



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Unidade Acadêmica de Engenharia Elétrica

Ferramenta para Edição de Cenários Gerados pelo OTS do Ambiente Simulop

Rômulo Afonso Luna Vianna de Omena

*Relatório referente ao **Trabalho de Conclusão de Curso** submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande.*

Orientadora:

Profa. Dra. Maria de Fátima Queiroz Vieira

Campina Grande, Paraíba, Brasil

©Rômulo Afonso Luna Vianna de Omena, Julho de 2011

Trabalho de Conclusão de Curso

Relatório apresentado à Coordenação do Curso de Engenharia Elétrica da Universidade Federal de Campina Grande em cumprimento às exigências para a obtenção do título de Engenheiro Eletricista.

Rômulo Afonso Luna Vianna de Omena

Aluno

Profa. Dra. Maria de Fátima Queiroz Vieira

Orientadora

Campina Grande, Paraíba, Brasil

Julho de 2011

Índice

Índice	ii
Índice de Tabelas	v
Índice de Figuras	vi
1 Introdução	1
1.1 Treinamento de Operadores nos Sistemas Elétricos	1
1.2 Ferramentas que Apoiam o Treinamento	2
1.2.1 ASTRO	3
1.2.2 STPO	3
1.2.3 Simu <i>Lihm</i>	3
1.2.4 STriPP	4
1.2.5 SIMULOP	4
1.3 Objetivos do Trabalho	5
1.4 Metodologia	6
1.5 Apresentação do Texto	6
2 Cenários de Treinamento	7
2.1 Eventos no OTS	8
2.1.1 Como Criar	9
2.1.2 Executar	15
2.1.3 Exportar	15
2.1.4 Importar	16
2.1.5 Como Criar Eventos Condicionais	17
2.2 Limitações do Editor de Cenários do OTS	25
2.3 Conclusão	26

3	Construção do Editor de Cenários	27
3.1	Ferramentas Utilizadas	28
3.1.1	Construção do Editor	28
3.1.2	O Sistema de Gerenciamento de Banco de Dados MySQL	28
3.2	Classes e Métodos	30
3.3	Construção da Interface Gráfica	33
3.4	Conclusão	34
4	Editor de cenários no OTS do Simulop	35
4.1	Estrutura do Editor de Cenários do LIHM	36
4.2	Recursos do Editor	36
4.2.1	Criar Eventos	37
4.2.2	Copiar Eventos	39
4.2.3	Replicar Evento	40
4.2.4	Alterar Ação dos Equipamentos e o Tempo ou Condição	40
4.2.5	Salvar Cenário	40
4.2.6	Abrir Cenário	41
4.2.7	Manipulação dos cenários em um banco de dados	42
4.3	Conclusão	44
5	Validação do Editor de Cenários do LIHM	45
5.1	Preparando o SIMULOP	45
5.2	Estudo de Caso	49
5.2.1	Subestação Ribeirão (RIB) - PE	49
5.2.2	Subestação Juazeiro da Bahia II (JZD) - BA	61
5.3	Salvando os Cenários de Validação no Banco de Dados	64
5.4	Utilização de Eventos Condicionais com o Editor de Cenários do LIHM	65
5.5	Conclusão	67
6	Considerações Finais	68
6.1	Considerações	68
6.2	Propostas de Trabalhos Futuros	70
A	Criação do Banco de Dados “cenariosSage”	72

B Integração do Editor de Cenários com o Banco de Dados	76
Referências Bibliográficas	78

Índice de Tabelas

2.1	Ações que podem ser aplicadas aos equipamentos	13
2.2	Tipos de disparo dos eventos	14
2.3	Tipos de equipamentos	19
2.4	Tipos de funções	21
2.5	Tipos de comparações	23

Índice de Figuras

1.1	Integração SAGE-OTS	4
1.2	Centro de Treinamento (de Oliveira et al., 2002)	5
2.1	Estrutura de cenários no OTS (Muniz e Miranda, 2009)	8
2.2	Tela inicial do OTS	9
2.3	Grupos de eventos do OTS	10
2.4	Inserindo a descrição do grupo de eventos	10
2.5	Eventos	11
2.6	Inserindo os dados de um evento	11
2.7	Caixa de combinação “ <i>Action</i> ”	12
2.8	Caixa de combinação “ <i>TriggerType</i> ”	14
2.9	Copiar grupo de eventos	14
2.10	Exportar eventos	16
2.11	Exemplo de um trecho de um arquivo texto de cenário gerado pelo OTS	16
2.12	Importar eventos	17
2.13	Esquema para construção de eventos condicionais (Muniz e Miranda, 2009)	18
2.14	Tabela das medidas	18
2.15	Editando as medidas	19
2.16	Tabela das funções	20
2.17	Editando as funções	21
2.18	Tabela das comparações	22
2.19	Editando as comparações	23
2.20	Tabela dos blocos lógicos	24
2.21	Editando os blocos lógicos	25
3.1	Interação entre cliente e servidor	29

3.2	Diagrama de classes do pacote editorinterface	31
3.3	Diagrama de classes do pacote comunicacao	32
3.4	Paleta do <i>Netbeans</i> de componentes do <i>Swing</i>	33
4.1	Fluxo de dados OTS - Editor LIHM - BD	35
4.2	Janela Principal do editor de cenários do LIHM	36
4.3	Eventos	37
4.4	Inserindo os dados do evento	38
4.5	Copiar Eventos	39
4.6	Copiar vários eventos	39
4.7	Replicar Evento	40
4.8	Alterar ação e tempo ou condição	41
4.9	Salvar cenário	41
4.10	Salvar cenário no banco de dados	42
4.11	Cenários armazenados no banco de dados	43
5.1	Tela inicial do Sage	46
5.2	Ativando a Base de Tempo Real do SIMULOP	46
5.3	Aba “Programas” do SIMULOP	47
5.4	Processos do SIMULOP	47
5.5	Executor de Programas do SAGE	48
5.6	Janela principal do OTS	48
5.7	Criando o grupo de eventos 1	50
5.8	Editando Evento	50
5.9	Eventos do grupo 1	51
5.10	Alterando a ação de todos os eventos	51
5.11	Grupos de eventos para validação	52
5.12	Salvando o cenário para validação	52
5.13	Arquivo texto do cenário de validação	53
5.14	Importando cenário de validação no OTS	53
5.15	Cenário criado no editor do LIHM e importado para o OTS	54
5.16	Visor de Telas do SAGE	55
5.17	Regiões das subestações da CHESF	55

5.18	Subestações da Região Leste	56
5.19	Subestação RIB antes da ativação dos eventos	57
5.20	Subestação RIB depois da ativação do grupo de eventos 1	57
5.21	Subestação RIB depois da ativação do grupo de eventos 4	58
5.22	Visor de Alarmes da região Leste	59
5.23	Visor de Alarmes geral indica a mudança de estado dos equipamentos na RIB	59
5.24	Alterando a ação e o tempo de disparo de uma sequência de eventos	60
5.25	Todos os eventos em um só grupo	60
5.26	Grupos de eventos para a segunda etapa de validação	61
5.27	Eventos do grupo 1	62
5.28	Subestação JZD antes da ativação dos eventos	63
5.29	Subestação JZD depois da ativação do grupo de eventos 1	63
5.30	Visor de Alarmes geral indica a mudança de estado dos equipamentos na JZD	64
5.31	Inserindo os parâmetros para salvar o cenário de validação no BD	64
5.32	Janela de cenários do editor do LIHM	65
5.33	Cenário gerado pelo editor do LIHM	66
5.34	Cenário gerado pelo editor do OTS	66
A.1	Tela de <i>login</i> do <i>phpMyAdmin</i>	72
A.2	Estrutura do banco de dados “cenariosSage”	73
A.3	MER do banco “cenariosSage”	73
A.4	Estrutura da tabela “cenarios”	74
A.5	Estrutura da tabela “gruposdeeventos”	74
A.6	Estrutura da tabela “eventos”	74
A.7	Tabela “cenarios”	75
A.8	Tabela “gruposdeeventos”	75
A.9	Tabela “eventos”	75

1

Introdução

Um sistema elétrico de potência é supervisionado por diversos centros de controle, os quais atuam sobre diferentes regiões deste sistema, possuindo níveis diferentes de hierarquia e complexidade. O desenvolvimento deste sistema, sua progressiva interligação e automatização dos centros de controle, elevou o trabalho de seus operadores a um grau de complexidade e responsabilidade consideravelmente maior do que o vigente há alguns anos atrás.

Conseqüentemente, a operação do sistema elétrico de potência atual requer crescentes e complexas tomadas de decisões visando encontrar o compromisso certo entre segurança e economia. Com o sistema sendo operado perto de seus limites físicos, seu funcionamento está cada vez mais sujeito a fenômenos tais como: perda de sincronismo, queda de frequência, colapso de tensão, corte de geradores e cargas, etc. Os operadores devem estar sempre prontos para responder rápida e corretamente sob estas condições (de Oliveira et al., 2002).

1.1 Treinamento de Operadores nos Sistemas Elétricos

Usualmente o treinamento de um novo operador para o sistema elétrico é feito com aulas teóricas e a prática é obtida operando o sistema real. Este processo é incompleto visto que um operador pode ter anos de trabalho sem nunca antes ter se deparado com uma situação grave de emergência. A experiência mostra que um eficiente treinamento só pode ser alcançado, de forma adequada e em curto prazo, empregando-se simuladores para treinamento de operadores (de Oliveira et al., 2002).

Para elaboração de treinamento utilizando simuladores é necessário elaborar os cenários de treinamento. Para este trabalho, o cenário de treinamento consiste na descrição dos eventos e da sequência de passos utilizados durante um treinamento. Cenários são considerados descrições evolutivas de situações em um ambiente sendo compostos por um conjunto ordenado de interações entre seus participantes (Leite et al., 1997).

Assim, objetivando uma melhor qualificação de seus operadores é importante para uma companhia a implantação de simuladores para treinamento e reciclagem periódica de seus operadores. O treinamento de novos operadores empregando simuladores é muito importante para permitir a rápida assimilação de conceitos e procedimentos de operação, bem como permitir o acúmulo de experiências em um curto prazo de tempo. Para operadores experientes, o simulador é empregado para reciclagem dos mesmos, permitindo a revitalização de procedimentos menos frequentes (contingências críticas, blecautes, recomposição, etc.). Outra importante atividade de capacitação é o treinamento de novos procedimentos operativos devido a alterações no sistema elétrico. Desta forma, esta ferramenta se torna bastante útil na empresa, inclusive permitindo estabelecer-se um processo bastante eficiente para a certificação de Operadores de Sistema (de Oliveira et al., 2002).

1.2 Ferramentas que Apoiam o Treinamento

Para apoiar o treinamento de operadores de sistemas elétricos de potência foram desenvolvidas algumas ferramentas de simulação, que permitem ao operador obter experiência para lidar com situações de emergência e condições anormais de operação do sistema, além de eliminar o risco de se trabalhar diretamente com o sistema real em funcionamento (Hirsch, 2004).

Estes ambientes de simulação têm como objetivo (de Oliveira et al., 2002):

- Educação primária do operador: assimilação de conceitos e procedimentos de operação;
- Adaptação ao ambiente da sala de controle;
- Treinamento complementar: confronto com situações extraordinárias;
- Treinamento preparatório para alterações futuras nos procedimentos e configurações da rede;

- Aumento de conhecimento sobre o sistema.

Algumas ferramentas utilizadas no treinamento de operadores do setor elétrico são descritas abaixo.

1.2.1 ASTRO

O Ambiente Simulado para Treinamento de Operadores (ASTRO) foi desenvolvido pela CEPEL (Centro de Pesquisas de Energia Elétrica) em conjunto com a Eletrosul. Ele foi dividido em três módulos: um editor de cenário; um programa de simulação da operação; e um analisador de desempenho. Os cenários são gerados a partir de autômatos para descrever os passos que devem ser executados durante um treinamento, porém a ferramenta não está acessível para uso do público geral (Silva et al., 2009). Autômato é um modelo matemático que representa uma máquina de estados finitos, que ao receber uma entrada, muda de estado de acordo com uma função de transição.

1.2.2 STPO

Desenvolvido na Universidade Federal do Ceará, o STPO (Simulador para Treinamento de Proteção e Operação) pode ser aplicado em treinamentos presenciais e a distância. O sistema é composto de um emulador de sistemas supervisórios denominado “EmulaScada” e o programa do simulador para treinamento de proteção e operação de sistemas elétricos (Bezerra et al., 2007). Esta ferramenta não possui gerador de cenários.

1.2.3 SimuLihm

O SimuLihm, simulador desenvolvido na Universidade Federal de Campina Grande pelo Laboratório de Interface Homem-Máquina (LIHM), diferentemente dos simuladores apresentados anteriormente, tem a proposta de treinar os técnicos nas manobras executadas dentro da subestação, como abertura de disjuntores e seccionadoras no pátio, a partir de um ambiente tridimensional. Também não possui um gerador de cenários de treinamento (Netto e Vieira, 2010).

1.2.4 STriPP

O Simulador Tridimensional em Primeira Pessoa foi desenvolvido pela Furnas Centrais Elétricas e está integrado ao SAGE (Sistema Aberto para Gerenciamento de Energia). O simulador é composto por um ambiente tridimensional onde o operador pode interagir com os objetos da instalação de uma subestação e uma ferramenta para gerenciar os cenários de treinamento (Júnior et al., 2010).

1.2.5 SIMULOP

O SIMULOP (Simulador para Treinamento de Operadores), foi desenvolvido pela CEPEL em conjunto com a CHESF (Companhia Hidro Elétrica do São Francisco). Disponibiliza um ambiente de simulação de sistemas elétricos para utilização no treinamento de operadores de sistema. Na CHESF, estas simulações são executadas a partir da integração de um simulador digital em tempo real de sistemas elétricos de potência, o OTS (*Operator Training Simulator*) desenvolvido pela EPRI (*Electric Power Research Institute*) e uma ferramenta EMS (*Energy Management System*) de supervisão e controle de sistemas de potência. A ferramenta EMS utilizada na CHESF é o SAGE. A Figura 1.1 ilustra a integração do EPRI-OTS e o SAGE, formando o SIMULOP.

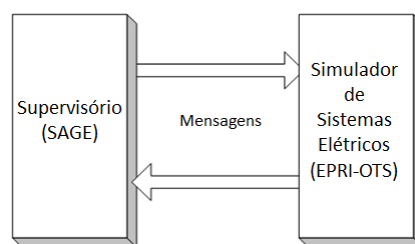


Figura 1.1: Integração SAGE-OTS

O OTS é um software simulador do comportamento dos sistemas elétricos de potência e responde em tempo real a novas condições operativas e eventos sistêmicos. Dessa forma, cada segundo transcorrido nos processos simulados corresponde a um segundo transcorrido na situação real (Muniz, 2004).

Na Figura 1.2 é ilustrada a estrutura de um centro de treinamento típico. Nele os operadores estão inseridos no mesmo ambiente de um centro de controle real, sendo que todas as suas atividades são acompanhadas por um tutor.

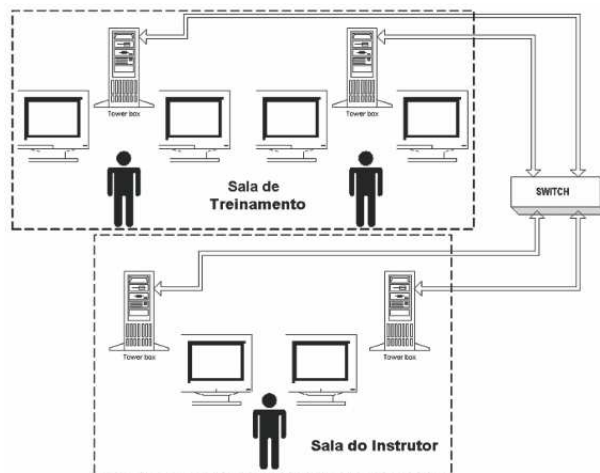


Figura 1.2: Centro de Treinamento (de Oliveira et al., 2002)

Este simulador é o foco deste projeto e a ferramenta construída será utilizada junto ao mesmo.

1.3 Objetivos do Trabalho

O editor de cenários de simulação do OTS apresenta algumas limitações, tornando a edição de cenários uma tarefa difícil, trabalhosa e que demanda tempo. Por exemplo, muitas vezes o que difere um evento de outro é apenas a ação sobre o equipamento. No editor do OTS, o tutor teria que criar este novo evento sem ter a possibilidade de copiar algum já existente e alterar apenas a ação. Dentre este e outros motivos, sentiu-se a necessidade de facilitar a edição de cenários.

O objetivo deste trabalho é construir uma ferramenta que permita a edição de cenários gerados pelo OTS para execução no ambiente SIMULOP. A edição do cenário permitirá “remover”, “inserir” e “modificar” trechos de cenário gerados pelo OTS. Por “trechos” entende-se a descrição de objetos utilizados no cenário e a descrição do seu estado. Para localizar os trechos a serem editados, a ferramenta deverá oferecer um recurso para “busca”.

O OTS exporta um cenário na forma de um arquivo texto para que possa ser importado sempre que necessário. A ferramenta deverá ser capaz de gerar um cenário na forma de um arquivo texto utilizando a mesma formatação que a do arquivo gerado pelo OTS. Dessa forma, o cenário gerado pela ferramenta poderá ser importado para o OTS. A ferramenta

também deve possibilitar que os cenários sejam salvos em um banco de dados.

1.4 Metodologia

Inicialmente foi realizado um estudo sobre treinamento de operadores em subestações elétricas e analisado alguns simuladores para este contexto. O foco principal foi o SIMULOP mais especificamente o gerador de cenários do OTS. Neste estudo foi avaliada a elaboração do cenário na ferramenta e o arquivo exportado por ela (formato do arquivo, codificação dos caracteres, formatação do documento) para isto foram consultados seus manuais e vídeos explicativos. Este estudo tinha como objetivo identificar as limitações e propor melhorias para a ferramenta.

Para dar início a construção da nova ferramenta de edição de cenários foram estudadas algumas tecnologias que serão apresentadas no capítulo 3 (em ferramentas utilizadas), dentre elas pode-se citar o Java e o SQL.

Posteriormente foram especificadas as funcionalidades que deveriam conter na nova ferramenta e foram elas implementadas. Nessa nova ferramenta foram inseridos recursos inexistentes no editor do OTS. Por fim validou-se a ferramenta, onde a validação consistiu na criação de cenários que foram importados para o OTS e utilizados em uma simulação. Ao término da validação, a ferramenta foi devidamente comentada e gerado o relatório final.

1.5 Apresentação do Texto

O capítulo 2 faz uma abordagem sobre o que é um cenário de treinamento e apresentação do editor de cenários da ferramenta OTS. O capítulo 3 descreve o processo de construção da nova ferramenta e quais foram os recursos utilizados. O capítulo 4 apresenta a nova ferramenta e os seus recursos. O capítulo 5 trata do processo de validação do editor. Por fim, no capítulo 6 têm-se as considerações finais e a proposição de trabalhos futuros.

2

Cenários de Treinamento

Um cenário de treinamento consiste de um ou mais grupos de eventos. Dentro de cada grupo existe um conjunto de eventos que ocorrerão durante a simulação. Um evento é uma ação, que pode ser desde a simples abertura de um disjuntor que interliga a subestação a uma linha, à simulação de um esquema de proteção de grande porte.

O ponto de partida para a elaboração de um cenário é um caso base. Este caso base estabelece o estado inicial do sistema e define o estado dos equipamentos, unidades geradoras entre outros. Um caso base pode ser obtido a partir de um “*snapshot*”, que é uma espécie de “fotografia” do estado do sistema. No caso de um disjuntor, este estado descreve se o mesmo está aberto, fechado ou se estará habilitado para manobra. Obtido os estados dos equipamentos o tutor poderá criar eventos que ocorrerão durante a simulação e que farão alterações sobre os estados destes equipamentos (Muniz, 2004). Na Figura 2.1 é ilustrada a estrutura dos cenários utilizados no OTS.

Este capítulo faz uma breve descrição sobre casos base e eventos. Em seguida é mostrado como criar e executar eventos e ainda como exportá-los para um arquivo texto. É mostrado também que cenários exportados para um arquivo texto podem ser importados para serem utilizados novamente. Outro tipo de eventos, os eventos condicionais, que necessitam de alguma condição para que sejam executados, é mostrado também aqui. Por fim, são descritas as limitações do editor de cenários do OTS.

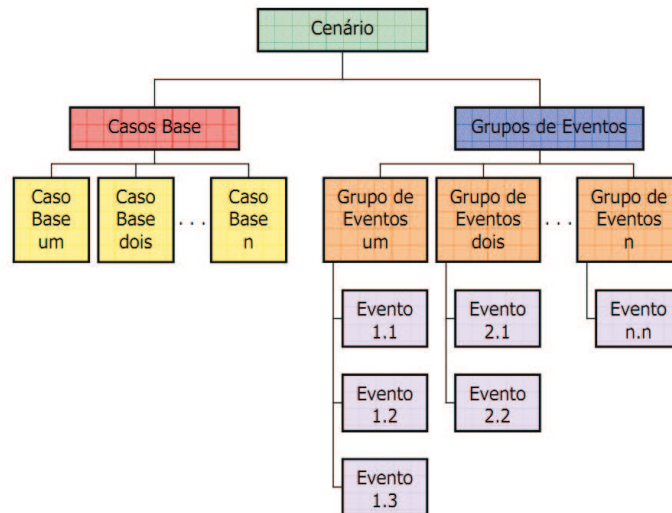


Figura 2.1: Estrutura de cenários no OTS (Muniz e Miranda, 2009)

2.1 Eventos no OTS

Um evento é uma ação, que pode ser desde a simples abertura de uma linha à simulação de um esquema de proteção de grande porte. O OTS possui um editor que possibilita ao tutor criar e editar estes eventos.

O tutor pode selecionar uma seqüência de eventos que ocorrerão durante a sessão de treinamento. Por conveniência, os eventos são separados em grupos (Figura 2.1). Um ou mais grupos de eventos podem ser selecionados para um dado cenário. Os grupos de eventos são preparados para acontecer em tempos pré-determinados e podem ser inicializados quando necessário durante a sessão de treinamento. Uma biblioteca de grupos de eventos pode ser desenvolvida para facilitar a preparação das sessões de treinamento (Muniz, 2004).

Cada evento pode ocorrer em tempo relativo ou absoluto. O tempo relativo é relacionado com o início da simulação. Exemplo, um evento pode ocorrer após 30 segundos dado o início da simulação. O tempo absoluto está relacionado com o relógio do simulador e não possui nenhuma relação com a hora atual (Muniz, 2004).

O OTS também dispõe de eventos condicionais. Estes são ativados quando determinadas condições são aplicadas, que podem ser valores de tensão, potência, corrente ou operação de dispositivos como a abertura de um disjuntor.

Em uma sessão de treinamento, os tutores podem iniciar ou parar a simulação, pausar ou

continuar a simulação, ativar grupos de eventos ou realizar “*snapshots*” para poder retornar facilmente a um determinado ponto da simulação.

2.1.1 Como Criar

A tela inicial do OTS é ilustrada na Figura 2.2. Detalhes de como iniciar o OTS serão mostrados adiante.

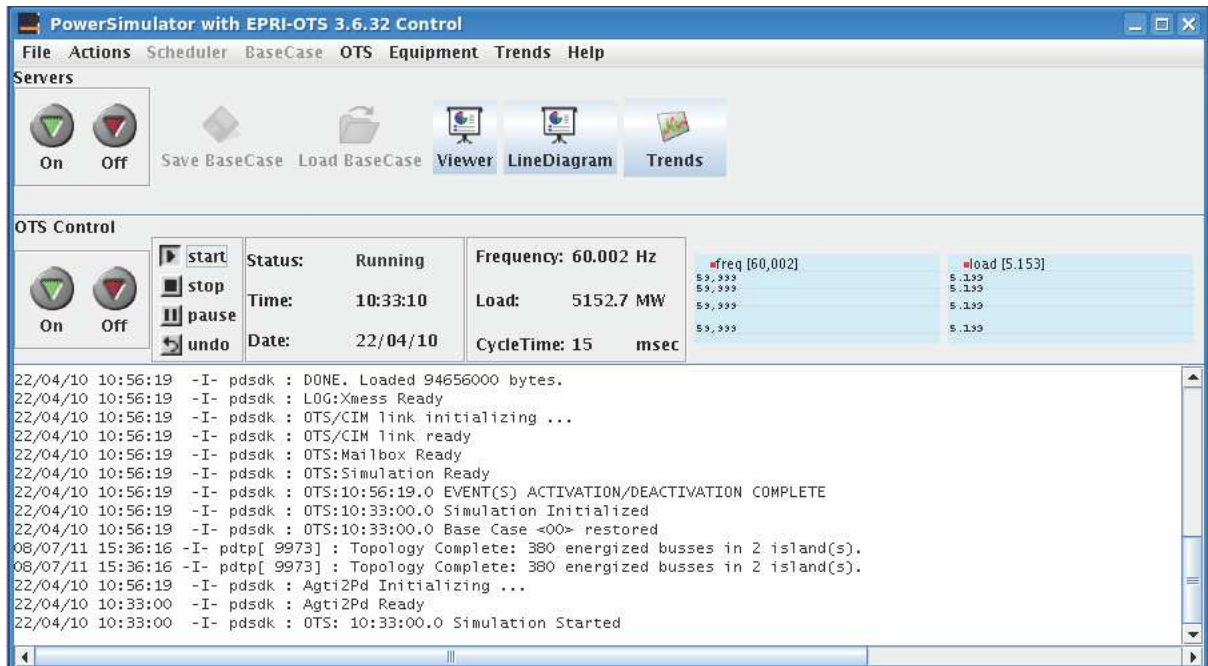
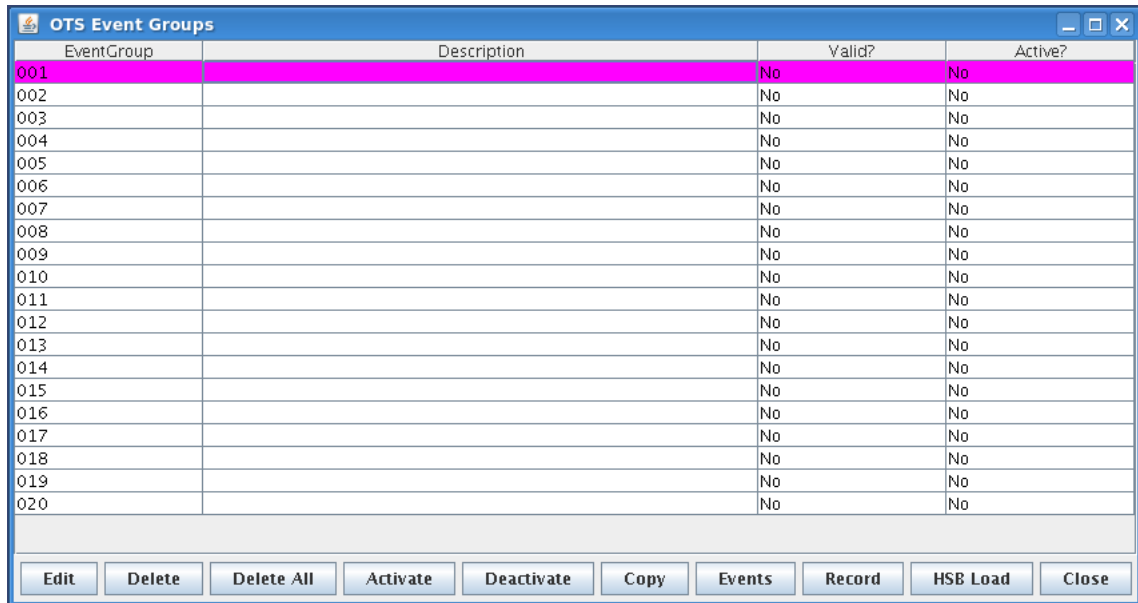


Figura 2.2: Tela inicial do OTS

Para iniciar a criação de eventos, no menu “OTS” escolhe-se “OTS *Events*” e depois “*Event Groups*”. A janela ilustrada na Figura 2.3 será aberta e nela existe uma tabela listando os grupos de eventos.

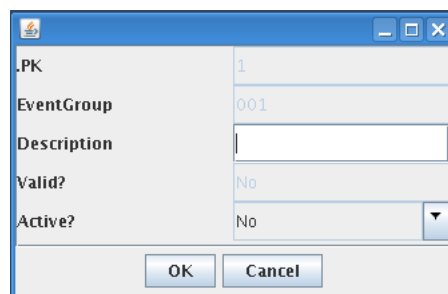


EventGroup	Description	Valid?	Active?
001		No	No
002		No	No
003		No	No
004		No	No
005		No	No
006		No	No
007		No	No
008		No	No
009		No	No
010		No	No
011		No	No
012		No	No
013		No	No
014		No	No
015		No	No
016		No	No
017		No	No
018		No	No
019		No	No
020		No	No

Figura 2.3: Grupos de eventos do OTS

Um cenário é dividido em grupos de eventos onde é possível ter um máximo de 20 grupos. Dentro de cada grupo, podem existir no máximo 99 eventos. Na tabela, a coluna “*Description*” lista a descrição dos grupos. O OTS faz uma validação dos dados inseridos nos eventos e indica se um grupo é válido através da coluna “*Valid?*”. A coluna “*Active?*” mostra se o grupo está ativo na simulação.

Para inserir a descrição de um grupo de eventos seleciona-se o grupo e clica-se em “*Edit*”. Será aberta uma nova janela (Figura 2.4) onde a descrição deverá ser inserida. Clica-se em “*OK*” e a coluna “*Description*” exibirá a descrição do grupo de eventos.



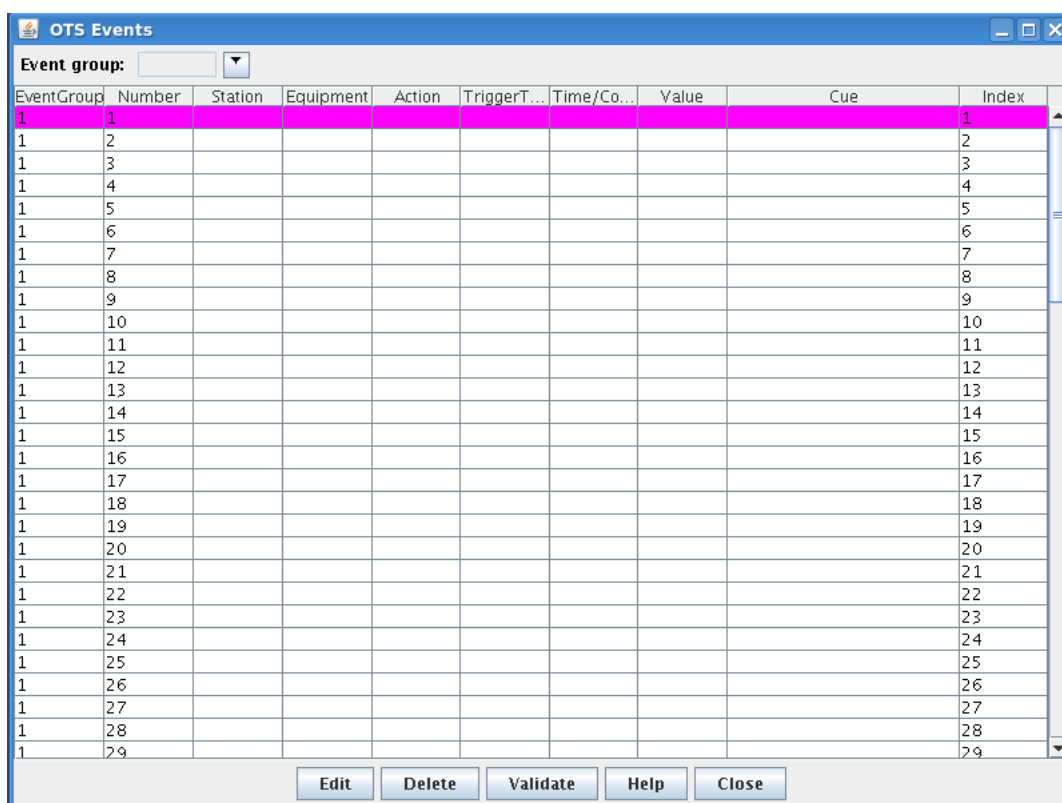
.PK	1
EventGroup	001
Description	
Valid?	No
Active?	No

OK Cancel

Figura 2.4: Inserindo a descrição do grupo de eventos

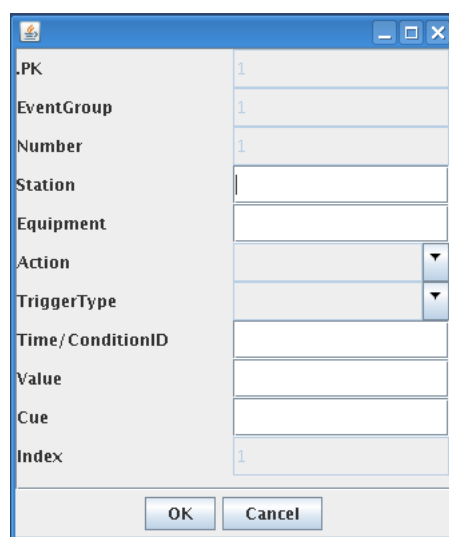
Para editar um evento, com o grupo de eventos ainda selecionado na tabela, clica-se em “*Events*”. Uma nova janela será aberta (Figura 2.5), na qual contém outra tabela que lista os 99 eventos correspondentes ao grupo de eventos selecionado na primeira tabela. Seleciona-se agora o evento que será editado e clica-se no botão “*Edit*”. A janela ilustrada na Figura 2.6

será aberta para que os dados sejam inseridos.



EventGroup	Number	Station	Equipment	Action	TriggerT...	Time/Co...	Value	Cue	Index
1	1								1
1	2								2
1	3								3
1	4								4
1	5								5
1	6								6
1	7								7
1	8								8
1	9								9
1	10								10
1	11								11
1	12								12
1	13								13
1	14								14
1	15								15
1	16								16
1	17								17
1	18								18
1	19								19
1	20								20
1	21								21
1	22								22
1	23								23
1	24								24
1	25								25
1	26								26
1	27								27
1	28								28
1	29								29

Figura 2.5: Eventos



.PK	1
EventGroup	1
Number	1
Station	
Equipment	
Action	
TriggerType	
Time/ConditionID	
Value	
Cue	
Index	1

Figura 2.6: Inserindo os dados de um evento

No campo “*Station*” é colocado o nome da subestação onde o evento deve ocorrer. No campo “*Equipment*” é colocado o nome do equipamento que será acionado neste evento. A CHESF faz uso de dois padrões para nomear os equipamentos. O padrão antigo que contém o nome do equipamento e da subestação a qual pertence, separados por um hífen “-” (Ex.: 12J5-JZD), e o padrão novo que contém o nome da subestação, o nome do equipamento e o código ANSI ¹ para os equipamentos de subestações, todos separados por dois pontos “:” (Ex.: RIB:12L5:52).

Na caixa de combinação “*TriggerType*” deve-se escolher o tipo de disparo do evento. No campo “*TimeCondition/ID*” coloca-se o tempo ou o identificador de uma condição, conforme o tipo de disparo escolhido. O tempo ou identificador da condição deve ter 6 caracteres. Se o tipo de disparo for absoluto ou relativo deve-se colocar o tempo na forma hhhmss (hora, minutos e segundos).

Se a ação deve alterar alguma grandeza, no campo “*Value*” é colocado o valor desta grandeza. No campo “*Cue*” pode ser inserida uma mensagem que será exibida na tela de alarmes do sistema supervisorio no momento em que a ação ocorrer.

A Figura 2.7 ilustra a janela onde se escolhe a ação aplicada ao equipamento, correspondente à caixa de combinação “*Action*” e a Tabela 2.1 lista e descreve estas ações. Nem todas estas ações podem ser aplicadas a todos os equipamentos. Caso seja escolhida uma ação que não seja aplicável ao equipamento selecionado, o OTS não validará o evento.

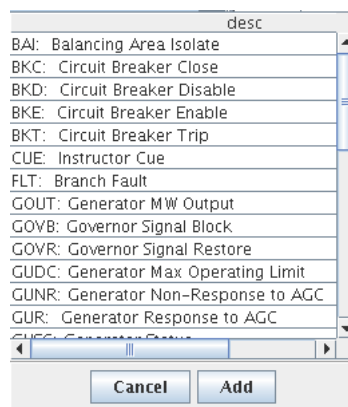


Figura 2.7: Caixa de combinação “*Action*”

¹American National Standards Institute

Tabela 2.1: Ações que podem ser aplicadas aos equipamentos

Código	Nome	Descrição
BAI	<i>Balancig Area Isolate</i>	Gera uma ilha no sistema, balanceando a mesma
BKC	<i>Circuit Breaker Close</i>	Fecha o disjuntor
BKD	<i>Circuit Breaker Disable</i>	Desabilita o comando do disjuntor
BKE	<i>Circuit Breaker Enable</i>	Habilita o comando do disjuntor
BKT	<i>Circuit Breaker Trip</i>	Abre o disjuntor
CUE	<i>Instructor Cue</i>	Envia mensagem ao treinando/tutor
FLT	<i>Branch Fault</i>	
GOUT	<i>Generator MW Output</i>	Modifica geração de uma unidade geradora
GOVB	<i>Governor Signal Blocked</i>	Bloqueio de comando para o gerador
GOVR	<i>Governor Signal Restore</i>	Retira o bloqueio do comando do gerador
GUDC	<i>Generator Max Operating Limit</i>	Máxima geração da máquina
GUNR	<i>Generator Non Response to AGC</i>	Gerador não responde ao CAG (Controle Automático de Geração)
GUR	<i>Generator Response to AGC</i>	Gerador responde ao CAG
LDD	<i>Load Disable</i>	Desabilita a carga de um alimentador
LDE	<i>Load Enable</i>	Habilita a carga de um alimentador
LDS	<i>Load MW Spike</i>	Modifica a potência ativa do alimentador
LDQ	<i>Load MVAR Spike</i>	Modifica a potência reativa do alimentador
LDZ	<i>Zone Load Step</i>	Coloca um degrau de carga de uma determinada zona
LDZS	<i>Zone Load Spike</i>	Modifica a carga de uma determinada zona
NISC	<i>Change Interchange Schedule</i>	Modifica o intercâmbio programado
PAUS	<i>Pause Simulation</i>	Pausa a simulação
PXA	<i>Replace External Analog Point</i>	Faz a varredura de um ponto analógico
PXD	<i>Replace External Digital Point</i>	Faz a varredura de um ponto digital
RLYD	<i>Disable Relay</i>	Desabilita um relé
RLYE	<i>Enable Relay</i>	Habilita um relé
RTL	<i>RTU Loss</i>	Causa a perda de uma remota
RTR	<i>RTU Restore</i>	Restaura uma remota
STOP	<i>Stop Simulation</i>	Para a simulação
VRGU	<i>Change Unit Regulating Voltage</i>	Modifica o status do regulador de tensão
VRMN	<i>Change LTC Regulating Voltage (min)</i>	Modifica o valor de tensão mínima do regulador
VRMX	<i>Change LTC Regulating Voltage (max)</i>	Modifica o valor de tensão máxima do regulador
TPD	<i>Disable LTC</i>	Desabilita o regulador de tensão
TPE	<i>Enable LTC</i>	Habilita o regulador de tensão
TPHI	<i>Tap High Side Change</i>	Eleva o <i>tap</i> do transformador ao valor solicitado
TPLO	<i>Tap Low Side Change</i>	Diminui o <i>tap</i> do transformador ao valor solicitado

A Figura 2.8 ilustra a janela de configuração do tipo de disparo, correspondente à caixa de combinação “*TriggerType*”, e a Tabela 2.2 lista as descrições para cada tipo de disparo.

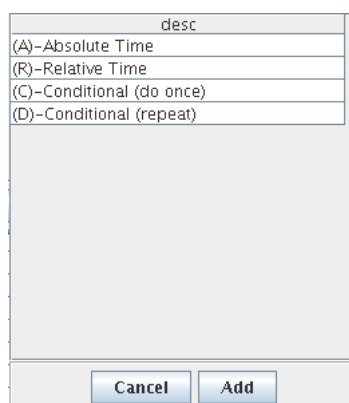


Figura 2.8: Caixa de combinação “*TriggerType*”

Tabela 2.2: Tipos de disparo dos eventos

Tipo de disparo	Descrição
(A) - Absolute Time	O evento acontece precisamente no horário determinado
(R) - Relative Time	O evento acontece depois do tempo selecionado, relativo à ativação do evento
(C) - Conditional (<i>once</i>)	O evento vai acontecer apenas uma vez, a partir da condição da expressão lógica definida tornar-se verdade
(D) - Conditional (<i>repeat</i>)	O evento vai acontecer sempre que a condição da expressão lógica definida tornar-se verdade

Ainda no editor de cenários do OTS, é possível fazer uma cópia de um grupo de eventos existente para um novo grupo. Na janela do editor, seleciona-se a linha do grupo que será copiado e depois se clica no botão “*Copy*”. Uma nova janela será aberta (Figura 2.9) e nela digita-se o número do grupo que receberá a cópia.

É possível também apagar um grupo de eventos selecionado ou apagar todos de uma só vez, clicando-se nos botões “*Delete*” e “*Delete All*”, respectivamente.

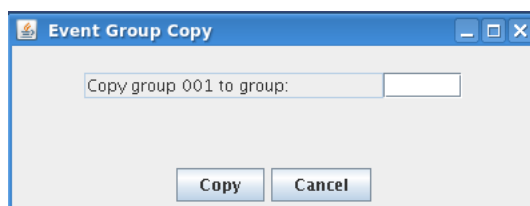


Figura 2.9: Copiar grupo de eventos

2.1.2 Executar

Para que um grupo de eventos seja executado é necessário ativá-lo. Ele será executado conforme sua configuração, ou seja, se por tempo após o tempo inicial decorrido, ou se por condição, caso a condição exigida seja atendida. Existem duas maneiras de ativá-lo ou desativá-lo, sendo uma através do botão “*Activate*” ou “*Deactivate*” (para desativar) da janela do editor, ou na janela onde se insere a descrição do grupo de eventos (Figura 2.4), na caixa de combinação “*Active*”.

2.1.3 Exportar

O OTS possibilita que os eventos sejam exportados para uma futura importação. Apenas os grupos de eventos desejados são exportados. Para exportar um grupo de eventos seleciona-se “*Export Events*” no menu OTS - OTS *Events* do OTS. Uma nova janela (Figura 2.10) será aberta e nela devem ser inseridos os grupos de eventos que serão exportados. Se nenhum grupo de eventos for colocado, todos os grupos serão exportados. Clicando no botão “*File*” escolhe-se o caminho onde o arquivo será armazenado. O arquivo exportado não possui extensão e armazena as informações dos eventos na forma de um texto.

Um exemplo de um trecho de um cenário exportado para um arquivo texto é ilustrado na Figura 2.11. Neste arquivo texto, os grupos de eventos são distribuídos seguindo a numeração e um grupo será exibido no texto caso exista um evento dentro deste grupo.

No arquivo texto um grupo de eventos é iniciado com a palavra “- - - EVENT GROUP - - - xxx”, onde “xxx” indica o número do grupo. Na linha em seguida ao lado do caractere “G:”, está escrita a descrição do grupo. Logo abaixo estão dispostas as linhas de texto com os dados dos eventos. Uma linha de evento é iniciada com o caractere “E”, seguido do caractere que indica o tipo de disparo da ação, o tempo ou identificador de uma condição, a ação sobre o equipamento, o nome da subestação, o nome do equipamento (padrão novo no exemplo ilustrado na Figura 2.11), o valor do ítem “*Value*” (está em branco no exemplo) e por fim a mensagem do ítem “*Cue*”, que será exibida na tela de alarmes do supervisor do sistema. Todos estes itens que compõem um evento são separados pelo caractere “:” e uma linha de evento é finalizada pelo caractere “!”.

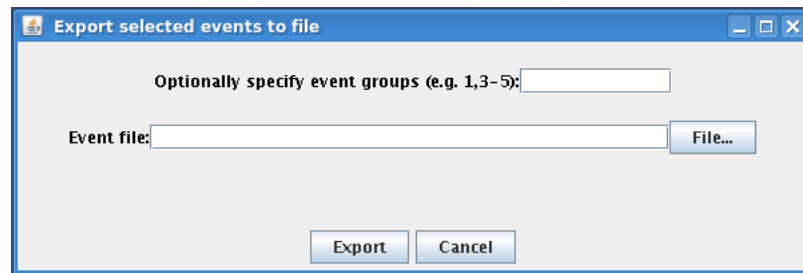


Figura 2.10: Exportar eventos

```

0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
1 | Export Event Library 09/08/20 17:26:11
2 * OTS Model EXP Database eprints/db/DBBA
3 * OTS Date 00/00/2009 TIME 00:00:00
4 * Maximum Number of Event Groups 20
5 * Maximum Number of Events Per Group 99
6 *
7 * ---- EVENT GROUP ---- 001
8 *
9 G:Evento 1
10 E:R:000001:BKT :RIB :RIB:12T2:52 : : :04T3-RIB TRIP 50/51 NEUTRO - 69kV !
11 E:R:000001:BKT :RIB :RIB:12T4:52 : : : : !
12 E:R:000001:BKT :RIB :RIB:12L1:52 : : :04T3-RIB RELE DE BLOQUEIO !
13 E:R:000001:BKT :RIB :RIB:12L2:52 : : :12T3-RIB FALHA ATUADO - 62BFX !
14 E:R:000001:BKT :RIB :RIB:12L3:52 : : :02BP-RIB RELE DE BLOQUEIO 86-02A1 ATUADO !
15 E:R:000001:BKT :RIB :RIB:12L4:52 : : :02BP-RIB SUBTENSÃO BARRA DISPARO !
16 E:R:000001:BKT :RIB :RIB:12L5:52 : : : !
17 E:R:000001:BKT :RIB :RIB:12L6:52 : : : !
18 E:R:000001:BKT :RIB :RIB:12L7:52 : : : !
19 E:R:000001:BKT :RIB :RIB:12L8:52 : : : !
20 E:R:000001:BKT :RIB :RIB:12L9:52 : : : !
21 E:R:000008:BKD :RIB :RIB:12T2:52 : : : !
22 E:R:000008:BKD :RIB :RIB:12T4:52 : : : !
23 E:R:000008:BKD :RIB :RIB:12L1:52 : : : !
24 E:R:000008:BKD :RIB :RIB:12L2:52 : : : !
25 E:R:000008:BKD :RIB :RIB:12L3:52 : : : !
26 E:R:000008:BKD :RIB :RIB:12L4:52 : : : !
27 E:R:000008:BKD :RIB :RIB:12L5:52 : : : !
28 E:R:000008:BKD :RIB :RIB:12L6:52 : : : !
29 E:R:000008:BKD :RIB :RIB:12L7:52 : : : !
30 E:R:000008:BKD :RIB :RIB:12L8:52 : : : !
31 E:R:000008:BKD :RIB :RIB:12L9:52 : : : !
32 E:C:COMP03:BKE :RIB :RIB:12T2:52 : : : !
33 E:C:COMP03:BKE :RIB :RIB:12T4:52 : : : !
34 E:C:COMP03:BKE :RIB :RIB:12L1:52 : : : !
35 E:C:COMP03:BKE :RIB :RIB:12L2:52 : : : !
36 E:C:COMP03:BKE :RIB :RIB:12L3:52 : : : !
37 E:C:COMP03:BKE :RIB :RIB:12L4:52 : : : !
38 E:C:COMP03:BKE :RIB :RIB:12L5:52 : : : !
39 E:C:COMP03:BKE :RIB :RIB:12L6:52 : : : !
40 E:C:COMP03:BKE :RIB :RIB:12L7:52 : : : !
41 E:C:COMP03:BKE :RIB :RIB:12L8:52 : : : !
42 E:C:COMP03:BKE :RIB :RIB:12L9:52 : : : !
43 E:R:000001:BKT :RIB :RIB:14T3:52 : : : !
44 E:R:000008:BKD :RIB :RIB:14T3:52 : : : !
45 E:C:COMP03:BKE :RIB :RIB:14T3:52 : : : !
46 *
47 * ---- EVENT GROUP ---- 002
48 *
49 G:Evento 2
50 E:R:000001:BKT :RIB :RIB:12L1:52 : : :04M1-RIB PROTEÇÃO DISTÂNCIA DISPARO !
51 E:R:000001:BKT :AGL :14S1-AGL : : :04M1-RIB CARRIER TRANSMISSÃO !
52 E:R:000001:BKT :RIB :RIB:14M1:52 : : :04S1-RIB CARRIER TRANSMISSÃO !
53 E:R:000001:BKT :RIB :RIB:12L4:52 : : :02BP-RIB SUBTENSÃO BARRA DISPARO !
54 E:R:000001:BKT :RIB :RIB:12L5:52 : : : !
55 E:R:000001:BKT :RIB :RIB:12L6:52 : : : !
56 E:R:000001:BKT :RIB :RIB:12L7:52 : : : !
57 E:R:000001:BKT :RIB :RIB:12L8:52 : : : !
58 E:R:000001:BKT :RIB :RIB:12L9:52 : : : !

```

Figura 2.11: Exemplo de um trecho de um arquivo texto de cenário gerado pelo OTS

2.1.4 Importar

Eventos que foram exportados podem ser importados para serem utilizados novamente. Para fazer a importação seleciona-se a opção “*Import Events*” no menu OTS - OTS “*Events*” do OTS. Na janela que será aberta (Figura 2.12) clica-se no botão “*File*” para localizar o arquivo armazenado.

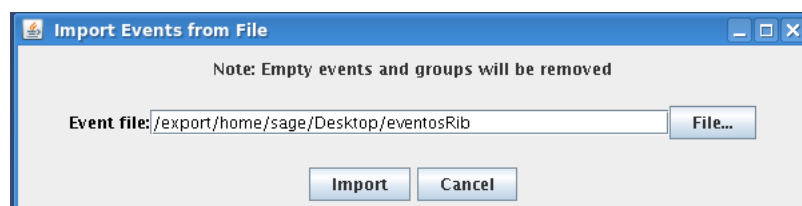


Figura 2.12: Importar eventos

2.1.5 Como Criar Eventos Condicionais

Os eventos condicionais permitem ao tutor programar várias ações automáticas. Os eventos condicionais podem ser editados no próprio OTS ou podem ser editados no arquivo texto e depois importados.

Através dos eventos condicionais é possível emular as seguintes situações:

- Impedir que em determinada manobra, o treinando faça alguma manobra dentro do cenário de treinamento que possa levar este ao colapso;
- Automatizar funções que possam trazer a simulação para contingências ou cenários mais realísticos;
- Sem a necessidade de carregar uma base para inserções de proteções em determinados pontos do sistema, emular estas através de eventos condicionais;

A lógica dos eventos condicionais é montada a partir das funcionalidades:

- *Measurements* (medidas);
- *Functions* (funções);
- *Comparison* (comparações);
- *Logic* (blocos lógicos);

Na Figura 2.13 é ilustrado um esquema para construção de eventos condicionais.

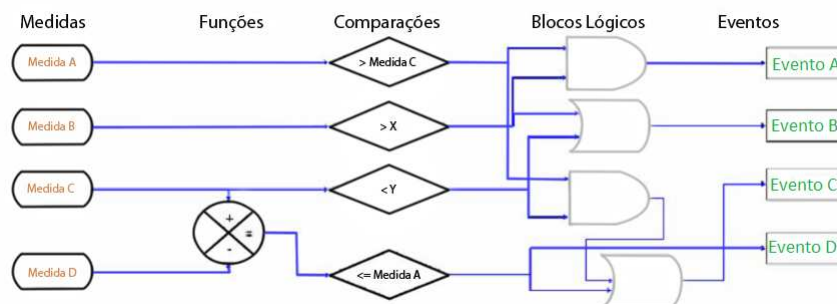


Figura 2.13: Esquema para construção de eventos condicionais (Muniz e Miranda, 2009)

Medidas

Medidas são os valores dos parâmetros monitorados na subestação e obtidos a partir de sensores. Aqui elas são utilizadas como dados de entrada para criação de uma condição. Para construir uma medida, no menu OTS escolhe-se a opção *Conditional Events - Measurements*. Será aberta uma janela (Figura 2.14) que contém uma tabela onde se encontram as medidas existentes. Para editar uma medida seleciona-se uma linha na tabela e clica-se em “*Edit Row*”. A coluna “Value” é possível visualizar o valor atual da medida.

Number	Measurement...	Equipment T...	Equipment St...	Equipment N...	Sub-ID	Measurement...	Value
1							0.0
2							0.0
3							0.0
4							0.0
5							0.0
6							0.0
7							0.0
8							0.0
9							0.0
10							0.0
11							0.0
12							0.0
13							0.0
14							0.0
15							0.0
16							0.0
17							0.0
18							0.0
19							0.0
20							0.0
21							0.0
22							0.0

Figura 2.14: Tabela das medidas

Na Figura 2.15 é ilustrada a janela de edição das medidas. Nela podem ser editados os seguintes itens:

- **MeasurementID**: nome adotado pela medida, devendo conter até 6 caracteres;

- **Equipment Type:** tipo de equipamento a qual a medida é referida e deve ser escolhido na caixa de combinação. Os tipos existentes encontram-se na Tabela 2.3;
- **Equipment Station:** subestação onde será realizada a medida;
- **Equipment Name:** nome do equipamento;
- **Measurement Type:** tipo de medida referente ao equipamento, podendo ser corrente, tensão, entre outras. Pode ser também uma medida digital, que no caso retornará o estado do equipamento.

Figura 2.15: Editando as medidas

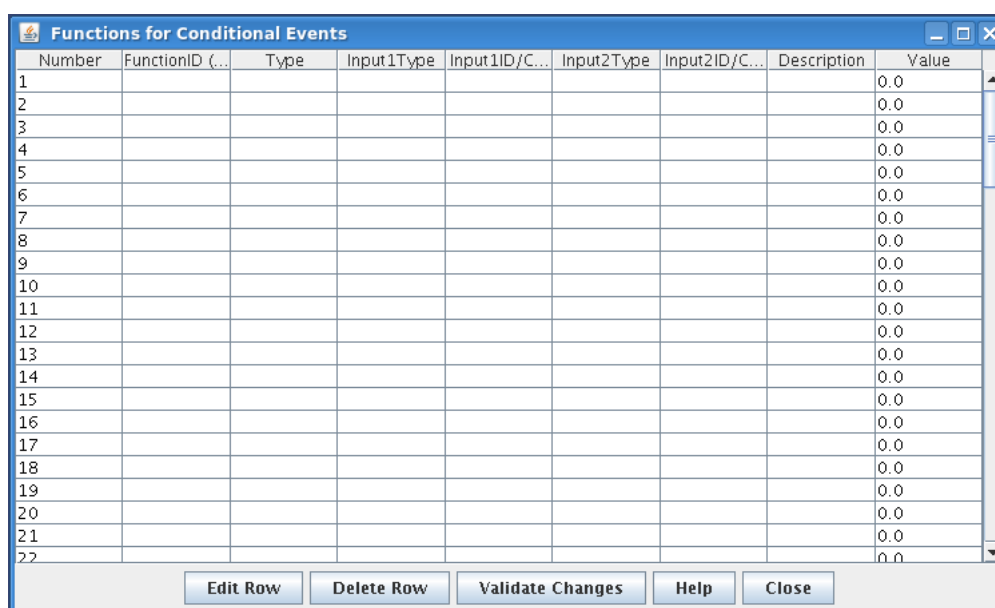
Tabela 2.3: Tipos de equipamentos

Tipo	Descrição
None	Nenhum
AR	Área de Controle
FD	Alimentador
GU	Unidade geradora
LD	Carga
LE	Extremo da linha
NV	Nó
PS	Defasador
SH	Shunt
SS	TCSC
SW	Chave ou Disjuntor
SY	Sistema
XA	Medida analógica
XD	Medida digital
XF	Transformador

Com estes dados a medida pode ser construída e ela é o ponto de partida para a criação de uma condicionalidade.

Funções

Para editar as funções, segue-se o caminho: menu OTS escolhe-se a opção *Conditional Events - Functions Blocks*. Será aberta a janela ilustrada na Figura 2.16, que contém uma tabela com as funções existentes. Para editar uma função, seleciona-se uma linha e clica-se em “*Edit Row*”.



Number	FunctionID (...)	Type	Input1Type	Input1ID/C...	Input2Type	Input2ID/C...	Description	Value
1								0.0
2								0.0
3								0.0
4								0.0
5								0.0
6								0.0
7								0.0
8								0.0
9								0.0
10								0.0
11								0.0
12								0.0
13								0.0
14								0.0
15								0.0
16								0.0
17								0.0
18								0.0
19								0.0
20								0.0
21								0.0
22								0.0

Figura 2.16: Tabela das funções

Na janela de edição das funções (Figura 2.17) inserem-se os seguintes dados:

- **FunctionID**: nome adotado para função, devendo ter até 6 caracteres;
- **Type**: tipo de função que será utilizado e deve ser escolhido na caixa de combinação. Os tipos estão descritos na Tabela 2.4;
- **Input1Type**: tipo da variável de entrada, podendo ser uma constante, uma medida ou outra função;
- **Input1ID/Constant**: nome da medida ou função, ou no caso de constante, o valor da mesma;

- ***Input2Type***: tipo da segunda variável de entrada;
- ***Input2ID/Constant***:: nome da segunda medida, função ou valor da constante;
- ***Description***: descrição para a função.

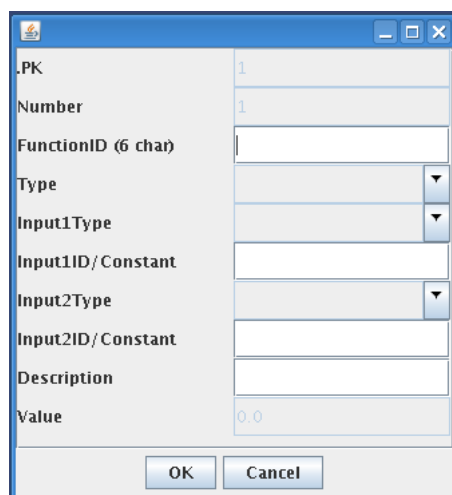


Figura 2.17: Editando as funções

Tabela 2.4: Tipos de funções

Tipo	Descrição
<i>Add</i>	Adição
<i>Divide</i>	Divisão
<i>Multiply</i>	Multiplicação
<i>Subtract</i>	Subtração

Comparações

Para editar as comparações, no menu OTS escolhe se a opção *Conditional Events - Comparison Blocks*. Será aberta a janela ilustrada na Figura 2.18, que contém uma tabela com as comparações existentes. Para editar uma comparação, seleciona-se uma linha na tabela e clica-se em “*Edit Row*”.

Number	Comparis...	Input 1 Ty...	Input 1 Co...	Comparis...	Input 2 Ty...	Input 2 Co...	Delay (sec)	Description	Value
1							0		False
2							0		False
3							0		False
4							0		False
5							0		False
6							0		False
7							0		False
8							0		False
9							0		False
10							0		False
11							0		False
12							0		False
13							0		False
14							0		False
15							0		False
16							0		False
17							0		False
18							0		False
19							0		False
20							0		False
21							0		False
22							0		False

Figura 2.18: Tabela das comparações

Na janela de edição das comparações (Figura 2.19) inserem-se os seguintes dados:

- **ComparisonID:** nome adotado para comparação, devendo ter até 6 caracteres;
- **Input 1 Type:** tipo da variável de entrada, podendo ser uma constante, uma medida ou outra função;
- **Input 1 Constant/FunctionID/MeasurementID:** nome da medida ou função, ou no caso de constante, o valor da mesma;
- **Comparison Type:** tipo de comparação feita entre as variáveis de entrada. Os tipos estão descritos na Tabela 2.5;
- **Input2Type:** tipo da segunda variável de entrada;
- **Input 2 Constant/FunctionID/MeasurementID:::** nome da segunda medida, função ou valor da constante;
- **Delay:** tempo que deve ser considerado para a comparação ser dada como válida;
- **Description:** descrição para a comparação.

Figura 2.19: Editando as comparações

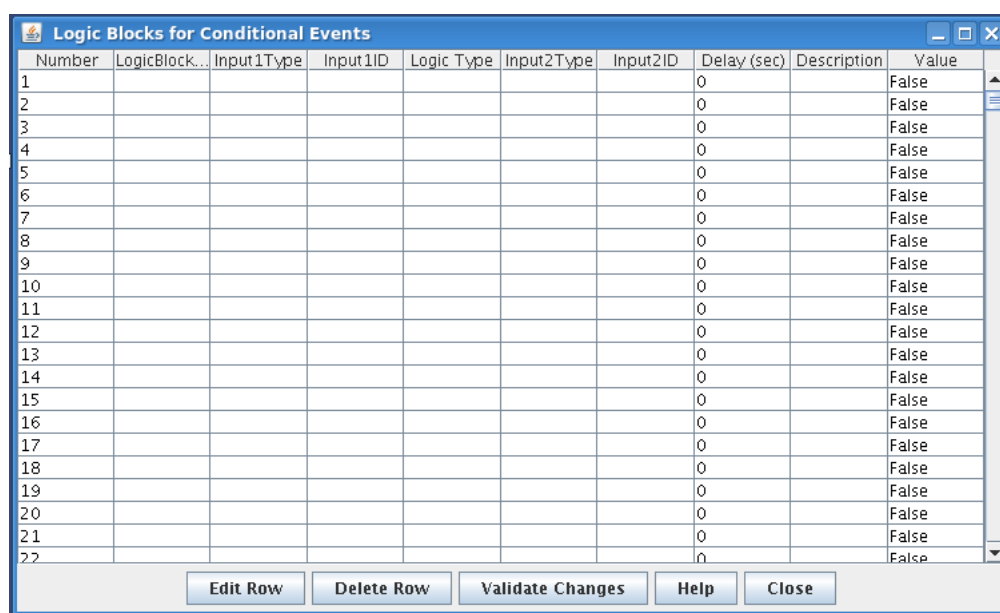
Tabela 2.5: Tipos de comparações

Tipo	Descrição
<i>equal to</i>	Igual à
<i>greater than ou equal to</i>	Maior ou igual à
<i>greater than</i>	Maior que
<i>less than or equal to</i>	Menor ou igual à
<i>less than</i>	Menor que
<i>not equal</i>	Diferente

Blocos Lógicos

Os blocos lógicos permitem a realização de operações lógicas com o resultados de comparações e de medidas digitais (estado do equipamento). A saída do bloco lógico normalmente é a chave de ativação da condição para atuação do evento.

Para editar os blocos lógicos lógico, no menu OTS escolhe se a opção *Conditional Events - Logic Blocks*. Será aberta a janela ilustrada na Figura 2.20, que contém uma tabela com os blocos lógicos existentes. Para editar um bloco lógico, seleciona-se uma linha na tabela e clica-se em “*Edit Row*”.



Number	LogicBlock...	Input1Type	Input1ID	Logic Type	Input2Type	Input2ID	Delay (sec)	Description	Value
1							0		False
2							0		False
3							0		False
4							0		False
5							0		False
6							0		False
7							0		False
8							0		False
9							0		False
10							0		False
11							0		False
12							0		False
13							0		False
14							0		False
15							0		False
16							0		False
17							0		False
18							0		False
19							0		False
20							0		False
21							0		False
22							0		False

Figura 2.20: Tabela dos blocos lógicos

Na janela de edição dos blocos lógicos (Figura 2.21) inserem-se os seguintes dados:

- **LogicBlockID**: nome adotado para o bloco lógico, devendo ter até 6 caracteres;
- **Input1Type**: tipo da variável de entrada, podendo ser uma comparação, ou mesmo outro bloco lógico;
- **Input1ID**: nome da comparação ou do bloco lógico;
- **Logic Type**: tipo de operação lógica aplicada às variáveis de entrada. Estas operações lógicas podem ser *AND*, *OR*, *NAND* e *NOR*;
- **Input2Type**: tipo da segunda variável de entrada;

- **Input2ID::** nome da segunda comparação ou bloco lógico;
- **Delay:** tempo que deve ser considerado para o bloco lógico ser dado como válido;
- **Description:** descrição do bloco lógico.

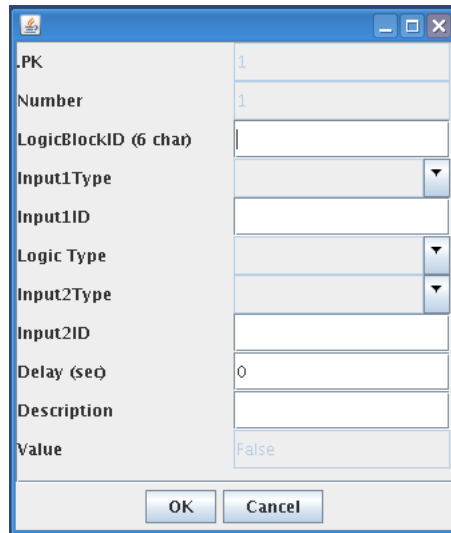


Figura 2.21: Editando os blocos lógicos

2.2 Limitações do Editor de Cenários do OTS

No editor de cenários atual do OTS a ausência de alguns recursos tornam a criação de eventos difícil e demorada. Como mostrado na seção anterior, os recursos oferecidos com relação à edição de eventos, são apenas copiar um grupo de eventos e apagar um ou todos os grupos.

Duas ações comuns nas sessões do treinamento é a abertura dos disjuntores de uma subestação seguida do bloqueio dos mesmos (no caso de um curto-circuito, por exemplo). Uma das formas de simular este processo é criar um grupo de eventos para a abertura e outro para o bloqueio. Neste caso tem-se um evento para cada disjuntor. Dessa forma, na criação do segundo grupo de eventos, o tutor poderá fazer a cópia do primeiro para um segundo grupo, e neste deve-se alterar cada um dos eventos para agora realizar o bloqueio dos disjuntores. Esta segunda etapa seria facilitada se fosse possível a alteração das ações sobre os equipamentos simultaneamente, ou seja, alterar de uma só vez as ações de cada evento do grupo.

Outra maneira de simular este processo é colocar todos estes eventos em um só grupo. Primeiro ter-se-iam os eventos que fariam a abertura dos disjuntores seguidos dos eventos que fariam o bloqueio dos mesmos, sendo estes últimos com um atraso no tempo em relação aos primeiros eventos. O editor do OTS não permite que a primeira sequência de eventos seja copiada para que posteriormente alterem-se apenas as ações dos eventos.

2.3 Conclusão

No EPRI-OTS, um cenário de treinamento é dividido em grupos de eventos, onde cada grupo possui um conjunto de eventos. Um caso base estabelece o estado inicial do sistema e é o ponto de partida para criação do cenário. Os eventos podem ocorrer em tempo relativo ou absoluto, ou podem ser condicionais, nos quais dependem que uma condição seja atendida para que sejam ativados.

No OTS existe um editor de cenários que permite ao tutor elaborar os eventos que ocorrerão durante uma sessão de treinamento. Após criados, os eventos podem ser ativados ou desativados. Ele permite que os eventos sejam exportados na forma de um arquivo texto, de forma que o cenário possa ser importado sempre que necessário. As condições para serem utilizadas junto aos eventos condicionais são criadas utilizando algumas funcionalidades do OTS. A ausência de alguns recursos no editor do OTS tornam a edição de cenários uma tarefa difícil e que demanda tempo.

3

Construção do Editor de Cenários

Este capítulo mostra como e quais foram as ferramentas utilizadas para a construção do editor de cenários, bem como as classes e métodos utilizadas no programa e o processo de construção da interface gráfica do editor. Contém ainda uma breve descrição sobre o sistema de gerenciamento de banco de dados MySQL, como ele é instalado e como ele pode ser administrado através de uma página WEB.

Como citado anteriormente, para a construção da ferramenta utilizou-se a linguagem de programação Java. A escolha do Java se deve a vários fatores. Dentre eles:

- Linguagem orientada a objetos;
- Existem vários IDE's ¹ gratuitas para desenvolvimento;
- É portátil, ou seja, o Java pode ser executado em qualquer plataforma ou equipamento que possua um interpretador Java instalado.

O sistema de gerenciamento de banco de dados MySQL também foi utilizado para que a ferramenta pudesse armazenar cenários em um banco de dados.

¹*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado

3.1 Ferramentas Utilizadas

3.1.1 Construção do Editor

Para a construção do editor foi utilizado o IDE *NetBeans* 7.0. O *NetBeans* é um IDE gratuito e disponibiliza vários recursos. É disponível para download na página:

<http://netbeans.org/downloads/>.

Em conjunto com o *NetBeans* foi utilizado o JDK 6.0 (*Java Development Kit*) que disponibiliza várias bibliotecas para a construção de aplicativos Java. É disponível para download gratuito na página: <http://www.oracle.com/>.

Para que os programas Java pudessem ser executados, foi utilizado o JRE 6.0 (*Java Runtime Environment*). O JRE funciona como um interpretador de programas Java. É também disponibilizado para download gratuito na página citada no parágrafo acima.

Na construção do diagrama de classes foi utilizado o software *astah community* 6.4. O *astah* é uma ferramenta que utiliza a linguagem UML (*Unified Modeling Language*) para a construção de diagramas. A versão *community* é gratuita e é disponibilizada na página: <http://astah.net/>.

3.1.2 O Sistema de Gerenciamento de Banco de Dados MySQL

Um banco de dados é uma coleção de dados estruturados. Ele pode ser qualquer coisa desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informação de uma rede corporativa. Para adicionar, acessar, e processar dados armazenados em um banco de dados de um computador, você necessita de um sistema de gerenciamento de bancos de dados como o Servidor MySQL (MyS, 1997-2006).

O MySQL é um servidor de banco de dados SQL, muito rápido, multi-tarefa e multi-usuário. É também o sistema mais popular de gerenciamento de banco de dados. Devido algumas características importantes do MySQL, optou-se por utilizá-lo neste trabalho. São elas:

- É um banco de dados relacional (separa os dados por tabelas);
- Multiplataforma (Windows, Linux, Mac OS, Solaris, etc.);

- Diversas linguagens de programação possuem API's ² para uso com o MySQL;
- Os clientes podem de qualquer plataforma se conectar ao servidor MySQL usando *sockets* TCP/IP (ponto de comunicação composto por um IP e uma porta);
- Disponibiliza uma versão gratuita e de código aberto.

O servidor MySQL na versão *community* (versão gratuita) é disponibilizado para download na página <http://dev.mysql.com/downloads/mysql/>. A instalação é simples e rápida. Na instalação algumas informações são solicitadas, tais como a senha para a administração do banco de dados e a porta onde o servidor realizará conexões.

Foi utilizada também uma ferramenta que possibilita que o banco de dados possa ser administrado através de uma página WEB. Esta ferramenta é denominada *phpMyAdmin*. Para que o *phpMyAdmin* possa ser acessado, é necessária a instalação de um servidor WEB. O servidor WEB mais utilizado é o *Apache* (download gratuito na página <http://httpd.apache.org/>). Este sistema de administração de banco de dados via WEB é escrito com a linguagem PHP, logo, deve-se instalar também um módulo PHP para o *Apache* (download gratuito na página <http://www.php.net/>). Por fim, coloca-se na pasta de configuração do Apache os arquivos do *phpMyAdmin* (download gratuito na página <http://www.phpmyadmin.net/>). Na Figura 3.1 está ilustrada a interação entre o cliente e o servidor. Seguindo estes procedimentos o banco de dados estará pronto para ser acessado e administrado pelo *phpMyAdmin*.

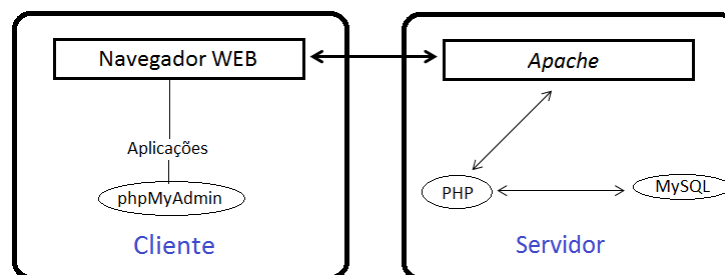


Figura 3.1: Interação entre cliente e servidor

No apêndice A é descrito como foi criado o banco de dados para armazenar os cenários. No apêndice B é mostrado como é feita a integração do programa Java com o banco de dados.

² *Application Programming Interface*, ou Interface de Programação de Aplicativos

3.2 Classes e Métodos

No *NetBeans* iniciou-se um projeto de nome “EditorInterface”. Neste projeto foi inserido um pacote denominado “editorinterface” e neste foi inserido um formulário *JFrame*. *JFrame* é uma classe do pacote *Swing* do Java e é utilizada para fazer a janela da interface gráfica de um programa.

Quando o formulário *JFrame* é inserido no pacote, automaticamente uma classe que herda a *JFrame* é criada. Esta classe foi denominada “EditorInterface”. A classe *EditorInterface* possui como um de seus atributos uma matriz de *Strings* que armazena todos os dados do cenário e métodos para armazenar e obter dados desta matriz.

Os métodos vinculados aos componentes da interface também fazem parte da classe *EditorInterface*. Dessa forma, sempre que acontece alguma mudança nestes componentes um método pode ser chamado. Por exemplo, ao clicar em um botão ou clicar duas vezes em uma linha da tabela, um método é chamado pelo programa.

Outras classes que possuem métodos necessários para a edição dos cenários também fazem parte do pacote “editorinterface”. Tais métodos fazem cópias dos eventos, apagam eventos, salvam o cenário no arquivo texto, dentre outros.

Os métodos vinculados aos componentes da interface realizam a chamada dos métodos das outras classes. Exemplo, ao clicar no botão “Copiar” da janela principal do editor, um método vinculado ao botão será chamado e este chamará o método da classe responsável pela cópia do grupo de eventos selecionado.

Para a manipulação dos cenários no banco de dados foi adicionado no projeto um pacote denominado “comunicacao”. Dentro deste pacote existe uma classe chamada “ServidorBD”, que possui métodos que realizam as transações com o banco de dados. A interação da ferramenta com o banco de dados será tratada adiante.

Utilizando o *astah* foram construídos os diagramas de classe para os dois pacotes utilizados. Nas Figuras 3.2 e 3.3 estão ilustrados estes diagramas. As classes *EditorInterface* e *ServidorBD* apontam com setas contínuas para as classes herdadas. A classe *EditorInterface* aponta com setas pontilhadas para as classes nas quais ela depende.

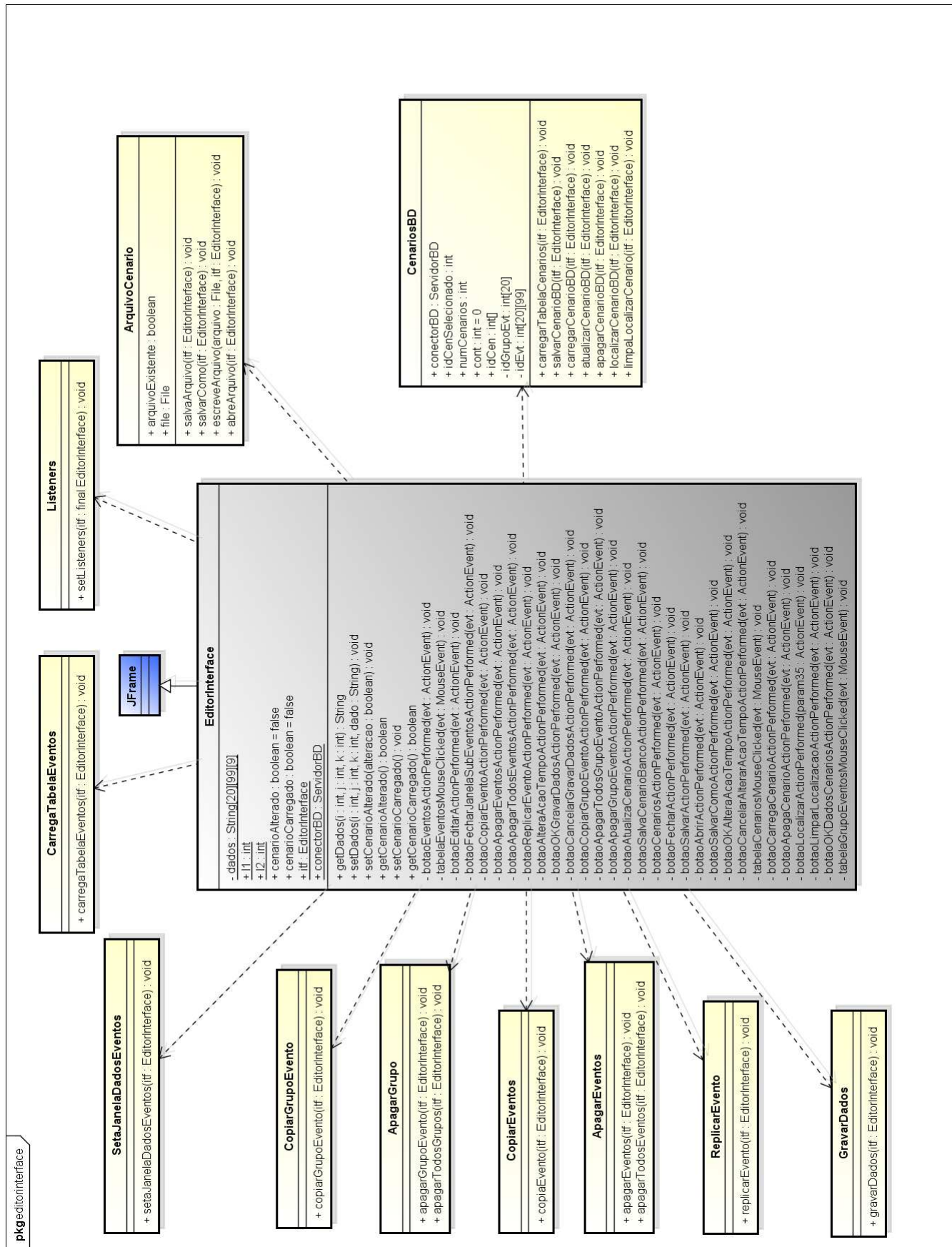


Figura 3.2: Diagrama de classes do pacote editorinterface

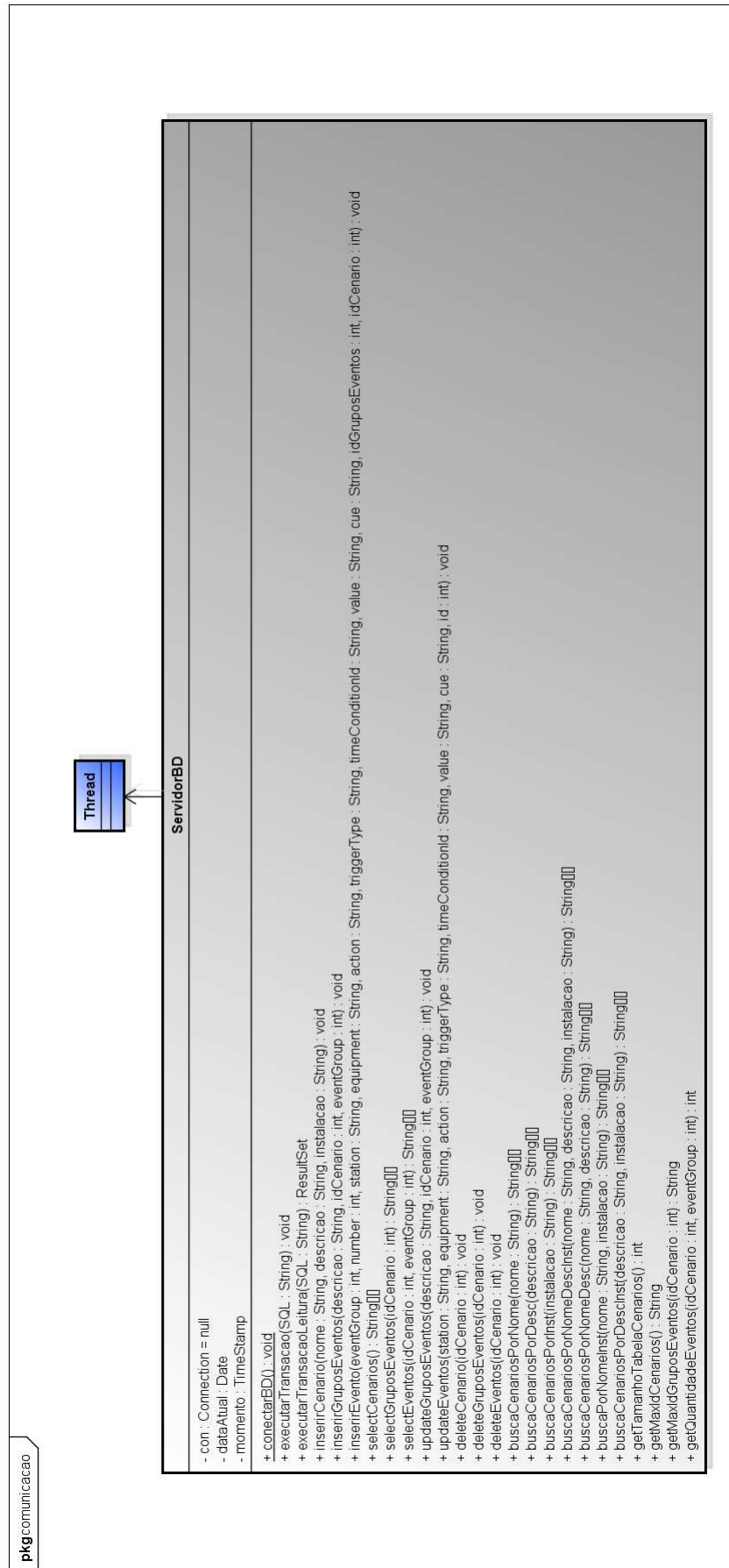


Figura 3.3: Diagrama de classes do pacote comunicacao

3.3 Construção da Interface Gráfica

Na construção da interface gráfica da ferramenta fez-se uso do pacote *Swing* do Java. As janelas são representadas pelo formulário *JFrame* e outros componentes do *Swing* são inseridos no formulário.

O *NetBeans* disponibiliza uma paleta de componentes que pertencem ao *Swing* e que podem ser utilizados para a construção da interface. Na Figura 3.4 é ilustrada esta paleta.

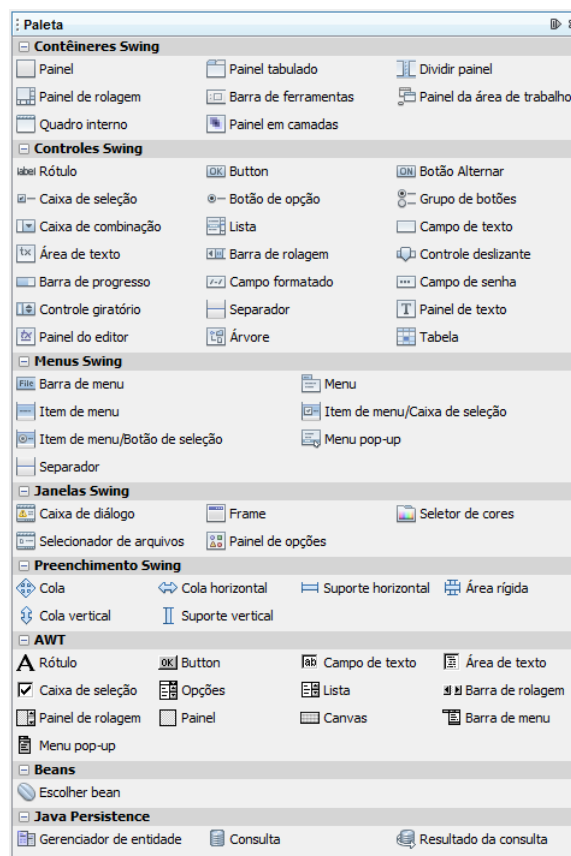


Figura 3.4: Paleta do *Netbeans* de componentes do *Swing*

A partir da paleta, rótulos, botões, tabelas, campos de texto, caixas de seleção, caixas de combinação, botões de opção, dentre outros componentes, foram inseridos no *JFrame*. Outros *JFrame*'s foram inseridos mas sendo o primeiro o principal.

3.4 Conclusão

Neste capítulo foi descrito o processo de construção do editor de cenários. Com o IDE *NetBeans* foi possível criar o programa Java e construir a interface gráfica. Com o *astah* foi construído o diagrama de classes, de forma a facilitar o entendimento do programa. Foi descrito também como instalar um servidor de banco de dados MySQL e como administrá-lo através de uma página WEB.

4

Editor de cenários no OTS do Simulop

Em razão das limitações do editor de cenários do OTS discutidas no Capítulo 2, sentiu-se a necessidade de criar uma nova ferramenta que trouxesse mais recursos e facilitasse a elaboração de cenários de treinamento pelos tutores.

O LIHM lançou a proposta de elaborar uma nova ferramenta que traz recursos ainda inexistentes no editor do OTS e permite ainda que os cenários sejam salvos em um banco de dados.

Na Figura 4.1 é ilustrado o fluxo de dados que ocorre entre o OTS, o editor de cenários e o banco de dados MySQL.

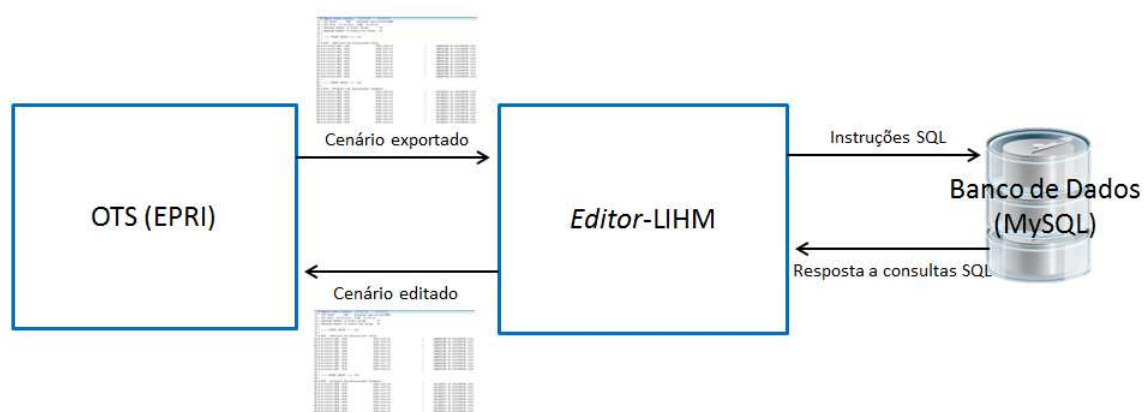


Figura 4.1: Fluxo de dados OTS - Editor LIHM - BD

4.1 Estrutura do Editor de Cenários do LIHM

A ideia inicial do editor do LIHM era mantê-lo com uma interface gráfica semelhante a do editor do OTS, de forma que os tutores pudessem estar familiarizados com o novo ambiente.

A janela principal do novo editor apresenta também a tabela com os 20 grupos de eventos, diferindo apenas nas colunas “*Valid*” e “*Active*”, inexistentes na tabela do novo editor. Estas colunas não foram possíveis já que o processo de validação e ativação são inerentes ao OTS.

Com relação ao editor antigo, na janela principal do novo editor foram retirados e adicionados botões. Os botões adicionados foram devido aos novos recursos oferecidos. Os botões retirados eram vinculados a recursos que trabalhavam em conjunto com o SIMULOP. Na Figura 4.2 é ilustrada a janela principal do editor do LIHM.

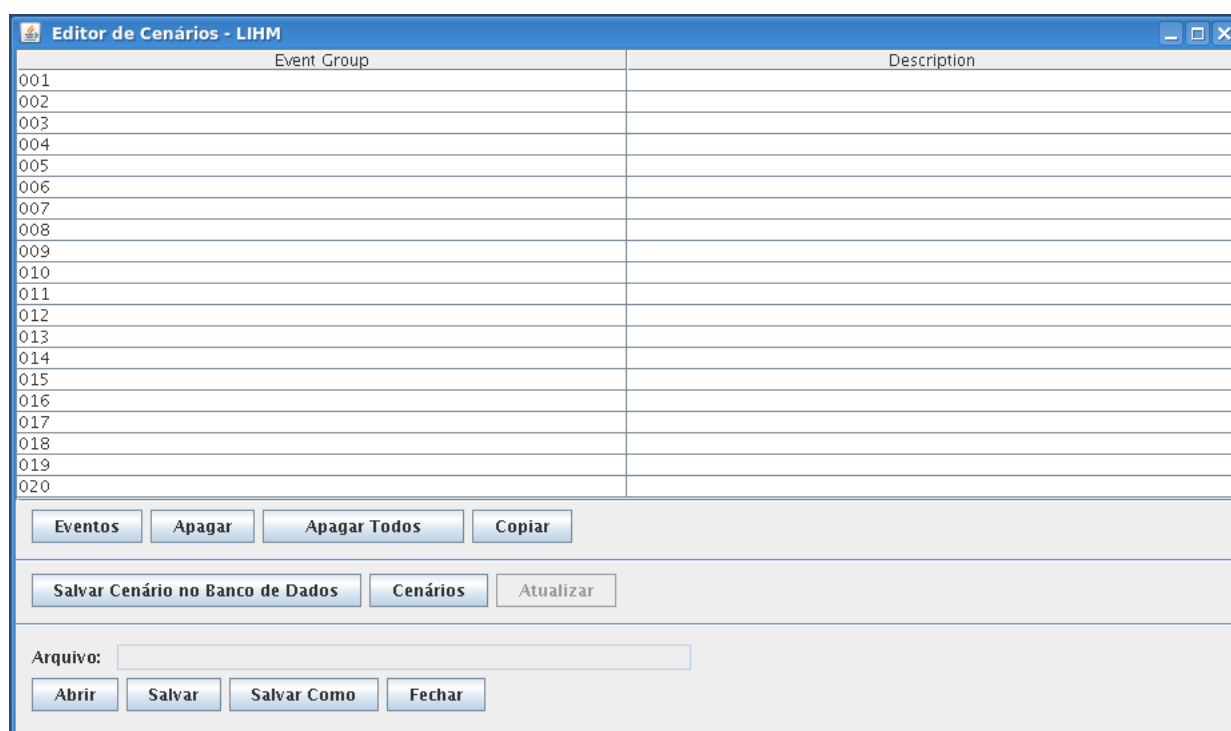


Figura 4.2: Janela Principal do editor de cenários do LIHM

4.2 Recursos do Editor

No editor do LIHM (Figura 4.2), os botões referentes à edição dos grupos de eventos ainda são mantidos. A importação ou exportação dos arquivos de cenário ficaram agora

incorporadas à janela principal do editor. No botão “Abrir” é possível abrir um arquivo de cenário existente, o botão “Salvar” salva um novo arquivo de cenário ou atualiza um cenário já aberto e editado. O botão “Salvar Como” salva com outro nome um arquivo de cenário editado. O campo de texto “Arquivo” exibe o caminho do diretório do arquivo aberto no momento. O botão “Fechar” encerra o editor.

O botão “Salvar Cenário no Banco de Dados” salva um cenário no banco de dados. Um cenário aberto a partir do banco de dados, após editado pode ser atualizado através do botão “Atualizar”.

O editor do LIHM permite ainda que um cenário também seja salvo em um banco de dados MySQL. Os cenários também podem ser carregados a partir do banco de dados. A manipulação dos cenários no banco de dados será tratada adiante.

4.2.1 Criar Eventos

A criação de eventos no editor do LIHM é semelhante a do OTS. Uma das diferenças é a inserção da descrição do grupo de eventos, que agora ocorre na própria tabela. Para editar os eventos, seleciona-se o grupo desejado e clica-se em “Eventos” ou um duplo clique na linha da tabela do grupo desejado e a janela ilustrada na Figura 4.3 será aberta.

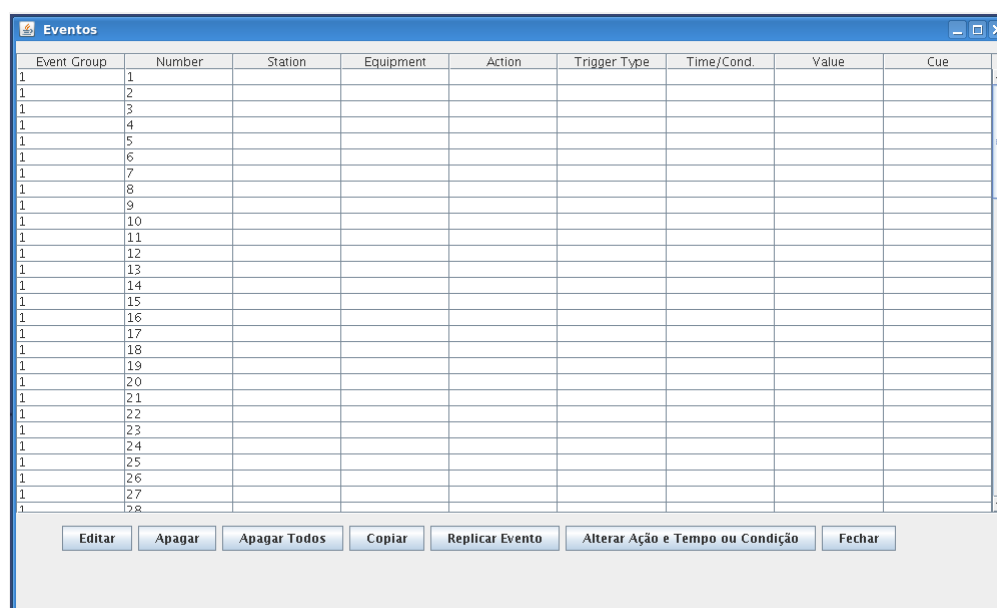


Figura 4.3: Eventos

Para editar um evento, seleciona-se o evento desejado na tabela e logo após clica-se no

botão “Editar”, ou um duplo clique na linha do evento e uma janela onde se insere os dados dos eventos será aberta (Figura 4.4).

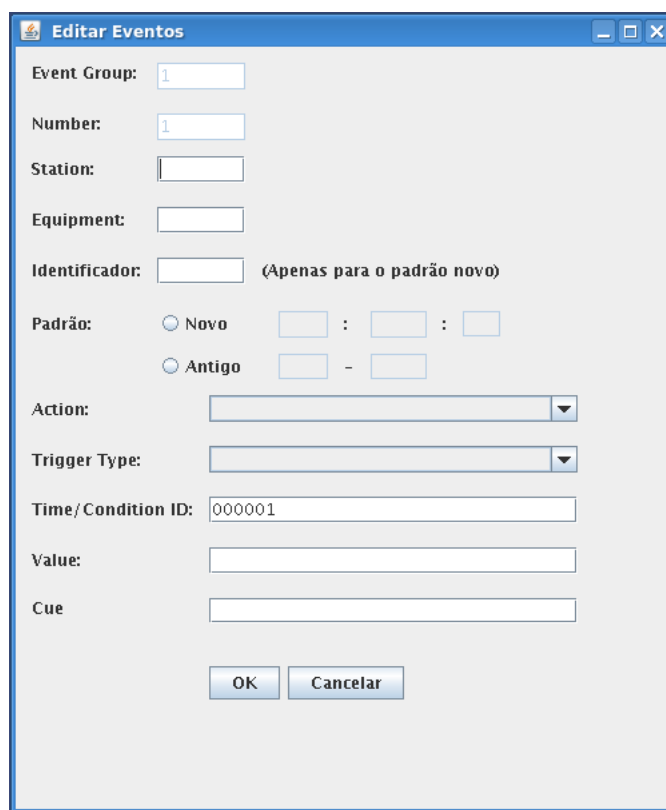


Figura 4.4: Inserindo os dados do evento

A escrita dos dados dos eventos acontece de forma semelhante a do editor do OTS. No campo “*Equipment*” agora é necessário informar apenas o código que identifica o equipamento na subestação, sem precisar repetir o nome da mesma, como acontecia no editor do OTS. O tutor ainda tem a possibilidade de configurar o padrão que a subestação utiliza, basta marcar “Padrão Novo” ou “Padrão Antigo” e os campos ao lado mostram uma previsão de como o nome do equipamento será exibido na tabela e no arquivo texto do cenário, conforme o padrão escolhido. Apenas o padrão novo possui o identificador do equipamento da tabela ANSI, então o campo “Identificador” é preenchido apenas neste caso. Se o padrão não for marcado e o usuário clicar em “OK”, uma mensagem de erro será exibida.

A seleção da ação sobre o equipamento, do tipo de disparo da ação e o preenchimento do restante dos campos não diferem do editor do OTS.

4.2.2 Copiar Eventos

O editor do LIHM permite ao tutor realizar cópias de eventos, enquanto que no editor do OTS só é possível copiar um grupo de eventos completo. Para copiar apenas um evento, seleciona-se o evento desejado e clica-se no botão “Copiar”. Será aberta uma nova janela solicitando o número do evento que receberá a cópia (Figura 4.5). Após informar o número e clicar em “OK” a cópia é concluída.

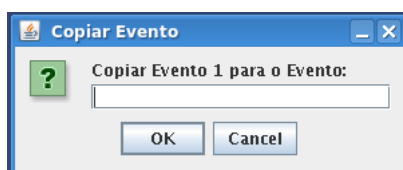


Figura 4.5: Copiar Eventos

Para copiar vários eventos por vez, selecionam-se na tabela os eventos a serem copiados. Para selecionar uma sequência de eventos, clica-se com o botão esquerdo do mouse e ainda mantendo o botão pressionado, arrasta-se o ponteiro do mouse sobre as linhas da tabela até que as linhas desejadas sejam selecionadas. Caso os eventos desejados não estejam em sequência, o usuário pode manter pressionada a tecla “Ctrl” e selecionar as linhas dos eventos a serem copiados. Após um destes passos, clica-se no botão “Copiar”. Será aberta uma nova janela informando quantos eventos foram selecionados e solicitando a partir de qual evento a cópia deve ser iniciada (Figura 4.6). Após informar o número e clicar em “OK” a cópia é concluída. Caso o número do evento inicial somado da quantidade de eventos a serem copiados ultrapasse o limite do 99º evento, uma mensagem de erro será exibida e a cópia não será possível.

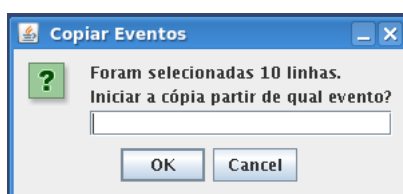


Figura 4.6: Copiar vários eventos

4.2.3 Replicar Evento

Um evento pode ser replicado em um intervalo desejado. Para fazer a réplica de um evento, seleciona-se o evento na tabela e clica-se no botão “Replicar Evento”. Será aberta uma janela solicitando o intervalo de eventos em que a réplica do evento será copiada (Figura 4.7). Os números do intervalo são separados pelo caractere “-”. Após informar o intervalo, clica-se em “OK” e o evento é replicado. Se um intervalo inválido for inserido o programa exibirá uma mensagem de erro.

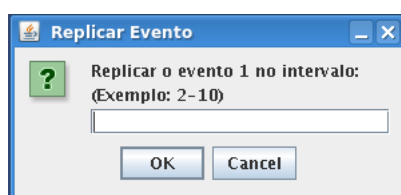


Figura 4.7: Replicar Evento

4.2.4 Alterar Ação dos Equipamentos e o Tempo ou Condição

Como discutido anteriormente, um grupo de eventos pode diferir de outro apenas no estado dos equipamentos. O editor do LIHM disponibiliza um recurso para alterar a ação dos equipamentos e o tempo ou condição, simultaneamente.

Para alterar a ação dos equipamentos e o tempo ou condição, selecionam-se na tabela os eventos desejados e clica-se no botão “Alterar Ação e Tempo ou Condição”. Será aberta uma janela com uma caixa de combinação com todas as ações aplicáveis aos equipamentos (Figura 4.8), da mesma forma que na caixa “Action” da janela da Figura 4.4 e com um campo para inserir o novo tempo ou condição. Para alterar um destes dois itens, deve se marcar as caixas de seleção correspondentes.

Desta forma, o tutor pode realizar a cópia de um grupo de eventos e nesta cópia altera-se a ação dos equipamentos de uma só vez, sem a necessidade de alterar um a um.

4.2.5 Salvar Cenário

Depois de feitas todas as edições desejadas o cenário pode agora ser salvo na forma de um arquivo texto. Este arquivo texto é gerado seguindo a exatamente mesma estrutura do

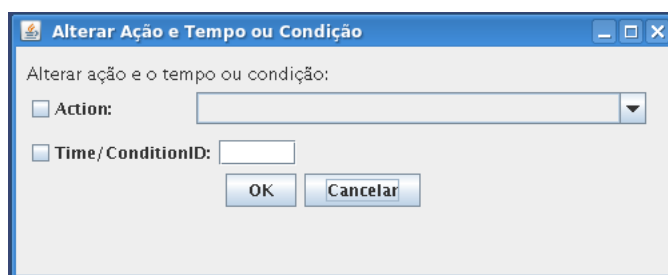


Figura 4.8: Alterar ação e tempo ou condição

arquivo gerado pelo OTS (exemplo de um arquivo gerado pelo OTS é ilustrado na Figura 2.11). Dessa forma, o tutor poderá editar um cenário a partir do editor do LIHM, salvá-lo, importá-lo no OTS e iniciar a simulação.

Para salvar um cenário, na janela principal do editor clica-se no botão “Salvar”. O seletor de arquivos ilustrado na Figura 4.9 será aberto e nele escolhe-se um nome para o arquivo e o diretório para salvá-lo.

Caso o cenário já esteja aberto no editor, clicando-se no botão “Salvar” as modificações serão salvas. Clicando-se no botão “Salvar Como” o cenário já aberto poderá ser salvo em outro arquivo.

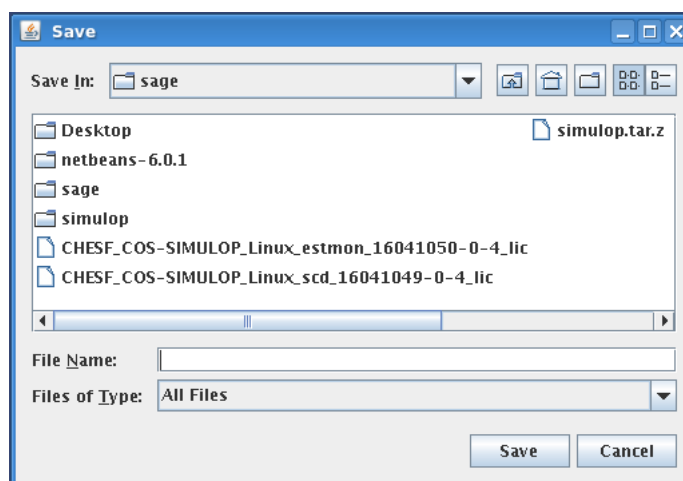


Figura 4.9: Salvar cenário

4.2.6 Abrir Cenário

Um cenário gerado pelo editor do OTS ou pelo editor do LIHM e que esteja salvo, poderá ser aberto para edição. Na janela principal do editor clica-se no botão “Abrir” e um seletor de arquivos semelhante ao ilustrado na Figura 4.9 será aberto para a escolha do arquivo.

4.2.7 Manipulação dos cenários em um banco de dados

Como discutido anteriormente, o editor do LIHM possibilita que os cenários sejam salvos em um banco de dados e também carregado a partir deste banco para um uso futuro.

No LIHM já se encontrava instalado um servidor de banco de dados MySQL na versão 5.1.38-community. A partir do *phpMyAdmin*, (sistema baseado em uma página WEB utilizado para administração de bancos de dados) foi criado um novo banco de dados chamado “cenariosSage”. O sistema de administração do banco de dados instalado do LIHM é acessado local ou remotamente através da página WEB <http://150.165.61.62/phpMyAdmin/>.

Salvar Cenário no Banco de Dados

No banco de dados “cenariosSage” um cenário contém três parâmetros, o nome para o cenário, a sua descrição e a instalação correspondente.

Para salvar um cenários no banco de dados, na janela principal do editor clica-se no botão “Salvar Cenário no Banco de Dados”. A janela ilustrada na Figura 4.10 será aberta e solicitará os parâmetros do cenário. Após inserir os dados clica-se em “OK” e o cenário será salvo.

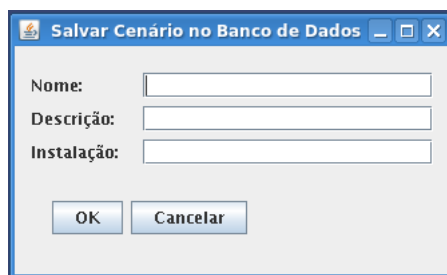


Figura 4.10: Salvar cenário no banco de dados

Carregar um Cenário a Partir do Banco de Dados

Para verificar os cenários armazenados no banco de dados e ainda carregá-lo para edição, na janela principal do editor clica-se no botão “Cenários”. A janela ilustrada na Figura 4.11 será aberta e nela existe uma tabela que é uma cópia da tabela armazenada no banco “cenariosSage”.

Para carregar um cenário, seleciona-se o cenário na tabela e clica-se no botão “Carregar Cenário” ou um duplo clique na linha da tabela e o cenário é carregado na janela principal do editor. Um cenário pode ainda ser removido do banco de dados, bastando selecionar o cenário desejado e clicar no botão “Apagar Cenário”.

Depois de feitas as edições no cenário e o usuário deseje salvar as atualizações no mesmo cenário carregado, na janela principal do editor clica-se no botão “Atualizar”. Este botão fica desabilitado e só é habilitado caso o cenário seja alterado.

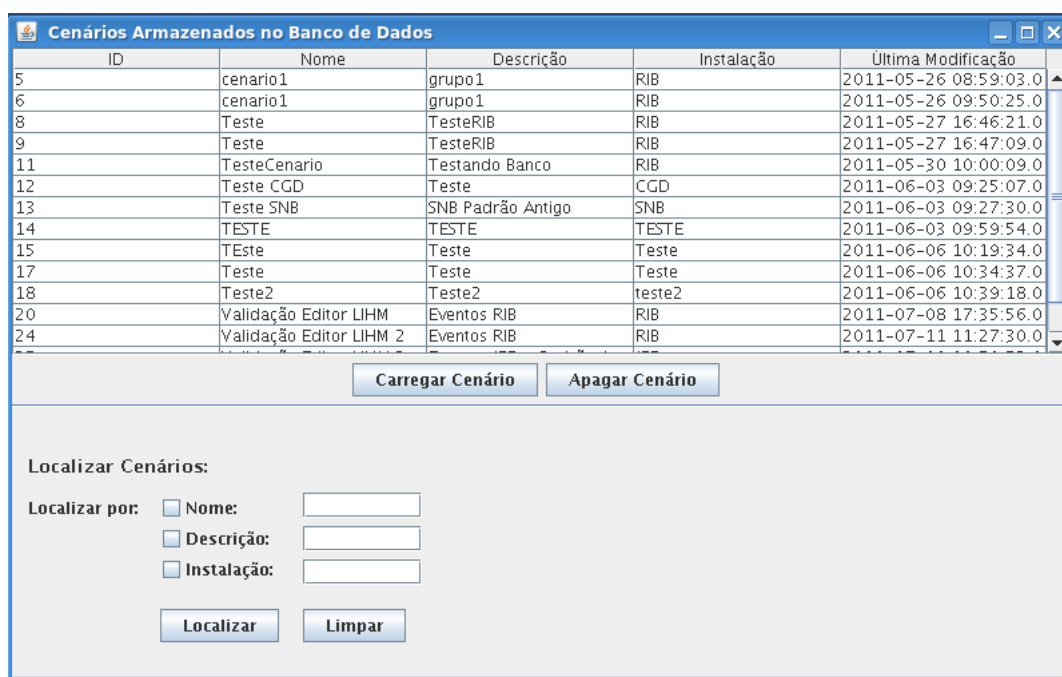


Figura 4.11: Cenários armazenados no banco de dados

Localizar um Cenário no Banco de Dados

Para facilitar a localização de um cenário armazenado no banco de dados, o editor do LIHM disponibiliza um recurso para localização de cenários. Os cenários podem ser localizados pelo nome, descrição, instalação ou pela combinação de qualquer um destes parâmetros.

Para localizar um cenário, na janela dos cenários (Figura 4.11) marcam-se as caixas correspondentes aos parâmetros no qual se deseja localizar o cenário e preenchem-se os campos correspondentes. Por fim, clica-se no botão “Localizar” e a tabela será atualizada com os cenários localizados. Para retornar a tabela com todos os cenários e limpar os campos

de localização, clica-se no botão “Limpar”.

4.3 Conclusão

Neste capítulo foram mostrados os recursos oferecidos pelo editor do LIHM para a edição de cenários. Com estes recursos a edição de cenários torna-se mais fácil e rápida. O cenário pode ser exportado para um arquivo texto e importado para o OTS. A partir de um banco de dados MySQL, os cenários podem ser armazenados e localizados facilmente.

5

Validação do Editor de Cenários do LIHM

No processo de validação do editor de cenários do LIHM foram utilizados os novos recursos oferecidos para gerar um cenário. Este cenário foi salvo em um arquivo, de forma a carregá-lo posteriormente no OTS, e no banco de dados. Além disto, a ferramenta foi testada nas plataformas Linux e Windows.

5.1 Preparando o SIMULOP

O SIMULOP (integração SAGE e OTS) funciona a partir do sistema operacional Linux. No LIHM encontrava-se um computador com a distribuição CentOS (<http://www.centos.org/>) do Linux e o SIMULOP instalados. Para utilizar o SIMULOP é necessário seguir uma sequência de passos rigorosa, caso contrário ele não iniciará corretamente.

No primeiro passo deve-se abrir o SAGE. A tela inicial ilustrada na Figura 5.1 será aberta. Para acessar o supervisor deve-se digitar o nome de usuário e senha. Após o *login* no sistema, na aba “Base de Tempo Real” ativa-se a base de tempo real no modo manual (Figura 5.2). Na aba “Programas” (Figura 5.3) clica-se no botão “Processos” para abrir a janela que lista todos os processos necessários para a simulação (Figura 5.4). Todos os processos devem ser selecionados e habilitados, exceto os dois últimos, que deverão ser habilitados após a inicialização do OTS.

Agora o OTS estará pronto para ser executado. Na aba “Programas” (Figura 5.3) clica-se no botão “Outros”. Na janela que será aberta (Figura 5.5) ativa-se o OTS e a janela principal do OTS abrirá (Figura 5.6). Nesta janela, em “Servers” e em “OTS Control” clica-se em “On” e depois em “Start”. Com o OTS funcionando podemos agora habilitar os dois últimos processos (Figura 5.4).



Figura 5.1: Tela inicial do Sage

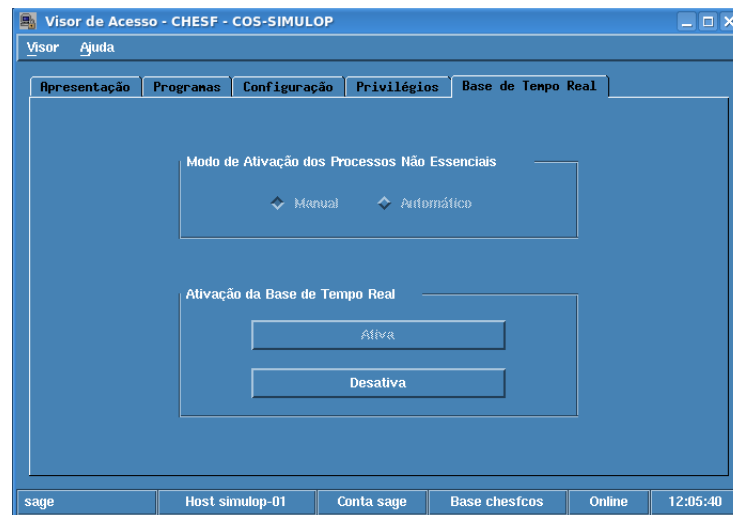


Figura 5.2: Ativando a Base de Tempo Real do SIMULOP

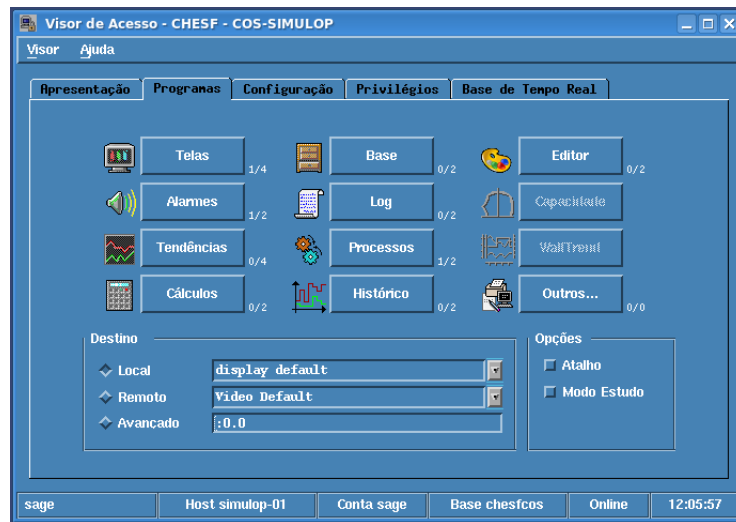


Figura 5.3: Aba “Programas” do SIMULOP

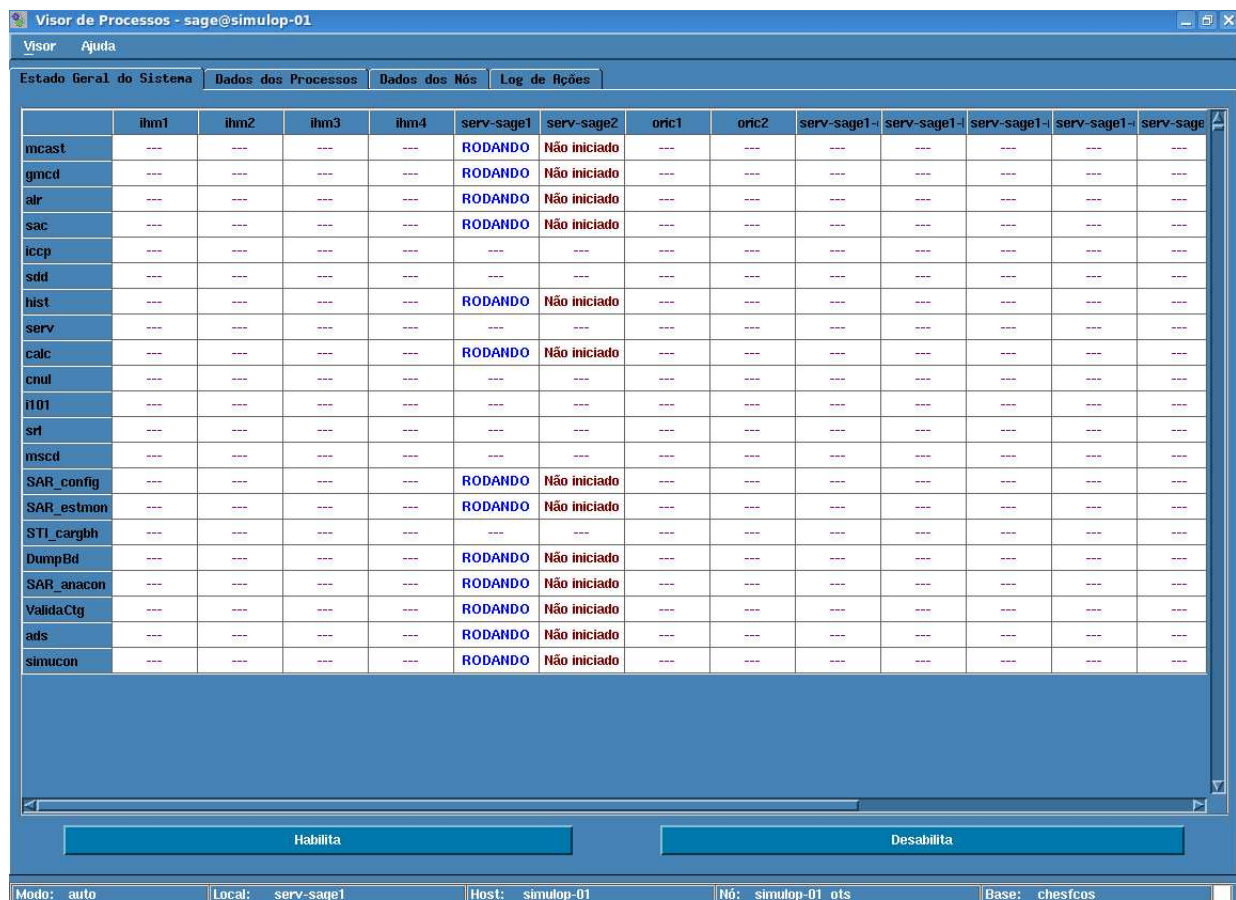


Figura 5.4: Processos do SIMULOP

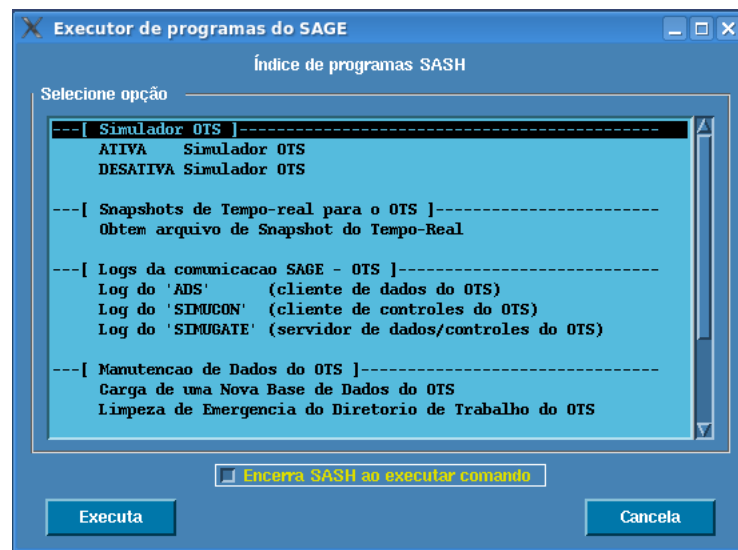


Figura 5.5: Executor de Programas do SAGE

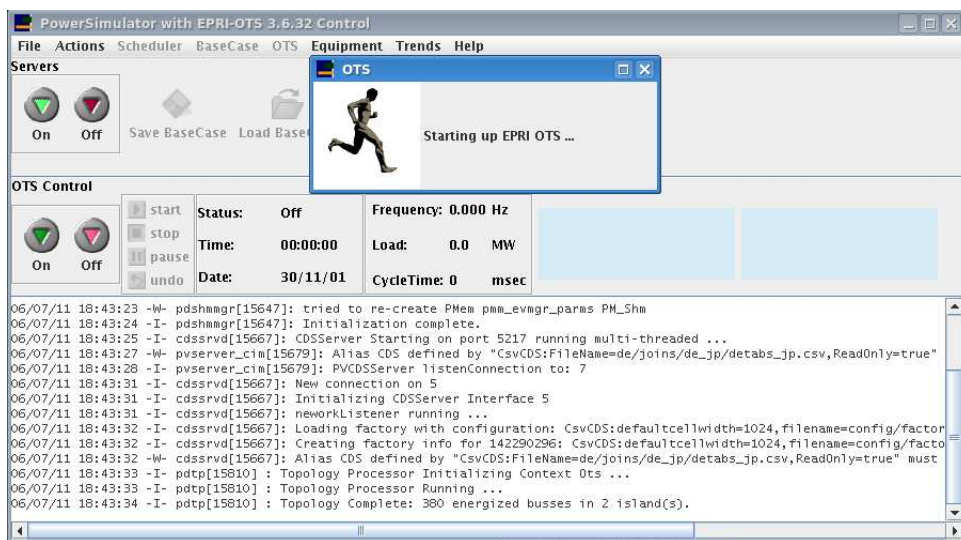


Figura 5.6: Janela principal do OTS

5.2 Estudo de Caso

O estudo de caso tem dois objetivos, testar o uso da ferramenta na geração de eventos e validar o arquivo gerado no SIMULOP. Para realizar esta validação foram criados eventos para duas subestações a Ribeirão e a Juazeiro II que possuem padrões de nomenclaturas diferentes, posteriormente os eventos serão executados no OTS para validar e atestar o documento gerado pela ferramenta desenvolvida.

5.2.1 Subestação Ribeirão (RIB) - PE

Na primeira etapa de validação foram criados alguns eventos para uma das subestações da CHESF. A subestação escolhida foi a Ribeirão, que se localiza na cidade de Ribeirão-PE. Esta subestação é identificada pela CHESF por “RIB” e utiliza o padrão mais novo para nomear os seus equipamentos e será simulado o desligamento da subestação com bloqueios dos disjuntores.

Foram criados 4 grupos de eventos com cada um destes contendo 11 eventos. Cada grupo foi criado para realizar uma ação sobre alguns disjuntores da subestação RIB. O primeiro grupo realizava a abertura dos disjuntores, o segundo o bloqueio após abertura, o terceiro o desbloqueio e o quarto o fechamento destes. Vale salientar que estes eventos foram criados baseados em treinamentos já elaborados e utilizados pela empresa.

O primeiro evento consiste na abertura dos disjuntores, os eventos foram criados no grupo de número “1” e a descrição é: “RIB - Abertura dos Disjuntores (Trip)” (Figura 5.7). Dentro deste grupo, foi criado um evento para um dos disjuntores, com a ação “BKT” e com tempo relativo ao início da simulação de 1s (Figura 5.8). Este evento foi replicado no intervalo 2-11 e em cada cópia da réplica alterou-se apenas o nome dos disjuntores. Na Figura 5.9 estão listados todos os disjuntores que deverão ser abertos após ativação.

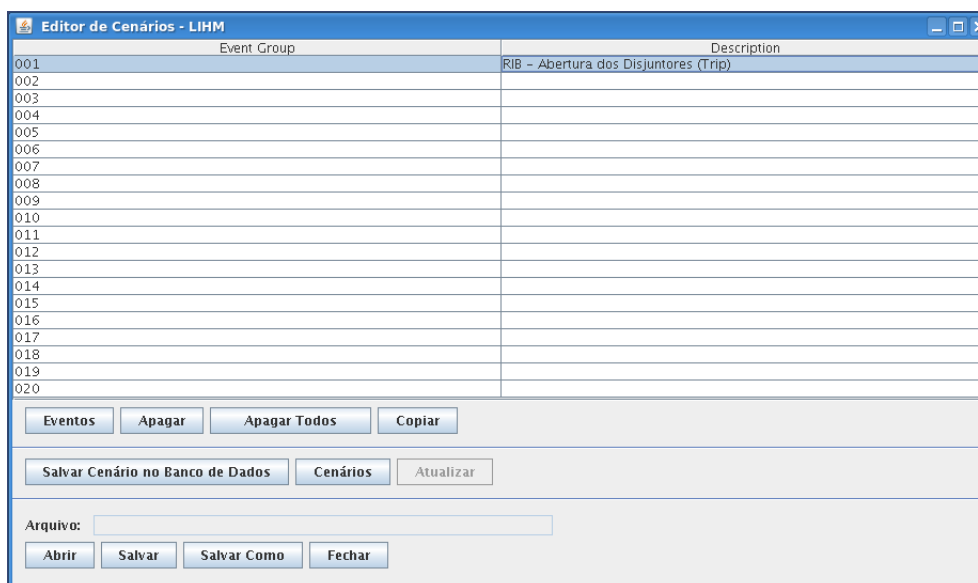


Figura 5.7: Criando o grupo de eventos 1

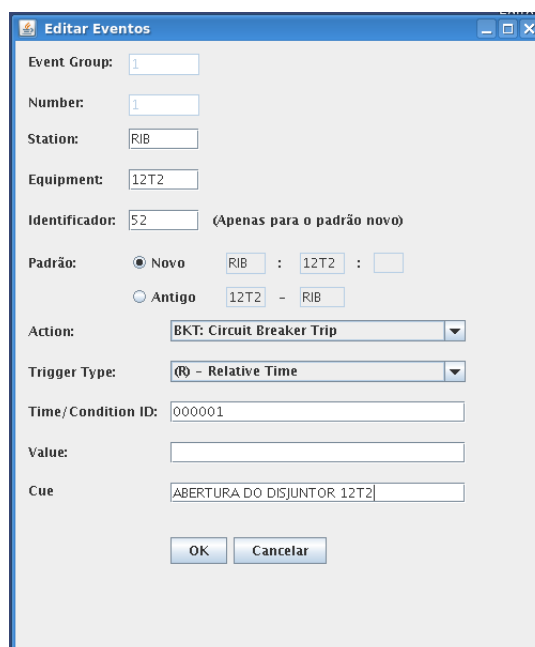


Figura 5.8: Editando Evento

Event Group	Number	Station	Equipment	Action	Trigger Type	Time/Cond.	Value	Cue
1	1	RIB ...	RIB:12T2:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12T2 ...
1	2	RIB ...	RIB:12T3:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12T3 ...
1	3	RIB ...	RIB:12L1:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L1 ...
1	4	RIB ...	RIB:12L2:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L2 ...
1	5	RIB ...	RIB:12L3:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L3 ...
1	6	RIB ...	RIB:12L4:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L4 ...
1	7	RIB ...	RIB:12L5:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L5 ...
1	8	RIB ...	RIB:12L6:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L6 ...
1	9	RIB ...	RIB:12L7:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L7 ...
1	10	RIB ...	RIB:12L8:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L8 ...
1	11	RIB ...	RIB:12L9:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L9 ...
1	12							
1	13							
1	14							
1	15							
1	16							
1	17							
1	18							
1	19							
1	20							
1	21							
1	22							
1	23							
1	24							
1	25							
1	26							
1	27							
1	28							

Figura 5.9: Eventos do grupo 1

Depois de criado o grupo 1, foi feita uma cópia do grupo 1 para o grupo 2. O grupo 2 representa o evento de bloqueio das chaves após a sua abertura. Assim, nos eventos do grupo 2, selecionou-se todos os eventos e alterou-se apenas a ação do equipamentos de “BKT” para “BKD” (Figura 5.10) e as mensagens que aparecerão na tela de alarmes do supervisor. Para criação dos grupos 3 e 4, foi realizado o mesmo procedimento anterior, alterando-se a ação para “BKE” e “BKC”, respectivamente. A tabela dos grupos de eventos ficou agora na forma ilustrada na Figura 5.11.

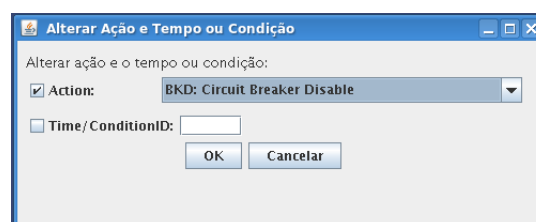


Figura 5.10: Alterando a ação de todos os eventos

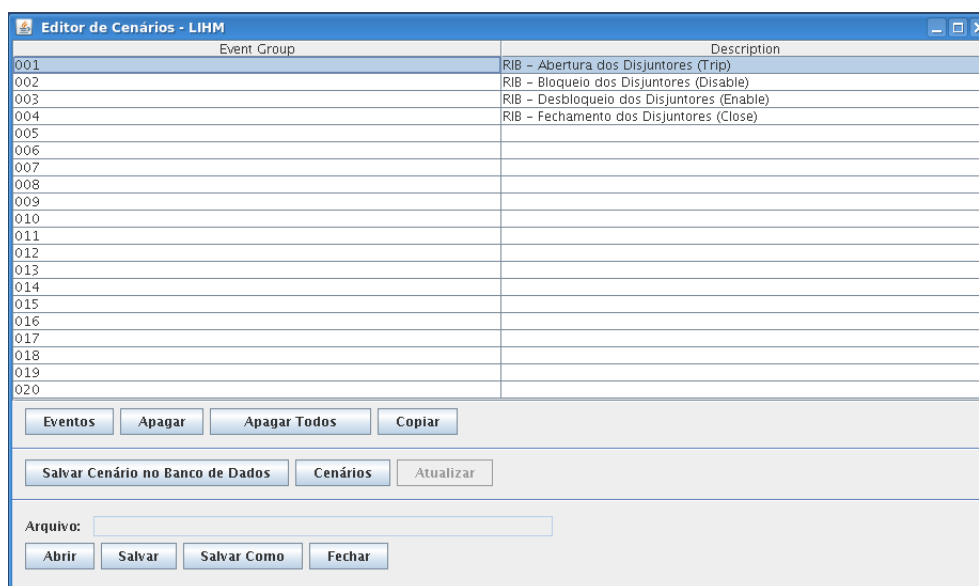


Figura 5.11: Grupos de eventos para validação

Para que estes eventos pudessem ser importados no OTS o cenário foi salvo no arquivo "Validacao_Eventos_RIB" (Figura 5.12). Um trecho do arquivo texto gerado é ilustrado na Figura 5.13.

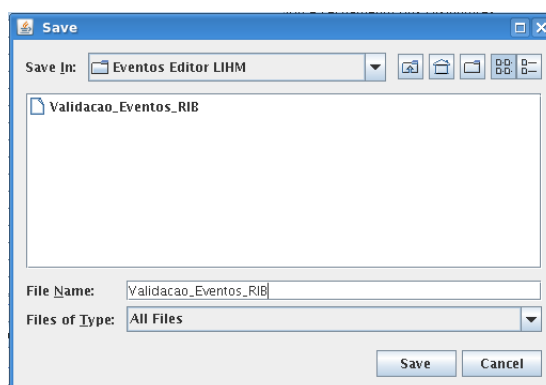


Figura 5.12: Salvando o cenário para validação

```

1 | Export Event Library 00/00/00 00:00:00
2 * OTS Model EXP Database eprlots/db/DBSA
3 * OTS Date 00/00/0000 TIME 00:00:00
4 * Maximum Number of Event Groups 20
5 * Maximum Number of Events Per Group 99
6 *
7 * ---- EVENT GROUP ---- 001
8 *
9 G:RIB - Abertura dos Disjuntores
10 E:R:000001:BKT :RIB :RIB:12T2:52 : : :ABERTURA DO DISJUNTOR 12T2
11 E:R:000001:BKT :RIB :RIB:12T3:52 : : :ABERTURA DO DISJUNTOR 12T3
12 E:R:000001:BKT :RIB :RIB:12L1:52 : : :ABERTURA DO DISJUNTOR 12L1
13 E:R:000001:BKT :RIB :RIB:12L2:52 : : :ABERTURA DO DISJUNTOR 12L2
14 E:R:000001:BKT :RIB :RIB:12L3:52 : : :ABERTURA DO DISJUNTOR 12L3
15 E:R:000001:BKT :RIB :RIB:12L4:52 : : :ABERTURA DO DISJUNTOR 12L4
16 E:R:000001:BKT :RIB :RIB:12L5:52 : : :ABERTURA DO DISJUNTOR 12L5
17 E:R:000001:BKT :RIB :RIB:12L6:52 : : :ABERTURA DO DISJUNTOR 12L6
18 E:R:000001:BKT :RIB :RIB:12L7:52 : : :ABERTURA DO DISJUNTOR 12L7
19 E:R:000001:BKT :RIB :RIB:12L8:52 : : :ABERTURA DO DISJUNTOR 12L8
20 E:R:000001:BKT :RIB :RIB:12L9:52 : : :ABERTURA DO DISJUNTOR 12L9
21 *
22 * ---- EVENT GROUP ---- 002
23 *
24 G:RIB - Bloqueio dos Disjuntores
25 E:R:000001:BKD :RIB :RIB:12T2:52 : : :BLOQUEIO DO DISJUNTOR 12T2
26 E:R:000001:BKD :RIB :RIB:12T3:52 : : :BLOQUEIO DO DISJUNTOR 12T3
27 E:R:000001:BKD :RIB :RIB:12L1:52 : : :BLOQUEIO DO DISJUNTOR 12L1
28 E:R:000001:BKD :RIB :RIB:12L2:52 : : :BLOQUEIO DO DISJUNTOR 12L2
29 E:R:000001:BKD :RIB :RIB:12L3:52 : : :BLOQUEIO DO DISJUNTOR 12L3
30 E:R:000001:BKD :RIB :RIB:12L4:52 : : :BLOQUEIO DO DISJUNTOR 12L4
31 E:R:000001:BKD :RIB :RIB:12L5:52 : : :BLOQUEIO DO DISJUNTOR 12L5
32 E:R:000001:BKD :RIB :RIB:12L6:52 : : :BLOQUEIO DO DISJUNTOR 12L6
33 E:R:000001:BKD :RIB :RIB:12L7:52 : : :BLOQUEIO DO DISJUNTOR 12L7
34 E:R:000001:BKD :RIB :RIB:12L8:52 : : :BLOQUEIO DO DISJUNTOR 12L8
35 E:R:000001:BKD :RIB :RIB:12L9:52 : : :BLOQUEIO DO DISJUNTOR 12L9

```

Figura 5.13: Arquivo texto do cenário de validação

Executando os eventos

Primeiramente o cenário editado no editor do LIHM foi importado para o OTS. A Figura 5.14 ilustra a importação do arquivo “Validacao_Eventos_RIB”. Depois disso, os grupos de eventos do OTS já se encontravam preenchidos com os dados do cenário importado (Figura 5.15).

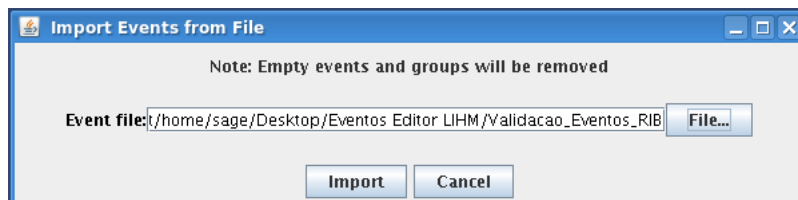
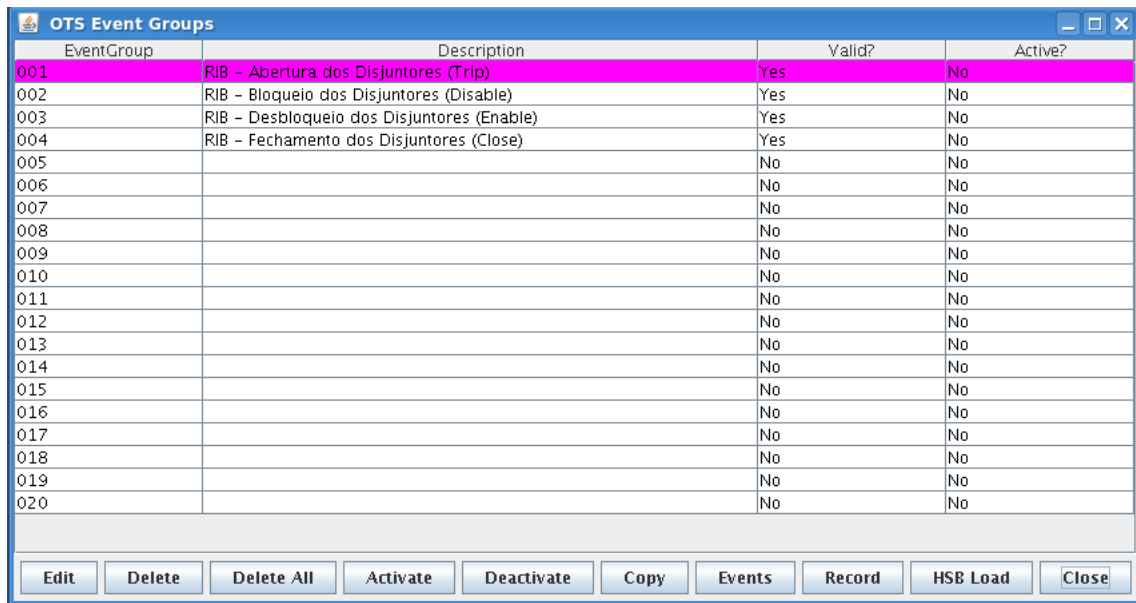


Figura 5.14: Importando cenário de validação no OTS



The screenshot shows a window titled "OTS Event Groups" with a table containing 20 rows. The first row is highlighted in pink. Below the table is a row of buttons: Edit, Delete, Delete All, Activate, Deactivate, Copy, Events, Record, HSB Load, and Close.

EventGroup	Description	Valid?	Active?
001	RIB - Abertura dos Disjuntores (Trip)	Yes	No
002	RIB - Bloqueio dos Disjuntores (Disable)	Yes	No
003	RIB - Desbloqueio dos Disjuntores (Enable)	Yes	No
004	RIB - Fechamento dos Disjuntores (Close)	Yes	No
005		No	No
006		No	No
007		No	No
008		No	No
009		No	No
010		No	No
011		No	No
012		No	No
013		No	No
014		No	No
015		No	No
016		No	No
017		No	No
018		No	No
019		No	No
020		No	No

Figura 5.15: Cenário criado no editor do LIHM e importado para o OTS

Quando importados, os grupos de eventos são automaticamente validados pelo OTS e tornam-se prontos para serem executados pelo tutor do treinamento.

Antes da execução dos eventos, no SAGE deve ser aberto o Visor de Telas de forma que o operador possa visualizar o estado dos equipamentos na subestação. Para visualizar a tela da subestação RIB, na aba “Programas” (Figura 5.3) clica-se no botão “Telas” e o Visor de Telas abrirá (Figura 5.16). No Visor de Telas, na caixa de combinação no canto superior direito, procura-se pelo item “INDICE_SUBSISTEMAS”. A próxima tela mostrará uma divisão das subestações em regiões. Para esta validação, será escolhida a região Leste e em seguida a subestação Ribeirão (Figuras 5.17 e 5.18).

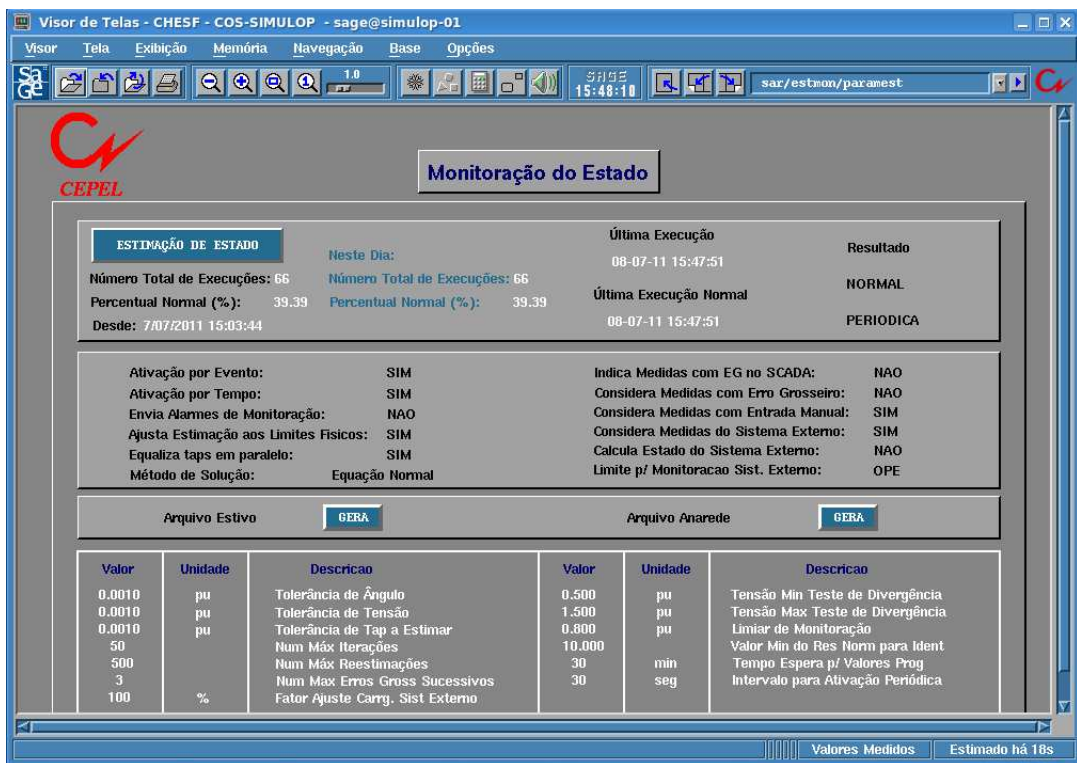


Figura 5.16: Visor de Telas do SAGE

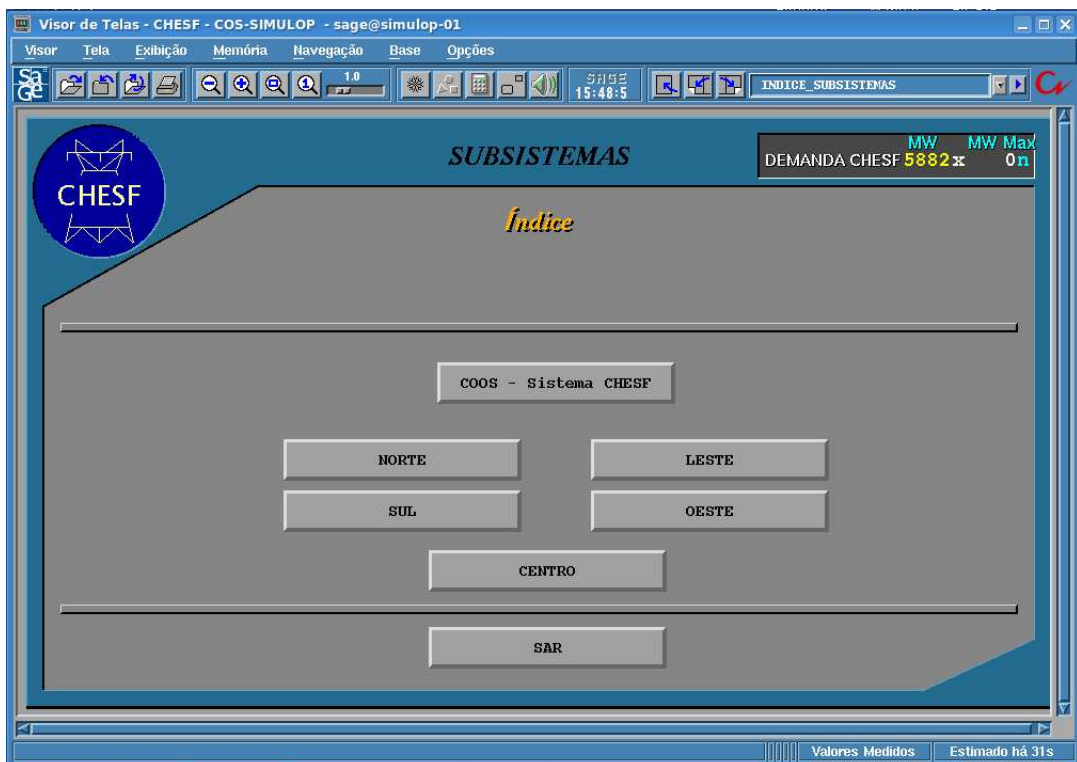


Figura 5.17: Regiões das subestações da CHESF

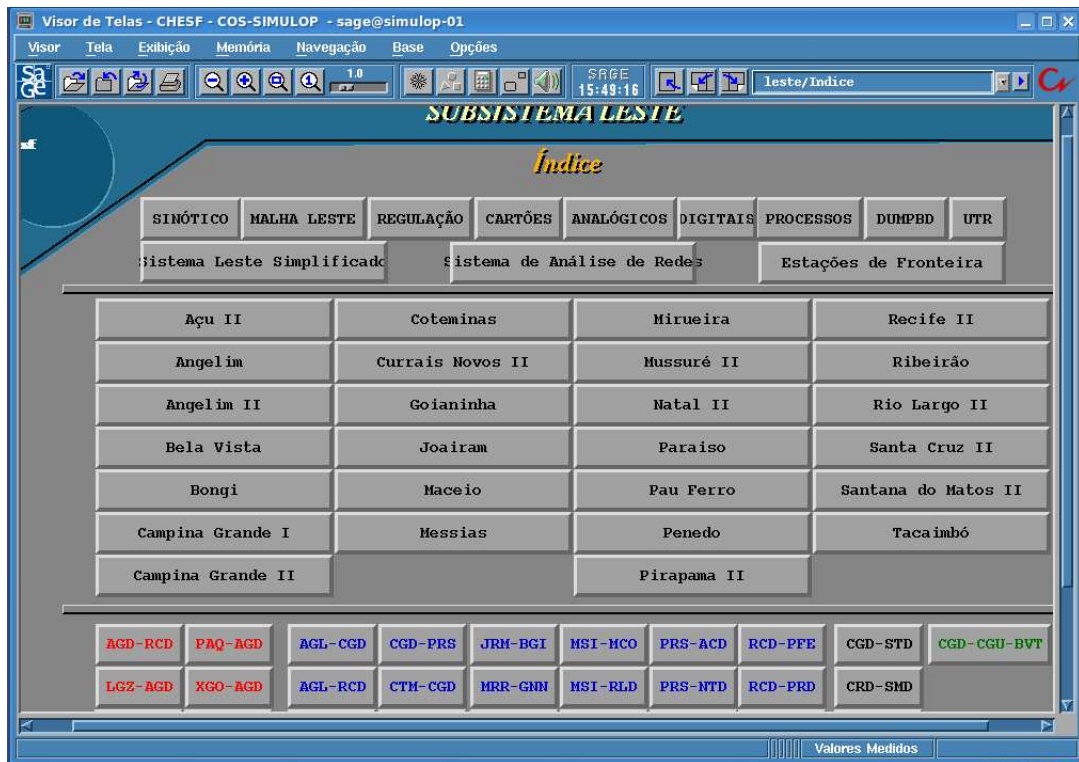


Figura 5.18: Subestações da Região Leste

A tela da subestação RIB (Figura 5.19) exibe o estado de alguns equipamentos antes dos eventos serem executados. A ilustração na forma de um quadrado corresponde a um disjuntor. A cor verde indica que o disjuntor está aberto e a vermelha que o mesmo está fechado.

Agora, o grupo de eventos “1” é ativado. Após 1s da ativação alguns disjuntores da RIB são abertos. A Figura 5.20 ilustra a tela da RIB após a ativação deste grupo. Os disjuntores abertos passam a ter agora a cor verde. O disjuntor 12L9 já se encontrava aberto antes da ativação do evento, então, continuou no mesmo estado.

Pode-se observar que a tensão na barra 02BP encontra-se em 0.0 kV, já que os disjuntores 12T2 e 12T3 que ligam a saída dos transformadores a barra, foram abertos.

Depois de ativado o grupo de eventos 1, ativou-se o grupo 2 e em seguida o grupo 3, que realizam o bloqueio e o desbloqueio de alguns disjuntores da RIB, respectivamente. Caso os disjuntores continuassem bloqueados não seria possível fechá-los. Com os disjuntores desbloqueados, foi ativado o grupo 4.

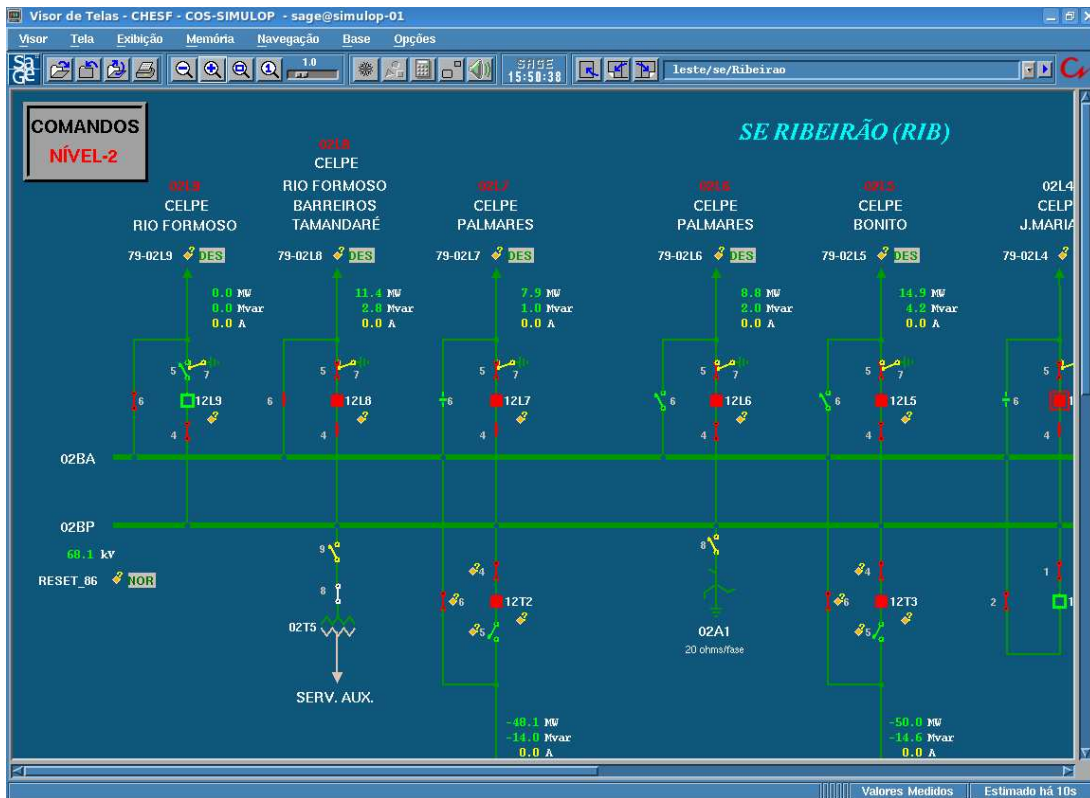


Figura 5.19: Subestação RIB antes da ativação dos eventos

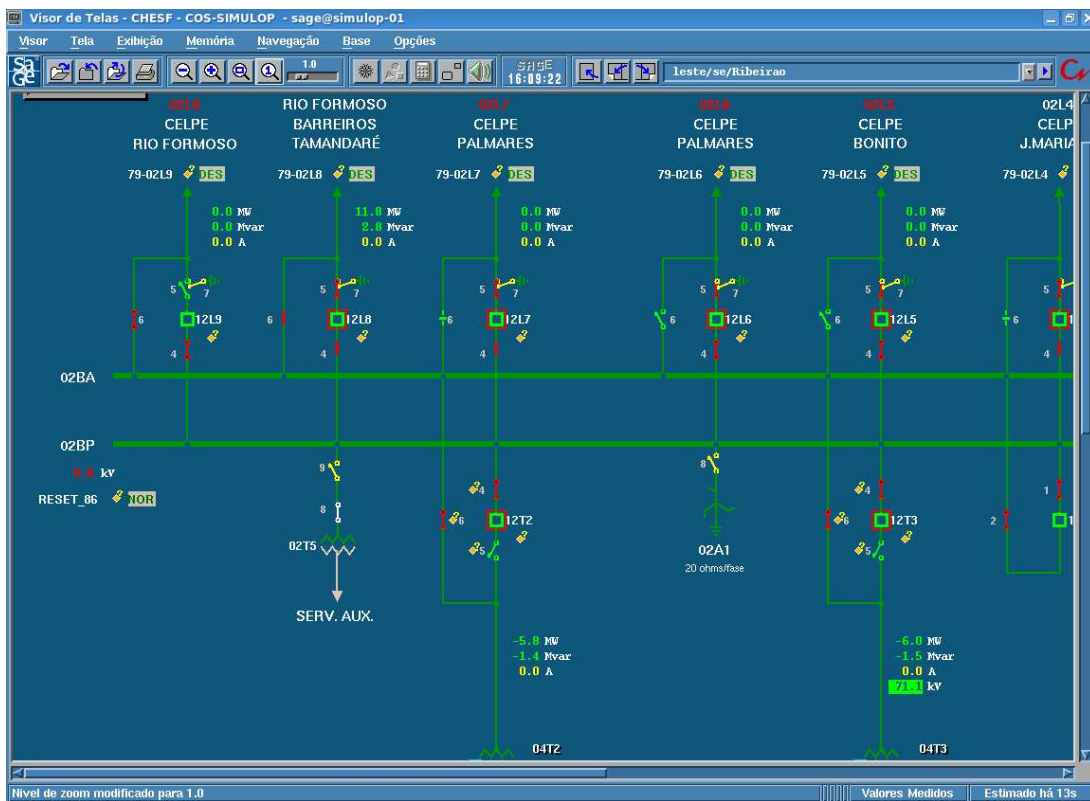


Figura 5.20: Subestação RIB depois da ativação do grupo de eventos 1

A Figura 5.21 ilustra a tela da RIB após a ativação do grupo 4. A tensão na barra 02BP agora difere da tensão de antes da ativação do grupo de eventos 1. Isto ocorreu devido à ligação de mais uma linha ao barramento através do fechamento do disjuntor 12L9.



Figura 5.21: Subestação RIB depois da ativação do grupo de eventos 4

O Visor de Alarmes do SAGE deve indicar a ocorrência destes eventos. Para abrir o Visor de Alarmes, na aba “Programas” do SAGE (Figura 5.3) clica-se no botão “Alarmes”. Os alarmes referentes à região Leste indicam a abertura e o fechamento dos disjuntores (Figura 5.22). Os alarmes gerais (Figura 5.23) indicam ainda as mensagens colocadas no campo “Cue” do editor de cenários. Um alarme sonoro é soado sempre que Visor de Alarmes indica alguma alteração no estado dos equipamentos.

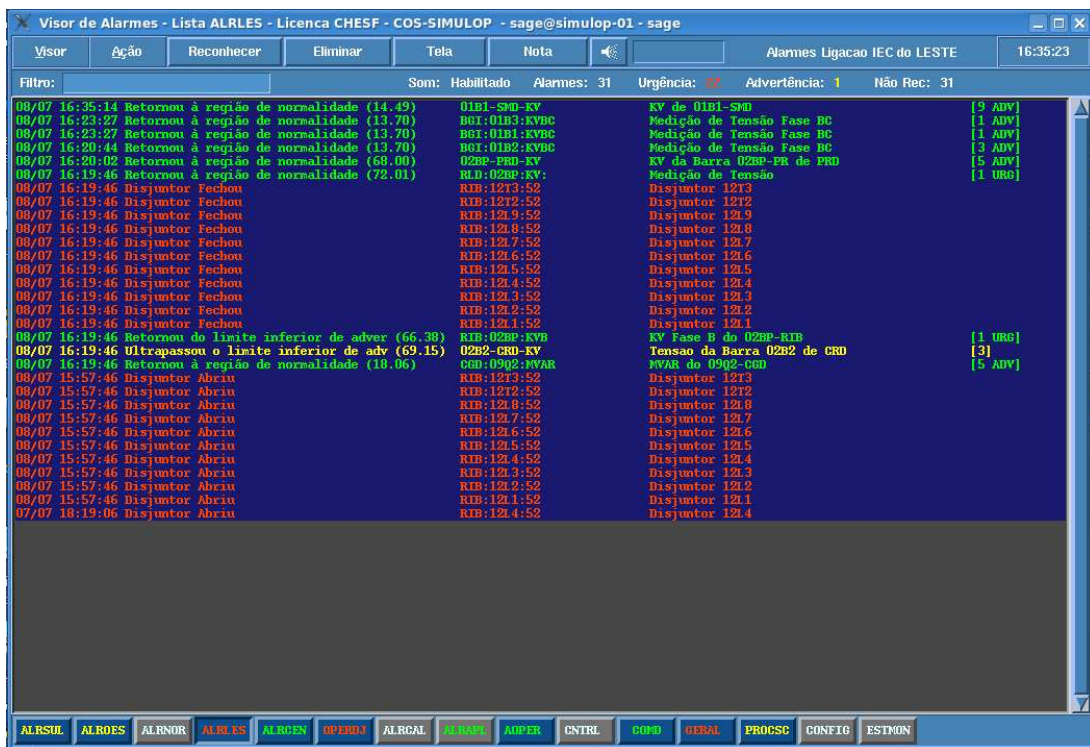


Figura 5.22: Visor de Alarmes da região Leste



Figura 5.23: Visor de Alarmes geral indica a mudança de estado dos equipamentos na RIB

Outra maneira de simular estes eventos é colocá-los em um só grupo e fazer com que cada ação seja executada após um tempo em relação à outra. Para isto, seleciona-se a sequência de eventos que realizam a abertura e os copiam para outro intervalo. Através do botão “Alterar Ação e Tempo ou Condição” altera-se a ação e o tempo de disparo da ação da nova sequência de eventos (Figura 5.24). O mesmo é feito para as próximas sequências. Na Figura 5.25 são ilustrados estes eventos distribuídos em um só grupo.

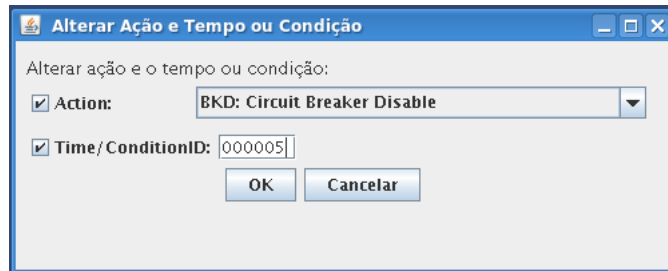


Figura 5.24: Alterando a ação e o tempo de disparo de uma sequência de eventos

Event Group	Number	Station	Equipment	Action	Trigger Type	Time/Cond.	Value	Cue
1	1	RIB ...	RIB: 12T2:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12T2 ...
1	2	RIB ...	RIB: 12T3:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12T3 ...
1	3	RIB ...	RIB: 12L1:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L1 ...
1	4	RIB ...	RIB: 12L2:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L2 ...
1	5	RIB ...	RIB: 12L3:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L3 ...
1	6	RIB ...	RIB: 12L4:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L4 ...
1	7	RIB ...	RIB: 12L5:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L5 ...
1	8	RIB ...	RIB: 12L6:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L6 ...
1	9	RIB ...	RIB: 12L7:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L7 ...
1	10	RIB ...	RIB: 12L8:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L8 ...
1	11	RIB ...	RIB: 12L9:52 ...	BKT	R	000001		ABERTURA DO DISJUNTOR 12L9 ...
1	12	RIB ...	RIB: 12T2:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12T2 ...
1	13	RIB ...	RIB: 12T3:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12T3 ...
1	14	RIB ...	RIB: 12L1:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L1 ...
1	15	RIB ...	RIB: 12L2:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L2 ...
1	16	RIB ...	RIB: 12L3:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L3 ...
1	17	RIB ...	RIB: 12L4:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L4 ...
1	18	RIB ...	RIB: 12L5:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L5 ...
1	19	RIB ...	RIB: 12L6:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L6 ...
1	20	RIB ...	RIB: 12L7:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L7 ...
1	21	RIB ...	RIB: 12L8:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L8 ...
1	22	RIB ...	RIB: 12L9:52 ...	BKD	R	000005		BLOQUEIO DO DISJUNTOR 12L9 ...
1	23	RIB ...	RIB: 12T2:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12T2 ...
1	24	RIB ...	RIB: 12T3:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12T3 ...
1	25	RIB ...	RIB: 12L1:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12L1 ...
1	26	RIB ...	RIB: 12L2:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12L2 ...
1	27	RIB ...	RIB: 12L3:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12L3 ...
1	28	RIB ...	RIB: 12L4:52 ...	BKE	R	000010		DES Bloqueio DO DISJUNTOR 12L4 ...

Figura 5.25: Todos os eventos em um só grupo

5.2.2 Subestação Juazeiro da Bahia II (JZD) - BA

Apenas para demonstrar a utilização do editor com os equipamentos nomeados com o padrão antigo, foram criados dois grupos de eventos e eles simularam o desligamento e posteriormente o religamento da subestação (abertura e fechamento dos disjuntores). A ação em todos os eventos é disparada após 1s relativo ao início da simulação. A mudança no padrão afeta apenas os nomes dos equipamentos e o padrão de cores para os estados dos equipamentos vistos nas telas do SAGE não é alterado.

Na Figura 5.26 é ilustrado o editor do LIHM com os dois grupos de eventos e a Figura 5.27 os eventos correspondentes ao grupo 1. Os eventos do grupo 2 utilizam os mesmos equipamentos do 1 e diferem apenas na ação, de “BKT” para “BKC”. O arquivo de cenário foi salvo com o nome “Validacao_Eventos_JZD_Padrao_Antigo”.

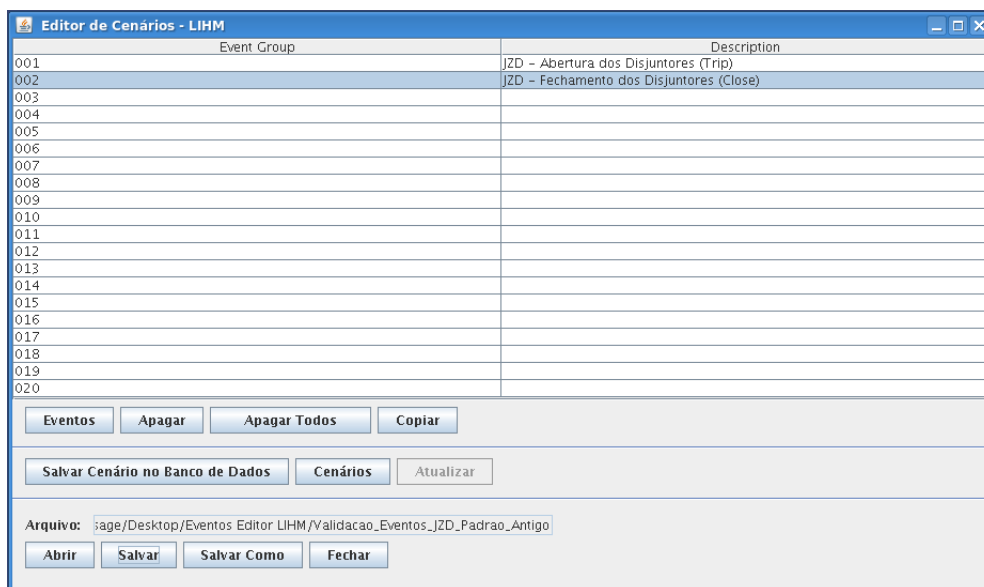


Figura 5.26: Grupos de eventos para a segunda etapa de validação

Da mesma forma que anteriormente, o arquivo de cenário foi importado para o OTS. Primeiro ativou-se o grupo de eventos que realiza a abertura dos disjuntores e em seguida o grupo que realizava o fechamento dos mesmos.

Event Group	Number	Station	Equipment	Action	Trigger Type	Time/Cond.	Value	Cue
1	1	JZD	12C1-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C1
1	2	JZD	12C2-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C2
1	3	JZD	12C3-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C3
1	4	JZD	12C4-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C4
1	5	JZD	12C5-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C5
1	6	JZD	12C6-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C6
1	7	JZD	12C7-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C7
1	8	JZD	12C8-JZD	BKT	R	000001		ABERTURA DO DISJUNTOR 12C8
1	9							
1	10							
1	11							
1	12							
1	13							
1	14							
1	15							
1	16							
1	17							
1	18							
1	19							
1	20							
1	21							
1	22							
1	23							
1	24							
1	25							
1	26							
1	27							
1	28							

Figura 5.27: Eventos do grupo 1

Nas Figuras 5.28 e 5.29 é ilustrada a subestação JZD antes e depois da ativação dos eventos, respectivamente. Após a ativação do segundo grupo de eventos (fechamento dos disjuntores) a subestação encontrava-se da mesma forma que antes da ativação do primeiro grupo. Os alarmes gerais indicam a mudança do estado nos disjuntores da JZD (Figura 5.30).

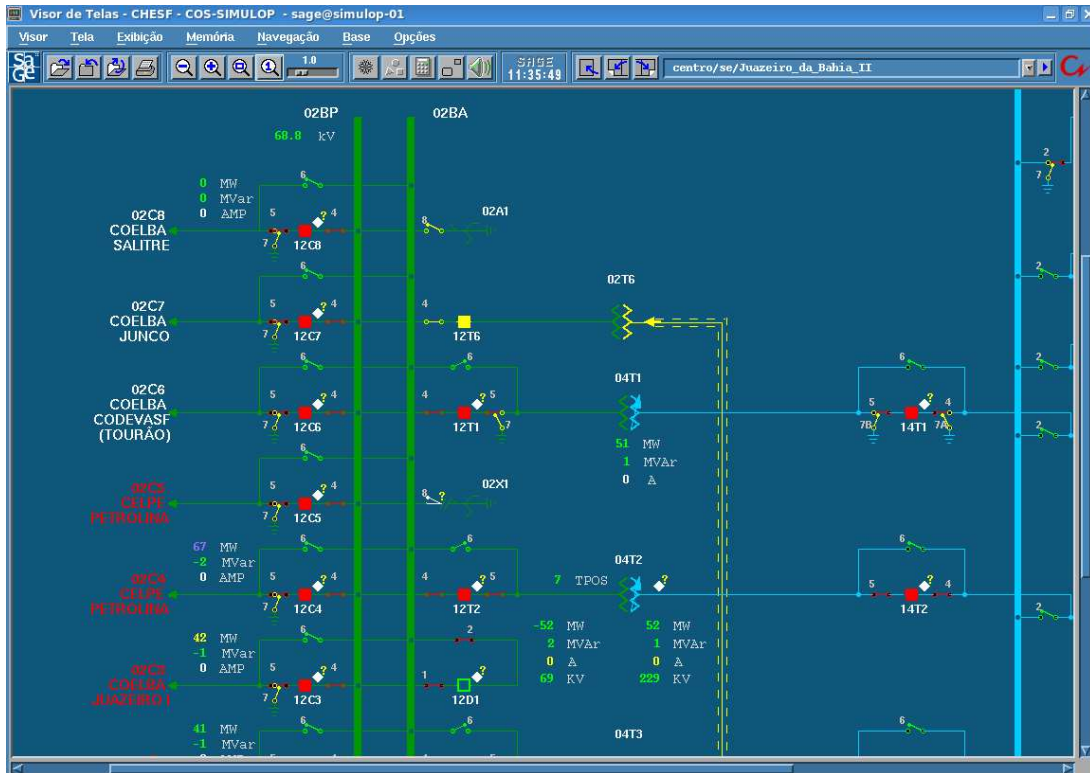


Figura 5.28: Subestação JZD antes da ativação dos eventos

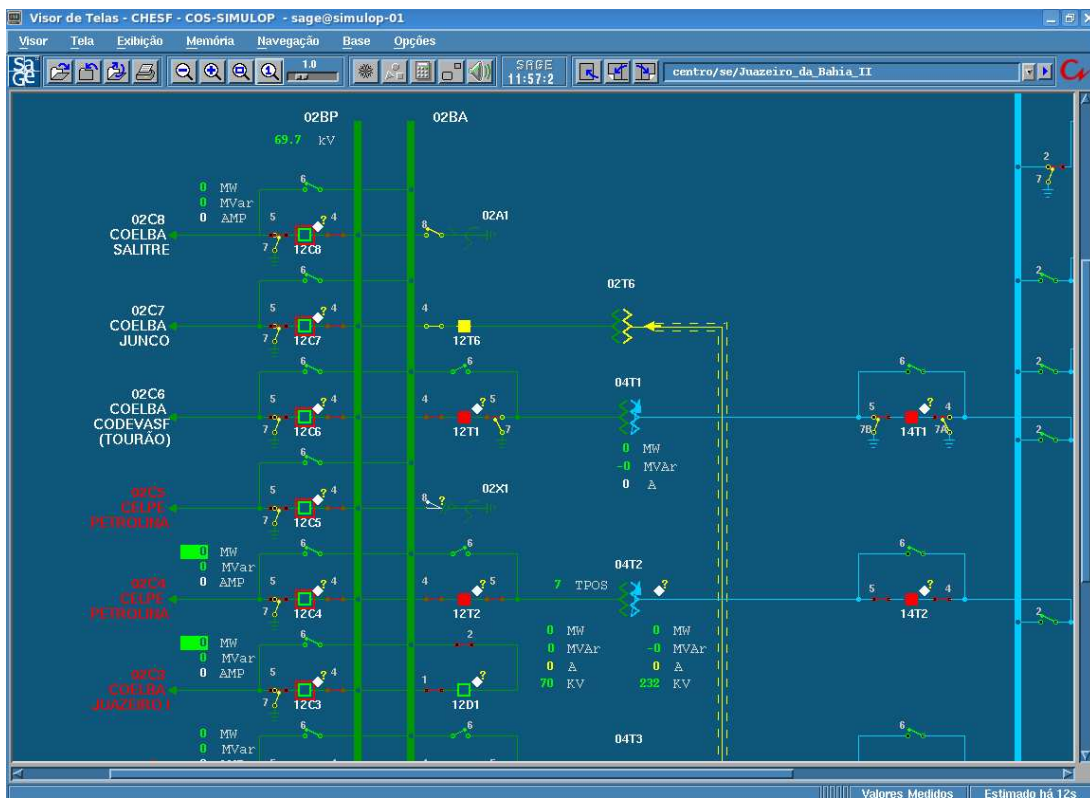


Figura 5.29: Subestação JZD depois da ativação do grupo de eventos 1



Figura 5.30: Visor de Alarmes geral indica a mudança de estado dos equipamentos na JZD

5.3 Salvando os Cenários de Validação no Banco de Dados

Para a validação da ferramenta utilizando o banco de dados, os cenários mostrados nas seções anteriores foram armazenados no mesmo.

Para salvar o primeiro cenário utilizado na RIB, na janela principal do editor, clicou-se no botão “Salvar Cenário no Banco de Dados”. Na janela ilustrada na Figura 5.31 os parâmetros do cenário foram inseridos.

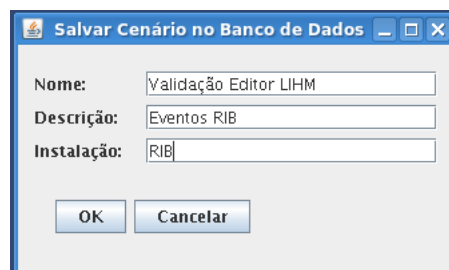


Figura 5.31: Inserindo os parâmetros para salvar o cenário de validação no BD

Prosseguiu-se da mesma forma para os outros cenários. A janela de cenários do editor (Figura 5.32) exibe a tabela com todos os cenários salvos.

ID	Nome	Descrição	Instalação	Última Modificação
8	Teste	TesteRIB	RIB	2011-05-27 16:46:21.0
9	Teste	TesteRIB	RIB	2011-05-27 16:47:09.0
11	TesteCenario	Testando Banco	RIB	2011-05-20 10:00:09.0
12	Teste CGD	Teste	CGD	2011-06-03 09:25:07.0
13	Teste SNB	SNB Padrão Antigo	SNB	2011-06-03 09:27:30.0
14	TESTE	TESTE	TESTE	2011-06-03 09:59:54.0
15	TEste	Teste	Teste	2011-06-06 10:19:34.0
17	Teste	Teste	Teste	2011-06-06 10:34:37.0
18	Teste2	Teste2	teste2	2011-06-06 10:39:18.0
20	Validação Editor LIHM	Eventos RIB	RIB	2011-07-08 17:35:56.0
24	Validação Editor LIHM 2	Eventos RIB	RIB	2011-07-11 11:27:30.0
25	Validação Editor LIHM 3	Eventos JZD - Padrão A...	JZD	2011-07-11 11:50:52.0
28	Validação Editor LIHM 4	Evento RIB com condicio...	RIB	2002-01-02 01:29:50.0

Figura 5.32: Janela de cenários do editor do LIHM

5.4 Utilização de Eventos Condicionais com o Editor de Cenários do LIHM

Na versão atual do editor não é possível criar medidas, funções, comparações e blocos lógicos para evento condicional. Uma forma de se utilizar eventos condicionais com o editor do LIHM é escolher o tipo de disparo “*TriggerType*” e colocar o identificador da condição no campo “*Time/ConditionID*”. Dessa forma as condições para que o evento seja disparado podem ser o resultado das operações lógicas criadas no próprio editor do OTS, citadas na subseção 2.1.5.

Para exemplificar será elaborado um evento condicional em um disjuntor de um transformador na subestação RIB. O evento consiste no desbloqueio de um disjuntor (12T3) a partir da relação com as suas chaves seccionadoras (32T3-4 e 32T3-5). O 12T3 está conectado em série com uma seccionadora em cada extremidade e só poderá ser desbloqueado quando as duas seccionadoras estiverem fechadas.

No editor do LIHM foi criado um evento que abre o disjuntor 12T3 com tempo relativo de 1s. Em outro grupo foi criado outro evento que bloqueia o mesmo disjuntor também com tempo relativo de 1s. Em um terceiro grupo foi criado o evento condicional. Este evento desbloqueia o disjuntor caso a condição seja atendida. Esta condição foi identificada como “COMP01”. A Figura 5.33 ilustra o arquivo texto do cenário salvo.

```

1 | Export Event Library 00/00/00 00:00:00
2 * OTS Model EXP Database eprlots/db/DBBA
3 * OTS Date 00/00/0000 TIME 00:00:00
4 * Maximum Number of Event Groups 20
5 * Maximum Number of Events Per Group 99
6 *
7 * ---- EVENT GROUP ---- 001
8 *
9 G:RIB - Abertura 12T3
10 E:R:000001:BKT :RIB :RIB:12T3:52 : :ABERTURA 12T3
11 *
12 * ---- EVENT GROUP ---- 002
13 *
14 G:RIB - Bloqueio 12T3
15 E:R:000001:BKD :RIB :RIB:12T3:52 : :BLOQUEIO 12T3
16 *
17 * ---- EVENT GROUP ---- 003
18 *
19 G:RIB - Desbloqueio de 12T3 com condicionante
20 E:C:COMP01:BKD :RIB :RIB:12T3:52 : :DESBLOQUEIO DE 12T3 CASO COMP01 SEJA VERDADE
21 *
22 * ---- CONDITIONAL EVENT BLOCKS ----
23 *

```

Figura 5.33: Cenário gerado pelo editor do LIHM

Foi importado para o editor do OTS o cenário elaborado e criado uma operação lógica também identificada por “COMP01”, ou seja, caso o resultado desta operação seja verdade, o evento condicional que tem o identificador de condição “COMP01”, será disparado quando o seu grupo for ativo no OTS. O cenário com os blocos dos eventos condicionais foi exportado pelo OTS e é ilustrado na Figura 5.34.

```

1 * Export Event Library 02/01/02 01:37:54
2 * OTS Model EXP Database eprlots/db/DBBA
3 * OTS Date 04/22/2010 TIME 12:03:09
4 * Maximum Number of Event Groups 20
5 * Maximum Number of Events Per Group 99
6 *
7 * ---- EVENT GROUP ---- 001
8 *
9 G:RIB - Abertura 12T3
10 E:R:000001:BKT :RIB :RIB:12T3:52 : :ABERTURA 12T3
11 *
12 * ---- EVENT GROUP ---- 002
13 *
14 G:RIB - Bloqueio 12T3
15 E:R:000001:BKD :RIB :RIB:12T3:52 : :BLOQUEIO 12T3
16 *
17 * ---- EVENT GROUP ---- 003
18 *
19 G:RIB - Desbloqueio de 12T3 com condicionante
20 E:C:COMP01:BKE :RIB :RIB:12T3:52 : :DESBLOQUEIO DE 12T3 CASO COMP01 SEJA VERDADE
21 *
22 * ---- CONDITIONAL EVENT BLOCKS ----
23 *
24 M:32T34 :SW:RIB :RIB:32T3-4:89 : :STATE!
25 M:32T35 :SW:RIB :RIB:32T3-5:89 : :STATE!
26 L:COMP01 : 0:C:32T34 :AND :C:32T35 : :

```

Figura 5.34: Cenário gerado pelo editor do OTS

A operação que tem identificador “COMP01” foi montada da seguinte forma: foram criadas duas medidas digitais para as seccionadoras 32T3-4 e 32T3-5 (verificam o estado das seccionadoras, se abertas ou fechadas). Estas medidas foram identificadas como “32T34” e “32T35”. Depois elas foram colocadas em um bloco lógico que realizava uma operação “AND” e este bloco lógico foi identificado como “COMP01”.

5.5 Conclusão

Este capítulo tratou do processo de validação do editor de cenários do LIHM. No início foi mostrado como o SIMULOP é iniciado e preparado para simulação. Em seguida, utilizando o editor do LIHM, foram criados eventos para uma subestação que utiliza o padrão novo para nomear os equipamentos e também para uma subestação que utiliza o padrão antigo. Os cenários foram importados para o OTS e validados e ativados com sucesso. A partir das telas do SAGE pôde-se verificar a mudança no estado dos equipamentos. Estes cenários foram salvos também no banco de dados. Por fim, foi mostrado como utilizar o editor do LIHM junto ao OTS para criar os eventos condicionais.

6

Considerações Finais

6.1 Considerações

Na elaboração deste projeto pôde-se constatar a importância do treinamento de operadores de sistemas elétricos. Nas sessões de treinamento, os operadores são capacitados para tomarem decisões complexas em diversas situações que podem vir acontecer no sistema real. O treinamento através de simuladores elimina ainda o risco de se realizar treinamentos utilizando o sistema real em funcionamento.

Antes das sessões, os tutores elaboram os cenários de treinamento ou fazem uso de uma biblioteca existente. A elaboração de um cenário de treinamento consiste em utilizar um caso base e a partir dele criar eventos que ocorrerão durante a simulação. Um caso base define o estado dos equipamentos antes da simulação. O caso base também pode ser obtido a partir do sistema real.

Os eventos que compõem um cenário podem ocorrer em tempo absoluto (acontecem em horário determinado), em tempo relativo (acontecem em tempo relativo ao início da simulação) ou podem ser eventos condicionais. Os eventos condicionais necessitam que uma condição seja satisfeita para que seja executado.

Na CHESF, as simulações são realizadas a partir de um simulador em tempo real de sistemas elétricos, o OTS, e um sistema de supervisão de sistemas elétricos, o SAGE. O OTS desprovê de alguns recursos para a edição de cenários, tornando isto uma tarefa difícil. Esta

foi a principal motivação para a criação deste trabalho.

Este trabalho teve como objetivo construir uma nova ferramenta que oferecesse mais recursos e facilitasse a edição de cenários. Com a linguagem de programação Java, foi criado o editor de cenários do LIHM. O editor do LIHM possui sua interface gráfica semelhante a do editor do OTS, de forma a tornar o novo ambiente familiar para os tutores. Ao final da edição, o cenário pode ser salvo na forma de um arquivo texto e importado para o OTS e utilizado na simulação.

Diferentemente do editor original, o editor do LIHM possibilita que os cenários sejam armazenados em um banco de dados (BD) MySQL. Isto viabiliza a organização de todos os cenários já elaborados, facilita a busca de treinamentos anteriores para serem utilizados novamente e servem para integrar os tutores que podem compartilhar os cenários elaborados.

Durante a elaboração do projeto surgiram algumas dificuldades. Uma delas foi analisar o documento gerado pelo OTS (que descreve o cenário) para ser carregado durante o treinamento. Para solucionar este problema foi avaliado o formato do arquivo, onde foi constatado que se tratava de um documento de texto sem formatação e extensão do tipo ANSI, para confirmar foi elaborado um código em Java para gerar um arquivo semelhante. Este arquivo foi importado para o OTS para verificar se era compatível com o formato aceito. Outra dificuldade foi entender o funcionamento do OTS, para solucionar este problema foi necessário um treinamento rápido com um tutor da empresa.

Outra dificuldade encontrada foi na aprendizagem da linguagem de programação Java, visto que o tempo para construção da ferramenta era limitado, e que no curso de Engenharia Elétrica foca-se pouco em linguagens de programação orientadas ao objeto.

Para validar a nova ferramenta foi feito um estudo de caso realizando simulações em duas subestações da CHESF. No editor de cenários do LIHM foram criados eventos para estas duas subestações e os mesmos foram importados para o OTS. No visor de telas do SAGE foi possível observar a mudança no estado dos equipamentos após a ativação das simulações. Os cenários utilizados na validação foram salvos no banco de dados.

Apesar de na versão atual do editor do LIHM não ser possível criar as condições para que eventos condicionais possam ser utilizados, o cenário pode ser editado no editor do LIHM e após importado, as condições podem ser construídas no editor do OTS.

Na versão atual da ferramenta pode-se constatar que ainda existe algumas limitações. A seção seguinte traz algumas propostas de trabalhos futuros para a continuidade deste trabalho.

6.2 Propostas de Trabalhos Futuros

Uma das propostas para continuidade deste trabalho é uma melhor organização da interface do editor de cenários. A interface poderá conter uma tela inicial onde o usuário poderá escolher, dentre várias opções, a operação desejada, ao invés de a tela inicial conter a tabela dos grupos de eventos, como ocorre na versão atual. Estas opções podem ser, por exemplo, abrir um cenário, criar um novo, acessar o banco de dados, etc. O editor do LIHM poderá conter uma barra de menus onde as operações possíveis possam ser distribuídas.

Outra proposta é possibilitar a criação das condicionalidades para utilização junto aos eventos condicionais no próprio editor do LIHM. Para isto, devem ser inseridas mais quatro janelas (*JFrame*) onde cada janela corresponde a uma das funcionalidades (medidas, funções, comparações e blocos lógicos). Cada janela deverá conter uma tabela onde estão preenchidos os dados da funcionalidade e para editar os dados destas tabelas devem existir mais quatro janelas para que os dados sejam inseridos (uma para cada funcionalidade), de maneira análoga à edição de eventos feita no editor atual, onde os dados dos eventos são inseridos em uma janela e depois os dados são preenchidos na linha da tabela correspondente. Para armazenar estes dados, devem ser inseridas também quatro matrizes de Strings. Na barra de menus pode ser inserido um menu para estas funcionalidades, onde escolhe-se uma das quatro opções para edição.

Para que as condicionalidades possam ser escritas no arquivo texto de cenário, no método “*escreveArquivo*” da classe “*ArquivoCenario*” do pacote “*editorinterface*”, o código deve ser completado. O código será semelhante ao da escrita dos eventos. No banco de dados devem ser criadas novas tabelas, uma para cada funcionalidade. Estas tabelas deverão ser vinculadas a um “*id*” da tabela dos cenários.

Outra limitação que pode ser observada é a mistura de idiomas, pois a ferramenta original é toda em inglês e durante a elaboração deste projeto tentou-se seguir o padrão que já era utilizado para facilitar o uso, deixando as palavras chaves ainda em inglês, mas pode-se

conversar com usuários do sistema para traduzir toda a ferramenta. Pois as funcionalidades que foram desenvolvidas estão todas em português.

A

Criação do Banco de Dados “cenariosSage”

No LIHM, todos os sistemas citados na subseção 3.1.2 encontravam-se instalados em um servidor. O servidor possui endereço IP 150.165.61.62 e o *phpMyAdmin* pode ser acessado através da página <http://150.165.61.62/phpMyAdmin/>. A tela inicial do *phpMyAdmin* ilustrada na Figura A.1, solicita o nome de usuário e senha para acesso ao sistema.



A imagem mostra a interface de login do phpMyAdmin. No topo, há o logotipo do phpMyAdmin com o texto "Bem vindo ao phpMyAdmin". Abaixo, há um menu suspenso para "Linguagem - Language" com "Português - Brazilian portuguese" selecionado. Na seção "Autenticação", há campos para "Usuário:" (contendo "cenariosSage") e "Senha:". Um botão "Executar" está localizado na parte inferior direita da seção de autenticação.

Figura A.1: Tela de *login* do *phpMyAdmin*

Acessando o *phpMyAdmin* com a conta de administrador foi criado um banco de dados denominado “cenariosSage”. Associado a este banco foi criado um usuário também com nome “cenariosSage” e com senha de acesso “cenarios!123”.

Dentro do banco de dados “cenariosSage” foram criadas três tabelas. A primeira com nome “cenarios”, outra com nome “gruposeventos” e a última com nome “eventos”. Cada linha destas tabelas possui uma chave primária que aqui foi chamada de “id”. A medida que linhas

são adicionadas nas tabelas, estas chaves primárias são incrementadas automaticamente. Caso alguma linha seja removida, a chave primária desta linha não será mais utilizada por outra linha adicionada posteriormente. Na Figura A.2 é ilustrada a visualização da estrutura do banco “cenariosSage” no *phpMyAdmin*.

Tabela	Ação	Registros ¹	Tipo	Collation	Tamanho	Sobrecarga
cenarios		14	InnoDB	utf8_general_ci	16.0 KB	-
eventos		1,104	InnoDB	utf8_general_ci	224.0 KB	-
gruposseventos		89	InnoDB	utf8_general_ci	16.0 KB	-
3 tabela(s)	Soma	1,207	InnoDB	utf8_general_ci	256.0 KB	0 Bytes

Figura A.2: Estrutura do banco de dados “cenariosSage”

Estas tabelas estão interligadas da seguinte maneira: uma linha da tabela “eventos” possui os dados de um evento que pertence a um grupo, e este grupo está armazenado em uma linha da tabela “gruposseventos”, que por sua vez pertence a um cenário que está armazenado em uma linha da tabela “cenarios”. Estas vinculações são feitas a partir das chaves primárias, ou seja, um evento está vinculado à chave primária de um grupo e este está vinculado à chave primária de um cenário. Na Figura A.3 é ilustrado o MER (Modelo Entidade-Relacionamento) do banco de dados “cenariosSage”, construído no *software DBDesigner* (download em <http://www.fabforce.net/downloads.php>). As Figuras A.4, A.5 e A.6, ilustram as visualizações no *phpMyAdmin* das estruturas das tabelas “cenarios”, “gruposseventos” e “eventos”, respectivamente.

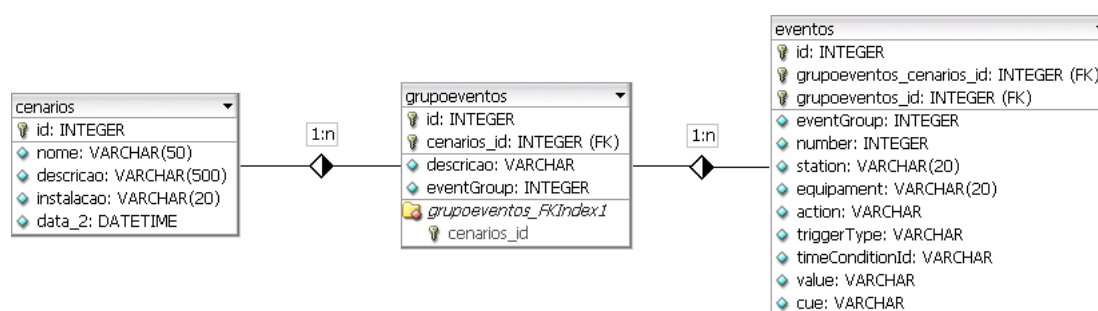


Figura A.3: MER do banco “cenariosSage”

Servidor: localhost ▶ Banco de Dados: cenariosSage ▶ Tabela: cenarios

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Operações Limpar Eliminar

Campo	Tipo	Collation	Atributos	Nulo	Padrão	Extra	Ação
<input type="checkbox"/> id	int(11)			Não	None	auto_increment	[Ícone]
<input type="checkbox"/> nome	varchar(50)	utf8_general_ci		Não	None		[Ícone]
<input type="checkbox"/> descricao	varchar(500)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> instalacao	varchar(20)	utf8_general_ci		Não	None		[Ícone]
<input type="checkbox"/> data	datetime			Não	None		[Ícone]

Figura A.4: Estrutura da tabela “cenarios”

Servidor: localhost ▶ Banco de Dados: cenariosSage ▶ Tabela: gruposeventos

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Operações Limpar Eliminar

Campo	Tipo	Collation	Atributos	Nulo	Padrão	Extra	Ação
<input type="checkbox"/> id	int(11)			Não	None	auto_increment	[Ícone]
<input type="checkbox"/> descricao	varchar(500)	utf8_general_ci		Não	None		[Ícone]
<input type="checkbox"/> idCenario	int(11)			Não	None		[Ícone]
<input type="checkbox"/> eventGroup	int(11)			Não	None		[Ícone]

Figura A.5: Estrutura da tabela “gruposeventos”

Servidor: localhost ▶ Banco de Dados: cenariosSage ▶ Tabela: eventos

Visualizar Estrutura SQL Procurar Inserir Exportar Importar Operações Limpar Eliminar

Campo	Tipo	Collation	Atributos	Nulo	Padrão	Extra	Ação
<input type="checkbox"/> id	int(11)			Não	None	auto_increment	[Ícone]
<input type="checkbox"/> eventGroup	int(11)			Não	None		[Ícone]
<input type="checkbox"/> number	int(11)			Não	None		[Ícone]
<input type="checkbox"/> station	varchar(20)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> equipment	varchar(20)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> action	varchar(200)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> triggerType	varchar(200)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> timeConditionId	varchar(20)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> value	varchar(200)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> cue	varchar(500)	utf8_general_ci		Sim	NULL		[Ícone]
<input type="checkbox"/> idGruposEventos	int(11)			Não	None		[Ícone]
<input type="checkbox"/> idCenario	int(11)			Não	None		[Ícone]

Figura A.6: Estrutura da tabela “eventos”

Na Figura A.7 é ilustrada a tabela “cenarios”. Já na Figura A.8 é ilustrada a tabela “gruposdeeventos” com os grupos de eventos pertencentes ao cenário com “id” = 20, que é o primeiro cenário utilizado na validação do editor do LIHM. Na Figura A.9 é ilustrado um trecho da tabela com os eventos também pertencentes ao cenário com “id” = 20. As consultas SQL inseridas no *phpMyAdmin* para a obter estas tabelas foram as seguintes: “SELECT * FROM gruposdeeventos WHERE idCenario = 20” e “SELECT * FROM eventos WHERE idCenario = 20”.

	id	nome	descricao	instalacao	data
<input type="checkbox"/>	5	cenario1	grupo1	RIB	2011-05-26 08:59:03
<input type="checkbox"/>	6	cenario1	grupo1	RIB	2011-05-26 09:50:25
<input type="checkbox"/>	8	Teste	TesteRIB	RIB	2011-05-27 16:46:21
<input type="checkbox"/>	9	Teste	TesteRIB	RIB	2011-05-27 16:47:09
<input type="checkbox"/>	11	TesteCenario	Testando Banco	RIB	2011-05-30 10:00:09
<input type="checkbox"/>	12	Teste CGD	Teste	CGD	2011-06-03 09:25:07
<input type="checkbox"/>	13	Teste SNB	SNB Padrão Antigo	SNB	2011-06-03 09:27:30
<input type="checkbox"/>	14	TESTE	TESTE	TESTE	2011-06-03 09:59:54
<input type="checkbox"/>	15	TEste	Teste	Teste	2011-06-06 10:19:34
<input type="checkbox"/>	17	Teste	Teste	Teste	2011-06-06 10:34:37
<input type="checkbox"/>	18	Teste2	Teste2	teste2	2011-06-06 10:39:18
<input type="checkbox"/>	20	Validação Editor LIHM	Eventos RIB	RIB	2011-07-08 17:35:56
<input type="checkbox"/>	24	Validação Editor LIHM 2	Eventos RIB	RIB	2011-07-11 11:27:30
<input type="checkbox"/>	25	Validação Editor LIMH 3	Eventos JZD - Padrão Antigo	JZD	2011-07-11 11:50:52

Figura A.7: Tabela “cenarios”

	id	descricao	idCenario	eventGroup
<input type="checkbox"/>	84	RIB - Abertura dos Disjuntores (Trip)	20	1
<input type="checkbox"/>	85	RIB - Bloqueio dos Disjuntores (Disable)	20	2
<input type="checkbox"/>	86	RIB - Desbloqueio dos Disjuntores (Enable)	20	3
<input type="checkbox"/>	87	RIB - Fechamento dos Disjuntores (Close)	20	4

Figura A.8: Tabela “gruposdeeventos”

	id	eventGroup	number	station	equipment	action	triggerType	timeConditionId	value	cue	idGruposEventos	idCenario
<input type="checkbox"/>	1088	1	1	RIB	RIB:12T2:52	BKT	R	000001		ABERTURA DO DISJUNTOR 12T2	84	20
<input type="checkbox"/>	1089	1	2	RIB	RIB:12T3:52	BKT	R	000001		ABERTURA DO DISJUNTOR 12T3	84	20
<input type="checkbox"/>	1090	1	3	RIB	RIB:12L1:52	BKT	R	000001		ABERTURA DO DISJUNTOR 12L1	84	20
<input type="checkbox"/>	1091	1	4	RIB	RIB:12L2:52	BKT	R	000001		ABERTURA DO DISJUNTOR 12L2	84	20
<input type="checkbox"/>	1092	1	5	RIB	RIB:12L3:52	BKT	R	000001		ABERTURA DO DISJUNTOR 12L3	84	20

Figura A.9: Tabela “eventos”

B

Integração do Editor de Cenários com o Banco de Dados

Através do pacote SQL do Java e da API JDBC (*Java Database Connectivity*) é possível fazer a integração do banco de dados com o programa Java.

Um objeto do tipo *Connection* (pacote SQL) gerencia a conexão entre o programa Java e o banco de dados. Este objeto é inicializado a partir do resultado de uma chamada para o método *static getConnection* da classe *DriverManager* (pacote SQL), que tenta conectar-se ao banco de dados especificado pela sua URL (*Uniform Resource Locator*). Esta URL possui todas as informações necessárias para se efetuar a conexão com o servidor, tais como: o endereço IP do servidor, a porta onde o servidor recebe conexões (3306 é a porta padrão para o MySQL), o nome do banco de dados e o nome de usuário e senha (aqui, o IP do servidor é o 150.165.61.62 e o nome do banco de dados e de usuário é “cenariosSage”). O método *getConnection* recebe três argumentos, uma *String* que especifica a URL do banco de dados, uma *String* que especifica o nome de usuário e outra *String* que especifica a senha (Deitel e Deitel, 2005).

O método *createStatement* do objeto *Connection* é chamado para obter um objeto que implementa a interface *Statement* (pacote SQL). O método *executeUpdate* do objeto *Statement* é utilizado para submeter SQL para o banco de dados. O método *executeUpdate* também do objeto *Statement* é utilizado para realizar consultas no banco de dados. Este método retorna um objeto que implementa a interface *ResultSet* e contém o resultado da

consulta (Deitel e Deitel, 2005).

As instruções SQL mais utilizadas neste projeto foram a `SELECT`, que realiza uma consulta nas tabelas; `INSERT`, que insere uma linha em uma tabela; `UPDATE`, que modifica os dados em uma tabela e `DELETE`, que remove linhas de uma tabela. Estas instruções possuem as seguintes sintaxes:

- **SELECT:** `SELECT nomeDaColuna1, nomeDaColuna2, ... FROM nomeDaTabela WHERE criterios`
- **INSERT:** `INSERT (nomeDaColuna1, nomeDaColuna2, ..., nomeDaColunaN) VALUES (valor1, valor2, valorN)`
- **UPDATE:** `UPDATE nomeDaTabela SET nomeDaColuna1 = valor1, nomeDaColuna2 = valor2, ..., nomeDaColunaN = valorN WHERE criterios`
- **DELETE:** `DELETE FROM nomeDaTabela WHERE criterios`

Os nomes das colunas na instrução `SELECT` podem ser substituídos por um “*” para que a consulta retorne todas as colunas. Mais detalhes em (MyS, 1997-2006). No programa Java estas instruções SQL são colocadas em uma *String*.

Referências Bibliográficas

(1997-2006) Manual de Referência do MYSQL 4.1, MYSQL AB.

Bezerra, H.; Barroso, G. C.; Sampaio, R. F.; Leão, R. P. e Soares, J. M. (2007) Sistema simulador para treinamento de proteção e operação de sistemas elétricos, ICECE 2007 Conferência Internacional em Educação em Engenharia e Computação, **Anais da Conferência Internacional em Educação em Engenharia e Computação, São Paulo - Brasil.**

Cay Horstmann, W. (2008) Big Java, John Wiley and Sons, 3^o edic..

Deitel, H. M. e Deitel, P. J. (2005) JAVA Como Programar, Prentice-Hall, 6^o edic..

Hirsch, P. (2004) User Guide for PowerSimulator with EPRI OTS Version 3.6, EPRI, Palo Alto - California.

Júnior, R.; Macedo, E. e. A. e Sampaio, n. A. (2010) Desenvolvimento de método para aperfeiçoamento de operadores através da imersão em ambiente de simulação, XI EDAO - Encontro para Debates de Assuntos de Operação.

Leite, J.; Rossi, G.; Balaguer, F.; Maiorana, V.; Kaplan, G.; Hadad, G. e Oliveros, A. (1997) Enhancing a requirements baseline with scenarios, Requirements Engineering, **Proceedings of the Third IEEE International Symposium on, pp. 44 - 53, Annapolis, MD - USA.**

Muniz, R. B. (2004) Funcionalidades do Simulador de Redes Elétricas EPRI-OTS e Sua Utilização Para Treinamento de Operadores na CHESF, CHESF/DOS/DOMO.

Muniz, R. B. e Miranda, A. R. (2009) Manual Básico de Utilização do OTS, CHESF/DOS/DOMO/COOS.

Netto, A. V. S. e Vieira, M. F. Q. (2010) Virtual reality training environment a proposed architecture. In: VECIMS Virtual Environments, Human-Computer Interfaces, and

- Measurement Systems, Taranto, Itália. Proceedings of the VECIMS, 2010. p. 77-81.
- de Oliveira, J.; Lima, L.; Pereira, L.; Sollero, R.; Leite, C.; Muniz, R.; Costa, C.; Cavalcante, M.; Carmo, U.; de Araujo, A. e Meireles, L. (2002) Treinamento e Certificação de Operadores no Sistema SAGE Empregando o Simulador EPRI/OTS, Centro de Pesquisas de Energia Elétrica - CEPEL.
- Silva, V. N. A.; Linden, R.; Ribeiro, G. F.; Pereira, M. F. L.; Lannes, R. S. e Standke, C. R. (2009) Simuladores para treinamento de operadores de sistemas e de instalações do setor elétrico, XIII ERIAC Décimo Tercer Encuentro Regional Iberoamericano de Cigré.