dadas por estudantes e operadores á pergunta "3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados." | 115 |
| Figura 66 - Comparação das respostas dadas por estudantes e operadores á pergunta "3.2. Capacidade de o software enviar mensagens de erro caso os mesmos ocorram " | 115 |
| Figura 67 - Comparação das respostas dadas por estudantes e operadores á pergunta "4.1. Tempo de resposta às ações do usuário" | 116 |
| Figura 68 - Comparação das respostas dadas por estudantes e operadores á pergunta "4.2. Fidelidade a uma subestação e um transformador de força reais" | 117 |
| Figura 69 - 4.4. Como avalia em termos globais a Portabilidade do software | 117 |
| Figura 70 - Comparação das respostas dadas por estudantes e operadores á pergunta "5.1. Facilidade de execução em diferentes ambientes (Executável Windows/Página web)" | 118 |
| Figura 71 - Comparação das respostas dadas por estudantes e operadores á pergunta "5.2. Como avalia em termos globais a Portabilidade do software" | 118 |
| Figura 72 - Comparação das respostas dadas por estudantes e operadores á pergunta "6.1. A importância que atribui ao software para o aprendizado em geral." | 119 |
| Figura 73 - Comparação das respostas dadas por estudantes e operadores á pergunta "6.2. A importância de treinar em Ambientes de Realidade Virtual" | 120 |

| | |
|--|-----|
| Figura 74 - Comparação das respostas dadas por estudantes e operadores á pergunta “6.3. A importância de treinar em um simulador de transformador de força” | 120 |
| Figura 75 - Comparação das respostas dadas por estudantes e operadores á pergunta “6.4. A importância geral do sistema no treinamento de um operador do sistema” | 121 |

Lista de tabelas

| | |
|--|-----|
| Tabela 1- estudo comparativo dos trabalhos relacionados | 49 |
| Tabela 2 - Comparativo entre as tecnologias analisadas..... | 72 |
| Tabela 3 - Diferenças gerais entre as versões do Unity 3D. | 73 |
| Tabela 4 - Diferenças entre as versões do Unity no que diz respeito ao desenvolvimento. | 73 |
| Tabela 5 - Conteúdo da pasta visão..... | 76 |
| Tabela 6 - conteúdo da pasta controlador..... | 91 |
| Tabela 7 - Conteúdo da pasta modelo | 93 |
| Tabela 8 - Comparação do Virtual Substation com demais sistemas. | 123 |

Glossário

| | |
|-------------------|---|
| Android | Sistema operacional da Google usado em smartphones e tablets |
| Aneel | Agência Nacional de Energia Elétrica |
| Apache | Servidor web onde as páginas ficam armazenadas e processadas para serem entregues ao navegador de internet |
| APE | Actionscript Physics Engine – Biblioteca que permite incorporar física a animações criadas em Flash |
| API | Application Programmers Interface – interface para programadores – conjunto de bibliotecas e documentações presentes em uma linguagem ou sistema que podem ser usadas por programadores para construir novos conteúdos para o/a mesmo/a |
| AV | Ambiente Virtual |
| CAD | Computer-Aided Design – Desenho auxiliado por computador |
| CAE | Computer-Aided Engineering – Engenharia auxiliada por computador |
| DOF | Dregres of Freedon – Graus de liberdade |
| Standalone | Programa completamente autossuficiente que para seu funcionamento não necessitam de um software auxiliar, como um interpretador, sob o qual teria que ser executado. |
| Framework | Conjunto de bibliotecas para programação |
| FTP | File Transfer Protocol – Protocolo usado para transferência de arquivos na internet |
| GPS | Global Positioning System – sistema de posicionamento global usado para rastreamento de dispositivos |
| GUI | Grphic User Interface – Interface gráfica com o usuário |
| Hibernate | Framework ORM usado para diminuir a quantidade de SQL escrito em softwares |
| HTML | Hypertext Markup Language - Linguagem de Marcação de Hipertexto. É uma linguagem de marcação utilizada para produzir páginas de internet |
| HTTP | Protocolo usado para transporte das páginas de internet do servidor até a máquina do internauta |
| IOs | Sistema Operacional da Apple usado em dispositivos como Iphone e Ipad |
| JAVA FX | Plataforma criada pela empresa Oracle que permite a criação de interfaces ricas de internet usando 3D com a linguagem Java |
| Javascript | Linguagem de programação em scripts bastante utilizada para conferir |

| | |
|----------------------|---|
| | interatividade a páginas HTML. Nenhuma relação com Java, a não ser o nome semelhante. |
| JDBC | Java Database Connectivity – Mecanismo que permite à linguagem Java se comunicar com SGDBs diversos |
| JE | Jogos Eletrônicos |
| Kernel | Componente central do sistema operacional da maioria dos computadores |
| LEDs | Light Emitting Diode - diodos de emissão de luz – usados para criação de painéis luminosos, monitores e TVs. |
| MVC | Modelo Visão Controlador – padrão de projetos utilizado em softwares que divide a implementação em três camadas distintas, de acordo com a função de cada uma. |
| MySQL | Banco de dados bastante utilizado em conjunto com PHP e Apache |
| Open source | Código aberto – refere-se a programas que podem ser modificados e redistribuídos livremente |
| ORM | Mapeamento Objeto Relacional – técnica que consiste em associar uma ou mais tabelas de um SGDB relacional a uma classe de programa |
| PAPERVISION | Biblioteca que permite a criação de animação 3D usando a plataforma Flash da Adobe |
| PHP | Personal Hypertext Preprocessor – linguagem de scripts utilizada para criar páginas web do lado servidor. Estas páginas web normalmente recebem requisições para gravação e recuperação de informações. |
| Playstation 3 | Videogame da Sony |
| PostgreSQL | SGDB gratuito bastante utilizado na indústria |
| RA | Realidade Aumentada |
| RV | Realidade Virtual |
| SCADA | Supervisory Control And Data Acquisition - Sistemas de Supervisão e Aquisição de Dados. São sistemas que utilizam software para monitorar e supervisionar as variáveis e os dispositivos de sistemas de controle conectados às SE's |
| SE | Subestações Elétricas |
| SGBD | Sistema Gerenciador de Banco de Dados |
| SQL | Structured Query Language – linguagem utilizada para inserção, deleção e recuperação de informações em um SGDB. |
| Stakeholder | Um <i>stakeholder</i> em uma arquitetura de software é uma pessoa, grupo ou entidade com um interesse ou preocupações sobre a realização da |

| | |
|-------------------------------|--|
| | arquitetura. |
| UNITY 3D | Ferramenta criada pela empresa Unity para a criação de jogos |
| VRML | Virtual Reality Modeling Language – linguagem para modelagem de Realidade Virtual |
| Wii mote ou Wii remote | Dispositivo usado para jogar no videogame Wii da Nintendo |
| WIMP | Window, Icon, Menu, Pointing Device – estilo de interação usado nos sistemas gráficos como Windows, Linux e Leopard. |
| X3D | Padrão aberto para distribuir conteúdo 3D |
| Xbox | Videogame da Microsoft |

Lista de Publicações Relacionadas a este Trabalho

1. SILVA, R. C.; CARDOSO, A.; LAMOUNIER, E. **VIRTUAL SUBSTATION: Um sistema de realidade virtual para treinamento de operadores de subestações elétricas.** Workshop de Realidade Virtual e Aumentada: Anais, Uberaba, MG, n. , p.1-5, 07 nov. 2011.
2. SILVA, R. C.; SILVA Jr., A. R. **TECNOLOGIAS PARA CONSTRUÇÃO DE MUNDOS VIRTUAIS:: UM COMPARATIVO ENTRE AS OPÇÕES EXISTENTES NO MERCADO. Fazu em Revista,** Uberaba, MG, n. 8, p.211-215, 01 jul. 2011. Anual.

Sumário

| | |
|---|----|
| Capítulo 1 - Introdução..... | 21 |
| 1.1 Motivação..... | 23 |
| 1.2 Objetivo central..... | 24 |
| 1.3 Objetivos específicos..... | 24 |
| 1.4 Organização da dissertação..... | 25 |
| Capítulo 2 - Fundamentos | 26 |
| 2.1 Introdução | 26 |
| 2.2 Realidade Virtual | 26 |
| 2.2.1 Os três pilares da Realidade Virtual..... | 27 |
| 2.2.2 Sistemas imersivos e não imersivos | 28 |
| 2.2.3 Vantagens e limitações da Realidade Virtual | 29 |
| 2.2.4 Subestações elétricas | 31 |
| 2.2.5 Transformador de força..... | 33 |
| 2.3 Conclusão..... | 33 |
| Capítulo 3 - Trabalhos relacionados | 35 |
| 3.1 Introdução..... | 35 |
| 3.2 Aplicações correlatas..... | 35 |
| 3.2.1 arqElectrical..... | 35 |
| 3.2.2 RV-COPEL..... | 38 |
| 3.2.3 MineInside | 40 |
| 3.2.4 Simulador Furnas 1..... | 43 |
| 3.2.5 Simulador Furnas 2..... | 45 |
| 3.3 Comparação entre os sistemas apresentados..... | 48 |
| 3.4 Conclusão..... | 49 |
| Capítulo 4 - Especificação do sistema proposto..... | 51 |
| 4.1 Introdução | 51 |

| | | |
|---|--|-----|
| 4.2 | Levantamento de requisitos | 51 |
| 4.3 | Arquitetura do sistema | 58 |
| 4.3.1 | Modelo MVC | 58 |
| 4.3.2 | Módulos do sistema | 58 |
| 4.3.3 | Gerenciador de login | 59 |
| 4.4 | Conclusão..... | 64 |
| Capítulo 5 - Detalhes da implementação..... | | 65 |
| 5.1 | Introdução | 65 |
| 5.2 | Tecnologia para construção da camada de visão | 66 |
| 5.2.1 | VRML..... | 66 |
| 5.2.2 | Papervision..... | 67 |
| 5.2.3 | Unity 3D | 67 |
| 5.2.4 | Java FX | 69 |
| 5.2.5 | Critérios para avaliação das ferramentas | 69 |
| 5.2.6 | Ferramenta escolhida para a construção do ambiente virtual | 72 |
| 5.3 | Tecnologias de apoio..... | 74 |
| 5.3.1 | Linguagem PHP..... | 74 |
| 5.3.2 | PostgreSQL | 74 |
| 5.4 | O modelo MVC implementado..... | 75 |
| 5.4.1 | Camada visão | 75 |
| 5.4.2 | Camada controlador | 91 |
| 5.4.3 | Camada modelo | 93 |
| 5.5 | Conclusão..... | 97 |
| Capítulo 6 - Resultados e limitações | | 98 |
| 6.1 | Introdução | 98 |
| 6.2 | Metodologia | 98 |
| 6.3 | Resultados | 100 |
| 6.3.1 | Funcionalidade..... | 100 |

| | | |
|---|--|-----|
| 6.3.2 | Usabilidade..... | 101 |
| 6.3.3 | Confiabilidade..... | 103 |
| 6.3.4 | Eficiência..... | 105 |
| 6.3.5 | Portabilidade..... | 107 |
| 6.3.6 | Sobre o sistema..... | 108 |
| 6.4 | Comparações entre as respostas de estudantes e operadores..... | 111 |
| 6.4.1 | Funcionalidade..... | 111 |
| 6.4.2 | Usabilidade..... | 112 |
| 6.4.3 | Confiabilidade..... | 114 |
| 6.4.4 | Eficiência..... | 116 |
| 6.4.5 | Portabilidade..... | 117 |
| 6.4.6 | Sobre o sistema..... | 119 |
| 6.4.7 | Resultado da comparação..... | 121 |
| 6.5 | Algumas sugestões dos usuários..... | 122 |
| 6.6 | Conclusão..... | 122 |
| Capítulo 7- Conclusões..... | | 124 |
| Capítulo 8 - Trabalhos futuros..... | | 126 |
| Referências..... | | 128 |
| Apêndices..... | | 137 |
| Apêndice 1 – Questionário aplicado..... | | 137 |



Capítulo 1 - Introdução

Segundo Aneel (2002) a energia, nas suas mais diversas formas, é indispensável à sobrevivência da espécie humana. E mais do que sobreviver, o homem procurou sempre evoluir, descobrindo fontes e formas alternativas de adaptação ao ambiente em que vive e de atendimento às suas necessidades. Dessa forma, a exaustão, escassez ou inconveniência de um dado recurso tendem a ser compensadas pelo surgimento de outro(s). Em termos de suprimento energético, a eletricidade se tornou uma das formas mais versáteis e convenientes de energia, passando a ser recurso indispensável e estratégico para o desenvolvimento socioeconômico de muitos países e regiões.

Segundo Eletrobrás (2012) a eletricidade se tornou a principal fonte de luz, calor e força utilizada no mundo moderno. Atividades simples como assistir à televisão ou navegar na internet são possíveis porque a energia elétrica chega até a sua casa. Fábricas, supermercados, shoppings e uma infinidade de outros lugares precisam dela para funcionar. Grande parte dos avanços tecnológicos que alcançamos se deve à energia elétrica.

No Brasil, a energia elétrica gerada nas usinas, sejam elas hidrelétricas ou térmicas, passa por diversos sistemas até sua chegada ao consumidor final em baixa tensão. De uma forma geral, a energia elétrica é disponibilizada ao sistema interligado nacional em tensões não superiores a 13,8 kV, sendo na sequência elevadas para níveis de alta tensão (até 750 kV no caso brasileiro) de forma a possibilitar sua transmissão por longas distâncias com perdas reduzidas. Após a fase de transmissão, o nível da tensão de fornecimento é novamente reduzido para valores não superiores a 138 kV, iniciando a etapa de distribuição de energia elétrica, passando por uma nova redução para valores não superiores a 34,5 kV. Finalmente, o nível da tensão de fornecimento é novamente reduzido para valores não superiores a 380 V, sendo esta a tensão final de atendimento às diversas unidades de consumo de baixa tensão, a exemplo de residências e pequenos comércios e indústrias.

A grande extensão da rede de transmissão no Brasil é explicada pela configuração do segmento de geração, constituído, na maior parte, de usinas hidrelétricas instaladas em localidades distantes dos centros consumidores. A principal característica desse segmento é a sua divisão em dois grandes blocos: o Sistema Interligado Nacional (SIN), que abrange a quase totalidade do território brasileiro, e os Sistemas Isolados, instalados, principalmente, na região Norte. Aneel (2008)

Segundo Rossi (2004) as interações eletromecânicas e as interligações físicas entre as áreas de produção, transmissão e de distribuição de energia elétrica, são realizadas através de **centros**, onde se processam as transformações de potências, os chaveamentos e manobras de

circuitos desse sistema de transmissão, envolvendo as funções de controle e supervisão, (medição e proteção) nos fluxos dessa energia passante e/ou transformada e ainda, centros onde ocorrem as principais tomadas de decisão sobre as condições operativas por eles monitoradas. Tais centros operativos são os que convencionalmente denominamos de **Subestações Elétricas**(SE's).

No sistema elétrico brasileiro atual existe uma diversidade muito grande de SE's. Cada uma delas foi projetada para se adequar à topografia e às necessidades energéticas de uma determinada região. Além disso, o custo para construir e manter uma SE é bastante elevado – na casa dos milhões de reais e sua operação exige um conhecimento altamente especializado. Isto demanda muito treinamento e implica em reduzir o número de pessoas que esteja apto a realizar este tipo de operação.

A diversidade de projetos e o alto custo para se construir uma máquina que simule o funcionamento de uma SE faz com que a tecnologia de Realidade Virtual (RV) se torne uma opção interessante, dada a flexibilidade que ela apresenta na construção dos mais diversos mecanismos e cenários e seu custo reduzido.

Para Kalawsky (1993), Realidade Virtual (RV) compreende um conjunto de tecnologias avançadas de Interface, que permitem ao usuário realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multissensoriais.

Leite, Silva e Ribeiro Filho (2010) afirmam que a indústria de há muito tempo tem utilizado esta ferramenta, principalmente, para as atividades de entretenimento, criando jogos que faz que o usuário desperte e se interesse em participar da atividade que está sendo proposta. Neste contexto a RV despertou ideia de ser utilizada em atividades educacionais, favorecendo um avanço no processo ensino aprendizagem. Assim surgiram os primeiros simuladores voltados à formação de técnicos em algumas áreas, favorecendo a assimilação e a vivência de situações virtuais comparáveis ao mundo real.

No início, o alto custo do *hardware* tornava o uso de aplicações de RV proibitivo. Capacetes, luvas e outros equipamentos eram extremamente caros. Mas, a tecnologia evoluiu e surgiu a possibilidade de se utilizar sistemas semi imersivos que utilizam hardware mais barato – necessita-se apenas de um computador com capacidade para processar gráficos3D – e obtém resultados tão bons quantos se o usuário estivesse conectado a um sistema totalmente imersivo.

Segundo Tori e Kirner (2006) a Realidade Virtual é categorizada como não imersiva, quando o usuário é transportado, parcialmente, ao mundo virtual, por meio de uma janela (monitor ou projeção, por exemplo), mas continua a sentir-se predominantemente no mundo real.



1.1 Motivação

Rossi (2004) afirma que para que o transporte do grande fluxo de energia pelas SE's de forma contínua e segura, essas SE's são providas hoje, de uma sofisticada estrutura de controle, realizada através de uma enorme quantidade de recursos computacionais, que auxiliam os seus processos supervisivos, os quais, operando em tempo real, monitoram as medições, as funções de controle, os alarmes, as proteções, os diversificados comandos, etc.

Segundo Costa, Pereira e Alves (2005) A qualificação dos operadores de sistemas elétricos de potência dos centros de operação tem sido uma preocupação constante nas últimas décadas, pois a cada dia que passa essa função de operador torna-se mais complexa. Cabe ao operador analisar as condições do sistema, implementar ações de controle, conhecer e aplicar, com segurança uma gama enorme de conceitos associados à operação do sistema.

Considerando o grande potencial da tecnologia de RV para criar interfaces mais intuitivas e ambientes realistas de alto desempenho e os baixos custos envolvidos na criação dos mesmos, a tecnologia de RV se mostra uma alternativa bastante adequada para a confecção de objetos de aprendizagem altamente interativos, como por exemplo, um Sistema de Realidade Virtual (SRV) para o treinamento dos operadores acima mencionados.

Assim outro ponto a favor da RV é o fato de que, usando tal SVR, o operador aprendiz pode treinar numa enorme variedade de cenários, simulando SE's de diversos tamanhos e adaptadas às mais diversas topografias.

De fato, as modalidades de treinamento tradicionais exigem deslocamento dos controladores de subestação. E estas modalidades, na maioria das vezes focam apenas na teoria, raramente, suportando interação e envolvimento com o objeto de aprendizagem como acontece naquela que fazem uso de um ambiente de RV.

Durlach e Mavor(1995) afirmam que tecnologias da informação como a RV podem ter importante função para melhorar a educação, considerando que é um potencial veículo através do qual o alcance das experiências em que os estudantes são expostos consegue ser amplamente aumentado; pode oferecer ambientes imersivos e interativos que fornecem grande contexto no qual problemas intelectuais interessantes aparecem naturalmente; e aumenta potencialmente a possibilidade de experiências colaborativas.

1.2 Objetivo central

O principal objetivo deste trabalho é investigar técnicas de RV aderentes à simulação de Subestações de Energia Elétrica.

Tal investigação implica em:

- a) Análise de ferramentas existentes em busca de uma metáfora de interação adequada;
- b) Levantamento de requisitos para a construção de uma ferramenta que atenda ao que é solicitado;
- c) Seleção de um ou mais padrões de projeto para software dentre os inúmeros existentes atualmente, que mais se adeque ao software que se pretende desenvolver;
- d) Escolha de uma linguagem de programação/ambiente de desenvolvimento que suporte a construção da metáfora no padrão de projeto escolhido;
- e) Validação junto aos *stakeholders* para verificação do cumprimento dos requisitos.

1.3 Objetivos específicos

São objetivos específicos:

- Investigar técnicas que permitam a construção do SRV;
- Avaliar as metáforas de interação adequadas ao treinamento de operadores de tais subestações:
 - Mostrar como as ações do operador no painel de controle do sistema supervisorio afetam a subestação como um todo (visualização *two-way*);
 - Permitir que o operador vivencie situações realísticas de perigo sem acarretar prejuízo financeiro e risco de vida;
- Definir estratégias para o desenvolvimento de SRV equivalentes a instalações reais de Subestações de Energia Elétrica;
- Permitir comunicação com um SGBD para que as ações do aprendiz possam ser persistentes e, posteriormente, recuperadas pelo instrutor.
- Avaliar o uso de tais ambientes virtuais no treinamento a distância de tais operadores.

Para isso, estabeleceram-se as seguintes metas:

- Fazer levantamento bibliográfico sobre as características de um bom sistema de RV;
- Avaliar aplicações correlatas a fim de determinar a contribuição do simulador proposto em relação a técnicas tradicionais;



- Verificar tecnologias que são mais adequadas para a construção do sistema protótipo;
- Identificar os componentes de uma SE e determinar como se dá a interação entre eles. Em seguida selecionar um deles para que seja construído um protótipo;
- Realizar a modelagem geométrica, física e lógica da simulação;
- Validar a simulação junto a estudantes, engenheiros e operadores de SE.

1.4 Organização da dissertação

Este trabalho está organizado da seguinte forma:

No capítulo 2 chamado **Fundamentos**, é realizada uma revisão de literatura para determinar as características desejáveis em um sistema de RV para treinamento. Nele, também é mostrada uma visão geral das tecnologias existentes para construção de ambientes virtuais. Finalizando o capítulo, também são apresentados os conceitos necessários para se compreender o que uma subestação e um transformador de força e qual a importância deste componente para o funcionamento da SE.

No capítulo seguinte, são analisadas aplicações com propostas parecidas e são levantadas suas principais características e contribuições.

No capítulo 4, **Especificação do sistema proposto** são discutidos os requisitos levantados junto aos usuários para a construção de um modelo realista e de que forma as diversas camadas da aplicação proposta devem interagir.

Em sequência, o capítulo 5, **Detalhes da implementação** discute sobre as tecnologias escolhidas para a construção do simulador e de que forma elas interagem entre si.

No capítulo 6, **Resultados e limitações**, são discutidos os resultados da avaliação feita pelos estudantes que possibilitarão descobrir as qualidades do simulador e seus eventuais defeitos, para que ele possa ser melhorado.

No Capítulo 7, **Conclusões** são discutidos os resultados alcançados e, finalizando, no capítulo 8 - **Trabalhos futuros** são comentadas novas pesquisas que podem se originar de melhorias no presente trabalho ou mesmo novas contribuições inéditas com a mesma proposta.



Capítulo 2 - Fundamentos

2.1 Introdução

Este capítulo tem por finalidade, apresentar, através de revisão de literatura, os principais conceitos inerentes à tecnologia conhecida como Realidade Virtual. Tais conceitos subsidiarão a construção do Sistema de Realidade Virtual (SRV) proposto, referenciado como Virtual Substation.

Em princípio, apresenta-se a definição formal de Realidade Virtual e quais as características que uma interface computacional deve ter para receber esta classificação. A seguir, apresenta-se um breve histórico e a distinção entre Realidade Virtual imersiva e não imersiva. Também são apresentados os três pilares da tecnologia (imersão, navegação e envolvimento). Após isto, são discutidas as vantagens e desvantagens deste tipo de ambiente computacional.

Finalizando o capítulo, são discutidos brevemente os conceitos de Subestação Elétrica e de Transformador de Força extremamente importantes para a compreensão do que do Funcionamento do Virtual Substation.

2.2 Realidade Virtual

Segundo Tori And Kirner (2006) a Realidade Virtual (RV) é, antes de tudo uma “interface avançada de usuário” para acessar aplicações encontradas no computador, tendo como características, a visualização de, e movimentação em ambientes tridimensionais em tempo real e interação com elementos deste ambiente.

Pimentel define Realidade Virtual como sendo o uso de tecnologia para convencer o usuário de que ele está em outra realidade. Em geral, Realidade Virtual refere-se a uma experiência interativa e imersiva baseada em imagens gráficas tridimensionais geradas em tempo real por computador (Pimentel, 1995).

Outro aspecto dos ambientes virtuais é que, para tornar a interação do usuário com o ambiente virtual mais próxima da experiência real, a RV prioriza a interação em tempo real em detrimento da qualidade das imagens e dos sons, por exemplo.

A RV se preocupa em proporcionar aos usuários uma interação mais natural, como uso de mãos em ambientes virtuais renderizados na tela do monitor, em projeções em tela, ou em projeções nos olhos, por meio de capacetes de Realidade Virtual (Head Mounted Display – HMD).

No entanto, para que o usuário consiga interagir com o ambiente virtual, é necessário o uso de dispositivos especiais multissensoriais, como luvas com rastreadores, dispositivos com reação de tato e força, mouses 3D, óculos estereoscópicos, fones de ouvido ou alto-falantes com sons espaciais, etc. (Bowman 2005).

2.2.1 Os três pilares da Realidade Virtual

Para Kalawsky (1993), imersão, navegação e interação são os três pilares da RV. E são elas que diferenciam um ambiente de RV de um ambiente computacional tradicional, em três dimensões.

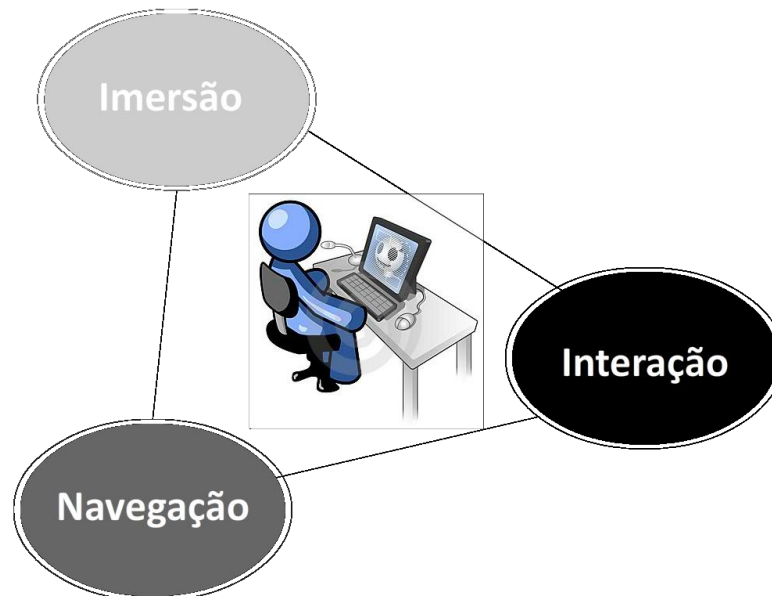


Figura 1 - Os três pilares da RV

Segundo Botega E Cruvinel (2009) a ideia de **imersão** está intimamente ligada ao sentimento de fazer parte do ambiente.

Já para Rodello E Brega (2011) pela característica da imersão, a RV permite uma associação mais estreita entre os meios e as mensagens, uma vez que os usuários entram no canal de mídia e se tornam uma parte dela.

Botega E Cruvinel (2009) acrescentam que a ideia de envolvimento está ligada ao grau de motivação para o engajamento de uma pessoa em determinada atividade. O envolvimento

pode ser passivo, como ler um livro ou assistir televisão, ou ativo, como participar de um jogo com algum parceiro. A RV tem potencial para os dois tipos de envolvimento ao permitir a exploração de um ambiente virtual e propiciar a interação do usuário com o mundo virtual mais intuitiva.

Botega E Chuvinhe (2009) salientam ainda que a **interação** está ligada à capacidade do computador detectar as entradas do usuário e modificar, instantaneamente o ambiente virtual em função das ações efetuadas sobre ele (capacidade reativa ou feedback).

Na prática, a RV permite que o usuário realize **navegação** e observe o um mundo tridimensional sob seis graus de liberdade (6 DOF). Na essência, a RV é um espelho da realidade física, na qual o indivíduo existe em três dimensões, tem a sensação de estar imerso no ambiente e tem a capacidade de interagir com o mundo ao seu redor. Os dispositivos de RV simulam essas condições, chegando ao ponto em que o usuário pode tocar, virtualmente, os objetos de um ambiente virtual e fazer com que eles respondam, ou mudem, de acordo com suas ações (Von Schweber, 1995).

2.2.2 Sistemas imersivos e não imersivos

Normalmente, um sistema imersivo é obtido com o uso de capacete de visualização, cavernas e projeções das cenas nas paredes, teto e piso (Cruz-Neira, 1992). Além do fator visual, dispositivos ligados aos demais sentidos também são importantes para o sentimento de imersão, principalmente o som (Begault, 1994; Gradecki, 1994), além do posicionamento do usuário e dos movimentos da cabeça. A visualização de uma cena 3-D em um monitor é considerada não imersiva (Leston, 1996).

Entretanto, dispositivos baseados nos demais sentidos podem introduzir algum grau de imersão à RV utilizando monitores (Robertson, 1993). Ainda assim, os mesmos ainda apresentam alguns pontos positivos, como o baixo custo e a facilidade de uso, evitando as limitações técnicas e problemas decorrentes do uso do capacete. Todavia, a tendência deve ser a utilização da RV imersiva, considerando que a imersão, aliada à interação, justifica-se como o grande propósito das aplicações em RV.

Segundo GRILO et al. (2001), as formas não-imersivas, amplamente utilizadas em jogos e na manipulação de maquetes eletrônicas via Internet, têm como vantagem o baixo custo do equipamento e a facilidade de acesso em qualquer tipo de computador, tal como uma máquina de médio porte e monitor, teclado e mouse para a entrada de dados.

Conforme Botega e Cruvinel (2009) estes dispositivos são periféricos de usabilidade comum, destinado aos usuários comuns e sistemas de propósito gerais, como: teclado, mouse, trackballs, digitalizadores de mesa (scanners), canetas digitais e microfones. No âmbito de sistemas de RV, os dispositivos não convencionais de entrada de dados podem contribuir para o desenvolvimento de ambientes interativos e não imersivos, porém, eles não provêm o realismo e o envolvimento tal qual fornecidos por sistemas virtuais dotados de dispositivos não convencionais.

Ainda segundo GRILO et al. (2001) no entanto, a RV imersiva ou semi imersiva proporciona uma maior amplitude de aplicações. Utilizando capacetes de Realidade Virtual (Head Mounted Displays), luvas (data gloves), BOOM (Binocula omni-orientation monitor) ou CAVEs (Cave Automated Virtual Environment), dentre outros equipamentos de imersão, tem-se uma integração do usuário com objetos virtuais, criando aplicações cuja intensidade sensorial possibilita uma experimentação mais profunda do ambiente virtual. Incluem-se, neste aspecto, fatores sensoriais de calor, frio, toque e deslocamento.

O sistema *Virtual Substation* proposto neste trabalho é um sistema não-imersivo. No entanto, pretende-se que ele não fique atrelado à plataforma PC e sim que possa ser executado numa grande variedade de ambientes computacionais, como os cada vez mais populares smartphones e tablets.

2.2.3 Vantagens e limitações da Realidade Virtual

Ao falar de ambientes virtuais, GRILO et al. (2001) salientam a principal vantagem dos modelos computadorizados deve-se à tridimensionalidade e a disponibilização de diferentes pontos de observação, tanto internos quanto externos. Renderizações e perspectivas podem ser, automaticamente, geradas com um mínimo de esforço humano. Animações e simulações podem ser geradas empregando o mesmo modelo. O realismo constitui um dos postos-chaves dos modelos gerados por computador, devido à tridimensionalidade, à representação dos elementos arquitetônicos em escala e à simulação de texturas e efeitos luminosos nas superfícies e volumes.

A despeito de todas as vantagens oferecidas por esta tecnologia, ela ainda possui uma série de limitações.

A primeira delas diz respeito ao fotorrealismo. Como os ambientes virtuais são projetados tendo em mente a interação em tempo real com o usuário, muitas vezes o realismo da simulação é prejudicado. Isto pode ser um problema em aplicações que tem necessidade de visualização de detalhes pequenos, como por exemplo, na medicina e na arquitetura. Com o



desenvolvimento de computadores com maior poder de processamento, este problema tende a diminuir. Ferramentas como Unity 3D já possibilitam a criação de ambientes virtuais com alto grau de realismo e tempo de resposta satisfatórios.

Outro problema diz respeito aos dispositivos usados, principalmente nos ambientes virtuais imersivos. Segundo Tori E Kirner (2006), a necessidade de fazer uso de aparatos tecnológicos para a interação do usuário com o ambiente virtual provoca restrições tanto pelo aspecto econômico e tecnológico quanto pelo desconforto. Porém permite ao usuário fazer coisas que eram impossíveis ou inviáveis.

Ainda sobre este aspecto, Tori E Kirner (2006) salienta que uma das limitações da RV é que dependendo da complexidade da aplicação, o uso simultâneo de diversos dispositivos (por exemplo, óculos estereoscópicos para imersão e um dispositivo háptico para interação) pode aumentar a demanda de tempo de processamento. Para Nunes et al (2011), a definição dos limites de influência de cada dispositivo são pontos estratégicos que devem ser planejados tecnicamente, pois exigem processamento paralelo e implementação cuidadosa, garantindo que cada equipamento possa proporcionar a sensação para a qual foi designado, no momento correto.

Este quadro está melhorando, com a invenção de dispositivos de controle como os joysticks do tipo Wii Remote e consoles para controle de aplicações computacionais por gestos como o *Kinect* da empresa Microsoft.

Segundo Wilcox (1998), avatar é toda representação de personagem controlado por alguém no mundo real.

Aguiar, Soares E Campos (2009) afirmam que todo esse conjunto de tecnologias do wiimote permite, na realidade, é que os movimentos do usuário sejam traduzidos mais fielmente para o mundo de jogo. A imersão e a identificação com seu avatar tende a ser maior, devido à maior associação entre suas ações e às do personagem controlado em tela.

Outro entrave enfrentado pelos ambientes de RV – principalmente os não imersivos – é que para haver imersão total é necessário certo grau de suspensão da descrença por parte do usuário.

Segundo Roman (2010), esse termo surgiu dos críticos do cinema e é definido como a vontade do espectador de acreditar nas imagens como sendo reais e desconsiderar o meio que o cerca. O espectador pensa dessa forma para substituir o julgamento pelo entretenimento.

Ainda Segundo Aguiar, Soares E Campos (2009) utilizando dispositivos de navegação do tipo wiimote, a suspensão de descrença, necessária para aproveitar melhor a narrativa apresentada, passa a ser mais facilmente atingida, devido a essa relação direta. E, até onde o



jogador é capaz de perceber, instantânea entre o que faz e a reprodução de suas ações no jogo, pelo seu avatar, e isso é útil em vários aspectos.

Outro entrave é a pouca quantidade de mão de obra capacitada para desenvolver software de Realidade Virtual.

2.2.4 Subestações elétricas

Segundo Brasil (2012), uma subestação é um conjunto de instalações elétricas em média ou alta tensão que agrupa os equipamentos, condutores e acessórios, destinados à proteção, medição, manobra e transformação de grandezas elétricas.

Cemig (2003) define subestação como uma estação com uma ou mais das funções de gerar, medir, controlar a energia elétrica ou transformar suas características, quando fazendo parte das instalações de utilização (instalações de propriedade do consumidor).

Segundo Mamede Filho (2010) subestação central de transmissão é aquela normalmente construída ao lado das usinas produtoras de energia elétrica, cuja finalidade é elevar os níveis de tensão fornecidos pelos geradores para transmitir a potência gerada aos grandes centros de consumo.

No presente estudo, será criado um simulador de uma subestação central de transmissão, conforme descrito por Brasil (2012).



Figura 2 - Subestação elevadora de Juiz de Fora (MG) extraído de <http://www.pracomprar.com/v2/abrir/site/blp-engenharia/servicos/1420>



Figura 3 - Subestação elétrica da usina de Igarapava - SP

As figuras 2 e 3, mostram exemplos de subestações elétricas. A Figura 2 apresenta uma imagem da subestação elétrica de Juiz de fora que será simulada pelo Virtual Substation. Na Figura 3, tem-se uma imagem da subestação da usina hidrelétrica de Igarapava. Por essas imagens, é possível perceber que existem vários arranjos de subestação. A subestação de Igarapava é uma subestação blindada e usa basicamente equipamentos a gás, enquanto que a subestação de Juiz de Fora é uma subestação a céu aberto e mesmo pertencendo a uma mesma categoria apresentam grandes diferenças entre si.

2.2.5 Transformador de força

Segundo Siemens (2012) a constante evolução tecnológica a nível mundial faz com que a energia elétrica seja uma real necessidade no presente e no futuro. Um componente fundamental no sistema de geração, transmissão e distribuição dessa energia são os transformadores de força. Este é o principal componente de uma subestação e por isso o protótipo inicial simulará a tela de controle de um transformador de força.

A Figura 4 apresenta a imagem de um transformador de força.



Figura 4 - Transformador de Força

O Transformador de Força é um transformador para uso em subestações com o objetivo de fazer a conexão entre os sistemas de transmissão e distribuição.

2.3 Conclusão

Neste capítulo foi apresentada a definição formal da Realidade Virtual. Mais de uma mera interface computacional tridimensional, ela se mostra uma rica alternativa para a construção de metáforas de interface, pois para ser considerado um SRV, um *software* deve apresentar um



ambiente de fácil navegação, altamente interativo e dentro do qual o usuário se sinta total ou parcialmente imerso.

Também foi apresentada a diferença entre ambiente virtual imersivo e não imersivo. Para a construção do protótipo optou-se pela segunda abordagem, uma vez que para interagir com o mesmo não será necessário o uso de óculos, luvas ou qualquer outro dispositivo de *hardware* adicional, que o que encareceria sua execução. Tal ausência não empobrecerá a experiência do usuário, pois muitas vezes a imersão depende mais de fatores psicológicos que físicos.

Finalizando, procurou-se apresentar os conceitos relativos a subestações elétricas e transformadores de força, o que possibilitará uma maior compreensão do restante deste trabalho - principalmente do ambiente que se pretende simular.



Capítulo 3 - Trabalhos relacionados

3.1 Introdução

Nesse capítulo, são apresentados cinco sistemas que, de alguma forma, têm relação com o simulador que se pretende construir. São analisadas ferramentas de diversos tipos: desde ferramentas para CAD como o *arqElectrical*, simuladores de operações rotineiras de manutenção elétrica como o RV-COPEL, sistemas de controle de mineração, como o MineInside® até simuladores de subestação e usina como os desenvolvidos por FURNAS.

Todas estas ferramentas se assemelham ao Virtual Substation, seja na forma como constroem suas metáforas de interação (*ArqElectrical*, MineInside®) como no seu propósito (RV-COPEL, simuladores Furnas 1 e 2). Além da apresentação de cada uma delas, e seus pontos fortes e eventuais deficiências são ressaltados.

Após a exposição das ferramentas, são elencados os quesitos que uma nova ferramenta deve atender para se mostrar uma contribuição relevante. São eles: possibilidade de visualização 3D, uso de *two-way*, comunicação com SGDB e a existência ou não de uma versão para internet. Estes quesitos são detalhados e ao final, é compilada uma tabela indicando quais deles as ferramentas analisadas atendem ou não.

3.2 Aplicações correlatas

3.2.1 *arqElectrical*

O *ArqElectrical* descrito em Silva (2006) é um sistema de treinamento proposto inicialmente, para que alunos de Engenharia Elétrica possam compreender melhor de que forma acontece o planejamento de uma instalação elétrica residencial. Para tanto, ele é capaz de ler a planta arquitetônica de uma residência em 2D e exibi-la simultaneamente em 2D e 3D para o aluno construir o projeto elétrico de uma residência.

Após ler a planta – que deve estar no formato .DXF – o sistema permite ao usuário cadastrar os pontos de energia e estas são armazenados como vértices de um grafo. O cálculo da potência das lâmpadas será em função da área e do tipo do cômodo da residência.

Além disso, o sistema ainda conta com um arquivo de onde pode ser consultada a potência dos aparelhos eletro eletrônicos.

Os eletrodutos que interligam os pontos de energia também são lançados internamente pelo sistema como arcos do grafo. Os pontos de energia no teto também formam uma malha de eletrodutos, ligados por meio de uma triangulação. Estes pontos de energia do teto também são lançados de forma incremental no grafo.

Sempre que o usuário quiser começar um novo projeto no ArqElectrical, ele deve obedecer a uma sequência padrão:

- a) O primeiro passo sempre é colocar uma lâmpada como primeiro ponto de energia.
- b) Em seguida, o sistema cria a tubulação.
- c) O quadro de energia só deverá ser lançado quando todos os pontos de energia tiverem sido colocados.
- d) Em seguida, por meio da interação do usuário com o sistema, o segundo calcula o tipo de fornecimento de energia por parte da concessionária.
- e) Após o lançamento do quadro e a definição de rotas na malha de eletrodutos do teto são formados os circuitos elétricos.
- f) Ao final desta operação, o sistema gera um relatório, contendo critérios e tabelas utilizadas.
- g) Além de permitir a visualização 3D, o sistema também permite a navegação, imersão e interação com o modelo 3D, permitindo ao usuário, a manipulação de objetos.

Abaixo, a Figura 5 mostra a interface de usuário do ArqElectrical.

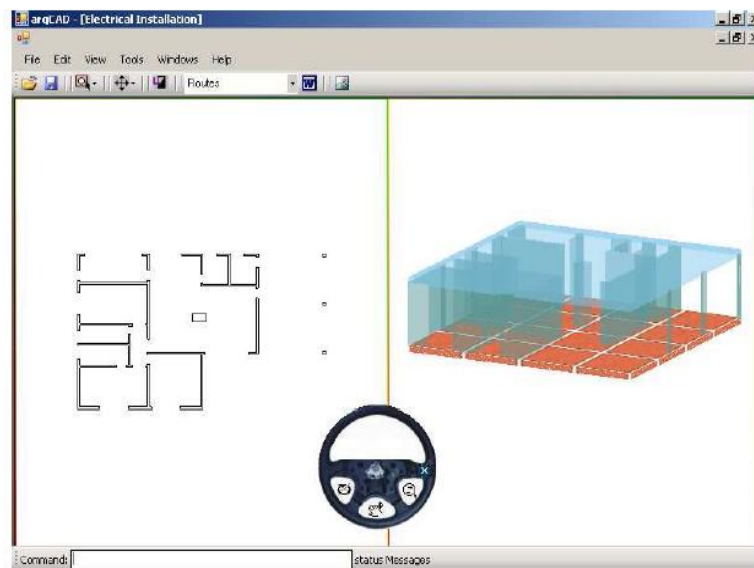


Figura 5 - interface gráfica com usuário (GUI) do sistema ArqElectrical

Fonte: (SILVA, 2006)

Esta interface é uma interface do tipo desktop e apresenta as já tradicionais barras de menu e de botões, típicas dos ambientes WIMP (Window, Icon, Menu, Pointing Device).

Abaixo, ficam a barra de desenhos com viewports em 2D e 3D e o volante de navegação.

No canto inferior da tela, ficam a barra de comando e a barra de status. Segundo seus autores, o sistema responde a certos comandos como importar, salvar, mover etc. Enquanto a barra de status mostra a ação corrente para o usuário.

O sistema também dá ao usuário a possibilidade de criar e manipular componentes elétricos – como lâmpadas e tomadas – além de oferecer suporte a operações de navegação. A visualização do projeto pode se dar em 2D ou 3D ou os dois simultaneamente, o que não é comum nas aplicações de CAD (Computer-Aided Design) e CAE (Computer-Aided Engineering) disponíveis no mercado.

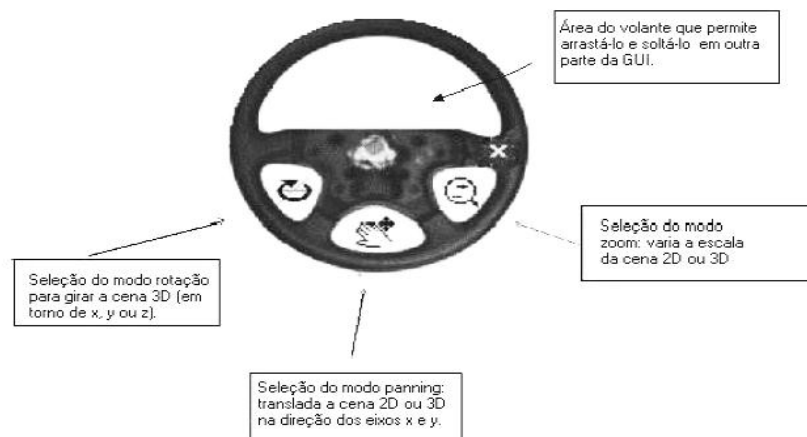


Figura 6 - Volante de navegação do sistema ArqElectrical

Fonte: (SILVA, 2006)

A principal função do volante mostrado na Figura 6 é permitir ao usuário selecionar o modo de navegação no ambiente tridimensional. Por meio de seus botões é possível rotacionar o ambiente 3D e aumentar o zoom, por exemplo, entre outras funções.

Internamente, o sistema possui implementado o Gerenciador de Restrições, que controla as ações do usuário executadas por meio da interface gráfica. Ele atualiza o gráfico de restrições segundo essas ações. Ele também é responsável por detectar colisões entre os eletrodutos do teto. Caso ocorra alguma colisão, os eletrodutos são redesenhados a fim de contornar obstáculos.

Segundo os autores do ArqElectrical:

O Grafo de Restrições mantém em seus vértices as informações geométricas e de engenharia das cargas elétricas. Além disso, ele mantém em seus arcos as informações referentes as restrições

(eletrodutos) entre duas cargas elétricas. Os vértices e os arcos possuem métodos de visualização automática em 2D e em 3D. (SILVA, 2006).

Apesar de permitir a visualização two-way, a persistência dos dados só pode ser feita em arquivos texto e a aplicação só pode ser executada em ambiente Windows.

3.2.2 RV-COPEL

O projeto RV-COPEL, descrito em Buriol E Rosendo (2009), segundo seus autores, tem por objetivo o desenvolvimento de um Ambiente Virtual para o treinamento de atividades de manutenção em linha viva, utilizando as técnicas de RV combinada com interfaces inovadoras.

A manutenção de equipamentos em linha viva (com circuitos energizados) além de ser extremamente perigosa, demanda conhecimento especializado. Dentre as várias atividades que podem ser realizadas as mais comuns são:

- troca de isolador de pino,
- troca de isolador de disco, troca de chave fusível,
- troca de para-raios,
- troca de cruzeta, poda de árvores,
- emenda e/ou reparo de condutores,
- instalação de chave faca,
- retirada de objetos estranhos da rede
- e substituição de chave unipolar.

Todas elas são atividades que requerem um conjunto de procedimentos distintos e específicos. Para a avaliação do protótipo, escolheu-se a troca de cruzeta.

Nesta operação, os vários passos que devem ser executados pelo treinando no AV se resumem a escolher um objeto e posicioná-lo no lugar certo. Estes passos devem ser realizados numa sequência correta e aspectos como peso e a inércia dos equipamentos são relevantes.

Para a construção do ambiente, parte-se da premissa de que o usuário poderá realizar a simulação livremente e seu desempenho será monitorado de forma transparente, e no fim, será fornecido um resultado.

O treinando já inicia a simulação próximo ao condutor energizado e indicará ao sistema qual equipamento será selecionado e onde ele será posicionado. Além disso, ele deverá estar atento para as distâncias e manipulação segura do objeto.

O treinamento acontece com apenas um profissional por vez, embora em situações do mundo real a manutenção em linha viva é realizada por equipes de três deles.

Para a manipulação dos elementos, é usado o controle Wii Remote.

Por meio da câmera infravermelha do Wii remote é possível rastrear a posição da cabeça do usuário e atualizar a posição da câmera virtual na cena. É possível se aproximar para aumentar o campo de visão ou se movimentar para o lado a fim de observar objetos ocultos na cena.

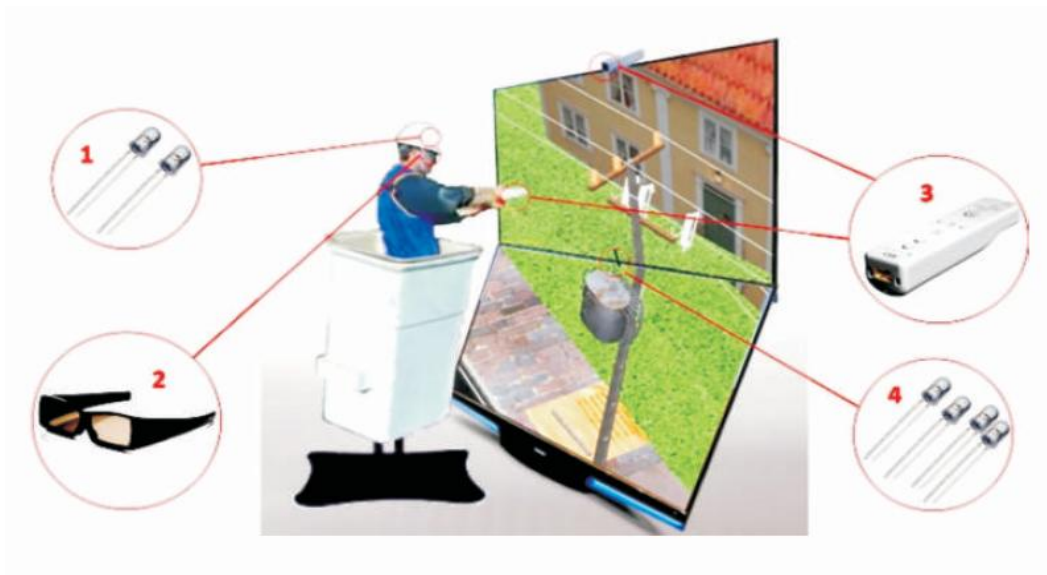


Figura 7 - Plataforma proposta pelo projeto RV-COPEL

Fonte: (BURIOL E ROSENDO, 2009)

A programação do RV-COPEL foi realizada buscando-se a integração de bibliotecas e kits de desenvolvimento – a maioria *open source* – para as seguintes funções:

- Rendering, gerenciamento e otimização de grafo de cena
- Simulação física (dinâmica e colisão);
- Controle de áudio;
- Controle de dispositivos de entrada e saída de dados

Uma das características interessantes da manutenção da linha viva é que neste tipo de atividade o electricista permanece em uma cesta isolada, o que limita seus movimentos e a posição da cabeça não ultrapassa meio metro em torno da posição central, o que propicia a utilização de um *headtracker* (rastreador da posição da cabeça). O ponto de vista do usuário é rastreado por meio da câmera infravermelha do Wii remote, que capta os sinais de um par de LEDs (diodos de emissão de luz) acoplado aos óculos do usuário. A desvantagem deste tipo de sistema é que o usuário precisa estar no campo de visão da câmera.



Figura 8 - Seleção dos objetos na cena com o wiimote no sistema RV-COPEL
Fonte: (BURIOL E ROSENDO, 2009)

Além de rastrear a posição da cabeça do usuário, o Wii remote (ou “wiimote”) também pode ser usado para manipular os objetos da cena, funcionando de forma análoga a um *mouse*. O pressionamento dos botões do wiimote permite ao usuário pegar os objetos na tela e soltá-los.

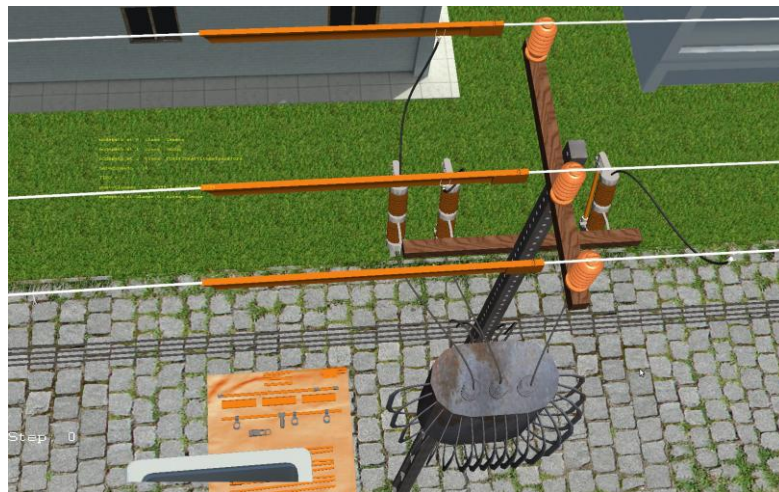


Figura 9 Vista do usuário próximo ao poste com os condutores cobertos no sistema RV-COPEL

Fonte: (BURIOL E ROSENDO, 2009)

3.2.3 MineInside

A atividade de mineração é uma das mais importantes para a manutenção da atual cadeia produtiva.

Toda mina possui uma sala de controle de onde a equipe de controle de processos controlam fatores como controle de tráfego operacional, manutenção de rotas, extração de minério, controle de qualidade, entre outros. Isso exige que o operador que deseja ingressar nesta equipe seja um profissional extremamente treinado. Todo esse processo é realizado utilizando-se telemetria e GPS, que utilizam softwares e hardware bastante específico.

A questão é que, enquanto está sendo treinado para utilizar todo este aparato, o operador em potencial nem sempre consegue ter uma visão do processo como um todo. Isso pode ser ainda mais dificultado pelo fato de em alguns lugares da mina – principalmente nas áreas de escavação, não ser possível a instalação de um circuito fechado de TV.

Na tentativa de fechar esta lacuna, foi desenvolvido o Mineinside – descrito em Coelho, ET al.(2011). Este software é usado atualmente na mina de Carajás, no estado do Pará, controlada pela companhia Vale.

O MineInside é uma plataforma de RV que permite ao operador ter uma visão completa de toda o processo de mineração e simula o ambiente de uma mina. Este AV funciona de forma interligada com todos os outros softwares, mostrando a operação de mineração em uma tela em tempo real. Segundo seus idealizadores o ambiente ajuda, inclusive a reduzir o tempo ocioso do maquinário, o que acaba por tornar mais eficiente o processo como um todo, além de tornar a operação mais segura.

Como a mina em questão é uma mina a céu aberto, e possui características diferentes, os autores do MineInside salientam que ele também leva em consideração informações topográficas e geológicas e suas mudanças durante a mineração.

Um ambiente virtual para ser usado em uma mina a céu aberto é complexo. Ele precisa mostrar camadas diferentes, como composição geológica, química e topográfica. Isso se torna ainda mais complicado, pelo fato de a extração de minério mudar a topografia da área a cada dia, por isso é necessário o uso de fotos de satélite e medições de GPS de alta precisão para fazer as mudanças necessárias no background do software toda semana.

A Figura 10 mostra uma tela do MineInside ® exibindo a operação de tráfego ao fundo e as máquinas selecionadas em zoom. Cada máquina selecionada pode mostrar dados como: velocidade, motorista, localização, se está carregada ou não. No caso de caminhões, o percurso e operação de um trator ou no caso de uma escavadeira os parâmetros geológicos e químicos do minério.

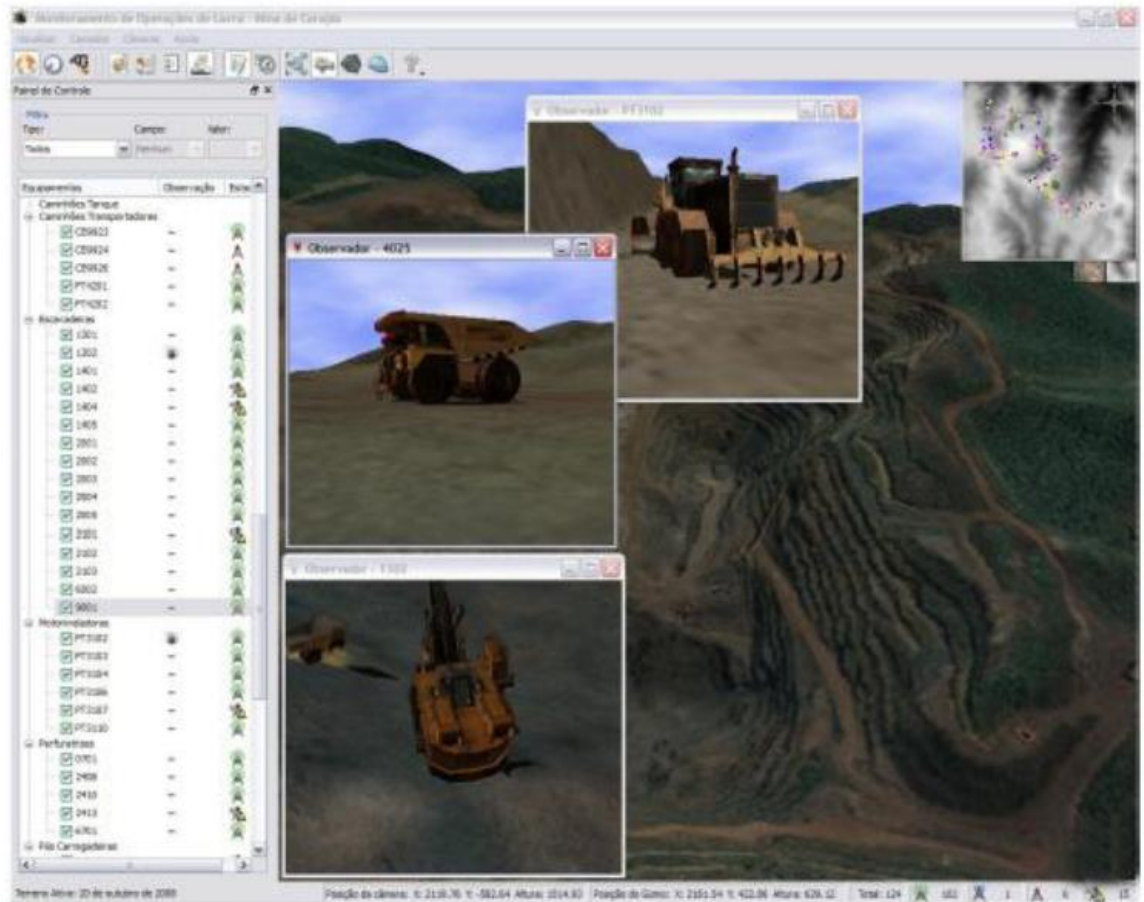


Figura 10 - Tela do MineInside®, mostrando a operação de mineração os diversos veículos no detalhe.

Fonte: (COELHO, ET al., 2011).

Cada tipo de máquina da área de campo (caminhões, tratores e escavadeiras) é modelado, e usando um kernel desenvolvido em C++ é inserido no plano de fundo. Os objetos virtuais são programados para mudar de posição de acordo com alterações nos dados do GPS da máquina correspondente. Nesta etapa, o software fornece ao operador uma câmera virtual e o parâmetro de velocidade de cada máquina. A fim de evitar congelamento de imagens, foram utilizadas técnicas específicas de processamento para reduzir o número de polígonos e texturas de cada máquina, enquanto o operador aumenta ou diminui o zoom.

O processo de automação utiliza diversos softwares diferentes e cada um deles fornece informações em uma tela diferente e usa uma base de dados específica para obter e armazenar informações. O MineInside® integra todas essas informações em uma única tela, o que torna o processo de operação mais amigável. Ele também integra todos os sistemas de telemetria das máquinas, aumentando o tempo de produção dessas máquinas e ajuda os operadores a evitar

falhas críticas. Se necessário, é possível desligar máquinas remotamente em emergências, como colisões, alta pressão de um pneu ou alta temperatura dos motores.

O Mineinside também pode ser usado para treinamento. É possível, por exemplo, recuperar informações de um turno anterior de serviço.

Isso permite aos gerentes entender um período de baixa produtividade ou um incidente específico e poderia ser usado pelos supervisores para treinamento e simulação do processo de controle.

O *software* só pode ser executado no ambiente *desktop* e não possui versão para internet, o que não é necessariamente uma desvantagem, já que em ambiente corporativo é bastante comum o uso de soluções fechadas.

3.2.4 Simulador Furnas 1

O simulador descrito por Ferrara et al. (2009) foi criado por uma equipe multidisciplinar da empresa FURNAS CENTRAIS ELÉTRICAS S.A.

Ele é totalmente integrado com o sistema SOL (também desenvolvido por Furnas), responsável por registrar todas as operações de um operador em campo. Ele funciona da seguinte forma:

Uma série de registros de histórico de uma situação real é obtida do banco de dados do sistema SOL e partir deste ponto o aluno deve interagir com o sistema como se aquela situação estivesse realmente em progresso.

Usando o simulador, o aluno pode manipular os seguintes valores:

- injeções de potência ativa e reativa,
- injeções de ajuste de potência ativa e reativa nas barras fronteiras e externas,
- tensões e taps de transformadores.

Para criar casos simulados e interagir com o treinando, o instrutor faz uso do programa SIMULASOL. Nele, é possível que o instrutor possa manipular dados analógicos ou digitais do sistema. Na Figura 11 é mostrada a interface do simulador de treinamento.

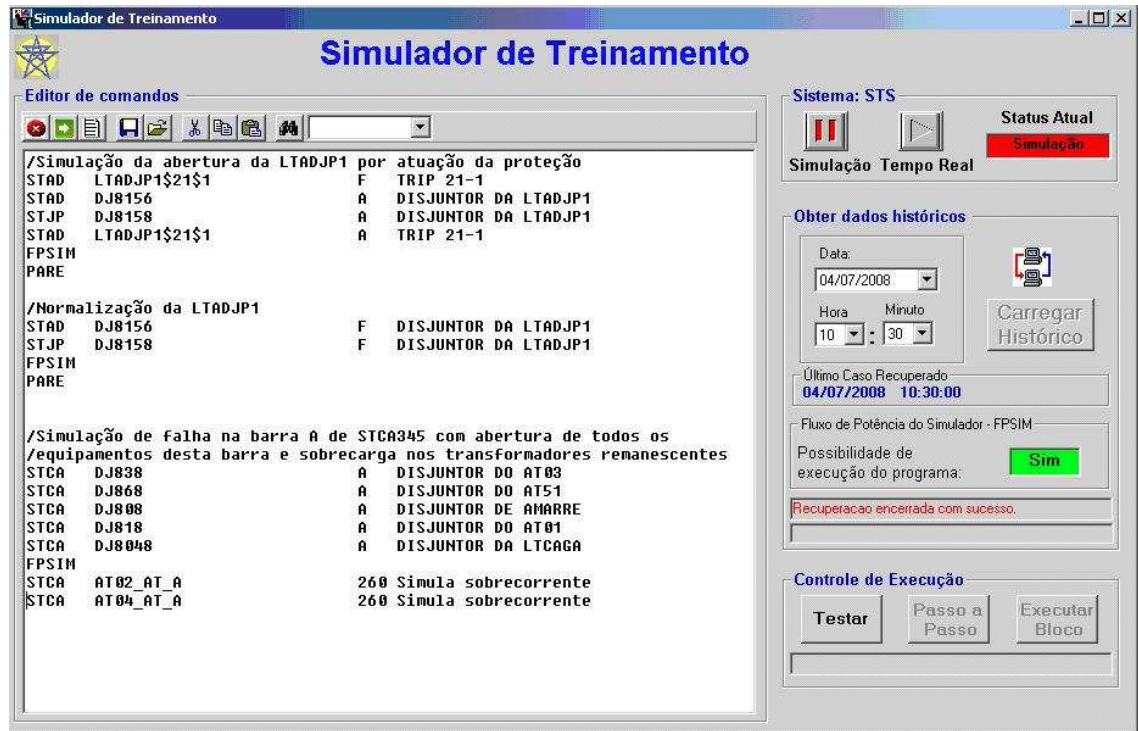


Figura 11 Interface para programação do Simulador de Treinamento FURNAS

Fonte: (FERRARA et al. , 2009)

O Simulador de Treinamento pode ser colocado em dois modos de execução: tempo real ou simulação.

No segundo caso, as ordens de manobras do treinando devem ser colocadas em um “script” de comandos que serão executados pelo programa. Este script pode ser editado para alterar os valores de

- estados digitais (disjuntores, seccionadoras e estado de proteções)
- medidas (tensão, corrente, MW e MVAR) do sistema, além de permitir simular a atuação da proteção.

Existe ainda a possibilidade de testar o script antes de executá-lo, ao se clicar um botão específico. A execução pode ser feita passo a passo ou em blocos de comando.

Os scripts podem ser salvos ou carregados de arquivos rtf (“rich text format”).

Uma coisa importante que se deve ressaltar sobre o simulador FURNAS é que se pretende simular de forma mais fiel possível o ambiente de trabalho do treinando e que o simulador descrito acima é apenas uma parte de uma estrutura maior, composta pelas seguintes partes:

- Sala de equipamentos, onde fica o Sistema de Treinamento e Simulações (STS). Ela possui a mesma estrutura de servidores e equipamentos de um Centro de Operações Real.

- Sala de treinamento: onde é reproduzido o ambiente da sala de controle, com um ou mais consoles de treinamento, idênticos aos consoles de operação, e é neste local que os treinandos realizam seus exercícios.
- Sala do instrutor: é um ambiente em separado, com um console idêntico ao do treinando, onde são preparados os exercícios e é conduzido o treinamento.

A Figura 12 mostra a sala de treinamento, reproduzindo o ambiente de trabalho de um operador.



Figura 12 - Sala de treinamento do simulador FURNAS

Fonte: (FERRARA et al. , 2009)

3.2.5 Simulador Furnas 2

O simulador Furnas descrito em Alves e Ribeiro Júnior (2011) é uma evolução do sistema apresentado no item anterior. No entanto, dada a mudança radical realizada na interface e no modo de operação, ele acabou merecendo um capítulo a parte.

Por meio deste simulador podem ser simuladas situações cotidianas e situações de crise.

O processo de treinamento é coordenado pela figura de um tutor. Este tutor será responsável por programar os cenários de treinamento que serão utilizados pelo treinando. Para esta programação, o sistema disponibiliza uma interface gráfica de usuário, mostrada na Figura 13 - Interface gráfica para programação de cenários.

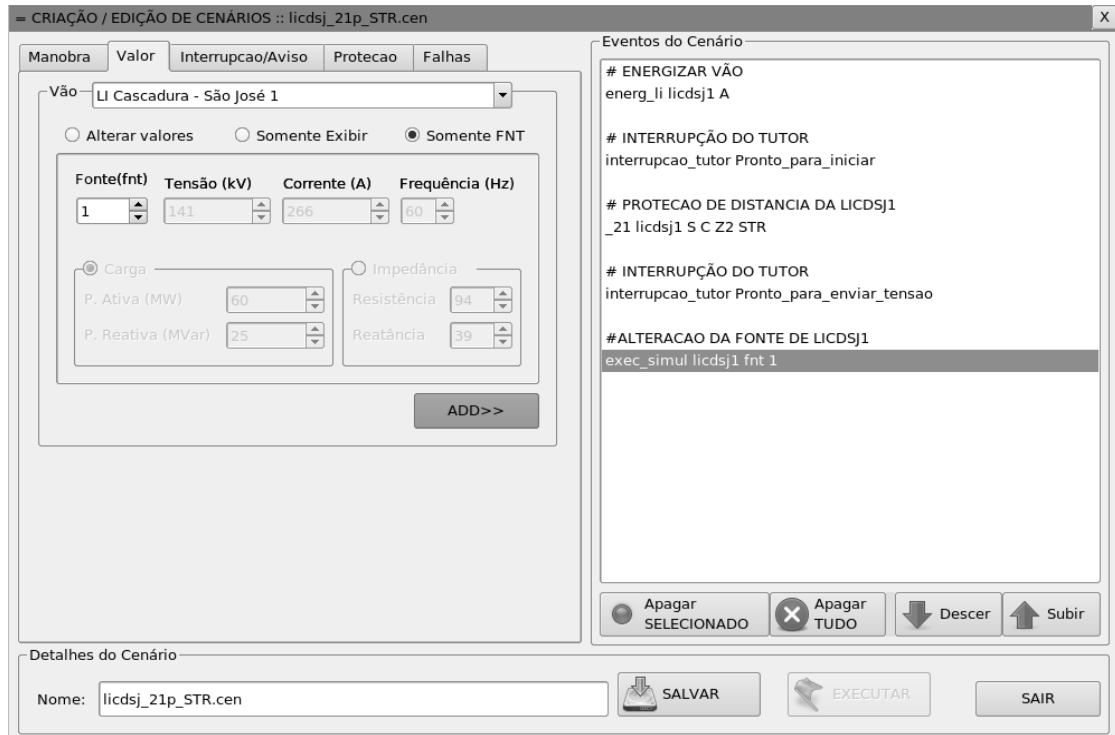


Figura 13 - Interface gráfica para programação de cenários

Fonte: (ALVES E RIBEIRO JÚNIOR, 2011)

Esta aplicação proporciona um ambiente gráfico para programação de eventos, o que permite criar ou modificar um Cenário de Simulação. O tutor normalmente fica numa sala separada do treinando onde há um terminal que permite ele monitorar toda a simulação.

Com o intuito de reproduzir o mais fielmente o ambiente de trabalho do operador, sua área de treinamento conta com um terminal telefônico, e rádio intercomunicador. Assim como no simulador anterior, há um terminal para acessar o Sistema Aberto de Gerenciamento de Energia (SAGE/CEPEL), que opera na modalidade de treinamento.

Já em Instalações onde os equipamentos do plantel são controlados de forma mista (parte dos equipamentos são controlados por meios computacionais, parte dos equipamentos são controlados por painéis elétricos), são necessárias duas estações de trabalho – em uma delas está instalado o SAGE na modalidade de treinamento e no equipamento restante está instalado o “Simulador Tridimensional em Primeira Pessoa” (STriPP).

Esta aplicação permite reproduzir a interação do Treinado com um painel elétrico e representa de forma realista, o ambiente de uma Sala de Controle, seus equipamentos, e seus componentes para proporcionar a um indivíduo a experiência de interagir de forma absolutamente segura com os punhos e botões em um “painel elétrico virtualizado”, facilitando a ele adquirir perícia na utilização deste tipo de interface de controle.

Dentro do ambiente STriPP, a navegação e o controle dos elementos da simulação é feita por meio de controles do tipo Wii mote (Figura 14), desenvolvidos pela empresa Nintendo. Os Wii motes são controles sem fio com capacidade de force feedback limitados – eles podem emitir vibração em várias intensidades.

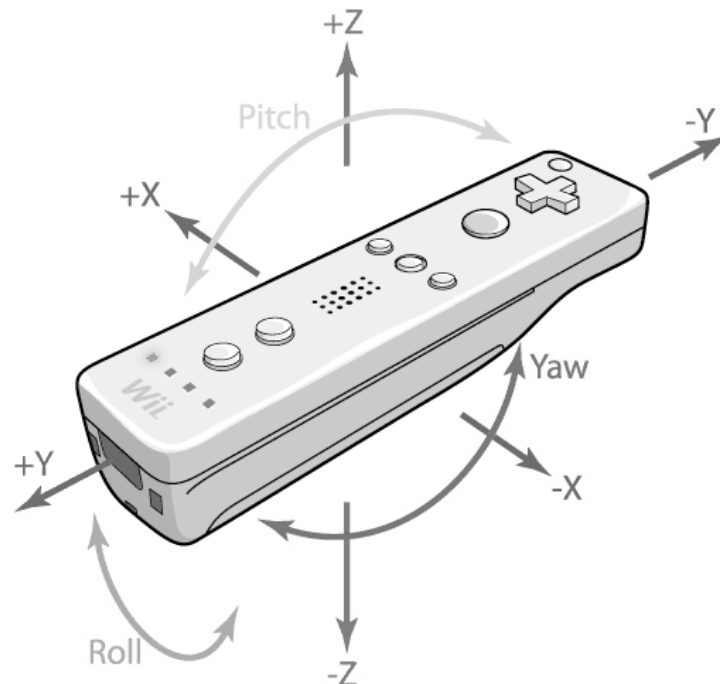


Figura 14 - Controle wii - ou “wiimote” e os graus de liberdade proporcionados por ele

Fonte: <http://futurelab.com.br/site/laboratorio-inovacao/tour-3D/>

O uso deste tipo de controle em detrimento dos habituais teclado e mouse visa simular de forma mais realista a manipulação de chaves de interruptores presentes no ambiente virtual.



Figura 15 -Treinando controlando o cenário virtual por meio do "wiimote"

Fonte: (ALVES E RIBEIRO JÚNIOR, 2011)

Das aplicações analisadas até agora, esta é a que mais se aproxima do modelo proposto. Até o momento da escrita deste texto, o ambiente STRiPP não contava com a visualização two-way.

3.3 Comparação entre os sistemas apresentados

Após análise das ferramentas descritas, foi compilada a Tabela 1, que leva em consideração os seguintes critérios:

- **Visualização 3D:** é necessário verificar se o ambiente/software proposto oferece a possibilidade de visualizar um modelo 3D do que está sendo simulado/projetado.
- **Two-way:** visualização simultânea de ambiente 3D e 2D. No caso da operação de subestações elétricas que se pretende simular, pretende-se que o operado visualize o painel de controle do sistema supervisor e qual será o impacto de determinada ação nos componentes da subestação.
- **Comunicação com SGDB:** a comunicação com banco de dados é extremamente necessária, para que as ações realizadas durante o treinamento possam ser armazenadas e posteriormente recuperadas para análise.
- **Versão para internet:** além de poderem ser usadas sem necessidade de deslocamento físico, aplicações com versões *web* possibilitam maior independência de sistema operacional, além de dispensarem a necessidade de instalação, o que facilita seu uso.

Tabela 1- estudo comparativo dos trabalhos relacionados

| Aplicação | Critérios de avaliação | | | |
|---------------------|------------------------|---------|----------------------|----------------------|
| | Visualização 3D | Two-way | Comunicação com SGDB | Versão para internet |
| ArqElectrical | ✓ | ✓ | ✗ | ✗ |
| MineInside® | ✓ | ✓ | ✓ | ✗ |
| RVCOPEL | ✓ | ✗ | ✓ | ✗ |
| Simulador Furnas 01 | ✗ | ✓ | ✓ | ✗ |
| Simulador Furnas 02 | ✓ | ✗ | ✓ | ✗ |

✓=sim ✗=não

Por meio da Tabela 1 é possível notar que nenhum dos sistemas apresentados atende totalmente às quatro características estudadas (visualização 3D, two-way, comunicação com SGDB e versão para internet), o que demonstra que a possível contribuição do Virtual Substation em relação aos mesmos seria apresentar um sistema que tivesse todas essas funcionalidades.

3.4 Conclusão

Nas seções anteriores foi realizada a análise de cinco ferramentas de terceiros na tentativa de se determinar quais as características que o Virtual Substation deve possuir para se constituir numa ferramenta de treinamento que apresente uma contribuição válida.

Os softwares analisados foram:

- *ArqElectrical* – aplicativo criado para o projeto correto de instalações elétricas residenciais;
- *RV-COPEL* – serious game criado para o ensino de procedimentos envolvendo linha viva;
- *MineInside* – interface para gerenciamento de uma mina;
- *Simulador Furnas 01* – ferramenta 2d criada para treinamento de operadores do sistema na operação de subestações e centros de operação;
- *Simulador Furnas 02* – software criado com o mesmo propósito do anterior, mas que apresenta interface 3d.



Ao final, compilou-se uma tabela comparativa com os quesitos que elas atendem ou não, levando-se em conta o contexto de Realidade Virtual e treinamento. São eles:

- possibilidade de visualização 3D;
- uso da estratégia *two-way* - visões simultâneas em 2D e 3D que se interagem entre si;
- Comunicação com SGDB para persistência e recuperação do que foi realizado no SVR;
- existência de uma versão para internet – possibilidade de usar navegadores de internet diversos para a execução da simulação, sem a necessidade de uma instalação local.

Pretende-se que a ferramenta criada possua todas estas características, ao contrário das ferramentas analisadas.



Capítulo 4 - Especificação do sistema proposto

4.1 Introdução

Após a revisão de literatura e levantamento de trabalhos relacionados, neste capítulo se inicia a real contribuição deste trabalho.

Primeiramente é descrita a análise dos requisitos da ferramenta que será criada. Esta análise é a primeira atividade técnica no desenvolvimento de qualquer *software*, sendo considerada a base do mesmo.

Segundo Paula Filho (2000) os requisitos são características que definem os critérios de aceitação de um produto.

Os requisitos são apresentados na forma de diagramas de caso de uso – uma notação bastante utilizada na indústria de *software*. Este tipo de diagrama descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.

Este levantamento de requisitos dará origem à especificação da arquitetura do sistema, que é detalhada na sequência. Na seção dedicada à arquitetura, detalham-se os três componentes principais do software (Área de login, área de treinamento e área do instrutor) e sua interação. Nesta seção, também é detalhado o padrão de projeto escolhido para a implementação do *software* (padrão MVC – Modelo-Visão-Controlador), bem como a forma como se dá o tráfego de informação entre as camadas deste modelo se dá na versão proposta pelos idealizadores deste projeto.

4.2 Levantamento de requisitos

Após entrevista junto ao *stakeholder* verificou-se que o Virtual Substation seria utilizado por duas classes de usuários:

- **Operador Trainee**
- **Instrutor**

O sistema proposto deverá permitir ao Operador Trainee a realização de uma sessão de treinamento e que cada passo dado por ele durante esta sessão possa ser armazenado para posterior recuperação por parte do instrutor.

Para o instrutor, deverá permitir a recuperação da sessão de treinamento passo a passo para verificar eventuais erros e acertos por parte de seu Trainee.

Além de cuidar do armazenamento e recuperação das sessões de treinamento, o sistema também precisará ser responsável pela segurança dos dados, impedindo o acesso indevido às informações por partes de terceiros, o que necessariamente implica na construção de um controle de *login*.

A Figura 16 mostra na forma de casos de uso, as funcionalidades disponibilizadas a cada um dos atores e suas eventuais dependências.

Para demonstrar a viabilidade do projeto, elegeu-se um dos componentes da subestação para ser controlado pelo Operador Trainee no protótipo. Por ser um dos componentes mais complexos e um dos mais importantes da subestação, optou-se pelo transformador de força.

Para criar uma simulação com maior realismo os componentes deste transformador, como tensão, corrente e outros serão alimentados por uma planilha chamada de curva de carga. Esta curva de carga demonstra o status dos componentes de um transformador real em uma determinada data coletado a cada dez minutos, como se ele houvesse sido obtido pelos sensores de um sistema SCADA real.

Além disso, será necessário que o sistema simule o efeito da alteração de um dos parâmetros do transformador – por exemplo, o comutador de tapes - terá no sistema como um todo, recalculando-os valores como tensão e corrente.

Assim como num sistema real, quando os valores de tensão se tornarem altos ou baixos demais o sistema também deverá ser capaz de alterar os valores de tapes, demonstrando ao Trainee de que forma isso acontece.

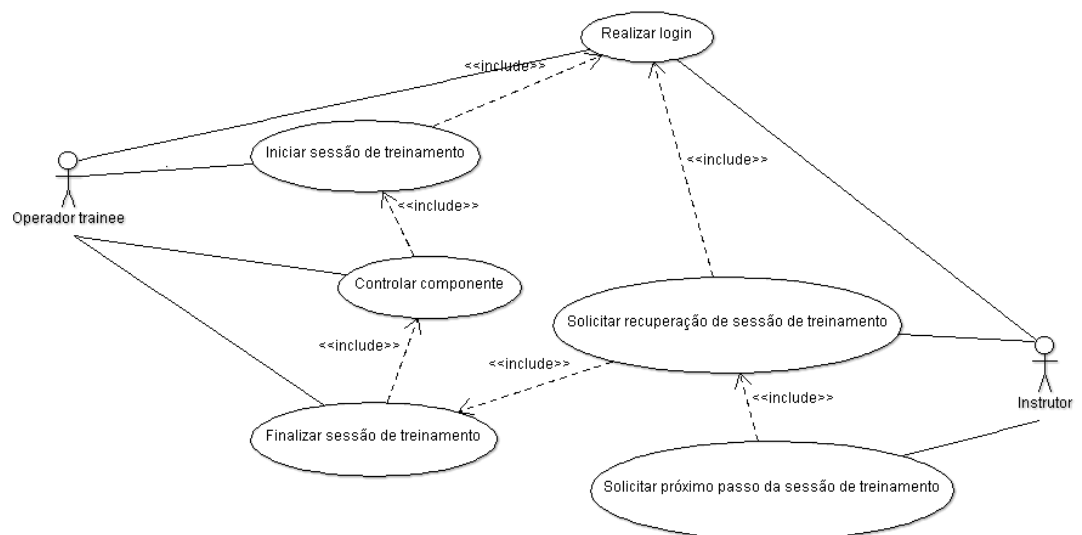


Figura 16 - Caso de uso do Virtual Substation



A seguir, são detalhados cada um dos casos de uso.

| | |
|-----------------------------|---|
| Nome do Caso de Uso | Realizar login |
| Ator Principal | Operador Trainee Instrutor |
| Atores Secundários | - |
| Resumo | Neste caso de uso são descritos os passos para realização de login no sistema |
| Pré-Condição | - |
| Pós-Condição | Só podem realizar treinamento operadores Trainees que realizaram login por este caso de uso. Só podem recuperar treinamento instrutores que tenham realizado login por este caso de uso. |
| Ações do Ator | Ações do Sistema |
| | 1- Exibir menu de opções |
| 2- Selecionar opção no menu | |
| 3- Digitar usuário e senha | |
| | 3- Se usuário e senha são válidos Se usuário é um operador Trainee - Direcionar para área de treinamento Senão - Se usuário é um instrutor - Direcionar para a área de recuperação de treinamento Fim-se Senão - Mostrar mensagem de erro Fim-se |
| Restrições / Validações | Não há |



| | |
|-----------------------------|--|
| Nome do Caso de Uso | Iniciar sessão de treinamento |
| Ator Principal | Operador Trainee |
| Atores Secundários | - |
| Resumo | Neste caso de uso é descrito o processo realizado pelo Operador Trainee para começar uma sessão de treinamento no sistema |
| Pré-Condição | - |
| Pós-Condição | - É gravado um “log” no sistema com a data e a hora em que Operador Trainee iniciou o treinamento - A curva de carga é recarregada periodicamente, simulando o comportamento de um transformador real |
| Ações do Ator | Ações do Sistema |
| 1- Selecionar opção no menu | |
| | 2- Recupera status inicial da curva de carga |
| | 3- Atualizar GUI para refletir a curva de carga |
| | 4- Gravar “log” de início de sessão |
| Restrições / Validações | Não há |

| | |
|---------------------|--|
| Nome do Caso de Uso | Alterar componente |
| Ator Principal | Operador Trainee |
| Atores Secundários | - |
| Resumo | Neste caso de uso é descrito o processo realizado pelo Operador Trainee interagir com um componente da GUI |
| Pré-Condição | - Sessão de treinamento deve ter sido previamente inicializada |
| Pós-Condição | - Este passo pode ser repetido várias vezes até que o Operador Trainee deseje encerrar |



| | |
|--|---|
| | <p>a sessão de treinamento</p> <ul style="list-style-type: none">- É gravado o status do sistema no SGDB- Também é gravado um “log” da ação no SGDB- São armazenados quais os valores alterados pelo usuário e qual é a diferença entre eles, para que, quando a curva de carga for recarregada, ela já possa refletir essa alteração |
| Ações do Ator | Ações do Sistema |
| 1- Selecionar componente na visão 3D ou digitar uma tecla na visualização 2D | |
| | 2- Calcular variáveis do transformador |
| | 3- Atualizar status do Sistema SCADA no SGDB |
| | 4- Gravar “log” da alteração |
| | 5- Atualizar visualização two-way |
| Restrições / Validações | Não há |

| | |
|------------------------------------|---|
| Nome do Caso de Uso | Finalizar sessão de treinamento |
| Ator Principal | Operador Trainee |
| Atores Secundários | - |
| Resumo | Neste caso de uso é descrito o processo realizado pelo Operador Trainee para encerrar a sessão de treinamento |
| Pré-Condição | - Sessão de treinamento deve ter sido previamente inicializada |
| Pós-Condição | - Também é gravado um “log” contendo a data e hora de encerramento da sessão de treinamento no SGDB |
| Ações do Ator | Ações do Sistema |
| 1- Selecionar comando de finalizar | |



| | |
|-------------------------|--|
| no menu | |
| | 2- Calcular variáveis de controle |
| | 3- Atualizar status do Sistema SCADA no SGDB |
| | 4- Gravar "log" da alteração |
| | 5- Fechar visualização two-way |
| Restrições / Validações | Não há |

| | |
|-------------------------|---|
| Nome do Caso de Uso | Solicitar recuperação de sessão de treinamento |
| Ator Principal | Instrutor |
| Atores Secundários | - |
| Resumo | Neste caso de uso é descrito o processo realizado pelo Instrutor para recuperar um treinamento efetuado previamente pelo Operador Trainee |
| Pré-Condição | - sessão de treinamento deve ter sido previamente finalizada |
| Pós-Condição | - Sistema entra na visualização two-way e fica aguardando pedidos de avanço por parte do Treinador |
| Ações do Ator | Ações do Sistema |
| | 1- Listar sessões de treinamento previamente gravadas |
| 2- Escolher sessão | |
| | 3- Recuperar status inicial do sistema no início da sessão de treinamento |
| | 4- Atualizar visualização GUI |
| Restrições / Validações | O sistema não permite que o Instrutor interaja com console ou mesmo com a visualização 3D, podendo apenas navegar por ela. A única opção de interação disponível é o |



| | |
|--|---|
| | botão AVANÇAR, cujo pressionamento dispara o caso de uso “Solicitar próximo passo da sessão de treinamento” |
|--|---|

| | |
|-------------------------|--|
| Nome do Caso de Uso | Solicitar próximo passo da sessão de treinamento |
| Ator Principal | Instrutor |
| Atores Secundários | - |
| Resumo | Neste caso de uso é descrito o processo realizado pelo Instrutor para recuperar um passo do aluno na sessão de treinamento |
| Pré-Condição | - instrutor deve ter solicitado recuperação prévia da sessão de treinamento |
| Pós-Condição | - Esta ação pode ser repetida várias vezes até chegar ao log final da sessão de treinamento - Sistema fica aguardando pedidos de avanço por parte do Treinador |
| Ações do Ator | Ações do Sistema |
| 1- Solicitar avanço | |
| | 2- Se for o último “log” da sessão Mostrar mensagem na tela Interromper o processo Fim-se |
| | 3- Recuperar próximo “log” da sessão de treinamento |
| | 4- Atualizar visualização 3D |
| Restrições / Validações | O sistema não permite que o Instrutor interaja com console ou mesmo com a visualização 3D, podendo apenas navegar por ela. A única opção de interação disponível é a tecla “ENTER”, cujo pressionamento dispara o caso de uso “Solicitar próximo passo da |



| | |
|--|------------------------|
| | sessão de treinamento” |
|--|------------------------|

4.3 Arquitetura do sistema

4.3.1 Modelo MVC

De modo a facilitar a programação e a manutenção do sistema Virtual Substation, foi adotado o padrão **MVC** (Model-View-Controller ou Modelo-Visão-Controlador).

O padrão MVC é um dos mais antigos padrões de projeto existentes no mercado. Já era utilizado na linguagem Smaltalk-80. Segundo Gamma (2000):

A abordagem MVC é composta por três tipos de objetos. O modelo é o objeto de aplicação, a visão é a apresentação na tela e o Controlador é o que define a maneira como a interface de usuário reage às entradas do mesmo. Antes da MVC os projetos de interface para usuário tendiam a agrupar estes objetos. A MVC separa estes objetos para aumentar a flexibilidade e a reutilização (Gamma; 2000):

No Virtual Substation, esta filosofia foi utilizada, com pequenas variações.

A camada **VISÃO** é responsável pela interface com o usuário. Nas modalidades de treinamento e recuperação de treinamento, ela também é responsável por exibir e atualizar modelos 3D e 2D. Maiores detalhes sobre a interface do sistema poderão ser obtidos no próximo capítulo.

A camada **Controlador** é responsável pela comunicação entre a camada de Visão e camada de Modelo. Em alguns casos ela realiza a validação das informações antes de enviá-las à camada de Modelo, mesmo que estas informações tenham sido previamente validadas na camada de visão, garantindo uma maior segurança. Na camada **Controlador** também é verificado se o usuário já realizou login ou não, pois se ele não tiver realizado login no sistema ou o tempo de sua sessão de treinamento tiver expirado, ele não deve ter acesso a nenhuma funcionalidade. Se isto acontecer, a camada **Controlador** deve enviar uma mensagem à camada de visão.

A camada **Modelo** compreende as classes de negócio. Ela também é responsável pelo armazenamento das informações em disco e recuperação das mesmas.

4.3.2 Módulos do sistema



Para atender os requisitos de usuário comentados na seção 4.2, o Virtual Substation foi dividido em três módulos:

- Gerenciador de Login
- Área de treinamento
- Área do instrutor

Abaixo, são discutidas as atribuições de cada um e a forma como ele implementa a filosofia MVC.

4.3.3 Gerenciador de login

O módulo **Gerenciador de login** é responsável pela segurança do sistema, permitindo que apenas usuários devidamente registrados tenham acesso às opções de Treinamento e Recuperação de Treinamento.

Ele também serve para direcionar o usuário até sua área específica. Assim que o Operador Trainee realiza login no sistema ele já é automaticamente direcionado à área de treinamento. O mesmo se dá com o Instrutor, que é direcionado para a área de Recuperação de treinamento.

Na Figura 17 é mostrado o diagrama MVC deste módulo

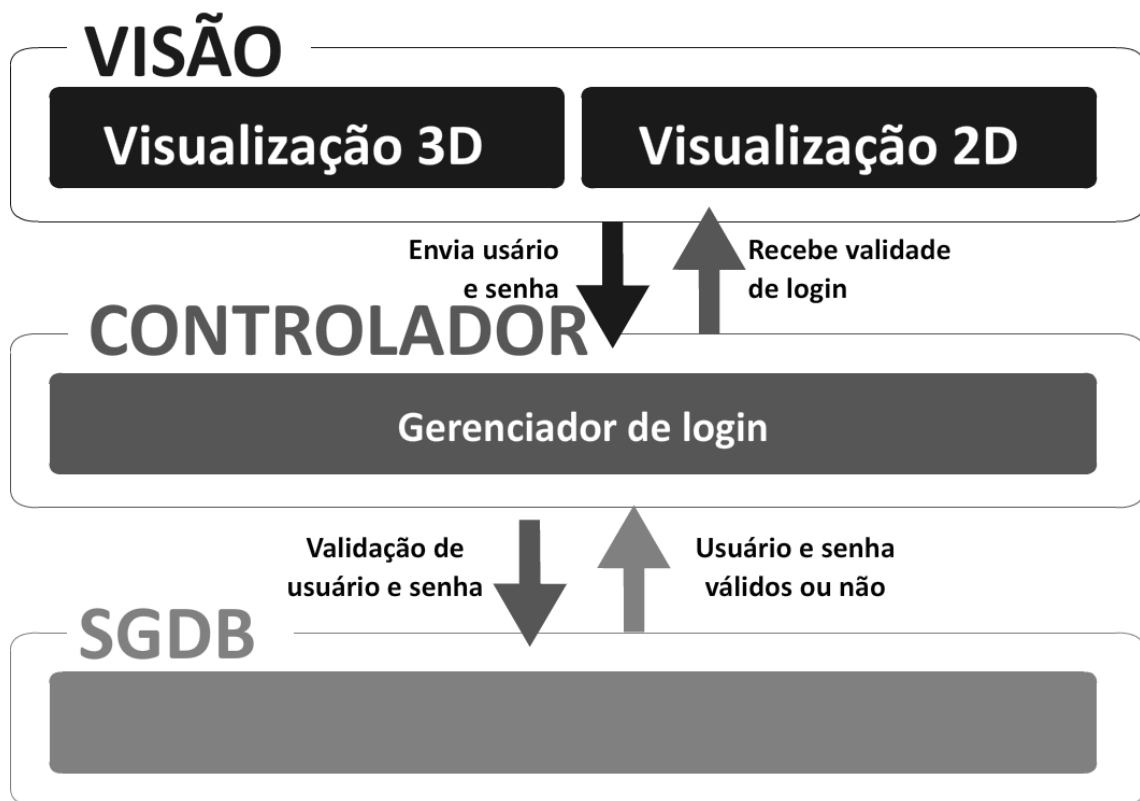


Figura 17 - Modelo MVC do gerenciador de login

A camada **Visão** exibe uma interface 2D onde podem ser digitados usuário e senha. Após o clique do botão “Fazer login” é realizada uma consulta à camada controlador. Se o login for válido, o usuário já é automaticamente redirecionado à área de treinamento ou a área de recuperação, de acordo com sua classe.

A camada **Controlador** deste módulo é composta por um único componente, o Gerenciador de Login, responsável por consultar o SGDB e verificar se o usuário e senha são válidos ou não. Caso não sejam, ele também é responsável por devolver uma mensagem à camada de visão.

4.3.3.1 Área de treinamento

A área de treinamento consiste numa representação 3D de uma subestação real, na qual o Operador Trainee pode navegar e com a qual pode interagir. Ela também possui em seu canto direito inferior, uma miniatura de um console SCADA semelhante ao que o operador teria em seu terminal dentro do centro de operações de uma subestação real. Também possui uma área textual no canto superior direito, onde dele pode verificar a condição de alguns indicadores. Cada um deles será detalhado no próximo capítulo.

Ao iniciar uma sessão de treinamento, a Área de treinamento recupera do banco de dados a posição da curva de carga em um determinado momento do tempo e a partir daí o Operador Trainee pode interagir com o modelo virtual.

O operador Trainee navega no modelo 3D utilizando teclado e mouse e interage com o terminal do sistema SCADA por meio do *mouse*. Ao solicitar um comando no terminal SCADA, automaticamente os indicadores são atualizados, e na vista 3D o componente que foi afetado começa a piscar.

Para cada ação realizada pelo Operador Trainee, é gravado no banco de dados uma linha com a data, a hora, o componente afetado e o valor do que foi afetado – o que chamamos de *Log*. Também é gravada uma nova linha com o status do sistema SCADA após a atualização realizada pelo Operador Trainee.

A Figura 18 mostra como as camadas MVC interagem na Área de Treinamento.

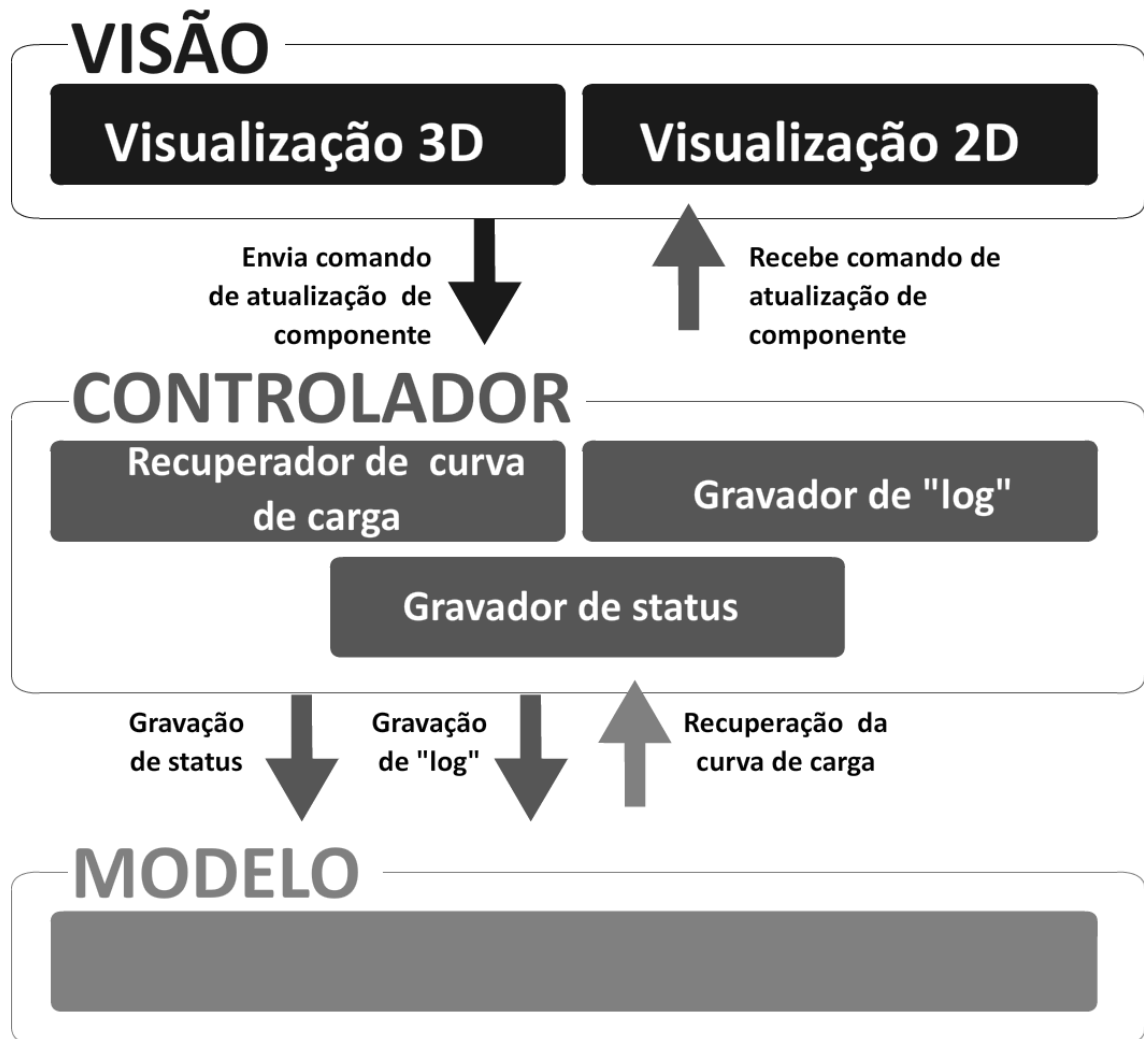


Figura 18 - Interação das camadas MVC na área de treinamento

Na camada de **visão** deverá haver uma interface dupla. Numa área maior da tela, será exibido o ambiente virtual com todos os elementos da subestação. Numa área menor, haverá uma interface gráfica 2d com todos os elementos do sistema supervisorio do transformador real. Por *default*, a visualização 2D deverá vir desligada, sendo acionada ao selecionar a Figura 3D do transformador.

As modificações realizadas no ambiente 2D interferem no sistema 3D. Por exemplo, ao alterar um indicador do painel 2D, o componente da subestação afetado pisca.

Há uma área no canto inferior esquerdo onde são mostradas mensagens de diálogo com o Trainee e logo acima dela, outra área onde se pode ver a situação da curva de carga.

A Figura 19 apresenta um esboço da tela do Virtual Substation.

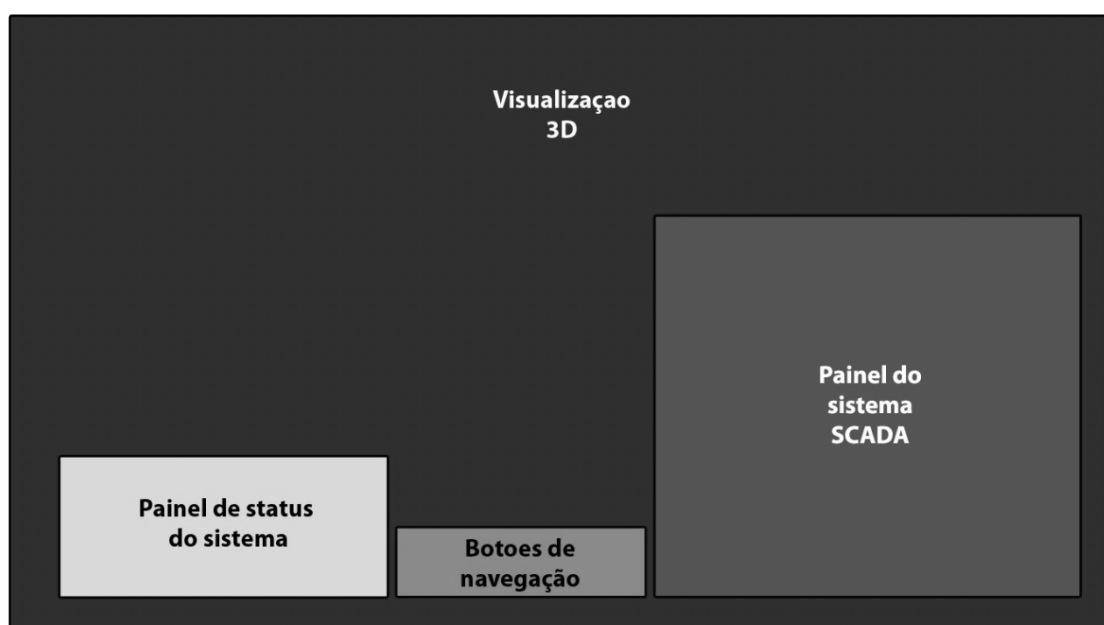


Figura 19 - esboço da tela do Virtual Substation

A interface também deverá possuir comandos para fazer a câmera saltar diretamente para o transformador de força, para uma chave seccionadora e para voltar ao ponto inicial da simulação. Também deve ter uma opção para o usuário terminar a sessão de treinamento.

A camada de **visão** também é responsável por recalcular as variáveis do sistema, emitir alertas quando o usuário realiza alguma ação ilegal e quando o sistema precisa atualizar algum valor automaticamente devido a alguma ação do Trainee.

Na camada controlador, há os seguintes elementos:

- **Recuperador de curva de carga:** responsável por recuperar um registro da curva de carga armazenado em SGDB. Este status é então recebido e processado pela camada de visão que automaticamente atualiza o painel do sistema SCADA.

- **Gravador de "log":** Responsável por armazenar as informações do "log" do sistema.
- **Gravador de status:** Responsável por armazenar a o status do sistema após modificação do usuário.

4.3.3.2 Área do instrutor

A área do instrutor é um local onde ele pode visualizar uma lista de sessões realizadas pelos seus Operadores Trainees.

Deverá ser mostrada uma lista de Trainees e suas respectivas sessões para que o instrutor possa fazer a seleção de que sessão de treinamento deseja recupera.

A tela de recuperação deverá ser semelhante à tela de treinamento, com a diferença de que o instrutor não poderá interagir com ela, podendo apenas recuperar uma a uma as ações realizadas pelo Trainee e verificar quais os componentes 2D e 3D foram afetados.

A Figura 20 ilustra de que forma as camadas MVC interagem nesta área.

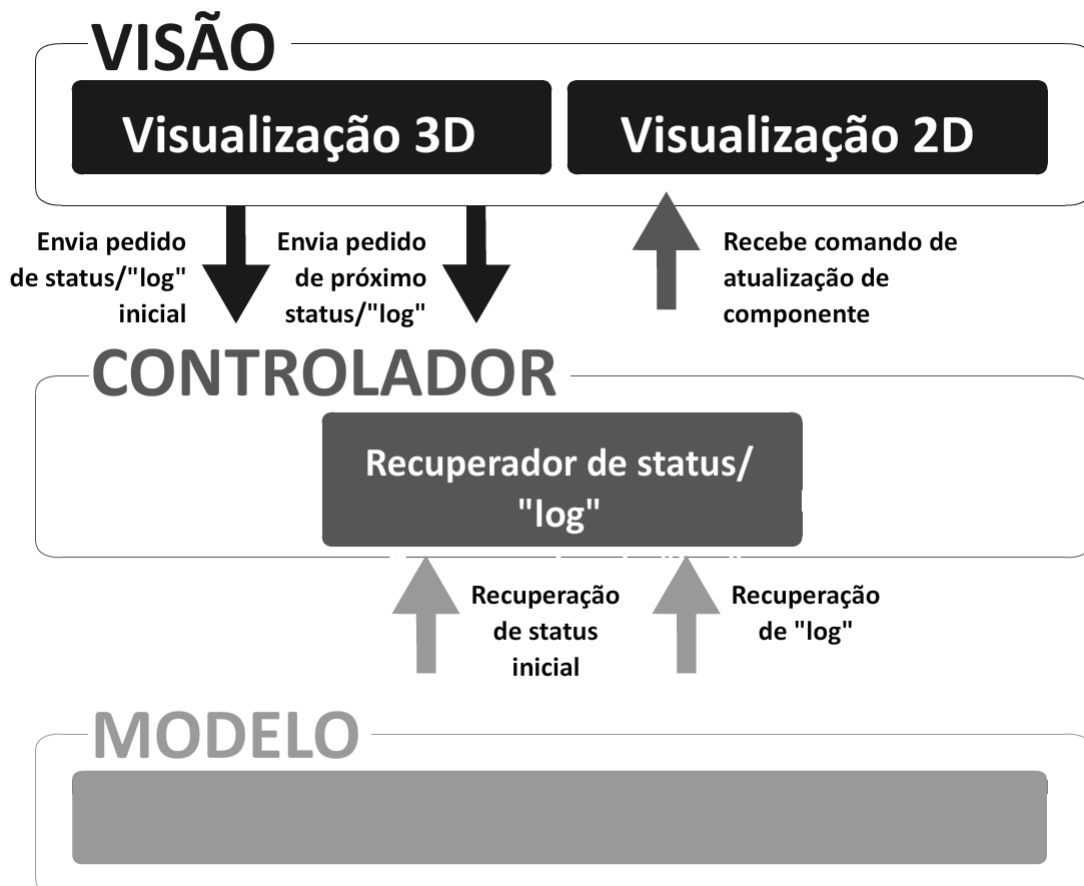


Figura 20 - Interação das camadas MVC do módulo de recuperação.



Conforme já mencionado, a camada **visão** recupera status inicial do treinamento e em seguida vai recuperando um a um todos os logs de treinamento gravado, atualizando as visualização 3D e 2D da GUI. Os elementos afetados tanto no painel 2D quanto 3D são destacados e é mostrado um texto na parte de baixo da tela, dizendo o que foi afetado.

A camada controlador é responsável por recuperar logs e status do SGDB e direcioná-los para a camada **Visão**.

4.4 Conclusão

Se fosse realizada uma analogia com um projeto arquitetônico, este capítulo constituiria a planta baixa do sistema. Nesta etapa, procurou-se determinar quem eram os atores do sistema e quais as ações que eles poderiam realizar ao interagir com os mesmos.

Após conversar com *stakeholders*, percebeu-se que o sistema deveria possuir duas áreas distintas:

- Área para treinamento – usada pelos operadores *trainees* ;
- Área do instrutor – usada pelos professores para recuperar o que foi realizado durante a interação do trainee com a área acima.

A seguir, procurou-se esboçar uma arquitetura do sistema antes de se partir para a implementação propriamente dita. Partindo do pressuposto que se deseja construir uma ferramenta com alta escalabilidade, a divisão do mesmo em três camadas, (denominadas Modelo, Visão e Controlador), mostra-se necessária, uma vez delas possui papel bastante distinto, o que facilita eventuais correções ou mesmo expansões das funcionalidades iniciais.



Capítulo 5 - Detalhes da implementação

5.1 Introdução

Após o levantamento de requisitos, todo desenvolvimento de *software* passa pela etapa da codificação – que é discutida neste capítulo.

O primeiro passo neste sentido é a escolha do ambiente de desenvolvimento para realizar a implementação do que se foi levantado. Levando-se em consideração o fato de que o objetivo principal deste trabalho é o de investigar técnicas que permitam a construção do simulador, esta é uma etapa que merece atenção especial, já que a escolha do ambiente de desenvolvimento certo impactará diretamente no tempo gasto e no na qualidade do produto final.

Existem inúmeros ambientes de desenvolvimento capazes de gerar ambientes interativos 3D com as características necessárias para a construção da camada de visão, de longe a mais complexa do projeto. Tal escolha não pode ser aleatória e deve levar em consideração alguns critérios, para que não se corra o perigo de escolher um ambiente incompatível com a solução que se pretende desenvolver. Para este estudo foram escolhidos os seguintes:

- Motor de física
- Linguagens de Programação Suportadas
- Importação de objetos criados em ferramentas externas
- Conexão com SGDB
- Geração de versão para internet
- Gratuidade

Cada um destes critérios é explicado em detalhes, no decorrer do capítulo e com base neles foram analisados os seguintes linguagens/ambientes de desenvolvimento: VRML, Papervision, Unity 3D e JavaFX. O resultado desta análise foi então compilado numa tabela comparativa.

Na sequência são discutidas as tecnologias usadas para construção das camadas de controlador e modelo, denominadas tecnologias de apoio.

Após isto, aspectos importantes como a modelagem geométrica e modelagem comportamental do Ambiente Virtual são apresentados.

Finalizando, o funcionamento das camadas MVC é apresentado e trechos de código relevante são elucidados.

5.2 Tecnologia para construção da camada de visão

Nesta seção são apresentadas algumas dos ambientes de desenvolvimento disponíveis no mercado para a construção de SRVs. Após a apresentação das tecnologias, é compilada uma tabela com todos os requisitos que a tecnologia deve atender e é escolhida uma delas para a construção da camada de visão da SE virtual.

5.2.1 VRML

Segundo Martins (2007), VRML é uma linguagem para modelagem em Realidade Virtual (*Virtual Reality Modeling Language*) que é usada para descrever objetos em 3 dimensões e combiná-los em cenários de ambientes virtuais. Foi uma linguagem pioneira para a construção de interfaces de Realidade Virtual.

Para Carrard (2012), a linguagem VRML não é reconhecida de forma padronizada pelos navegadores web. Desta forma, é necessário configurar o navegador para que ele compreenda o conteúdo do arquivo que está recebendo. Para isto é necessário um navegador VRML que deverá, tipicamente, ser adicionado ao navegador HTML na forma de *plug-in*. Plug-ins nada mais são do que programas que trabalham de forma associada ao navegador HTML, para que este entenda arquivos de tipos diferenciados. Dentre os plug-ins disponíveis no mercado há o Cosmo Player e a Corona 3D, que estão disponíveis apenas para plataforma Windows e Linux, o que inviabiliza a sua utilização em outros tipos de ambientes computacionais.

Outro aspecto que deve se considerar é que a modelagem de objetos deve ser feita em editores de texto ou em ambientes de desenvolvimento pagos.

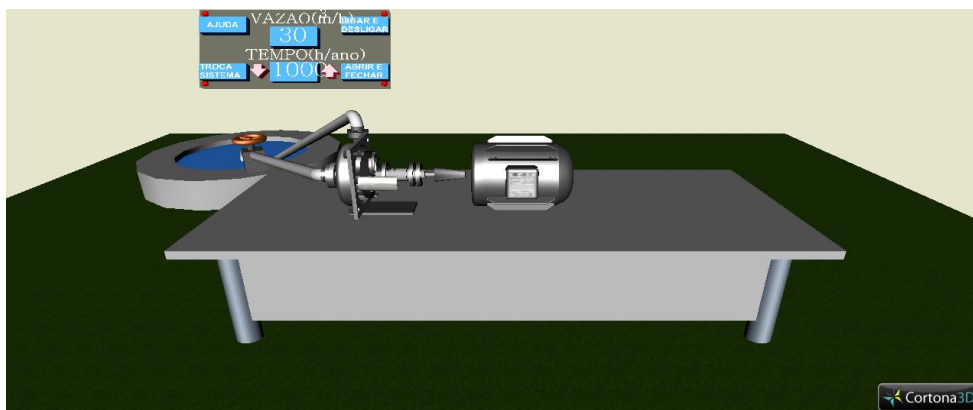


Figura 21 - Sistema desenvolvido em VRML -

Fonte <http://www.alexandre.eletrica.ufu.br/delaiba/principal.wrl>

A Figura 21 mostra um exemplo de ambiente virtual criado com a linguagem VRML.

5.2.2 Papervision

Para Winder E Tondeur (2009), Papervision é um motor *open source* que trouxe características 3D ao Flash Player. Papervision é uma biblioteca de uso simples, escrita em ActionScript 3.0 que permite aos desenvolvedores criar 3D em flash. Papervision permite que se construam modelos 3D em tempo real, provendo experiências 3D excitantes e interativas.

Devido a sua facilidade de uso e à grande popularidade do Flash player, Papervision tem sido bastante utilizado para construir ambientes virtuais.

Devido a restrições no número de polígonos, objetos complexos podem não ser importados e renderizados corretamente.

Para poder visualizar uma animação em Papervision, o usuário precisa ter o plug-in do Flash® instalado em seu computador, ou a animação precisa ser exportada para o formato .exe.



Figura 22 - Ambiente virtual simulando o fundo do mar criado em Papervision.

Fonte: <http://papervision3D.org/>

A Figura 22 mostra um exemplo de um cenário representando o fundo do mar construído com Papervision.

5.2.3 Unity 3D

Segundo Bento (2011) o software UNITY 3D simplifica o processo de criação de jogos 3D ou outras aplicações através de uma ferramenta de edição, fácil de usar, que permite inserir elementos pré-fabricados, como por exemplo, chuva ou terreno arborizado, e combiná-los com outros recursos de forma a criar o ambiente de um jogo. Para iniciantes e amadores na

programação de jogos, esta ferramenta torna-se bastante acessível permitindo desenvolver com facilidade aplicações que podem funcionar nos mais variados ambientes, on-line ou off-line, Mac, PC, iPhone e IPAD, TV e consoles de jogos.

Não é possível realizar modelagem geométrica no Unity 3D. O ambiente funciona mais como um agregador de conteúdos criados em outras plataformas, sendo capaz de importar nativamente arquivos mp3, jpeg, gif, mov, avi, wav, TTF, entre outros. Também é possível importar modelos 3D criados em programas de modelagem, tais como Blender, Maya, e3D Max e Cheetah.

Barros (2011) salienta que a partir do editor integrado, é possível criar todos os cenários e posicionar os objetos que vão povoar o mundo dentro da aplicação. Isto permite ter um *feedback* imediato e uma representação exata do que será visto em *runtime*.

Passos et al. (2009) aponta esta tecnologia como um motor de jogos (*game engine*) criado para auxiliar no desenvolvimento de jogos 3D. Mesmo usando a versão mais barata do Unity 3D, os jogos podem ser desenvolvidos para PC, Mac ou mesmo embutidos numa página web.

O motor usado pelo Unity é o PhysX (maiores informações sobre ele podem ser obtidas em <http://www.geforce.com/hardware/technology/physx>). Segundo Barros(2011) este engine consegue tornar o cenário bem mais realista sem grande esforço adicional. Além disso, o Unity inclui também um motor de áudio capaz de reproduzir sons em 3D e vários métodos para a criação de aplicações multiplayer.

Alguns recursos do Unity3D estão disponíveis apenas na versão paga.



Figura 23 - Exemplo de ambiente virtual desenvolvido em Unity 3D para o celular Iphone.

Fonte: <http://unity3D.com/support/resources/example-projects/>

Na Figura 23 vê-se uma demonstração de um ambiente virtual construído utilizando-se o Unity 3D.

5.2.4 Java FX

Clarke, Connors E Bruno (2010) definem JavaFX como:

“uma plataforma que inclui uma linguagem de scripts declarativa e de alto desempenho para oferecer e criar uma nova geração de aplicações para internet ricas. O foco do JavaFX é tornar o desenvolvimento de interface gráfica de usuário fácil e adotar características mais atraentes como efeitos visuais, som e animação.”

Eles acrescentam ainda que ela tem sido bastante usada para a criação de jogos pedagógicos. É necessário que o usuário tenha instalado uma máquina virtual Java superior à versão 6.0, para poder executar um programa criado nesta tecnologia.

A Figura 24 mostra um exemplo de utilização de JavaFX para a criação de um personagem virtual interativo.



Figura 24 - Jogo desenvolvido com JavaFX.

Fonte: <http://javafx.com/samples/FishSim/index.html>

5.2.5 Critérios para avaliação das ferramentas

Kirner E Siscoutto (2007) afirmam que a modelagem dos ambientes virtuais, usando linguagens como VRML (Virtual Reality Modeling Language) e sua sucessora, X3D, além de outras linguagens e ferramentas de autoria, permite, ao usuário, visualizar ambientes tridimensionais, movimentar-se dentro deles e manipular seus objetos virtuais. Os objetos virtuais podem ser animados, apresentando comportamentos autônomos ou ativados por eventos.



Todas as tecnologias analisadas possuem a capacidade de criar ambientes virtuais, nos quais o usuário é capaz de navegar em primeira pessoa como se estivesse em um cenário real. Todas as ferramentas analisadas permitem essa modelagem.

Para realizar esta avaliação, foram usados os seguintes critérios:

- Motor de física;
- Linguagem de programação;
- Importação de objetos criados em ferramentas externas;
- Conexão com SGDB;
- Geração de versão para internet
- Gratuidade;

Abaixo são detalhados cada um deles.

5.2.5.1 Motor de física:

O ambiente virtual deve ter um comportamento bastante semelhante à de uma subestação real. Isto significa que ela estará sujeita a forças como massa, gravidade, velocidade e fricção. Segundo Goldstone (2009) para desenvolvedores trabalhando com motores de jogo, os motores de física provêm um complemento para simular respostas realísticas para objetos em jogos.

5.2.5.2 Linguagem de programação

Tendo em vista que esta não é uma simulação estática, toda a parte de interação com o operador envolverá programação. Para possibilitar um maior ganho de tempo no desenvolvimento é essencial que o ambiente possa dar suporte a uma ou mais linguagens de programação conhecidas pela equipe de desenvolvimento, além de possibilitar maior compatibilidade e reaproveitamento de eventuais códigos-fonte escritos por terceiros.

5.2.5.3 Importação de objetos criados em ferramentas externas

É fato que modelos demoram muito tempo para serem construídos de forma como muitas vezes é feito em VRML. Visando simplificar, padronizar e até mesmo distribuir melhor as tarefas de modelagem física entre várias pessoas, é de suma importância que o ambiente virtual seja capaz de importar arquivos gerados em ferramentas 2d como Photoshop ou em ferramentas 3D populares como 3D Max, Blender ou Maya.



5.2.5.4 Conexão com SGDB

Um dos objetivos da simulação é que o treino realizado pelo operador possa ser armazenado, e posteriormente recuperado. Por exemplo, pelo supervisor do operador ou pelo próprio operador para que se possa verificar onde ele, eventualmente, possa ter errado ou acertado. Dito isso, é de suma importância que esses dados possam ser persistidos em um SGDB. Hoje em dia, a grande maioria dos programas comunica-se com os usuários por meio de interfaces gráficas de janelas. Entretanto, normalmente, os programas não contém todo código referente à exibição dos dados nelas. De fato, utilizam gerenciadores de interface de usuário – conjuntos de rotinas que incluem as funcionalidades que um programador vai necessitar frequentemente, ao construir este tipo de funcionalidade. Da mesma forma, para comunicar-se com processos remotos, usam gerenciadores de comunicação. Segundo Heuser (2009) para manter grandes repositórios compartilhados de dados, ou seja, para manter bancos de dados, são usados Sistemas de Gerência de Banco de Dados (SGBD).

5.2.5.5 Geração de versão para internet

Em princípio pretende-se que a simulação possa ser executada em ambientes desktop (com SGDB e aplicação, ambos instalados na máquina do usuário) e web (com a aplicação executando embutida em uma página HTML e SGDB instalado num servidor remoto). Mas considerando-se que existe uma grande variedade de dispositivos com poder computacional variado, como aparelhos celulares, *smartphones*, *tablets* e mesmo consoles para jogos, é importante que o produto final da simulação não esteja restrito a uma única plataforma, possibilitando que no futuro ela possa ser portada para outros ambientes.

5.2.5.6 Gratuidade

O custo de um software é um fator determinante na execução de um projeto, pois ele pode impactar em custos muito altos que podem vir a inviabilizar o projeto. Felizmente, a maioria das tecnologias analisadas possui licença *open source* e pode ser utilizada de forma gratuita – além de disporem de editores para elas, também gratuitos. A única exceção é o Unity 3D, que possui uma versão gratuita e totalmente funcional, mas que tem várias versões pagas com recursos adicionais. (veja nas tabelas 2,3 e 4 as diferenças entre as versões do Unity 3D).

5.2.6 Ferramenta escolhida para a construção do ambiente virtual

Conforme se percebe pela Tabela 2, apesar de possuir uma versão gratuita com menos recursos, o motor Unity 3D se mostra mais adequado ao projeto proposto.

A biblioteca Papervision se mostra uma alternativa interessante, pois possui uma integração nativa com páginas web que poderiam servir de camada intermediária de conexão com o SGDB. Apesar de não possuir um motor de física nativo, ela pode ser expandida por meio da utilização da biblioteca APE - ActionScript Physics Engine (disponível em <http://www.cove.org/ape/>). Apesar disso, foi descartada por apresentar um problema sério ao importar modelos geométricos complexos gerados em editores 3D externos. Isto pode comprometer a visualização da simulação, bem como seu tempo de resposta. Papervision é perfeito par demonstrações e interfaces simples. Agora é potencialmente possível criar interfaces 3D impactantes, interativas, com um mínimo tempo de carga. Isto é algo que os desenvolvedores de internet e os clientes desejavam por um longo tempo. Ainda tem grandes limitações, porque a aceleração não é feita por hardware. Porém está em desenvolvendo uma nova versão do Papervision3D, baseada na API do Flash Player 10, com a possibilidade de reutilizar linhas de código C e C++ e a execução de PixelBender; e assim se espera o aumento significativo da qualidade dos novos sites 3D na internet. Sobre este caminho, nasce uma nova gama de interfaces, e Papervision3D se torna muito importante no desenvolvimento de websites. Osa (2011).

Tabela 2 - Comparativo entre as tecnologias analisadas.

| Ferramenta | Critérios de avaliação | | | | | |
|-------------|------------------------|--------------------------------------|---|------------------|---------------------------------|----------|
| | Motor de física | Linguagens de Programação Suportadas | Importação de objetos criados em ferramentas externas | Conexão com SGDB | Geração de versão para internet | Gratuita |
| VRML | × | JAVASCRIPT | × | × | × | ✓ |
| PAPERVISION | × | ACTION SCRIPT | ✓ | ✓ | ✓ | ✓ |
| UNITY 3D | ✓ | JAVASCRIPT, C#, BOOL | ✓ | ✓ | ✓ | ✓ |
| JAVA FX | × | JAVA | × | ✓ | ✓ | ✓ |



✓=sim ✗=não

A linguagem JavaFX também apresenta a possibilidade de conexão com SGDB nativa, devido ao fato de rodar sobre uma máquina virtual Java, podendo desfrutar de recursos como JDBC (Java Database Connectivity). Embora Java seja considerada uma tecnologia com maiores recursos e processamento disponíveis do que a tecnologia Flash, esta tecnologia possui problemas de incompatibilidade entre versões. Além disso, a linguagem JavaFX conta ainda com poucos materiais para estudo e pouca penetração no mercado, o que torna difícil averiguar o potencial de tal tecnologia.

Posto isso, o Unity 3D se mostra mais adequado, por possuir um motor de física nativo bastante maduro, além de possibilitar a conexão com banco por meio de comunicação com uma página web escrita em PHP ou JAVA por meio de sua função nativa WWWForm, viabilizando a arquitetura proposta para o sistema.

Um possível entrave para o uso do unity3D seria que alguns recursos só estão disponíveis nas versões pagas. As tabelas 7 e 8 mostram quais recursos estão disponíveis em cada versão (3D, 2011).

Todas as ferramentas necessárias (motor de física, conectividade com web e possibilidade de conexão com o SGDB) estão presentes na função gratuita, o que não inviabiliza o uso do engine para a construção da simulação.

Tabela 3 - Diferenças gerais entre as versões do Unity 3D.

| General | Unity Pro | Unity | iOS Pro | iOS | Android Pro | Android |
|---|-----------|-------|-------------------|-----|----------------|---------|
| ☐ Physics | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ☐ Audio (3D Positional and Classic Stereo) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ☐ Audio Filters | ✓ | | ✓ | | ✓ | |
| ☐ Video Playback and Streaming | ✓ | | ✓ ^{2, 3} | | ✓ ³ | |
| ☐ Multiplayer Networking with RakNet | ✓ | ✓ | ✓ | | ✓ | |
| ☐ Fully Fledged Streaming with Asset Bundles | ✓ | | ✓ | | ✓ | |
| ☐ May be licensed and used by companies or incorporated entities that had a turnover in excess of US\$100,000 in their last fiscal year | ✓ | | ✓ | | ✓ | |

Tabela 4 - Diferenças entre as versões do Unity no que diz respeito ao desenvolvimento.

| Deployment | Unity Pro | Unity | iOS Pro | iOS | Android Pro | Android |
|---------------------------|-----------|-------|---------|-----|-------------|---------|
| ☐ One-Click Deployment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ☐ Web Browser Integration | ✓ | ✓ | | | | |
| ☐ Custom Splash Screen | ✓ | | ✓ | | ✓ | |
| ☐ Build Size Stripping | | | ✓ | | ✓ | |

5.3 Tecnologias de apoio

Além da ferramenta para construir o SRV, ainda seria necessário definir uma ferramenta para a construção da controladora e da camada modelo, responsável pela persistência dos dados da aplicação proposta. Nas seções seguintes são apresentadas a Linguagem PHP e o SGBD PostgreSQL.

5.3.1 Linguagem PHP

PHP significa Hypertext Preprocessor. Segundo Welling e Thomson (2005) PHP é uma linguagem de criação de scripts no lado servidor que foi projetada especificamente para a Web. Dentro de uma página Web você pode embutir código PHP que será executado toda vez que a página for visitada. O código PHP é interpretado pelo servidor web e gera HTML ou outra saída que o visitante geral.

Portanto, um dos requisitos para o funcionamento do interpretador PHP é a presença de um servidor web. No caso do Virtual Substation, optou-se por usar o servidor Apache, por se tratar também de uma plataforma gratuita e de código-fonte aberto.

Segundo Infowester (2012) ao acessar qualquer site, há um servidor por trás daquele endereço responsável por disponibilizar as páginas e todos os demais recursos que você pode acessar. Assim, quando você envia um e-mail através de um formulário, coloca uma mensagem em um fórum de discussão, faz uma compra on-line, etc., um servidor Web (ou um conjunto de servidores) é responsável por processar todas essas informações.

O autor ainda acrescenta que como servidor Web, o Apache é o mais conhecido e usado. Os motivos incluem sua excelente performance, segurança, compatibilidade com diversas plataformas e todos os seus recursos.

O servidor Apache é capaz de executar código em PHP, Perl, Shell Script e até em ASP e pode atuar como servidor FTP, HTTP, entre outros. Sua utilização mais conhecida é a que combina o Apache com a linguagem PHP e o banco de dados MySQL, mas ele pode ser perfeitamente usado com outros SGBDs como o PostgreSQL, descrito na próxima seção.

5.3.2 PostgreSQL

Segundo Biazus (2012) o PostgreSQL é um SGBD objeto-relacional de código aberto, com mais de 15 anos de desenvolvimento. É extremamente robusto e confiável, além de ser extremamente flexível e rico em recursos. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos,

como herança e tipos personalizados. A equipe de desenvolvimento do PostgreSQL sempre teve uma grande preocupação em manter a compatibilidade com os padrões SQL92/SQL99.

5.4 O modelo MVC implementado

Conforme mencionado no capítulo anterior, para a criação do Virtual Substation, optou-se por utilizar o padrão de projeto MVC. Abaixo, na Figura 25 é mostrada a estrutura de diretórios do Virtual Substation.

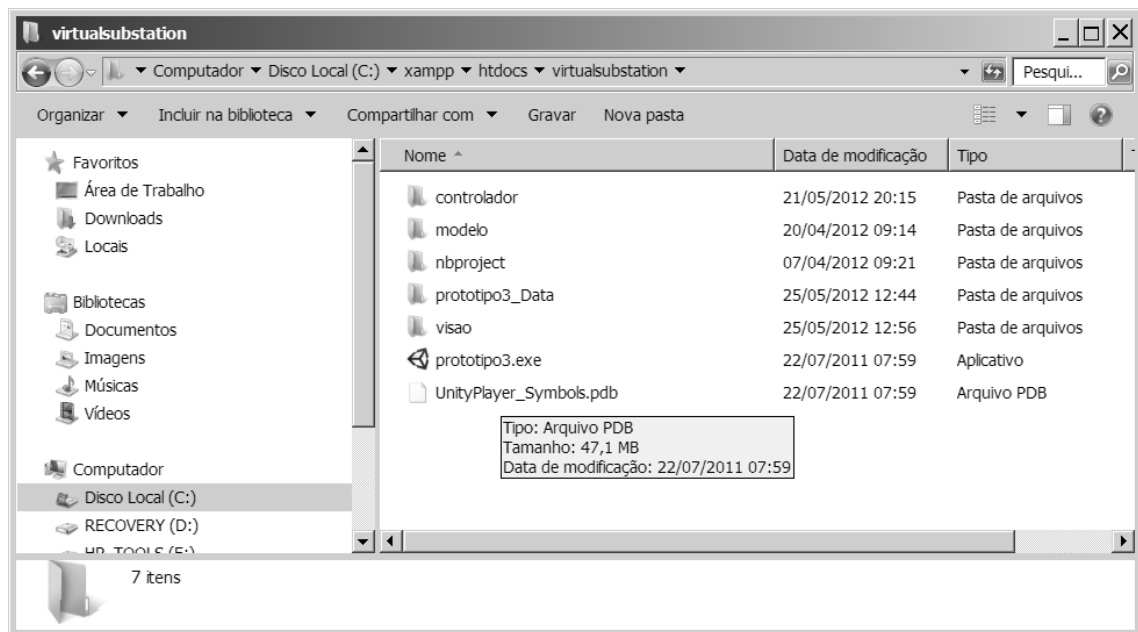


Figura 25 - Pasta do Virtual Substation no gerenciador de arquivos

Como pode ser observado, existem as pastas, **visão**, **modelo** e **controlador**. A pasta `nbproject` foi criada pelo editor Netbeans, usado para codificar os scripts PHP. Também existe programa executável gerado para o Windows e um arquivo `.pdb`, gerado pelo exportador do Unity 3D.

A seguir é detalhado o conteúdo de cada pasta e as tecnologias usadas para construir cada camada.

5.4.1 Camada visão

A Tabela 5 detalha o conteúdo da pasta **visão**:

A camada visão foi construída usando-se a ferramenta Unity. Ela pode ser usada na forma de um executável Windows *Standalone* ou por meio de uma arquivo `.unity3d` embutido em



uma página web. Para ser executada no navegador, é necessário que o Unity Player esteja instalado. O *download* de seu instalador pode ser feito em <http://unity3D.com/webplayer/>.

Tabela 5 - Conteúdo da pasta visão

| Arquivo | Função |
|-------------------|--|
| crossdomain.xml | Arquivo de políticas de segurança que permite ao Unity Player acessar informações de um determinado domínio na web |
| index.html | Página HTML responsável por chamar a simulação gerada no Unity |
| WebPlayer.unity3D | Arquivo da simulação gerada pelo Unity 3D |

Também é possível criar um pacote *apk* para ser instalado no sistema *Android* e um programa executável para a plataforma *Mac* da Apple. Mas no presente teste, essas duas opções não foram usadas.

A criação da camada de visão envolveu as seguintes etapas

- Modelagem geométrica
- Modelagem comportamental

5.4.1.1 Modelagem geométrica

A modelagem geométrica abrange a descrição da forma dos objetos virtuais através de polígonos, triângulos ou vértices, e sua aparência, usando textura, reflexão da superfície, cores, etc.

A modelagem geométrica da SE foi realizada com o uso do software 3d Max da Autodesk. Esta modelagem foi realizada com o uso de fotos obtidas durante uma visita de campo à subestação de Juiz de Fora como referência. Normalmente, num projeto de SRV esta é a etapa que consome mais tempo. Por isso é importante contar com o uso de um ambiente consagrado escala, como o 3d Max.

Na Figura 26 são mostrados detalhes da modelagem criada dentro deste aplicativo:

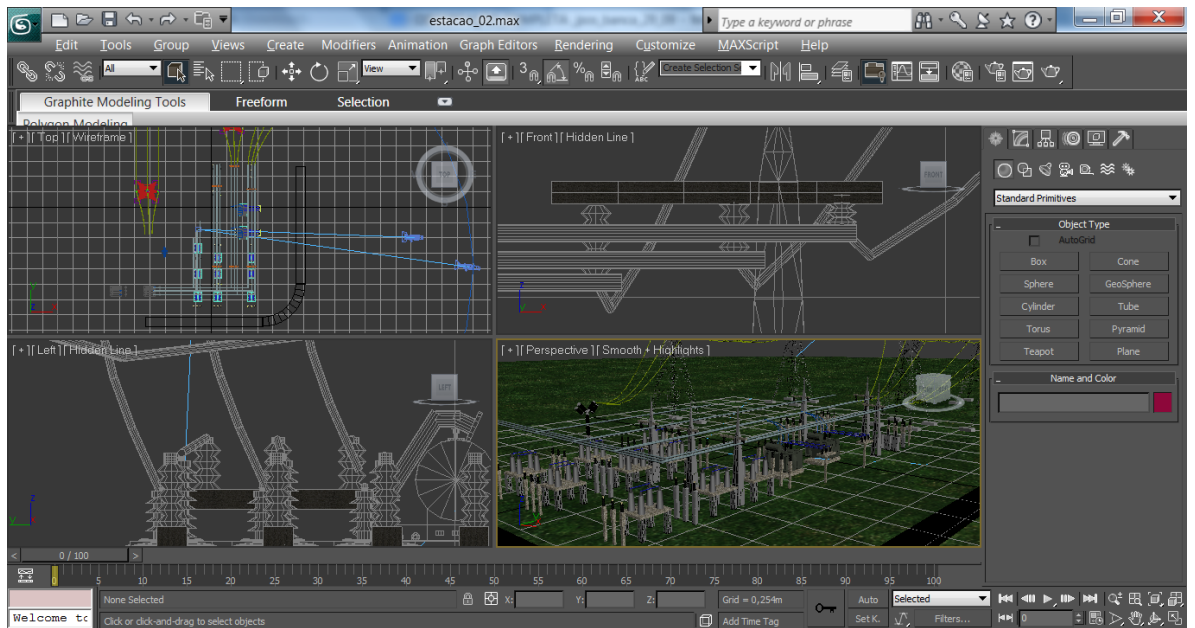


Figura 26 - Modelagem geométrica utilizando-se 3d Max

Nativamente o Unity 3D suporta a importação direta dos arquivos .max gerados pelo 3D max, mas no caso desta modelagem, este processo mostrou-se demorado, visto que alguns dos componentes modelados apresentam geometrias bastante complexas, o que tornou o modelo bastante pesado. Por isso, optou-se por converter o modelo para o formato .obj. Trata-se de um tipo de arquivo mais leve, mas que preserva as principais características do modelo geométrico. Este arquivo .obj foi importado para dentro do ambiente do Unity 3D sem problemas. Na Figura 27 pode-se ver a modelagem já dentro do projeto Unity.

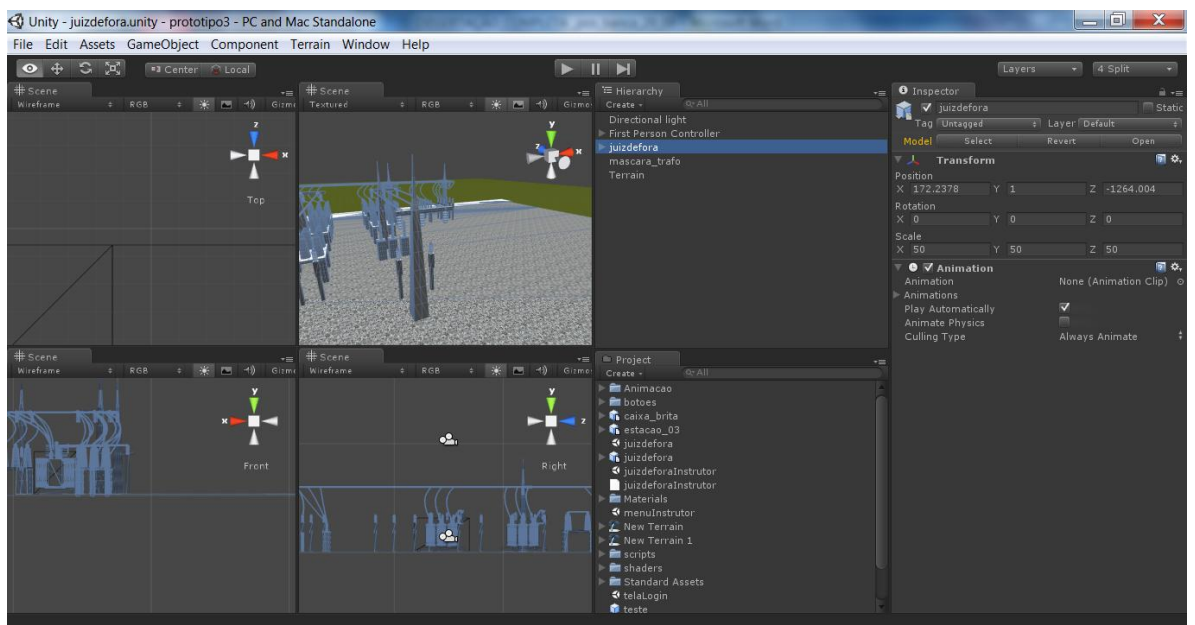


Figura 27 - Modelagem importada dentro do ambiente Unity 3d

5.4.1.2 Modelagem comportamental

Este tipo de modelagem determina como se dará a experiência do usuário com o ambiente virtual. Neste ponto normalmente são definidas as formas de navegação e interação com os objetos virtuais. Deve-se determinar, por exemplo, se o usuário navegará pelo ambiente através de uma câmera em primeira pessoa ou se será representado pela figura de um avatar, se ele pode atravessar algum componente ou não e o que acontece se ele vier a colidir com tal componente.

Neste aspecto, a escolha do Unity mostra-se acertada, já que ele traz as duas forma de navegação citadas anteriormente em seus *Assets* nativos *First Person Controller* e *3rd Person Controller*.

Para o Virtual Substation optou-se pelo uso do *First Person Controller*. Este componente provê um cilindro com uma câmera no alto que reage a um script predefinido, mas que pode ser customizado. Através do cilindro, é possível a detecção de colisões. Na Figura 28 é mostrado este objeto:

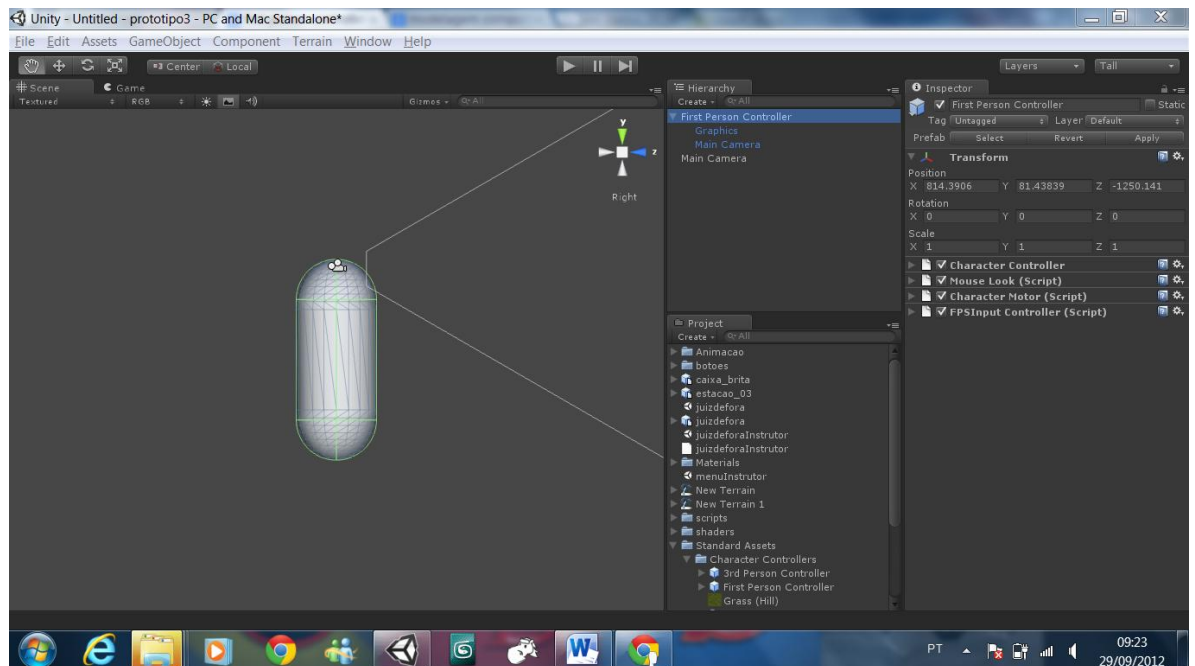


Figura 28 - objeto First Person Controller do Unity 3d

Para caminhar pelo SRV, o usuário usa as teclas A,S,D e W ou as setas direcionais do teclado.

Como seria interessante trocar o ponto de vista do observador, podendo em alguns momentos visualizar a SE virtual de cima, foi acrescentada uma segunda câmera à já existente no objeto *First Person Controller*. Esta câmera inicialmente ficaria inativa e o intercâmbio entre as



duas se daria através do pressionamento da tecla ENTER. O script que realiza esta ação pode ser visto na Figura 29.

```
1. //variáveis criadas para o efeito de alternar camera
2. var multicom : Camera[];
3. var contador:int = 0;
4. var controle:int = 0;
5. (...)
6.
7. function Update () {
8. (...)
9. for (cameras in multicom)
10.    {
11.
12.        if (cameras !=multicom [controle])
13.        {
14.
15.            cameras.enabled = false;
16.            cameras.GetComponent(AudioListener).enabled = false;
17.        }
18.
19.        if (Input.GetKeyDown ("return")){
20.            //print(controle);
21.            if (controle==1) {
22.                controle=0;
23.            }
24.            else {
25.                controle=1;
26.            }
27.
28.            multicom[controle].enabled = true;
29.            multicom[controle].GetComponent(AudioListener).enabled = true;
30.        }
31. (...)
32. }
```

Figura 29 - script criado para alternar entre as câmeras

Outro detalhe importante é que a câmera *default* do First Person Controller está configurada para girar a perspectiva obedecendo ao movimento do mouse, o que pode tornar o uso dos botões um tanto quanto inquietante. Neste caso, esta rotação também foi desligada via script, como mostra a Figura 30.

```
1. //variavel para controlar rotação
2. static var rotacao:boolean = false;
3. (...)
4. function Update () {
5. (...)
6.
7.     //liga/desliga a rotação da perspectiva
8.     GameObject.Find("First Person
   Controller").GetComponent("MouseLook").enabled = rotacao;
9.     GameObject.Find("Main Camera").GetComponent("MouseLook").enabled =
   rotacao;
10. (...)
11. }
```

Figura 30 - script criado para o desligamento da rotação de câmera

Caso deseje ligar novamente a rotação, o usuário poderá fazê-lo através de um botão específico da GUI.

Para evitar que o usuário atravessasse objetos que no mundo real sabe-se ser sólido, como, por exemplo, o transformador de força, estes objetos foi transformados em um corpos rígidos (Rigidy Bodies) e foram aplicadas Malhas de colisão (Mesh Colliders) nos mesmos.

5.4.1.3 Curva de carga

Cotrin (2009) define **Curva de carga** com a curva que apresenta demanda em função do tempo $D(t)$ para todo período T . No caso do virtual Substation, esta curva é representada por uma planilha com os valores de corrente nas fases A,B e C e tensão nas mesmas fases obtidos de um transformador real durante 24 horas de seu funcionamento. Tais dados foram colhidos a cada 10 minutos deste tempo. Ela é usada para dar maior realismo à simulação.

Na Figura 31 é mostrada esta curva de carga na forma de planilha.



| 1 | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|----|------|-------------------|-------------------|-------------------|---|-----------------|-----------------|-----------------|---|---|------|-------|---|---|
| 2 | Hora | Corrente - Fase A | Corrente - Fase B | Corrente - Fase C | | Tensão - Fase A | Tensão - Fase B | Tensão - Fase C | | | | | | |
| 3 | | Amperes | Amperes | Amperes | | kV | kV | kV | | | tape | kV | | |
| 3 | 0:00 | 486,54 | 508,65 | 508,65 | | 67,72 | 67,05 | 67,38 | | | +16 | 95,67 | | |
| 4 | 0:10 | 464,42 | 497,60 | 486,54 | | 70,95 | 70,24 | 70,59 | | | +15 | 94,67 | | |
| 5 | 0:20 | 453,37 | 475,48 | 475,48 | | 72,68 | 71,95 | 72,32 | | | +14 | 93,67 | | |
| 6 | 0:30 | 442,31 | 464,42 | 464,42 | | 74,50 | 73,75 | 74,12 | | | +13 | 92,67 | | |
| 7 | 0:40 | 442,31 | 464,42 | 464,42 | | 74,50 | 73,75 | 74,12 | | | +12 | 91,67 | | |
| 8 | 0:50 | 431,25 | 464,42 | 453,37 | | 76,41 | 75,64 | 76,02 | | | +11 | 90,67 | | |
| 9 | 1:00 | 431,25 | 453,37 | 453,37 | | 76,41 | 75,64 | 76,02 | | | +10 | 89,67 | | |
| 10 | 1:10 | 431,25 | 453,37 | 453,37 | | 76,41 | 75,64 | 76,02 | | | +9 | 88,67 | | |
| 11 | 1:20 | 431,25 | 442,31 | 442,31 | | 78,42 | 77,63 | 78,02 | | | +8 | 87,67 | | |
| 12 | 1:30 | 420,19 | 442,31 | 442,31 | | 78,42 | 77,63 | 78,02 | | | +7 | 86,67 | | |
| 13 | 1:40 | 420,19 | 442,31 | 431,25 | | 78,20 | 77,42 | 77,81 | | | +6 | 85,67 | | |
| 14 | 1:50 | 409,13 | 431,25 | 431,25 | | 77,75 | 76,97 | 77,36 | | | +5 | 84,67 | | |
| 15 | 2:00 | 409,13 | 431,25 | 431,25 | | 78,33 | 77,55 | 77,94 | | | +4 | 83,67 | | |
| 16 | 2:10 | 409,13 | 431,25 | 431,25 | | 79,03 | 78,24 | 78,64 | | | +3 | 82,67 | | |
| 17 | 2:20 | 409,13 | 431,25 | 420,19 | | 79,10 | 78,31 | 78,70 | | | +2 | 81,67 | | |
| 18 | 2:30 | 409,13 | 431,25 | 420,19 | | 79,23 | 78,44 | 78,83 | | | +1 | 80,67 | | |
| 19 | 2:40 | 409,13 | 431,25 | 431,25 | | 79,29 | 78,50 | 78,90 | | | 0 | 79,67 | | |
| 20 | 2:50 | 409,13 | 431,25 | 420,19 | | 79,73 | 78,93 | 79,33 | | | -1 | 78,67 | | |
| 21 | 3:00 | 409,13 | 431,25 | 420,19 | | 80,54 | 79,73 | 80,13 | | | -2 | 77,67 | | |
| 22 | 3:10 | 409,13 | 431,25 | 420,19 | | 80,25 | 79,45 | 79,85 | | | -3 | 76,67 | | |
| 23 | 3:20 | 409,13 | 431,25 | 420,19 | | 80,44 | 79,63 | 80,03 | | | -4 | 75,67 | | |
| 24 | 3:30 | 409,13 | 431,25 | 420,19 | | 80,27 | 79,47 | 79,87 | | | -5 | 74,67 | | |
| 25 | 3:40 | 409,13 | 431,25 | 420,19 | | 80,54 | 79,73 | 80,13 | | | -6 | 73,67 | | |
| 26 | 3:50 | 409,13 | 420,19 | 420,19 | | 79,29 | 78,50 | 78,90 | | | -7 | 72,67 | | |
| 27 | 4:00 | 398,08 | 420,19 | 420,19 | | 79,73 | 78,93 | 79,33 | | | -8 | 71,67 | | |
| 28 | 4:10 | 398,08 | 420,19 | 420,19 | | | | | | | | | | |

Figura 31 - curva de carga no formato de planilha

Esta planilha, foi convertida em na tabela `vs_curva_carga` do SGDB que é recarregada periodicamente pela GUI do Virtual Substation. Na Figura 32 vemos esta tabela no gerenciador de BD pgAdmin.

| id | datahora | correnteFase | correnteFase | correnteFase | tensaoFase | tensaoFase | tensaoFase |
|-------------|--------------|--------------|--------------|--------------|------------|------------|------------|
| PK1 integer | timestamp | real | real | real | real | real | real |
| 1 | 2011-12-30 0 | 67.72 | 67.38 | 67.38 | 486.54 | 508.65 | 508.65 |
| 2 | 3 | 70.95 | 70.59 | 70.59 | 464.42 | 497.6 | 486.54 |
| 3 | 4 | 72.68 | 72.32 | 72.32 | 453.37 | 475.48 | 475.48 |
| 4 | 5 | 74.5 | 74.12 | 74.12 | 442.31 | 464.42 | 464.42 |
| 5 | 6 | 74.5 | 74.12 | 74.12 | 442.31 | 464.42 | 464.42 |
| 6 | 7 | 76.41 | 76.02 | 76.02 | 431.25 | 464.42 | 453.37 |
| 7 | 8 | 76.41 | 76.02 | 76.02 | 431.25 | 453.37 | 453.37 |
| 8 | 9 | 76.41 | 76.02 | 76.02 | 431.25 | 453.37 | 453.37 |
| 9 | 10 | 76.41 | 76.02 | 76.02 | 431.25 | 453.37 | 442.31 |
| 10 | 11 | 78.42 | 78.02 | 78.02 | 420.19 | 442.31 | 442.31 |
| 11 | 12 | 78.42 | 78.02 | 78.02 | 420.19 | 442.31 | 431.25 |
| 12 | 13 | 78.2 | 77.81 | 77.81 | 409.13 | 431.25 | 431.25 |
| 13 | 14 | 77.75 | 77.36 | 77.36 | 409.13 | 431.25 | 431.25 |
| 14 | 15 | 78.33 | 77.94 | 77.94 | 409.13 | 431.25 | 431.25 |
| 15 | 16 | 79.03 | 78.64 | 78.64 | 409.13 | 431.25 | 420.19 |
| 16 | 17 | 79.1 | 78.7 | 78.7 | 409.13 | 431.25 | 420.19 |
| 17 | 18 | 79.23 | 78.83 | 78.83 | 409.13 | 431.25 | 431.25 |
| 18 | 19 | 79.29 | 78.9 | 78.9 | 409.13 | 431.25 | 420.19 |
| 19 | 20 | 79.73 | 79.33 | 79.33 | 409.13 | 431.25 | 420.19 |
| 20 | 21 | 80.54 | 80.13 | 80.13 | 409.13 | 431.25 | 420.19 |

Figura 32 - tabela `vc_curva_carga` criada com base na planilha de curva de carga

5.4.1.4 Detalhes da GUI

Abaixo, a Figura 33 mostra uma tela do Virtual Substation sendo executada na plataforma Windows.

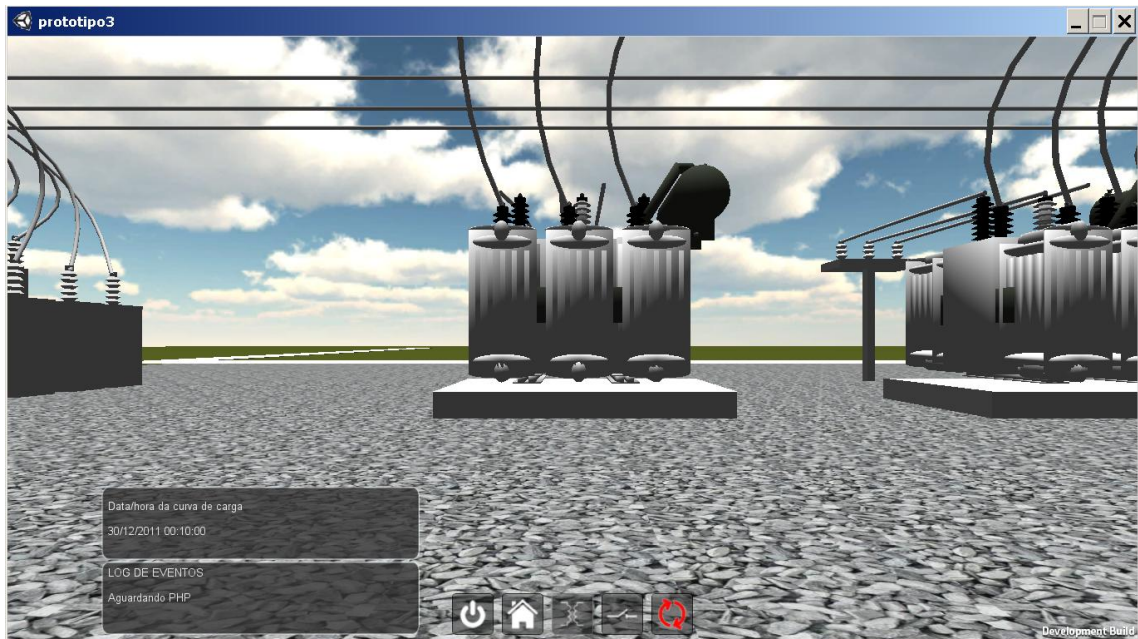


Figura 33 - Virtual Substation em execução no ambiente Windows

Por outro lado, a Figura 34 apresenta o simulador em versão web:

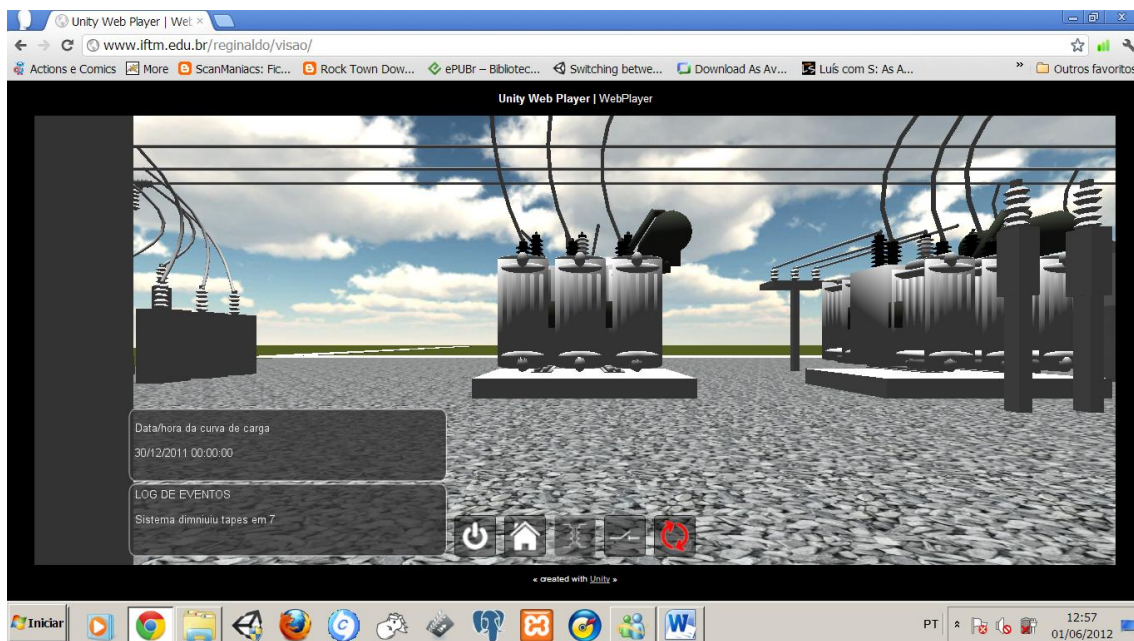


Figura 34 - virtual Substation em execução no navegador

É possível perceber que, pelo fato de o simulador ao ser executado no navegador, não há nenhum tipo de perda aparente e as duas interfaces são renderizadas de forma bem semelhante.

O executável Windows, bem como a versão web, farão uso da camada de modelo e da camada de controladores, ambas implementadas na linguagem PHP. Isso significa que para acessar o aplicativo é necessário o usuário estar conectado à internet.

Conforme já foi dito também, a camada visão é dividida em três módulos:

- Gerenciador de Login,
- Área do Operador Trainee,
- Área do Instrutor.

Nas próximas seções, estas áreas são descritas com maiores detalhes.

5.4.1.5 Tela de Login

A tela de login é composta de uma caixa de diálogo com campos para usuário e senha e dois botões. Logo abaixo, em amarelo, são mostradas as imagens do sistema. Uma imagem desta tela é mostrada na Figura 35.

Ao clicar em **Fazer Login**, é chamado o controlador Gerenciar Login, que testa o usuário e a senha e devolve uma mensagem ao sistema. Se o usuário logado for um trainee, o sistema é redirecionado para a área de treinamento. Se for um instrutor, o usuário é redirecionado para sua área específica.

Ao clicar no botão **Sair**, o sistema é fechado.



Figura 35 - tela de Login



Na Figura 36, é mostrado um trecho do código Javascript que é executado pelo Unity quando o botão **Fazer Login** é pressionado. Logo após, ele é comentado com maiores detalhes.

```
12. function Login() {
13.     (...)
14.     validações de campo omitidas
15.     (...)
16.     var form = new WWWForm();
17.     form.AddField("nomeUsuario", campoNomeUsuario);
18. form.AddField("senha", Utils.Md5Sum( campoSenha));
19. var paginaWeb = new WWW('http://localhost/virtualsubstation/
20.     controlador/GerenciadorDeLogin.php', form);
21. textoStatus = "Conectando com Servidor";
22. yield paginaWeb;
23.
24. textoStatus = "Dados recebidos com sucesso!";
25. if (paginaWeb.error !=null) {
26. textoStatus = "Erro ao enviar dados! "+paginaWeb.error;
27. }
28. else {
29. var texto = paginaWeb.text;
30. var pedacos = texto.Split("|"[0]);
31. if (pedacos.Length !=2) {
32. textoStatus = "Página web retornou no formato errado.(1)";
33. textoStatus = paginaWeb.text;
34. }
35. else {
36. if (pedacos[0]=='erro') {
37.     textoStatus = "PHP retornou o erro \n"+pedacos[1];
38. }
39. else {
40. if (pedacos[0]=="ok"){
41. var dados = pedacos[1].Split(";"[0]);
42. if (dados.Length==1) {
43.     textoStatus = dados[0];
44.     }
45.     else {
46.     if (dados[3]=='Trainee') {
47.     textoStatus = "Login bem-sucedido....";
48.     (...)
49.     Application.LoadLevel(1);
50.
```



```
51.         }
52.         else if (dados[3]=='Instrutor') {
53.             textoStatus = "Login bem-sucedido...";
54.             (...)
55.             Application.LoadLevel(2);
56. }
57. }
58. }
59. }
60. }
61. }
62.     (...)
63. }
```

Figura 36 - Código Javascript responsável por fazer Login de usuários

Utilizando-se a classe *WWWForm* – nativa do Unity 3D(usada na linha 5 da Figura 36) são passados os campos *nomeUsuario* e *senha* para o arquivo PHP *GerenciadorDeLogin.php* que responde se eles existem ou não. Esta resposta é tratada (linha 14 da Figura 36) e o resultado é mostrado na tela.

Se tudo estiver correto e for possível determinar a classe do usuário logado, o sistema é redirecionado para o nível de treinamento ou o nível de instrutor por meio do comando `Application.LoadLevel(numeroDoNivel)`; (presente nas linhas 38 e 44 da Figura 36).

Maiores detalhes sobre o formato da resposta do controlador são descritos na Seção que trata dos controladores.

5.4.1.6 Área de Treinamento

A seguir, é apresentada a tela da área de treinamento e seus componentes:

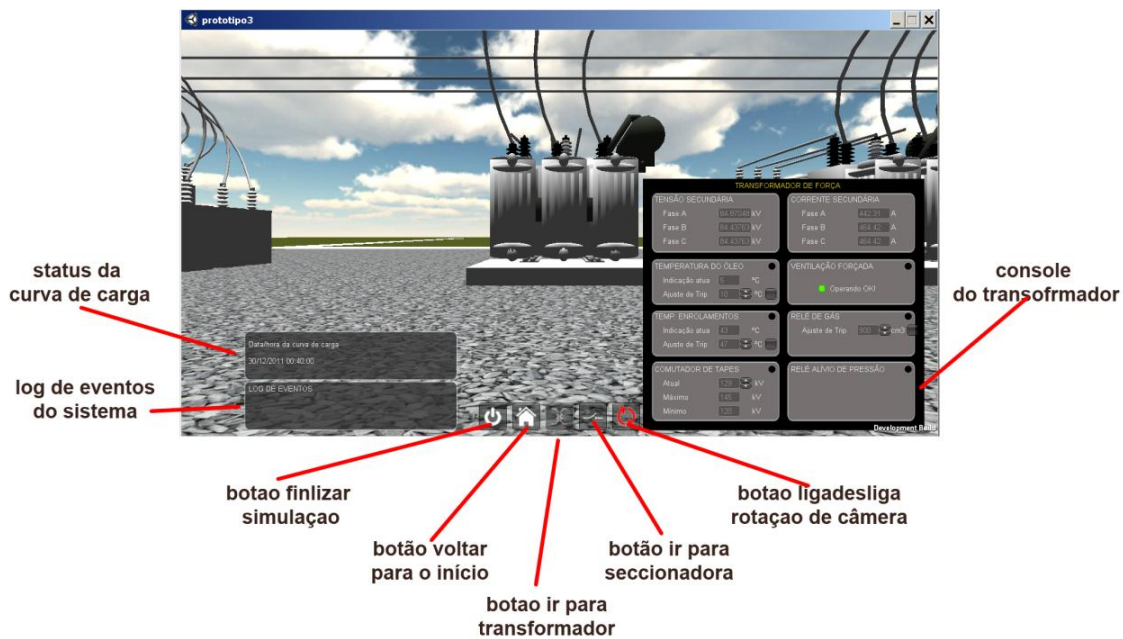


Figura 37 - área do Operador Trainee

Do lado esquerdo inferior da tela, existem dois painéis: status da curva de carga e log de eventos do sistema.

No painel **status da curva de carga** são mostradas as informações da curva de carga, tais como a data e hora da curva que está ativa no sistema. Se houver algum erro no carregamento da curva de carga, também será mostrado ali.

No painel **log de eventos** do sistema são mostradas mensagens do sistema, tais como, eventuais ações ilegais cometidas pelo usuário. Quando o sistema grava alguma informação de log ou de status o progresso é mostrado neste painel. O painel **log de eventos** ainda serve para mostrar algum possível erro ocorrido durante a comunicação com o controlador.

Ao clicar no **botão finalizar simulação**, o usuário encerra o treinamento e volta para tela de login, onde poderá começar outro treinamento ou sair do sistema.

Os botões **voltar para o início**, **ir para o transformador** e **ir para seccionadora** possibilitam uma navegação rápida, guiando o usuário, automaticamente, até um ponto específico do AV ao serem acionados.

Por *default*, o ponto de vista da câmera é sempre frontal e usuário navega apenas com teclado. Neste caso, o *mouse* é usado somente para interagir com os componentes 3D do AV. Mas pode ser interessante alterar o ponto de vista câmera, como é comum em jogos de primeira pessoa, para, por exemplo, visualizar melhor algum detalhe. Quem controla se câmera está fixa ou não é o botão **ligar/desligar rotação da câmera**.



O **console do transformador** é aberto somente quando o usuário clica sobre a imagem do transformador de força no AV. Se houver um novo clique, este painel é fechado.

Na console são mostradas todas as informações de operações possíveis em um transformador. Entretanto, na maioria das vezes, um sistema SCADA não permitira controlar tantas opções.

Por meio da manipulação dos diversos campos deste painel é possível alterar suas propriedades. Quando uma propriedade do transformador é alterada, a imagem dele no AV piscará, demonstrando qual dos componentes 3D foi afetado pela ação do usuário.

Um destes campos é o comutador de taps.

Comutador de Tapes sob Carga (OLTC, do inglês *On-Load Tap Changer*): Dispositivo que altera a relação de transformação em um número pré-definido de passos e desta forma altera também a tensão em um determinado barramento.

Ao alterar o comutador de tapes, as tensões secundárias das fases A, B e C são, automaticamente, recalculadas na tentativa de simular mais fielmente um transformador real. Para o cálculo das novas tensões é usada a seguinte fórmula:

$$tensão\ fase = \left(\frac{valor\ tap\ antigo}{valor\ tap\ novo} \right) \times tensão\ fase \quad (5.1)$$

É importante lembrar que a curva de carga é recarregada automaticamente a cada 10 minutos. Se por acaso, o usuário aumentou ou diminuiu as tensões, esta diferença é aumentada ou diminuída na próxima carga dos valores da curva, tornando o sistema mais interativo.

Quando há alteração de tensão, seja por parte do usuário ou quando o sistema carrega uma nova tupla da curva de carga, é verificado se as novas tensões não estão muito altas ou baixas em relação ao valor de tensão nominal (valor eficaz de tensão pelo qual o sistema é projetado, expresso em volts ou quilovolts.) que é

$$tensao\ nominal = \frac{138}{\sqrt{3}} \quad (5.2)$$

Neste caso, o sistema incrementa ou decrementa automaticamente o valor do comutador de tapes até que os valores de tensão cheguem o mais próximo possível do desejável.

Quando há a intervenção do sistema, é gravado um *log* dizendo que isso aconteceu, para que o instrutor possa estar ciente deste fato, quando for recuperar a sessão de treinamento.

Caso o usuário, ao alterar o valor do comutador de tapes, tente inserir um valor que ultrapasse os níveis máximos e mínimos deste componente, o sistema interromperá a alteração



e mostrará uma mensagem no painel **log de eventos do sistema**. Paralelamente, começa a piscar um alarme, próximo ao comutador de tapes, mostrado na Figura 38.

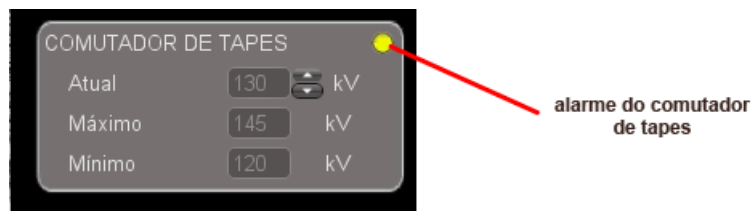


Figura 38 - alarme mostrando que os limites do comutador de tapes foram infringidos

5.4.1.7 Área do instrutor

Ao entrar na área do instrutor, é mostrado um menu com todos os Trainees que realizaram sessões, conforme mostra a Figura 39.

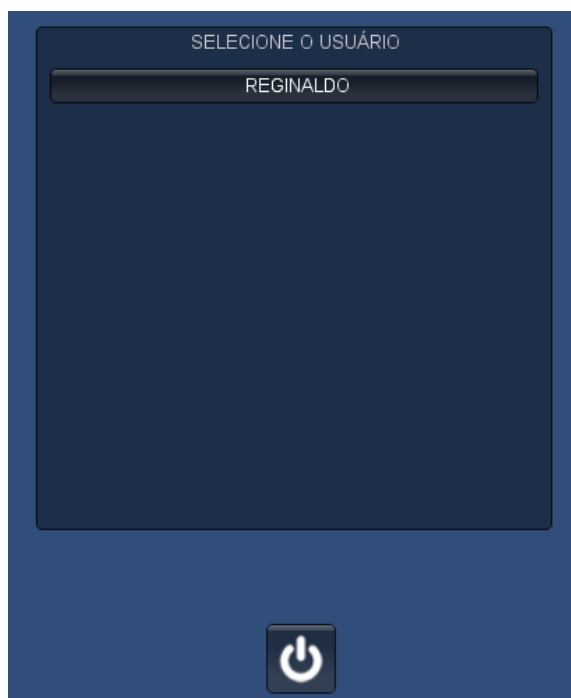


Figura 39 - Menu de Trainees na área do instrutor.

Após selecionar um trainee na lista, é mostrada uma lista com todas as sessões realizadas por ele, com data e hora, conforme se vê na Figura 40.



Figura 40 - Lista de sessões realizadas pelo Trainee

Ao escolher uma sessão na lista, o instrutor é então, levado para a área de recuperação da sessão, mostrada na Figura 41.

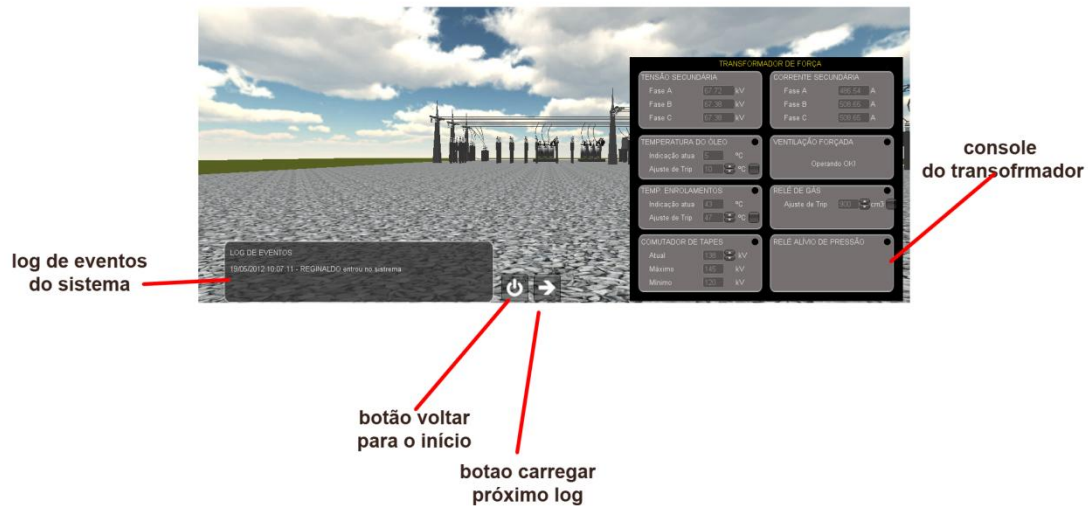


Figura 41 - Área de recuperação da sessão

Esta área é análoga á área de treinamento, com algumas diferenças. Nela, o instrutor pode navegar pela simulação, mas não interagir com ela.

No painel **log de eventos do sistema** são carregadas, uma a uma as ações realizadas pelo trainee durante a realização de seu treinamento. O sistema carrega a primeira ação e fica aguardando o instrutor solicitar a próxima.

Nesta tela, o botão **voltar para o início** também serve para deixar a área de recuperação e voltar para o menu de seleção da sessão. Já o botão **carregar próximo log** serve para buscar a próxima ação realizada pelo Trainee. O painel **log de eventos** quanto o **console do transformador** são, automaticamente, atualizados, de acordo com o que foi buscado.

Tanto as alterações realizadas pelo trainee quanto as realizadas pelo sistema (por exemplo, calibragem automática do comutador de tapes) são mostradas ao instrutor, indicando quem foi o autor da mudança.

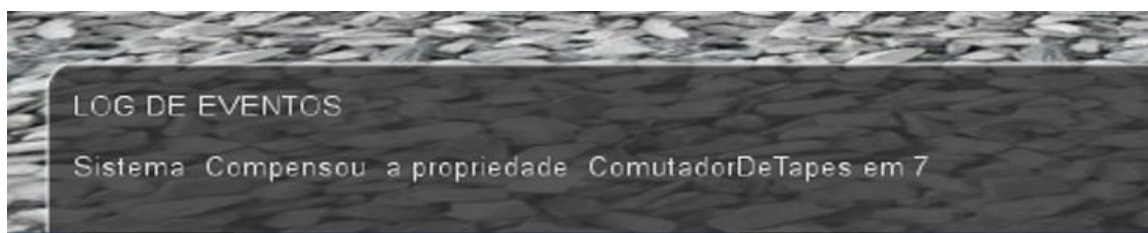
É importante salientar que são carregadas no log de eventos, apenas as interações realizadas pelo trainee com o painel do transformador de força e eventuais compensações realizadas pelo sistema, uma vez que informações relativas à navegação dele pelo SVR e desligamento ou não da rotação da câmera não irrelevantes para o instrutor.

Na Figura 42 é mostrado o detalhe do log de eventos com uma ação realizada pelo trainee sendo recuperada pelo instrutor.



Figura 42 - detalhe do log de eventos mostrando ação realizada pelo trainee

Na Figura 42 é mostrado o detalhe do log de eventos com uma ação realizada pelo sistema sendo recuperada pelo instrutor.



Quando o instrutor chegar ao último log da sessão, é mostrada uma mensagem dizendo que a sessão chegou ao fim.

Ao contrário do que acontece na área de treinamento, o **console do transformador** fica aberto o tempo todo e nenhum dos seus componentes pode ser alterado pelo instrutor.

5.4.2 Camada controlador

A camada controlador é responsável por receber dados para persistir no SGDB e também por recuperar dados. Ela deve também validar as informações recebidas e fazer o controle de sessão, para impedir que usuários que não estejam logados tenham acesso indevido ao sistema.

A Tabela 6 - conteúdo da pasta controlador mostra os arquivos que compõem a pasta controlador e explica a função de cada um.

Tabela 6 - conteúdo da pasta controlador

| Arquivo | Função |
|------------------------------------|--|
| EncerradorSessao.php | Script que encerra a sessão do usuário logado |
| GerenciadorDeLogin.php | Script responsável por verificar se usuários e senhas são válidos, antes de logar no Virtual Substation |
| GravadorDeLog.php | Responsável por gravar todos os “logs” do operador Trainee durante sua sessão de treinamento |
| GravadorDeStatus.php | Responsável por gravar o status do transformador de força durante um dado instante da simulação |
| recuperadorCurvaCarga.php | Responsável por recuperar um registro da curva de carga num determinado instante de tempo. Se não receber um número de id, ele traz o primeiro registro da curva de carga. Em caso contrário, ele recupera o próximo registro da curva de carga após aquele id |
| RecuperadorDeSessoes.php | Recupera as sessões realizadas por um determinado usuário na área do orientador |
| RecuperadorDeUsuarios.php | Recupera a lista de usuários que realizaram sessões de treinamento no sistema para a área do orientador |
| RecuperadorPrimeiroLogDaSessao.php | Recupera o primeiro log e o primeiro Status de uma sessão de treinamento |
| RecuperadorProximoLogDaSessao.php | Recupera o próximo log e o próximo status do transformador armazenados em uma sessão de treinamento |

Conforme se percebe pela extensão dos arquivos, esta camada foi implementada usando-se a linguagem PHP, descrita na seção 5.3.1.

Na Figura 43 tem-se o exemplo do código-fonte do controlador Gerenciador de Login um controlador escrito em PHP.

```
1. <?php
```



```
2. error_reporting(0);
3. //usar utf8_encode quando houver acentuação para não dar problema
   com o Unity
4. $nomeUsuario = $_POST['nomeUsuario'];
5. $senha = $_POST['senha'];
6. try {
7.   if (!isset($nomeUsuario)) {
8.     throw new Exception("Nome de usuário vazio!");
9.   }
10.
11.  if (empty($senha)) {
12.    throw new Exception("Senha vazia!");
13.  }
14.
15.  include_once '../modelo/VsUsuario.php';
16.  $resultado = VsUsuario::testarLogin($nomeUsuario, $senha);
17.  if ($resultado == null) {
18.    throw new Exception("Usuário não encontrado");
19.  } else {
20.    //criando a sessão de treinamento
21.    $agora = time();
22.    include '../modelo/VsSessao.php';
23.    $sessao = new VsSessao();
24.    $sessao->setVs_usuario_fk_id($resultado->getId());
25.    $sessao->setDtInicio($agora);
26.    $sessao->persiste();
27.
28.    echo "ok|" . $resultado->getId() . "|";
29.    $resultado->getNomeusuario() . "|";
30.    $resultado->getSenha() . "|";
31.    $resultado->getClasse() . "|";
32.    $sessao->getId();
33.  }
34.  exit;
35. } catch (Exception $e) {
36.  echo "erro|" . $e->getMessage();
37.  exit;
38. }
39.
40. echo "ok|sem resultado";
41. ?>
```

Figura 43 - Código fonte de GerenciadorDeLogin.php responsável pela autenticação dos usuários no sistema



Note que, o script valida os dados recebidos da camada **visão**-linhas 4 e 5 – e invoca o método estático `testaLogin` da classe `VsUsuario` (linha 16), que deve devolver `null`, se o usuário não existe ou um novo objeto do tipo `VsUsuario`, caso ele exista no SGDB. No primeiro caso, é devolvida à camada **visão** uma mensagem de erro. No segundo, é criado um novo objeto do tipo `VsSessao` (linhas 21 a 25). Esta sessão é então persistida no SGDB usando-se o método `persiste()` da classe `VsSessao` (linha 26).

Se tudo corre bem, é devolvida ao Unity uma mensagem com todas as informações do Usuário que realizou login e o Id da sessão (linhas 28 a 32).

A estrutura `try...catch` garante que mesmo que aconteça um erro ou uma exceção, seja devolvido à camada de visão uma resposta. As respostas são formatadas da seguinte forma:

Status | resultado

Sendo que o status pode ser **ok** ou **erro**. Se o status for **ok**, ele será seguido por um texto ou uma coleção de objetos separados por `\n` (enter). Se o status for **erro** ele será seguido somente por um texto com a natureza do erro. Esta resposta deverá ser tratada pela camada de visão.

Este formato é seguido por todos os controladores e por uma questão de espaço eles não serão listados aqui.

5.4.3 Camada modelo

Para a implementação da camada **modelo** também foi usada a linguagem PHP. Abaixo, uma lista detalhada do conteúdo da pasta e a finalidade de cada arquivo:

Tabela 7 - Conteúdo da pasta modelo

| Arquivo | Função |
|-------------------------------------|---|
| <code>conexao.php</code> | Arquivo responsável por estabelecer conexão com o SGDB |
| <code>VsLog.php</code> | Classe PHP responsável por recuperar e persistir as informações de um log no SGDB |
| <code>VsStatusScadaTrafo.php</code> | Classe PHP responsável por recuperar e persistir as informações do status do transformador de força no SGDB |
| <code>VsUsuario.php</code> | Classe PHP responsável por recuperar e persistir as informações dos usuários no SGDB |

Na Figura 44, é detalhado o código-fonte da classe `VSUusuário`, responsável por obter as informações do usuário do SGDB.



```
1. <?php
2.
3. class VsUsuario {
4.
5.     private $id;
6.     private $nomeusuario;
7.     private $senha;
8.     private $classe;
9.
10. ...
11. //funções get e set omitidas
12.
13.     public static function testaLogin($nomeUsuario, $senha) {
14. include_once 'conexao.php';
15.
16.         $sql = "SELECT
17. id,
18. nomeusuario,
19. senha,
20. classe
21.         FROM
22. vs_usuario
23.         WHERE
24. nomeusuario = '$nomeUsuario'
25. and
26. senha = '$senha'";
27.         //throw new Exception($sql);
28.         $result = pg_query($dbcon, $sql);
29. if (!$result) {
30. throw new Exception("Não foi possível selecionar");
31. }
32.
33.         $qtos = 0;
34.         $resultado = "";
35.         while ($linha = pg_fetch_array($result)) {
36. $resultado = "ok|" . $linha['id'] . "|" . $linha['nomeusuario'] . "|" .
37.         $linha['classe'];
38.         //echo $resultado;
39.
40.
41.         $vsUsuario = new VsUsuario();
```

```
42.         $vsUsuario->setId($linha['id']);
43.         $vsUsuario->setNomeusuario($linha['nomeusuario'] );
44.         $vsUsuario->setSenha($linha['senha']);
45.         $vsUsuario->setClasse($linha['classe']);
46.
47. if (isset($_SESSION)) {
48. session_destroy();
49. }
50. session_start();
51.
52.         $_SESSION['id'] = $linha['id'];
53.         $_SESSION['nomeusuario'] = $linha['nomeusuario'] ;
54.         $_SESSION['senha'] = $linha['senha'];
55.         $_SESSION['classe'] = $linha['classe'];
56.
57.
58. return $vsUsuario;
59.
60.         //return $resultado;
61.         $qtos++;
62.         //exit;
63. }
64. if ($qtos == 0) {
65.         //throw new Exception("Usuário não encontrado");
66. return null;
67. }
68.
69. pg_close($dbcon);
70. return $resultado;
71. }
72.
73. }
74. ?>
```

Figura 44 - código responsável por obter as informações do usuário junto ao SGDB

A despeito das diferenças de sintaxe do PHP esta é uma classe comum, com atributos e métodos. Por questão de espaço, os métodos get, set e construtor foram omitidos. Por meio do método estático `testaLogin()` – linha 13, a classe recebe um usuário e senha, se conecta ao SGDB – linha 14 - e executa um comando SQL – linha 16.

Segundo Alvarez (2012) SQL ou *Structured Query Language* não é mais do que uma linguagem padrão de comunicação com base de dados. Segundo ele, trata-se de uma linguagem

normalizada que nos permite trabalhar com qualquer tipo de linguagem (ASP ou PHP) em combinação com qualquer tipo de base de dados (MS Access, SQL Server, MySQL).

A instrução SQL da linha 16 pode trazer a tupla correspondente ao usuário e senha ou nenhum. Em seguida é criado um objeto do tipo VSUsuário e que é devolvido ao controlador que chamou a classe.

O SGDB foi escolhido o PostgreSQL, descrito na seção 5.3.2.

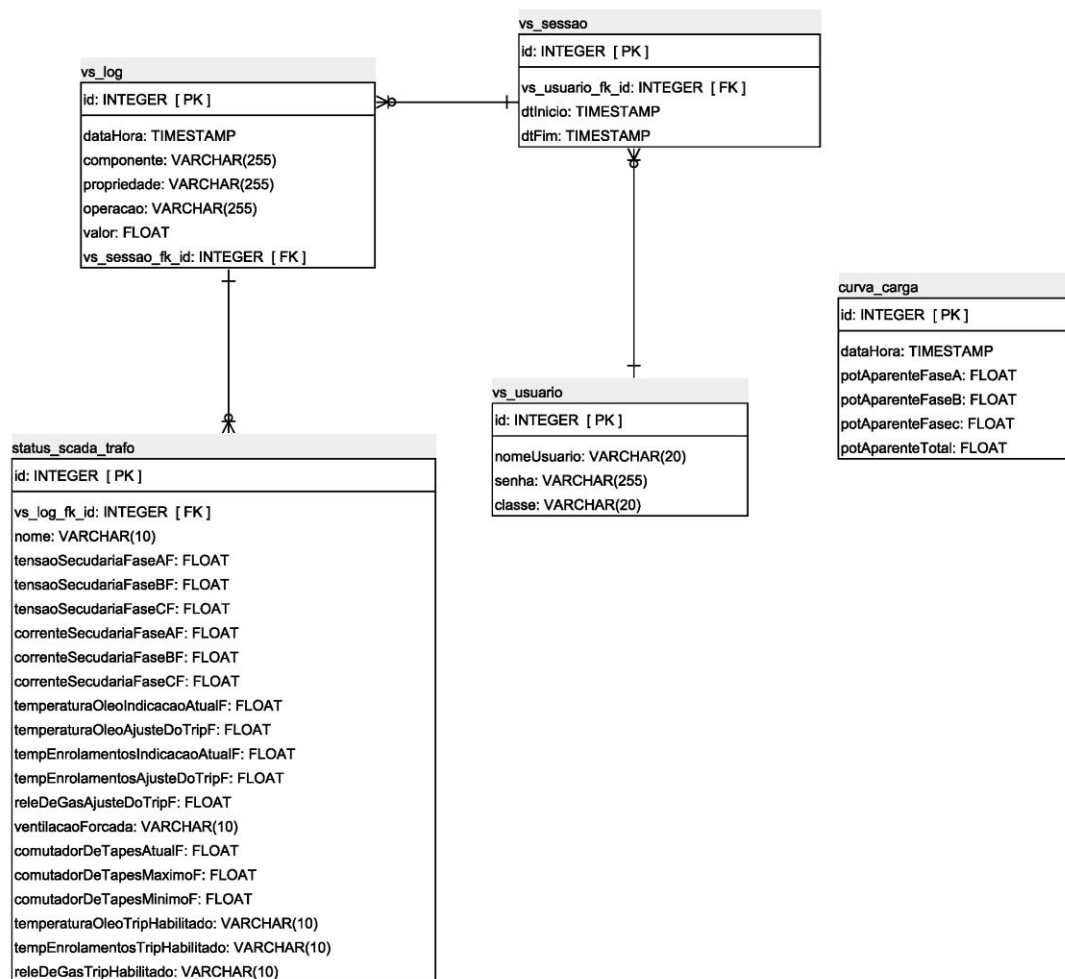


Figura 45 - Diagrama físico do SGDB.

É importante notar que para cada tabela do SGDB existe uma classe associada, com a mesma função dos itens da tabela 8, responsável pela recuperação de suas informações, da mesma forma como é feito quando se usa um *framework* de mapeamento Objeto-Relacional como o *Hibernate*.

Segundo King et al. (2012) trabalhar com software orientado a objetos e um banco de dados relacional pode ser pesado e desperdiçar muito tempo nos ambientes corporativos atuais.



O *Hibernate* é uma ferramenta para mapeamento Objeto/Relacional para ambiente Java. O termo Objeto/Relacional (ORM) se refere à técnica de mapear uma representação de um modelo baseado em objetos para um modelo relacional, com um esquema baseado em SQL.

Ainda segundo os autores, o *Hibernate* não apenas cuida do mapeamento das classes Java para as tabelas do banco (e dos tipos de dados Java para os tipos de dados do SQL), como também provê consulta a dados e recuperação dos mesmos. Isso pode reduzir significativamente o tempo de desenvolvimento que seria gasto manipulação manual de dados em SQL e JDBC.

Nesta implementação em particular foi usado SQL tradicional e não este tipo de *framework*, mas o uso desta tecnologia significaria uma melhoria significativa, principalmente se o número de classes utilizadas para persistência crescer muito, dificultando a manutenção do sistema.

5.5 Conclusão

Neste capítulo foram discutidas de forma mais detalhada cada uma das camadas do modelo MVC que compõem o sistema, bem como as tecnologias utilizadas para construir cada uma delas. Na camada de visão, são mostrados detalhes de interface de cada uma das três áreas que compõem o sistema, bem como pequenos trechos de código que demonstram a interação entre esta camada e a camada de controlador. Na camada controlador, é mostrada a função de cada um dos arquivos criados para fazer o intermédio entre a visão e a camada de modelo, também com pequenos trechos de código. Já na camada de modelo é discutido o modelo físico do banco de dados e cada uma das classes com as quais ele está relacionando.

O que se percebe é que, do ponto de vista técnico, a implementação atendeu a todos os requisitos levantados junto ao usuário, provendo formas de persistência dos passos realizados pelo Trainee durante sua sessão de treinamento com o Transformador de Força e de recuperação destes passos pelo Instrutor, em sua área específica.

No próximo capítulo é realizada outra etapa natural do processo de desenvolvimento de qualquer software: a validação. Nesta etapa, é assegurado que tal software cumpra com suas especificações e atenda às necessidades dos usuários.



Capítulo 6 - Resultados e limitações

6.1 Introdução

Neste capítulo são realizados testes e validação do Virtual Substation por parte dos usuários. Esta é uma etapa crucial para determinar ou não o sucesso do projeto.

A NBR ISO/IEC 9126-1 define **validação** como a confirmação, por exame e fornecimento de evidência objetiva, de que os requisitos específicos para um determinado uso pretendido são atendidos.

Para tanto, inicialmente é apresentada a metodologia estabelecida para a entrevista de alunos, professores e operadores de subestação.

Em seguida, os dados são analisados em sua totalidade através de critérios estabelecidos pela norma ISO citada acima, em sua totalidade.

Na seção seguinte, os dados de alunos e profissionais não analisados em separado, visando determinar eventuais diferenças entre as opiniões destas duas classes.

Complementando a análise, são mostradas algumas das sugestões e considerações dos entrevistados no intuito de melhorar as funcionalidades do sistema virtual.

Encerrando o capítulo, é discutida a contribuição trazida pelo Virtual Substation e realizada uma comparação com as ferramentas correlatas analisadas no capítulo 3.

6.2 Metodologia

Para o processo de validação, foram escolhidos seis alunos do curso de Engenharia Elétrica, do laboratório de Qualidade da Energia Elétrica do Programa de Pós Graduação em Engenharia Elétrica da UFU. Estes alunos, por possuírem maior conhecimento na área de fornecimento e transmissão de energia, estavam habilitados a analisar o protótipo, sob a perspectiva de potenciais Trainees.

O protótipo também foi avaliado por um professor de Engenharia Elétrica da Universidade de Uberaba (UNIUBE) que tem ampla experiência no projeto e construção de subestações de pequeno e médio porte.

Complementando a amostragem, o projeto também foi avaliado por dois operadores de subestação que já trabalham na área há anos.

Para registrar a opinião dos entrevistados, foi elaborado um questionário baseado na ISONORM 9126-1 (adotada para avaliação de software) e foram também considerados alguns aspectos de software educacional descritos por Campos e Campos (2001).

Esta norma estabelece que, no que diz respeito à avaliação de *software*:

As medições sejam objetivas, empíricas e utilizem uma escala válida e, ainda, sejam reprodutíveis.

- para serem objetivas, deve haver um procedimento escrito e acordado para assinalar o número ou categoria ao atributo do produto;
- para serem empíricas, os dados devem ser obtidos por observação ou através de um questionário com validação psicométrica;
- para usarem uma escala válida, os dados devem ser baseados em itens de valor igual ou itens com um valor conhecido. Se uma lista de verificação for utilizada para fornecer dados, convém que os itens sejam ponderados, se necessário;
- para serem reprodutíveis, os procedimentos de medição devem resultar nas mesmas medidas (dentro de tolerâncias adequadas), sendo obtidas por pessoas diferentes fazendo as mesmas medições do produto de *software* em diferentes ocasiões. (ABNT, 2003)

A partir disto, um conjunto de 16 questões de múltipla escolha foi elaborado e uma questão discursiva para que o entrevistado pudesse registrar suas considerações sobre o *software*, apresentado críticas e sugestões.

As questões de múltipla escolha foram agrupadas conforme os seguintes critérios:

- **Funcionalidade:** onde se verifica se o conjunto de funções do sistema atende às necessidades explícitas e implícitas para o qual foi projetado;
- **Usabilidade:** para avaliar a facilidade de uso do *software*;
- **Confiabilidade:** onde se nota se o desempenho do sistema se mantém no decorrer do tempo;
- **Eficiência:** para verificar se os recursos e os tempos envolvidos são compatíveis com o tempo de desempenho requerido para o produto.
- **Portabilidade:** onde se mede a capacidade do produto de *software* para ser transferido de um ambiente para o outro.
- **Sobre o sistema:** onde se analisa o sistema dentro de um contexto de treinamento.

Nestas questões, as únicas alternativas possíveis eram:

1. Muito Satisfeito
2. Satisfeito
3. Insatisfeito

Antes de responderem ao questionário, os entrevistados foram apresentados à proposta do projeto e assistiram a uma rápida demonstração do sistema. Em seguida, eles puderam executar o aplicativo em seus próprios computadores, primeiramente, em versão web e depois versão *desktop*.

6.3 Resultados

Esta seção apresenta os resultados da avaliação do sistema, a partir de dados que foram coletados dos questionários e apresentados em forma de gráficos.

6.3.1 Funcionalidade

Quando responderam à pergunta *O software auxilia no processo de aprendizagem na finalidade proposta?*, a maioria dos entrevistados se mostrou muito satisfeita, o que demonstra que a ferramenta pode ser de auxílio no ensino do funcionamento de uma subestação elétrica, conforme pode-se verificar na Figura 46. É interessante notar que não houve nenhum entrevistado que se mostrou insatisfeito. Os 14% restantes se mostraram apenas satisfeitos, o que indica que ainda há pontos que podem ser melhorados no simulador. Exemplos de melhorias foram sugeridos por usuários como, por exemplo, mostrar o caminho percorrido pela energia na subestação até chegar ao transformador e a possibilidade de incluir formulários e ordens de trabalho que são parte integrante do trabalho do operador em campo.

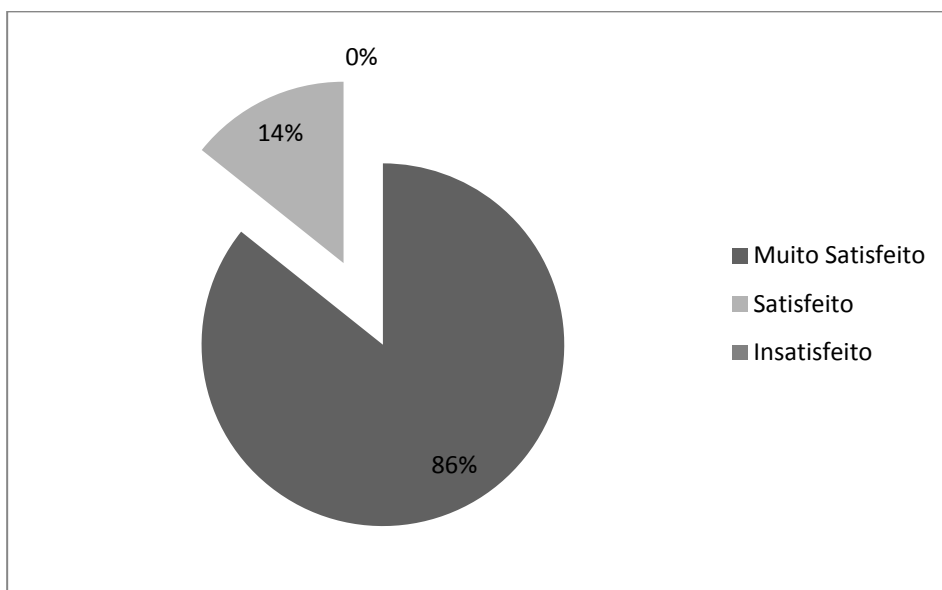


Figura 46 - Respostas dos entrevistados à questão "1.1. O software auxilia no processo de aprendizagem na finalidade proposta?"

Já ao responder a pergunta *Como avalia globalmente a funcionalidade do software?*, A maioria dos usuários se mostrou apenas satisfeita o que reforça o que foi percebido na pergunta anterior, que ainda existem pontos do simulador que podem ser refinados, como, por exemplo, a navegação que poderia ser mais ágil, conforme foi relatado por um dos entrevistados.

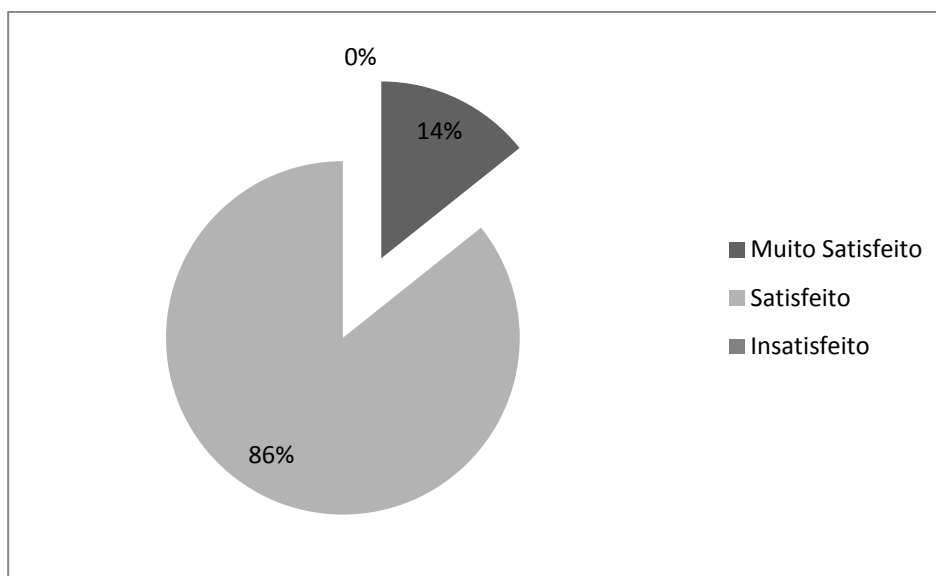


Figura 47 - Resposta dos entrevistados à pergunta "1.2. Como avalia globalmente a funcionalidade do software"

Por meio da observação da Figura 46 e da Figura 47 é possível concluir que a função do *software* está clara e que a maioria dos entrevistados entende seu propósito e o visualiza como uma ferramenta válida para o aprendizado.

6.3.2 Usabilidade

Quando questionados *Em relação à facilidade de entender com funciona o programa*, a maioria dos entrevistados se mostrou muito satisfeita, o que quer dizer que eles o consideraram muito fácil de usar. Os 29% restantes mostraram-se apenas satisfeitos, o que indica que eles consideram que a interface ainda pode melhorar para que seu funcionamento fique mais claro. Um dos usuários menciona que é necessário indicar no console do sistema SCADA qual é o número do transformador de força que está sendo manipulado, uma vez que na simulação existem vários componentes iguais.

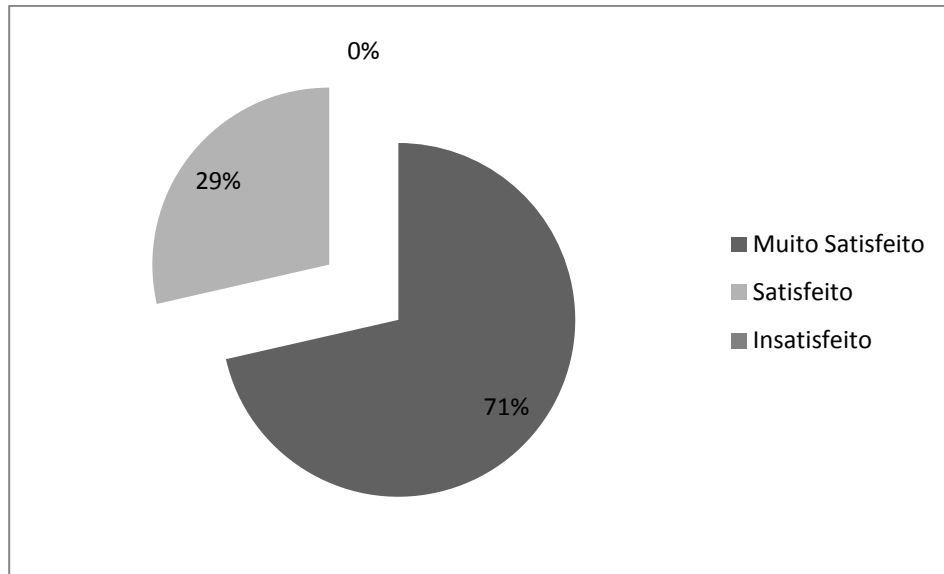


Figura 48 - Respostas dos entrevistados à pergunta "2.1. Em relação à facilidade de entender com funciona o programa"

Já na pergunta *Quanto à navegação na cena e manipulação dos objetos* (facilidade de uso) a maioria dos usuários mostrou-se apenas satisfeita, o que quer dizer que há espaço para melhorias. Uma melhoria neste sentido, mencionada por um dos entrevistados, seria a possibilidade de acelerar a velocidade com que a câmera caminha pelo cenário. Segundo o entrevistado, mesmo tendo a possibilidade de saltar diretamente para um dos componentes, por meio de botões próprios, ao caminhar livremente, o usuário demora em se deslocar pela subestação. Isto poderia ser resolvido criando-se movimentos de corrida através do clique duplo do mouse, como acontece em alguns jogos em primeira pessoa. Outro usuário sugeriu a possibilidade de a simulação utilizar mouse 3D, que possui maior grau de movimentação que o mouse tradicional o que poderia enriquecer a experiência de navegação no simulador. Os outros entrevistados mostraram-se muito satisfeitos. Os resultados dessa pergunta podem ser vistos na Figura 49.

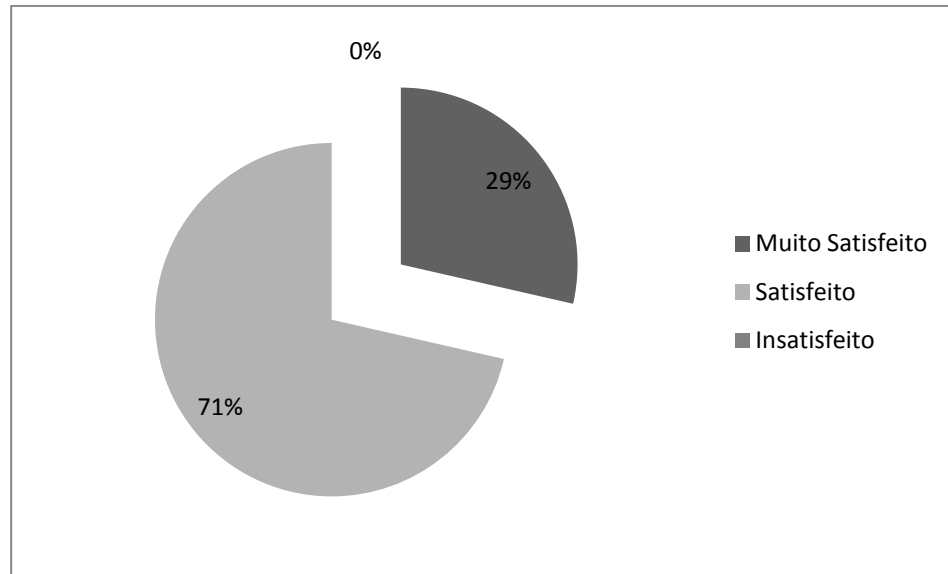


Figura 49 - Respostas dos entrevistados à questão "2.2. Quanto à navegação na cena e manipulação dos objetos (facilidade de uso)"

6.3.3 Confiabilidade

No que tange à *Capacidade de continuar a funcionar corretamente* após erros do próprio software ou erros de manipulação de dados (Figura 50), o que se verifica é que a maioria dos entrevistados mostrou-se apenas satisfeita, o que demonstra, que, para estas pessoas, embora o software continue funcionando bem após um erro, ele não comunica de forma clara o suficiente o que ocorreu. Com isto conclui-se que este aspecto deve ser mais bem trabalhado em versões futuras. Os 43% restantes mostram-se muito satisfeitos. Assim como em todas as alternativas restantes não houve insatisfeitos.

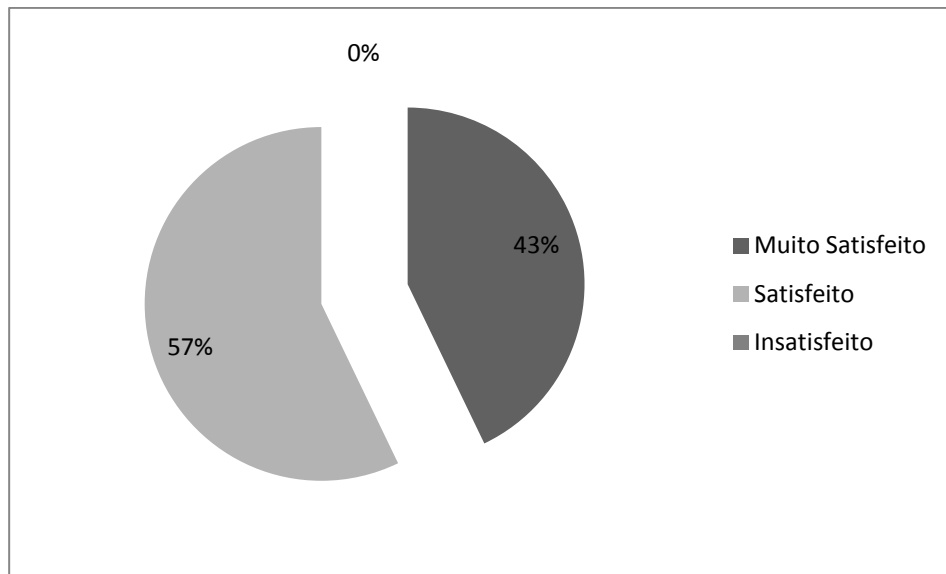


Figura 50 - Respostas dos entrevistados à questão "3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados."

Quando questionados sobre a *Capacidade de o software enviar mensagens de erro caso os mesmos ocorram* (vide Figura 51), a maioria dos entrevistados se mostrou satisfeita, o que reforça o que já foi apurado na pergunta anterior – que a comunicação entre o software pode ser aperfeiçoada, através de mensagens de erro mais claras e mais visíveis na tela, pois, por exemplo, um erro ao carregar a curva de carga é mostrado apenas no painel no canto esquerdo, não ficando tão visível para um usuário desatento.

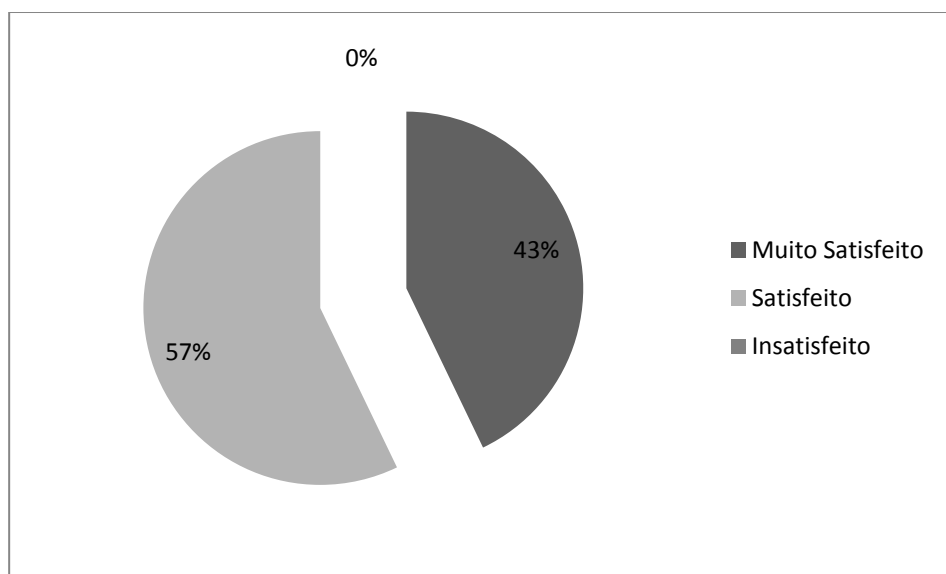


Figura 51 - Resposta dos entrevistados à questão "3.2. Capacidade de o software enviar mensagens de erro caso os mesmos ocorram "

6.3.4 Eficiência

A maioria dos entrevistados considerou o *Tempo de resposta às ações do usuário* muito satisfatório, o que demonstra que o *software* não apresenta gargalos ou demoras ao processar respostas à interação dos usuários. O restante mostrou-se apenas satisfeitos, uma vez que no entendimento destas pessoas o tempo de resposta ainda pode ser melhorado. Mais uma vez, conforme demonstra a Figura 52 não houve usuários insatisfeitos.

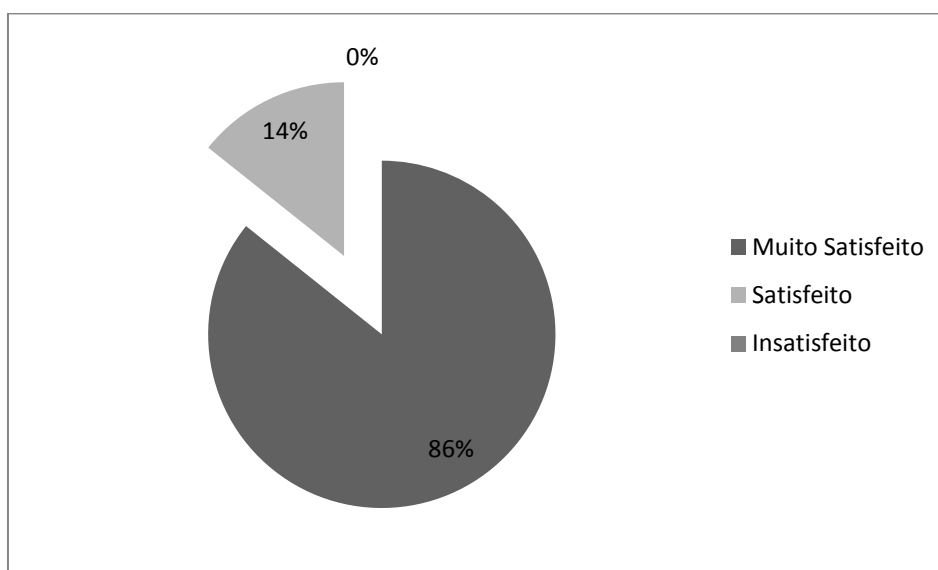


Figura 52 - Respostas dos entrevistados à questão "4.1. Tempo de resposta às ações do usuário"

No que concerne à *Fidelidade a uma subestação e um transformador de força real* a maioria dos usuários (71%) se mostrou apenas satisfeitos. Um dos usuários afirma que a presença de torres de força no modelo 3D ajudaria a tornar o cenário mais realista. No entender de outro usuário, alguns equipamentos não parecem suficientemente conectados, o que indica que há erros na modelagem geométrica. Já para um terceiro, não fica clara a movimentação de peças articuladas – por exemplo, chaves seccionadoras. Isto é compreensível, uma vez que o modelo testado não demonstrou forma de interação com nenhuma peça deste tipo. Conforme mostra a Figura 53, uma porcentagem menor (29%) se mostrou muito satisfeita com a fidelidade a uma subestação e não houve ninguém que tenha considerado que o simulador não ofereça fidelidade alguma aos objetos do mundo real.

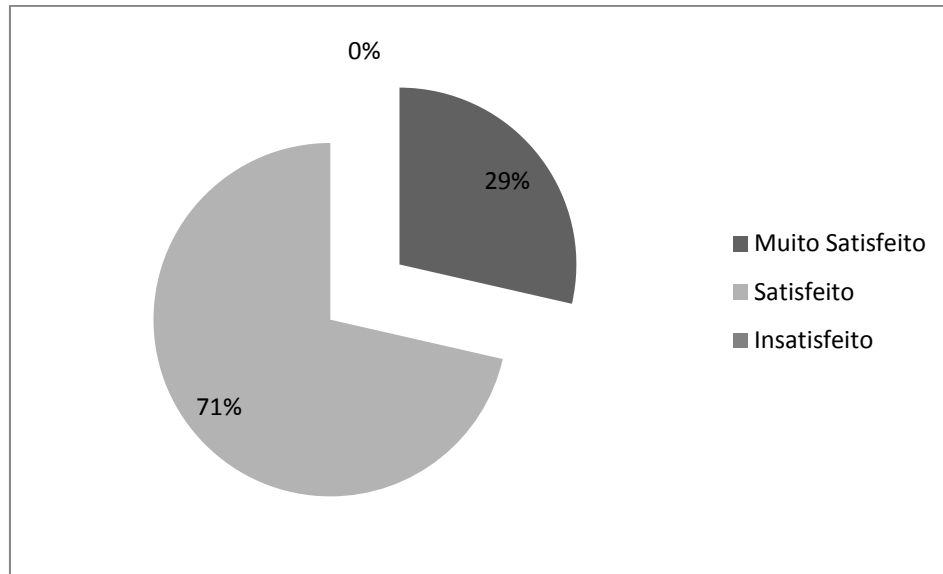


Figura 53 - Respostas dos entrevistados à questão "4.2. Fidelidade a uma subestação e um transformador de força reais"

Conforme demonstra a Figura 54, ao serem perguntados *Como avalia em termos globais a eficiência do software* a maioria dos usuários se mostrou que embora esta eficiência seja satisfatória, ela pode ser melhorada. Isso demonstra que aspectos isolados, como a comunicação de erros, navegação e fidelidade interferem na visão global do simulador. Os outros 29% restantes e mostraram muito satisfeitos com o *software*.

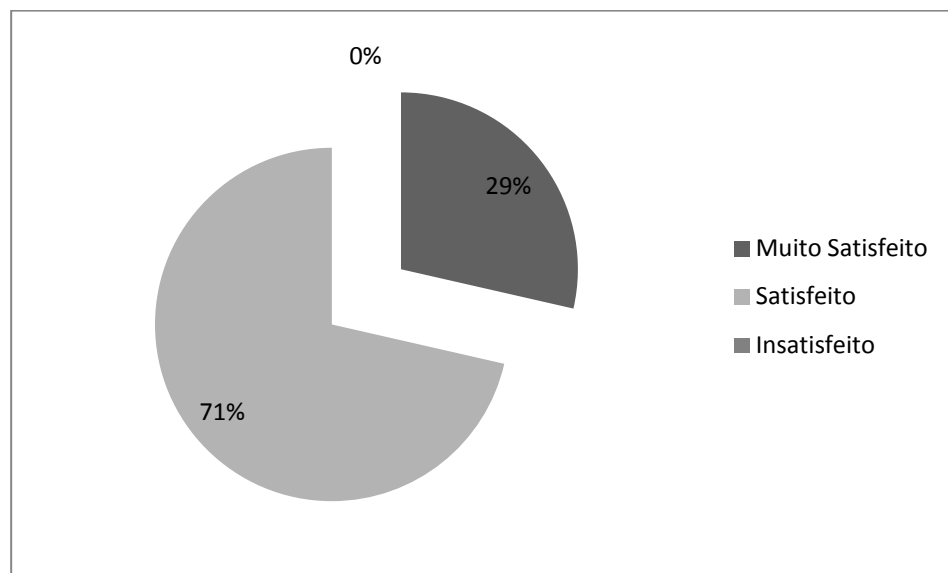


Figura 54 - Respostas dos entrevistados à questão "4.3. Como avalia em termos globais a eficiência do software"

6.3.5 Portabilidade

A maioria dos entrevistados se mostrou satisfeitos com a *Facilidade de execução do software em diferentes ambientes (Executável Windows/Página web)*. Os entrevistados restantes se mostraram muito satisfeitos e não houve nenhuma demonstração de insatisfação. Isso demonstra que no geral, o software não sofre muitas mudanças ao ser transposto de um ambiente para outro. Os resultados dessa pergunta podem ser vistos na Figura 55. Um aspecto que contribuiu para o baixo percentual de muito satisfeito foi a dificuldade que alguns usuários tiveram ao instalar o plug-in *Unity Player* para executar a versão web. Este processo se mostrou demorado em locais com baixa velocidade de conexão com a internet e confuso para alguns deles.

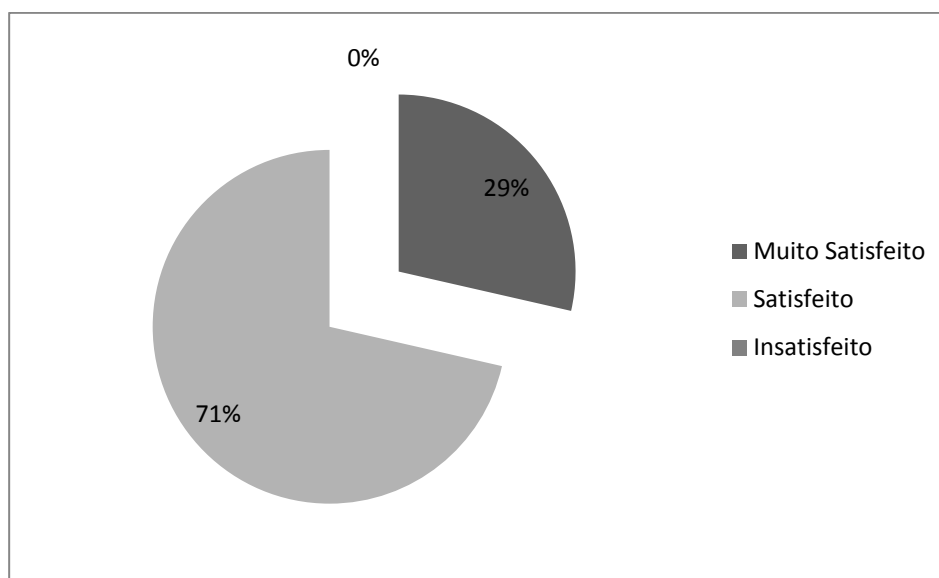


Figura 55 - Respostas dos entrevistados à questão "5.1. Facilidade de execução em diferentes ambientes (Executável Windows/Página web)"

Quanto á pergunta *Como avalia em termos globais a Portabilidade do software?*, a maioria dos entrevistados se mostrou satisfeita e os usuários restantes se mostraram muito satisfeitos, o que reforça a ideia de que as funcionalidades do software não sofrem nenhuma mudança ao serem portados para um novo meio computacional, como pode ser visto na Figura 56.

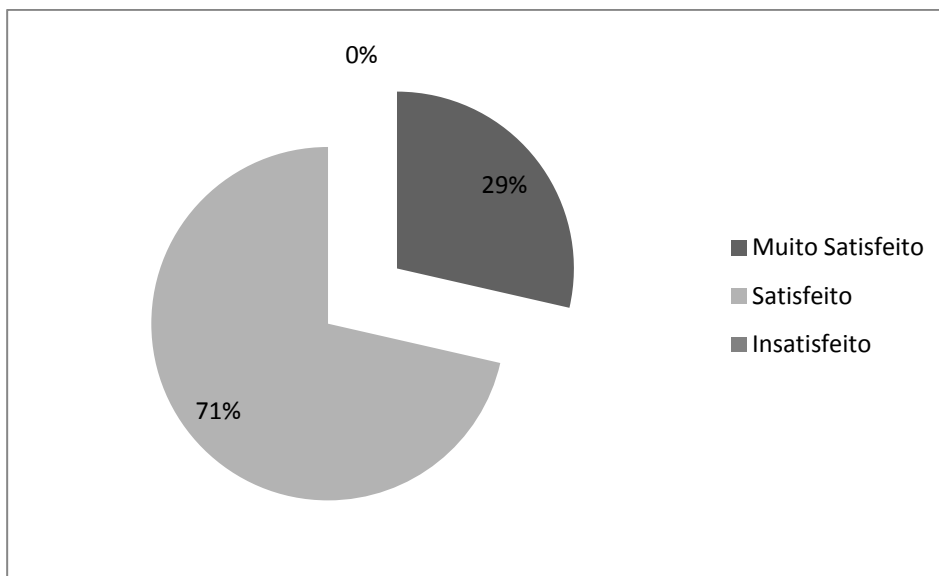


Figura 56 - Respostas dos entrevistados à questão "5.2. Como avalia em termos globais a Portabilidade do software"

6.3.6 Sobre o sistema

A Figura 57 demonstra que, quando perguntados sobre *A importância que atribui ao software para o aprendizado em geral*, a maioria dos usuários considerou muito importante, enquanto os demais consideraram como importante. Isso demonstra que, no entender dos entrevistados, o aplicativo analisado é relevante como ferramenta pedagógica.

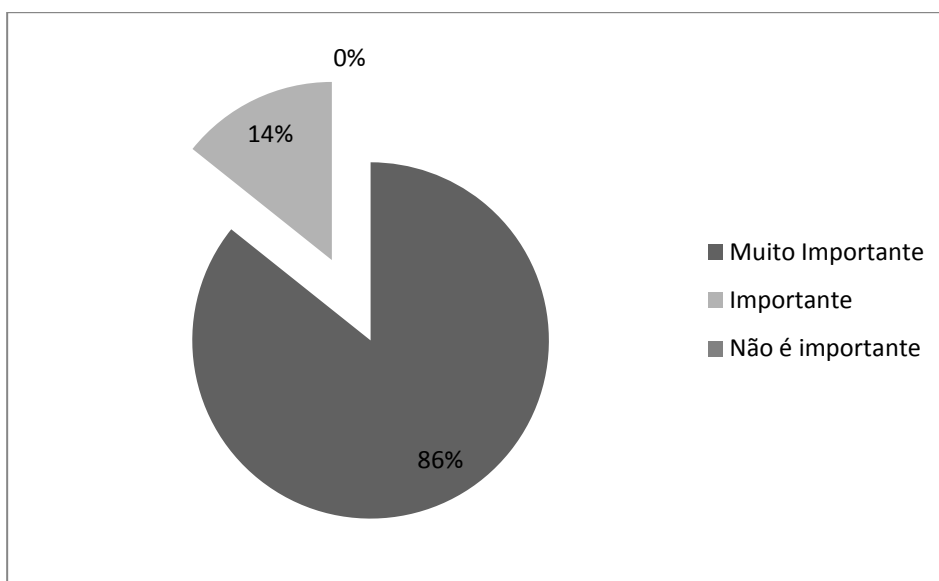


Figura 57 - Respostas dos entrevistados à questão "6.1. A importância que atribui ao software para o aprendizado em geral."

Já sobre *A importância de treinar em Ambientes de Realidade Virtual*, a maioria considerou muito importante ou importante, o que demonstra que compreendem a importância desse tipo de ferramenta para o treinamento, conforme pode ser visto na Figura 58.

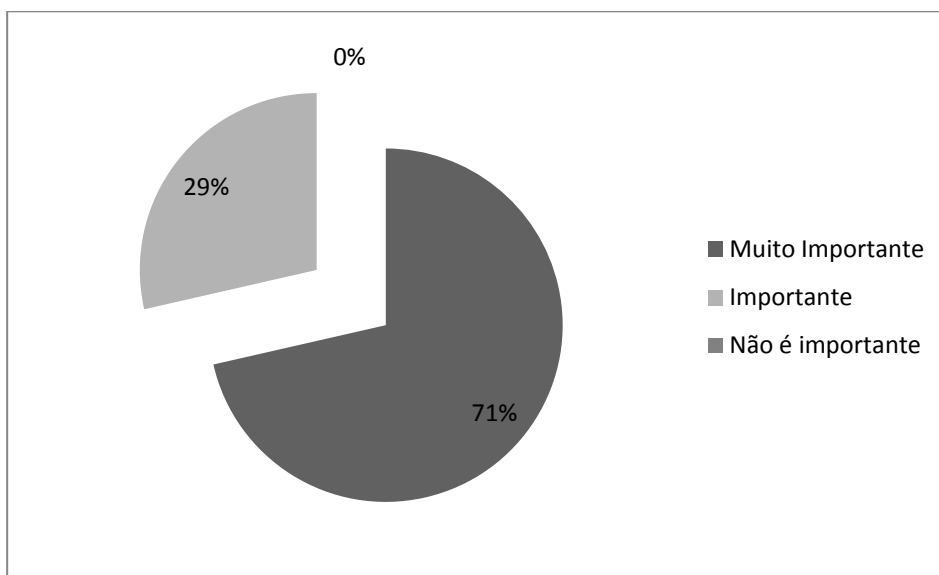


Figura 58 - Resposta dos entrevistados à questão "6.2. A importância de treinar em Ambientes de Realidade Virtual"

Em resposta à pergunta *A importância de treinar em um simulador de transformador de força* maioria dos entrevistados achou muito importante. O restante achou importante e nenhum deles assinalou que não é importante. Isto demonstra que no entender dos entrevistados esta utilização traria grandes ganhos ao processo de aprendizagem de novos operadores e na formação continuada dos já existentes, uma das principais justificativas para a criação do protótipo em questão.

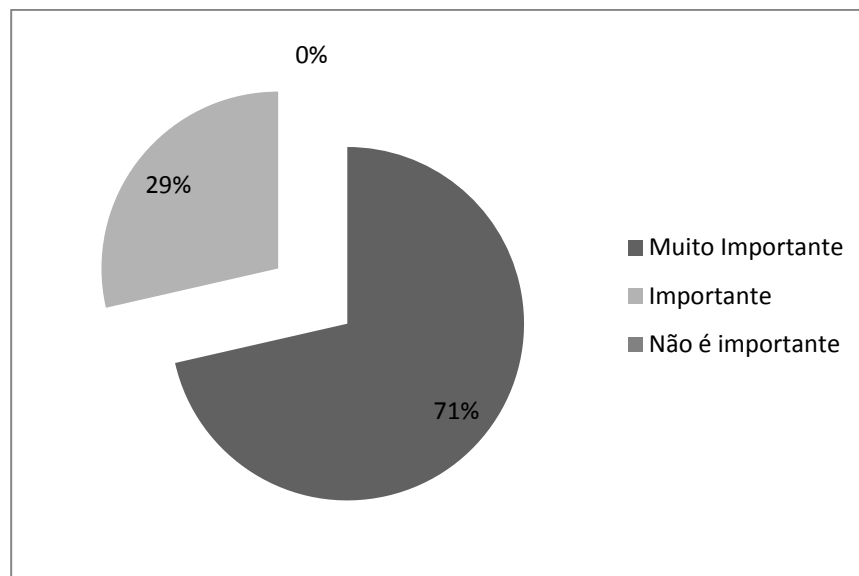


Figura 59 - Resposta dos entrevistados à questão "6.3. A importância de treinar em um simulador de transformador de força"

Finalmente, quando questionados sobre *A importância geral do sistema no treinamento de um operador do sistema* a maioria considerou-o muito importante. Os outros entrevistados demonstraram que é importante. Isso reforça a validade do *Virtual Substation* como ferramenta de treinamento, conforme já foi apurado na questão anterior.

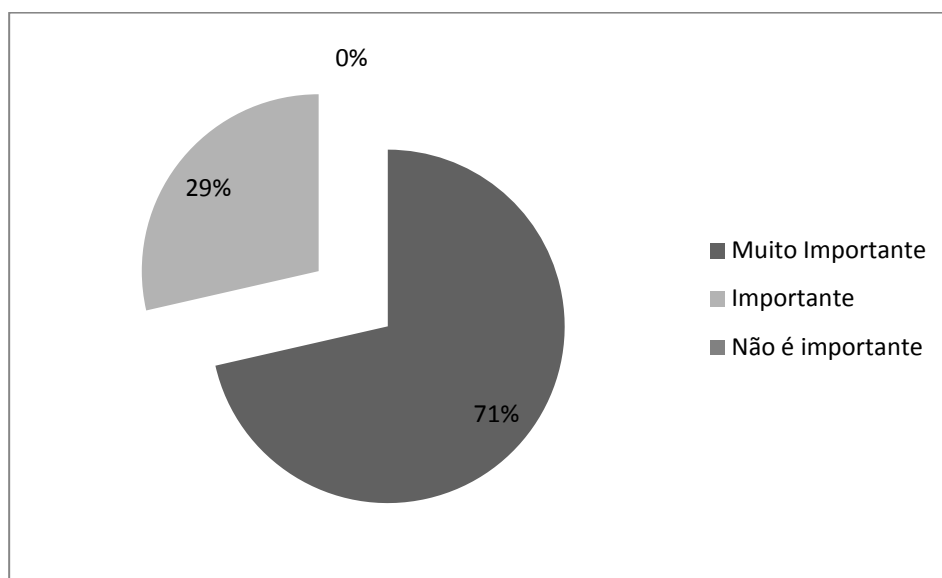


Figura 60 - Respostas dos entrevistados à questão "6.4. A importância geral do sistema no treinamento de um operador do sistema "

Por meio da análise dos dados mostrados nas Figuras Figura 58, Figura 59 e Figura 60, nota-se que a maioria dos entrevistados, mesmo aqueles que não estão familiarizados com

Realidade Virtual considera o simulador como uma ferramenta importante e válida para ser usada em treinamento e no aprendizado de operadores de subestação, demonstrando que o protótipo cumpre o objetivo pelo qual foi projetado.

6.4 Comparações entre as respostas de estudantes e operadores

Nesta seção, é realizada uma comparação entre as respostas dos estudantes, professores e operadores de subestação. Para isto, os entrevistados foram divididos em dois grupos:

- Alunos – que contempla os estudantes de Engenharia Elétrica entrevistados;
- Operadores – que contempla os funcionários de SE e o professor entrevistado.

Esta análise se dará pelos mesmos critérios da realizada anteriormente e se procurará verificar quais aspectos agradam ou desagradam mais cada uma das classes entrevistadas.

6.4.1 Funcionalidade

Ao responder a pergunta *O software auxilia no processo de aprendizagem na finalidade proposta?* a totalidade dos alunos se mostrou muito satisfeita, juntamente com a maioria dos operadores. Apenas 33% por cento dos operadores se mostrou apenas satisfeita, conforme se verifica na Figura 61. Isto demonstra que na opinião da maioria dos entrevistados, a contribuição do software para o aprendizado é totalmente válida e que para alguns dos operadores ele pode ser melhorado para contribuir mais.

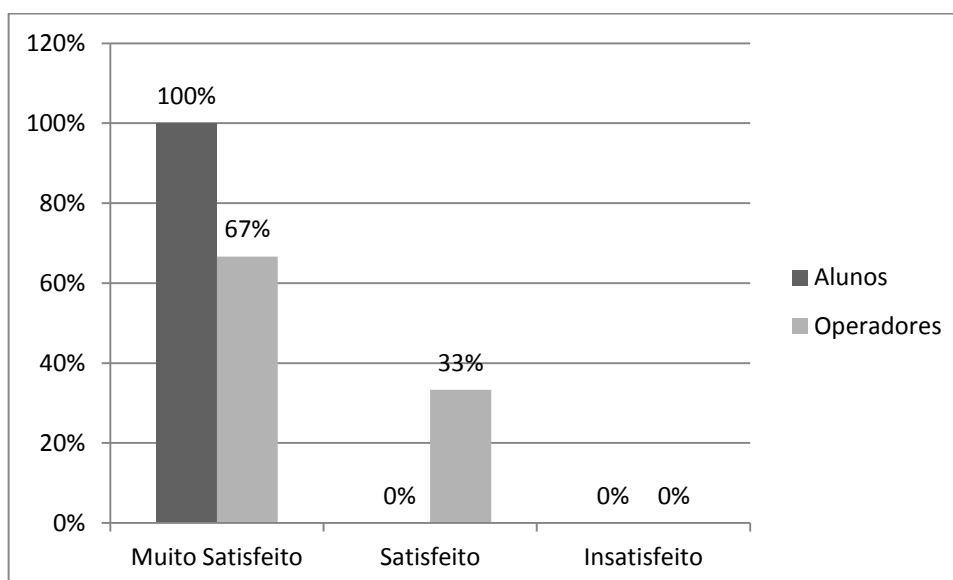


Figura 61 - Comparação das respostas dadas por estudantes e operadores á pergunta “1.1. O software auxilia no processo de aprendizagem na finalidade proposta?”

Já em resposta a *Como avalia globalmente a funcionalidade do software* (Figura 62) a totalidade alunos se mostrou apenas satisfeita, bem como a maioria dos operadores. Apenas 33% dos últimos achou o simulador muito satisfatório neste aspecto. Isso provavelmente se deve ao fato mencionado por alguns dos operadores de que eles julgaram faltar à simulação algumas funcionalidades com as quais estão familiarizados, como por exemplo formulários para abertura de ordem de serviço e seleção de equipamentos de proteção individual. É provável que a adição destes detalhes daria maior credibilidade ao simulador.

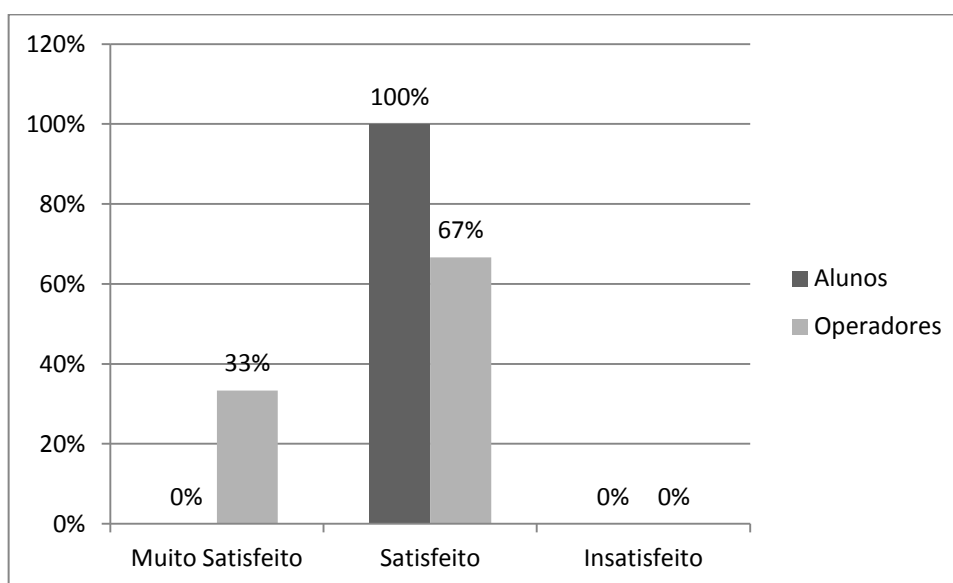


Figura 62 - Comparação das respostas dadas por estudantes e operadores á pergunta “1.2. Como avalia globalmente a funcionalidade do software”

6.4.2 Usabilidade

Na Figura 63 são mostradas as repostas dadas à questão *Em relação à facilidade de entender com funciona o programa*. Através da análise da figura percebe-se que, a maioria tanto de alunos quanto de operadores ficou muito satisfeita com a facilidade de uso do software. No entanto, também entre os dois grupos, uma minoria significativa (40% por cento dos alunos e 33% dos operadores) ficou apenas satisfeita o que significa que para atender aos dois grupos, melhorias precisam ser implementadas na interface do programa, para torna-lo realmente fácil de usar.

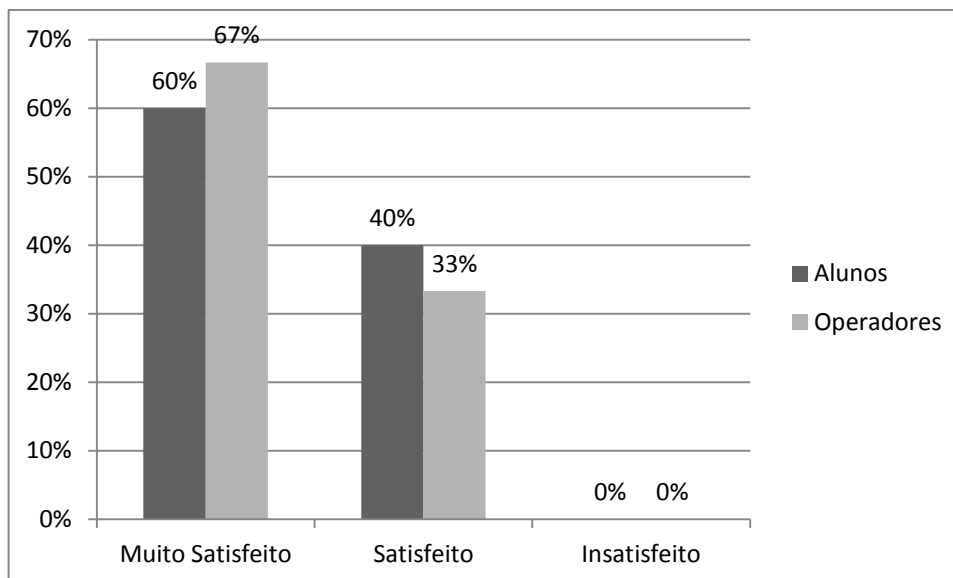


Figura 63 - Comparação das respostas dadas por estudantes e operadores á pergunta “2.1. Em relação à facilidade de entender com funciona o programa”

Já em relação à questão *Quanto à navegação na cena e manipulação dos objetos (facilidade de uso*, a totalidade dos alunos se mostrou apenas satisfeita, juntamente com a maioria dos operadores. Apenas um percentual de 33% dos operadores ficou muito satisfeito com o Virtual Substation neste aspecto. Esse índice baixo de satisfação por parte dos alunos provavelmente se deve ao fato de muitos deles já estarem familiarizados com interfaces 3D interativas complexas, como jogos, por exemplo, bem como alguns de seus pares, operadores de SE. Neste sentido, tanto operadores quanto alunos ofereceram sugestões para melhorar a navegação que, consideraram muito lentas, como por exemplo, a implementação de clique duplo para acelerar a velocidade com que a câmera se desloca pelo ambiente. Também foi solicitada a possibilidade de, ao se clicar num componente, ser aberta uma janela em separado onde o mesmo poderia ser rotacionado, para que se pudesse ter uma visão geral de seus detalhes. Estas respostas podem ser vistas na Figura 64.

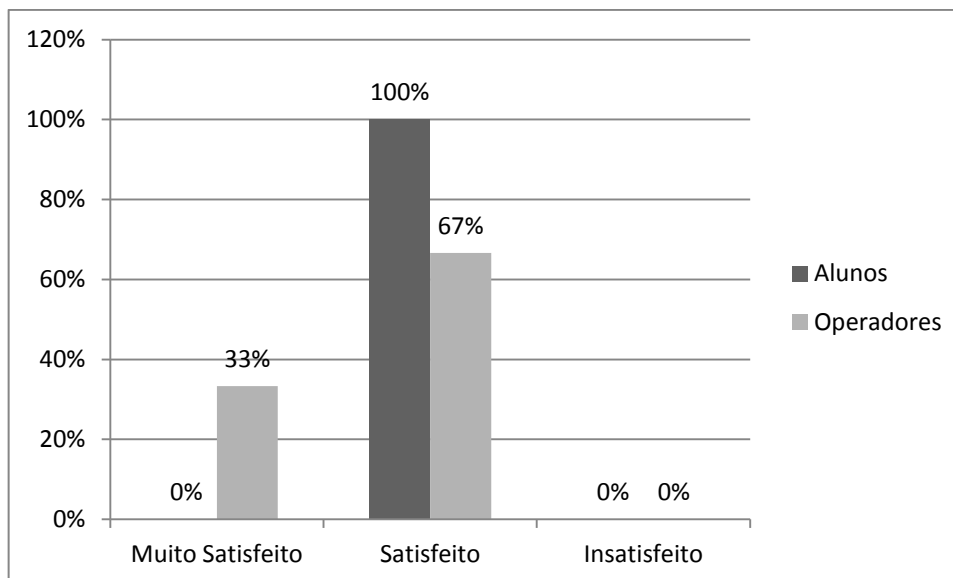


Figura 64 - Comparação das respostas dadas por estudantes e operadores á pergunta “2.2. Quanto à navegação na cena e manipulação dos objetos (facilidade de uso)”

6.4.3 Confiabilidade

Quanto à pergunta *Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados* (Figura 65) as opiniões dos dois grupos se mostram divergentes, uma vez que a maioria dos alunos achou apenas satisfatória enquanto que a maioria dos operadores achou muito satisfatória. Este tipo de resposta também se deve ao fato de o estudantes terem expectativas maiores com relação a ambientes 3D, uma vez que a maioria deles já está acostumada a interagir com jogos 3D. Nenhum dos dois grupos foi questionado sobre este tipo de experiência com ambientes tridimensionais, portanto, na realização de um novo questionário para validar futuras implementações deste software, seria interessante incluir esta pergunta.

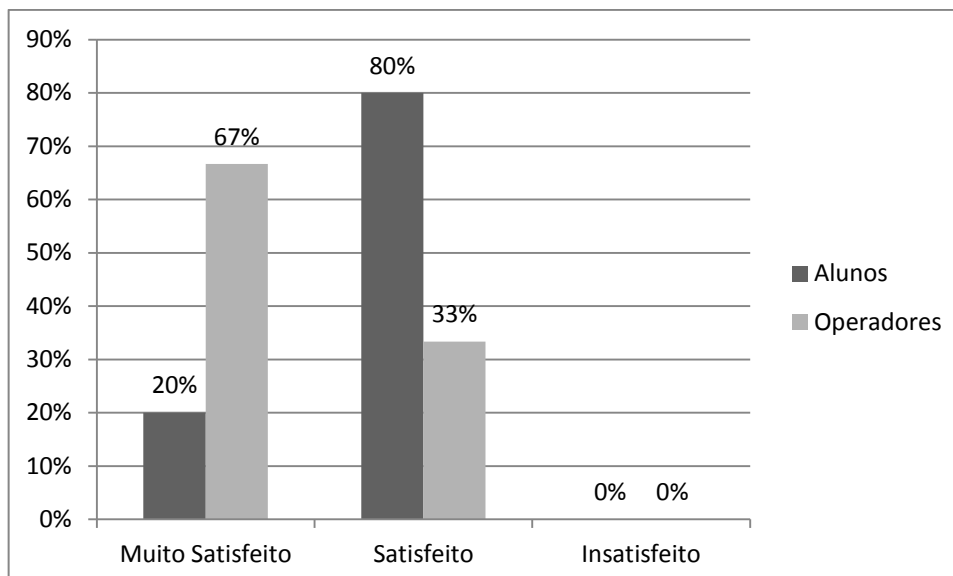


Figura 65 - Comparação das respostas dadas por estudantes e operadores á pergunta “3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados.”

A Figura 66 apresenta uma nova divergência de opiniões entre os grupos quando são questionados sobre a *Capacidade de o software enviar mensagens de erro caso os mesmos ocorram* uma vez que para a maioria dos alunos ela é apenas satisfatória, enquanto que, para a maioria dos operadores ela é muito satisfatória. Especula-se que mais uma vez isso se deva aos fatores comentados na questão anterior.

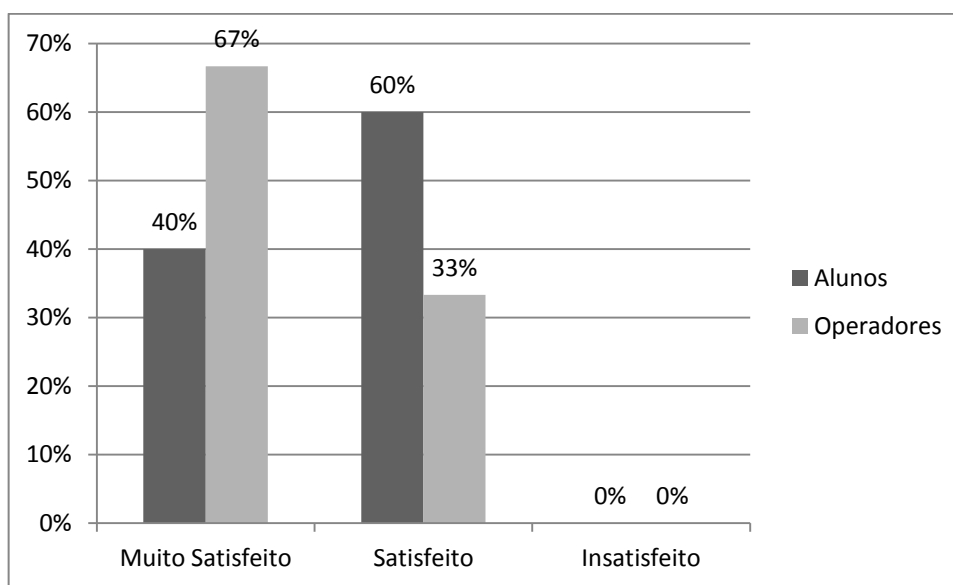


Figura 66 - Comparação das respostas dadas por estudantes e operadores á pergunta “3.2. Capacidade de o software enviar mensagens de erro caso os mesmos ocorram “

6.4.4 Eficiência

Em relação ao *Tempo de resposta às ações do usuário* a maioria dos alunos concorda com a totalidade dos operadores que tal tempo é muito satisfatório, embora na opinião de uma porcentagem significativa dos alunos considerem apenas satisfatória. No entender destes, ajustes devem ser necessários para melhorar a eficiência do sistema, conforme mostra a Figura 67. Nesse aspecto, é necessária maior investigação, uma vez que o tempo de resposta, principalmente quando se está manipulando um elemento do transformador de força, não depende apenas do poder de processamento da máquina local, dependendo também de fatores como velocidade da internet, no caso do uso da simulação via web.

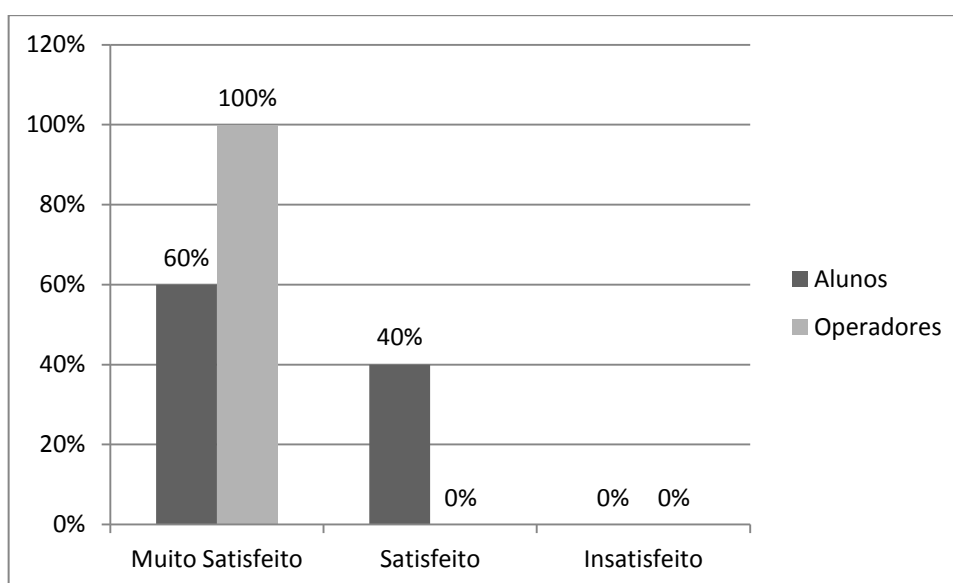


Figura 67 - Comparação das respostas dadas por estudantes e operadores á pergunta "4.1. Tempo de resposta às ações do usuário"

No que diz respeito à *Fidelidade a uma subestação e um transformador de força reais* (Figura 68) ambos os grupos concordam que ela é apenas satisfatória, embora uma porcentagem pequena dos operadores se mostre muito satisfeita. Isso provavelmente se deve ao fato de o modelo geométrico possuir falhas, o que incomodou principalmente aos estudantes.

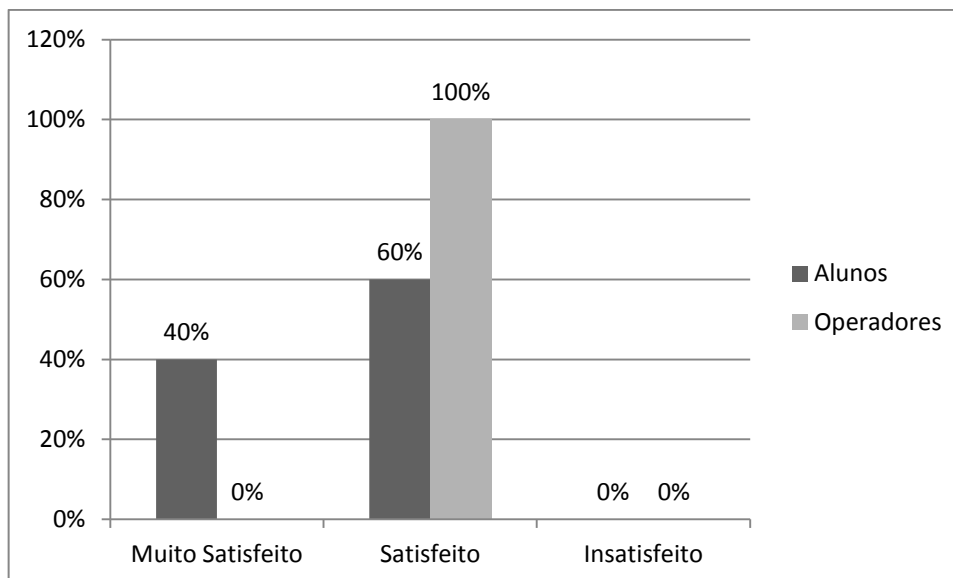


Figura 68 - Comparação das respostas dadas por estudantes e operadores á pergunta "4.2. Fidelidade a uma subestação e um transformador de força reais"

Quando responderam ao questionamento *Como avalia em termos globais a eficiência do software* alunos e operadores se mostraram apenas satisfeitos, enquanto que uma minoria se encontrava muito satisfeita. Isso significa que ajustes na eficiência do software são necessários, embora apenas os estudantes tenham assinalado isto na questão anterior. Estes resultados podem ser vistos na Figura 69.

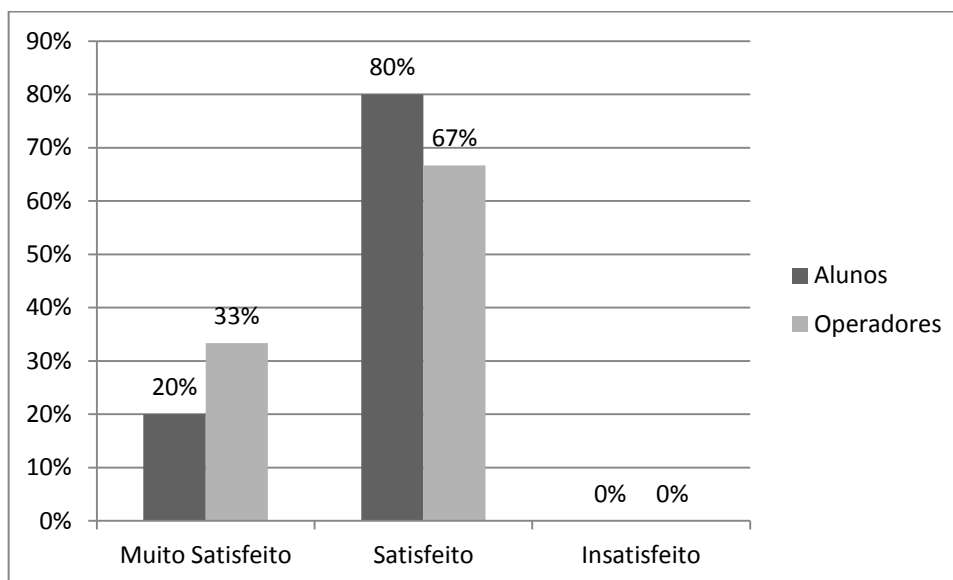


Figura 69 - 4.4. Como avalia em termos globais a Portabilidade do software

6.4.5 Portabilidade

Quando questionados sobre a *Facilidade de execução em diferentes ambientes (Executável Windows/Página web)* - Figura 70 - operadores e alunos concordam que ela é apenas

satisfatória, uma vez que mesmo alunos acostumados a usar páginas web no seu cotidiano mostram dificuldades com a instalação do plug-in do Unity.

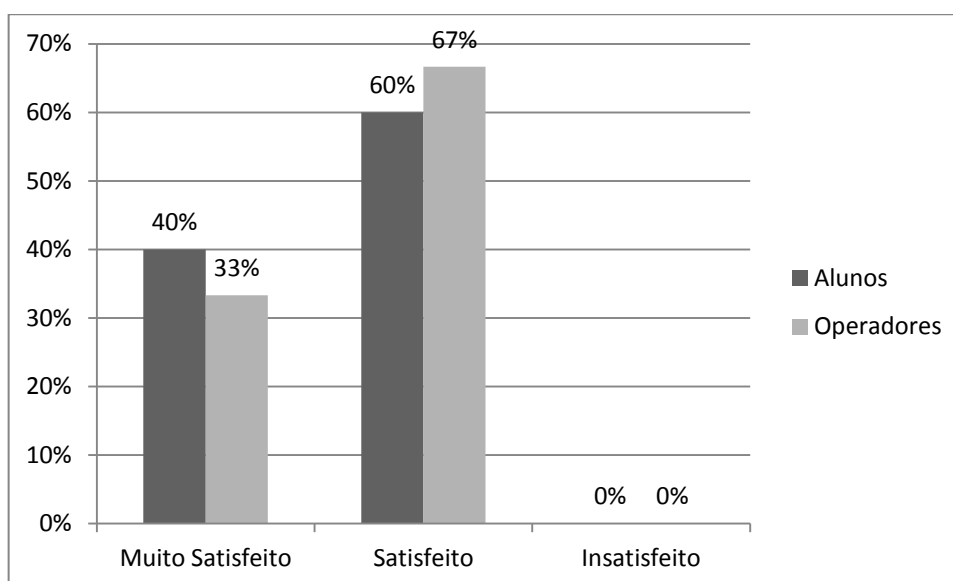


Figura 70 - Comparação das respostas dadas por estudantes e operadores á pergunta “5.1. Facilidade de execução em diferentes ambientes (Executável Windows/Página web)”

Esta concordância se mantém nas respostas à pergunta *Como avalia em termos globais a Portabilidade do software* uma vez que a maioria, tanto de alunos quanto de operadores se mostra apenas satisfeita com questão da portabilidade. Este é um aspecto que precisa ser melhorado. A inclusão de instruções para a instalação antes da tela de login do software foi, inclusive apontada por um aluno como uma forma de minimizar os problemas detectados pela instalação do plug-in.

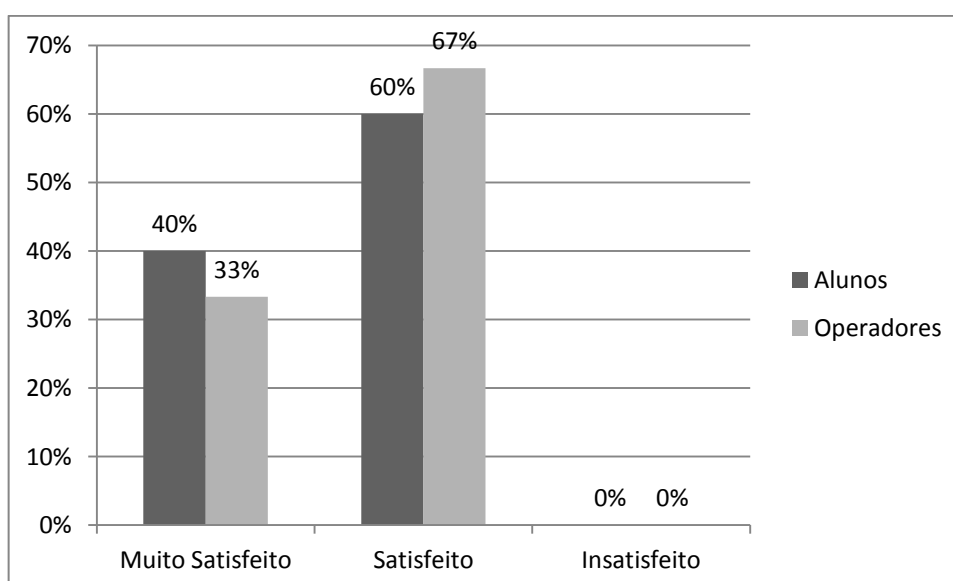


Figura 71 - Comparação das respostas dadas por estudantes e operadores á pergunta “5.2. Como avalia em termos globais a Portabilidade do software”

6.4.6 Sobre o sistema

Em suas respostas à questão *A importância que atribui ao software para o aprendizado em geral*. (Figura 72) a maioria de alunos e operadores afirmou achar muito importante, embora o percentual de operadores esteja 13% menor. Mesmo assim, isso demonstra que no entender de ambas as classes a utilização do simulador como ferramenta pedagógica é relevante.

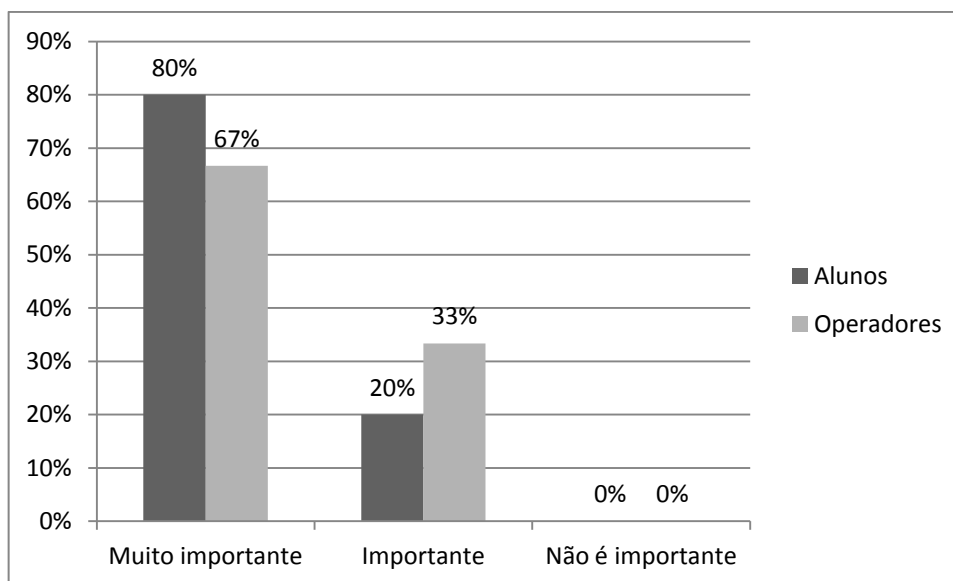


Figura 72 - Comparação das respostas dadas por estudantes e operadores à pergunta “6.1. A importância que atribui ao software para o aprendizado em geral.”

Já quanto à questão *A importância de treinar em Ambientes de Realidade Virtual* a situação se inverte e um porcentagem significativa de alunos (40%) achar apenas importante, conforme pode se observar na Figura 73. É interessante notar que, embora os estudantes provavelmente estejam mais habituados ao convívio com ambientes virtuais, foram os operadores que se mostraram mais entusiasmados.

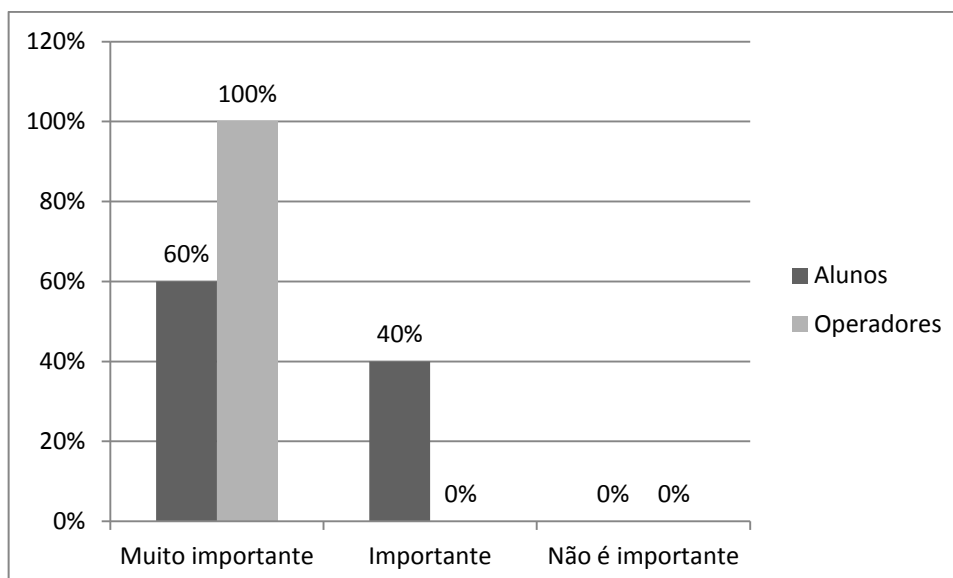


Figura 73 - Comparação das respostas dadas por estudantes e operadores á pergunta “6.2. A importância de treinar em Ambientes de Realidade Virtual”

Também é interessante notar que, quanto *A importância de treinar em um simulador de transformador de força* (Figura 74) os patamares da questão anterior se mantêm. Mais habituados a interfaces 2D, este provavelmente é o primeiro contato de alguns deles com um SRV, o que pode explicar um maior entusiasmo com a riqueza do ambiente 3D e a experiência como um todo.

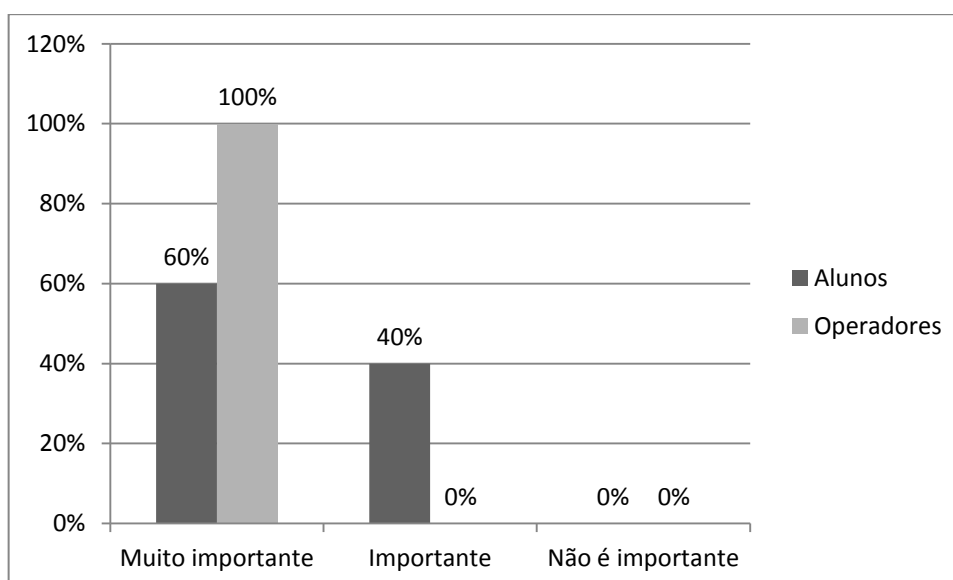


Figura 74 - Comparação das respostas dadas por estudantes e operadores á pergunta “6.3. A importância de treinar em um simulador de transformador de força”

Para finalizar, quanto questionados sobre *A importância geral do sistema no treinamento de um operador do sistema* as duas classes parecem concordar que é muito importante, embora um percentual maior de alunos considere apenas importante, Embora isto aconteça, o

percentual de aceitação demonstra que a contribuição do Virtual Substation para a formação deste tipo de profissional não possa ser desprezada.

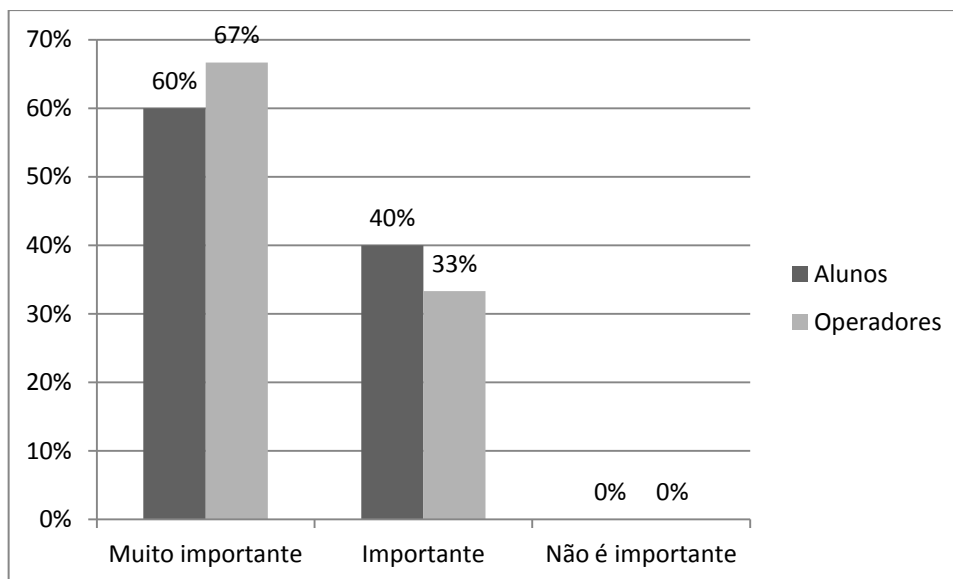


Figura 75 - Comparação das respostas dadas por estudantes e operadores á pergunta “6.4. A importância geral do sistema no treinamento de um operador do sistema”

6.4.7 Resultado da comparação

No quesito *funcionalidade*, o software agradou mais aos alunos que aos operadores na questão “O software auxilia no processo de aprendizagem na finalidade proposta?. e na avaliação global da funcionalidade de software, os operadores se mostraram mais satisfeitos.

No quesito *usabilidade*, o simulador também agradou mais aos operadores. Embora, provavelmente tenham mais experiências com jogos e ambientes 3D, os alunos consideraram o software mais difícil de usar e sua navegação menos satisfatória.

No que diz respeito a *confiabilidade*, mais uma vez os operadores ficaram mais satisfeitos, principalmente no que diz respeito ao envio de mensagens de erro.

Esta situação se inverteu no que diz respeito à *eficiência*. Acostumados a trabalhar com sistemas de interface 2D, que provavelmente apresentam menor tempo de resposta, os operadores acharam o sistema menos eficiente que os alunos.

Ambas as categorias concordam que o sistema atende à *portabilidade* apenas satisfatoriamente, o que indica que este é um aspecto que deve ser mais trabalhado em futuras versões.

Em relação à *seção sobre o sistema* os operadores mais uma vez se mostraram mais entusiasmados com as potencialidades pedagógicas da do sistema e da RV como um todo.

O que se percebe é que de um modo geral, os operadores se mostraram um grau maior de satisfação. No entanto, por se tratarem de um percentual menor da amostra seria interessante, na avaliação de uma próxima versão do sistema, verificar se este nível de satisfação se mantém.

Outro aspecto que se mostrou relevante durante esta comparação e que não foi tratado pelo questionário diz respeito à experiência prévia do entrevistado com ambientes 3D. Numa versão posterior deste estudo esta questão deve ser levantada, uma vez que, quanto maior o conhecimento que o entrevistado tenha travado com ambientes deste tipo, maior aumenta seus níveis de expectativa e de exigência.

6.5 Algumas sugestões dos usuários

Ao final do questionário foi disponibilizado um espaço para comentários e sugestões. Abaixo segue uma transcrição das respostas de alguns usuários:

4. *Ao clicar em um equipamento, seria interessante que fosse aberta uma janela menor com este equipamento em separado girando em 360º.*
5. *Sugiro melhorar a navegação (controle de direção e ângulo de visão simultâneo) e movimentação das peças articuladas.*
6. *Melhorar a navegação com opção de acelerar a caminhada, por exemplo, com clique duplo a câmera correr, como em jogos de primeira pessoa.*
7. *Tentar implementar mouse 3D.*
8. *Acrescentar torres de transmissão para mostrar de onde a energia vem.*
9. *Mostrar de alguma forma o caminho que a energia percorre na subestação.*
10. *Melhorar a conexão de alguns equipamentos que parecem desconectados.*
11. *Indicar no console do sistema SCADA qual é o número do transformador de força que está sendo manipulado, pois há mais de um componente igual dentro da subestação.*
12. *De posse dos procedimentos para treinamento, implementar não somente a parte técnica como também a parte documental e de segurança, como por exemplo:*
 - a. *Abertura de permissões de trabalho;*
 - b. *Utilização de equipamentos de proteção individual;*
 - c. *Sinalizações.*

6.6 Conclusão

Após análise dos dados dos entrevistados e em comparação com outras ferramentas disponíveis, acredita-se que a contribuição do simulador *Virtual Substation* é oferecer uma

ferramenta que suporte a visualização 3D do ambiente, múltiplas formas de interação simultâneas (*two-way*), persistência dos dados em SGDB e a possibilidade de ser executado em mais de um ambiente (em desktops, uma página web e em dispositivos executando os sistemas operacionais iOS e Android, em futuras versões).

Um estudo comparativo entre o *Virtual Substation* e as outras aplicações pode ser verificado na Tabela 8, destacando-se a contribuição do sistema proposto.

Tabela 8 – Comparação do Virtual Substation com demais sistemas.

| Aplicação | Critérios de avaliação | | | |
|---------------------|------------------------|---------|----------------------|----------------------|
| | Visualização 3D | Two-way | Comunicação com SGDB | Versão para internet |
| ArqElectrical | ✓ | ✓ | ✗ | ✗ |
| MineInside® | ✓ | ✓ | ✓ | ✗ |
| RVCOPEL | ✓ | ✗ | ✓ | ✗ |
| Simulador Furnas 01 | ✗ | ✓ | ✓ | ✗ |
| Simulador Furnas 02 | ✓ | ✗ | ✓ | ✗ |
| Virtual Substation | ✓ | ✓ | ✓ | ✓ |

✓=sim ✗=não



Capítulo 7- Conclusões

Após a utilização do questionário com estudantes, professores e operadores de subestação, ficou claro que as técnicas empregadas (camada de visão construída em Unity 3D se comunicando páginas PHP) se mostraram adequadas para construir o simulador de SE proposto.

Também pôde-se verificar que todos os requisitos levantados foram cumpridos, embora ajustes sejam necessários.

Também se verificou que a camada de visão deva passar por ajustes finos no que diz respeito à navegação, ao realismo e à reportagem de erros.

Sobre a metáfora de comunicação bilateral entre o sistema de RV e os dados oriundos do SGDB, o que se verifica é que ela é adequada, embora também deva passar por ajustes a fim de melhorar seu desempenho.

A questão da portabilidade merece atenção especial, uma vez que a maioria dos usuários considerou-a apenas satisfatória e muitos deles reportaram problemas ou dúvidas no momento da instalação do plug-in necessário para rodar a versão web.

Apesar de o único componente testado, por meio de manipulação direta, ter sido o transformador de força, observou-se que é possível e atrativo em um ambiente virtual desta natureza explorar parâmetros de controle para respostas imediatas a problemas de interesse do usuário. Em particular, quanto ao transformador, foi possível verificar que o sistema de RV proposto mostrou de forma fiel, em tempo real e satisfatória o impacto que mudanças nos seus parâmetros têm em uma subestação, principalmente, no que diz respeito a tensões e correntes.

A persistência correta dos dados e a recuperação correta destes demonstram que a estratégia de camadas MVC se mostrou como uma forte candidata para a construção de sistemas virtuais que visam os mesmos princípios de funcionamento. Por meio desta estratégia, a manutenção e posterior expansão do modelo proposto podem ser gerenciadas de forma mais segura.

Quanto ao uso ferramenta Unity 3D, pode-se concluir que a mesma é adequada para os propósitos de simulação e controle da subestação virtual, dada sua flexibilidade para trabalhar com modelos e Figuras criados externamente. A possibilidade de exportar o modelo criado para duas versões (web e *desktop*) também é um ponto positivo desta ferramenta, pois conforme se percebeu os dois modelos, ao serem manipulados pelos entrevistados, não apresentaram diferenças significativas.

A possibilidade de se comunicar com páginas web também se mostra uma funcionalidade interessante, caso seja necessário realizar a integração do modelo com sistemas



legados. A persistência também se mostra um bom acréscimo no que diz respeito ao aprendizado, possibilitando a recuperação dos dados para análise de instrutores e até mesmo para geração de estatísticas sobre os erros mais cometidos, o que permitiria também um maior refinamento do treinamento.

Através das entrevistas, também se percebe que o *Virtual Substation* é uma ferramenta relevante no que diz respeito ao aprendizado de controle e operação de subestações de energia elétrica. Além disso, as técnicas aqui apresentadas podem ser claramente utilizadas para a construção de outros objetos de aprendizagem que não sejam, necessariamente, da área de fornecimento de energia.

Finalmente, pôde-se observar que as principais características de sistemas de RV (navegação, imersão e o envolvimento) demandaram pouco esforço de programação em função da arquitetura proposta e das ferramentas utilizadas para suportá-la.



Capítulo 8 - Trabalhos futuros

Após analisadas as considerações dos entrevistados pode se perceber que embora a ferramenta se mostrasse satisfatória, ainda existem pontos que podem ser melhorados e caminhos que podem ser percorridos para a evolução da ferramenta apresentada.

São eles:

1. **Construção do comportamento de outros componentes da SE**, pois apenas foi modelado o transformador de força. A construção de outros componentes enriqueceria ainda mais o modelo proposto, possibilitando maior aprendizagem por meio do operador de subestação e de estudantes da área de engenharia elétrica em geral.
2. **Investigar se a utilização do *mouse 3D* e outros periféricos como *joysticks* poderia enriquecer a experiência do usuário com o SRV**, já que muitas das aplicações relacionadas apresentadas no Capítulo 3 - Trabalhos relacionados usam este tipo de hardware para interação.
3. **Incorporar rotinas comuns ao operador de subestação que não foram cobertas na construção do protótipo** como formulário de abertura de permissões de trabalho, seleção de equipamentos de prevenção individual antes de ir a campo e sinalizações.
4. **Aperfeiçoamento das formas de interação**: melhoria da navegação, utilização de sistemas de partículas para demonstrar o percurso da energia através dos ramais da subestação e mudanças deste percurso quando, por exemplo, uma chave seccionadora é acionada.
5. **Aplicação de estereoscopia ao modelo virtual**, pois se trata de um SRV não imersivo, que ainda não faz uso das possibilidades de imersão que os atuais óculos 3D podem oferecer.
6. **Colocação de um cenário de fundo no modelo proposto** e verificação se este ajudaria mais a aumentar a sensação de imersão do usuário no sistema proposto ou apenas o poluiria mais visualmente
7. **Criação de uma versão para *tablet* do protótipo**, uma vez que este tipo de dispositivo se torna cada vez mais popular e oferece mecanismos de interação diferenciados, como por exemplo tela sensível ao toque.

Além disto, as técnicas investigadas podem ser usada para a construção de novos trabalhos, como, por exemplo:

1. **Ferramenta para controle de subestação real**: uma vez que o protótipo criado possui todas as características de um transformador de força verdadeiro, com algumas
-

adaptações, a camada controladora poderia ser modificada para enviar e receber informações para um transformador em tempo real.

2. **Criação de ferramentas para treinamento áreas, que não envolvam necessariamente componentes elétricos:** a capacidade de persistir e recuperar dados de um ambiente virtual abre um imenso leque de possibilidades. Com esta abordagem, podem ser criados simuladores de procedimentos diversos, tais como: análises laboratoriais, fabricação de produtos alimentícios, manutenção de computadores, operação de plataformas de petróleo, operação de um navio até procedimentos de segurança em caso de incêndio,



Referências

3D, Unity. **License Comparisons**. Disponível em: <<http://unity3D.com/unity/licenses.html>>. Acesso em: 09 abr. 2011.

ABNT. **NBR ISO/IEC 9126-1**: Engenharia de software - Qualidade de produto. Rio de Janeiro: Abnt - Associação Brasileira de Normas Técnicas, 2003.

AGUIAR, Bernardo; SOARES, Nilson; CAMPOS, Fábio. Possibilidades de interface e imersão em novas tecnologias no design de jogos1. **III Simpósio Nacional Abciber**: Anais do simpósio, São Paulo - Sp, n. , p.1-15, 18 nov. 2009.

ALVES, Ângelo Andelnyr Sampaio; RIBEIRO JÚNIOR, Enéas Macedo. DESENVOLVIMENTO DE METODOLOGIA PARA APERFEIÇOAMENTO DE OPERADORES ATRAVÉS DA IMERSÃO EM AMBIENTE DE SIMULAÇÃO. **Anais do IX Simpase**: Simpósio de Automação de sistemas Elétricos, Curitiba - Pr, n. , p.1-8, 14 ago. 2011.

ANEEL. **Atlas de energia elétrica do Brasil**. Brasília: Agência Nacional de Energia Elétrica (, 2002.

ANEEL. **Atlas de Energia Elétrica do Brasil**. 3a Brasília, 2008. 236 p.

ARAUJO, Regina Borges de. **Computação Ubíqua: Princípios, Tecnologias e Desafios**. Natal, Rn: Ufrn, 2003.

BENTO, João José Fernandes. **Desenvolvimento e avaliação de um ambiente de aprendizagem 3D**. Bragança , Pt: Instituto Politécnico de Bragança, 2011.

BARROS, Rui Tiago de Cruz. **Jogos 3D em tempo real para iPhone / iPad baseados em sensores**. Porto, Portugal: Faculdade De Engenharia Da Universidade Do Porto, 2011.

BERTI, Cláudia Beatriz; NUNES, Fátima L.s.. **Visualização de informação de bases de imagens médicas utilizando realidade virtual**. Disponível em: <<http://www.ime.uerj.br/professores/cecas/V%20WIM/poster/8375.pdf>>. Acesso em: 12 maio 2012.

- BIAZUS, Diogo de Oliveira. **PostgreSQL: Introdução e Histórico**. Disponível em: <http://wiki.postgresql.org/wiki/Introdu%C3%A7%C3%A3o_e_Hist%C3%B3rico>. Acesso em: 08 jun. 2012.
- BOGONI, Tales Nereu et al. Simulador para Treinamento de Operadores de Colheitadeira Axial de Grãos. **Anais do Wrvva'2010: 7 Workshop de Realidade Virtual e Aumentada**, São Paulo - Sp, n. , p.209-213, 08 nov. 2010.
- BOTEGA, Leonardo Castro; CRUVINEL, Paulo Estevão. Realidade Virtual: Histórico, Conceitos e Dispositivos. **Xi Simpósio De Realidade Virtual E Aumentada: Livro do Pré-Simpósio**, Porto Alegre-rs, n. , p.9-30, 25 maio 2009.
- BOWMAN, D. et al., 3D User Interfaces: Theory and Practice, Addison-Wesley, 2005.
- BUCCIOLI, Arthur A. B.; ZORZAL, Ezequiel R.; KIRNER, Claudio. **Usando Realidade Virtual e Aumentada na Visualização da Simulação de Sistemas de Automação Industrial**. Disponível em: <<http://www.realidadeaumentada.com.br/artigos/16585.pdf>>. Acesso em: 12 maio 2012.
- BURIOL, T. ; ROSENDO, M. ; De Geus, K. ; SCHEER, S. ; Felsky, C. . Proposta de Plataforma Baseada em Realidade Virtual Para Treinamento de Atividades em Linha Viva. In: Proposta de Plataforma Baseada em Realidade Virtual Para Treinamento de Atividades em Linha Viva, 2009, Búzios, RJ. **Anais do 30 CILAMCE**, 2009.
- CAMARGO, Clarissa Avelino Xavier de et al. Aplicações de Realidade Aumentada para Ensino de Física no Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Jataí. **Anais do Wrvva'2010: 7 Workshop de Realidade Virtual e Aumentada**, São Paulo - Sp, n. , p.166-171, 08 nov. 2010.
- CAMPOS, F. C. A., CAMPOS, G. H. B. (2001) **Qualidade de Software Educacional** in Rocha, A. R. C. da, Maldonado, J. C. , Weber, K.C. (Orgs.) **Qualidade de Software : Teoria e Prática**. 1. ed. S.,o Paulo: Prentice Hall, p. 124-130.
- CARDOSO, A.e LAMOUNIER JR., E.. **Aplicações na Educação e Treinamento**, chapter 9, pages 343–357. Realidade Virtual e Aumentada Uma abordagem tecnológica, SBC, 2008

CARDOSO, A.; LAMOUNIER, E. Aplicações na Educação e Treinamento. **Realidade Virtual e Aumentada**: Uma Abordagem Tecnológica, João Pessoa – Pb, n. , p.343-357, 13 maio 2008.

CARDOSO, A. KIRNER, C. LAMOUNIER, E. KELNER J.. **Tecnologias para o desenvolvimento de sistemas de Realidade Virtual e aumentada**. Recife: Ed. Universitária da UFPE, 2007.

CARRARD, Marcos. **Tutorial VRML 2.0**. Disponível em:

<www.lcvdata.kinghost.net/downloads/tutorial_vrml.pdf>. Acesso em: 14 jul. 2012.

CLARKE, Jim; CONNORS, Jim; BRUNO, Eric. **Java FX: Desenvolvendo Aplicações de Internet Ricas**. Rio de Janeiro: Alta Books, 2010. 324 p. (Java Direto da Fontes).

COELHO, D. R. COELHO R. R. and CARDOSO, A. "Use of Virtual Reality in process control in a mine in Brazil – a case study". **Anais do XIII Symposium on Virtual and Augmented Reality**, Uberlandia, MG,

CORTI, K., **Games-based Learning: A Serious Business**

Application. <http://www.pixelearning.com/docs/seriousgamesbusinessapplications.pdf>, 2006. Acesso em novembro de 2011.

COSTA, M.R. da; PEREIRA, L A C; ALVES, J M T. **A importância e a evolução de funções de análise de redes no sistema de supervisão e controle**. Anais do Xiii Edao, Recife, Pe, n. , p.1-5, 06 mar. 2005.

COTRIN, Ademaro A. M. B.. **Instalações elétricas**. 5. ed. São Paulo: Pearson, Prentice Hall, 2009.

DIAS, Diego Roberto Colombo et al. ChemCAVE3D: Sistema de Visualização Imersivo e Interativo de Moléculas 3D. **Anais do Wvrva'2011**: 8 Workshop de Realidade Virtual e Aumentada, Uberaba, Mg, n. , p.1-5, 07 nov. 2010.

DURLACH, N. IMAVOR, A. S. **Virtual Reality – Scientific and Technological Challenges**, National AcademyPress, Washington, 1995.



ELETROBRAS. **Importância da energia elétrica**. Disponível em: <<http://www.elektrobras.com/elb/natrilhadaenergia/energia-eletrica/main.asp?View={B1E5C97A-39C6-49BE-9B34-9BC51ECC124F}>>. Acesso em: 14 jul. 2012.

EXHIBITORS: **Virtual reality in manufacturing research and education**, August, 1997. (<http://www.ivri.me.uic.edu/symp96/preface.html>)

FERRARA, Carlos Alberto et al. EXPERIÊNCIA DE FURNAS NO DESENVOLVIMENTO E UTILIZAÇÃO DE UM SIMULADOR DE TREINAMENTO PARA OS CENTROS DE OPERAÇÃO. **Anais do Viii Simpase**: Simpósio de Automação de Sistemas Elétricos, Rio de Janeiro - Rj, n. , p.1-11, 09 ago. 2009.

GAMMA, Erich et al. **Padrões de projeto**: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 370 p.

GOLDSTONE, Will. **Unity Game Development Essentials: Build fully functional, professional 3D games with realistic environments, sound, dynamic effects, and more!**. Birmingham,: Packt Publishing Ltd., 2009. 316 p.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. (edição Digital): Artmed, 2009.

GRILO, L.; MONICE, S.; SANTOS, E. T.; MELHADO, S. Possibilidades de Aplicação e Limitações da Realidade Virtual na Arquitetura e na Construção Civil. **Anais do II Simpósio Brasileiro de Gestão da Qualidade do Trabalho no Ambiente Construído (SIBRAGEQ'2001)**, set. 2001, Fortaleza, CE (CD-ROM).

INFOWESTER. **Conhecendo o Servidor Apache (HTTP Server Project)**. Disponível em: <<http://www.infowester.com/servapach.php>>. Acesso em: 08 jun. 2012.

KALAWSKY, Roy S. (1993) **The Science of Virtual Reality and Virtual Environments**, Addison-Wesley, Reading, MA, 1993.

KING, Gavin et al. **Hibernate Reference Documentation**. Disponível em: <http://docs.jboss.org/hibernate/orm/3.5/reference/en-US/pdf/hibernate_reference.pdf>. Acesso em: 09 jun. 2012.

KIRNER, C. **Realidade Virtual e Aumentada**, Acesso em Janeiro 2012, Disponível em <<http://www.realidadevirtual.com.br>>.

KIRNER, Claudio; KIRNER, Tereza Gonçalves. **Evolução e Tendências da Realidade Virtual e da Realidade Aumentada. Xiii Simpósio De Realidade Virtual E Aumentada: Livro do Pré-Simpósio**, Uberlândia - Mg, n. , p.10-25, 23 maio 2011.

KIRNER, Claudio; SISCOOTTO, Robson Augusto. **Fundamentos de Realidade Virtual e Realidade Aumentada. Realidade Virtual e Aumentada: Uma Abordagem Tecnológica**, João Pessoa - Pb, n. , p.1-20, 13 maio 2008.

KIRNER, Claudio; SISCOOTTO, Robson Augusto. **Fundamentos de Realidade Virtual e Aumentada. In: Symposium On Virtual And Augmented Reality, 9.**, 2007, Petrópolis, Rj. Livro do pré-simpósio. Petrópolis, Rj: Sbc, 2007. v. 1, p. 2 - 21.

LEITE, Marco A. M.; SILVA, Fabio do Nascimento; RIBEIRO FILHO, Manoel. **Utilização de Realidade Virtual em Treinamento de Operadores e Mantenedores na Usina Hidrelétrica de Tucuruí. Anais do Iii Simpósio Brasileiro de Sistemas Elétricos**, Belém - Pr, n. , p.1-4, 18 maio 2010.

MAMEDE FILHO, Joao. **Instalações Elétricas Industriais**. 7. ed. Rio de Janeiro: Ltc, 2010.

MARÇAL, Edgar; ANDRADE, Rossana; RIO, Riverson. **Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual**. Disponível em: <<http://www.cin.ufpe.br/~cadcn/files/Pesquisas/IC%20-%20M-Learning/IC%20-%20Te%F3rica/IC/PIBIC/mLearning/Aprendizagem%20utilizando%20Dispositivos%20M%F3veis%20com%20Sistemas%20de%20Realidade%20Virtual.pdf>>. Acesso em: 14 jan. 2012.

MARTINS, Vânia. **Realidade Virtual em educação: Criando objetos de aprendizagem com VRML**. Colabor@: A Revista Digital da CVA-RICESU, Rio de Janeiro, v. 4, n. 15, p.1-11, 01 set. 2007. Semestral.

MICHAEL, D., CHEN, S., **Serious Games: Games That Educate, Train and Inform**. Course Technology PTR, 2005.

NETTO, Antonio Valerio; TAHARA, Creusa Sayuri; PORTO, Arthur J. Vieira. **Realidade Virtual e suas aplicações na área de manufatura, treinamento, simulação e desenvolvimento de produto**. Disponível em: <<http://www.scielo.br/pdf/gp/v5n2/a02v5n2.pdf>>. Acesso em: 12 maio 2012.

NUNES, Fátima L. S. et al. Desenvolvendo aplicações de RVA para saúde: imersão, realismo e motivação. **Xiii Simpósio De Realidade Virtual E Aumentada**: Livro do Pré-Simpósio, Uberlândia, Mg, n. , p.82-95, 23 maio 2011.

NUNES, Fátima L. S. et al. Desenvolvendo aplicações de RVA para saúde: imersão, realismo e motivação. **Xiii Simpósio De Realidade Virtual E Aumentad**: Livro do Pré-Simpósio, Uberlândia, Mg, n. , p.82-95, 23 maio 2011.

OLIVEIRA, Fábio Henrique M. et al. Uso de Realidade Aumentada na melhoria do processo de ensino-aprendizagem de motores elétricos. **Anais do Wrva'2010: 7 Workshop de Realidade Virtual e Aumentada**, São Paulo - Sp, n. , p.42-46, 08 nov. 2010.

OSA, David Daniel Pérez. **Interfaces 3D en los navegadores de internet**. Disponível em: <<http://david5.com/labs/pv3D/investigacion/interfaces3D.pdf>>. Acesso em: 09 abr. 2011.

PANTELIDIS V.S.; **Reasons to Use Virtual Reality in Education, VR in the Schools**, June 1995. Disponível em <<http://www.soe.ecu.edu/vr/vrits/1-1Pante.htm> >

PARREIRA, Rafael Tomaz et al. Sistema de Ensino de Física Óptica Geométrica da Reflexão em Espelhos Usando Realidade Virtual. **Anais do Wrva'2010: 7 Workshop de Realidade Virtual e Aumentada**, São Paulo - Sp, n. , p.213-217, 08 nov. 2010.

PASSOS, Erick Baptista et al. José Ricardo da Silva Jr.: Fernando Emiliano Cardoso Ribeiro. In: **BRAZILLIAN SYMPOSIUM ON GAMES AND DIGITAL ENTERTAINMENT**,8., 2009, Rio de Janeiro. Tutorials. Rio de Janeiro: Scb, 2009. v. 1, p. 1 - 30.



PAULA FILHO, Wilson de Pádua. **Engenharia de Software::** fundamentos, métodos e padrões (. Rio de Janeiro: Ltc, 2000.

PIMENTEL, K; TEIXEIRA, K. **Virtual Reality - through the new looking glass .2nd edition.**New York, McGraw-Hil, 1995.

RIZZATO, Andréia C.; NUNES, Fátima L. S..**Realidade Virtual aplicada à educação: reflexões sobre o estado da arte e o futuro.** Wrrva'07: Workshop de Realidade Virtual e Aumentada, Itumbiara, Mg, n. , p.81-84, 20 nov. 2007.

RODELLO, Ildeberto Aparecido; BREGA, José Remo Ferreira. Realidade Virtual e Aumentada em Ações de Marketing. **Realidade Virtual e Aumentada: Aplicações e Tendências: XIII SIMPÓSIO DE REALIDADE VIRTUAL E AUMENTADA**, Uberlândia, Mg, n. , p.44-57, 23 maio 2011.

RODELLO, Ildeberto Aparecido; BREGA, José Remo Ferreira. Realidade Virtual e Aumentada em Ações de Marketing. **Xiii Simpósio De Realidade Virtual E Aumentada: Livro do Pré-Simpósio**, Uberlândia-mg, n. , p.45-58, 23 maio 2011.

ROMAN, Fernando José. **Cave Desktop para jogos 3D em primeira pessoa.**Porto Alegre: Universidade Federal Do Rio Grande Do Sul, 2010.

ROSSI, Ronaldo. **SUBESTAÇÕES ELÉTRICAS DE ALTA TENSÃO: OPERAÇÃO E MANUTENÇÃO.** Itajubá: Itajubá, 2004.

RUSSO, E. E. R.; et al; 2004, "A Realidade Virtual na Indústria de Exploração e Produção de Petróleo". In: Realidade Virtual: Conceitos e Tendências – **Livro do Pré-Simpósio SVR 2004**, Cap. 21, p.283-288. Editora Mania de Livro, São Paulo, 2004

SILVA, A. L. da. **O USO DO PARADIGMA RESTRIÇÕES GEOMÉTRICAS PARA APOIO A PROJETOS DE INSTALAÇÕES ELÉTRICAS.** Uberlândia - Mg: Universidade De Uberlândia, 2006.

SILVA, Arquimedes Lopes da. **O uso do paradigma restrições geométricas para apoio a projetos de instalações elétricas.** Uberlândia - Mg: Universidade Federal De Uberlândia, 2006. 162 p.

SILVEIRA, Ismar Frango; BERGAMASCHI, Marcelo Pereira; SILVA, Luciano. **Aplicações de Realidade Virtual e Aumentada na Educação**. Uberaba, Mg: Ufu, 2011.

SILVESTRI, Filipe Ghesla. **Bibliotecas de Animação em Flash**. Disponível em: <http://www.filipesilvestrim.com/resources/public/filipe_silvestrim_artigo_bibliotecas_animacao_flash.pdf>. Acesso em: 09 abr. 2011.

SISCOUTTO, R. A.; KIRNER, C. **Fundamentos de Realidade Virtual e Realidade Aumentada. Realidade Virtual e Aumentada: Uma Abordagem Tecnológica**, João Pessoa – Pb, n. , p.1-20, 13 maio 2008

SOUZA, Everton Silva de; CARDOSO, Alexandre; LAMOUNIER, Edgard. Aplicações de RV no entretenimento. **Aplicações de Realidade Virtual e Aumentada: Livro do pré-simpósio, XI Symposium on Virtual and Augmented Reality**, Porto Alegre,, n. , p.90-107, 25 maio 2008.

TORI, R., KIRNER, C. **Fundamentos de Realidade Virtual**, Fundamentos e tecnologia de Realidade Virtual e Aumentada, pp. 9-28, maio 2006.

TORI, Romero et al. Jogos e Entretenimento com Realidade Virtual e Aumentada. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações: “Livro do pré-simpósio, IX Symposium on Virtual and Augmented Reality**, Porto Alegre, 2007., n. , p.192-222, 28 maio 2007.

TORI, Romero; KIRNER, Cláudio. Fundamentos de Realidade Virtual. **Fundamentos e Tecnologias de Realidade Virtual e Aumentada: Livro do pre-simpósio, VIII Symposium on Virtual Reality**, Belém - Pa, n. , p.2-21, 02 maio 2006.

VON SCHWEBER, L. e VON SCHWEBER, E. **Cover story: Realidade Virtual**, PC Magazine Brasil, pp. 50-73, v. 5, n. 6, 1995

WELLING, Luke; THOMSON, Laura. **PHP e MySQL, desenvolvimento web**. 6. ed. Rio de Janeiro: Elsevier, 2005. 754 p.

WILCOX, S.K. **Web developer's guide to 3D avatars**. New York: Wiley. 1998.



WINDER, Jeff; TONDEUR, Paul. **Papervision3D Essentials: Create interactive Papervision3D applications with stunning effects and powerful animations**. Birmingham: Packt Publishing Ltd., 2009. 428 p.

YOUNGBLUT C.; **Educational Uses of Virtual Reality Technology**. *VR in the Schools*. June 1997. Disponível em <<http://www.soe.ecu.edu/vr/vrits/3-1Young.htm>>

ZORZAL, Ezequiel Roberto et al. **O Uso da Realidade Virtual e Aumentada na Interação e Visualização de Informações em Redes de Computadores**. Disponível em: <<http://www.realidadeaumentada.com.br/artigos/24495.pdf>>. Acesso em: 12 maio 2012.



Apêndices

Apêndice 1 – Questionário aplicado

| Avaliação do Sistema Virtual Substation 1.0 / Unity 3D | |
|--|-----------------|
| Avaliador: | Data Avaliação: |
| () professor () aluno () engenheiro | |

Assinale, por favor, a opção que melhor traduz a sua opinião.

| 1. FUNCIONALIDADE - Evidência que o conjunto de funções atende às necessidades explícitas e implícitas para a finalidade que se destina o software (treinamento de operadores). | Muito Satisfeito | Satisfeito | Insatisfeito |
|--|------------------|------------|--------------|
| 1.1. O software auxilia no processo de aprendizagem na finalidade proposta? | | | |
| 1.2. Como avalia globalmente a funcionalidade do software. | | | |

| 2. USABILIDADE - Evidência a facilidade de utilização de software | Muito Satisfeito | Satisfeito | Insatisfeito |
|---|------------------|------------|--------------|
| 2.1. Em relação à facilidade de entender com funciona o programa | | | |
| 2.2. Quanto à navegação na cena e manipulação dos objetos (facilidade de uso) | | | |

| 3. CONFIABILIDADE - Evidência que o desempenho se mantém ao longo do tempo em condições estabelecidas. | Muito Satisfeito | Satisfeito | Insatisfeito |
|---|------------------|------------|--------------|
| 3.1. Capacidade de continuar a funcionar corretamente, após erros do próprio software ou erros de manipulação de dados. | | | |
| 3.2. Capacidade de o software enviar mensagens de erro caso os mesmos ocorram | | | |

| 4. EFICIÊNCIA - Evidência que os recursos e os tempos envolvidos são compatíveis com o tempo de desempenho requerido para o produto. | Muito Satisfeito | Satisfeito | Insatisfeito |
|---|------------------|------------|--------------|
| 4.1. Tempo de resposta às ações do usuário | | | |
| 4.2. Fidelidade a uma subestação e um transformador de força reais | | | |
| 4.3. Como avalia em termos globais a eficiência do software | | | |



| | | | |
|--|--|--|--|
| 5.6. Como avalia em termos globais a Portabilidade do software | | | |
|--|--|--|--|

| 5. Portabilidade - A capacidade do produto de software para ser transferido de um ambiente para o outro | Muito Satisfeito | Satisfeito | Insatisfeito |
|--|------------------|------------|--------------|
| 5.1. Facilidade de execução em diferentes ambientes (Executável Windows/Página web) | | | |
| 5.6. Como avalia em termos globais a Portabilidade do software | | | |

| 6. Sobre o Sistema – Metrifica o sistema em um contexto de treinamento | Muito Importante | Importante | Não é importante |
|---|------------------|------------|------------------|
| 6.1. A importância que atribui ao software para o aprendizado em geral. | | | |
| 6.2. A importância de treinar em Ambientes de Realidade Virtual | | | |
| 6.3. A importância de treinar em um simulador de transformador de força | | | |
| 6.4. A importância geral do sistema no treinamento de um operador do sistema | | | |

| |
|---------------------------|
| Comentários/ Observações: |
|---------------------------|

Este questionário foi elaborado com base na ISONORM 9126-1 e também considerou alguns aspectos de software educacional descritos por Campo e Campos (2001).