

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Unidade Acadêmica de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Dissertação de Mestrado

**Desenvolvimento de um módulo supervisor para um ambiente
de treinamento de operadores**

Flávio Torres Filho

Orientadora: Maria de Fátima Queiroz Vieira, Ph.D.

Campina Grande, Março de 2011.

Desenvolvimento de um módulo supervisor para um ambiente de treinamento de operadores

Flávio Torres Filho

Dissertação de mestrado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências no domínio da Engenharia Elétrica.

Área de Concentração
Engenharia da computação.

Orientadora
Maria de Fátima Queiroz Vieira, PhD.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

T693d Torres Filho, Flávio.

Desenvolvimento de um módulo supervisor para um ambiente de treinamento de operadores / Flávio Torres Filho. — Campina Grande, 2011. 99 f.: il. color.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Orientadora: Prof^a. PhD. Maria de Fátima Queiroz Vieira.

Referências.

1. Simuladores para Treinamento. 2. Sistemas Supervisórios. 3. Realidade Virtual. 4. Operação de Sistemas Elétricos. I. Título.

CDU – 681.5.017(043)

**DESENVOLVIMENTO DE UM MÓDULO SUPERVISÓRIO PARA UM AMBIENTE
DE TREINAMENTO DE OPERADORES**

FLÁVIO TORRES FILHO

Dissertação Aprovada em 30.03.2011

Maria de Fátima Q. Vieira
MARIA DE FÁTIMA QUEIROZ VIEIRA, Ph.D., UFCG
Orientador


ANGELO PERKUSICH, D.Sc., UFCG
Componente da Banca

Hyggo Oliveira de Almeida
HYGGÓ OLIVEIRA DE ALMEIDA, D.Sc., UFCG
Componente da Banca

CAMPINA GRANDE - PB
MARÇO -2011

Dedicatória

Aos meus pais.

Agradecimentos

Agradeço a Deus, pela oportunidade de realizar esse trabalho e pelo fortalecimento nos momentos difíceis.

Aos meus pais, Maria Ana da Conceição e Flávio Torres, e aos meus irmãos, Ítalo e Ianny, com os quais aprendi as mais valiosas lições de amor, coragem e determinação.

À minha professora orientadora, Maria de Fátima Queiroz Vieira, pela gentileza, generosidade e pela confiança depositada em mim na realização desse trabalho.

Aos meus companheiros e companheiras do LIHM pelas inestimáveis contribuições a esse trabalho, em especial, Ademar Netto, Raffael Carvalho, Daniel Cavalcante, Diego Lopes, Daniel Sherer, Alves Netto e Yuska Aguiar.

Aos grandes amigos e companheiros Manuel Pereira, André Moreira e Anderson Cavalcanti; e às amigas Fernanda Baracuy, Cecília Teresa, Taciana Rached e Vanessa Kelly, que me acompanharam nessa caminhada e sempre me incentivaram a prosseguir.

Muito obrigado a todos.

Resumo

Durante a operação do sistema elétrico, o operador está sujeito a intensa demanda por desempenho que pode desencadear decisões erradas. Neste contexto, o erro humano pode ter graves consequências ambientais, humanas e materiais, o que torna evidente a necessidade de intensificar a capacitação e o treinamento dos operadores. Atualmente são encontrados simuladores que apóiam o treinamento especificamente no uso de sistemas supervisórios. No entanto, para que seja mais realista e completo, o ambiente simulado deve abranger também o treinamento nos painéis de comando; assim como a operação integrada nestes dois níveis de controle (painéis e supervisório). Neste trabalho, teve-se como objetivo o desenvolvimento de um módulo capaz de integrar um software supervisório a um ambiente de simulação para treinamento de operadores. O projeto do simulador reconstitui, em realidade virtual, a sala de controle de uma subestação onde os operadores, em treinamento, podem interagir com painéis de controle, visualizar mostradores analógicos e digitais, ouvir o disparo de alarmes. Como resultado deste trabalho, o ambiente do simulador permite executar um software supervisório e operá-lo de forma integrada ao ambiente de simulação. Por integrado significa que as ações realizadas nos painéis repercutem no ambiente do supervisório e vice-versa, viabilizando um treinamento mais abrangente e facilitado a imersão do treinando nos cenários de treinamento. A inclusão do supervisório no ambiente simulado foi avaliada, do ponto de vista do usuário, a partir de um experimento no qual os usuários foram solicitados a utilizar o ambiente de treinamento e realizar tarefas utilizando tanto o supervisório quanto os painéis. Como resultado desta avaliação preliminar foi constatado que a funcionalidade do módulo supervisório atendeu aos principais requisitos propostos para o módulo.

Palavras chave: Simuladores para treinamento, Sistemas supervisórios, Realidade Virtual, Operação de sistemas elétricos.

Abstract

During the operation of an electrical system, the operator is subject to intense demand for performance, which in turn may lead to poor decisions. In this operational context, human error can have serious consequences (environmental, human and material), which highlight the need to intensify operator training. Currently the simulators focus specifically in the use of supervisory systems. However, to make the training more realistic and complete the simulated environment should also include the possibility of operating on control panels, as well; as an integrated operation on these two levels. Thus, this study aimed to integrate a supervisory software to the simulator developed for operator training. The simulator project offers a training environment, in virtual reality, that represents the control room of an electric substation, where the operators, can interact with control panels, display digital and analog displays, and listen to alarms. As a result of this work, the simulator project was extended to allow the interaction with supervisory software, and to support the training activity in an integrated environment. For integrated it means that the actions performed on the panels reflect in the supervisory software and vice versa, allowing a more comprehensive training and facilitated the immersion of the trainee in the scenarios. The integration of a supervisory software in the simulated environment was validated on an experiment during which a user performed tasks using both the supervisory software and the virtual reality representation of the substation panels.

Key Words: Supervisory Systems, Operator Training System, Virtual Reality.

Lista de abreviaturas

3D: Três dimensões

API: *Application Programming Interface*

AV: Ambiente Virtual

BD: Banco de Dados

CEPEL: Centro de Pesquisas de Energia Elétrica

CHESF: Companhia Hidro Elétrica do São Francisco

CPN: *Colored Petri Net*

GIHM: Grupo de Interface Homem Máquina

IP: *Internet Protocol*

ISO: *International Organization for Standardization*

JDBC: *Java DataBase Connectivity*

LIHM: Laboratório de Interface Homem-Máquina

MSC: *Message Sequence Charts*

RV: Realidade Virtual

SAGE: Sistema Aberto de Supervisão e Controle

SAI: *Scene Access Interface*

SCADA: *Supervisory Control And Data Aquisition*

SGBD: Sistema de Gerenciamento de Banco de Dados

SML: *Standard ML*

STPO: Simulador para Treinamento de Proteção e Operação

TCP: *Transmission Control Protocol*

UML: *Unified Modeling Language*

VRML: *Virtual Reality Modeling Language*

X3D: *eXtensible 3D*

XML: *eXtensible Markup Language*

Lista de Figuras

Figura 2-1 – Arquitetura geral do SIMULOP (LEITE; RODRIGUES; OLIVEIRA, 2007).....	23
Figura 2-2 – Editor de Cenários do sistema ASTRO (SILVA et al., 2009)	24
Figura 2-3 – Estrutura do Simulador de treinamento de FURNAS (FERRARA et al., 2009).	25
Figura 2-4 – Tela do diagrama unifilar do simulador de treinamento STPO (BEZERRA et al., 2007).....	26
Figura 3-1 – Sala de controle virtual de uma subestação elétrica, representada no ambiente do simulador	28
Figura 3-2 – Sala de controle da subestação Campina Grande II.....	29
Figura 3-3 – Painel de controle e supervisório representados em VRML, na versão inicial do simulador	30
Figura 3-4 – Arquitetura do simulador em desenvolvimento pelo LIHM.....	31
Figura 3-5 – Ambiente do operador no simulador para treinamento de operadores de sistemas elétricos.....	32
Figura 3-6 – Ambiente do tutor no simulador para treinamento de operadores de sistemas elétricos.....	33
Figura 3-7 – Comunicação entre modelos CPN, Supervisório e Painéis de Controle.....	35
Figura 4-1 – Digrama unifilar de uma subestação representado no supervisório SAGE (Fonte: Chesf)	39
Figura 4-2 – Exemplo de uma tela de sinótico com informações mais detalhadas sobre uma linha de transmissão (Fonte: CHESF)	40
Figura 4-3 – Exemplo de uma tela de alarmes e eventos (Fonte: SAGE/Cepel).....	41
Figura 4-4 – Software para edição de telas do SAGE (Fonte: SAGE/Cepel)	42
Figura 4-5 – Diagrama de casos de uso no módulo supervisório	44
Figura 4-6 – Diagrama de sequência para a execução do comando de abertura de um disjuntor através do supervisório	46
Figura 4-7 – Estação de trabalho do operador representada na sala de controle virtual	47
Figura 4-8 – Tela de Login do supervisório	48
Figura 4-9 – Tela de acesso aos submódulos do supervisório.....	48
Figura 4-10 – Tela com o diagrama unifilar da subestação, implementada no <i>InTouch</i>	49
Figura 4-11 – Janela <i>pop-up</i> para o controle do disjuntor 14D1, implementada no <i>InTouch</i> ..	50
Figura 4-12 – Tela de alarmes no supervisório do simulador, implementada no <i>InTouch</i>	51
Figura 4-13 – Representação do ciclo de execução do método run() da classe <i>supervisórioTutor.java</i>	53
Figura 4-14 – Representação do ciclo de execução do método run() da classe <i>supervisórioTreinando.java</i>	54
Figura 4-15 – classe <i>ddeDriver.java</i>	56
Figura 5-1 – Representação da linha de transmissão 04V2 com os dispositivos no estado inicial do cenário	60
Figura 5-2 – Representação da linha de transmissão 04V2 após a realização da manobra de liberação do disjuntor 14V2	62

Lista de Tabelas

Tabela 5-1 – Papel desempenhado pelos usuários em cada sessão de teste	58
Tabela 5-2 – Plano de coleta de dados e critérios de avaliação.....	59
Tabela 5-3 – Dispositivos a serem configurados através do supervisor.....	60

Sumário

Capítulo 1	<i>Introdução</i>	13
1.1	Motivação	14
1.2	Objetivos.....	16
1.3	Metodologia	17
1.4	Estrutura da Dissertação.....	18
Capítulo 2	<i>O Treinamento de operadores em subestações elétricas</i>	19
2.1	Métodos para o treinamento de operadores.....	19
2.2	Simuladores para apoio ao treinamento.....	20
2.2.1	Principais características e requisitos fundamentais.....	21
2.2.2	Exemplos de simuladores.....	22
Capítulo 3	<i>Ambiente Virtual para Treinamento de Operadores do Setor Elétrico</i>	27
3.1	Sala de controle virtual de uma subestação elétrica.....	27
3.2	Arquitetura do simulador.....	30
3.2.1	Ambiente do operador.....	32
3.2.2	Ambiente do tutor.....	32
3.2.3	Banco de Dados.....	34
3.2.4	Motor de simulação baseado em modelos.....	34
3.2.4.1	Modelo dos painéis.....	36
3.2.4.2	Modelo do Campo.....	37
Capítulo 4	<i>Sistema supervisorio no ambiente do simulador</i>	38
4.1	Características de sistemas supervisorios em subestações elétricas.....	38
4.2	Análise dos requisitos para o módulo supervisorio do simulador.....	42
4.3	Especificação e implementação do módulo supervisorio.....	44
4.3.1	Implementação da interface com o operador e com o tutor.....	47

4.3.2	Estratégia de integração com os demais módulos da arquitetura	51
Capítulo 5	<i>Avaliação funcional do módulo supervisor do simulador</i>	57
5.1	Planejamento do experimento	57
5.1.1	Critérios de avaliação	59
5.2	Apresentação das tarefas de teste.....	59
5.3	Análise dos resultados	62
Capítulo 6	<i>Considerações finais.....</i>	65
	Trabalhos futuros	68
	<i>Referências.....</i>	69
	<i>Apêndice A: Ferramentas de desenvolvimento para projeto do simulador.....</i>	73
	<i>Apêndice B: Modelo CPN do campo.....</i>	76
	<i>Apêndice C: Planejamento e artefatos do teste de validação</i>	82
	<i>Apêndice D: Classes que gerenciam a troca de informações com o supervisor</i>	90

Capítulo 1 Introdução

Durante o trabalho de supervisão e controle de sistemas elétricos um grande número de informações é disponibilizado aos operadores, os quais precisam responder de forma rápida e correta aos eventos, aos quais o sistema é submetido.

Além disso, são frequentes as mudanças na configuração do sistema, por vezes considerável, através da adição de novos equipamentos e da expansão de determinadas áreas. Também é comum a troca de operadores em subestações elétricas, que são realocados em função das necessidades da companhia de energia. Situações como estas dificultam o entendimento das características do sistema e do comportamento da rede pelos operadores.

Por outro lado, o erro humano durante a operação de sistemas elétricos pode ter sérias implicações, com perdas materiais, ambientais, ou mesmo de vidas humanas. E, em recente estudo de relatórios de falhas do sistema na empresa CHESF – Companhia Hidro Elétrica do São Francisco, realizado por NASCIMENTO NETO (2010), foi verificado que 20% das falhas resultam de erro humano.

Assim, dada a complexidade das tarefas a serem realizadas, e os graves acidentes que o erro humano pode causar, o treinamento eficiente e eficaz desses profissionais é essencial.

Neste contexto, um simulador pode desempenhar um papel importante, recriando uma variedade de cenários de treinamento. Pressupõe-se que caso o operador já tenha passado por uma situação, ainda que simulada, em tese, ele estará mais preparado para enfrentar uma situação real. Também durante uma simulação, o operador em treinamento poderá se sentir à vontade para explorar as funcionalidades do sistema de supervisão e controle, e verificar as implicações de determinadas decisões no comportamento do sistema, algo evidentemente impensável na operação de um sistema elétrico real.

Os cenários de treinamento podem consistir em situações de rotina, com o sistema elétrico operando em condições normais, e em situações de exceção ou críticas, a exemplo de ocorrências que demandam a recomposição do sistema. Dessa forma, a capacitação do profissional deve ser abrangente e os simuladores são úteis tanto na qualificação de operadores iniciantes como na reciclagem de operadores experientes.

Além disso, o uso de simuladores elimina os riscos associados à operação do sistema real durante a realização de um treinamento, evitando que um profissional, ainda em treinamento, seja exposto ao risco ou possa comprometer a integridade do sistema.

Este trabalho está inserido no contexto de desenvolvimento de um simulador no Laboratório de Interfaces Homem-Máquina (LIHM). Este simulador apoiará o treinamento de operadores de subestações elétricas, e o estudo do erro humano na operação destes sistemas.

O ambiente de simulação reproduz, em realidade virtual, a sala de controle de uma subestação elétrica, com os níveis de operação via supervisor e via painéis de controle. Imersos em uma réplica virtual do ambiente de trabalho, os operadores podem exercitar e demonstrar o domínio sobre os procedimentos e habilidade necessários à operação do sistema.

Enquanto ferramenta de apoio ao estudo do erro humano o simulador disponibiliza um ambiente controlado no qual é possível evidenciar a relação entre o erro e o projeto da interface do usuário com o sistema, visando a concepção de interfaces mais ergonômicas.

Na sala de controle de uma subestação elétrica, os operadores interagem com sistemas supervisórios e com painéis de controle para a operação do sistema elétrico de potência. A representação de painéis de controle no ambiente virtual do simulador, e a comunicação com o motor de simulação do sistema, foram exploradas respectivamente nos trabalhos (NASCIMENTO NETO, 2010) e (SILVA NETTO, 2010).

Nesse trabalho, será focalizada a representação do **sistema supervisor** nesse ambiente virtual. Dado o crescente nível de automação e informatização nas salas de controle em subestações elétricas é necessário oferecer treinamento no uso dos sistemas supervisórios. Este tipo de treinamento já é oferecido pela maioria dos simuladores, porém o supervisor, representado nesses simuladores, não é inserido em uma representação virtual do ambiente de trabalho, nem associado a uma representação de painéis, como aqui proposto.

1.1 Motivação

Os centros de supervisão e controle de energia elétrica, assim como ocorreu em outras áreas da indústria na última década, evoluíram da operação através de painéis de controle distribuídos ao longo do sistema de produção para os sistemas supervisórios, que coletam dados do processo, formatam esses dados e os disponibilizam em terminais de computador (MAMEDE FILHO, 2007).

Na CHESF, por exemplo, assim como ocorreu em outras empresas geradoras e transmissoras de energia elétrica, novas subestações foram projetadas para serem totalmente

gerenciadas remotamente por operadores interagindo com sistemas supervisórios em centros de operação. Subestações mais antigas, que possuem o controle convencional, também estão sendo modernizadas e a introdução de software supervisórios nesses ambientes é cada vez mais comum (SANTOS; CARMO, 2005).

As tarefas realizadas pelos operadores ainda são essencialmente as mesmas, porém, através da interação com supervisórios, a forma como são realizadas foi modificada. Neste cenário, torna-se clara a necessidade de se oferecer capacitação dos profissionais no uso desses sistemas.

Como será apresentado no capítulo seguinte, muitas empresas do setor elétrico já possuem simuladores com o propósito de treinamento. Entretanto, nesses simuladores, o supervisório não está associado a uma representação do ambiente da subestação, nem aos painéis de controle. Isso significa que, numa situação de treinamento, ao atuar apenas no supervisório, o operador não pode perceber os alarmes sonoros e visuais que são disparados nos painéis, nem tão pouco interagir com eles.

Por outro lado, na rotina de operação de uma subestação elétrica, em grande parte das tarefas que são realizadas, os operadores precisam executar alguns comandos no supervisório, e outros nos painéis. E isso é necessário porque nem todos os equipamentos da subestação estão preparados para serem operados remotamente (SANTOS; CARMO, 2005). Também é possível que ocorra falha na comunicação do supervisório com o dispositivo, sendo necessário que o operador se desloque até o painel de controle, ou mesmo até o equipamento para operá-lo diretamente. Portanto, o treinamento oferecido pela maioria dos simuladores é incompleto e não atende a todas as necessidades de capacitação dos operadores.

Nesse sentido, o simulador desenvolvido no LIHM pretende oferecer treinamento aos operadores de sistemas elétricos nos dois níveis de controle, via supervisório e via painéis. Em um ambiente tridimensional, inspirado em uma típica sala de controle de subestação elétrica, o operador pode se deslocar e interagir com representações virtuais dos painéis de controle (VIEIRA et al, 2010). E, em um trabalho anterior, desenvolvido por Freitas (2006), foi esboçada uma representação do ambiente supervisório no mundo virtual, porém tratava-se apenas de uma prova de conceito que precisava ser explorada para os propósitos do ambiente do simulador.

Esse trabalho é então motivado pela necessidade de disponibilizar a representação adequada de um sistema supervisório ao projeto de simulador desenvolvido no LIHM, associando-o ao ambiente virtual, ao motor de simulação, e ao banco de dados do sistema.

Dessa forma, o operador poderá se aproximar mais da realidade de operação de uma subestação elétrica, ora atuando em painéis de controle, ora no supervísório. E poderá perceber a repercussão das ações tomadas de um nível de controle no outro.

1.2 Objetivos

Nesse trabalho, o principal objetivo é o desenvolvimento de um ambiente supervísório, com as características dos sistemas supervísórios usados na operação de subestações elétricas, e a sua integração ao simulador para treinamento de operadores, desenvolvido no LIHM.

Dessa forma, os usuários do simulador, instrutores e treinandos, poderão participar de sessões de treinamento em que devem monitorar e controlar uma subestação elétrica, representada em um mundo virtual, em dois níveis de controle diferentes (painéis e supervísório). Diferenciando-se da grande maioria dos sistemas de apoio ao treinamento de operadores, que permitem a simulação de manobras apenas no nível de controle supervísório. Quando, no entanto, na rotina de operação de uma subestação elétrica, os operadores precisam também atuar sobre os painéis de controle. Especialmente nas manobras mais complexas, o operador deve realizar um conjunto de ações no supervísório e outro nos painéis, sendo frequente e necessário o deslocamento entre os dois níveis de controle.

Para atingir este objetivo geral, foram definidos os seguintes objetivos específicos:

- Reproduzir em um sistema supervísório comercial as telas do ambiente supervísório com o qual o operador interage no seu ambiente de trabalho.
- Associar o ambiente de supervisão desenvolvido ao banco de dados do simulador, permitindo representar cenários de treinamento armazenados pelo tutor e a atualização dos dados a partir das ações dos usuários durante a simulação.
- Desenvolver um modelo simplificado da planta industrial e integrá-lo com ao modelo que representa o comportamento dos objetos nos painéis de controle. O motor de simulação do sistema baseia-se na execução desses modelos.
- Permitir que os comandos efetuados pelos usuários no supervísório sejam executados no motor de simulação do sistema.

- Integrar a representação virtual dos painéis com o ambiente supervísório, de tal modo que a interação em qualquer um dos níveis (painel ou supervísório) repercute sobre o estado da chave correspondente no outro nível de controle.
- Permitir que o supervísório seja usado, durante uma sessão de treinamento, por diferentes usuários (tutor e treinandos) e que os comandos efetuados por um dos usuários possa ser visualizado pelos demais.

Como objetivo secundário, pretende-se oferecer aos usuários do simulador uma representação o mais fiel possível àquela do ambiente real, facilitando a imersão do operador no contexto de trabalho, durante a realização das sessões de treinamento.

1.3 Metodologia

Para a execução deste trabalho foi adotada a seguinte metodologia:

1. Compor o motor de simulação a partir de modelos CPN:
 - 1.1. Construir um modelo simplificado da planta industrial;
 - 1.2. Realizar a comunicação entre os modelos de interface e o modelo da planta industrial.
2. Estudar temas relacionados ao projeto do simulador
 - 2.1. Simuladores para Treinamento de operadores;
 - 2.2. Realidade Virtual: Formalismos para construção de mundos virtuais e ferramentas de software usadas no projeto do simulador;
 - 2.3. Sistemas supervísórios: Principais características, recursos para concepção de telas, conexão com banco de dados.
3. Propor e analisar estratégias para representação de um sistema supervísório no ambiente virtual do simulador;
4. Selecionar e implementar a estratégia de representação do supervísório no ambiente do simulador:
 - 4.1. Conceber as telas do supervísório para um cenário de treinamento;
 - 4.2. Integrar o supervísório ao simulador (Conectar com o mundo virtual e com o Banco de dados).
5. Validar o uso do supervísório no ambiente do simulador através de um teste com usuários, durante o qual será executado um roteiro de manobras.

1.4 Estrutura da Dissertação

Este trabalho está estruturado da seguinte forma. No Capítulo 2 é apresentada uma breve revisão de projetos de simuladores para treinamento de operadores de sistemas elétricos, ressaltando suas características relevantes para o trabalho proposto. A seguir, no Capítulo 3, é apresentado o projeto do simulador atualmente em desenvolvimento no LIHM – Laboratório de Interfaces Homem-Máquina. No Capítulo 4, são apresentadas as características de sistemas supervisórios voltados para a operação de sistemas elétricos, os requisitos considerados, a especificação e a implementação do sistema supervisório que foi integrado aos demais módulos do simulador. Para validação do trabalho realizado, no Capítulo 5, é descrito o teste realizado com usuários e os dados coletados são analisados. Finalmente, no Capítulo 6, são apresentadas as considerações finais e sugestões de trabalhos futuros.

Capítulo 2 O Treinamento de operadores em subestações elétricas

Neste Capítulo, apresentam-se os métodos mais comuns na formação, reciclagem e certificação de operadores de sistemas elétricos de potência. E destaca-se o uso de *software* de simulação por empresas do setor elétrico na realização de treinamentos, com exemplos de aplicações e algumas das suas principais características.

2.1 Métodos para o treinamento de operadores

No contexto desta dissertação, entende-se por *treinamento* a capacitação de pessoas, ou o desenvolvimento de suas habilidades, para realizarem um determinado trabalho, em particular, a operação de sistemas elétricos.

Com algumas variações, os métodos usados pelas empresas do setor elétrico para o treinamento de novos operadores são muito semelhantes (MOREALE, 2007). Embora, a descrição apresentada nessa seção tenha como referência informações obtidas junto a operadores da CHESF.

Ao ingressarem na empresa, os novos operadores iniciam um processo de formação com aulas teóricas, aprendizagem dos procedimentos operacionais, visitas técnicas às subestações da empresa, e estudo de ocorrências anteriores. Essa etapa busca transmitir aos profissionais as informações minimamente necessárias para iniciarem suas atividades.

Na etapa seguinte, o operador iniciante passa a vivenciar a rotina da subestação, normalmente supervisionado por um operador mais experiente. A prática é obtida operando o sistema real, a começar pelas manobras mais simples e frequentes. O operador supervisor instrui e acompanha o iniciante na realização dessas manobras. E, na medida em que adquirem experiência e domínio dos procedimentos operacionais, os iniciantes ganham maior autonomia e podem realizar manobras mais complexas (SILVA et al., 2009).

Esse processo é incompleto, deixando algumas lacunas na formação do operador, uma vez que o profissional pode ter anos de experiência sem nunca ter se deparado com uma situação de risco e emergência (LEITE; RODRIGUES; OLIVEIRA, 2007). E, portanto, pode

não estar preparado para lidar com o *stress* e, em pouco tempo, ser capaz de usar o conhecimento que possui para enfrentar as particularidades da ocorrência.

Nesse sentido, é comum a dramatização de ocorrências para treinamento e reciclagem dos operadores (MIRANDA; VIEIRA; OLIVEIRA, 2008). Esse método, também usado na formação de bombeiros e salva-vidas, consiste na simulação de uma ocorrência, em que os operadores falam e agem com se estivesse enfrentando uma situação real, devem identificar a razão do problema e indicar o que fariam para resolvê-lo, ainda que não atuem efetivamente sobre os equipamentos. O objetivo é manter os operadores bem treinados, submetê-los a situações pouco frequentes, apesar de possíveis, e avaliar a capacidade individual para solucionar os problemas.

Além disso, são realizadas reuniões periódicas dos operadores com a equipe de engenheiros responsáveis pela subestação, onde são apresentadas e discutidas as últimas ocorrências, as não conformidades observadas naquele período, e eventuais modificações nos procedimentos operacionais. O objetivo é manter os operadores sempre atualizados e evitar que falhas humanas possam se repetir.

2.2 Simuladores para apoio ao treinamento

Dada a crescente preocupação das empresas do setor elétrico com a capacitação de seus operadores, ferramentas específicas para treinamento vêm sendo desenvolvidas, sobretudo simuladores (SILVA et al., 2009). Durante sessões de treinamento utilizando simuladores, os usuários têm a oportunidade de enfrentar diferentes problemas e aplicar seus conhecimentos interagindo com interfaces que reproduzem aquelas normalmente usadas na operação do sistema real.

A utilização desses *softwares* de simulação facilita assimilação dos procedimentos de operação e permite que os treinandos possam praticar com uma réplica do sistema sem a preocupação de interferir no sistema real. Ou seja, a segurança do operador e do sistema ficam resguardadas, e o fornecimento de energia elétrica aos consumidores não é interrompido. Ao mesmo tempo, o operador se familiariza com a prática de operação e adquire maior autoconfiança.

Aliado às aulas teóricas e ao estudo de ocorrências anteriores, os operadores podem demonstrar o aprendizado em simulações que reconstituem as situações críticas ou de emergência. Apesar de pouco frequentes, esses casos demandam do operador a capacidade de tomar as decisões corretas e rápidas para trazer sistema às condições normais de operação.

Assim, os simuladores possibilitam um treinamento mais abrangente, os operadores ganham maior experiência e estarão mais preparados para enfrentar situações reais.

Nas subseções a seguir, analisa-se algumas das características e requisitos desejáveis para simuladores de apoio ao treinamento de operadores de sistemas elétricos e destaca-se algumas aplicações já usadas por empresas nacionais e internacionais.

2.2.1 Principais características e requisitos fundamentais

Para atingir os objetivos do treinamento em um software de simulação, é imprescindível convencer o operador da verossimilhança das situações simuladas. Ou seja, o operador deve estar seguro de que o modelo que representa o comportamento e a interface dos objetos no simulador são compatíveis com aqueles encontrados no ambiente real. Isso permitirá que o treinando possa ser mais facilmente envolvido na situação simulada e posteriormente seja capaz de reproduzir no ambiente real, o aprendizado adquirido no treinamento (SILVA *et al.*, 2009).

Nesse sentido, o simulador deve oferecer os recursos que o operador teria disponível para realizar suas tarefas durante ocorrências reais, como a operação sobre botões em painéis e equipamentos, visualização de mostradores analógicos e digitais, visualização de telas de um sistema supervisório, consulta a manuais, comunicação com centros de operação, etc.

Também é necessário disponibilizar recursos para o instrutor criar uma variedade de cenários de treinamento com diferentes propósitos (instrução, certificação e reciclagem), de acordo com as necessidades de competências do ambiente de operação (LEITE; RODRIGUES; OLIVEIRA, 2007). Os novos operadores, por exemplo, podem ser treinados em situações de rotina, nas quais são realizadas manobras frequentes, e também em situações nas quais é necessário realizar manobras complexas e raras, como recompor e normalizar o sistema elétrico após desligamentos imprevistos.

Por outro lado, operadores experientes podem se beneficiar de simuladores para exercitar manobras pouco frequentes, ou no treinamento de novos procedimentos resultantes de modificações no sistema elétrico. A empresa também pode adotar o simulador no processo de certificação de seus operadores para atuar na operação da planta (OLIVEIRA, *et al.*, 2005).

Uma vez que concessionárias e usinas geradoras de energia normalmente possuem subestações elétricas geograficamente distantes, um requisito importante para um simulador é permitir a realização de treinamentos à distância (BEZERRA *et al.*, 2007). Com isso, evita-se

que operadores e instrutores se ausentem de seus postos de trabalho para participar de treinamentos em outras localidades.

É também um requisito importante permitir que as sessões de treinamento possam ser assistidas por um instrutor em tempo real, inclusive podendo disparar eventos durante a simulação. Por outro lado, é importante também oferecer o treinamento isolado, respeitando a conveniência do operador e o seu ritmo de trabalho individual.

Tendo realizado o treinamento, o software de simulação deve oferecer meios para a análise e qualificação do desempenho do operador no cenário simulado. Do ponto de vista dos operadores, essas informações permitirão corrigir falhas, tirar dúvidas e melhorar a qualidade de seu desempenho. Para o tutor, estas informações permitirão avaliar o progresso de cada treinando, e apoiar a criação de novos cenários de treinamento.

2.2.2 Exemplos de simuladores

Várias empresas do setor elétrico, no Brasil e no exterior, utilizam simuladores no processo de treinamento e certificação de operadores. Dentre estas estão: Operador Nacional do Sistema – ONS (LORENZO et al., 2008), ELETRONORTE (MOUTINHO, 2008), ITAIPU Binacional (ISASI et al., 2007), Swedish National Grid (WALVE; ADSTROM, 1999), Electricité de France – EDF (KOSTIC et al., 1997) e na China Henan Electric Power (HUA; ZHOU; YU, 2004).

Em sua maioria, os simuladores são voltados para o treinamento no ambiente do software supervisor. Neles, os operadores em treinamento interagem com uma representação do sistema supervisor, que simula a operação real do sistema elétrico de potência. A seguir, apresenta-se uma descrição sucinta de alguns destes simuladores.

O SIMULOP (LEITE; RODRIGUES; OLIVEIRA, 2007) foi desenvolvido pelo CEPEL e usado na CHESF (Companhia Hidro Elétrica do Rio São Francisco) para treinamento e certificação de operadores. Este sistema é resultado da integração de uma ferramenta EMS (Energy Management System) para supervisão e controle de sistemas elétricos de potência, o SAGE (Sistema Aberto de Gerenciamento de Energia), com um simulador digital de redes elétricas, o OTS (Operator Training Simulator) desenvolvido pelo EPRI (Electric Power Research Institute).

Para o processo de integração entre o sistema supervisor SAGE e o simulador OTS foi necessário o desenvolvimento de: filtros estáticos, filtros dinâmicos, protocolos de comunicação, telas especiais, dentre outros módulos.

Na Figura 2-1, é representada a arquitetura geral do sistema SIMULOP, onde são ilustrados seus principais componentes:

- Base SAGE – base de dados distribuída do sistema, em tempo real;
- EPRI-API – biblioteca de rotinas com interfaces para acesso à base de dados do simulador;
- Simulador OTS – simulador digital em tempo real de sistemas elétricos de potência;
- Filtro Estático – aplicativo que faz a transferência dos dados cadastrais da base de dados do sistema SAGE para a base de dados do simulador OTS;
- Filtro Dinâmico – aplicativo que faz a conexão, em tempo real, dos pontos de comando e supervisão do sistema SAGE ao sistema elétrico representado no simulador.

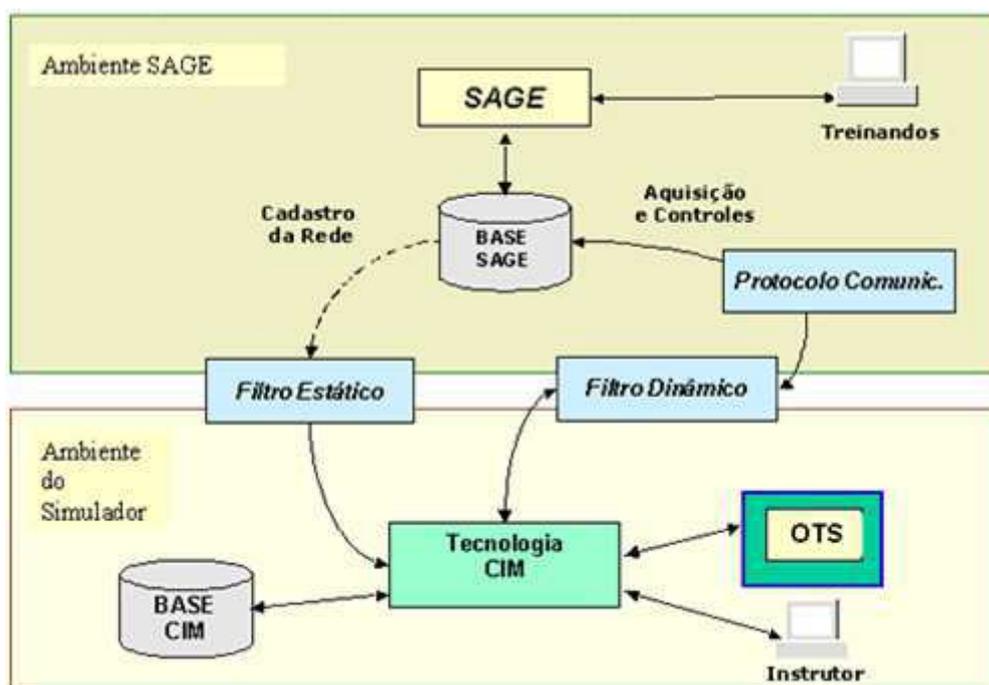


Figura 2-1 – Arquitetura geral do SIMULOP (LEITE; RODRIGUES; OLIVEIRA, 2007)

Outro exemplo de simulador é o ASTRO (Ambiente Simulado para Treinamento de Operadores) desenvolvido pelo CEPEL, em parceria com a ELETROSUL, para funcionar de forma integrada ao SAGE (SILVA et al., 2009). O ASTRO é composto por três módulos principais:

- Editor de cenários – permite que os instrutores possam criar situações de treinamento (Figura 2-2), definindo um estado inicial do sistema e possíveis caminhos dentro de uma simulação;

- Programa de simulação da operação – funciona de forma integrada ao SAGE, permitindo que o operador execute a simulação como se estivesse enfrentando uma situação real;
- Analisador de desempenho – permite ao instrutor estabelecer critérios para análise e qualificação de uma simulação realizada, fornecendo ao operador uma avaliação do seu desempenho.

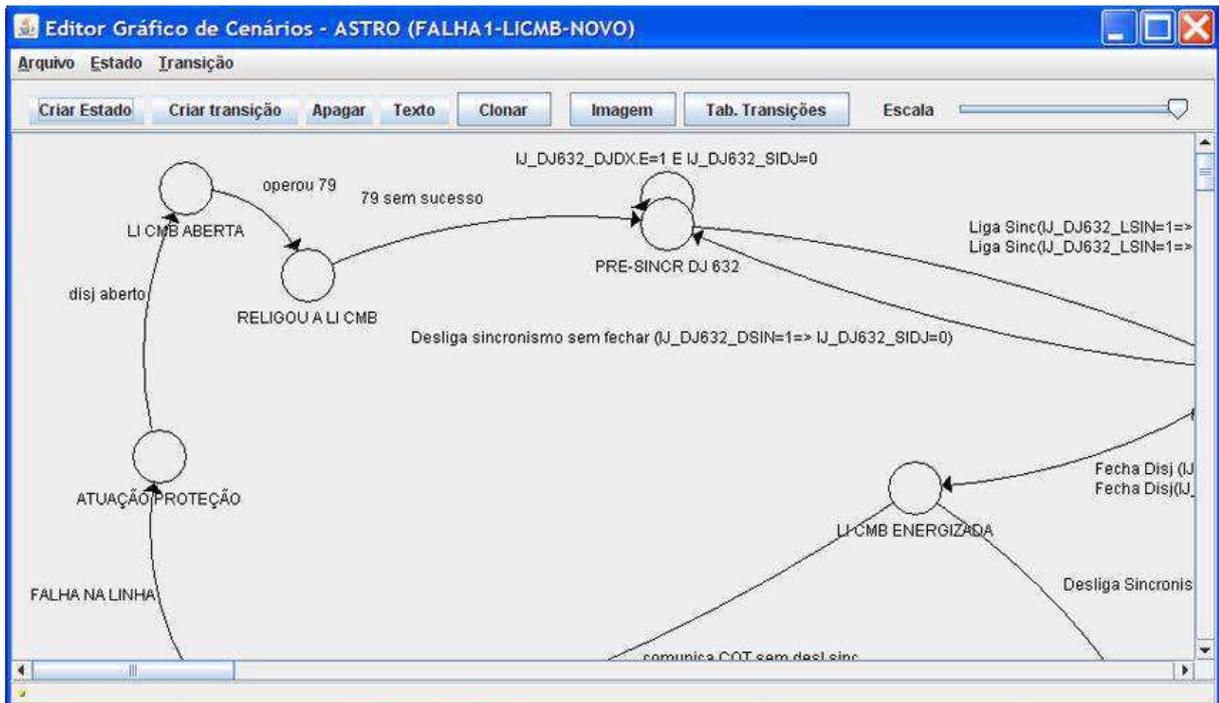


Figura 2-2 – Editor de Cenários do sistema ASTRO (SILVA et al., 2009)

Esse sistema não possui um simulador de redes elétricas embutido, cabe ao responsável pela criação do cenário atribuir os valores assumidos pelos pontos do SAGE após a ação de um operador. Esses valores podem ser levantados a partir de dados históricos, estudos, ou simulações, caso o instrutor tenha acesso a um simulador de redes elétricas.

De forma semelhante ao SIMULOP e ao ASTRO, o simulador para treinamento desenvolvido por Furnas Centrais Elétricas foi integrado ao sistema de supervisão e controle usado nessa empresa – o sistema SOL (FERRARA et al., 2009). Esse simulador também permite a criação de cenários a partir de dados históricos do sistema, para estabelecer grandezas analógicas e digitais em determinados pontos. A arquitetura desse simulador está representada na Figura 2-3, a seguir.

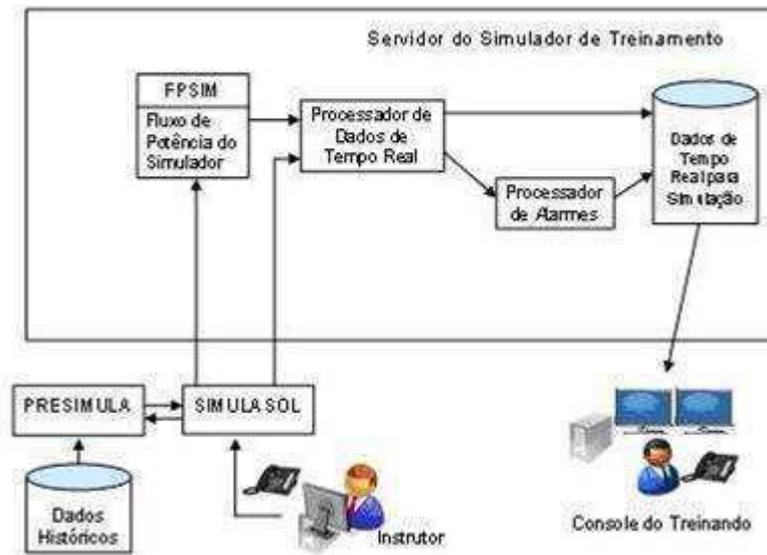


Figura 2-3 – Estrutura do Simulador de treinamento de FURNAS (FERRARA et al., 2009)

O simulador de Furnas é composto pelos seguintes módulos:

- SIMULASOL – programa que permite ao instrutor criar um cenário de treinamento, baseando-se em dados históricos que foram armazenados pelo sistema SOL;
- PRESIMULA – programa que prepara o ambiente para a simulação. É acionado pelo programa SIMULASOL para restabelecer os dados históricos de data e hora;
- FPSIM – programa que permite o cálculo de tensões, correntes e fluxos de potência ativa e reativa após alterações na topologia do sistema elétrico para concepções de diferentes cenários de treinamento.

Há ainda o sistema STPO (Simulador para Treinamento de Proteção e Operação de Sistemas Elétricos) desenvolvido na linguagem DELPHI para representar uma subestação da COELCE. Nele é possível simular falhas e reconfigurar o sistema elétrico (BEZERRA et al., 2007).

No STPO é apresentado um diagrama unifilar simplificado de uma subestação padrão COELCE de 69-13,8kV com seus alimentadores e dispositivos de proteção e manobra (Figura 2-4). Esse simulador foi também integrado a um ambiente virtual de ensino-aprendizagem para permitir o treinamento à distância.

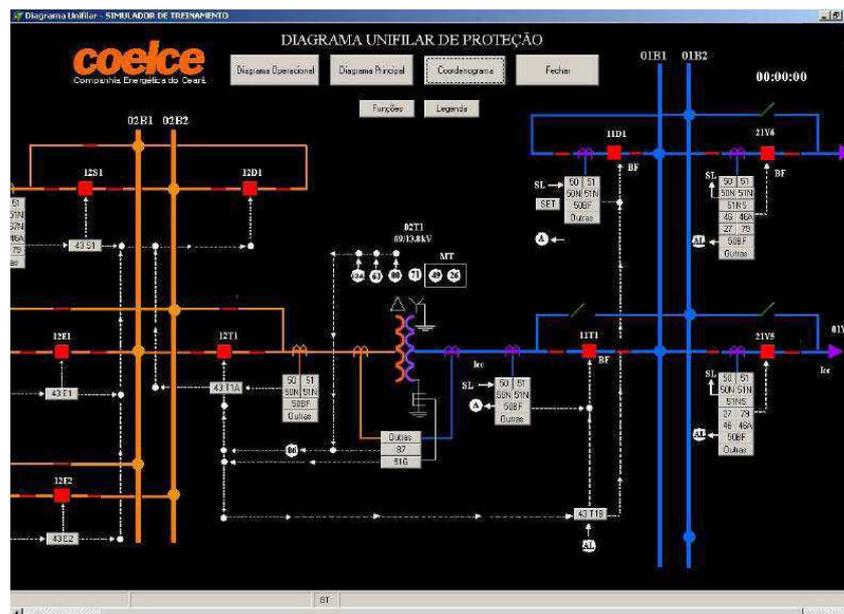


Figura 2-4 – Tela do diagrama unifilar do simulador de treinamento STPO (BEZERRA et al., 2007)

Observa-se uma preocupação das empresas do setor elétrico em treinar os operadores no uso do sistema supervisor. Nos projetos citados, os operadores em treinamento interagem com uma representação do sistema supervisor, simulando a operação real do sistema elétrico de potência.

Por outro lado, o simulador desenvolvido no LIHM permite o treinamento de operadores na interação com painéis de controle da subestação, mas oferecia uma representação rudimentar e ainda limitada do sistema supervisor. Nesse trabalho o módulo supervisor foi expandido e sua representação aprimorada, permitindo o treinamento nos dois níveis de operação do sistema elétrico de forma satisfatória.

Além disso, contrastando com os trabalhos citados, cujos motores de simulação foram construídos com base em linguagens de programação, o motor de simulação do sistema em desenvolvido no LIHM, apresentado no capítulo seguinte, foi construído a partir de modelos formais, os quais representam o comportamento de alguns dos objetos do ambiente simulado.

Capítulo 3 Ambiente Virtual para Treinamento de Operadores do Setor Elétrico

No presente capítulo, o projeto do simulador em realidade virtual desenvolvido no LIHM para o treinamento de operadores de sistemas elétricos é apresentado.

O sistema aqui descrito tem dois propósitos principais. O primeiro visa desenvolver um ambiente controlado que permita o estudo do comportamento do operador quando submetido a situações críticas ou de emergência e a relação do erro humano com o projeto da interface do usuário com o sistema. Comprovada e identificada essa relação, podem-se adotar estratégias para prevenção do erro e propor diretrizes de projeto na concepção de interfaces mais ergonômicas. O segundo objetivo, como mencionado anteriormente, é disponibilizar uma infraestrutura para treinamento de operadores de sistemas elétricos.

Além dos operadores que serão treinados na realização de manobras no sistema elétrico, o simulador também permite que usuários tutores possam criar, salvar, selecionar e editar cenários de simulação, assim como acompanhar a execução do cenário em tempo real e disparar eventos durante o treinamento.

O ambiente de simulação e os módulos que compõem a arquitetura da versão atual do simulador serão discutidos, ressaltando as tecnologias utilizadas e as ferramentas de desenvolvimento que foram adotadas.

Em especial, o módulo que simula um software supervisor para o controle do sistema elétrico é introduzido nesse capítulo, destacando-se sua relação com os demais módulos do sistema. Entretanto, a especificação e o processo de desenvolvimento desse módulo serão detalhados no capítulo seguinte.

3.1 Sala de controle virtual de uma subestação elétrica

O simulador desenvolvido no LIHM para o treinamento de operadores de sistemas elétricos oferece aos usuários um ambiente de simulação que reproduz, em realidade virtual, uma típica sala de controle de uma subestação distribuidora de energia elétrica.

Nesse ambiente virtual tridimensional (3D), os operadores, em treinamento, podem deslocar-se, interagindo com o mouse ou com o teclado, e realizar suas tarefas de forma semelhante ao que fariam no ambiente real. Para isso, eles interagem com representações virtuais dos painéis de controle e também, como resultado deste trabalho, com a representação de um sistema supervisório.

Na Figura 3-1, é apresentada uma vista desse ambiente de simulação, que foi inspirado no ambiente real da sala de controle e supervisão da subestação Campina Grande II (CGD II), da companhia CHESF (Figura 3-2).

Nessa subestação, como é comum nas instalações mais antigas, existem equipamentos com mais de 30 anos que operam paralelamente a outros equipamentos de tecnologia recente (NASCIMENTO, 2010). A modernização da subestação, através da adição de novos equipamentos ou da substituição dos equipamentos obsoletos, é gradativa e ocorre em função das necessidades e prioridades da companhia. A disparidade tecnológica, e também a presença de equipamentos de diferentes fabricantes, implicam em formas diferentes de interação do operador com o sistema.



Figura 3-1 – Sala de controle virtual de uma subestação elétrica, representada no ambiente do simulador

A Figura 3-2 ilustra o ambiente real da sala de controle desta subestação com seus painéis, quadro de alarmes, estações de trabalho, etc.

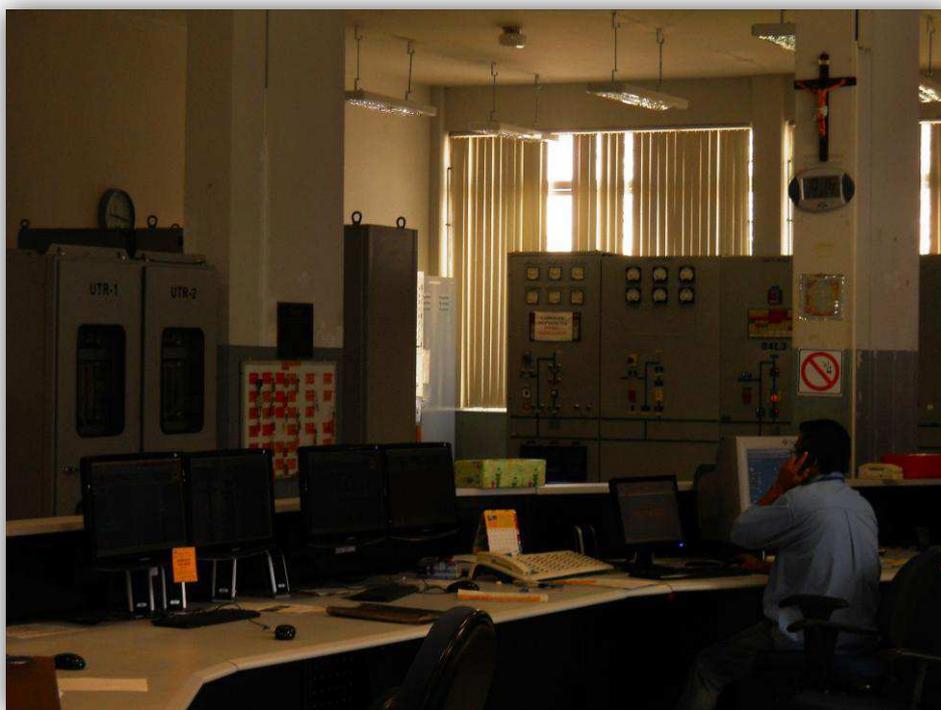


Figura 3-2 – Sala de controle da subestação Campina Grande II

Na versão original do simulador, foi utilizada a linguagem VRML para construção do mundo virtual (FREITAS, 2006). Entretanto, na versão atual, o mundo virtual é descrito na linguagem X3D (BRUTZMAN; DALY, 2007), que é baseada no padrão XML (*eXtensible Markup Language*) e considerada a evolução da VRML.

O X3D(*eXtensible 3D*) é um padrão aberto, reconhecido pela norma ISO/IEC 19776 e desenvolvido pelo Web3D Consortium (2011) que permite descrever em um arquivo as formas e comportamentos dos elementos que compõem um ambiente virtual.

A visualização do ambiente virtual descrito em X3D é possível através de *browsers*, aplicações desenvolvidas para processarem os arquivos x3d e apresentarem os modelos tridimensionais, possibilitando também a navegação no espaço virtual e a interação com os elementos que o compõem.

No projeto do simulador foi adotado como browser o Xj3D (BRUTZMAN; DALY, 2007), *software* de código aberto, desenvolvido na linguagem Java e que oferece uma API SAI (*Scene Access Interface*). Esta API, especificada pela Web 3D e implementada na linguagem Java, oferece um conjunto de métodos que permitem acessar a cena 3D e definir o comportamento dos objetos representados.

A SAI, oferecida pelo browser Xj3D, também permite que o visualizador seja instanciado e configurado para atender as necessidades da aplicação do desenvolvedor.

Essas facilidades de adaptação do *browser* e integração com aplicações Java motivaram a escolha pelo Xj3D. Outras ferramentas usadas no projeto do simulador estão descritas no Apêndice A.

A seguir, apresenta-se a arquitetura desse sistema e os módulos que a compõem.

3.2 *Arquitetura do simulador*

Em sua versão inicial, o simulador apresentava uma representação simplificada e limitada do ambiente virtual e do sistema supervisório (Figura 3-3), apenas como prova de conceito (FREITAS, 2006).

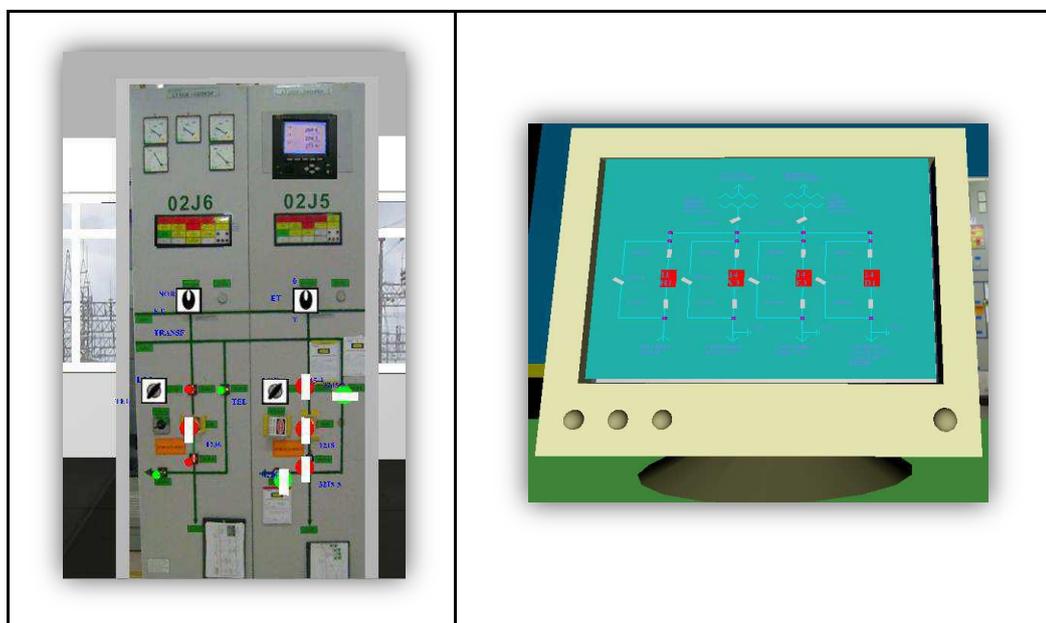


Figura 3-3 – Painel de controle e supervisório representados em VRML, na versão inicial do simulador

Em (SILVA NETTO, 2010) foi proposta outra arquitetura para esse simulador. Os modelos 3D, representando os painéis de controle no ambiente virtual, foram aperfeiçoados e foi desenvolvido um *driver* de comunicação desses modelos com o motor de simulação do sistema. Entretanto, a arquitetura proposta por SILVA NETTO (2010) não incluía um módulo relativo ao supervisório.

Assim, o projeto do simulador foi alterado para acomodar o Módulo Supervisório proposto nesse trabalho. A nova arquitetura também sofreu alterações em virtude da especificação e implementação do Módulo Tutor, outro trabalho em andamento no LIHM. A Figura 3-4 representa esta nova arquitetura, atualmente em desenvolvimento.

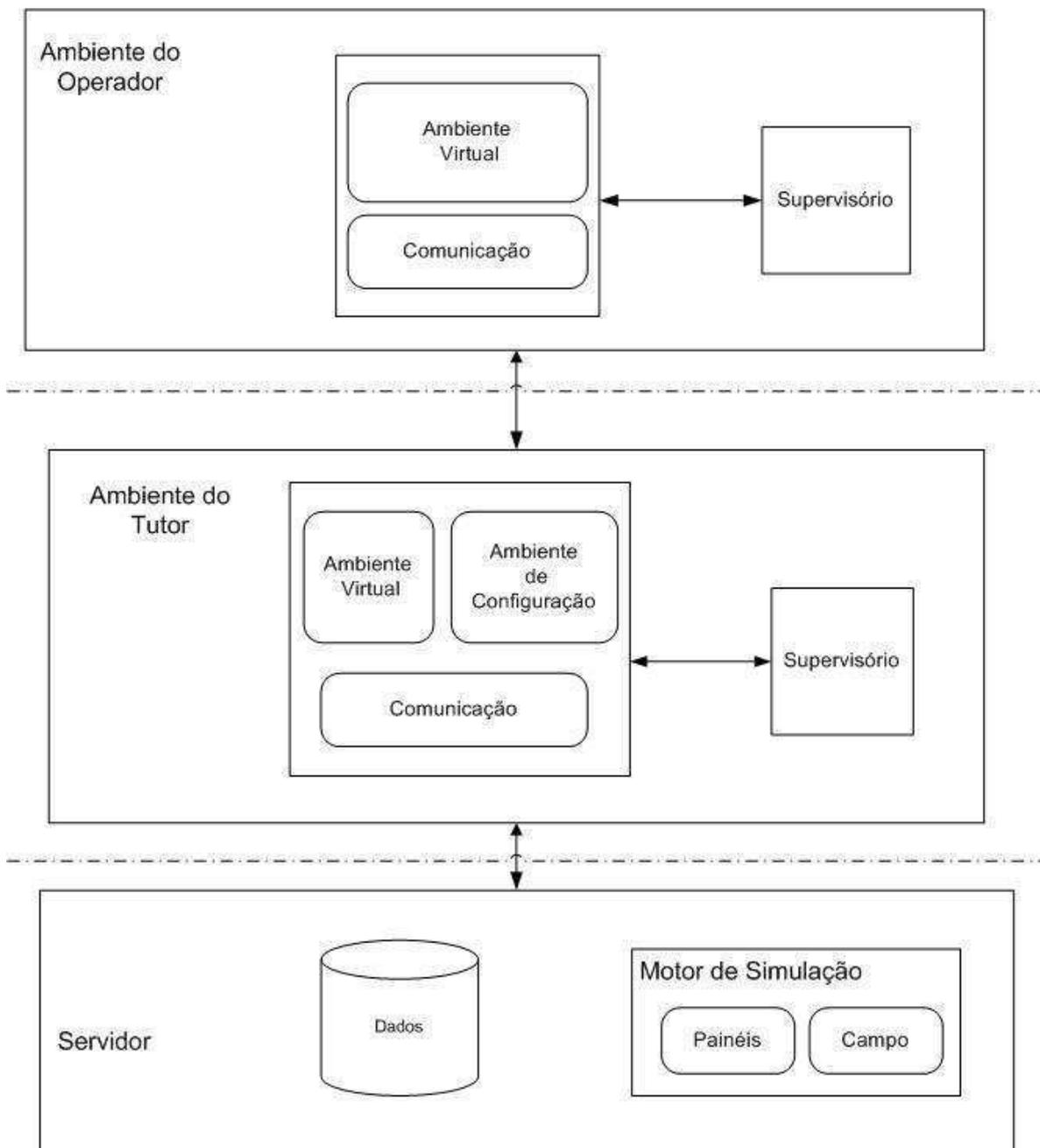


Figura 3-4 – Arquitetura do simulador em desenvolvimento pelo LIHM

A arquitetura, ilustrada na Figura 3-4, é do tipo cliente-servidor e a comunicação entre o servidor, o ambiente do tutor e o ambiente do operador se dá através de *socket* utilizando protocolo TCP/IP (*Transmission Control Protocol/ Internet Protocol*).

Segundo a arquitetura apresentada, o simulador é composto dos módulos descritos a seguir.

3.2.1 Ambiente do operador

Consiste em uma aplicação JAVA com o visualizador Xj3D integrado, permitindo ao usuário navegar e interagir com objetos tridimensionais representados na sala de controle virtual (Figura 3-4). Para aumentar o nível de realismo durante a interação com o simulador, um módulo supervisor é também disponibilizado e pode ser acessado a partir do mundo virtual. Classes Java gerenciam a comunicação do ambiente do operador, que inclui supervisor e mundo virtual, com o ambiente do tutor.

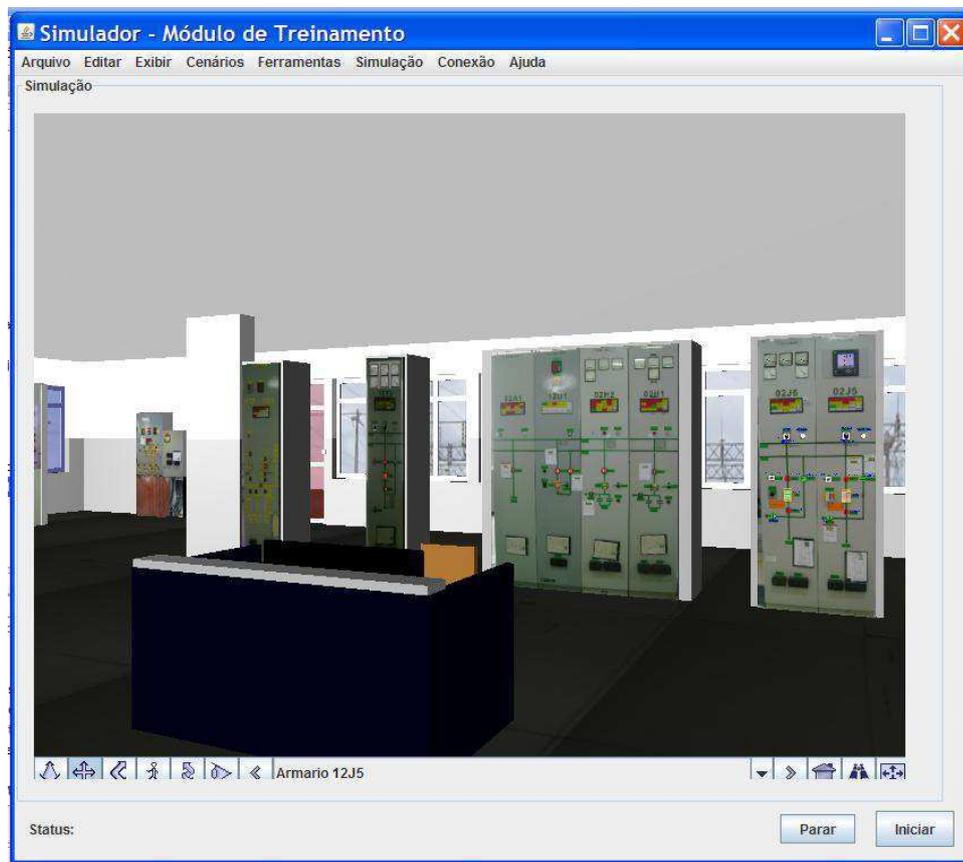


Figura 3-5 – Ambiente do operador no simulador para treinamento de operadores de sistemas elétricos

3.2.2 Ambiente do tutor

Ambiente no qual um instrutor cadastra treinamentos, cria novos cenários de treinamento ou edita cenários armazenados em um banco de dados. O usuário tutor pode acompanhar a execução do treinamento em tempo real, inclusive podendo disparar eventos durante a simulação.

Nesse ambiente, o visualizador Xj3D também está disponível, o que permite que o tutor possa navegar no ambiente virtual para configurar o estado inicial dos objetos nos painéis e no supervisório em um cenário de treinamento específico. Opcionalmente, o tutor também poderá configurar o estado desses objetos através de um menu lateral (Figura 3-6).

Para acompanhar a sessão de treinamento, o tutor pode navegar no ambiente virtual onde o operador está inserido e visualizar as ações de comando sobre os objetos nos painéis ou no supervisório em tempo real. Existe também uma tela de Log que registra todas as ações do operador durante o treinamento.

Classes Java gerenciam a comunicação com o ambiente do operador, com o banco de dados e com o motor de simulação do sistema.

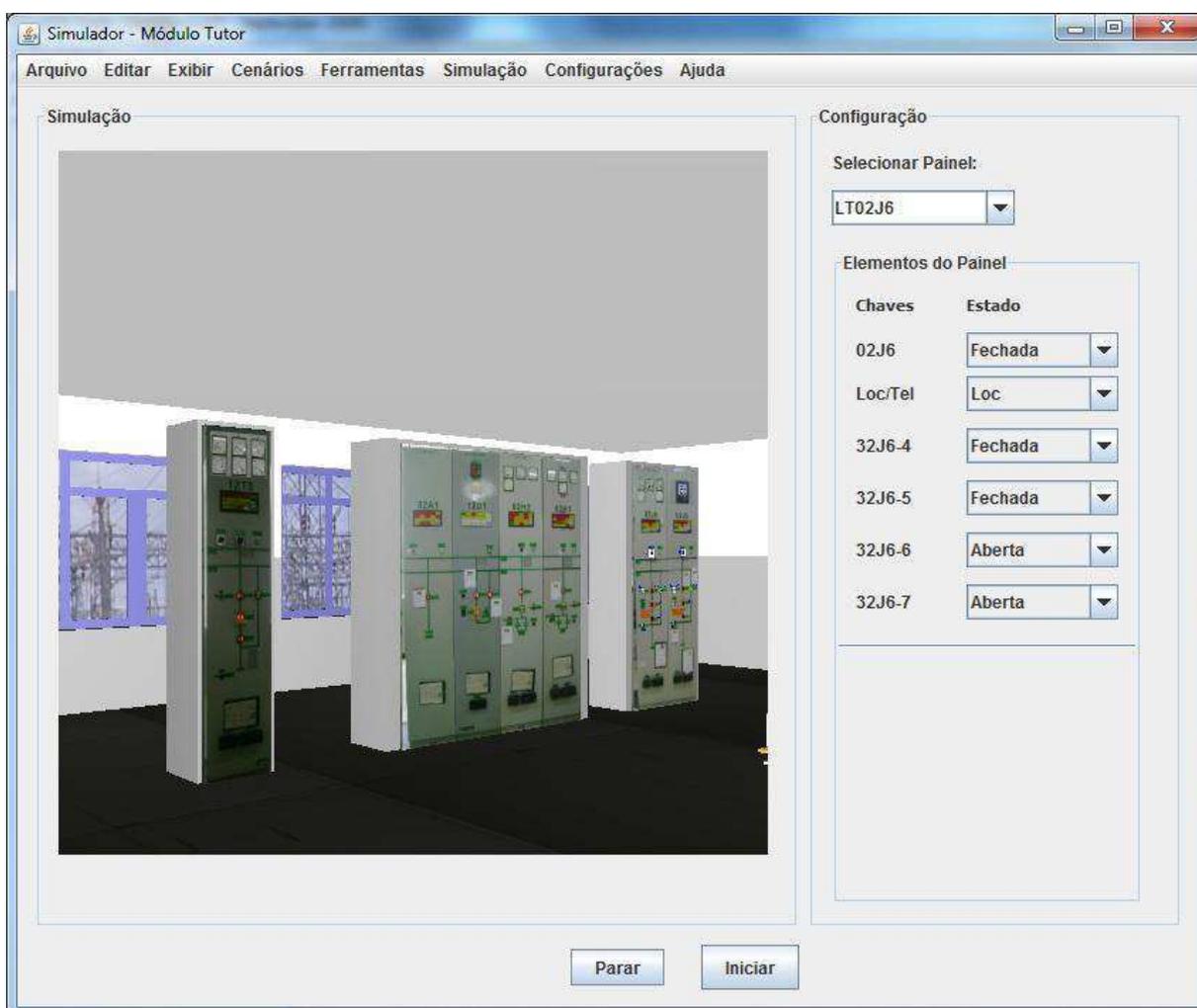


Figura 3-6 – Ambiente do tutor no simulador para treinamento de operadores de sistemas elétricos

3.2.3 Banco de Dados

O banco de dados do simulador armazena cenários de treinamento, o estado dos objetos presentes na interface do ambiente modelado durante uma sessão de treinamento, alarmes gerados na simulação e o histórico de treinamentos realizados, inclusive dados relativos às sessões individuais de treinamentos de operadores.

3.2.4 Motor de simulação baseado em modelos

Uma característica do simulador que distingue seu projeto dos demais é o uso de modelos formais na construção do motor de simulação que define o comportamento dos objetos com representação virtual.

Na construção dos modelos foi usado o formalismo de Redes de Petri Coloridas – CPN (JENSEN, 1997). Os modelos representam o comportamento dos objetos que são comumente encontrados em painéis de controle (chaves, botoeiras, mostradores, painéis de alarme, entre outros) (NASCIMENTO NETO, 2004); e também o comportamento da planta (TORRES FILHO; VIEIRA, 2010). Esses dois modelos integrados compõem o motor de simulação do sistema.

Esse formalismo possibilitou a representação de comportamentos concorrentes e paralelos de eventos no mundo virtual. A notação gráfica e matemática das redes de Petri possibilitam a verificação e a validação do comportamento dos modelos. Além disso, a representação de comportamentos em CPN minimizou o esforço de modelagem do ambiente de simulação reutilizando bibliotecas construídas em trabalhos anteriores do LIHM. Por sua vez, o uso de modelos já verificados e validados minimizou a possibilidade de ocorrerem erros de modelagem.

Na figura 3-7 é apresentado um esquema que ilustra a comunicação entre os modelos, no motor de simulação e entre estes e suas representações gráficas, a qual é descrita a seguir.

Durante uma sessão de treinamento, o usuário (tutor ou treinando) pode interagir com objetos representados nos painéis de controle ou no supervisório. Essa interação resulta no envio de uma mensagem ao módulo de comunicação.

No ambiente do tutor, as mensagens de comando que chegam ao módulo de comunicação são enviadas diretamente ao motor de simulação do sistema. E, no ambiente do operador, o módulo de comunicação transmite a mensagem a aplicação do tutor, que posteriormente encaminha a mensagem ao motor de simulação.

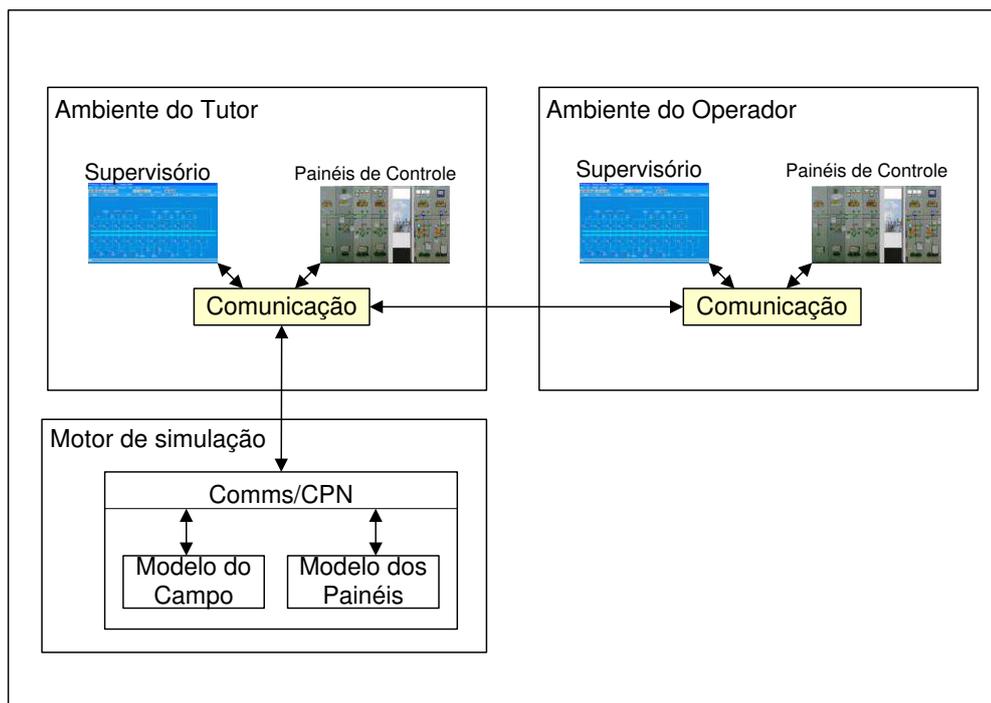


Figura 3-7 – Comunicação entre modelos CPN, Supervisório e Painéis de Controle

Se a mensagem foi originada de uma interação com o supervisor, ela será então transmitida diretamente ao modelo do campo. Entretanto, se a mensagem foi originada de uma interação com um painel, ela será primeiro enviada ao modelo CPN dos painéis, que muda de estado e, só então, envia uma solicitação ao modelo do campo.

Por sua vez, ao receber a nova mensagem, o modelo do campo poderá executar o comando solicitado pelo operador ou não (no caso de simulação de uma falha). Sendo o comando executado com sucesso, uma mensagem de resposta é transmitida ao módulo de comunicação.

Posteriormente, esse módulo realimenta o supervisor e os painéis de controle, atualizando suas representações visuais de acordo com as mensagens recebidas do motor de simulação, finalizando a operação.

Detalhes de implementação do módulo de comunicação serão apresentados no próximo Capítulo.

Composição do motor de simulação

O processo de composição do motor de simulação, a partir da integração dos modelos CPN, foi realizado no contexto deste trabalho e se encontra detalhado em (TORRES FILHO e VIEIRA, 2010).

Na construção dos modelos e durante sua execução foi utilizado o ambiente CPN Tools (2011), o qual permite que as simulações sejam executadas em tempo real. O CPN Tools também oferece ferramentas para geração e análise do espaço de estados das redes.

Enquanto que a comunicação entre os modelos foi viabilizada através da biblioteca COMMS/CPN (Gallasch; Kristensen, 2001), que permite a troca de mensagens TCP/IP entre modelos CPN. Essa biblioteca também possibilita que um modelo CPN se comunique com um processo externo através de mensagens no padrão TCP/IP. Essa característica viabilizou a comunicação com o módulo de comunicação do ambiente tutor, conforme ilustrado na Figura 3-7, e através dele a comunicação com os demais módulos do simulador.

Vale ressaltar que, na versão original do simulador, existia um terceiro modelo que representava os mecanismos de navegação entre as telas de um sistema supervisor e o comportamento de alguns dos elementos de interação disponíveis ao operador do sistema (disjuntores, seccionadoras, chaves de comando local/telecomando). No contexto da época, essa solução foi adotada porque se pretendia construir as telas do supervisor dentro do ambiente virtual e, assim, o modelo CPN serviria para *emular* a navegação no supervisor. Entretanto, na versão atual, esse modelo não será necessário, uma vez que as telas do supervisor serão implementadas em um supervisor real e serão associadas aos elementos do ambiente 3D, conforme é apresentado no Capítulo 4.

A seguir descreve-se de forma sucinta o modelo dos painéis e o modelo do campo, os quais compõem o motor de simulação do sistema.

3.2.4.1 Modelo dos painéis

Os elementos de interação representados neste modelo são aqueles tipicamente encontrados nos painéis de comando dentro de uma subestação elétrica (botões, chaves, etc.) e disponíveis aos operadores para supervisão e controle de dispositivos e equipamentos (disjuntores, seccionadoras, banco de capacitores, etc). Este modelo está detalhado em (NASCIMENTO NETO, 2004).

3.2.4.2 Modelo do Campo

Em subestações elétricas, os equipamentos localizados no pátio da subestação compõem a planta industrial. Estes equipamentos e dispositivos podem ser operados remotamente via programa supervisorio ou via painéis na sala de controle, ou ainda diretamente no painel de comando do equipamento.

Em um projeto anterior, realizado no LIHM, foi iniciada a modelagem da planta para o projeto do simulador (subestação do sistema elétrico), no entanto este projeto não foi concluído. Na literatura são encontradas modelagens de subestações em redes de Petri (SAMPAIO, 2002), no entanto, neste trabalho pretende-se associar à representação dos objetos manipulados pelo operador no ambiente virtual, o comportamento destes objetos no campo, incluindo estados de indisponibilidade por quebra ou mau-funcionamento.

Neste trabalho foram modelados disjuntores, chaves seccionadoras e linhas de transmissão, compondo um modelo de planta simplificado. A descrição detalhada desse modelo, e sua comunicação com os modelos de interface, podem ser encontrados em (TORRES FILHO e VIEIRA, 2010) e no Apêndice B.

Capítulo 4 Sistema supervisório no ambiente do simulador

Neste capítulo são apresentadas as características de sistemas supervisórios voltados para a operação de sistemas elétricos. Também serão discutidos os requisitos, a especificação e a implementação do sistema supervisório que foi integrado aos demais módulos do simulador para treinamento de operadores de subestações elétricas.

4.1 Características de sistemas supervisórios em subestações elétricas

Em subestações elétricas, o sistema elétrico de potência pode ser operado em diferentes níveis: através de terminais de computador, executando software supervisórios, através de painéis na sala de controle ou através da manipulação direta de equipamentos no pátio da subestação.

Um software supervisório concentra informações de um processo industrial, as quais são processadas e apresentadas ao operador do sistema através de terminais de computadores, permitindo o monitoramento e o controle remoto da planta industrial. A supervisão é viabilizada através da apresentação de representações gráficas do processo (sinóticos), com diferentes níveis de detalhes, tabelas com os valores das variáveis de interesse em tempo real, gráficos de tendência, dados históricos, tela de alarmes e relatórios, além da visualização do estado e controle de equipamentos remotos.

No contexto de sistemas elétricos, a planta industrial é representada na tela do supervisório através de diagramas unifilares, cujos elementos retratados são equipamentos de controle, proteção e distribuição de energia. Como exemplo, a Figura 4–1 ilustra o diagrama unifilar de uma subestação da CHESF representado no sistema SAGE – Sistema Aberto de Gerenciamento de Energia (LEITE; RODRIGUES; OLIVEIRA, 2007).

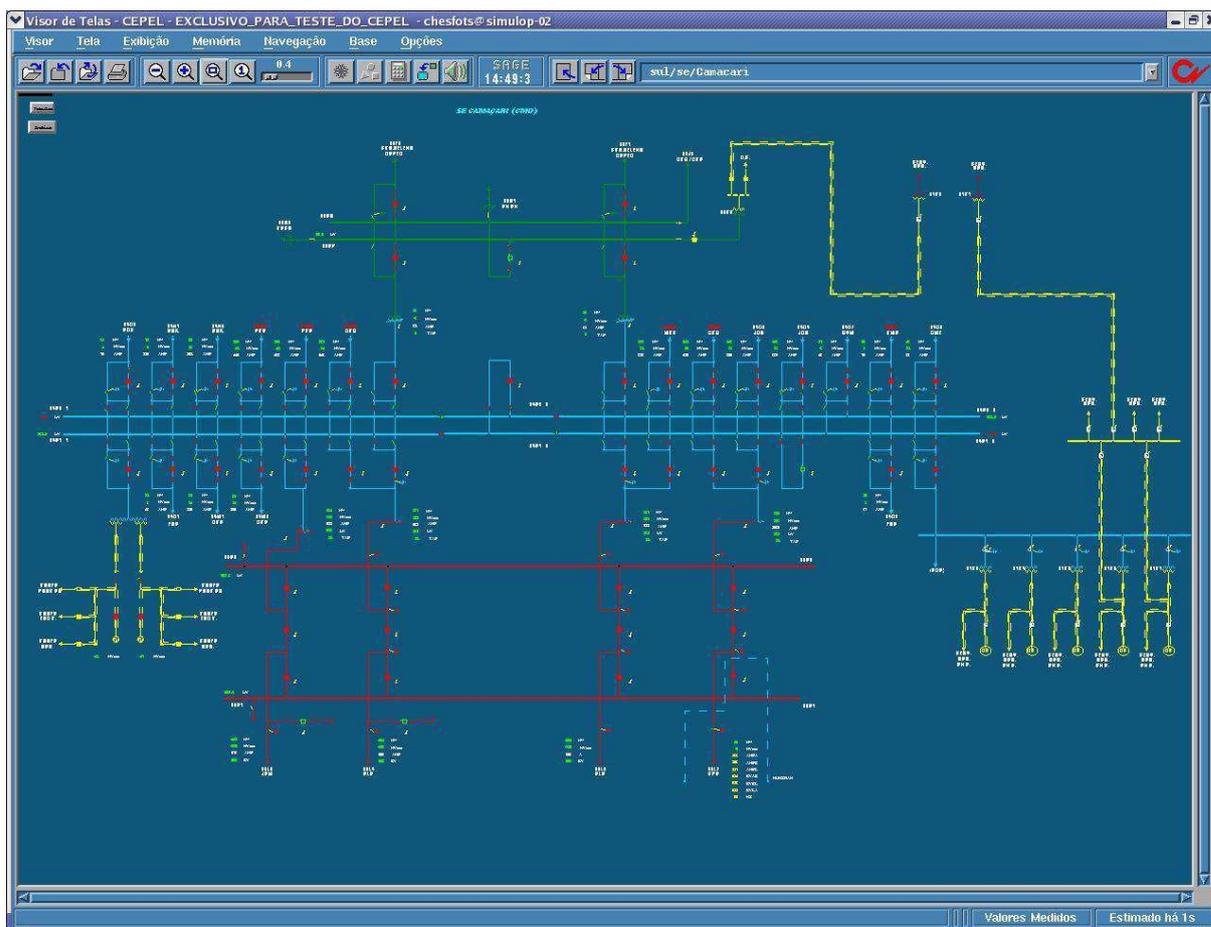


Figura 4-1 – Digrama unifilar de uma subestação representado no supervisorio SAGE (Fonte: Chesf)

Durante um turno de trabalho, o operador da subestação pode monitorar através dessa tela o comportamento de variáveis como potência, tensão e corrente nas linhas de transmissão. Além disso, para manter a estabilidade do sistema e o fornecimento de energia elétrica aos clientes, o operador deve reconfigurar o sistema à medida que seu estado evolui. A reconfiguração também pode ser realizada através do comando remoto dos equipamentos representados nessa tela, como a abertura ou fechamento de disjuntores e chaves seccionadoras.

Essa é uma das telas mais importantes do sistema elétrico que está sendo supervisionado, uma vez que oferece uma visão geral de seu comportamento e concatena informações de algumas das variáveis mais importantes da rede. Normalmente, um código de cores é associado ao estado dos equipamentos e as tensões nas linhas.

O operador também pode navegar entre telas dedicadas a representação de linhas de transmissão específicas, com informações mais detalhadas sobre a linha e seus equipamentos

de proteção. Nessas telas, também são apresentados os alarmes especificamente associados àquela linha de transmissão. Um exemplo é ilustrado na Figura 4-2.

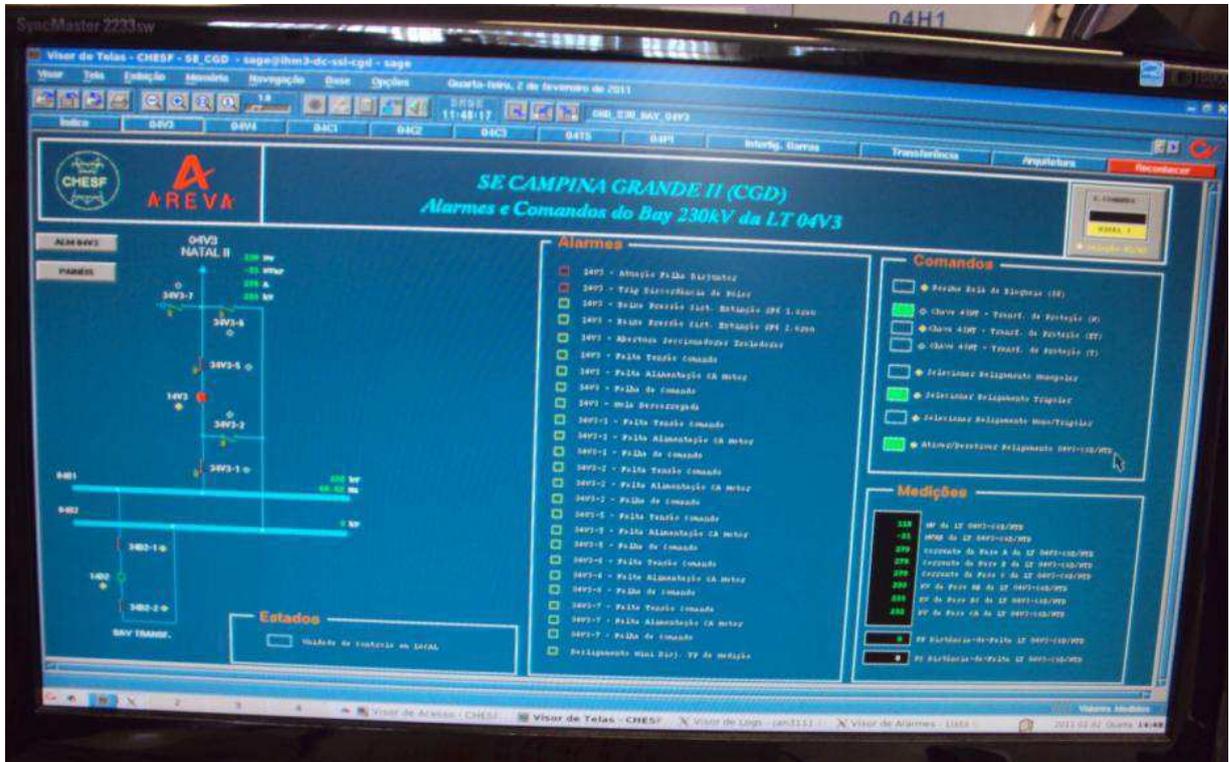


Figura 4-2 – Exemplo de uma tela de sinótico com informações mais detalhadas sobre uma linha de transmissão (Fonte: CHESF)

A propósito dos alarmes, é comum em sistemas supervisorios telas com uma lista dos alarmes gerados durante o período de supervisão e controle. A lista de alarmes é atualizada em tempo real e informa o operador dos eventos ocorridos no sistema elétrico. Como pode ser observado no exemplo da Figura 4 – 3, cada item da lista possui a data e a hora da ocorrência, a identificação do equipamento associado ao evento, seu estado atual e uma mensagem de descrição.

Os alarmes são categorizados pelos projetistas da interface e um código de cores também é associado à ordem de prioridade do alarme. Aqueles alarmes mais importantes são destacados, o operador pode realizar a ação de reconhecimento do alarme, retirando o destaque que o distingue dos demais. Essa ação do operador não acarretará nenhuma mudança no sistema elétrico, mas indica que ele tem conhecimento do evento e de seu significado.

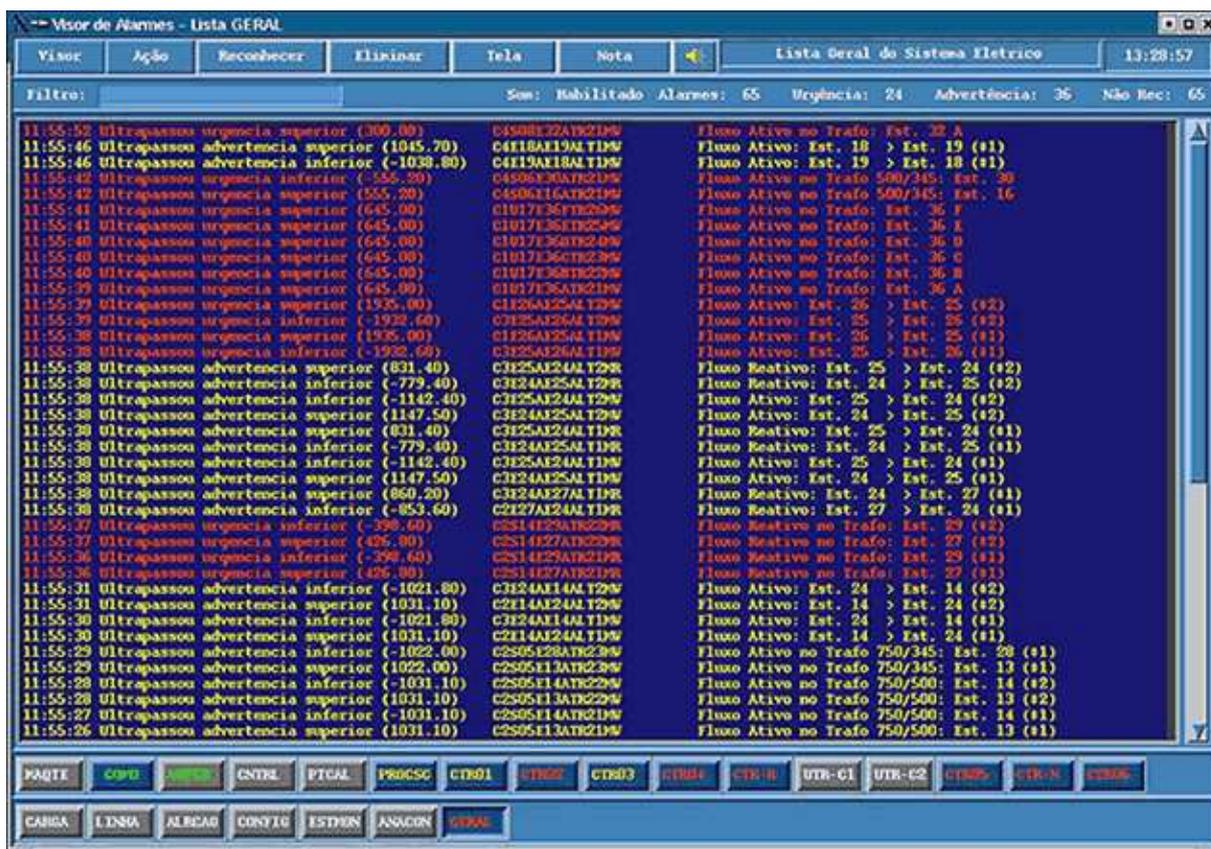


Figura 4-3 – Exemplo de uma tela de alarmes e eventos (Fonte: SAGE/Cepel)

O processo de configuração e implantação de um sistema supervisorio como o SAGE em uma subestação elétrica segue as seguintes etapas:

- **banco de dados:** modelagem e carregamento da base de dados associada ao processo e a configuração do sistema;
- **telas:** edição de telas e associação dos pontos dinâmicos com a Base de Dados de Referência;
- **interface gráfica:** configuração do controle de acesso, das fontes, das cores, das telas e dos relatórios.

A edição de telas do SAGE, segunda etapa descrita acima, é feita através do software SigDraw (Figura 4-4) em cada subestação da CHESF que possui o SAGE operando como software supervisorio. Entretanto, devido à inexistência de um documento organizacional formal que oriente o desenho das telas do sistema elétrico (TORRES FILHO, 2008), observa-se a falta de padronização das interfaces gráficas em uso nas diversas subestações da empresa.

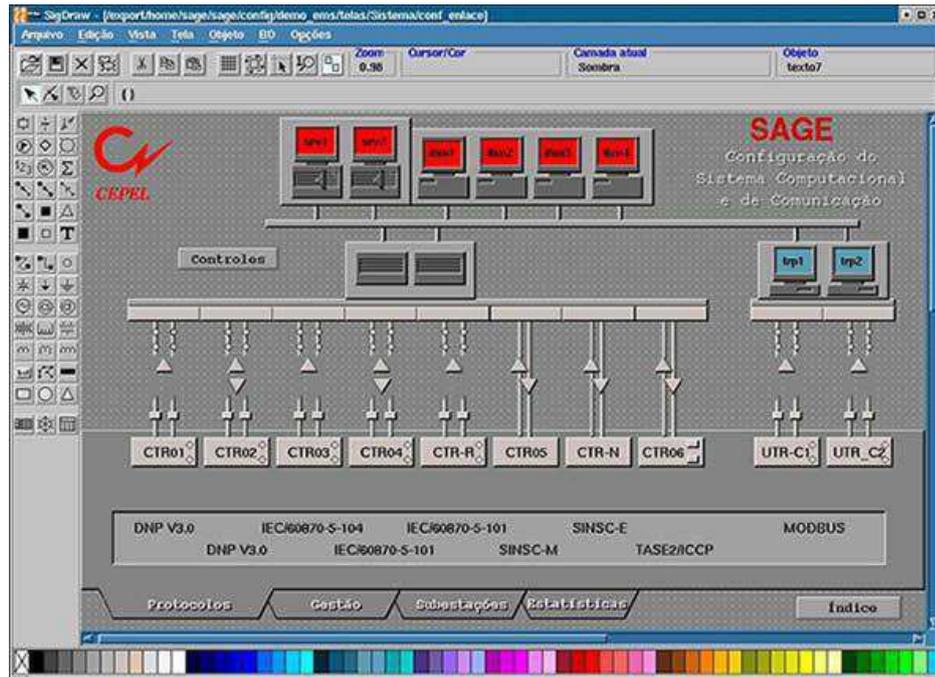


Figura 4-4 – Software para edição de telas do SAGE (Fonte: SAGE/Cepel)

4.2 Análise dos requisitos para o módulo supervisorio do simulador

A seguir, apresentam-se os requisitos considerados para concepção do módulo supervisorio do simulador e sua integração com os demais módulos.

Interface com o usuário:

- para aumentar o nível de realismo e facilitar a imersão do operador na situação simulada, as telas do supervisorio deverão ser inspiradas nas telas do SAGE, supervisorio amplamente utilizado por empresas do setor elétrico;
- o supervisorio deverá ser acessível a partir do ambiente virtual;
- o usuário poderá navegar entre as telas do supervisorio e retornar ao ambiente virtual quando desejar;
- deve-se obedecer ao código de cores associado ao estado dos objetos representados nas telas do supervisorio.

Comunicação:

- o módulo supervisorio deve comunicar-se com o motor de simulação baseado em modelos CPN. Dessa forma, para cada interação do usuário com elementos

representados no sinótico, como disjuntores ou chaves seccionadoras, uma mensagem de comando é enviada ao modelo da planta. Em resposta a essa mensagem, o modelo da planta evolui, executando o comando sobre o dispositivo, e retorna uma mensagem de confirmação;

- a solução adotada deve considerar que o módulo supervisorio será utilizado por usuários em treinamento e também pelo usuário tutor;
- permitir atualização dos estados dos objetos representados nas telas do supervisorio a partir do banco de dados do simulador;
- os estados iniciais dos objetos da interface devem refletir um cenário armazenado no banco de dados e selecionado pelo usuário tutor.

Funcionalidades:

- acesso ao módulo supervisorio através de login e senha do usuário;
- acessar telas com diagramas unifilares e efetuar comandos sobre os equipamentos representados nesses diagramas;
- as ações do operador interagindo com o supervisorio deverão refletir sobre os objetos representados no ambiente virtual;
- a mudança de estado dos objetos representados no ambiente virtual podem ser percebidos através do ambiente supervisorio: através de sinóticos e tela de alarmes;
- visualizar alarmes gerados durante a simulação e permitir que o usuário possa reconhecê-los;
- permitir que o usuário tutor possa configurar o estado inicial dos dispositivos a partir do diagrama unifilar da subestação representado na tela do supervisorio.

Essas funcionalidades estão representadas no diagrama de casos de uso, ilustrado na Figura 4-5.

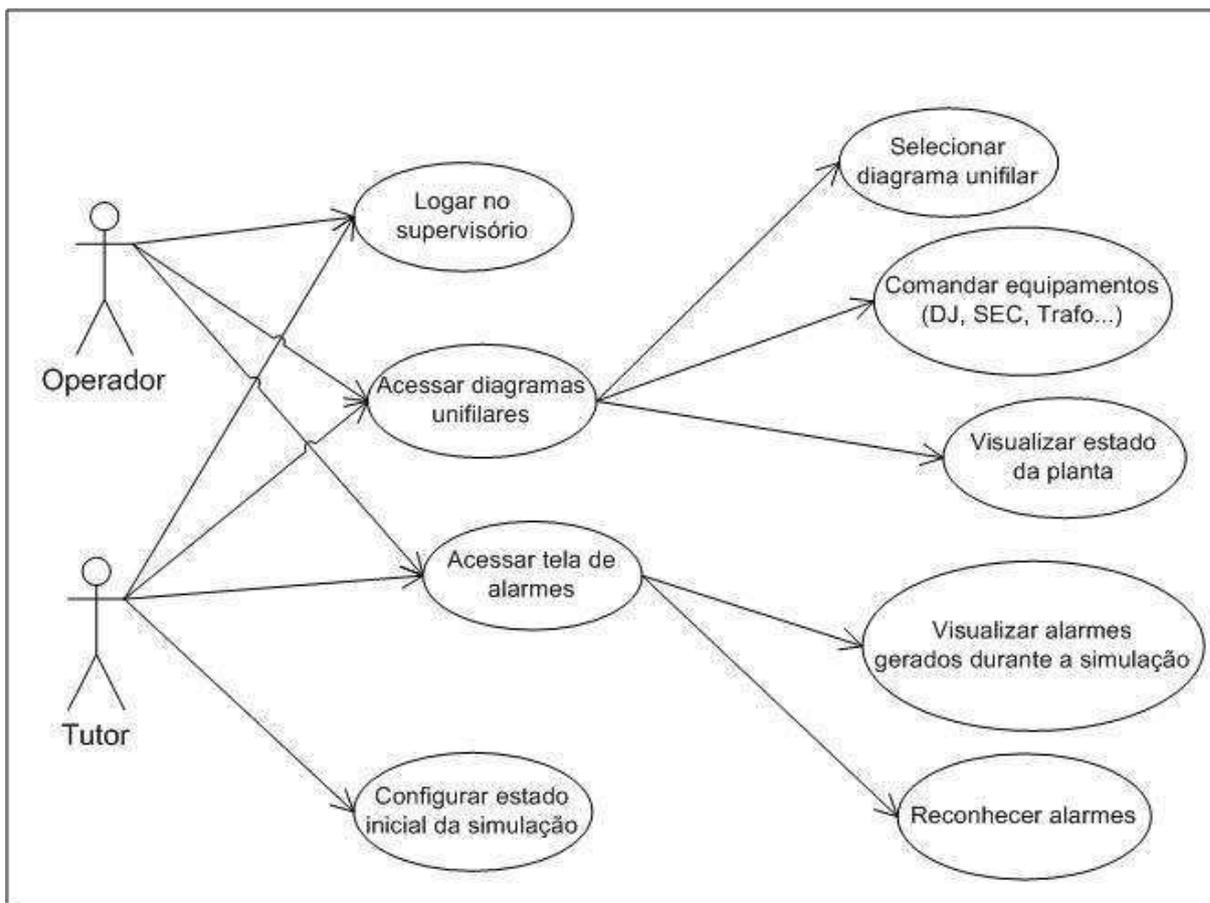


Figura 4-5 – Diagrama de casos de uso no módulo supervisório

4.3 Especificação e implementação do módulo supervisório

A arquitetura do simulador proposta por SILVA NETTO (2010) foi alterada para acomodar o módulo supervisório no ambiente de simulação. A nova arquitetura também sofreu alterações em virtude da especificação e implementação do ambiente do tutor, outro trabalho em andamento no LIHM. A Figura 3–4 representa a arquitetura atual do simulador.

Durante uma situação de treinamento, os dois tipos usuários do simulador, treinando e tutor, estão inseridos no mesmo ambiente virtual (sala de controle de uma subestação), assim um pode visualizar as ações do outro. A comunicação entre o ambiente do tutor e o ambiente do treinando se dá através de *socket* utilizando protocolo TCP/IP (*Transmission Control Protocol/ Internet Protocol*).

No diagrama de sequência, apresentado na Figura 4-7, é demonstrado como ocorre a troca de mensagens entre os módulos do simulador, quando o operador comanda a abertura de um disjuntor através da tela do supervisório.

Nessa situação, a interação do operador com o disjuntor representado na tela do supervisório resulta em uma mensagem enviada ao módulo de comunicação, que gerencia o tráfego de mensagens. A informação é transmitida ao módulo de comunicação do ambiente do tutor, que por sua vez encaminha a mensagem ao modelo CPN do campo, solicitando a execução do comando.

Sob condições normais de operação, o modelo do campo executa o comando do operador e envia uma mensagem ao módulo de comunicação do tutor, em resposta a solicitação.

Em seguida, o estado do equipamento comandado é atualizado no banco de dados do simulador. Através do envio de mensagens, o módulo de comunicação também atualiza a representação do estado do equipamento nos painéis do ambiente virtual e no supervisório do tutor. Além disso, encaminha a informação para o módulo de comunicação do ambiente do operador que, por sua vez, também atualiza a representação do estado do equipamento nos dois níveis (painel e supervisório).

Além de visualizar a mudança de estado dos equipamentos, o usuário tutor pode atuar sobre eles, enviando comandos ao motor de simulação (modelos CPN). E, de forma análoga a situação descrita anteriormente, o módulo de comunicação do ambiente tutor transmite a informação aos módulos do ambiente de simulação do operador.

A seguir, serão detalhados o processo de concepção da interface do usuário com o supervisório, as telas que foram implementadas e como o usuário interage com elas. Também são apresentados as classes que gerenciam a comunicação com o supervisório e o *driver* que foi implementado com esse objetivo.

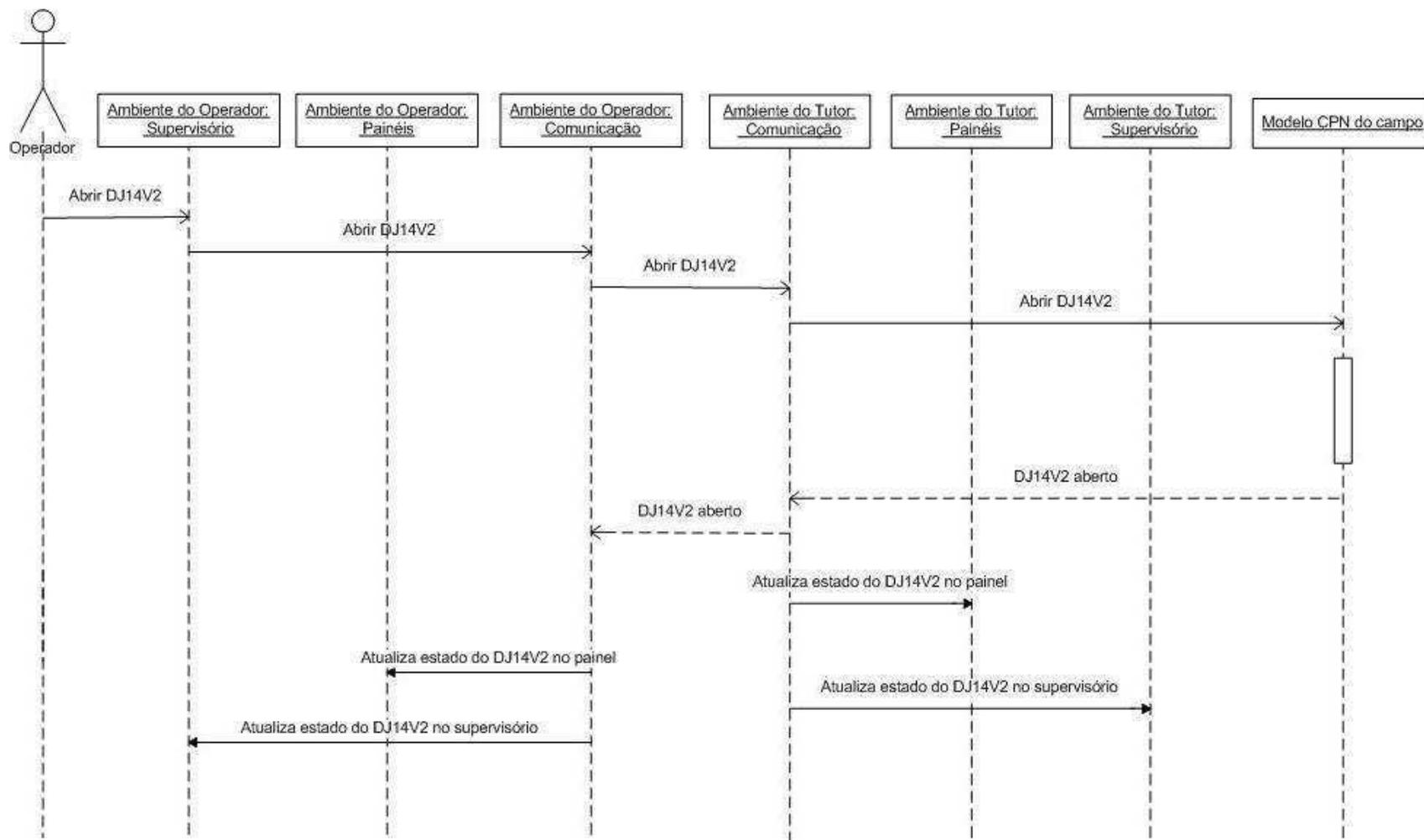


Figura 4-6 – Diagrama de sequência para a execução do comando de abertura de um disjuntor através do supervisório

4.3.1 Implementação da interface com o operador e com o tutor

A interface do usuário com o módulo supervisorio do simulador foi concebida com base no sistema SAGE – Sistema Aberto de Gerenciamento de Energia, supervisorio desenvolvido pelo CEPEL – Centro de Pesquisas de Energia Elétrica, e adotado por diversas empresas do setor elétrico do Brasil.

O sistema SAGE é composto por diferentes aplicativos conhecidos como “Visores” (CEPEL, 2011), dos quais os três mais usados pelos operadores foram modelados para compor o módulo supervisorio do simulador:

- **visor de acesso:** Aplicação através da qual o usuário tem acesso à interface gráfica do SAGE. Permite a verificação de senhas e privilégios dos usuários, e controla o acesso aos demais visores do supervisorio;
- **visor de telas:** Aplicação na qual o usuário tem acesso a diagramas unifilares do sistema elétrico permitindo a monitoração e o controle do mesmo;
- **visor de alarmes:** Aplicação em que o operador pode visualizar, em tempo real, os alarmes e eventos do sistema elétrico.

Na sala de controle virtual, existe uma representação em 3D da estação de trabalho do operador (Figura 4-8). Ao clicar com o mouse sobre a tela do monitor dessa estação de trabalho, o usuário inicia o submódulo (ou visor) de acesso ao supervisorio do simulador.

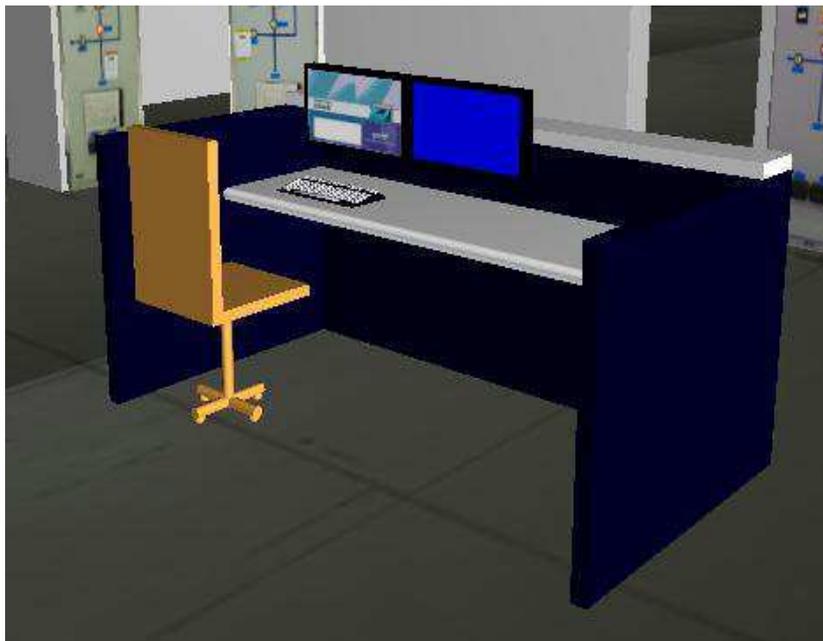


Figura 4-7 – Estação de trabalho do operador representada na sala de controle virtual

A Tela de Login é apresentada ao usuário (Figura 4-9), que deverá entrar com seus dados nos campos “Usuário” e “Senha”. Se for autenticado pelo sistema, o usuário será direcionado a aba “Programas”, onde poderá selecionar outros submódulos do supervisorio (Figura 4 - 10).

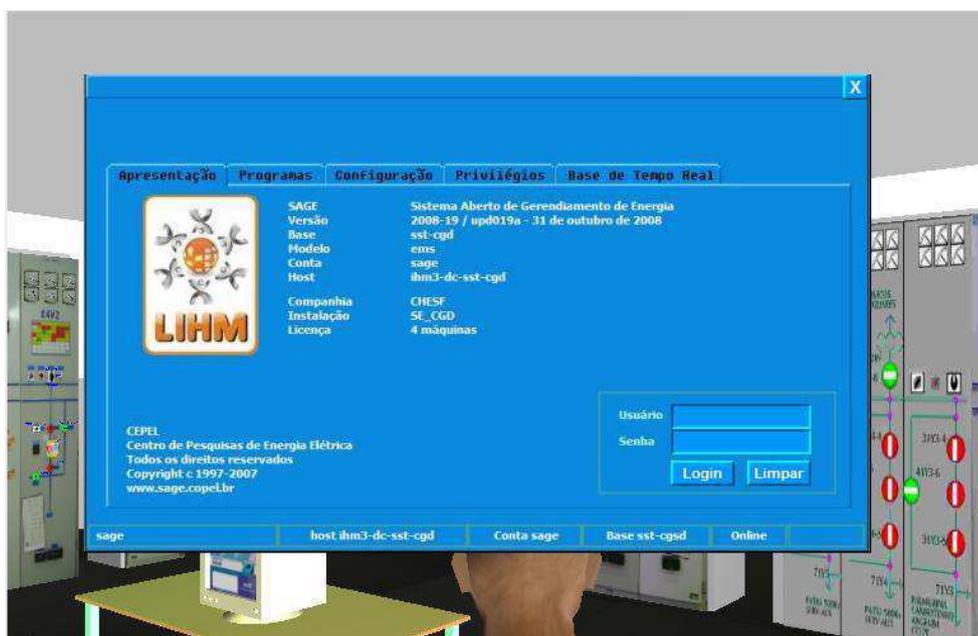


Figura 4-8 – Tela de Login do supervisorio

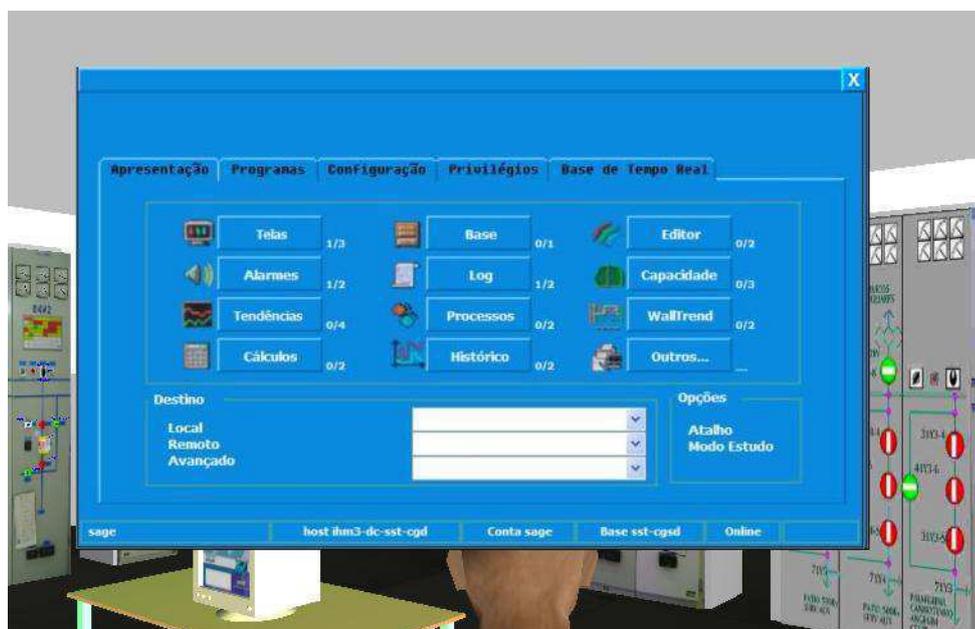


Figura 4-9 – Tela de acesso aos submódulos do supervisorio

Para acessar as telas com diagramas unifilares da subestação, por exemplo, o usuário deve clicar na opção “Telas”, dentro da aba “Programas”. Já para iniciar o visor de alarmes, o usuário deve clicar sobre a opção “Alarmes”.

Essas telas, que permitem ao usuário acessar o supervisorio do sistema simulado, foram implementadas a partir de classes na linguagem Java. Entretanto, as telas com diagramas unifilares e a tela de alarmes foram concebidas no *InTouch* (WONDERWARE, 2011), sistema supervisorio comercial também bastante aceito na área industrial.

Assim, ao clicar sobre os botões “Telas” ou “Alarmes”, o usuário iniciará a execução do *InTouch*. Outros *softwares* supervisorios também poderiam ter sido escolhidos para construção da interface, optou-se pelo *InTouch* devido à disponibilidade de licenças comerciais para uso no LIHM.

A Figura 4-10 ilustra uma dessas telas, com um diagrama unifilar representando as linhas transmissão de 230kV da subestação Campina Grande II.

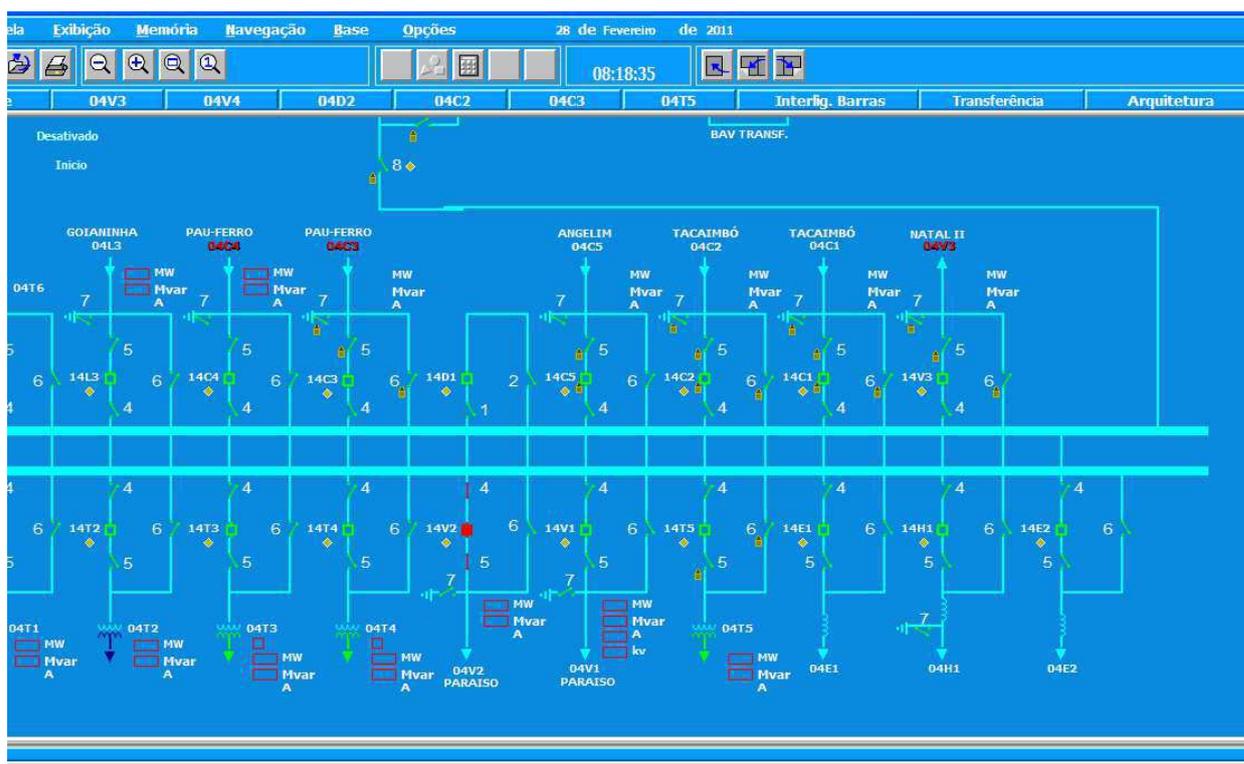


Figura 4-10 – Tela com o diagrama unifilar da subestação, implementada no *InTouch*

Em telas como a da Figura 4-10, o operador pode monitorar e controlar o estado dos equipamentos que estão representados. Um código cores é associado ao estado atual desses

equipamentos. Para disjuntores e seccionadoras, por exemplo, a cor vermelha indica que o equipamento encontra-se no estado “Fechado”, e a cor verde no estado “Aberto”.

Para efetuar o comando sobre um equipamento em particular, modificando seu estado atual, o operador deve selecioná-lo na tela, clicando com o *mouse*. Feito isso, uma janela do tipo *pop-up* surge no monitor, onde o operador deve confirmar sua intenção de abrir ou fechar o equipamento.

A Figura 4-11 ilustra uma dessas janelas para o controle dos equipamentos. Ao confirmar a ação, uma mensagem de comando é enviada.

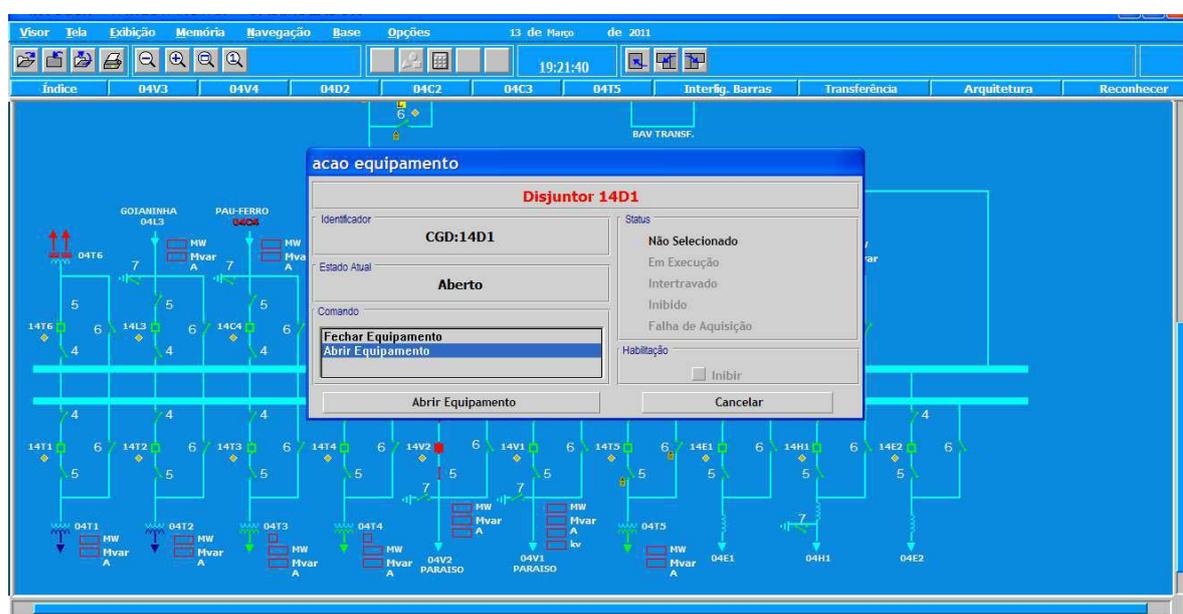


Figura 4-11 – Janela *pop-up* para o controle do disjuntor 14D1, implementada no *InTouch*

O Visor de Alarmes, representado na Figura 4-12, é iniciado quando o usuário clica sobre o botão “Alarmes”, no Visor de Acesso (Figura 4-10).

O supervisorio está configurado para acessar o banco de dados (BD) do simulador, assim sempre que um novo evento ocorre, e o estado de um equipamento é modificado no BD, um novo alarme é gerado na tela de alarmes. Cada alarme possui a data e hora em que ocorreu, a identificação do equipamento ao qual está associado e o novo estado desse equipamento.

Antes de serem reconhecidos pelo usuário, os caracteres que compõem o alarme surgem em destaque, alternando entre a cor branca e vermelha. Quando o usuário clica sobre o botão “Reconhecer” (menu superior da tela), indicando que o evento foi identificado, o alarme perde esse destaque e os caracteres adquirem a cor preta.



Figura 4-12 – Tela de alarmes no supervisorio do simulador, implementada no *InTouch*

4.3.2 Estratégia de integração com os demais módulos da arquitetura

O módulo de comunicação, presente no ambiente do tutor e do operador, possui classes que gerenciam a troca de informação entre os demais módulos do simulador. Particularmente, as classes responsáveis por enviar e receber dados do supervisorio foram desenvolvidas nesse trabalho, quais sejam: *ddeDriver*, *supervisorioTutor* e *supervisorioTreinando*.

A classe *ddeDriver* estabelece uma interface de comunicação com o supervisorio baseando-se no protocolo DDE (*Dynamic Data Exchange*). Os métodos dessa classe permitem estabelecer uma conexão com o supervisorio, desconectar e enviar/receber dados no formato de *string*. Detalhes de implementação da classe "*ddeDriver*" serão apresentados a seguir. Mas, antes disso, iremos apresentar a função desempenhada por cada método das classes "*supervisorioTutor*" e "*supervisorioTreinando*".

A classe *supervisorioTutor.java*

Esta classe está presente no módulo de comunicação do ambiente do usuário Tutor, e os métodos são:

- `abrirVisorDeTelas()` – permite iniciar o supervisorio *InTouch* com as telas associadas aos diagramas unifilares do sistema elétrico simulado;
- `abrirVisorDeAlarmes()` – permite iniciar o supervisorio *InTouch* com a tela de alarmes do sistema elétrico simulado;
- `conectarCampo()` – estabelece um canal de comunicação com a rede CPN que modela o comportamento do campo;
- `recCampo(mensagem: String)` – permite informar ao supervisorio mensagem recebida do modelo CPN do campo;
- `envCampo(mensagem: String)` – permite enviar mensagens ao modelo CPN do campo;
- `recTreinando(mensagem: String)` – recebe mensagens vindas do ambiente do operador e encaminha para o modelo CPN do campo;
- `iniciarAlarmes` – inicializa tabela com os alarmes no banco de dados do simulador;
- `atualizarAlarmes` – atualiza a tabela de alarmes no banco de dados do simulador;
- `run()` – monitora o estado do supervisorio e gerencia a troca de dados com o *InTouch* durante a execução da simulação. O diagrama na Figura 4-14 representa a execução do método.

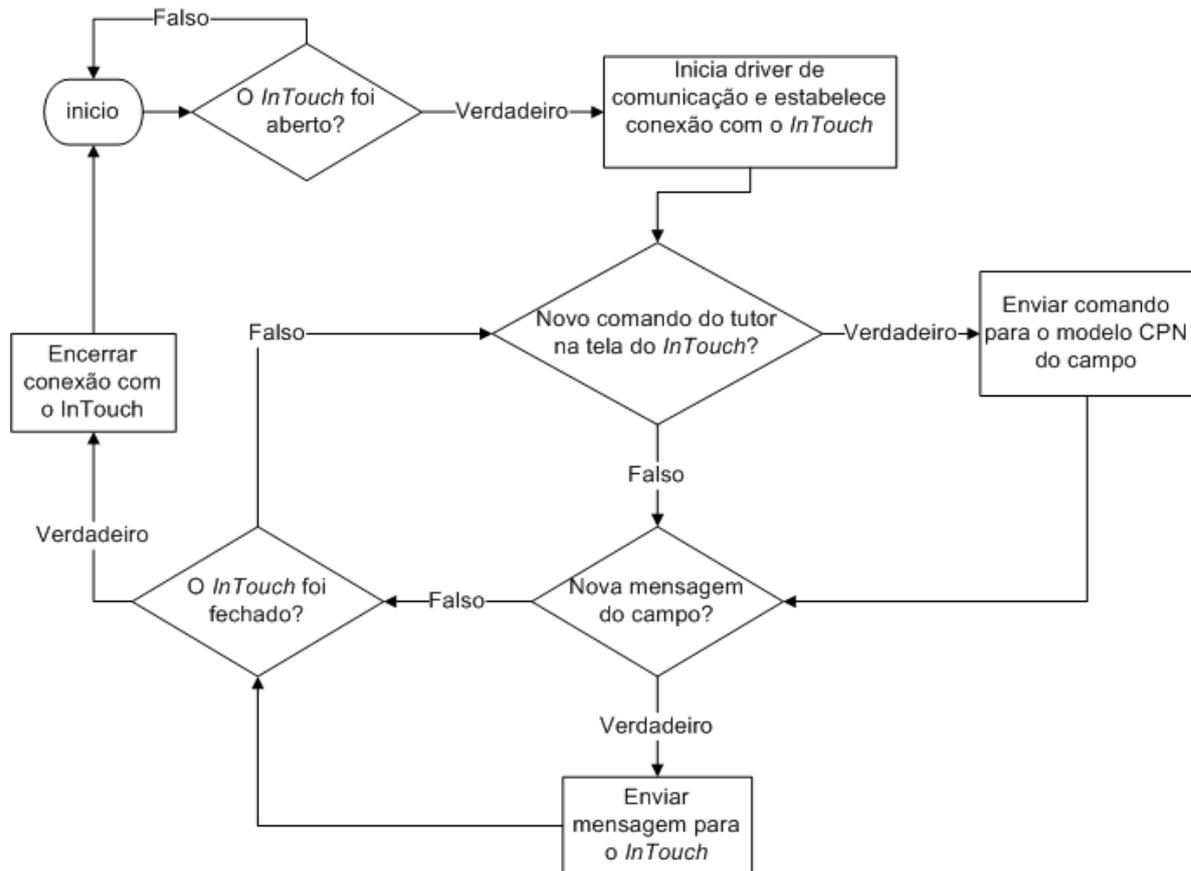


Figura 4-13 – Representação do ciclo de execução do método run() da classe *supervisorioTutor.java*

A classe *supervisorioTreinando.java*

Esta classe está presente no módulo de comunicação do ambiente do usuário Treinando, e os métodos são:

- `abrirVisorDeTelas()` – mesmo método presente na classe *supervisorioTutor.java*;
- `abrirVisorDeAlarmes()` – mesmo método presente na classe *supervisorioTutor.java*;
- `envTutor(mensagem: String)` – permite enviar mensagens ao módulo de comunicação do ambiente do tutor na ocorrência de um comando do operador no supervisorio;
- `recTutor(mensagem: String)` – permite informar ao supervisorio mensagem recebida do módulo de comunicação do ambiente do tutor;

- run() – monitora o estado do supervisorio e gerencia a troca de dados com o *InTouch* durante a execucao da simulacao. Esse metodo, que tambem esta presente na classe *supervisorioTutor.java*, sofreu algumas alteracoes (em destaque, na Figura 4-15) uma vez que o ambiente do operador nao troca informacoes diretamente com os modelos CPN;

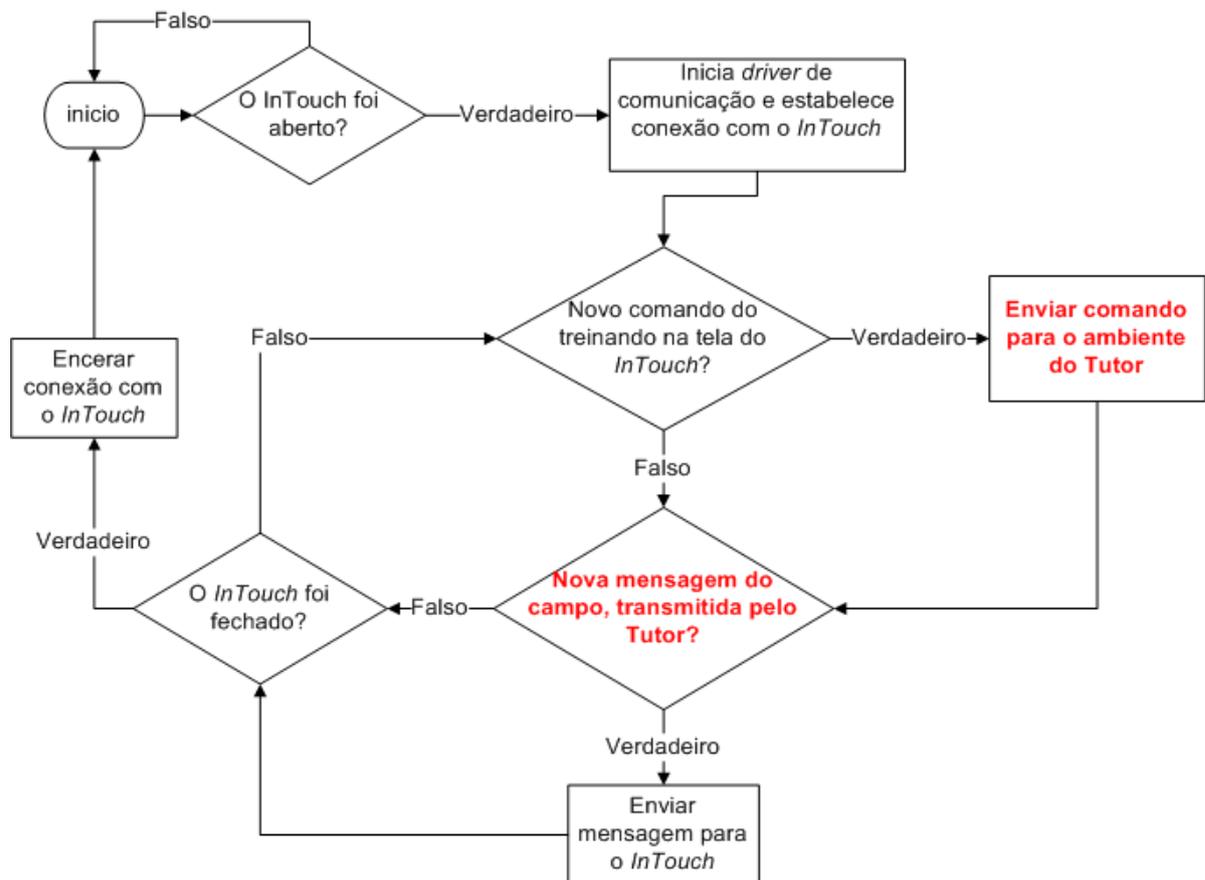


Figura 4-14 – Representacao do ciclo de execucao do metodo run() da classe *supervisorioTreinando.java*

Driver de comunicacao baseado no protocolo DDE

Todo software de supervisao possui interfaces de comunicacao que possibilitam a troca de informacao com outras aplicacoes. O *InTouch* permite a leitura/escrita de dados em aplicacoes externas atraves do protocolo de comunicacao DDE (*Dynamic Data Exchange*), nativo dos sistemas operacionais da *Microsoft*, ou pelo protocolo proprietario *suitelink*, da *Wordware*.

O protocolo de comunicação DDE implementa uma relação cliente-servidor entre dois aplicativos que estejam simultaneamente em execução. O aplicativo de servidor fornece dados e aceita pedidos de qualquer outra aplicação interessada nos seus dados. Essas aplicações, que solicitam dados do servidor, são chamadas clientes. Uma aplicação *InTouch* pode ser simultaneamente um cliente e um servidor.

O protocolo DDE permite o intercâmbio dinâmico de dados a partir da configuração de três parâmetros básicos:

- **aplicação** – nome do programa servidor;
- **tópico** – nome do tópico de acesso;
- **item** – endereço da variável.

Foi desenvolvida uma DLL (*Dynamic-Link Library*), que tem a função de criar uma interface de comunicação, baseada no protocolo DDE, entre o supervisorio e as classes em java que compõem o módulo de comunicação do simulador. Uma DLL é uma biblioteca dinâmica de funções que podem ser chamadas a partir de outra aplicação. A DLL permite que programas possam acessar as suas funções, independentemente da linguagem de desenvolvimento destes programas e da linguagem de desenvolvimento da DLL. Dessa forma, é possível utilizar funções do código desenvolvido utilizando as bibliotecas do C++ através do Java. A biblioteca dinâmica que foi implementada é chamada “*ddeDriver.dll*”, e o código na linguagem C++ encontra-se no Apêndice D.

A Figura 4-16 apresenta a classe “*ddeDriver.java*” que se comunica com a DLL e executa os métodos.

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package Supervisorio;
7
8  /**
9   *
10  * @author Flavio
11  */
12  public class DdeDriver {
13
14      static {
15          System.loadLibrary("DdeDriver");
16      }
17
18
19      public static native void inicializar();
20      public static native void conectar(String App, String Topico);
21      public static native void enviar(String Item, String Dado);
22      public static native String receber(String Item);
23      public static native void desconectar();
24
25  }
26
```

Figura 4-15 – classe *ddeDriver.java*

Na Linha 15 foi inicializada a DLL; observa-se que os parâmetros dos métodos também são passados para o Java, conforme as linhas 20, 21 e 22. Os métodos inicializam a comunicação, criam uma conexão com o *InTouch*, enviam e recebem dados.

Capítulo 5 Avaliação funcional do módulo supervisor do simulador

Com o objetivo de avaliar o trabalho que foi desenvolvido, foram realizados testes com usuários que desempenharam os papéis de um operador de sistemas elétricos iniciante e de um tutor conduzindo uma sessão de treinamento.

No cenário de teste planejado, o usuário no papel de tutor deve configurar o ambiente do simulador para realização do treinamento, gerar a ocorrência de um alarme e acompanhar as ações do treinando. Para isso, ele poderá interagir com o supervisor, com os painéis de controle no ambiente virtual e com um menu de configuração disponível na interface do tutor.

Por outro lado, o usuário no papel de operador deverá simular a realização de uma manobra no sistema elétrico que foi representado no simulador, interagindo com os painéis de controle e com o supervisor.

No que segue, será descrito o experimento que foi realizado, a configuração do ambiente de teste, o perfil dos usuários participantes e a manobra que deveriam executar. Em seguida, apresentam-se os critérios adotados para avaliação, os dados coletados e a análise dos resultados.

5.1 Planejamento do experimento

O principal objetivo na realização desse experimento é validar o módulo supervisor que foi desenvolvido e avaliar sua integração com os demais módulos do simulador. Como objetivos específicos, pretende-se avaliar se:

- os recursos e funcionalidades oferecidos pelo supervisor são suficientes para representar uma situação real;
- a representação do supervisor no ambiente de simulação é adequada;
- o tempo de resposta da interface é suficiente e tolerável pelo usuário;
- os estados dos dispositivos representados no supervisor são coerentes com as correspondentes representações nos painéis de controle, com os modelos do motor de simulação e com o banco de dados do sistema;

- existe erro na comunicação do supervisor com demais módulos do simulador;
- as informações são apresentadas na tela em uma sequência coerente e agradável ao usuário.

O ensaio de avaliação foi realizado no Laboratório de Interfaces Homem-Máquina (LIHM), que possui a infraestrutura necessária para gravação da sessão de teste em áudio e vídeo para posterior análise. Os usuários, tutor e treinando, realizaram as tarefas do teste em uma sala com câmeras e microfones, permitindo que os avaliadores pudessem acompanhar o experimento de uma sala vizinha (sala de controle), através de monitores.

Na sala de teste também existiam dois computadores, um executando a aplicação do tutor e o outro a do treinando. Antes de iniciarem a execução das tarefas, os usuários foram informados do propósito do experimento, o papel que deveriam desempenhar, os recursos e funcionalidades do simulador. Os avaliadores também demonstraram como navegar no ambiente 3D, como interagir com os objetos virtuais e como iniciar o supervisor.

Foram realizadas duas sessões de teste com três usuários homens, na faixa etária de 25 a 35 anos, com ensino superior completo e com conhecimento na área de sistemas elétricos em nível médio, mas sem nenhuma experiência de operação. Essas características são similares a de um operador iniciante.

Dos três usuários selecionados, apenas o UT1 (usuário de teste 1) tinha experiência no uso da versão atual do simulador. Na Tabela 1, pode ser observado o papel desempenhado pelos usuários em cada sessão de teste.

Tabela 5-1 – Papel desempenhado pelos usuários em cada sessão de teste

	Sessão A	Sessão B
Tutor	UT1	UT2
Treinando	UT2	UT3

UTn – usuário de teste n

5.1.1 Critérios de avaliação

Para atingir os objetivos do experimento, foram adotados alguns critérios de avaliação, os quais estão reunidos na Tabela 5-2, que apresenta também como os dados são coletados.

Tabela 5-2 – Plano de coleta de dados e critérios de avaliação

Critério de avaliação	Como coletar?
Consistência dos dados	Observação e registro audiovisual
Corretude da informação apresentada	Observação e registro audiovisual
Tempo de resposta da interface	Análise do questionário pós-teste, registro audiovisual e entrevista
Facilidade de aprendizado	Após breve apresentação do simulador (recursos de navegação, interação e acesso ao ambiente supervisor), verificar o desempenho do usuário na tarefa.
Facilidade de uso e navegação	Análise do questionário pós-teste, registro audiovisual e entrevista
Facilidade de compreensão	Análise do questionário pós-teste, registro audiovisual e entrevista

5.2 Apresentação das tarefas de teste

Para orientar os usuários na realização das tarefas, foram produzidos os documentos “Roteiro de tarefa de teste – Tutor” e “Roteiro de tarefa de teste – Operador”, os quais se encontram no Apêndice C.

A seguir, apresenta-se uma descrição sucinta das tarefas realizadas pelos dois tipos de usuários em que eles interagem com o supervisor.

Usuário Tutor: Configurar o Cenário

O usuário é solicitado a configurar um cenário de treinamento, antes que o operador inicie a execução de uma manobra. O processo de configuração consiste em verificar e atribuir os estados corretos aos dispositivos presentes no cenário simulado. Nesse caso, o usuário, interagindo com o supervisor, deve configurar os dispositivos presentes na linha de transmissão LT04V2 de acordo com a Tabela 5-3.

Tabela 5-3 – Dispositivos a serem configurados através do supervisor

Nome	Tipo	Estado inicial
14D1	Disjuntor	Aberto
14V2	Disjuntor	Fechado
34V2-4	Seccionadora	Fechado
34V2-5	Seccionadora	Fechado
34V2-6	Seccionadora	Aberto
34V2-7	Seccionadora	Aberto

A Figura 5-1 representa a linha LT04V2 já com seus dispositivos corretamente configurados. Observe que as chaves seccionadoras são identificadas no diagrama apenas com o último dígito de seu nome.



Figura 5-1 – Representação da linha de transmissão 04V2 com os dispositivos no estado inicial do cenário

Usuário Tutor: Acompanhar o treinamento

Nessa tarefa, o usuário tutor deve acompanhar as ações do operador durante a execução da manobra. Apesar de não estar explicitamente indicado no roteiro de tarefa (Ver Apêndice C), o tutor tem a opção de abrir o supervisor para visualizar a mudança de estado dos equipamentos e os alarmes que são gerados.

Usuário Treinando: Executar o roteiro de manobra

O usuário é solicitado a realizar a manobra de “Liberação do 14V2”. Para isso, ele deverá realizar a sequência de passos enumerados a baixo.

- I. Realizar login no supervisor e acessar o “Visor de Telas”.
 - a. Usuário: operador
 - b. Senha: 1234
 - c. Confirmar se o disjuntor 14D1 está aberto, na linha LT04V2 (Diagrama Unifilar de 230 kV)
- II. Fechar 34V2-6, no painel 04V2.
 - a. Na “Tela de Alarmes” do supervisor, visualizar que um alarme foi gerado em decorrência do fechamento do 34V2-6.
 - b. Reconheça esse alarme clicando no botão “Reconhecer”.
- III. Colocar chave 43 -14D1 na posição 'TRANSFERÊNCIA', no painel 14D1.
- IV. Colocar chave 43 -14V2 na posição 'TRANSFERÊNCIA', no painel 04V2.
- V. Fechar disjuntor 14D1 da linha 04V2 no supervisor.
- VI. Abrir disjuntor 14V2 da linha 04V2 no supervisor.
- VII. Colocar chave CLT-14V2 na posição 'LOC' no painel 04V2.
- VIII. Abrir seccionadoras 34V2-4 e 34V2-5 no painel 04V2.
- IX. Acessar a tela de alarmes do supervisor.
 - a. Visualizar os alarmes que foram gerados em decorrência da mudança de estado dos equipamentos.
 - b. Reconhecer os alarmes clicando no botão “Reconhecer”.
- X. Informar conclusão liberação 14V2

A Figura 5-2 representa a linha de transmissão LT04V2 após a realização da manobra, com o disjuntor 14V2 liberado.



Figura 5-2 – Representação da linha de transmissão 04V2 após a realização da manobra de liberação do disjuntor 14V2

Usuário Treinando: Alarmes de Eventos

O usuário deve identificar o painel onde um alarme sonoro foi disparado pelo tutor e depois desligá-lo. Para ajudar a identificar o equipamento associado ao alarme/evento, foi sugerido o acesso a tela de alarmes do supervisor.

5.3 Análise dos resultados

Nesse experimento foi avaliado a correteza das informações apresentadas nas telas do supervisor e a consistência dos dados. Os seguintes pontos foram observados nas sessões de teste e confirmados pelos usuários em um questionário pós-teste:

- os comandos efetuados na tela do sinótico foram corretamente executados pelo simulador;
- os alarmes gerados na tela de alarmes correspondem aos eventos observados nos painéis;
- os estados dos dispositivos representados nos painéis e no supervisor são concordantes;
- não houve erro de comunicação entre o supervisor e o ambiente virtual.

Os usuários realizaram todas as tarefas com sucesso e as intervenções dos avaliadores durante o teste serviram apenas para tirar dúvidas relacionadas à compreensão do roteiro de tarefas. Sendo que apenas um dos usuários possuía experiência prévia no uso do supervisor do simulador, os demais foram informados e assistiram a uma breve demonstração dos recursos oferecidos pelo simulador e, em especial, pelo supervisor. Esse fato comprova a facilidade de aprendizado e interação com o sistema.

O UT1, que desempenhou o papel de tutor na primeira sessão de teste, declarou preferir usar o supervisor na tarefa de configuração do cenário de teste em detrimento das outras formas possíveis (painéis e menus de configuração). De fato, no supervisor o tutor tem disponível uma representação gráfica do cenário (diferenciando-se dos menus) o que facilita a compreensão e a verificação do processo de configuração. Além disso, em uma única tela o tutor pode atribuir o estado inicial de todos os dispositivos envolvidos no cenário, sem precisar se deslocar pelo ambiente virtual a procura dos painéis de interesse.

Apesar disso, o UT1 demonstrou insatisfação com o tempo necessário para inicializar o supervisor (de 5 a 10 segundos) e comentou: “A primeira vez que abro o supervisor é mais demorado”. Porém, quanto ao tempo de resposta da interface, necessário para atualização dos estados dos dispositivos representados na tela, o UT1 considerou adequado.

O UT2, que na primeira sessão atuou como tutor e na segunda como operador, classificou como baixo o nível de dificuldade na realização de todas as tarefas, o que demonstra a facilidade de uso do simulador. Porém, considerou ruim a identificação dos dispositivos nos painéis de controle e ressaltou que identificá-los é mais fácil no supervisor.

Quanto à navegação entre o ambiente virtual e o supervisor, o UT2 classificou como adequada e intuitiva. Também considerou fácil o acesso às telas do supervisor a partir do monitor da estação de trabalho, representado na sala de controle virtual.

Em relação ao visor de alarmes do supervisor, o UT2 observou que a lista de alarmes seguia uma ordem alfabética, de acordo com os nomes dos dispositivos associados aos alarmes. Entretanto, os alarmes deveriam ser apresentados na ordem de ocorrência (sequência cronológica), como acontece nos sistemas reais. Essa falha foi corrigida após o teste.

O UT3, que atuou no papel de operador na segunda sessão de teste, declarou que o cenário executado foi consistente, completo e bastante realista. O usuário comentou após o experimento: “Me senti realmente envolvido com a situação”.

Na ocorrência do alarme sonoro, o UT3 demonstrou nervosismo e irritação porque não conseguia localizar o painel associado ao evento, onde ele deveria desligar o som do alarme. E comentou que identificou facilmente o dispositivo que gerou o alarme na tela de alarmes do supervísório, mas demorou a localizá-lo em um painel da sala de controle.

Pelo exposto, observou-se que o supervísório foi capaz de representar e simular a execução de uma manobra real. Ficou comprovada a integração do supervísório com o ambiente virtual, mantendo-se a consistência dos dados e a corretude das informações.

Capítulo 6 Considerações finais

Nesse trabalho, foi desenvolvido um módulo supervisorio com as características dos supervisórios usados na operação de subestações elétricas e integrado a um simulador para treinamento de operadores.

Esse simulador possui um ambiente de simulação descrito em realidade virtual e construído com base na sala de controle de uma subestação elétrica. Assim, estão retratados no ambiente virtual todos os objetos utilizados pelo operador da subestação para realização de suas tarefas. Dentre esses objetos se encontram: painéis de controle, chaves diversas, quadro de eventos, mostradores analógicos e digitais.

A interface do simulador permite que o usuário possa se deslocar no ambiente virtual e interagir com os objetos nele representados, simulando a realização das tarefas que faria no ambiente real. Entretanto, antes do desenvolvimento desse trabalho, a representação do software supervisorio no simulador era apenas ilustrativa e ainda inadequada para o propósito de treinamento.

Na solução proposta, as telas do supervisorio, usadas por operadores em uma subestação da CHESF, foram reproduzidas em um software comercial que é iniciado a partir do ambiente virtual da sala de controle. Dessa forma, os usuários podem realizar parte dos procedimentos de uma tarefa interagindo com objetos dos painéis de controle (ambiente virtual) e iniciar o software supervisorio para realizar os procedimentos restantes ou confirmar os resultados das ações efetuadas nos painéis.

Assim, permitiu-se que o operador, em uma sessão de treinamento, possa utilizar indistintamente quaisquer dos dois níveis de controle para execução de manobras no simulador. Exatamente como ocorre na rotina de operação de uma subestação elétrica.

Apesar de não ser exatamente o mesmo supervisorio usado por operadores de uma subestação, a reprodução das telas em outro sistema supervisorio comercial (*InTouch*) facilitou o desenvolvimento, com as réplicas muito semelhantes as telas reais.

Alternativamente, considerou-se a possibilidade de se recriar as telas do supervisorio no próprio ambiente virtual, utilizando-se a linguagem X3D. Entretanto, adotar essa solução implicaria em um esforço de programação muito elevado para alcançar níveis aceitáveis de

realismo na representação das telas. Isso se deve à necessidade de modelar no mundo virtual a representação de uma grande variedade de telas com seus elementos e objetos de interação. Tipicamente, em um sistema supervisão real, os projetistas de interface dispõem de uma vasta biblioteca de objetos com comportamentos previamente definidos, e programam seus comportamentos utilizando recursos disponibilizados pelo ambiente.

Usuários de teste, que conheciam as telas do supervisão que foram reproduzidas, consideraram as réplicas bastante realistas, embora ainda seja necessária a realização de testes com operadores reais, que estão habituados a interagir com esses sistemas. Esses testes podem revelar distorções de comportamento dos objetos representados nas telas que foram desenvolvidas e auxiliar no refinamento do projeto.

As telas que compõem a interface do ambiente supervisão dividem-se em três grupos: as telas de acesso, que permitem o *login* do usuário e a navegação para outras telas, as telas com diagramas unifilares, que resumem o estado do sistema elétrico que está sendo supervisionado, e a tela de alarmes que reúnem os alarmes gerados em decorrência de eventos no sistema.

Segundo os próprios operadores, e também observado durante a realização de visitas técnicas a subestações da CHESF, essas são as telas mais frequentemente usadas por eles na operação da subestação. Embora, outras telas possam ainda ser desenvolvidas e incorporadas a esse ambiente supervisão, a exemplo de telas de log, telas de oscilografia, telas de configuração e outras.

No instante em que se realiza a chamada de execução do supervisão, as variáveis são iniciadas a partir de consultas ao banco de dados do simulador. Depois disso, o supervisão passa trocar dados com um módulo de comunicação que gerencia o tráfego de mensagens. Para permitir o envio e o recebimento de mensagens com o supervisão, foi desenvolvido um *driver* de comunicação baseado no protocolo DDE (*Dynamic Data Exchange*), suportado pelo supervisão onde as telas foram projetadas e programadas, o *InTouch*.

A partir da associação do supervisão com o banco de dados do sistema, permitiu-se que o ambiente supervisão representasse cenários de treinamento configurados e armazenados pelo usuário tutor, um dos objetivos específicos nesse trabalho. E, através do *driver* de comunicação que foi desenvolvido, permitiu-se que os comandos efetuados na tela do supervisão fossem enviados ao motor de simulação do sistema, outro objetivo específico considerado nesse trabalho.

A propósito, a composição do motor de simulação do sistema, a partir de modelos CPN, apresenta-se como uma solução mais flexível que a codificação em uma linguagem de programação. Com o formalismo de redes de Petri foi também possível realizar a verificação e a validação dos modelos constituintes.

Na solução proposta, também considerou-se que o simulador é um ambiente multiusuário, ou seja, em uma sessão de treinamento, existem pelo menos dois usuários envolvidos, o tutor e o operador. Assim, as ações do operador, em qualquer um dos dois níveis de controle, são transmitidas ao ambiente do tutor, e vice-versa. Para o usuário tutor, em particular, o supervisório facilitou a tarefa de configuração dos estados iniciais dos dispositivos envolvidos em um cenário de treinamento.

As funcionalidades do módulo supervisório foram avaliadas através da realização de dois testes com usuários em uma situação de treinamento. Os testes foram realizados no laboratório de usabilidade da UFCG utilizando critérios adotados em testes de usabilidade, embora o propósito principal tenha sido avaliar a se as funcionalidades dos módulos estavam completas e corretas. Nos experimentos realizados, os usuários atuaram tanto no papel de tutor, como no de um operador em treinamento.

O cenário de teste envolvia uma manobra frequentemente realizada em subestações, a liberação de um disjuntor, e todas as funcionalidades propostas para o módulo supervisório foram testadas. Por outro lado, considerações a respeito da usabilidade do módulo supervisório demandariam a realização de novos testes, com uma quantidade maior de usuários.

A partir da revisão bibliográfica realizada neste trabalho, constatou-se que existe a preocupação e o interesse das empresas do setor elétrico em treinar seus operadores no uso de software de supervisão, que são sistemas complexos. Inclusive, alguns sistemas com o propósito de treinamento no uso de supervisórios foram citados nesse trabalho. Entretanto, esses sistemas não estão integrados a uma representação do ambiente de trabalho do operador, nem tão pouco possibilita a realização de manobras em painéis de controle.

Por outro lado, foi constatado que os operadores não podem realizar todas as suas tarefas em apenas um dos níveis de controle, sendo necessário que, numa mesma manobra no sistema, alguns comandos sejam efetuados nos painéis e outros no supervisório. E o deslocamento do operador de um nível de controle para o outro é constante e essencial.

Portanto, pode-se concluir que a realização de sessões de treinamento no uso sistemas supervisórios é fundamental para a formação do operador de subestações elétricas. E, como

resultado deste trabalho, o simulador do LIHM pode apoiar a realização de sessões de treinamento mais abrangentes e realistas que a maioria dos simuladores disponíveis atualmente, envolvendo cenários em que o treinando deve simular a operação em painéis de controle e no supervisão.

Trabalhos futuros

Como propostas de continuidade desse trabalho destacam-se os seguintes temas:

- expandir o “visor de telas” do supervisão, representando um conjunto maior de linhas de transmissão e permitindo que outras manobras sejam realizadas;
- desenvolver o “visor de log” do supervisão e associá-lo ao banco de dados do simulador;
- expandir o modelo da planta, incluindo novos equipamentos e dispositivos encontrados no pátio de subestações, a exemplo de transformadores, banco de capacitores, dentre outros. A partir desta extensão pretende-se construir uma biblioteca de modelos de equipamentos e dispositivos que facilite a representação no simulador de uma variedade de instalações;
- investigar a adequação de outros motores de simulação para o projeto do simulador. Especialmente aqueles em que o comportamento do sistema elétrico é simulado através do cálculo do fluxo de potência;
- propor um método de desenvolvimento que facilite a adaptação de simulador para diferentes contextos de aplicação.

Referências

BEZERRA, H. et al. Simulador para Treinamento Presencial e a Distância de Proteção e Operação de Sistemas Elétricos In: Encontro Regional Ibero-Americano Do CIGRÉ, 12., 2007, Foz do Iguaçu. **Anais do Décimo Segundo Encontro Regional Ibero-americano do CIGRÉ**. Foz do Iguaçu: XII Eriac, 2007. p. 1-8.

BRUTZMAN, Don; DALY, Leonard. **X3D: Extensible 3D Graphics for Web Authors**. California: Morgan Kaufmann Publications, 2007.

CEPEL (Org.). **SAGE - Sistema Aberto de Gerenciamento de Energia**. Disponível em: <<http://www.sage.cepel.br/>>. Acesso em: 10 jan. 2011.

FREITAS, Rodrigo Choji de. **Mecanismo para visualização e comunicação bidirecional entre modelos em redes de petri coloridas e modelos em realidade virtual**. 2006. 104 f. Dissertação (Mestrado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2006.

HUA, Bin; ZHOU, Jianzhong; YU, Jing. Integration of exist SCADA/EMS with dispatcher training simulator system. In: POWER SYSTEMS CONFERENCE AND EXPOSITION, 1., 2004, New York. **IEEE Proc. on Power Systems Conference and Exposition**. New York: Ieee, 2004. v. 2, p. 829 - 838.

ISASI, Ramon Antonio G. et al. Treinamento na Operação: A Experiência da Usina Hidrelétrica ITAIPU Binacional. In: Encontro Regional Ibero-Americano Do CIGRÉ, 12., 2007, Foz do Iguaçu. **Anais do Décimo Segundo Encontro Regional Ibero-americano do CIGRÉ**. Foz do Iguaçu: XII Eriac, 2007. p. 01 - 10.

JENSEN, Kurt. **Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use**. 2. ed. Berlin; Heidelberg; New York; Barcelona; Budapest; Hong Kong; London; Milan; Paris; Santa Clara; Singapore; Tokyo: Springer, 1997. 1 v.

KOSTIC, Tatjana et al. Decision Aid Function for Restoration of Transmission Power Systems: Conceptual Design and Real Time Considerations. In: IEEE CONFERENCE ON POWER INDUSTRY COMPUTER APPLICATIONS, 20., 1997, Ohio. **IEEE Proc. on 20th Conference on Power Industry Computer Applications**. Ohio: Ieee, 1997. p. 112 - 118.

LEITE, Carlos Roberto R.; RODRIGUES, João José; OLIVEIRA, Jaldemir Gomes de. O Uso de Simuladores no Treinamento de Operadores da CHESF como Ferramenta para Disseminação de Conhecimentos na Operação do Sistema Elétrico. In: Seminário internacional: reestruturação e regulação do setor de energia elétrica e gás natural, 2., 2007, Rio de Janeiro. **Anais Eletrônicos do II Seminário Internacional: Reestruturação e Regulação do Setor de Energia Elétrica e Gás Natural**. Rio de Janeiro: UFRJ, 2007. p. 1-27. Disponível em: <http://www.nuca.ie.ufrj.br/gesel/eventos/seminariointernacional/2007/artigos/pdf/carlosrobertoleite_ousodesimuladores.pdf>. Acesso em: 10 jan. 2011.

LORENZO, Mário Lúcio de. et al. Desenvolvimento e Preparação de Operadores do SIN - ONS. In: Encontro para debates de assuntos de operação, 10., 2008, São Paulo. **Anais do X Encontro para debates de assuntos de operação**. São Paulo: X EDAO, 2008. p. 1 - 8.

MAMEDE FILHO, João. **Instalações elétricas industriais**. 7. ed. Rio de Janeiro: LTC, 2007.

MIRANDA, Antonio Ricardo C.; VIEIRA, Tarcísio Vasconcelos; OLIVEIRA, Jaldemir. Experiência da CHESF com treinamentos simulados de operadores, através da utilização de simuladores interligados e simulação de manobras reais nas instalações e centros de operação. In: Encontro para debates de assuntos de operação, 10., 2008, São Paulo. **Anais do X Encontro para debates de assuntos de operação**. São Paulo: X EDAO, 2008. p. 1 - 6.

MIRANDA, Antonio Ricardo C.; VIEIRA, Tarcísio Vasconcelos; OLIVEIRA, Jaldemir. Experiência da CHESF com treinamentos simulados de operadores, através da utilização de simuladores interligados e simulação de manobras reais nas instalações e centros de operação. In: Encontro para debates de assuntos de operação, 10., 2008, São Paulo. **Anais do X Encontro para debates de assuntos de operação**. São Paulo: X EDAO, 2008. p. 1 - 6.

MOREALE, Michel Dos Santos. **Técnicas para treinamento de operadores de sistema elétrico utilizando simulador com base na interface de tempo real**. 2007. 127 f. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2007.

MOUTINHO, Joaquim Américo Pinto. Simulador de sistemas de proteção, controle e supervisão: uma ferramenta para capacitação da operação e manutenção. In: Encontro para debates de assuntos de operação, 10., 2008, São Paulo. **Anais do X Encontro para debates de assuntos de operação**. São Paulo: X EDAO, 2008. p. 1 - 5.

NASCIMENTO NETO, José Alves do. **Modelagem da Interface Homem-Máquina de uma Subestação Elétrica**. 2004. 128 f. Dissertação (Mestrado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2004.

NASCIMENTO NETO, José Alves do. **Processo para concepção de estratégias para prevenção do erro na operação de sistemas elétricos**. 2010. 139 f. Tese (Doutorado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2010.

SANTOS, Sergio Velho B. G. dos; CARMO, Ubiratan Alves do. Revisão do plano diretor de automação da CHESF (PDAO) face novo modelo do setor elétrico e experiência da CHESF na implantação do mesmo. In: Encuentro Regional Iberoamericano De CIGRÉ, 11, 2005, Ciudad Del Este. **Anales del undécimo encuentro regional iberoamericano de CIGRÉ**. Ciudad Del Este: XI ERIAC, 2005. p. XI/B5-01/01-06.

SCAICO, Alexandre. **Aplicação de métodos formais no projeto de interfaces para sistemas industriais críticos**. 2007. 150 f. Tese (Doutorado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2007.

SILVA NETTO, Ademar Virgolino da. **Arquitetura para um ambiente de treinamento representado em realidade virtual**. 2010. 132 f. Dissertação (Mestrado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2010.

SILVA, Victor Navarro A. L. Da. et al. Simuladores para treinamento de operadores de sistema e de instalações do setor elétrico. In: Encuentro Regional Iberoamericano De CIGRÉ, 13, 2009, Puerto Iguazú. **Anales del décimo tercer encuentro regional iberoamericano de CIGRÉ**. Puerto Iguazú: XIII Eriac, 2009. p. XIII/PI-C2-04/01-10.

TORRES FILHO, Flávio; VIEIRA, Maria de Fátima Queiroz. Motor de simulação baseado em modelos CPN aplicado a um sistema para treinamento de operadores. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 18., 2010, Bonito. **Anais do XVIII Congresso Brasileiro de Automática**. Bonito: CBA2010, 2010. p. 1252 - 1259.

VIEIRA, Maria F. Q. et al. A Real-time Interface Simulator for Operator Training: A Proposed Architecture. **Simulation**, 86: p. 53-63. 01 Jan. 2010. Disponível em: <<http://sim.sagepub.com/content/86/1/53>>. Acesso em: 10 jan. 2011.

WALVE, Kenneth; ADSTROM, Andres. The training simulator ARISTO-design and experiences. In: Power Engineering Society 1999 Winter Meeting, 1999, New York. **IEEE Proc. on Power Engineering Society Winter Meeting**. New York: IEEE, 1999. v. 1, p. 545 - 547.

WEB3D CONSORTIUM. **Web3D Consortium: Open Standards for Real-Time 3D Communication**. Disponível em: <<http://www.web3d.org/>>. Acesso em: 20 jan. 2011.

WONDERWARE. **InTouch HMI da Wonderware**. A Wonderware é líder de mercado em softwares para a gestão de operações em tempo real. Disponível em: <<http://global.wonderware.com/BR/Pages/WonderwareInTouchHMI.aspx>>. Acesso em: 12 jan. 2011.

A.2 InTouch

O *InTouch* é uma das ferramentas SCADA mais usadas na indústria. Seus módulos principais são:

- O *WindowMaker* – ambiente para a criação, edição e programação das telas;
- O *WindowViewer* – ambiente configurável que permite a execução das aplicações;
- O *Tagname Dictionary* – repositório com todas as variáveis da aplicação;
- Os *I/O Servers* – correspondem aos *drivers* de comunicação específicos para cada dispositivo de controle do mercado.

O *InTouch* oferece todos os recursos típicos para a construção de telas de operação e supervisão de processos industriais, tais como telas de sinópticos, gráficos de tendência, telas de alarmes, etc. A Figura A-2 ilustra a interface do ambiente de edição de telas do *InTouch*, o *WindowMaker*.

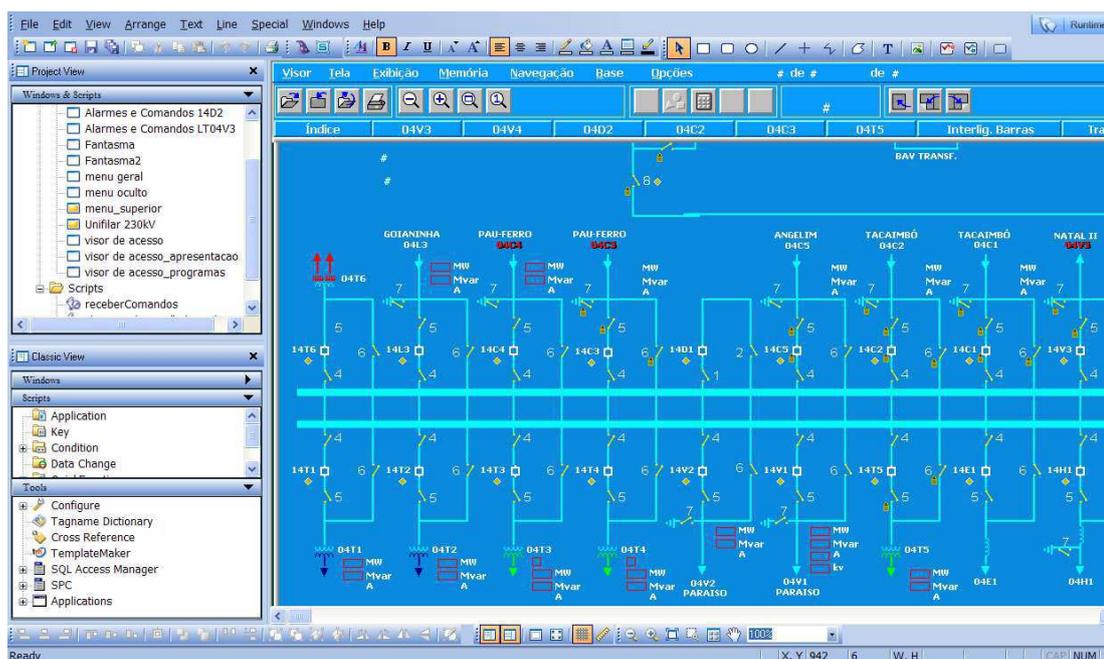


Figura A-2 – Tela com o diagrama unifilar da subestação

A.3 NetBeans

Ambiente de desenvolvimento integrado de aplicações em Java. Assim como o Eclipse , é um software gratuito, de código aberto e multi-plataforma, porém existe uma tendência de adoção do NetBeans devido às suas funcionalidades (Mobile Pack, NetBeans GUI Designer, NetBeans Plataforma, JRuby Editor, etc.) que facilitam a criação de interfaces gráficas, a comunicação com banco de dados, a depuração dos códigos, etc. Outro fator determinante para escolha do NetBeans como plataforma de desenvolvimento neste trabalho é sua documentação de boa qualidade, disponível em Português.

A.4 SQL Server

Sistema gerenciador de banco de dados (SGBD) usado na versão atual do simulador. Entre os fatores que influenciaram na escolha deste SGBD destaca-se a funcionalidade XML e a fácil integração com sistemas supervisórios, fundamental para o projeto.

A.5 X3D Edit

É um software gratuito que possui um conjunto de ferramentas que apóiam a criação, e edição de ambientes virtuais descritos em X3D. Dentre suas funcionalidades destacam-se a validação de arquivos X3D, menus com componentes X3D e um *browser Xj3D* integrado, que permite a visualização dos modelos tridimensionais.

Apêndice B: Modelo CPN do campo

Em subestações elétricas, os equipamentos localizados no pátio da subestação compõem a planta industrial. Estes poderão ser operados remotamente via programa supervisorio; via painéis na sala de controle, ou diretamente no seu respectivo painel de comando, no caso de falhas de comunicação. Neste trabalho foram modelados os dispositivos: disjuntores, chaves seccionadoras e as linhas de transmissão, compondo um modelo de planta simplificado.

O modelo CPN que representa o comportamento dos disjuntores é ilustrado na Figura B-1. Os lugares *Aberto*, *Fechado* e *Defeituoso* modelam o estado dos disjuntores. E as fichas que ocupam esses lugares são do tipo *Equipamento*. Estas fichas consistem de uma dupla (*LT*, *Disp*) na qual o primeiro elemento identifica a linha de tensão onde se encontra o dispositivo e segundo elemento identifica o disjuntor propriamente dito. Essa estrutura permite compor uma representação única para cada disjuntor no pátio da subestação.

As transições *Abrir* e *Fechar* representam a ação do operador ao manipular o equipamento, permitindo-lhe comandar a abertura e fechamento respectivamente. As transições *Falha_A* e *Falha_B* modelam ocorrências que deixam o dispositivo inoperante. Por fim, a transição *Evento* modela distúrbios na rede de distribuição que causam a abertura do disjuntor. As funções *send_campo_P* e *send_campo_S* enviam uma mensagem aos modelos: painel de comando e supervisorio, respectivamente; sempre que o disjuntor muda seu estado de *Aberto* para *Fechado* (ou vice-versa), mantendo a coerência entre os estados dos modelos. Na implementação dessas funções foram utilizadas outras funções da biblioteca COMMS/CPN, descrita na próxima sessão.

O modelo das chaves seccionadoras (Figura B-2) é semelhante ao modelo que representa o comportamento de disjuntores, já apresentado. Entretanto, este modelo possui os lugares *Abrindo* e *Fechando*, os quais representam estados transitórios durante as operações de abertura e fechamento desses equipamentos. O lugar de fusão *DJ_Aberto* é necessário para garantir que uma chave seccionadora seja aberta apenas quando o disjuntor, a ela associado na mesma linha de tensão, já estiver aberto.

Na figura B-3 é apresentado o modelo CPN para linhas de transmissão. Nesse modelo, as linhas de transmissão possuem os dois estados, *Energizada* e *Desenergizada*. E a mudança de estado depende da situação dos equipamentos diretamente ligados à linha.

Uma linha de transmissão energizada deverá ter necessariamente as chaves seccionadoras e disjuntores fechados. E para efetivar a mudança de estado de uma linha de *Energizada* para *Desenergizada* o disjuntor deve estar aberto.

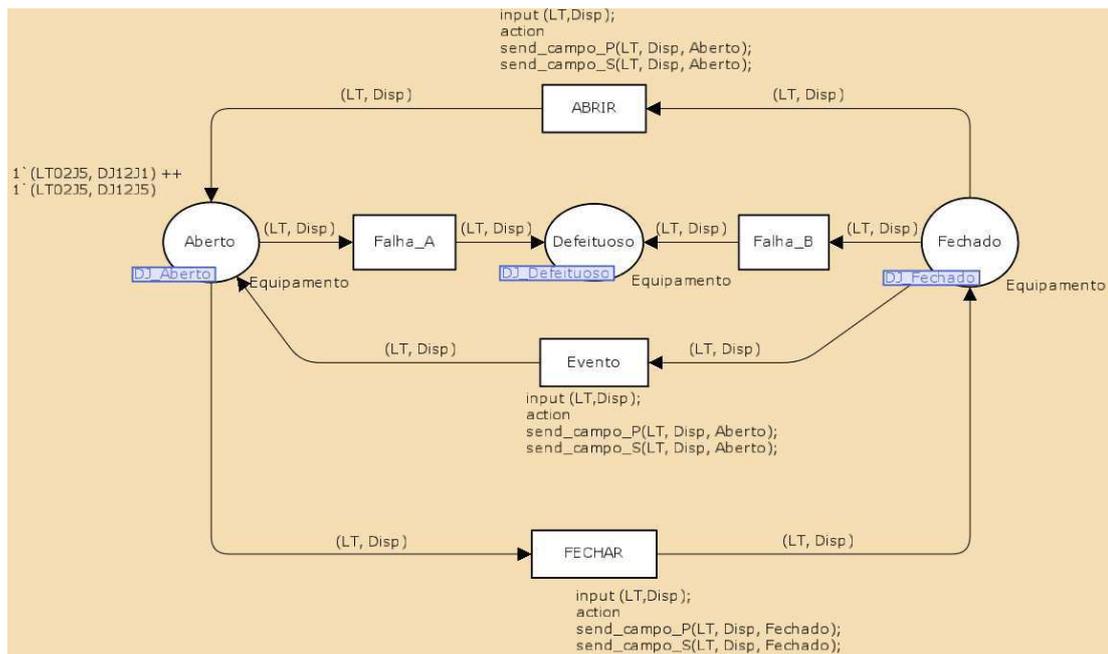


Figura B-1: Modelo de Disjuntores

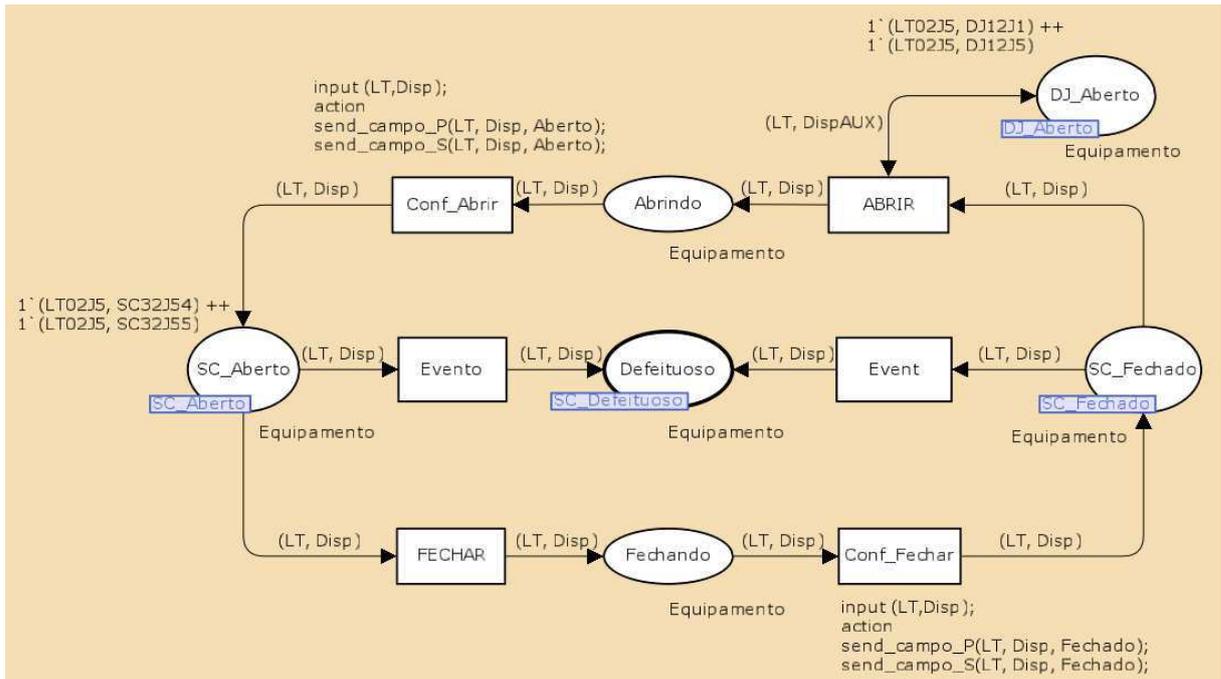


Figura B-2: Modelo de chaves seccionadoras

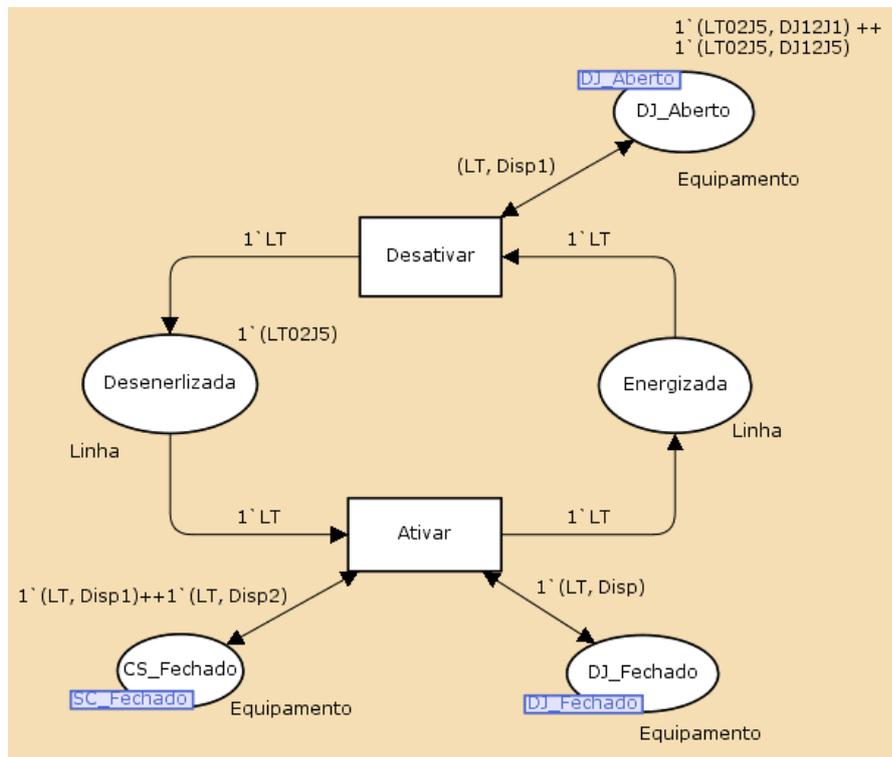


Figura B-3: Modelo de linhas de transmissão

A Figura B-4 ilustra uma página do modelo da planta, cuja sequência de lugares e transições promove o recebimento de mensagens oriundas dos modelos de interface. Esta rede consiste de duas partes: um segmento relativo ao recebimento de mensagens vindas do

modelo dos painéis (à esquerda) e outro relativo ao recebimento de mensagens vindas do modelo do supervisor (à direita).

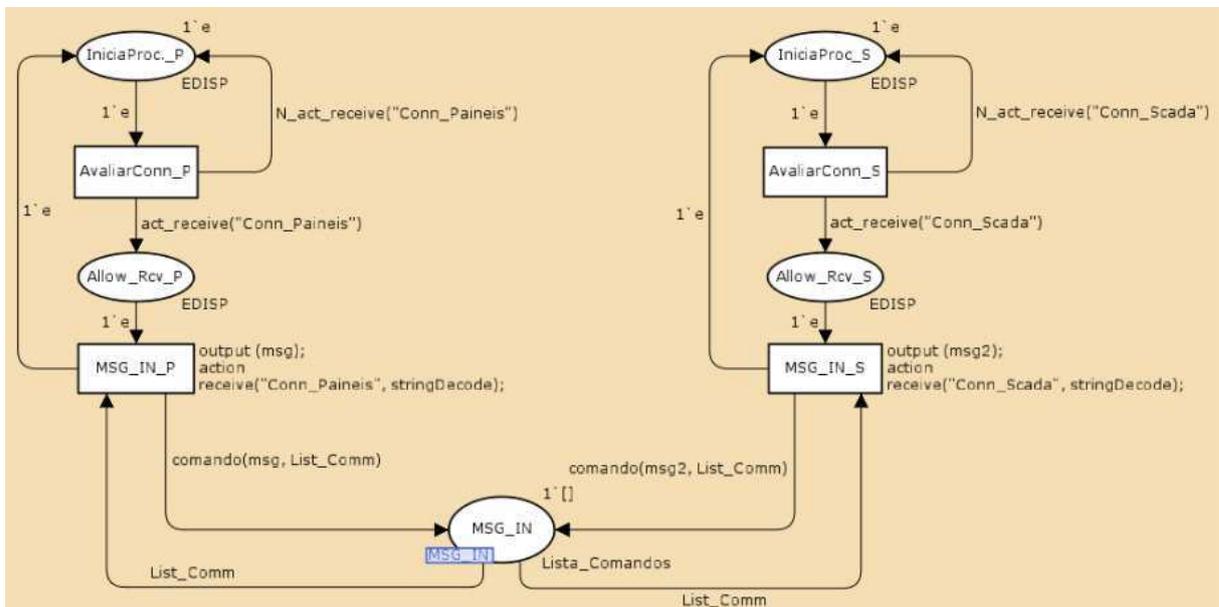


Figura B-4: Página do modelo de campo para o recebimento de mensagens enviadas pelos modelos de interface

A seguir, são descritos os componentes desta rede.

- Lugares *IniciaProc_P* e *IniciaProc_S*: responsáveis por habilitarem as transições *AvaliarConn_P* e *AvaliarConn_S*, respectivamente, quando a rede entra no modo de simulação.

- Transições *AvaliarConn_P* e *AvaliarConn_S*: que ao serem disparadas, causam a execução das funções associadas aos arcos de saída e avaliam se há um novo dado para ser recebido no canal de comunicação correspondente. Se houver, a função *act_receive* retorna uma ficha do tipo *EDISP* e a função *N_act_receive* retorna *empty*. Não existindo um novo dado, a função *act_receive* retorna *empty* e a função *N_act_receive* retorna uma ficha do tipo *EDISP*.

- Lugares *Allow_Rcv_P* e *Allow_Rcv_S*: quando uma ficha ocupa esses lugares, as transições *MSG_IN_P* e *MSG_IN_S* são respectivamente habilitadas, permitindo que uma nova mensagem seja recebida na conexão correspondente.

- Transições *MSG_IN_P* e *MSG_IN_S*: quando uma dessas transições é disparada ocorre o recebimento de uma mensagem através do canal de comunicação correspondente.

As mensagens, ou comandos, recebidos ocupam uma lista no lugar de fusão *MSG_IN*, na ordem de chegada. Esse lugar aparece em outras duas páginas do modelo, nas quais a execução dos comandos recebidos do meio externo ocorre. *Proc.MSG(DJ)* e *Proc.MSG(Sec)* realizam o tratamento dos comandos de abertura ou de fechamento, relativos aos disjuntores e chaves seccionadoras, respectivamente.

Na Figura B-5 é ilustrada a rede da página *Proc.MSG(DJ)*. Os lugares *Aberto* e *Fechado* representam os estados dos disjuntores e realizam a fusão com os lugares homônimos da Figura B-1.

A depender da lista de comandos, que ocupa o lugar *MSG_IN*, poderão ser habilitadas as transições *ABRIR* ou *FECHAR*, as quais realizam, respectivamente, a abertura e o fechamento de disjuntores. Ou seja, com o disparo de uma dessas transições um comando realizado sobre um painel, ou no supervisor, é executado. No instante do disparo, os modelos da IHM, painéis e supervisor, são também realimentados com o novo estado do dispositivo. Isso ocorre através da execução das funções associadas a estas transições: *send_campo_P* e *send_campo_S*.

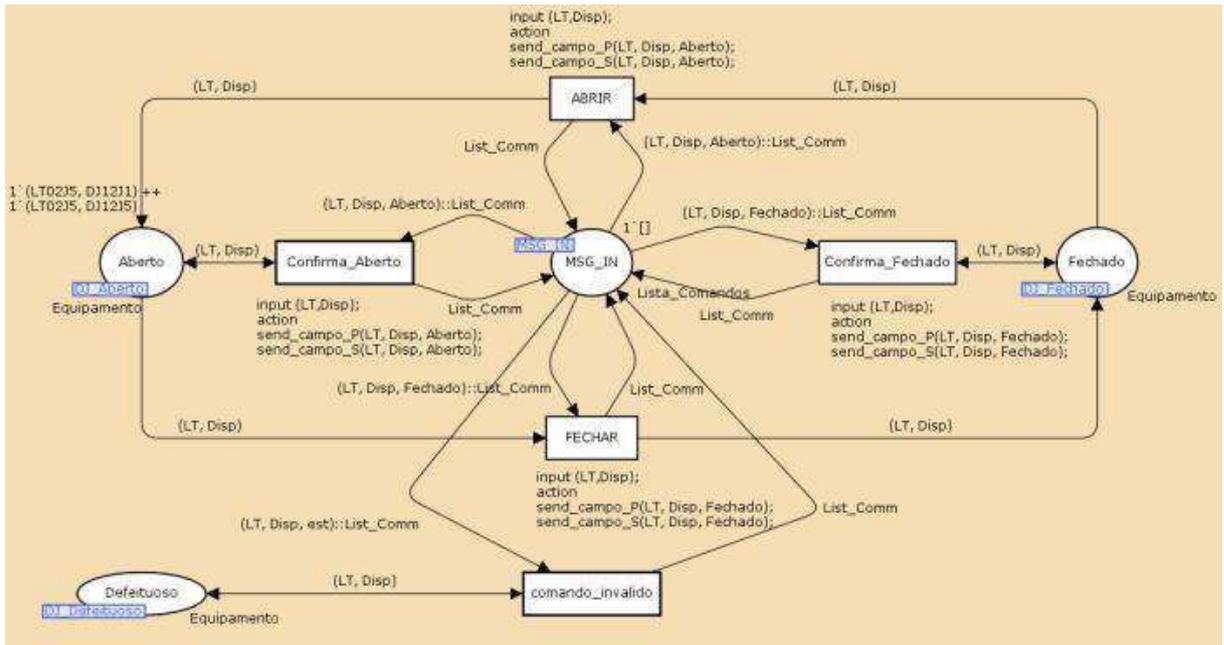


Figura B-5: Página do modelo de campo para execução de comandos nos disjuntores

Se ocorrer a situação na qual um comando causa um estado no qual o dispositivo já se encontra, as transições *Confirma_Aberto* e *Confirma_Fechado*, retiram o comando da lista e confirmam o estado do dispositivo para os modelos da interface.

Se for recebido um comando para um dispositivo que se encontre no estado *Defeituoso* a transição *Comando_Invalido* é habilitada e ao disparar retira o comando da lista. Como resultado, o comando não é executado e nenhuma mensagem é enviada aos modelos da IHM.

Apêndice C: Planejamento e artefatos do teste de validação

	<input type="checkbox"/> Hardware	<input checked="" type="checkbox"/> Software		
	<input type="checkbox"/> Outro _____			
Atividade				
Descrição da atividade	<p>- O usuário treinando deve simular a realização de manobras no sistema elétrico, interagindo com painéis virtuais e um sistema supervisorio;</p> <p>- O usuário tutor deve montar um cenário de treinamento, acompanhar as ações do treinando e também pode interagir com os painéis e com o supervisorio.</p>			
3 Realização da atividade	<input checked="" type="checkbox"/> Individual	<input type="checkbox"/> Coletiva		
Envolvidos na realização da atividade	Um usuário treinando. Integrante do LIHM atuando com Tutor.			
É necessário o uso de uma linguagem específica?	<input checked="" type="checkbox"/> Sim	<input type="checkbox"/> Não		
Em caso afirmativo, qual?	Própria dos operadores de subestações			
É necessário o uso de equipamentos auxiliares?	<input type="checkbox"/> Sim	<input checked="" type="checkbox"/> Não		
Em caso afirmativo, qual(is) e para quê?				
Universo amostral				
Descrição do perfil desejado (características)	Pessoas com conhecimentos básicos de sistemas elétricos; Operadores de subestações			
Descrição do perfil indesejado (características)	Pessoas sem os conhecimentos básicos de sistemas elétricos.			
Categoria dos usuários de teste	<input checked="" type="checkbox"/> Inexperiente			
	<input type="checkbox"/> Intermediário			
	<input checked="" type="checkbox"/> Experiente (operadores de subestações)			
4 Dimensão do universo amostral	3 usuários			
<i>Estratégia de recrutamento</i>				
Meio de contato	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Endereço	Telefone	Email	Outro. Qual?
<i>Treinamento</i>				
Existe a necessidade de treinamento do universo amostral?	<input checked="" type="checkbox"/> Sim	<input type="checkbox"/> Não		
Avaliação				
Objetivo geral	Validar a integração do módulo supervisorio ao ambiente do simulador.			
5 Objetivo específico	i. Verificar a facilidade de interação e navegação no módulo supervisorio;			
	ii. Analisar imersão do usuário durante a realização da manobra;			
	iii. Mensuração do nível de satisfação após a execução das tarefas.			
Natureza da avaliação	<input checked="" type="checkbox"/> Somativa		<input type="checkbox"/> Formativa	
	<input checked="" type="checkbox"/> Objetiva		<input checked="" type="checkbox"/> Subjetiva	
	<input checked="" type="checkbox"/> Qualitativa		<input checked="" type="checkbox"/> Quantitativa	
Hipóteses	i. A representação do supervisorio no ambiente de simulação é adequada;			
	ii. O tempo de resposta da interface é			

			suficiente e tolerável pelo usuário;
	iii.		As informações são apresentadas na tela em uma sequência coerente e agradável ao usuário;
	iv.		Não existe erro na comunicação do supervisor com demais módulos do simulador;
	v.		Os estados de todos os dispositivos estão coerentemente representados em todos os módulos do simulador;
	vi.		Todas as telas, objetos de interação e alarmes, necessários a realização da manobra escolhida, foram representados.
Tempo estimado		40 min	
Métodos de avaliação com usuário			
	<input checked="" type="checkbox"/> Observação direta	<input checked="" type="checkbox"/> Perfil do usuário	<input checked="" type="checkbox"/> Satisfação do usuário
Meta			
Objetivo			
Resultados previstos			
Natureza dos testes	<input checked="" type="checkbox"/> Em laboratório		
	<input type="checkbox"/> Em campo		
	<input type="checkbox"/> In loco		
Número de sessões de teste	2 testes efetivos + 1 teste piloto		
Ambiente de teste (descrição)	LIHM		
Tempo estimado	40 minutos		
Características a explorar	Pessoais (Nível de conhecimento em informática)		
	Profissionais (formação acadêmica, área de atuação)		
	Uso do produto (experiência no uso do produto ou similar)		
Métodos de avaliação com especialista			
	<input type="checkbox"/> Inspeção	<input type="checkbox"/> Heurísticas	<input type="checkbox"/> Guias
Características a explorar			
Planilha de responsabilidades			
Avaliador	Papel	Responsabilidades	
7	Flávio Torres	Avaliador	Elaborar o planejamento da avaliação;
			Elaborar roteiro de teste; Observador; Elaborar e aplicar questionários (pré- e pós-teste); Análise de dados.
	Raffael Carvalho	Avaliador	Observador; Aplicar questionários (pré- e pós-teste); Análise de dados.
	Fátima Vieira	Coordenadora	Elaborar o planejamento da avaliação; Elaborar roteiro de teste;

*Elaborar questionários (pré- e pós-teste);
Análise de dados.*

Cronograma

	Planejamento	21 – 25 de Janeiro
	Preparação do material do ensaio	28 – 02 de Fevereiro
8	Validação do ensaio (Teste Piloto)	03 de Fevereiro
	Realização do ensaio	04 de Fevereiro
	Análise dos dados	05 – 08 de Fevereiro
	Relato do ensaio	

Roteiro de tarefa de teste – Tutor

O produto a ser testado é um simulador para treinamento de operadores na sala de comando de subestação elétrica. Você é o tutor no treinamento e tem como função elaborar e configurar cenários de treinamento, gerar eventos de alarmes e acompanhar a execução dos treinamentos realizados pelos operadores. O roteiro está dividido em quatro tarefas.

Tarefa 01 – Configuração do Cenário

A primeira tarefa consiste em configurar o cenário de treinamento para que o operador possa executar a manobra. Para isto é fornecido um guia com a configuração de todos os objetos que devem ser configurados. A configuração consiste em conferir e atuar para que os objetos estejam no estado desejado.

Para efetuar esta configuração existem três modos possíveis, através da navegação no ambiente virtual, no menu lateral ou através do ambiente supervisorio. Como se pode observar no guia, em cada ação é definido onde você deve atuar e para que o teste seja realizado com sucesso é necessário que você siga todos os passos como descrito.

CONSULTAR GUIA DE MONTAGEM DO CENÁRIO.

Após configurar o cenário, o tutor deve salvar todas as alterações efetuadas.

Tarefa 02 – Iniciar conexão com Operador

Agora o tutor começará interagir com o operador. Inicialmente o tutor deve iniciar o servidor para que o operador possa se conectar. Após iniciar a conexão o tutor deve avisar ao operador que o servidor está iniciado e que ele deve tentar se conectar. Por fim o tutor deve identificar na aplicação que o operador está conectado e informar ao operador o sucesso da operação concluindo esta tarefa.

Tarefa 03 – Acompanhar o treinamento

Agora que o operador esta conectado, você deve dar informar ao operador para iniciar a execução da manobra e você deve acompanhar as ações executadas pelo operador e fazer as anotações que achar pertinente.

Roteiro de Manobras

Origem: CROL

Equipamento: 14V2-CGD

Motivo da Revisão: Substituição do 14D1

Configuração:

- 14D1 aberto com chaves associadas fechadas, 14V2 energizando LT 04V2 CGD/NTD e todas as chaves by Pass 230kV abertas.
- Chave 43T-CS do 14D1 na posição DESATIVADO.
- Chave 43T-C/I do 14D1 na posição REATOR.

1. LIBERACAO

- 1.1. CGD: Receber do responsavel solicitacao liberacao 14V2.
- 1.2. CGD: Solicitar CROL liberacao 14V2.
- 1.3. CROL: Informar COSR-NE liberaçao 14V2/CGD.
- 1.4. CROL: Autorizar CGD liberacao 14V2.
- 1.5. CGD: Confirmar 14D1 aberto.
- 1.6. CGD: Fechar 34V2-6.
- 1.7. CGD: Colocar chave 43 -14D1 na posição 'TRANSFERÊNCIA'.

- 1.8. CGD: Colocar chave 43 -14V2 na posição 'TRANSFERÊNCIA'.
- 1.9. CGD: Fechar 12D1.
- 1.10. CGD: Abrir 14V2. (INICIAR A TAREFA 04).
- 1.11. CGD: Colocar chave CLT-14V2 na posição 'LOC'.
- 1.12. CGD: Abrir 34V2-4 e 34V2-5.
- 1.13. CGD: Entregar 14V2 isolado ao responsável.
- 1.14. CGD: Informar CROL conclusão liberação 14V2.

Tarefa 04 – Gerar um evento de alarme

Quando o operador estiver executando a manobra e for executar a operação 1.10 que é: **Abrir o 14V2**. Você deve acionar o evento de alarme através da boteira de alarme no Painel 12J6 (botão branco na parte superior do painel) para identificar se o alarme foi devidamente acionado você ouvirá um alarme sonoro e o botão ficará com a luz acesa (amarelo).

Após o operador identificar a origem do alarme, ele lhe informará em qual painel o alarme foi acionado e você deve pedir para ele desligá-lo e continuar a executar a manobra, pois o evento poderá ser resolvido posteriormente.

Roteiro de tarefa de teste – Operador

O produto a ser testado é um simulador para treinamento de operadores na sala de comando de subestação elétrica. Você é um operador de subestação elétrica e tem como objetivo executar as manobras programadas no simulador, identificar e resolver problemas com a execução de manobras de emergência. O roteiro será dividido em três tarefas.

Tarefa 01 – Conectar ao Servidor

Inicialmente você deve esperar o contato do tutor para iniciar a conexão. Quando for permitido você deve tentar iniciar a conexão ao servidor, clicando na opção “Conexões > Conectar ao Servidor”.

Você pode verificar na barra de status (canto inferior esquerdo) se a conexão foi realizada com sucesso.

Tarefa 02 – Executar o roteiro de manobra

Como existe no simulador dois modos para executar a manobra (através do supervisório ou do ambiente 3D), no roteiro é descrito o local por onde você deve executar cada passo. A sequência deve ser seguida para que a tarefa seja realizada com sucesso.

Obs: Caso algum alarme sonoro seja disparado durante a execução dessa manobra, interrompa sua execução e passe para a próxima tarefa. Você poderá continuar esta tarefa após reconhecer o painel e desligar o alarme.

1. Liberação do disjuntor 14V2
 - 1.1. Realizar login no supervisório e acessar o “Visor de Telas”
 - 1.1.1. Usuário: operador
 - 1.1.2. Senha: 1234
 - 1.1.3. Confirmar se o disjuntor 14D1 está aberto, na linha LT04V2 (Unifilar de 230 KV)
 - 1.2. CGD: Fechar 34V2-6, no painel 04V2.
 - 1.2.1. Na “Tela de Alarmes” do supervisório, visualizar que um alarme foi gerado em decorrência do fechamento do 34V2-6.
 - 1.2.2. Reconheça esse alarme clicando no botão “Reconhecer”.
 - 1.3. Colocar chave 43 -14D1 na posição 'TRANSFERÊNCIA', no painel 14D1
 - 1.4. Colocar chave 43 -14V2 na posição 'TRANSFERÊNCIA', no painel 04V2
 - 1.5. Fechar disjuntor 14D1 da linha 04V2 no supervisório.
 - 1.6. Abrir disjuntor 14V2 da linha 04V2 no supervisório.
 - 1.7. Colocar chave CLT-14V2 na posição 'LOC' no painel 14V2.
 - 1.8. Abrir chaves seccionadoras 34V2-4 e 34V2-5 no painel 14V2.
 - 1.9. Acessar a tela de alarmes do supervisorio.
 - 1.9.1. Visualizar os alarmes que foram gerados em decorrência da mudança de estado dos equipamentos.
 - 1.9.2. Reconhecer os alarmes clicando no botão “Reconhecer”.

Tarefa 03 – Alarme de Eventos

Caso ouça algum alarme sonoro você deve identificar o painel ao qual o evento esta associado (para identificar o alarme no painel existe uma botoeira de alarme que esta localizada na parte superior do mesmo e estará acesa - indicação em amarelo), após identificar deve informar ao tutor e perguntar qual o procedimento a ser executado.

Se desejar, você também pode acessar a tela de alarmes do supervisório para tentar identificar o equipamento associado ao alarme/evento.

Apêndice D: Classes que gerenciam a troca de informações com o supervisor

```
/*
 * File: DdeDriver.c
 * Author: Flavio
 *
 * Created on December 29, 2010, 11:42 AM
 */

#include "windows.h"
#include "ddeml.h"
#include <stdio.h>
#include <sys/unistd.h>
#include "DdeDriver.h"

/*
 *
 */
DWORD idInst;
HSZ hszApp, hszTopic;
HCONV hConv;

HDDEDATA CALLBACK DdeCallback(
    UINT uType, // Transaction type.
    UINT uFmt, // Clipboard data format.
    HCONV hconv, // Handle to the conversation.
    HSZ hsz1, // Handle to a string.
    HSZ hsz2, // Handle to a string.
    HDDEDATA hdata, // Handle to a global memory object.
    DWORD dwData1, // Transaction-specific data.
    DWORD dwData2) // Transaction-specific data.
{
    return 0;
}

JNIEXPORT void JNICALL Java_Supervisorio_DdeDriver_inicializar
    (JNIEnv *env, jclass cl)
{
    idInst=0;
    UINT iReturn;

    iReturn = DdeInitialize(&idInst, (PFNCALLBACK) DdeCallback,
        APPCLASS_STANDARD | APPCMD_CLIENTONLY, 0) ;

    if (iReturn!=DMLERR_NO_ERROR)
    {
        printf("DDE Initialization Failed: 0x%04x\n", iReturn);
    }
}
```

```
        Sleep(1500);
    }
    else { /**printf("DDE Initialization ok\n");*/ }

}
JNIEXPORT void JNICALL Java_Supervisorio_DdeDriver_conectar
    (JNIEnv *env, jclass cl, jstring Japp, jstring Jtopic)
{
    const char* Capp;
    const char* Ctopic;

    Capp = (*env)->GetStringUTFChars(env, Japp, NULL);
    Ctopic = (*env)->GetStringUTFChars(env, Jtopic, NULL);

    hszApp = DdeCreateStringHandle(idInst, Capp, 0);
    hszTopic = DdeCreateStringHandle(idInst, Ctopic, 0);
    hConv = DdeConnect(idInst, hszApp, hszTopic, NULL);
    DdeFreeStringHandle(idInst, hszApp);
    DdeFreeStringHandle(idInst, hszTopic);
    if (hConv == NULL)
    {
        //printf("DDE Connection Failed.\n");
        Sleep(1500); DdeUninitialize(idInst);
    }
    else { /**printf("DDE Connection Ok.\n");*/ }

}

JNIEXPORT void JNICALL Java_Supervisorio_DdeDriver_enviar
    (JNIEnv *env, jclass cl, jstring Jitem, jstring Jdado)
{
    char* Citem;
    char* Cdado;

    Citem = (*env)->GetStringUTFChars(env, Jitem, NULL);
    Cdado = (*env)->GetStringUTFChars(env, Jdado, NULL);

    HSZ hszItem = DdeCreateStringHandle(idInst, Citem, 0);
    DdeClientTransaction((LPBYTE)Cdado, (DWORD)(lstrlen(Cdado)+1),
        hConv, hszItem, CF_TEXT,
        XTYP_POKE, 3000, NULL);
    DdeFreeStringHandle(idInst, hszItem);
}

JNIEXPORT jstring JNICALL Java_Supervisorio_DdeDriver_receber
    (JNIEnv *env, jclass cl, jstring Jitem)
{
    char* Citem;
    char Cdado[255];
    jstring Jdado;
```

```
Citem = (*env)->GetStringUTFChars(env, Jitem, NULL);

HSZ hszItem = DdeCreateStringHandle(idInst, Citem, 0);
HDDEDATA hData = DdeClientTransaction(NULL,0,hConv,hszItem,CF_TEXT,
    XTYP_REQUEST,5000 , NULL);
if (hData==NULL)
{
    //printf("Request failed: %s\n", Citem);
}
else
{
    DdeGetData(hData, (unsigned char *)Cdado, 255, 0);
}
Jdado = (*env)->NewStringUTF(env, Cdado);
return Jdado;
}

JNIEXPORT void JNICALL Java_Supervisorio_DdeDriver_desconectar
    (JNIEnv *env, jclass cl)
{
    //DDE Disconnect and Uninitialize.
    DdeDisconnect(hConv);
    DdeUninitialize(idInst);
    Sleep(3000);
    //printf("DDE Desconectado!!\n");

}

JNIEXPORT void JNICALL Java_Supervisorio_DdeDriver_testes
    (JNIEnv *env, jclass cl, jstring Japp, jstring Jtopic)
{
    printf("\nMétodo de testes...\n");

    const char* Capp;
    const char* Ctopic;

    Capp = (*env)->GetStringUTFChars(env, Japp, NULL);
    Ctopic = (*env)->GetStringUTFChars(env, Jtopic, NULL);

    printf("\n%s: %s\n", Capp, Ctopic);

}
}
```

```
/*
 *
 *
 */

package Supervisorio;

import java.io.File;
import java.util.logging.Level;
import java.util.logging.Logger;
import comunicacao.Servidor;
import comunicacao.EncodeDecode;

/**
 *
 * @author Flavio
 */
public class supervisorioTutor extends Thread {

    private String DIR_WindowsViewer = "C:\\\\WindowViewer.Ink";
    private String DIR_ProjetoIntouch = "C:\\\\Simulador"; //"C:\\\\SUPERVISÓRIO";
    private String status = "Closed"; //Closed or Running
    private String novaInteracao = "Inicio";
    private String novaMsgCampo = "Inicio";
    private String msgRecCampo = "";
    private String linha, disp, estado;
    private Servidor conector;
    private boolean homeSinotico = false, homeAlarmes = false;

    public void abrirVisorDeTelas () {

        try {
            Processo supervisorio = new Processo();

            supervisorio.abrir(DIR_WindowsViewer, DIR_ProjetoIntouch);
            homeSinotico = true; homeAlarmes = false;

            sleep(10000); //DELAY PARA APLICATIVO SER INICIADO
            //sleep(10);

        } catch (InterruptedException ex) {
            Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

}
```

```
public void abrirTelaDeAlarmes () {  
  
    try {  
        Processo supervisorio = new Processo();  
  
        supervisorio.abrir(DIR_WindowsViewer, DIR_ProjetoIntouch);  
        homeAlarmes = true; homeSinotico = false;  
  
        sleep(10000); //DELAY PARA APLICATIVO SER INICIADO  
        //sleep(10);  
  
    } catch (InterruptedException ex) {  
        Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);  
    }  
  
}  
  
//Estabelecer canal de comunicacao do Supervisorio com o campo (Rede CPN)  
public void conectarCampo () {  
  
    String hostName = "150.165.61.89";  
    int portaEnvio = 9092;  
  
    conector = new Servidor(hostName, portaEnvio);  
    conector.conectar2();  
  
}  
  
//Envia ao supervisorio msgs recebidas do campo  
public void recCampo (String msg) {  
  
    if(this.status.equals("Running")){  
        this.novaMsgCampo = "Ativado";  
        this.msgRecCampo = msg;  
    }  
  
}  
  
public void recTreinando(String msg) {  
  
    conector.envia(msg);  
}  
  
public void envCampo(String msg) {  
    conector.envia(linha + disp + estado);  
    System.out.println("Scada.novaInteracao: " + this.linha + ", " + this.disp + ", " +  
this.estado);  
  
}
```

```
public void recTreinando(String msg) {
    conector.envia(msg);
}

public Servidor getConector() {
    return conector;
}

public void iniciaAlarmes() {
    String SQL = "UPDATE dbo.Alarmes SET ativado='False', reconhecido='False' where
id=idDispositivo";

    bd.executarTransacao(SQL);
}

public void atualizarAlarmes(String dispositivo) {
    String[] atributos = bd.SelectAtributosDispositivo(dispositivo);
    String SQL = "UPDATE dbo.Alarmes SET ativado='True', reconhecido='False' where
id=" + atributos[0] + """;
    //System.out.println(SQL);
    bd.executarTransacao(SQL);
}

@Override
public void run() {
    try {
        System.out.println("A Thread supervisor foi iniciada!");

        DdeDriver.inicializar();

        DdeDriver.conectar("VIEW", "TAGNAME");

        while (true) {
            Thread.sleep(5000);
            this.status = DdeDriver.receber("status");

            if (this.status.equals("Running")) {
                if (homeSinotico) {
                    DdeDriver.enviar("showJanela", "telaDeUnifilares");
                    homeSinotico = false;
                }
                if (homeAlarmes) {
                    DdeDriver.enviar("showJanela", "telaDeAlarmes");
                    homeAlarmes = false;
                }
            }
        }
    }
}
```

```
        this.novaInteracao = DdeDriver.receber("novaInteracao");

        if (this.novaInteracao.contains("Ativado")) {

            this.linha = DdeDriver.receber("linha");
            this.disp = DdeDriver.receber("disp");
            this.estado = DdeDriver.receber("estado");
            this.novaInteracao = "Desativado";
            DdeDriver.enviar("novaInteracao", this.novaInteracao);
            this.envCampo(linha+disp+estado);

        }

        if (this.novaMsgCampo.equals("Ativado")) {
            DdeDriver.enviar("novaMsgCampo", novaMsgCampo); //Flag de nova
msg vinda do campo
            DdeDriver.enviar("msgRecCampo", msgRecCampo); //Msg vinda do
campo

            System.out.println("supervisorio.recCampo: " + msgRecCampo);
            this.novaMsgCampo = "Desativado";
        }
    } else {
        DdeDriver.desconectar();
        sleep(1000);
        DdeDriver.inicializar();
        DdeDriver.conectar("VIEW", "TAGNAME");
    }

}
} catch (InterruptedException ex) {
    Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);
}

System.out.println("A Thread supervisorio foi finalizada!");
}
}

/*
 *
 *
 */
package Supervisorio;

import java.util.logging.Level;
import java.util.logging.Logger;
import comunicacao.Servidor;
import comunicacao.EncodeDecode;
```

```
import interfaceTutor.jfTreinando;
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

/**
 *
 * @author Flavio
 */
public class supervisorioTreinando extends Thread {

    private String DIR_WindowsViewer = "C:\\\\WindowViewer.Ink";
    private String DIR_ProjetoIntouch = "C:\\\\Simulador";
    private String status = "Closed"; //Closed or Running
    private String novaInteracao = "Inicio";
    private String novaMsgCampo = "Inicio";
    private String msgRecCampo = "";
    private String linha, disp, estado;
    private Servidor conector;
    public static ObjectOutputStream output;
    private boolean homeSinotico = false, homeAlarmes = false;

    public void abrirVisorDeTelas() {

        try {
            Processo supervisorio = new Processo();

            supervisorio.abrir(DIR_WindowsViewer, DIR_ProjetoIntouch);
            homeSinotico = true;
            homeAlarmes = false;

            sleep(10000); //DELAY PARA APLICATIVO SER INICIADO

        } catch (InterruptedException ex) {
            Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

    public void abrirTelaDeAlarmaes() {

        try {
            Processo supervisorio = new Processo();

            supervisorio.abrir(DIR_WindowsViewer, DIR_ProjetoIntouch);
            homeAlarmes = true;
            homeSinotico = false;

            sleep(10000); //DELAY PARA APLICATIVO SER INICIADO
            //sleep(10);

        } catch (InterruptedException ex) {
            Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

}
```

```
}

//Envia ao supervisorio msgs recebidas do campo

public void recTutor(String msg) {

    if (this.status.equals("Running")) {
        this.novaMsgCampo = "Ativado";
        this.msgRecCampo = msg;
    }

}

public void envTutor (String msg) {

    try {

        output.writeObject(msg);
        output.writeObject("Supervisorio");
    } catch (IOException ex) {
        Logger.getLogger(supervisorioTreinando.class.getName()).log(Level.SEVERE,
null, ex);
    }

    System.out.println("Scada.novaInteracao: " + this.linha + ", " + this.disp + ", " +
this.estado);
}

public void initSocket(ObjectOutputStream outputTreinando) {
    output = outputTreinando;
}

@Override
public void run() {
    try {

        System.out.println("A Thread supervisorio foi iniciada!");

        DdeDriver.inicializar();

        DdeDriver.conectar("VIEW", "TAGNAME");

        while (true) {

            Thread.sleep(5000);
            this.status = DdeDriver.receber("status");

            if (this.status.equals("Running")) {
```

```
        if (homeSinotico) {
            DdeDriver.enviar("showJanela", "telaDeUnifilares");
            homeSinotico = false;
        }
        if (homeAlarmes) {
            DdeDriver.enviar("showJanela", "telaDeAlarmes");
            homeAlarmes = false;
        }

        this.novaInteracao = DdeDriver.receber("novaInteracao");

        if (this.novaInteracao.contains("Ativado")) {

            this.linha = DdeDriver.receber("linha");
            this.disp = DdeDriver.receber("disp");
            this.estado = DdeDriver.receber("estado");
            this.novaInteracao = "Desativado";
            DdeDriver.enviar("novaInteracao", this.novaInteracao);
            this.envTutor(linha+disp+estado);
            System.out.println("Scada.novaInteracao: " + this.linha + ", " +
this.disp + ", " + this.estado);
        }

        if (this.novaMsgCampo.equals("Ativado")) {
            DdeDriver.enviar("novaMsgCampo", novaMsgCampo); //Flag de nova
msg vinda do campo
            DdeDriver.enviar("msgRecCampo", msgRecCampo); //Msg vinda do
campo

            System.out.println("supervisorio.recCampo: " + msgRecCampo);
            this.novaMsgCampo = "Desativado";
        }
    } else {
        DdeDriver.desconectar();
        sleep(1000);
        DdeDriver.inicializar();
        DdeDriver.conectar("VIEW", "TAGNAME");
    }

}
} catch (InterruptedException ex) {
    Logger.getLogger(Supervisorio.class.getName()).log(Level.SEVERE, null, ex);
}
}
System.out.println("A Thread supervisorio foi finalizada!");
}
}
```