



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Ana Carolina Vieira Dal Pont

**Desenvolvimento do Simulador de Equipamentos e Auxiliares para o Sistema  
de Controle Centralizado da Linha 5 do Metrô de São Paulo**

Florianópolis  
2020

Ana Carolina Vieira Dal Pont

**Desenvolvimento do Simulador de Equipamentos e Auxiliares para o Sistema de Controle Centralizado da Linha 5 do Metrô de São Paulo**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Max Hering de Queiroz, Dr.  
Supervisor: Thiago Israel Ramalho Bacic, Eng.

Florianópolis  
2020

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Dal Pont , Ana Carolina Vieira  
Desenvolvimento do Simulador de Equipamentos e  
Auxiliares para o Sistema de Controle Centralizado da  
Linha 5 do metrô de São Paulo / Ana Carolina Vieira Dal  
Pont ; orientador, Max Hering de Queiroz, 2020.  
73 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Engenharia de Controle e Automação,  
Florianópolis, 2020.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Automação  
Industrial. 3. Sistemas SCADA. 4. Software Elipse E3. I.  
Hering de Queiroz, Max. II. Universidade Federal de Santa  
Catarina. Graduação em Engenharia de Controle e Automação.  
III. Título.

Ana Carolina Vieira Dal Pont

**Desenvolvimento do Simulador de Equipamentos e Auxiliares para o Sistema de Controle Centralizado da Linha 5 do Metrô de São Paulo**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 14 de dezembro de 2020.

---

Prof. Hector Bessa Silveira, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Max Hering de Queiroz, Dr.  
Orientador  
UFSC/CTC/DAS

---

Thiago Israel Ramalho Bacic, Eng.  
Supervisor  
Radix Engenharia e Desenvolvimento de Software

---

Prof. Ademar Gonçalves da Costa Junior, Dr.  
Avaliador  
Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)

---

Prof. Marcelo de Lellis Costa de Oliveira, Dr.  
Presidente da Banca  
UFSC/CTC/DAS

Este trabalho é dedicado à minha família e aos meus  
queridos amigos.

## **AGRADECIMENTOS**

Aos meus pais, Vanderlei e Marlete, e ao meu irmão Renato, agradeço pelo suporte, dedicação e, principalmente, por sempre acreditarem no meu potencial. Sem vocês esta jornada não teria sido possível.

Aos meus amigos, com os quais divido inúmeras boas memórias e experiências incríveis, o meu muito obrigada.

Aos meus colegas da Radix por confiarem no meu trabalho e me acolheram tão bem em São Paulo.

Por fim, agradeço à Universidade Federal de Santa Catarina e ao corpo docente do Departamento de Automação e Sistemas por todos os ensinamentos e aprendizados.

*“Recognize and embrace your uniqueness. Being a woman, on a team of all men, means that you are going to have a unique voice. It’s important to embrace that.” (Erin Teague)*

## RESUMO

Desde a segunda metade do século XVIII, o mundo passou por grandes transformações que possibilitaram a descoberta de novas tecnologias. A segunda revolução industrial possibilitou o aprimoramento de técnicas, o surgimento das máquinas e a introdução de novos meios de produção. Sistemas SCADA são capazes de se comunicar com os dispositivos de campo, coletando, monitorando e analisando as informações de um processo produtivo. De maneira geral, quanto mais extenso for o processo que se deseja supervisionar, maior será a complexidade do sistema supervisor a ser desenvolvido. Neste âmbito, o objetivo principal deste documento é apresentar os passos realizados para finalizar o desenvolvimento do Simulador de Equipamentos e Auxiliares (SIMEA) desenvolvido para o Sistema de Controle Centralizado (SCC) da Linha 5 do metrô de São Paulo. O SIMEA foi desenvolvido utilizando a plataforma Eclipse E3 e tem como objetivo principal simular as concentradoras de dados existentes em campo, sendo o ambiente utilizado para simulação, treinamento e testes de estratégias operacionais do SCC. A metodologia de projeto é apresentada, bem como os passos realizados para instanciar as estações da Linha 5 utilizando o Eclipse E3, juntamente com o código elaborado para automatizar o processo de instanciação e as lógicas desenvolvidas para garantir o correto funcionamento do SIMEA. Os resultados obtidos com o Teste de Aceitação em Fábrica do simulador indicam que o desempenho do sistema foi satisfatório, uma vez que ele é capaz de replicar os sinais provenientes do campo de forma idêntica ao modelo real, respeitando as lógicas básicas de funcionamento do SCC.

**Palavras-chave:** Automação Industrial. Sistemas SCADA. Eclipse E3.



## ABSTRACT

Since the second half of the 18th century, the world has undergone major transformations that made possible the discovery of new technologies. The second industrial revolution made possible the improvement of techniques, the appearance of machines, and the introduction of new ways of production. SCADA systems are able to communicate with field devices, collecting, monitoring, and analyzing information from a production process. In general, the bigger the process you want to supervise, the greater the complexity of the supervisory system to be developed. In this context, the main objective of this document is to present the steps taken to finalize the development of the Equipment and Auxiliary Simulator (SIMEA) developed for the Centralized Control System (SCC) of Line 5 of the São Paulo subway. SIMEA was developed using the Elipse E3 platform and its main objective is to simulate the data concentrators existent on field, being used for simulation, training, and testing of SCC operational strategies. The project methodology is presented, as well as the steps that are necessary to instantiate Line 5 stations using Elipse E3, along with the code designed to automate the instantiation process and the logic developed to ensure the correct functioning of SIMEA. The results obtained with the Factory Acceptance Test of the simulator show that the system performance was satisfactory since it is able to replicate the signals coming from the field in the same way as the real model, respecting the basic logic of SCC operation.

**Keywords:** Industrial Automation. SCADA Systems. Elipse E3.

## LISTA DE FIGURAS

Figura 1 – Pirâmide de automação industrial. . . . .	16
Figura 2 – Exemplo de Arquitetura do Elipse E3. . . . .	22
Figura 3 – Super domínio no Elipse E3. . . . .	24
Figura 4 – Aba de associações no Elipse E3. . . . .	26
Figura 5 – Exemplo de associação simples no Elipse E3. . . . .	26
Figura 6 – Exemplo de associação reversa no Elipse E3. . . . .	27
Figura 7 – Exemplo de associação por tabela no Elipse E3. . . . .	28
Figura 8 – Mapa da Linha 5 do metrô de São Paulo. . . . .	28
Figura 9 – Tela inicial do SCC. . . . .	30
Figura 10 – Concentradoras de dados. . . . .	33
Figura 11 – Configuração do domínio remoto no Elipse E3. . . . .	34
Figura 12 – Fluxo de dados no SIMEA. . . . .	35
Figura 13 – Tela de edição dos cenários. . . . .	37
Figura 14 – Tela de edição das sequências de eventos. . . . .	38
Figura 15 – Tela de edição das simulações. . . . .	39
Figura 16 – Tela de gerenciamento da simulação. . . . .	40
Figura 17 – Passos necessários para instanciar uma estação no SIMEA. . . . .	43
Figura 18 – Veneziana SVTE modelada no SIMEA. . . . .	45
Figura 19 – Instâncias da Veneziana SVTE na estação Largo Treze. . . . .	46
Figura 20 – Exemplo de associações entre o objeto de dados e o <i>driver</i> de comunicação. . . . .	47
Figura 21 – Objeto de dados do Sistema de Controle Elétrico da estação Santa Cruz. . . . .	48
Figura 22 – Associações criadas de forma automatizada para o equipamento Disjuntor Tramo Comando F11 da estação Santa Cruz. . . . .	49
Figura 23 – Mensagem indicando que a verificação foi efetuada com sucesso. . . . .	49
Figura 24 – Estado padrão do equipamento Contator Equalização. . . . .	51
Figura 25 – Exemplo de associação para pontos de alarmes que são atuados quando recebem o valor analógico 1. . . . .	52
Figura 26 – Exemplo de associação para pontos de alarmes que são atuados quando recebem o valor analógico 0. . . . .	53
Figura 27 – Exemplo de <i>script</i> do tipo <i>On Property Change</i> que é utilizado em propriedades de alarmes que são atuados quando recebem o valor analógico 0. . . . .	53
Figura 28 – <i>PlaceHolder</i> do SIMEA. . . . .	54
Figura 29 – Fluxo de dados entre o SCC e as concentradoras. . . . .	56
Figura 30 – Exemplo de uma tela modal de comando de um equipamento. . . . .	57

Figura 31 – Fluxo de dados entre o SCC e o SIMEA. . . . .	57
Figura 32 – <i>PlaceHolder</i> com as propriedades do equipamento Botoeira Emergência Geral. . . . .	58
Figura 33 – Registro de eventos do SCC após o comando Manual ser enviado ao equipamento. . . . .	59
Figura 34 – Associações no objeto de dados do equipamento Botoeira Emergência Geral no SCC. . . . .	60
Figura 35 – Associações no objeto de dados do equipamento Botoeira Emergência Geral no SIMEA. . . . .	60
Figura 36 – Instâncias do <i>XObject Delay</i> na classe da Botoeira Emergência Geral. . . . .	61
Figura 37 – Associações no objeto de dados do objeto <i>Delay</i> para o comando manual da Botoeira de Emergência Geral. . . . .	61
Figura 38 – Tela modal de comando da Botoeira de Emergência Geral no SCC. . . . .	62
Figura 39 – Fluxo dos comandos no SIMEA. . . . .	63
Figura 40 – Exemplos de casos de testes relacionados à navegação entre telas. . . . .	65
Figura 41 – Arquitetura de rede para o TAF. . . . .	66
Figura 42 – Trello do SIMEA. . . . .	68

## LISTA DE QUADROS

Quadro 1 – Situação de cada uma das estações da Linha 5 no SIMEA. . . . .	43
Quadro 2 – Equipamentos sem <i>XObject</i> associado no SIMEA. . . . .	44
Quadro 3 – Equipamentos comandáveis. . . . .	56
Quadro 4 – Exemplos de pendências geradas ao longo do TAF. . . . .	67

## LISTA DE TABELAS

Tabela 1 – Resultados quantitativos da primeira rodada de teste. . . . .	66
--	----

## LISTA DE ABREVIATURAS E SIGLAS

ABV	Estação Alto da Boa Vista
APN	Estação Adolfo Pinheiro
BGA	Estação Borba Gato
BRK	Estação Brooklin
CKB	Estação Chácara Klabin
CLP	Controlador Lógico Programável
CPB	Estação Campo Belo
CPL	Estação Campo Limpo
CPR	Estação Capão Redondo
ECT	Estação Eucaliptos
GGR	Estação Giovani Gronchi
HSP	Estação Hospital São Paulo
LTR	Estação Largo Treze
MOE	Estação Moema
SCADA	Sistema de Supervisão e Aquisição de Dados
SCC	Sistema de Controle Centralizado
SCE	Sistema de Controle Elétrico
SCL	Sistema de Controle Local
SCT	Sistema de Controle e Regulação de Trens
SCZ	Estação Santa Cruz
SEA	Sistema de Equipamentos e Auxiliares
SER	Estação AACD Servidor
SFP	Sistema de Fluxo de Passageiros
SIMEA	Simulador de Equipamentos e Auxiliares
SIMTR	Simulador de Trens e Trilhos
STA	Estação Santo Amaro
TAF	Teste de Aceitação em Fábrica
VBE	Estação Vila das Belezas

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	JUSTIFICATIVA	17
1.2	OBJETIVOS	17
1.3	RADIX ENGENHARIA E DESENVOLVIMENTO DE SOFTWARE	18
1.4	ESTRUTURA DO DOCUMENTO	19
<b>2</b>	<b>SISTEMA SUPERVISÓRIO DA LINHA 5 DO METRÔ DE SÃO PAULO</b>	<b>20</b>
2.1	SISTEMAS SCADA	20
2.2	SOFTWARE ELIPSE E3	21
<b>2.2.1</b>	<b>Domínios</b>	<b>23</b>
<b>2.2.2</b>	<b>Bibliotecas de Símbolos e Usuários</b>	<b>24</b>
<b>2.2.3</b>	<b>Associações</b>	<b>25</b>
2.3	SISTEMA DE CONTROLE CENTRALIZADO - SCC	28
<b>3</b>	<b>SIMULADOR DE EQUIPAMENTOS E AUXILIARES - SIMEA</b>	<b>32</b>
3.1	CENÁRIOS	35
3.2	EVENTOS	37
3.3	SIMULAÇÃO	38
<b>4</b>	<b>DESENVOLVIMENTO E IMPLEMENTAÇÃO DO SIMEA</b>	<b>41</b>
4.1	METODOLOGIA DE DESENVOLVIMENTO DO PROJETO	41
4.2	MODELAGEM DOS EQUIPAMENTOS E INSTANCIAMENTO DAS ESTACIONES	42
4.3	LÓGICAS DESENVOLVIDAS PARA OS EQUIPAMENTOS	50
<b>4.3.1</b>	<b>Estados Padrões dos Equipamentos</b>	<b>50</b>
<b>4.3.2</b>	<b>Lógica dos alarmes</b>	<b>51</b>
<b>4.3.3</b>	<b>Comando dos Equipamentos</b>	<b>55</b>
<b>5</b>	<b>TESTE DE ACEITAÇÃO EM FÁBRICA E RESULTADOS</b>	<b>64</b>
5.1	TESTE DE ACEITAÇÃO EM FÁBRICA (TAF)	64
5.2	RESULTADOS	68
<b>6</b>	<b>CONCLUSÃO</b>	<b>69</b>
	<b>REFERÊNCIAS</b>	<b>71</b>

## 1 INTRODUÇÃO

O contínuo desenvolvimento do conhecimento humano nos últimos anos resultou em um crescente desenvolvimento científico e tecnológico. Com o aumento da competição no mercado, torna-se obrigatório reduzir, continuamente, os custos e melhorar o processo produtivo sempre que possível. O sucesso depende muito da habilidade para acessar, entender e interpretar o grande volume de informações geradas pela operação do processo. Por isso, surge nas indústrias a necessidade de agregar tecnologia da informação aos processos de automação industrial, abrangendo também a automação do negócio (MORAES C.C.; CASTRUCCI, 2006).

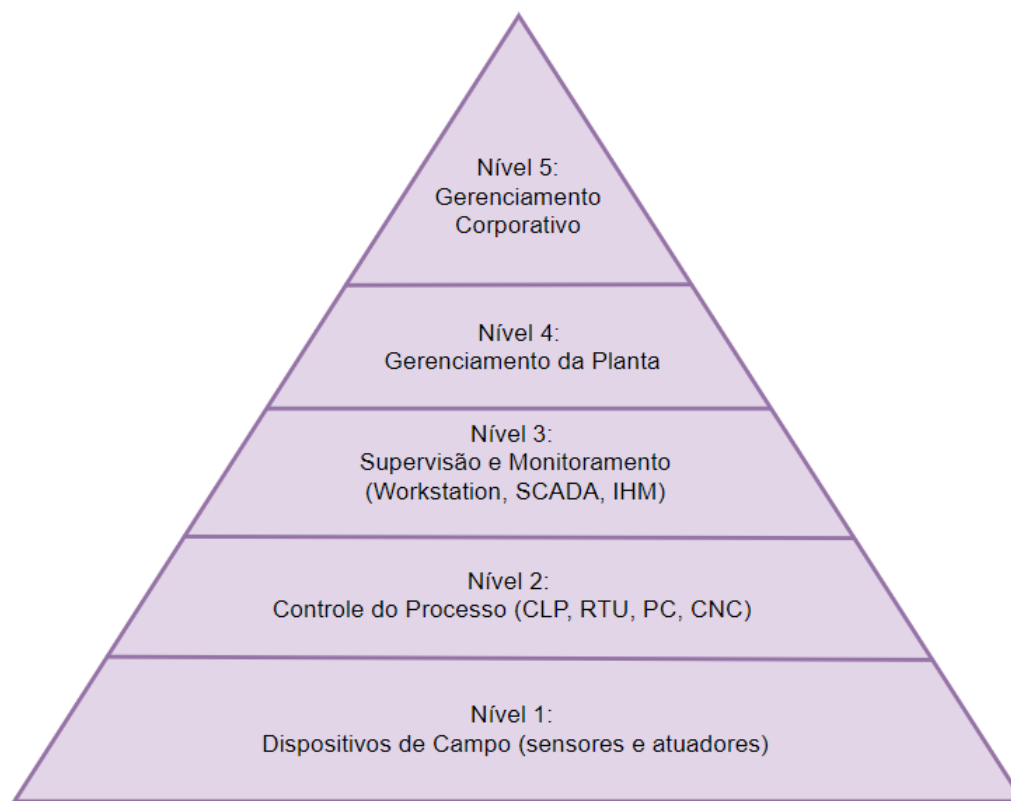
A introdução das primeiras formas de automação deu-se nas indústrias de processo, por meio do desenvolvimento de equipamentos de controle e medição elétrica. O conceito de automação é constantemente confundido com o de automatização. Automatizar é realizar movimentos repetitivos e mecânicos, implicando em uma ação cega e sem correção. A automação, por sua vez, possui um conjunto de técnicas por meio das quais se constroem sistemas ativos, capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre qual atuam (ROSÁRIO, 2009).

Desta forma, sistemas de supervisão e aquisição de dados possuem um papel muito importante. Sistemas SCADA (do inglês *Supervisory Control and Data Acquisition*) são responsáveis pela coleta de dados em tempo real em uma grande variedade de plantas, disponibilizando esses dados para operadores localizados nos mais diferentes locais a todo instante de tempo. São considerados sistemas críticos, pois são componentes vitais de inúmeras infraestruturas como, por exemplo, controle de refinarias, sistemas de transporte e instalações químicas (ABBAS H.; SHAHEEN, 2015).

Neste contexto, a pirâmide de automação industrial ilustrada na Figura 1, tem como objetivo principal estruturar, de forma hierárquica, as principais etapas de um processo de automação industrial. A pirâmide de automação é composta por cinco níveis e sua arquitetura permite demonstrar a interdependência existente entre cada um deles. Conforme pode ser visualizado na Figura 1, à medida que o topo vai sendo alcançado, os dados coletados sofrem um melhor tratamento, assim, no decorrer da pirâmide, as informações obtidas terão maior qualidade e significado.



Figura 1 – Pirâmide de automação industrial.



Fonte – Adaptado de (MORAES C.C.; CASTRUCCI, 2006).

O nível 1 é popularmente conhecido como o "chão de fábrica", visto que é nele que estão localizados as máquinas e dispositivos de campo, como sensores e atuadores. O controle dos equipamentos de automação do nível 1 é de responsabilidade do nível 2, uma vez que é ali que se encontram os controladores digitais, dinâmicos e lógicos. No nível 3 estão localizados os sistemas SCADA, os quais são responsáveis pelo controle do processo produtivo da planta, armazenando as informações repassadas pelos equipamentos dos níveis 1 e 2 em banco de dados e, posteriormente, repassando-as para os níveis administrativos. Além do mais, o nível 3 também é responsável por fazer a interface com o operador do sistema, permitindo que os dados sejam apresentados em tempo real. Por fim, o nível 4 é responsável pela programação e planejamento da produção, enquanto o nível 5 é responsável pela tomada de decisão e gerenciamento de todo o sistema.

Como mencionado anteriormente, os sistemas supervisórios possibilitam que as informações de um processo produtivo ou instalação física sejam monitoradas e rastreadas. Para tal finalidade, o sistema deve ser capaz de se comunicar com os dispositivos de campo a fim de coletar as informações julgadas necessárias e, em seguida, manipulá-las, analisá-las, armazená-las e, posteriormente, apresentá-las ao

usuário.

De modo geral, quanto mais extenso for o processo que se deseja controlar e monitorar, maior será a complexidade do sistema supervisorio a ser desenvolvido. Conseqüentemente, maior será a quantidade de *tags* de comunicação e telas que terão de ser criadas, bem como a quantidade de dados históricos que serão armazenados. Logo, por se tratar de um sistema considerado crítico, é desejável que o operador que irá manuseá-lo possua conhecimentos suficientes para operá-lo com tranquilidade e segurança.

Portanto, o projeto de fim de curso descrito neste documento tem como objetivo principal apresentar o Simulador de Equipamentos e Auxiliares (SIMEA) desenvolvido para a simulação, treinamento e testes de estratégias operacionais do sistema supervisorio da Linha 5 do metrô de São Paulo.

No contexto apresentado, os conhecimentos adquiridos nas disciplinas de informática industrial, redes de computadores para automação, programação e banco de dados são relevantes.

## 1.1 JUSTIFICATIVA

A principal motivação deste projeto reside em proporcionar uma interface amigável de testes e simulação para quem irá, eventualmente, operar o Sistema de Controle Centralizado (SCC) da Linha 5 do metrô de São Paulo. O SCC é um sistema muito extenso, possuindo em média 1200 telas e mais de 40000 pontos de comunicação. Cada estação da Linha 5 é composta por três grandes sistemas, sendo que estes ainda são divididos em subsistemas. Além do mais, cada uma das 17 estações existentes possui suas peculiaridades, logo, é desejável ter uma certa experiência antes de operá-lo.

Assim, o SIMEA deve proporcionar um ambiente que retrate fielmente o funcionamento do SCC em campo, permitindo que o operador adquira conhecimentos necessários para operar o sistema com tranquilidade e segurança, além de ensiná-lo a lidar com eventuais situações consideradas críticas.

## 1.2 OBJETIVOS

O objetivo geral deste projeto de fim de curso é finalizar o desenvolvimento do SIMEA. Por se tratar de um sistema muito extenso e complexo, sua concepção já havia sido iniciada por outros membros da equipe porém, devido à circunstâncias internas à empresa, o projeto havia sido pausado. É de responsabilidade da aluna, então, finalizar o desenvolvimento do SIMEA e, para atingir tal objetivo, os seguintes objetivos específicos devem ser alcançados:

- Realizar um levantamento para verificar quais equipamentos não estão presentes no SIMEA e, em seguida, modelá-los;

- Desenvolver o código que permite automatizar parte do processo de instanciação das estações;
- Verificar quais estações da Linha 5 ainda não estão contempladas no SIMEA e, em seguida, instanciá-las;
- Desenvolver as lógicas de comando e dos alarmes dos equipamentos;
- Efetuar o Teste de Aceitação em Fábrica (TAF) do sistema, listar as pendências geradas no teste (caso existam) e corrigi-las.

### 1.3 RADIX ENGENHARIA E DESENVOLVIMENTO DE SOFTWARE

A Radix é uma multinacional de tecnologia e engenharia, com sedes no Rio de Janeiro, São Paulo, Belo Horizonte, Atlanta e Houston. A empresa oferece serviços e soluções altamente qualificados e com independência tecnológica, para atender as principais indústrias de processo do Brasil e do mundo. Chegou ao mercado em abril de 2010 e conta com um time de gerentes e consultores multidisciplinares com mais de 30 anos de experiência em projetos de engenharia, automação, TI industrial e desenvolvimento de software. Seu nome vem do latim e significa "raiz" ou "origem".

No ano de 2011 recebeu as certificações ISO 9001, ISO 14001 e OHSAS 18001. No mesmo ano, foi eleita a segunda melhor empresa de tecnologia para se trabalhar no Brasil e a melhor empresa para se trabalhar no estado do Rio de Janeiro pelo *Great Place to Work Institute*. Neste ano de 2020, foi eleita a melhor empresa para se trabalhar no Brasil (categoria médias empresas) e a segunda melhor na categoria TI e Telecom, ambos prêmios pelo instituto *Great Place to Work*.

A Radix também mantém um estreito relacionamento com o meio acadêmico brasileiro. O objetivo dessa proximidade é retribuir o ensino dado por essas universidades aos seus profissionais, assim como estimular o desenvolvimento de novas tecnologias, disseminar novos conhecimentos e contribuir para a formação de jovens profissionais mais capacitados, complementando o ensino oferecido nos centros acadêmicos com o conhecimento técnico e prático da empresa. Por isso, a empresa patrocina equipes de competição, apoia eventos e ministra palestras e minicursos em universidades e escolas.

A empresa investe não apenas em soluções tecnológicas, mas também no suporte estratégico à tomada de decisão do cliente. As tecnologias desenvolvidas pela Radix aumentam a eficiência e agregam valor ao negócio. Algumas delas surgiram de soluções, *frameworks* e metodologias elaboradas de forma customizada para empresas, sempre considerando o que já existe no mercado, atentando à relação custo e benefício e seguindo os melhores conceitos de engenharia de projetos. Atualmente,

atua em diversos setores do mercado tais como óleo e gás, mineração, petroquímica, transportes, saúde, entretenimento e energia.

Além do mais, para ajudar a equilibrar a rotina de trabalho, a Radix oferece uma série de atividades na empresa, que vão desde encontros para debater novas tecnologias, chamados de *meet ups*, bem como *hackathons* e eventos de confraternização.

A Radix é dividida em três unidades de negócios (UNs), cada uma focada em uma diferente área de atuação da empresa, sendo elas: UN de Serviços, UN de Óleo e Gás e UN de Metais e Mineração. O projeto de fim de curso que será descrito neste documento foi desenvolvido na UN de Serviços, cujo foco é oferecer soluções tecnológicas para o setor terciário da economia.

#### 1.4 ESTRUTURA DO DOCUMENTO

Este documento está dividido em seis capítulos. No Capítulo 2 é realizada uma breve revisão da literatura, destacando os métodos e ferramentas utilizadas e que são necessárias para compreensão do problema em questão. No Capítulo 3, por sua vez, são apresentados, de maneira breve, os principais objetivos e funcionalidades do SIMEA. O desenvolvimento e a implementação são mostrados no Capítulo 4, enquanto que os resultados do Teste de Aceitação em Fábrica (TAF) são apresentados no Capítulo 5. Por fim, as considerações finais e perspectivas de trabalhos futuros são abordadas no Capítulo 6.

## 2 SISTEMA SUPERVISÓRIO DA LINHA 5 DO METRÔ DE SÃO PAULO

Neste capítulo, são apresentados as principais características de sistemas SCADA, bem como os principais aspectos do software Elipse E3. Além disso, é realizada uma breve descrição do Sistema de Controle Centralizado (SCC) responsável por monitorar a Linha 5 do metrô de São Paulo, elencando seus principais objetivos e funcionalidades.

### 2.1 SISTEMAS SCADA

O termo SCADA é proveniente do inglês “*Supervisory Control And Data Acquisition*” e significa Controle Supervisório e Aquisição de Dados. SCADA é um tipo de sistema que permite supervisionar e controlar um processo produtivo ou instalação física, mediante à troca de informação entre uma estação central (denominada também de Unidade Terminal Mestre, MTU) e uma ou mais unidades remotas (CLP, RTU, IED). Por meio desses equipamentos, a aquisição de dados dos dispositivos de campo é realizada, fazendo-se necessário a utilização de redes de comunicação para atingir esse objetivo. As informações adquiridas são manipuladas, analisadas, armazenadas e, posteriormente apresentadas ao operador em multiplicidade de formas, mediante uma interface de alto nível (COELHO, 2010).

Os primeiros sistemas SCADA eram basicamente telemétricos e permitiam informar, periodicamente, o estado corrente do processo industrial, monitorando sinais representativos de medidas e estados de dispositivos, sem que houvesse nenhuma interface de interação com o operador. Com a evolução da tecnologia, os computadores passaram a ter um papel importante na supervisão dos sistemas, coletando e tornando disponíveis os dados do processo. O acesso remoto aos dados facilita tanto o monitoramento quanto o controle do processo, fornecendo, em tempo útil, o estado atual do sistema através de gráficos, previsões ou relatórios, viabilizando tomadas de decisões, seja automaticamente ou por iniciativa do operador (SOUZA, 2005).

Atualmente, os sistemas de automação industrial utilizam tecnologias de computação e comunicação para automatizar o monitoramento e controle dos processos industriais, efetuando a aquisição de dados em ambientes complexos, eventualmente dispersos geograficamente, e a respectiva apresentação de modo amigável para o operador, com recursos gráficos elaborados (interfaces homem-máquina) e conteúdo multimídia. Para permitir isso, os sistemas SCADA identificam todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação através de *tags*, podendo executar funções computacionais (operações matemáticas, lógicas, com vetores ou *strings*, dentre outros) ou representar pontos de entrada/saída de dados do processo que está sendo controlado. Neste caso, correspondem às variáveis do processo real (como por exemplo temperatura, nível ou vazão), se comportando como a ligação entre o con-

trolador e o sistema. É com base nos valores das *tags* que os dados adquiridos são apresentados ao usuário (COELHO, 2010).

Dentre as principais funções dos sistemas SCADA, (CONSTAIN, 2011) destaca as seguintes:

- Configuração da interface de comunicação, variáveis de entrada e saída, bem como dos operadores que terão acesso ao sistema;
- Construção de sinóticos, representando graficamente os processos industriais;
- Geração de alarmes, avisando ao usuário do sistema quando uma variável ou condição do processo de produção está fora dos valores previstos;
- Geração de *scripts* ou programas, que desenvolvem alguma lógica para o controle do processo;
- Geração de históricos, que registram os eventos relevantes;
- Produção de relatórios a partir dos dados da planta, dos alarmes ocorridos em um intervalo de tempo e dos acessos ao sistema por parte dos operadores;
- Geração de informação para níveis gerenciais.

De acordo com (COELHO, 2010), a partir do momento em que o monitoramento e o controle de um processo são feitos com a ajuda de um sistema supervisório, o processamento das variáveis de campo é mais rápido e eficiente. Qualquer evento imprevisto no processo é rapidamente detectado e mudanças nos *set-points* são imediatamente providenciadas pelo sistema supervisório, no sentido de normalizar a situação. Ao operador fica a incumbência de acompanhar o processo de controle da planta, como o mínimo de interferência, excetuando-se casos em que sejam necessárias tomadas de decisão de atribuição restrita ao operador.

## 2.2 SOFTWARE ELIPSE E3

O Elipse E3 é uma plataforma IHM/SCADA responsável por monitorar e controlar processos. Uma vez que oferece escalabilidade e constante evolução para diversos tipos de aplicação, pode ser utilizada desde em simples interfaces humano máquinas até em complexos centros de operação em tempo real (SALVADOR, 2019). O software possui três componentes principais, sendo eles:

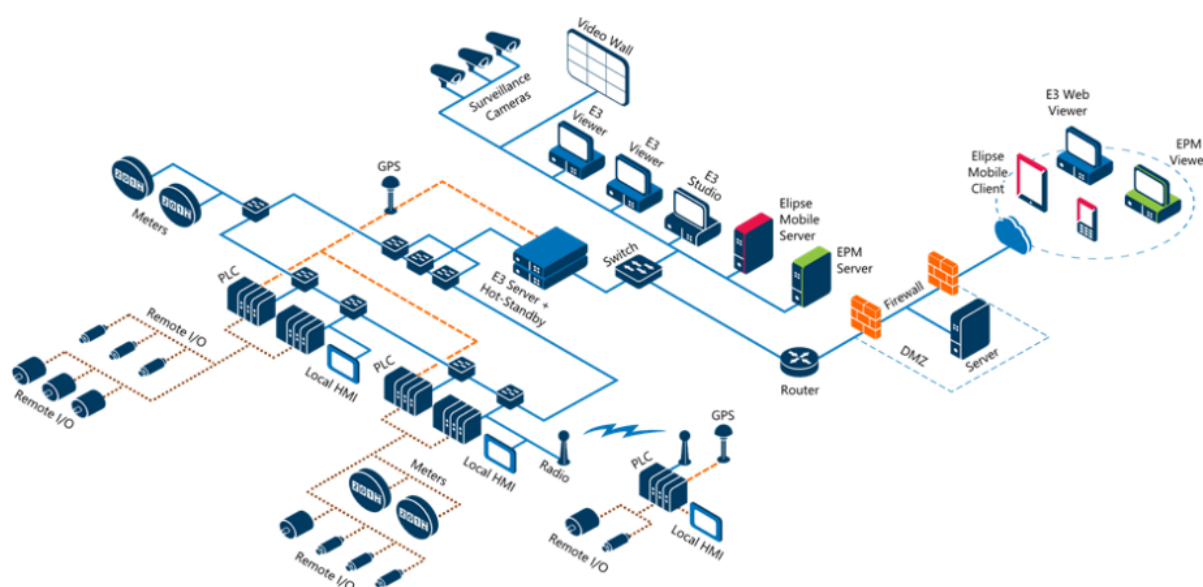
1. **E3 Studio:** É a ferramenta de configuração e desenvolvimento do sistema. Inclui desde a configuração da comunicação até editores de *scripts* e de gráficos para a criação das telas de operação. Além do mais, permite que uma mesma aplicação

seja editada por diversos usuários simultaneamente, facilitando o trabalho em equipe.

2. **E3 Server:** É o servidor de aplicações, onde são gerenciados os principais processos do sistema. Realiza a comunicação com os diversos pontos de aquisição de dados e é responsável pelo sincronismo de alarmes e da base de dados. Oferece grande robustez visto que permite que as informações gráficas sejam enviadas ininterruptamente aos clientes em qualquer local.
3. **E3 Viewer:** É a interface de operação com o usuário. Permite visualizar e operar a aplicação que está no servidor em qualquer computador, tanto na intranet quando na internet, via browser. Sendo assim, não existe a necessidade de instalar o projeto na máquina cliente, visto que todos os componentes e bibliotecas do sistema são baixados do servidor e atualizados automaticamente.

A Figura 2 apresenta um exemplo de arquitetura do Elipse E3, ilustrando a integração entre os componentes descritos.

Figura 2 – Exemplo de Arquitetura do Elipse E3.



Fonte: Elipse Software (2019).

Uma das grandes vantagens da ferramenta é a possibilidade de se comunicar com centenas de dispositivos de controle e aquisição de dados existentes no mercado. Além do mais, é possível gerenciar alarmes, desenvolver diversos tipos de interfaces gráficas - tanto em rede local quanto via internet - bem como gerar relatórios e armazenar as informações em um banco de dados (ELIPSE, 2019).

O software possui diversas ferramentas que possibilitam que a aplicação seja desenvolvida da forma mais robusta possível, sendo que as consideradas mais importantes para a compreensão deste documento são descritas a seguir.

### 2.2.1 Domínios

Sistemas supervisórios precisam realizar diversas tarefas, extrair dados úteis do processo e se comunicar entre si. No passado, para realizar tais funções, cada servidor SCADA deveria possuir uma cópia (parcial ou completa) da aplicação configurada na base de dados local, sendo que cada servidor só poderia executar uma base de dados configurada por vez. Isto pode resultar em problemas de administração, como, por exemplo, realizar o controle de versões das aplicações ou mesmo trabalhar com diferentes fabricantes de software.

O Eclipse E3 consegue solucionar estes problemas ao utilizar o conceito de domínio que é, basicamente, o conjunto de projetos e bibliotecas que compõem determinada aplicação. Portanto, o domínio é composto pela base de dados utilizada no projeto e está localizado dentro de um servidor que executa tarefas em tempo real. É possível executar mais de um projeto no mesmo domínio, sendo que a qualquer instante pode-se adicionar ou remover projetos e bibliotecas sem interromper ou prejudicar a execução do sistema (ELIPSE, 2019).

Quando dois ou mais projetos estão dentro de um domínio, eles podem acessar as propriedades e objetos um do outro como se residissem em uma única base de dados. Isso é possível porque um objeto pode estabelecer associação com o outro. Ou seja, se ambos os objetos existem e estão rodando, a associação estará ativa e qualquer mudança de valores é enviada assincronamente entre as partes. Se um dos objetos é destruído ou parado, a associação é interrompida e a aplicação, notificada.

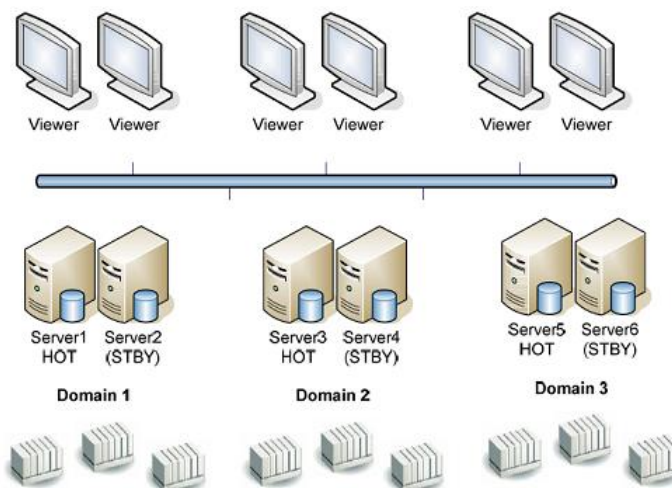
Um domínio é composto, fisicamente, por um servidor operando isoladamente, ou por dois servidores em uma configuração *dual standby*. Cada *Viewer*, na verdade, conecta-se ao domínio em si, e não apenas ao servidor. Uma vez estabelecida a conexão, o servidor inicia o carregamento dos objetos e bibliotecas definidas pelo usuário. O cliente inicia baixando a interface gráfica e salvando-a em um diretório cache. O procedimento pode levar alguns instantes dependendo do tamanho da aplicação, porém este tempo será recompensado nas próximas vezes em que a tela for aberta, visto que o *Viewer* executará o cache de disco e de memória. Além do mais, caso o *Viewer* não seja fechado, o tempo de paginação na próxima vez em que a mesma tela for aberta será menor, visto que ela já foi criada em memória. No entanto, caso a tela tenha sido reconfigurada, o *Viewer* irá baixá-la novamente e o processo, reiniciado.

Cada domínio é capaz de conversar com outros domínios, o que cria uma estrutura denominada de “super domínio”, exemplificada na Figura 3. Um *Viewer* conectado a um domínio isolado pode manipular informações residentes somente no servidor



ativo, de acordo com os direitos de acesso do usuário. Entretanto, caso o domínio esteja inserido em um domínio com outros irmãos, ele pode atuar como uma ponte para os outros servidores. Assim, quando é requisitada alguma informação inexistente no domínio local, o servidor é capaz de se comunicar em tempo real com outros *Viewers*, habilitando uma rede SCADA entre diferentes aplicações e sistemas.

Figura 3 – Super domínio no Elipse E3.



Fonte: Elipse Software (2019).

### 2.2.2 Bibliotecas de Símbolos e Usuários

O Elipse E3 possui ferramentas que permitem transformar qualquer objeto ou conjunto de objetos de uma aplicação em uma biblioteca do usuário. O conceito de biblioteca de usuário - também chamada de ElipseX - é oriundo das linguagens de programação orientadas a objetos, nas quais diversos componentes podem ser compartilhados entre diferentes projetos sem que haja a necessidade de desenvolver cada um dos objetos novamente (SALVADOR, 2019).

O uso de bibliotecas é aconselhável, uma vez que proporcionam diversas vantagens para a aplicação, tais como:

- Reutilização de código;
- Minimização de testes durante o desenvolvimento;
- Hierarquização de objetos;
- Criação de interfaces padrão para os objetos desenvolvidos.

As bibliotecas no E3 são arquivos com a extensão *.LIB* e podem ser de dois tipos: *XControls* e *XObjects*.

Os *XControls* são bibliotecas gráficas executadas no Viewer e que definem uma interface gráfica com o usuário. Uma vez criados e registrados, é possível incluí-los em uma tela e associar suas propriedades a variáveis externas. Qualquer mudança na biblioteca é automaticamente reproduzida em todas as instâncias do objeto, criando uma conexão de cada instância com a definição original. Além do mais, pode-se utilizar o mesmo *XControl* inúmeras vezes e em diferentes projetos ao mesmo tempo.

Os *XObjects* são bibliotecas de dados executadas no servidor e que possuem - além de propriedades, métodos e *scripts* - objetos relacionados ao servidor, tais como histórico de eventos e registros de alarmes. É possível definir seu comportamento ou associá-los a variáveis externas. Assim como com os *XControls*, qualquer mudança na definição do *XObject* original é automaticamente reproduzida em todas as instâncias do objeto.

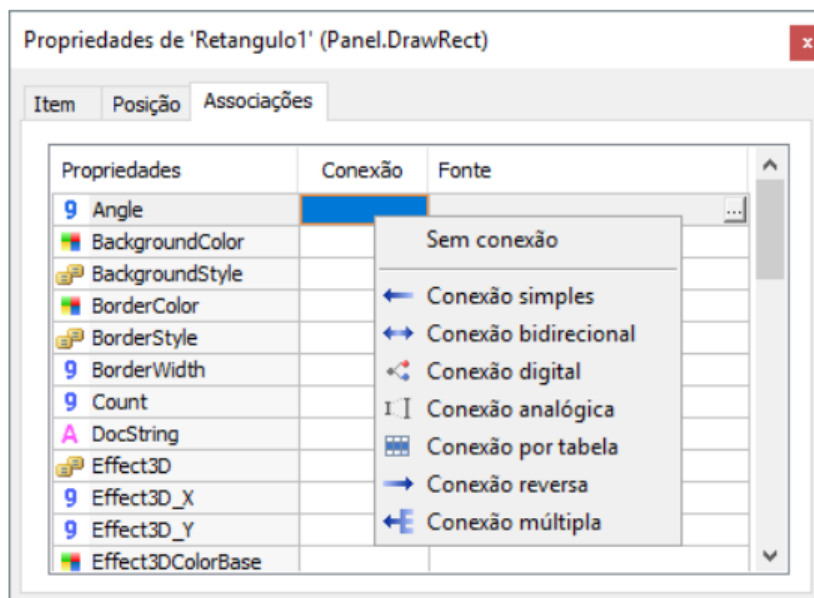
Outra grande vantagem do uso de bibliotecas no E3, é que elas podem ser associados entre si para melhorar o desempenho em aplicações complexas. Logo, é possível fazer com que um ou mais *XControls* acessem informações de um ou mais *XObjects*.

O Eclipse E3 também oferece uma poderosa ferramenta de *scripts* orientada a eventos e objetos, permitindo operações matemáticas, lógicas e manipulação de estruturas. Com ela, é possível desenvolver seus próprios eventos, executados em mudanças de estados de variáveis ou a intervalos fixos. Além disso, é possível utilizar *scripts* dentro das bibliotecas apresentadas anteriormente, dando maior poder aos componentes desenvolvidos de forma a evitar a repetição de código dentro da aplicação.

### 2.2.3 Associações

Associações são conexões realizadas entre propriedades e objetos ou entre outras propriedades. As associações trazem grande facilidade ao criar animações e outros tipos de lógicas comuns, minimizando a utilização de *scripts* (ELIPSE, 2020). Como é possível visualizar na Figura 4, existem sete tipos de associações no Eclipse E3: simples, bidirecional, digital, analógica, conexão por tabela, reversa ou múltipla. Nesta subseção são descritos os tipos de associações mais relevantes para este projeto.

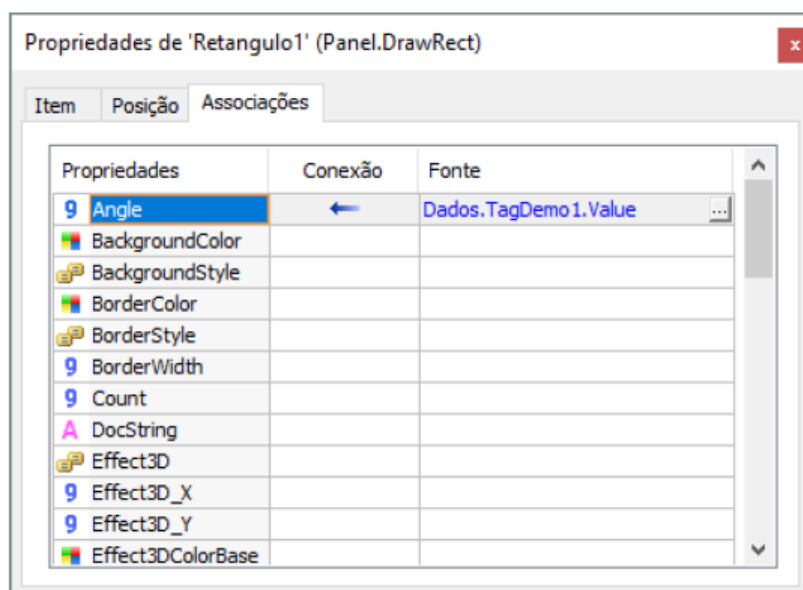
Figura 4 – Aba de associações no Elipse E3.



Fonte: Elipse Software (2020).

Na associação do tipo simples, o fluxo de dados é da fonte para a propriedade. Ou seja, o valor do campo fonte é transmitido para a propriedade associada a ele toda vez que for modificado. Na Figura 5, por exemplo, o valor da Tag Demo 1 será copiado para a propriedade *Angle* sempre que sofrer alteração.

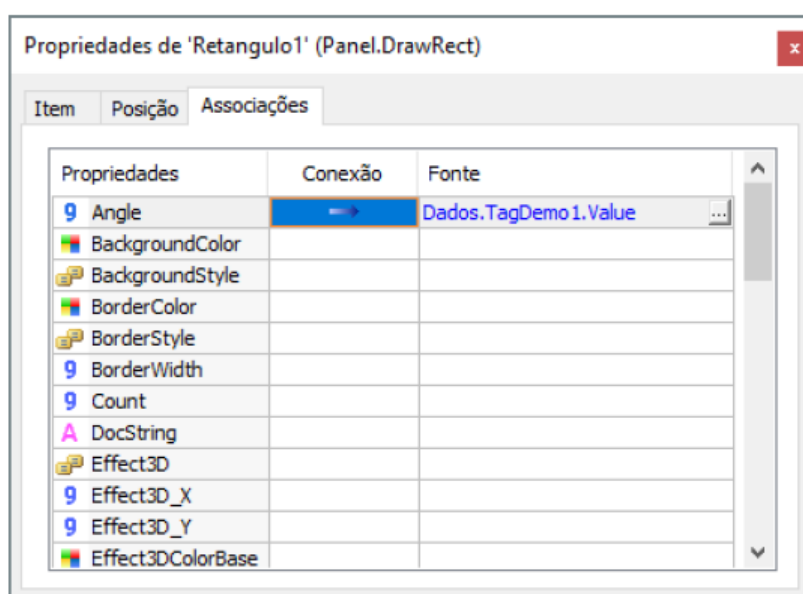
Figura 5 – Exemplo de associação simples no Elipse E3.



Fonte: Elipse Software (2020).

Por outro lado, a associação do tipo reversa funciona de maneira inversa à associação simples. Ou seja, sempre que o valor da propriedade for modificado, este será copiado para a fonte. Na Figura 6, por exemplo, o valor da propriedade *Angle* será repassado para a Tag Demo 1 toda vez que sofrer alteração.

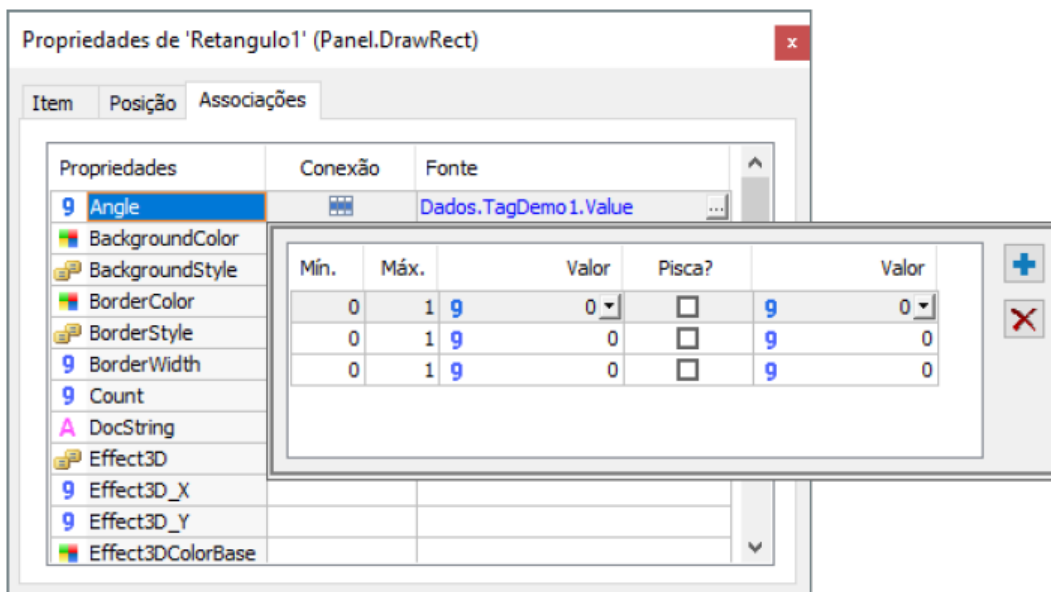
Figura 6 – Exemplo de associação reversa no Eclipse E3.



Fonte: Eclipse Software (2020).

Na associação por tabela é possível especificar um conjunto de intervalos em função de valores máximo e mínimo da fonte e, para cada um destes intervalos, o valor a ser assumido pela propriedade (ELIPSE, 2020). A Figura 7 ilustra um exemplo de uma associação por tabela no Eclipse E3.

Figura 7 – Exemplo de associação por tabela no Elipse E3.

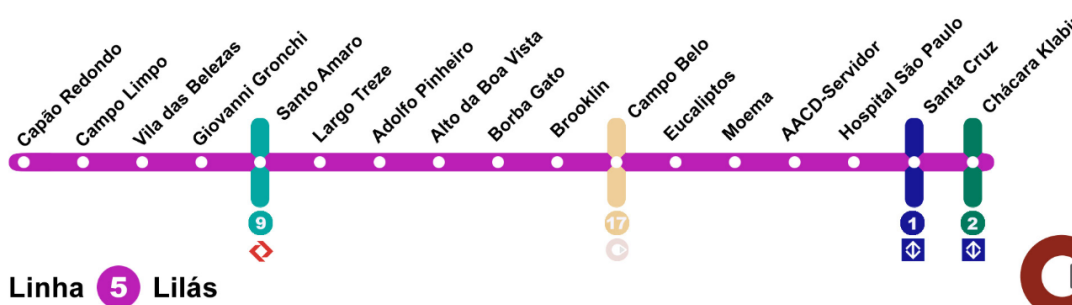


Fonte: Elipse Software (2020).

### 2.3 SISTEMA DE CONTROLE CENTRALIZADO - SCC

O Sistema de Controle Centralizado (SCC) é o sistema supervisório responsável por monitorar as 17 estações, apresentadas na Figura 8, da Linha 5 do metrô de São Paulo. Além do monitoramento das estações, o SCC é responsável também por supervisionar dois pátios e duas subestações de energia.

Figura 8 – Mapa da Linha 5 do metrô de São Paulo.



Fonte: (CPTM, 2019).

A Linha 5 está localizado na zona sul da cidade de São Paulo e possui 20 km de extensão, sendo que o trecho entre as estações Capão Redondo e Santo Amaro é elevado, enquanto que entre Largo Treze e Chácara Klabin é subterrâneo. A estação

Santo Amaro foi construída em uma ponte estaiada, por onde a linha passa exatamente sobre a confluência dos rios Jurubatuba e Guarapiranga, que formam o Rio Pinheiros. Além do mais, a linha 5 faz conexão com as linhas azul e verde, ligando diversas áreas movimentadas da cidades como complexos hospitalares, centros empresariais e polos comerciais. Estima-se que mais de 800 mil passageiros são transportados diariamente durante seu horário de funcionamento (domingo a sexta-feira das 4h40 à 0h00, e sábados das 4h40 à 1h00).

O SCADA do Sistema de Controle Centralizado é composto por três grandes sistemas. O Sistema de Controle Elétrico (SCE) supervisiona e controla o recebimento e a distribuição de energia, monitorando e controlando o consumo elétrico, bem como a demanda do sistema de transporte. O Sistema de Equipamentos Auxiliares (SEA), por sua vez, supervisiona e controla os equipamentos de serviços auxiliares, tais como: ventilação, bombas e iluminação. Além disso, monitora os sistemas de transmissão de dados, comunicação via rádio, gravação de som e detecção de incêndio. Por fim, o Sistema de Fluxo de Passageiros (SFP) é o responsável pela garantia de deslocamento seguro do usuário dentro da estação. É capaz de bloquear passagens, permitir deslocamentos verticais, bem como auxiliar pessoas com deficiência física por meio dos equipamentos monitorados. Para o controle e a supervisão do fluxo de passageiros são utilizados escadas rolantes, elevadores e catracas. Câmeras de monitoramento são utilizadas para garantir a segurança dos usuários do metrô, bem como para supervisionar equipamentos e as rotas do trem. O SFP também dispõe de comunicação audiovisual, possibilitando que os operadores se comuniquem por mensagem de voz e escrita.

Os três sistemas descritos anteriormente são operados de forma independente entre si. Ou seja, o usuário que estiver navegando pelo SCE não terá acesso à informações do SEA, nem do SFP e vice-versa. Além disso, é responsabilidade dos servidores do SCE, SEA e SFP armazenar informações sobre os equipamentos monitorados, sendo que o SCC deve garantir a correta operação destes. Em caso de situações anormais ou não esperadas, o SCC gera alarmes na tela e permite que o operador execute manobras para normalizar o sistema.

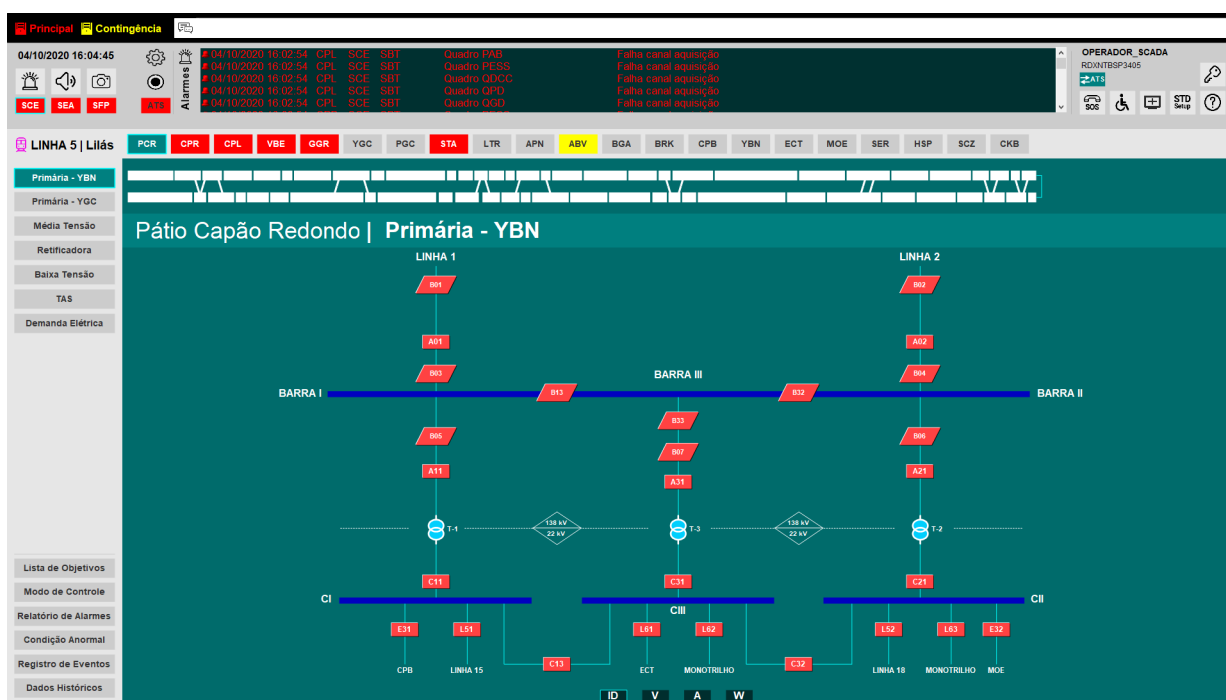
O SCC foi desenvolvido utilizando a plataforma Eclipse E3, sendo composto em média por 1200 telas e mais de 40000 *tags* de comunicação. Os dispositivos de controle e monitoramento utilizados em campo se conectam às concentradoras de dados presentes em cada uma das estações. A comunicação é feita utilizando o protocolo IEC-104, regido pela norma IEC-60870-5-104 (IEC, 2006), sendo do tipo mestre-escravo. Neste caso, o servidor SCADA atua como mestre e as concentradoras como escravos. Além do protocolo IEC-104, o SCC possui integração através de banco de dados e protocolo OPC.

Ainda, de modo a garantir maior disponibilidade, o SCC possui servidores redun-

dantes que funcionam no esquema de “hot standby”, possibilitando que dois servidores atuem um como contingência do outro. Ou seja, se o servidor principal falhar, o servidor de *backup* assume imediatamente.

A Figura 9 ilustra a tela na qual o usuário será direcionado após efetuar o *login* no sistema. A parte superior da tela é composta por botões funcionais, botões de navegação, janela reduzida de alarmes, informações do usuário conectado, *status* dos servidores, data e hora, barra de mensagens, dentre outras funcionalidades. A parte superior do menu lateral, localizado no lado esquerdo da tela, é formada por botões de navegação, agrupados em três categorias: botões dos subsistemas do sistema selecionado, botões com opções do subsistema selecionado e botões fixos comuns aos três sistemas. Por sua vez, na parte inferior do menu lateral é possível visualizar o relatório de alarmes e o registro de eventos do sistema, por exemplo.

Figura 9 – Tela inicial do SCC.



Fonte: Sistema de Controle Centralizado (2020).

Neste capítulo foi apresentado o Sistema de Controle Centralizado responsável por monitorar as 17 estações da Linha 5 do Metrô de São Paulo. O sistema foi desenvolvido utilizando a plataforma Elipse E3, sendo que esta é dividida em três componentes principais: E3 Studio, E3 Server e E3 Viewer. O SCC é composto por três grandes sistemas que operam de forma independente entre si. O SCE é responsável por monitorar a parte elétrica e a demanda do sistema de transporte; o SEA supervisiona e controla os equipamentos de serviços auxiliares como bombas e ventilação;

por fim, o SFP é responsável pela garantia de deslocamento seguro do usuário dentro da estação. O próximo capítulo tem como objetivo principal apresentar as principais funcionalidades do SIMEA.



### 3 SIMULADOR DE EQUIPAMENTOS E AUXILIARES - SIMEA

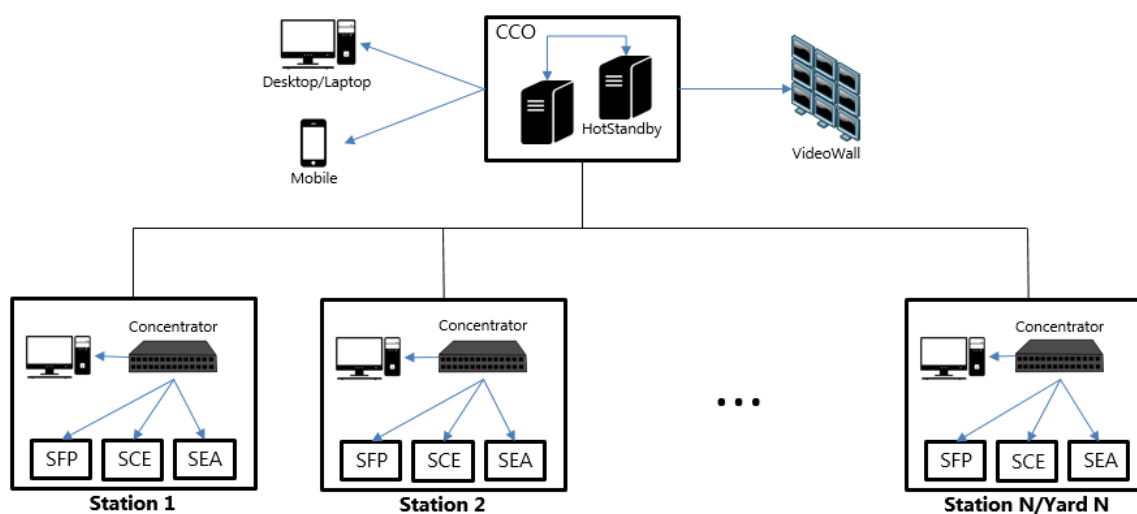
Neste capítulo, os principais objetivos do Simulador de Equipamentos e Auxiliares (SIMEA) são apresentados, juntamente com suas principais características e funcionalidades.

O SIMEA é responsável por simular todos os sinais provenientes do campo recebidos pelo Sistema de Controle Centralizado (SCADA/SCC) da Linha 5 do Metrô de São Paulo, sendo a ferramenta utilizada para simulação, treinamento e teste de estratégias operacionais dos sistemas SCE, SEA e SFP. Portanto, o simulador deve se comportar de forma similar ao campo, replicando lógicas de funcionamento e sendo capaz de reproduzir cenários programados e/ou acontecimentos armazenados na base histórica do SCC, possibilitando a análise de sequências de eventos da operação comercial. O ambiente de treinamento foi desenvolvido na plataforma Eclipse E3 e é composto pelos seguintes sistemas:

- SIMEA: o simulador para o SCADA do SCC;
- SIMTR: o simulador para o Sistema de Controle e Regulação de Trens (SCT), que deve integrar ao SIMEA;
- Cópia do SCADA do SCC: uma versão levemente modificada da aplicação do SCADA (mudanças apenas em IPs e nomes de servidores) do SCC do ambiente de produção que deve atender aos mesmos requisitos desta.

No SCC, as informações dos equipamentos são adquiridas pelas concentradoras do Sistema de Controle Local (SCL) e disponibilizadas em endereços pré-determinados. Existe uma concentradora de dados para cada uma das estações, subestações e pátios, conforme pode ser visualizado na Figura 10, sendo que estas são responsáveis por centralizar os dados obtidos dos Controladores Lógicos Programáveis (CLPs) de cada um dos subsistemas do SCC. O objetivo principal do SIMEA é substituir tais concentradoras sem que haja diferença perceptível para o usuário.

Figura 10 – Concentradoras de dados.



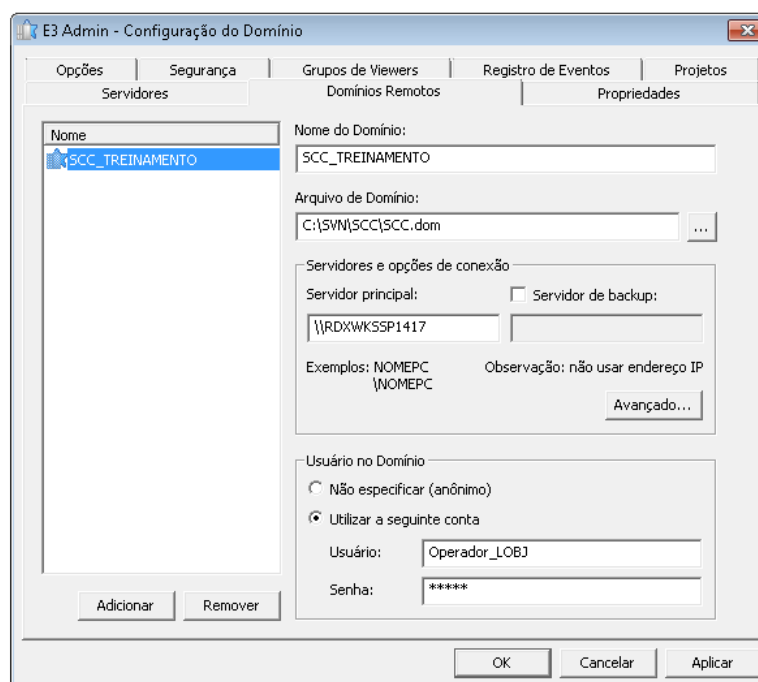
Fonte: Arquivo da Radix (2019).

O simulador também deve ser capaz de reproduzir cenários programados e possuir a funcionalidade *playback*, a qual permite a reprodução de acontecimentos reais armazenados na base de dados. Essa funcionalidade torna possível a análise de sequências de eventos da operação comercial.

A comunicação entre a cópia da aplicação do SCADA/SCC e o SIMEA é realizada por meio do protocolo IEC-104, o qual utiliza um modelo de comunicação mestre-escravo. Assim como as concentradoras, o SIMEA atua como escravo. Para que a comunicação entre os *drivers* seja devidamente estabelecida, é necessário configurar um domínio remoto. Domínios remotos são uma funcionalidade do Elipse E3 que possibilitam que exista comunicação entre diferentes servidores e *Viewers*, de modo que uma aplicação seja destinada a fornecer dados, e a outra a recebê-los. O domínio que fornece dados, neste caso o servidor do SCADA/SCC, é denominado de Domínio Servidor; já o domínio que utiliza esses dados, nesse caso o domínio do SIMEA, é chamado de Domínio Cliente.

A configuração das conexões com outros domínios do Elipse E3 deve ser realizada no Domínio Cliente, conforme mostrado na Figura 11.

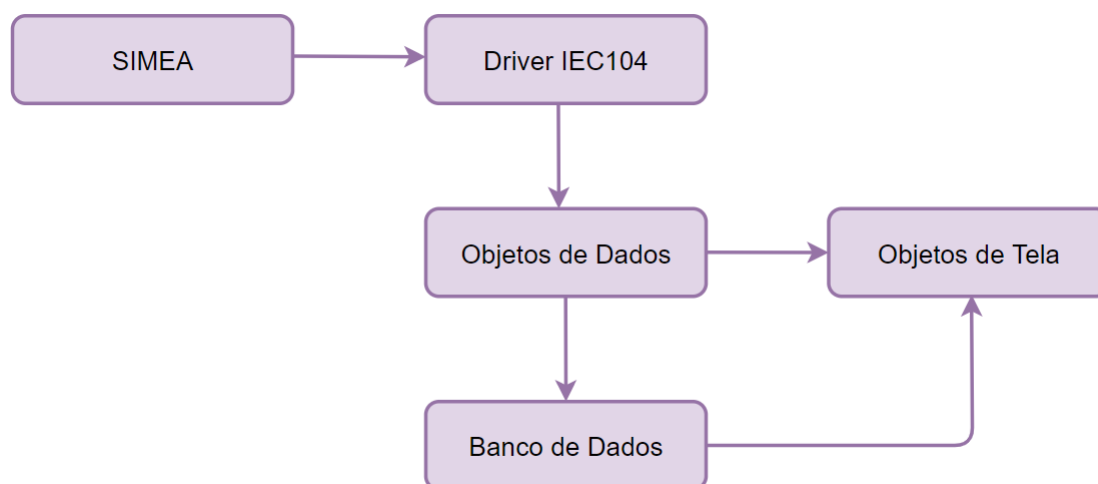
Figura 11 – Configuração do domínio remoto no Elipse E3.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

A informação de tempo real deve ser tratada da seguinte forma: ao atuar em um equipamento no SIMEA, a informação é enviada ao SCC por meio do *driver* de comunicação IEC-104 e, uma vez que a *tag* de *driver* atuada possui uma associação com o objeto de dados equivalente, a informação é recebida no SCC e registrada como evento no banco de dados. Ainda, caso a *tag* de *driver* atuada possua associação com um objeto de tela (chamados também de *XControl*), o equipamento poderá piscar, trocar de cor ou gerar alarmes, por exemplo. A Figura 12 apresenta o fluxo de dados descrito.

Figura 12 – Fluxo de dados no SIMEA.



Fonte: Elaborado pela autora (2020).

Os estados dos equipamentos são alterados conforme os comandos realizados pelo usuário ou através da reprodução de eventos pré-programados para a simulação. O SIMEA deve, portanto, ser capaz de:

- Editar e gerenciar cenários;
- Editar e gerenciar eventos do ambiente de simulação;
- Definir o cenário e a sequência de eventos que irão compor a simulação.

Além do mais, o SIMEA apresenta a funcionalidade de controle de acesso. Ou seja, somente usuários com permissão podem acessá-lo, protegendo assim os dados salvos e as configurações de simulações. Existe um usuário administrador e um grupo de usuários denominado de Instrutores. O usuário administrador é o único usuário capaz de criar e deletar usuários. Os usuários dentro do grupo Instrutores são capazes de utilizar todas as funcionalidades do SIMEA sem restrição. Para isso, parte-se do princípio que somente os instrutores possuem acesso ao SIMEA e que os alunos em treinamento somente interagem com a cópia da aplicação do SCADA/SCC.

### 3.1 CENÁRIOS

O termo cenário refere-se a um quadro estático de estados do SCC. O simulador possibilita que o usuário crie, edite, salve, exclua e carregue cenários. Para iniciar uma simulação, deve-se sempre partir de um cenário inicial, no qual todos os estados monitorados pelo SCC estejam definidos. Um cenário pode ser de quatro naturezas distintas:

- Cenário padrão;
- Cenário histórico;
- Cenário programado;
- Cenário de simulação salvo.

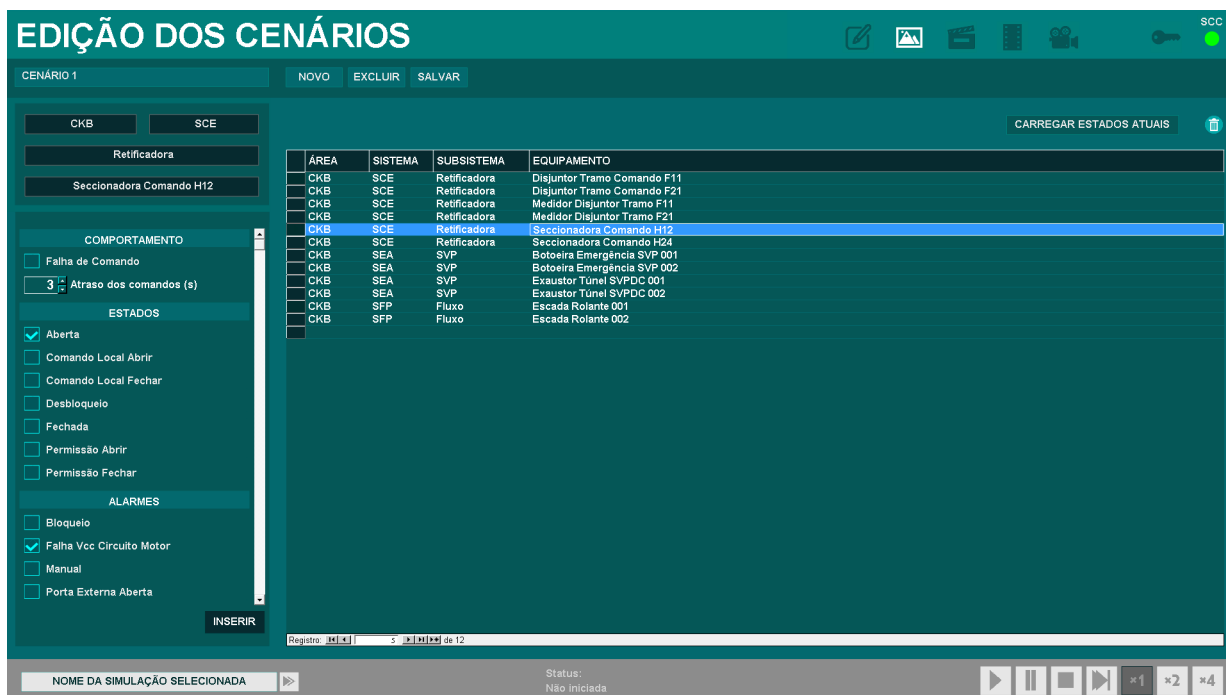
O SIMEA possui um cenário padrão que contém um valor predeterminado para todos os estados do sistema. Esse cenário será utilizado para iniciar qualquer simulação caso nenhum outro seja carregado. Por ser um cenário nativo do SIMEA, não está sujeito a alterações.

Por outro lado, um cenário histórico é aquele no qual os estados terão seus valores provenientes de dados históricos armazenados no banco de dados, sendo definido pela seleção de data e horários específicos. Um cenário programado, por sua vez, é proveniente de configurações de estados realizadas pelo usuário, sendo possível criá-lo a partir de qualquer cenário existente. Todos os cenários salvos podem ser excluídos ou alterados individualmente.

Além disso, durante qualquer momento da simulação, é possível salvar o valor dos estados naquele momento, criando um cenário de simulação salvo. Um cenário de simulação salvo pode ter sua origem de qualquer um dos cenários previamente apresentados. Dessa forma, o usuário pode carregar um cenário inicial, executar a simulação, e salvar o estado final dos equipamentos em um novo cenário. Assim como os cenários programados, os cenários salvos podem ser excluídos individualmente.

A Figura 13 ilustra a tela de edição dos cenários do SIMEA. Na parte superior esquerda é possível criar, excluir ou salvar um novo cenário. Para editar um cenário existente basta selecioná-lo, realizar as modificações necessárias e clicar no botão de salvar. No menu lateral direito, deve-se selecionar a estação, sistema, subsistema e o equipamento desejado. As propriedades de estado, bem como os alarmes do equipamento (caso existam), serão então exibidas na parte inferior esquerda da tela.

Figura 13 – Tela de edição dos cenários.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

### 3.2 EVENTOS

Um evento é definido como a mudança de estado em algum ponto de supervisão monitorado pelo SCC. Em outras palavras, ocorre um evento sempre que um estado mudar de valor. O SIMEA permite a alteração nos valores de qualquer ponto de supervisão da cópia do SCADA/SCC, respeitando a natureza do ponto de supervisão. Por exemplo, um ponto de supervisão do tipo booleano pode apenas receber os valores 0 e 1. No caso de valores analógicos, é possível também atribuir aos pontos um valor fora do *range* definido como padrão a fim de simular o comportamento do sistema para valores inválidos.

O SIMEA também permite a reprodução de sequências de eventos. As sequências podem ser determinadas a partir de um histórico de eventos reais armazenados no banco de dados, a partir de um histórico de eventos simulados previamente, ou podem, ainda, serem manualmente programadas no simulador. Enquanto a reprodução de eventos estiver ocorrendo, o SIMEA não estará simulando o sistema, apenas reproduzindo fielmente a sequência de eventos. Caso o SCC envie algum comando durante a reprodução, esse será ignorado pelo SIMEA. O usuário do SIMEA pode controlar a velocidade de reprodução da sequência de eventos, ou interrompê-la completamente.

As sequências de eventos permitem implementar padrões de comportamentos

no SIMEA. Uma sequência de eventos pode ter, no máximo, duas horas de duração, e é possível deixar salvas, simultaneamente, um total de 10 sequências, de forma que para se salvar uma nova após esse limite é necessário apagar uma antiga.

A Figura 14 ilustra a tela de edição das sequências de evento no SIMEA, bem como um exemplo de uma sequência de eventos. Tal como na tela de edição dos cenários, na parte superior esquerda é possível criar, excluir ou salvar uma nova sequência de eventos. No menu lateral esquerdo, deve-se selecionar a estação, sistema, subsistema e o equipamento desejado. As propriedades de estados, comandos e alarmes (caso existam) serão, então, exibidas na parte inferior esquerda da tela. Para adicionar um evento na sequência de eventos, basta selecionar a propriedade em que se deseja atuar e clicar na seta existente ao lado da respectiva propriedade. É possível, ainda, configurar o tempo de atraso de atuação do evento. Por exemplo, na Figura 14, o estado **Ligada** do equipamento Bomba Águas Pluviais 11 será atuado 20 segundos após a propriedade **Tensão Via** do Medidor Disjuntor Tramo 11 ter sido atuada para o valor de 220 V.

Figura 14 – Tela de edição das sequências de eventos.

ORDEM	ÁREA	SISTEMA	SUBSISTEMA	EQUIPAMENTO	PROPRIEDADE	VALOR	ATRASO
1	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Comando Abrir	True	5
2	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Permissão Fechamento	True	10
3	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Comando Fechar	True	5
4	APN	SCE	Retificadora	Medidor Disjuntor Tramo F11	Corrente	100	10
5	APN	SCE	Retificadora	Medidor Disjuntor Tramo F11	Tensão Via	220	10
6	APN	SEA	Bombas	Bomba Águas Pluviais 11	Ligada	True	20
7	APN	SEA	Bombas	Bomba Águas Pluviais 11	Falha Comando	True	3
8	APN	SEA	Bombas	Bomba Águas Pluviais 11	Manutenção	True	2
9	APN	SEA	Bombas	Bomba Águas Pluviais 11	Falha Comando	False	5
10	APN	SEA	Bombas	Bomba Águas Pluviais 11	Comando Desligar	True	16
11	APN	SFP	SME	Camera Fixa 001	Falha Comunicação	False	0
12	APN	SFP	SME	Camera Fixa 002	Falha Comunicação	False	0
13	APN	SFP	Fluxo	Elevador 001	Falha Técnica	True	4
14	APN	SFP	Fluxo	Elevador 002	Inspeção Manutenção	True	4

Fonte: Simulador de Equipamentos e Auxiliares (2020).

### 3.3 SIMULAÇÃO

Uma simulação pode ser utilizada para treinamento ou testes de novas estratégias. O usuário do SIMEA, chamado Instrutor, precisa selecionar um cenário para

iniciar a simulação. Assim, antes de a simulação ser iniciada, os estados dos equipamentos serão alterados de acordo com o cenário inicial selecionado. O usuário do SCADA/SCC pode, então, utilizar sua aplicação como se estivesse operando o sistema real, permitindo assim o contato com o sistema sem risco de prejudicar a operação.

O Instrutor tem liberdade de configurar os estados do simulador conforme desejar, seja para testar o comportamento do SCADA/SCC em situações específicas ou para ensinar os alunos a utilizarem-no. Ele pode, manualmente, alterar o valor dos estados que desejar, alterar cenários previamente configurados, ou executar sequências de eventos salvas no SIMEA.

A tela de edição das simulações do SIMEA é apresentada na Figura 15. A simulação exemplificada tem o Cenário 1 como cenário inicial e, uma vez que o Instrutor iniciá-la, as cinco sequências de eventos listadas na parte central da tela serão reproduzidas. Note que o instante inicial, localizado na parte superior esquerda da tela, representa o momento em que a simulação será iniciada, mas não necessariamente o momento em que a primeira sequência de eventos será executada. Isso se deve ao fato de que é possível configurar o tempo de atraso entre as sequências de eventos. Na Figura 15, por exemplo, a primeira sequência de eventos será executada 15 segundos após o instante inicial. A segunda sequência de eventos, por outro lado, será executada 20 segundos após a primeira e assim por diante.

Figura 15 – Tela de edição das simulações.

The screenshot displays the 'EDIÇÃO DAS SIMULAÇÕES' (Simulation Editor) interface. The main window is titled 'SIMULAÇÃO 1' and includes buttons for 'NOVA', 'EXCLUIR', and 'SALVAR'. The 'CENÁRIO INICIAL' (Initial Scenario) is set to 'CENÁRIO 1' with an 'INSTANTE INICIAL' (Initial Instant) of '18/10/2020 14:30:00'. A table lists the event sequences to be reproduced:

ORDEM	NOME	ATRASO
1	REPRODUÇÃO CENÁRIOS BGA	15
2	REPRODUÇÃO CENÁRIOS VBE	20
3	REPRODUÇÃO CENÁRIOS APN	60
4	REPRODUÇÃO CENÁRIOS BRK	45
5	DADOS HISTÓRICOS MEDIDORES	150

The interface also features a sidebar with 'SEQUÊNCIAS DE EVENTOS' (Event Sequences) and a control bar at the bottom showing the current time '18/10/2020 14:30:00' and the status 'Status: Simulação pronta para execução'.

Fonte: Simulador de Equipamentos e Auxiliares (2020).



A Figura 16 apresenta a tela de gerenciamento da simulação. Nela, é possível acompanhar a execução dos eventos em tempo real. Na parte superior direita é mostrado o número de eventos restantes, bem como a hora que o último evento será executado. Na parte inferior da tela, o Instrutor tem a opção de controlar a velocidade da simulação, bem como pausá-la ou avançá-la.

Figura 16 – Tela de gerenciamento da simulação.

Nome da Simulação em Andamento: SIMULAÇÃO 1  
 Número de Eventos Restantes: 3634  
 Hora do Último Evento: 18/10/2020 14:38:03

Grandezas Analógicas:

HORA	ÁREA	SISTEMA	SUBSISTEMA	EQUIPAMENTO	PROPRIEDADE	VALOR
18/10/2020 14:30:15	BGA	SFP	SME	Câmera Fixa 001	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 004	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 005	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 006	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 007	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 008	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 009	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 010	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 011	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 012	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 013	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 014	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 015	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 016	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 017	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 018	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 019	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 020	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 021	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 022	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 023	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 024	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 025	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 026	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 027	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 028	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 029	Falha Comunicação	True
18/10/2020 14:30:17	BGA	SFP	SME	Câmera Fixa 030	Falha Comunicação	True
18/10/2020 14:30:22	BGA	SFP	SME	Câmera Fixa 002	Falha Comunicação	True
18/10/2020 14:30:27	BGA	SFP	SME	Câmera Fixa 003	Falha Comunicação	True
18/10/2020 14:30:47	VBE	SEA	Bombas	Bomba Consumo 002	Manutenção	True
18/10/2020 14:30:57	VBE	SEA	Bombas	Bomba Consumo 002	Alarme Térmico	False
18/10/2020 14:31:02	VBE	SEA	Bombas	Bomba Consumo 002	Manutenção	True
18/10/2020 14:31:02	VBE	SEA	Bombas	Bomba Consumo 002	Ligada	False
18/10/2020 14:31:10	VBE	SEA	Bombas	Bomba Consumo 002	Ligada	True
18/10/2020 14:31:25	VBE	SEA	Bombas	Bomba Consumo 002	Aberto PCL	True
18/10/2020 14:32:30	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Comando Abrir	True
18/10/2020 14:32:40	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Permissão Fechamento	True
18/10/2020 14:32:45	APN	SCE	Retificadora	Disjuntor Tramo Comando F11	Comando Fechar	True

Status: Executando simulação

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Neste capítulo foram apresentados os principais objetivos do SIMEA, juntamente com suas funcionalidades básicas. O simulador deve ser capaz de simular as concentradoras de dados existentes em campo, atuando como escravo no protocolo de comunicação IEC-104. Além do mais, para que a comunicação entre os *drivers* seja estabelecida corretamente, domínios remotos devem ser configurados. Os conceitos de cenários e eventos foram definidos e as principais telas do sistema, apresentadas. O Capítulo 4 será, então, dedicado a apresentar a contribuição da aluna ao projeto.

## 4 DESENVOLVIMENTO E IMPLEMENTAÇÃO DO SIMEA

No Capítulo 3, foram apresentadas as principais funcionalidades do SIMEA. Como mencionado previamente neste documento, por se tratar de um sistema muito complexo, outros membros da equipe ajudaram a desenvolvê-lo ao longo deste ano. Quando a aluna entrou no projeto, muitas das funcionalidades já estavam desenvolvidas porém, o sistema não contemplava todas as estações da Linha 5 e, além disso, apresentava erros que precisavam ser corrigidos. Portanto, este capítulo tem o propósito de apresentar a contribuição da aluna no projeto, evidenciando o desenvolvimento necessário para atingir os objetivos específicos listados na Seção 1.2.

### 4.1 METODOLOGIA DE DESENVOLVIMENTO DO PROJETO

Uma metodologia de projetos consiste em um padrão de implantação, desenvolvimento e uso de ferramentas ao executar um projeto. Assim, as metodologias de projetos podem ser resumidas em um conceito: sistematização. Ao organizar a ordem de execução das tarefas, além do tempo investido em cada uma e os materiais usados, por exemplo, é mais fácil manter a execução do projeto alinhada com as expectativas do cliente, além de integrar as equipes e acelerar as entregas (SANTELLA, 2019).

Uma metodologia ágil, ao contrário das tradicionais, é adaptável ao rumo que o projeto toma durante a execução. O planejamento é desenvolvido de forma iterativa, ou seja, as características são definidas de acordo com as entregas. Se uma etapa do projeto acaba levando mais tempo para ser finalizada, por exemplo, o cronograma se adapta a esse imprevisto. Por isso, a tolerância a mudanças é muito maior (SANTELLA, 2019).

A primeira etapa do projeto consistiu em se reunir com o cliente com o intuito de especificar as funcionalidades do sistema, bem como alinhar as expectativas. Ao longo do projeto, os seguintes documentos foram elaborados:

1. Arquitetura do Sistema;
2. Especificação de Requisitos para o Simulador;
3. Requisitos Funcionais do Simulador;
4. *Template* de Telas do Simulador;
5. Procedimento de Testes do Simulador;
6. Relatório de Testes do Simulador;

A metodologia ágil utilizada neste projeto foi a do tipo Scrum, baseada no planejamento iterativo e composta por reuniões chamadas de *Sprints*. Nessas reuniões,

são elencadas a lista de atividades a serem desenvolvidas em determinada etapa do projeto. Em seguida, tais atividades são distribuídas entre os membros da equipe de desenvolvimento, que devem executá-las em um período de tempo definido também na reunião. Assim, é possível ter uma estimativa de quanto tempo a equipe levará para executar determinada tarefa, possibilitando a redução de riscos e o melhor planejamento de tempo e custos do projeto.

## 4.2 MODELAGEM DOS EQUIPAMENTOS E INSTANCIAMENTO DAS ESTAÇÕES

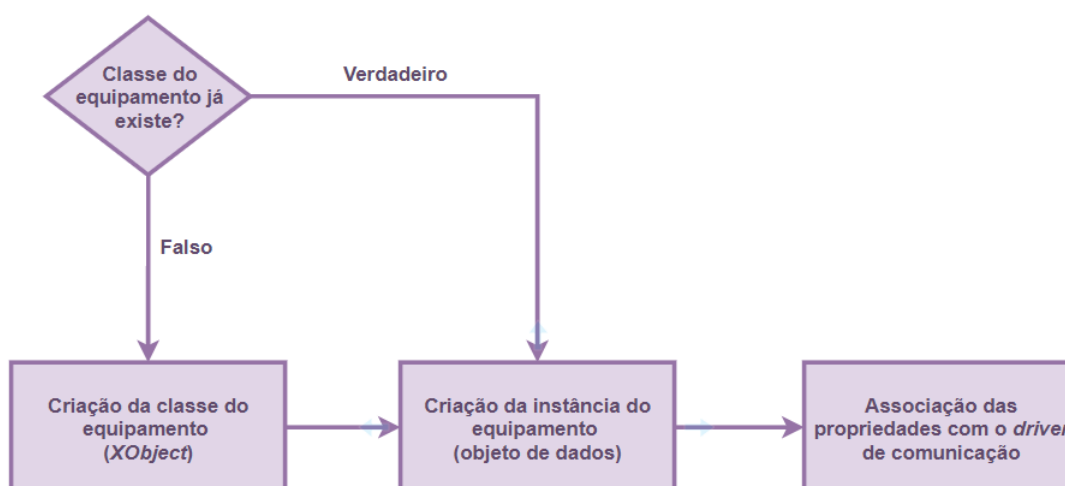
Em programação orientada a objetos, uma classe é definida como um conjunto de objetos com características comuns. Conforme explicado na Subseção 2.2.2 deste documento, no Eclipse E3 tais classes são denominadas de *XControls* e *XObjects*. Todos os equipamentos presentes no SCC devem possuir um *XObject* correspondente no SIMEA, sendo esse responsável por armazenar todas as propriedades, métodos e scripts necessários para que o equipamento tenha o comportamento desejado.

Uma vez que o *XObject* equivalente ao equipamento foi criado, é possível definir suas instâncias. Em programação orientada a objetos, a instância de uma classe é um objeto cujo comportamento e estado são definidos pela classe. Ou seja, a instância herda todos os métodos e atributos definidos na classe. Uma analogia interessante é pensar na classe como sendo a planta de uma casa: a casa ainda não existe, porém a estrutura necessária para criá-la sim. Um objeto, por sua vez, é a casa construída de acordo com os parâmetros definidos previamente na planta.

Após a criação das instâncias, é necessário criar um *link* entre cada uma de suas propriedades com as *tags* de comunicação correspondentes. Esse *link*, chamado de associação no Eclipse E3, irá permitir que um sinal enviado pelo SIMEA seja percebido pelo SCC e vice-versa.

A Figura 17 ilustra o procedimento descrito e que deve ser seguido sempre que for necessário instanciar uma nova estação no SIMEA.

Figura 17 – Passos necessários para instanciar uma estação no SIMEA.



Fonte: Elaborado pela autora (2020).

O Quadro 1 ilustra a situação de cada uma das estações da Linha 5 no SIMEA, indicando quais estavam ou não instanciadas no simulador.

Quadro 1 – Situação de cada uma das estações da Linha 5 no SIMEA.

<b>Estação</b>	<b>Instanciada no SIMEA?</b>
AACD-Servidor	Sim
Alto da Boa Vista	Sim
Adolfo Pinheiro	Sim
Borba Gato	Sim
Brooklin	Sim
Chácara Klabin	Sim
Campo Belo	Não
Campo Limpo	Não
Capão Redondo	Não
Eucaliptos	Sim
Giovanni Gronchi	Não
Hospital São Paulo	Sim
Largo Treze	Não
Moema	Sim
Santa Cruz	Não
Santo Amaro	Não
Vila Das Belezas	Não

Fonte – Elaborado pela autora (2020).

É válido ressaltar que as estações da Linha 5 possuem muitas divergências entre si. Logo, o tipo de equipamento que elas possuem também pode diferir. Em ou-

tras palavras, um equipamento presente na estação Adolfo Pinheiro, por exemplo, não necessariamente está presente em Vila das Belezas e vice-versa. Portanto, primeiramente foi realizado um levantamento para verificar quais equipamentos não possuíam seu modelo correspondente no SIMEA. O Quadro 2 ilustra o resultado do levantamento realizado.

Quadro 2 – Equipamentos sem *XObject* associado no SIMEA.

<b>Equipamento</b>	<b>Estação que está presente</b>
Contator Equalização	CPR, LTR e VBE
Cubículo	CPR
Inversor	CPL, CPR, GGR, LTR, STA e VBE
Medidor Barra	CPL, CPR e GGR
Medidor Grupo Retificador	CPR, LTR e VBE
Medidor SAT	VBE e CPB
Transferência Automática de Linha	CPB e LTR
Transferência Automática de Linha CPTM	CPB e LTR
Transformador SAT	CPB
Botoeira Emergência	LTR
Botoeira Emergência Geral	LTR
Jato Ventilador	LTR
Matriz Áudio Dante	CPL, GGR, STA e VBE
Painel Luz Bloco 4	CPL, CPR, GGR, LTR, STA e VBE
Ponte Estaiada	STA
Sistema Painéis Bloco 4	CPL, CPR, GGR, LTR, STA e VBE
Veneziana SVTE	LTR
Ventilador SVTE	LTR
Elevador Cadeirante	STA
Elevador Hidráulico	CPR, GGR, LTR e VBE
Escada Rolante Bloco 4	CPL, CPR, GGR, LTR, STA e VBE
Grupo Escada Rolante	CPL, CPR, GGR, LTR, STA e VBE

Fonte – Elaborado pela autora (2020).

Para cada um dos equipamentos mostrados no Quadro 2 foi criado um modelo correspondente no Elipse E3. Modelar um equipamento significa, de maneira geral, criar o seu *XObject* equivalente, juntamente com as propriedades de estado e comando associadas a ele. Na Figura 18 é possível visualizar o equipamento Veneziana SVTE modelado no SIMEA. As propriedades com o prefixo **stat** são pontos de estado enquanto que aquelas com o prefixo **cmd** representam pontos de comando. Essas propriedades também estão presentes no SCC, porém as propriedades com os prefixos **private**, **drv**, **prop** e **hst** estão presentes apenas no SIMEA visto que estão associadas com lógicas internas do simulador.

Figura 18 – Veneziana SVTE modelada no SIMEA.

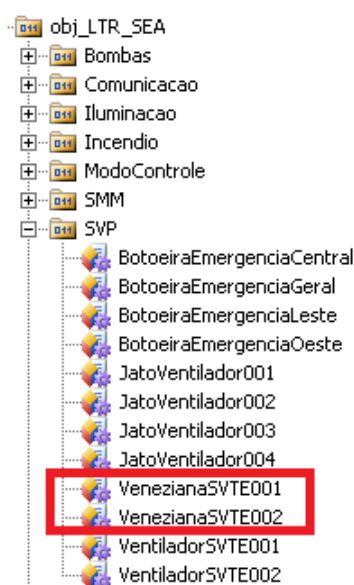
Nome	Tipo	Valor inicial	Texto de Ajuda
stat_AlarmesInibidos	Boolean	False	0: Alarme habilitado; 1: Não habilitado
prop_CodigoMetro	String	Empty	Nomenclatura mostrada em tela, alarmes e relatórios
prop_Area	String	INI	Estação ou pátio onde se encontra
prop_Sistema	String	SEA	Sistema do equipamento
prop_Subsistema	String	SVP	Subsistema do equipamento
stat_LigadoPCL	Boolean	False	0: Desligado; 1: Ligado
stat_AutomaticoCCM	Boolean	False	1: Automático
stat_ManualCCM	Boolean	False	1: Manual
stat_DesligadoCCM	Boolean	False	0: Ligado; 1: Ligado
stat_Automatico	Boolean	False	1: Automático
stat_Manual	Boolean	False	1: Manual
stat_Emergencia	Boolean	False	0: Normal; 1: Atuado
stat_Aberta	Boolean	False	0: Desligado; 1: Ligado
stat_ManutencaoPCL	Boolean	False	1: Manutencao
stat_SomatorioDefeitos	Boolean	False	0: Normal; 1: Atuado
stat_Fechada	Boolean	False	0: Desligado; 1: Ligado
stat_FalhaEletrica	Boolean	False	0: Normal; 1: Alarme
stat_DiscordanciaPosicao	Boolean	False	0: Normal; 1: Alarme
cmd_Abrir	Boolean	False	1: Abre
cmd_Fechar	Boolean	False	1: Fecha
prop_ValorComando	Integer	4	Esta variável serve indicar qual tela de comando deve abrir
stat_TravarLigar	Boolean	False	1: Trava Abrir
stat_TravarDesligar	Boolean	False	0: Trava Fechar
stat_InvalidoVeneziana	Boolean	False	Variável para avisar que a veneziana está inválida
prop_FonteEstadoPadrao	simea_VenezianaSVTE	EstadosPadres.sime	
prop_NomeAmigavel	String	"	Texto para identificação do equipamento na interface
drv_Abrir	Boolean	False	1: Abre
drv_Fechar	Boolean	False	1: Fecha
hst_Abrir	Boolean	False	cmd para Dados Históricos
hst_Fechar	Boolean	False	cmd para Dados Históricos
hst2_Abrir	Boolean	False	cmd para Dados Históricos
hst2_Fechar	Boolean	False	cmd para Dados Históricos
drv_Emergencia	Boolean	False	Variável auxiliar
private_Delay	Integer	0	Variável auxiliar de atraso dos comandos
private_OperadorComandoEnviado	String	SIMEA	Operador que enviou comando caso seja Dados Historicos; SIMEA caso seja outra forma de comando
private_FalhaComando	Boolean	False	Variável auxiliar de falha de comando
private_CmdAbrirExecutado	Boolean	False	Variável auxiliar de comando abrir executado
private_CmdFecharExecutado	Boolean	False	Variável auxiliar de comando fechar executado

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Como mencionado anteriormente, uma vez que a classe do equipamento foi definida, é possível criar quantas instâncias forem necessárias para o objeto, possibilitando ganho de performance, reutilização de códigos, e a otimização do processo.

A Figura 19 ilustra as instâncias criadas para as Venezianas SVTE da estação Largo Treze.

Figura 19 – Instâncias da Veneziana SVTE na estação Largo Treze.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

O próximo passo é realizar as associações entre o objeto de dados e o *driver* de comunicação. Existem sete tipos diferentes de associações no Eclipse E3, porém neste projeto a maior parte das associações são do tipo simples e reversa. Além disso, para implementar algumas lógicas específicas, também fez-se o uso de associações por tabela. No entanto, para esse caso em específico, diz-se que as associações por tabela são do tipo *fake*, visto que servem apenas como variáveis auxiliares. O funcionamento das associações *fake* será detalhado na Subseção 4.3.2.

Na Figura 20 é possível visualizar como foram realizadas as associações entre o objeto de dados e o *driver* de comunicação para a Veneziana SVTE 001 de Largo Treze.

Figura 20 – Exemplo de associações entre o objeto de dados e o *driver* de comunicação.

Item	Área de Alarmes	Associações
Propriedades	Conex...	Fonte
cmd_Abrir		"Associacao Para Exibir em Tela"
cmd_Fechar		"Associacao Para Exibir em Tela"
drv_Abrir	←	drv_IJC_LTR_SEA_SVP.VenezianaSVTE001.cmd_AbreFecha.CMD.BR00
drv_Emergencia	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_Emergencia].[Value]
drv_Fechar	←	drv_IJC_LTR_SEA_SVP.VenezianaSVTE001.cmd_AbreFecha.CMD.BR02
prop_FonteEstadoPadrao	⇒	EstadosPadres.simea_VenezianaSVTE
stat_Aberta	→	[drv_IJC_LTR].[SVP].[VenezianaSVTE001].[stat_Aberta].[Value]
stat_Automatico	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_Automatico].[Value]
stat_AutomaticoCCM	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_AutomaticoCCM].[Value]
stat_DesligadoCCM	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_DesligadoCCM].[Value]
stat_DiscordanciaPosicao	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_DiscordanciaPosicao].[Value]
stat_Emergencia	⇒	"Associacao Para Exibir em Tela"
stat_FalhaEletrica	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_FalhaEletrica].[Value]
stat_Fechada	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_Fechada].[Value]
stat_LigadoPCL	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_LigadoPCL].[Value]
stat_Manual	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_Manual].[Value]
stat_ManualCCM	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_ManualCCM].[Value]
stat_ManutencaoPCL	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_ManutencaoPCL].[Value]
stat_SomatorioDefeitos	→	[drv_IJC_LTR].[SEA].[SVP].[VenezianaSVTE001].[stat_SomatorioDefeitos].[Value]
stat_TravarDesligar	←	obj_LTR_SEA_SVP.VenezianaSVTE001.stat_Fechada
stat_TravarLigar	←	obj_LTR_SEA_SVP.VenezianaSVTE001.stat_Aberta
Count		

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Como pode ser visualizado na Figura 20, o processo de associar os objetos de dados com os *drivers* de comunicação é altamente manual, visto que para cada uma das *tags* de comunicação existentes é necessário criar um *link* com a propriedade da instância correspondente. No entanto, o software Elipse E3 aceita a importação de arquivos CSVs, permitindo, assim, que as associações e objetos de dados necessários sejam criados de forma automatizada. Desse modo, a solução encontrada foi a de desenvolver um código macro no Excel que combinasse as informações da lista de pontos da estação (enviada pelo Metrô de São Paulo) com as informações presentes no SCC.

O código desenvolvido possui diferentes funções que realizam diversas verificações de acordo com o sistema e estação que se está instanciando, sendo possível utilizá-lo em qualquer uma das estações da Linha 5. Ao final da execução, são gerados três arquivos CSVs referentes aos três grandes sistemas do Sistema de Controle Centralizado (SCE, SEA e SFP). Uma vez finalizada essa etapa, basta realizar a importação dos arquivos para o Elipse E3 e as instâncias dos objetos são automaticamente criadas e suas propriedades associadas com as *tags* de comunicação correspondentes.

Utilizando o código desenvolvido as estações que ainda não estavam presentes no SIMEA, conforme apresentado no Quadro 1, foram instanciadas. Ao importar o arquivo CSV gerado para o sistema SCE da estação Santa Cruz (SCZ), por exemplo, automaticamente são criadas as pastas e objetos mostrados na Figura 21.



Figura 21 – Objeto de dados do Sistema de Controle Elétrico da estação Santa Cruz.

Nome	Valor
[-] obj_SCZ_SCE	
[+] PCC	
[+] ModoControle	
[+] SBT	
[+] SMT	
DisjuntorCmdMediaTensaoE11	
DisjuntorCmdMediaTensaoE71	
DisjuntorCmdMediaTensaoE81	
DisjuntorCmdMediaTensaoL08	
DisjuntorCmdMediaTensaoL11	
DisjuntorCmdMediaTensaoL21	
DisjuntorCmdMediaTensaoL71	
DisjuntorCmdMediaTensaoL81	
Transformador3Enrol4T3	
Transformador3Enrol6T3	
Transformador3Enrol9T3	
MedidorDisjuntorE71	
MedidorDisjuntorE81	
MedidorDisjuntorE11	
MedidorDisjuntorL11	
MedidorDisjuntorL71	
MedidorDisjuntorL81	
MedidorDisjuntorL21	
[+] Retificadora	
DetectorArcoB01_1_2	
DetectorArcoB03a7	
SeccionadoraComandoJ11R11	
DisjuntorTramoComandoF11	
DisjuntorTramoComandoF21	
DisjuntorTramoComandoF31	
DisjuntorTramoComandoF41	
DisjuntorTramoComandoF02	

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Para cada um das instâncias presentes no objeto de dados, as propriedades foram devidamente associadas com a *tag* de comunicação correspondente, conforme pode ser visualizado na Figura 22.

Figura 22 – Associações criadas de forma automatizada para o equipamento Disjuntor Tramo Comando F11 da estação Santa Cruz.

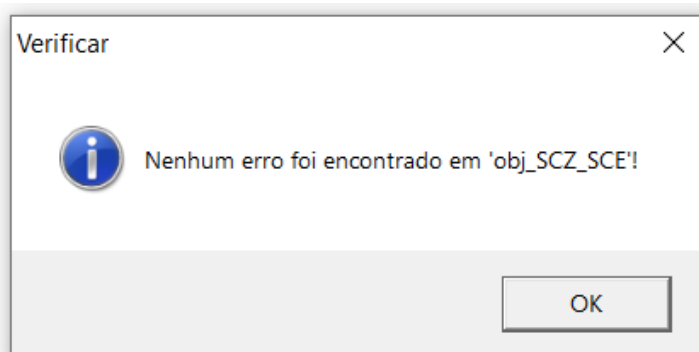
Propriedades de 'DisjuntorTramoComandoF11' (simea\_sce.simea\_DisjuntorTramoComando)

Item	Área de Alarmes	Associações
Propriedades	Conexão	Fonte
<input checked="" type="checkbox"/> alm_Bloqueio	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_Bloqueio].[Value]
<input checked="" type="checkbox"/> alm_CarrinhoPosicaoFalha	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_CarrinhoPosicaoFalha].[Value]
<input checked="" type="checkbox"/> alm_CarrinhoPosicaoIndefinida	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_CarrinhoPosicaoIndefinida].[Value]
<input checked="" type="checkbox"/> alm_CarrinhoPosicaoManutencao	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_CarrinhoPosicaoManutencao].[Value]
<input checked="" type="checkbox"/> alm_Desligamento	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_Desligamento].[Value]
<input checked="" type="checkbox"/> alm_DesligamentoBloqueio	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DesligamentoBloqueio].[Value]
<input checked="" type="checkbox"/> alm_DesligamentoMecanico	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DesligamentoMecanico].[Value]
<input checked="" type="checkbox"/> alm_DesligamentoNaoReligamento	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DesligamentoNaoReligamento].[Value]
<input checked="" type="checkbox"/> alm_DesligamentoReligamentoAuto	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DesligamentoReligamentoAuto].[Value]
<input checked="" type="checkbox"/> alm_DeteccaoArcoInterno	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DeteccaoArcoInterno].[Value]
<input checked="" type="checkbox"/> alm_DeteccaoIDS	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DeteccaoIDS].[Value]
<input checked="" type="checkbox"/> alm_DPFAlarmeTermico	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_DPFAlarmeTermico].[Value]
<input checked="" type="checkbox"/> alm_FalhaCaboBloqueio	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaCaboBloqueio].[Value]
<input checked="" type="checkbox"/> alm_FalhaCircuitoBobina	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaCircuitoBobina].[Value]
<input checked="" type="checkbox"/> alm_FalhaComando	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaComando].[Value]
<input checked="" type="checkbox"/> alm_FalhaDDLMaisI	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaDDLMaisI].[Value]
<input checked="" type="checkbox"/> alm_FalhaDDLMaisT	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaDDLMaisT].[Value]
<input checked="" type="checkbox"/> alm_FalhaDDLMenosI	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaDDLMenosI].[Value]
<input checked="" type="checkbox"/> alm_FalhaDDLMenosT	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaDDLMenosT].[Value]
<input checked="" type="checkbox"/> alm_FalhaIED	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaIED].[Value]
<input checked="" type="checkbox"/> alm_FalhaIMaxMais	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaIMaxMais].[Value]
<input checked="" type="checkbox"/> alm_FalhaIMaxMaisMais	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaIMaxMaisMais].[Value]
<input checked="" type="checkbox"/> alm_FalhaIMaxMenos	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaIMaxMenos].[Value]
<input checked="" type="checkbox"/> alm_FalhaIMaxMenosMenos	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaIMaxMenosMenos].[Value]
<input checked="" type="checkbox"/> alm_FalhaITripTermico	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FalhaITripTermico].[Value]
<input checked="" type="checkbox"/> alm_FusivelITensaoVia	→	[drv_IEC_S CZ].[SCE].[Retificadora].[DisjuntorTramoComandoF11].[alm_FusivelITensaoVia].[Value]

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Ao importar as planilhas é possível que ocorram erros de importação. Por exemplo, se alguma propriedade presente no objeto de dado não tenha sido criada anteriormente na classe do objeto, o Eclipse E3 apontará um erro. Para resolvê-lo, basta criar a propriedade no *XObject* do equipamento em questão. Resolvidos os erros de importação, é recomendado realizar uma verificação na pasta correspondente à estação e sistema que está sendo instanciado para garantir que nenhum erro tenha passado despercebido. Caso a mensagem mostrada na Figura 23 apareça na tela, a estação foi instanciada corretamente.

Figura 23 – Mensagem indicando que a verificação foi efetuada com sucesso.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

O procedimento análogo foi realizado para o restante das estações. Assim, a macro desenvolvida permitiu a criação de mais de 3000 instâncias de objetos de forma automatizada, possibilitando ganho de tempo e a otimização do processo.

### 4.3 LÓGICAS DESENVOLVIDAS PARA OS EQUIPAMENTOS

Como mencionado anteriormente, o SIMEA deve ser capaz de reproduzir as lógicas de funcionamento do SCC, uma vez que é a ferramenta responsável por simular os sinais provenientes do campo. Consequentemente, deve ser capaz de gerar alarmes no SCC, atuar nos estados dos equipamentos e também comandá-los (caso o equipamento seja comandável). Assim, as lógicas desenvolvidas para garantir o correto funcionamento do simulador são descritas nas próximas subseções.

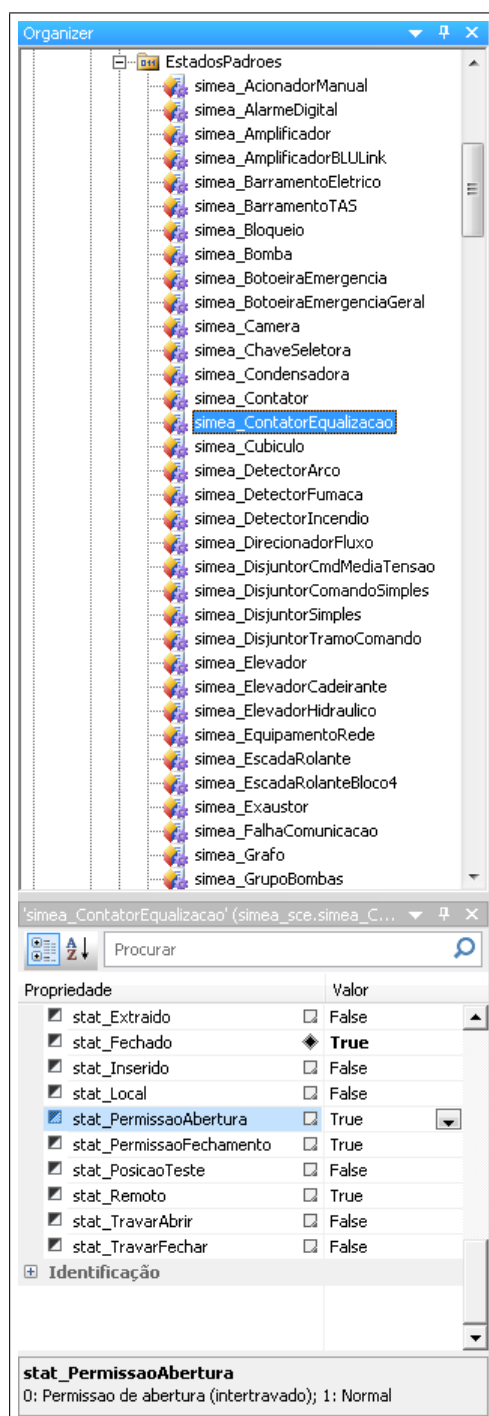
#### 4.3.1 Estados Padrões dos Equipamentos

Como mencionado na Seção 3.1, o SIMEA possui um cenário padrão que contém um valor predeterminado para todos os estados dos equipamentos sistema. Assim, caso nenhum cenário seja especificado no início de uma simulação, os estados dos equipamentos serão automaticamente setados conforme o definido na pasta de Estados Padrões.

A pasta Estados Padrões é composta por uma instância de cada um dos equipamentos do SIMEA, possuindo todas as propriedades analógicas e booleanas presentes no modelo do *XObject*. Logo, após efetuar a modelagem de um novo equipamento no sistema, é necessário definir o seu estado padrão. A Figura 24 ilustra o estado padrão do equipamento Contator Equalização. Note que as propriedades **stat\_Fechado**, **stat\_PermissaoAbertura**, **stat\_PermissaoFechamento** e **stat\_Remoto** estão configuradas para *True*, enquanto que o valor lógico do restante das propriedades está configurado para *False*. Assim, todos os equipamentos do tipo Contator Equalização existentes ao longo da aplicação serão iniciados com a mesma configuração.

Um código responsável por configurar os estados dos equipamentos para os estados padrões foi desenvolvido. Assim que o domínio do SIMEA é iniciado, ele percorre todas as pastas dos objetos de dados das estações e configura o valor das propriedades conforme o especificado. Tal fato permite a otimização do processo, visto que sem o código seria necessário ir manualmente até cada uma das pastas de objeto de dados e setar cada uma das propriedades para o valor desejado. O tempo total de configuração é, em média, de três minutos.

Figura 24 – Estado padrão do equipamento Contator Equalização.



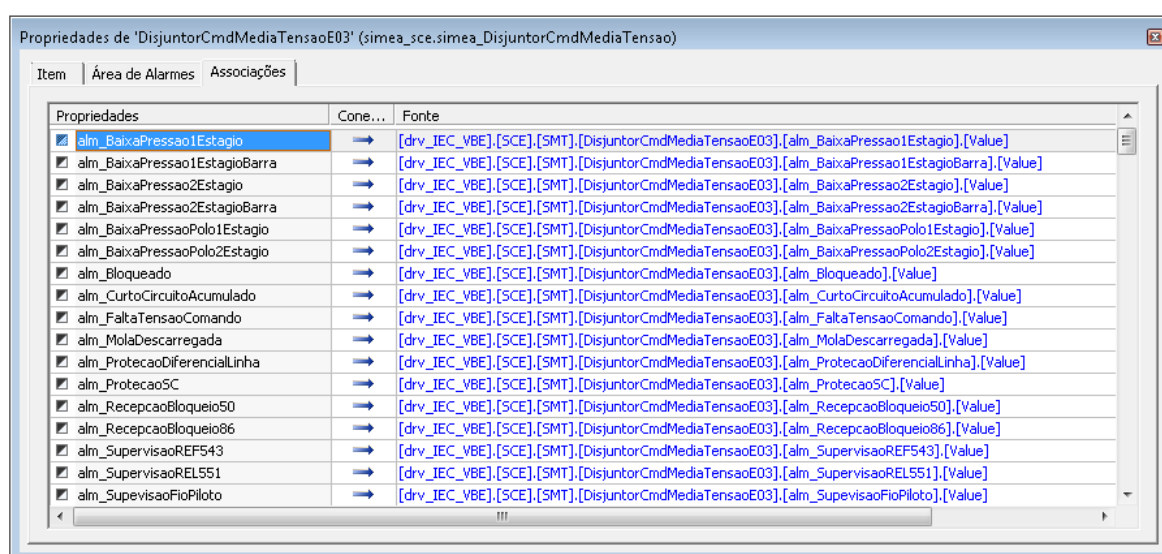
Fonte: Simulador de Equipamentos e Auxiliares (2020).

### 4.3.2 Lógica dos alarmes

As propriedades do SCC são, em sua maioria, do tipo booleanas e, consequentemente, recebem somente valores do tipo 0 ou 1. A exceção se aplica às propriedades dos medidores analógicos - que recebem uma grandeza analógica como valor

de entrada - e a algumas *tags* de equipamentos específicos. Todos os pontos que representam um alarme no sistema são booleanos, sendo que a maior parte deles são atuados quando a propriedade recebe o valor lógico 1. A Figura 25 ilustra um exemplo de como são feitas as associações no objeto de dados para pontos que alarmam em 1. Note que a associação é do tipo reversa, ou seja, ao atuar na propriedade **alm\_BaixaPressao1Estagio**, por exemplo, é enviado um sinal para a *tag* de *driver* associada a ela e um alarme é gerado no SCC.

Figura 25 – Exemplo de associação para pontos de alarmes que são atuados quando recebem o valor analógico 1.

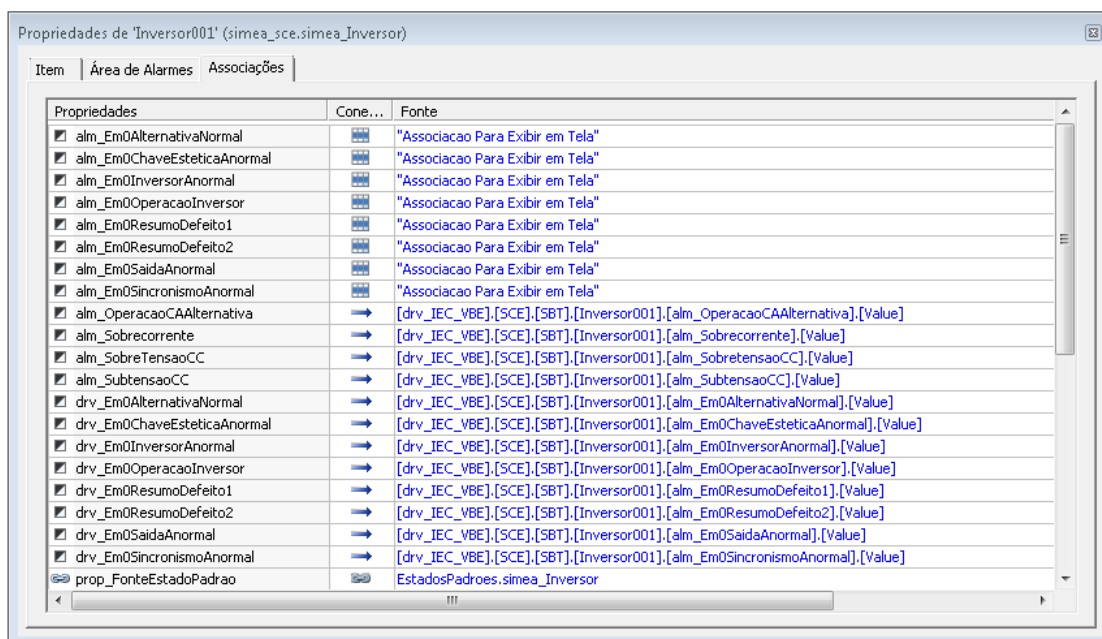


Fonte: Simulador de Equipamentos e Auxiliares (2020).

No entanto, alguns alarmes apresentam o funcionamento contrário, ou seja, são atuados quando a propriedade recebe o valor lógico 0. Logo, a lógica de funcionamento dos alarmes precisou ser revista para tais pontos. A Figura 26 ilustra como foram desenvolvidas as associações para os pontos que alarmam em 0. Note que, para esse caso em específico, a associação reversa entre o alarme e o driver de comunicação é feita por meio das propriedades com o prefixo **drv\_Em0**. As propriedades com esse prefixo estão presentes somente no SIMEA, uma vez que desempenham o papel de variáveis auxiliares e, conseqüentemente, fazem parte de lógicas inerentes ao simulador. Para cada um dos alarmes que atuam em 0 foi criado, na classe do equipamento, um *script* do tipo *On Property Change*. Esse tipo de *script* é executado toda vez que a propriedade associada a ele tiver seu valor alterado. Conforme pode ser visualizado na Figura 27, a propriedade **drv\_Em0SaidaAnormal** do inversor recebe o valor negado da propriedade **alm\_Em0SaidaAnormal**. Ou seja, quando o SIMEA enviar 1 para a *tag* **alm\_Em0SaidaAnormal**, a *tag* **drv\_Em0SaidaAnormal** receberá o valor analó-

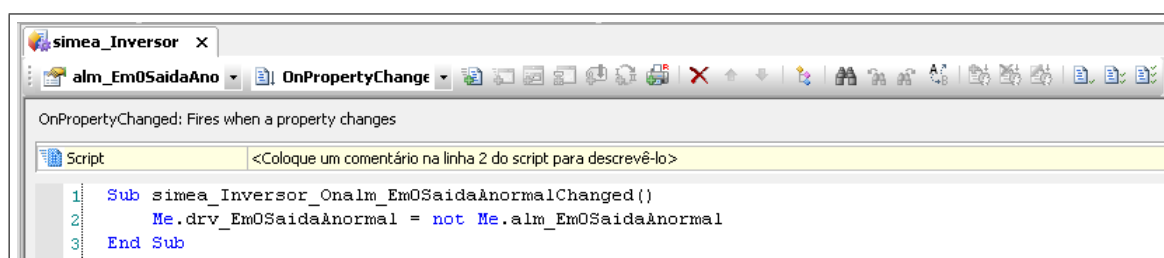
gico 0 e o alarme será atuado no SCC. Caso contrário, a tag **drv\_Em0SaidaAnormal** receberá o valor analógico 1 e o alarme será normalizado.

Figura 26 – Exemplo de associação para pontos de alarmes que são atuados quando recebem o valor analógico 0.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

Figura 27 – Exemplo de *script* do tipo *On Property Change* que é utilizado em propriedades de alarmes que são atuados quando recebem o valor analógico 0.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

Todas as propriedades de estado, alarme e comando referentes a um equipamento são exibidas em um objeto de tela chamado de *PlaceHolder*. Assim, cada vez que o usuário selecionar a estação, sistema, subsistema e equipamento, as propriedades referentes ao respectivo equipamento serão apresentadas na tela. A Figura 28

ilustra, por exemplo, o *PlaceHolder* preenchido com as propriedades do equipamento Disjuntor Comando Simples M02 da estação Vila das Belezas.

Figura 28 – *PlaceHolder* do SIMEA.

VBE SCE

SBT

Disjuntor Comando Simples M02

COMPORTAMENTO

Falha de Comando

Atraso dos comandos (s)

COMANDOS

Comando Abrir

Comando Fechar

ESTADOS

Aberto

Extraído

Fechado

Inserido

Posição Teste

ALARMES

Proteção Atuada

Circuito Medição Aberto

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Como mencionado anteriormente, as propriedades com os prefixos **prop**, **private** e **drv** representam lógicas internas ao simulador e, portanto, não devem ser exibidas no *PlaceHolder*. O código desenvolvido para configurar os equipamentos para os estados padrões é capaz de solucionar esse problema, uma vez que faz uma varredura nas pastas de objetos de dados das estações assim que o domínio do SIMEA é iniciado. Logo, caso a propriedade possua associação no objeto de dados, o código irá filtrá-la de acordo com o prefixo, populando o *PlaceHolder* apenas com aquelas que iniciam com **alm**, **cmd** e **stat**. É devido a tal fato que algumas das associações por tabela ilustradas na Figura 26 apresentam o texto "Associação Para Exibir em Tela" e, conseqüentemente, são denominadas de associações do tipo *fake*, visto que

não atuam em nenhuma propriedade de *driver*, sendo utilizadas apenas para que as propriedades associadas a elas sejam exibidas no *PlaceHolder*.

Com o objetivo de tornar o processo mais eficiente, a lógica dos alarmes que atuam quando recebem o valor lógico 0 foi adicionada na macro de instanciação desenvolvida (mencionada anteriormente na Seção 4.2. Primeiramente, devem ser verificados quais pontos atuam em 0 no SCC. Em seguida, tais pontos são associados conforme descrito anteriormente: a propriedade com o prefixo **drv\_Em0** é associada com a *tag* do *driver* de comunicação e a propriedade com o prefixo **alm\_Em0** recebe a associação *fake*.

É válido ressaltar que, além dos alarmes, algumas propriedades de estado também atuam em 0. A lógica para esses pontos é a mesma dos alarmes e, portanto, o código de instanciação é capaz de associá-las da maneira correta.

### 4.3.3 Comando dos Equipamentos

Conforme mencionado anteriormente neste documento, o objetivo principal do SIMEA é simular as concentradoras de dados existentes em campo. Portanto, além de reproduzir a atuação de pontos de estado e alarmes dos equipamentos, o simulador deve retratar também o comportamento de pontos de comando (caso o equipamento seja comandável). Por exemplo, ao comandar o fechamento de um disjuntor no SCC sem que haja nenhum intertravamento ativo, o SIMEA deve, automaticamente, configurá-lo para o estado de fechado. Um intertravamento é uma condição ou um conjunto de condições que verificam se o comando pode ser enviado ao equipamento. De maneira geral, o comando de fechar um disjuntor só poderá ser executado caso o equipamento esteja em modo remoto, aberto e sem falha de comando, por exemplo.

Existem três formas de enviar comandos a um equipamento: por meio da reprodução de dados históricos armazenados no banco de dados; por meio de uma sequência de eventos de uma simulação; ou, ainda, mediante à atuação direta no *PlaceHolder*. O Quadro 3 mostra os equipamentos que foram modelados no SIMEA, conforme mencionado na Seção 4.2, e que possuem pontos de comando. Para esses equipamentos, a lógica de comandos foi desenvolvida e será nesta subseção.



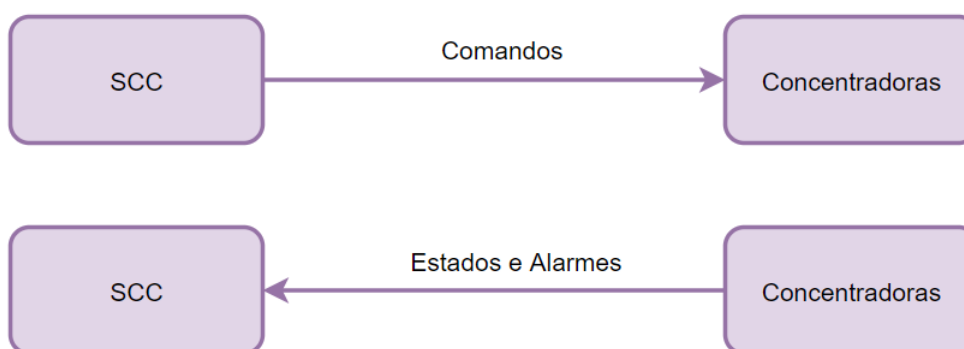
Quadro 3 – Equipamentos comandáveis.

Sistema do SCC	Equipamento
SCE	Contator Equalização
SEA	Botoeira Emergência
SEA	Botoeira Emergência Geral
SEA	Jato Ventilador
SEA	Painel Luz Bloco 4
SEA	Sistema Painéis Bloco 4
SEA	Veneziana SVTE
SEA	Ventilador SVTE
SFP	Elevador Hidráulico
SFP	Escada Rolante Bloco 4
SFP	Grupo Escada Rolante

Fonte – Elaborado pela autora (2020).

A Figura 29 ilustra o fluxo de dados entre o SCC e as concentradoras. Note que, para atuar em ponto de estado ou alarme de um equipamento, a atuação deve ser feita na concentradora de dados e o sinal deve chegar no SCC. Por outro lado, pontos de comando possuem o comportamento oposto, ou seja, a atuação deve ser realizada no SCC e o sinal deve chegar até a concentradora de dados.

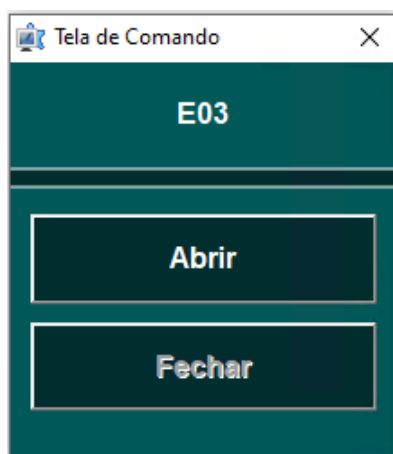
Figura 29 – Fluxo de dados entre o SCC e as concentradoras.



Fonte: Elaborado pela autora (2020).

Todos os equipamentos comandáveis possuem uma tela modal de comando associada a eles no SCC, conforme exemplificado na Figura 30. Assim, ao clicar no botão de **Abrir** ou no botão de **Fechar**, o comando de abrir/fechar o equipamento deve chegar na concentradora de dados.

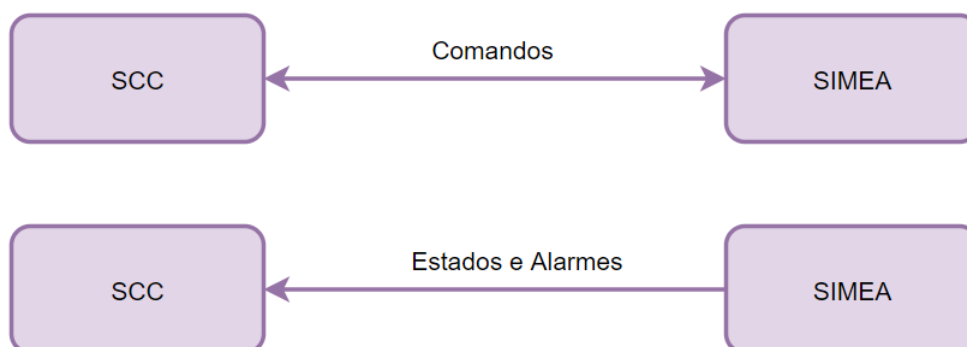
Figura 30 – Exemplo de uma tela modal de comando de um equipamento.



Fonte: Elaborado pela autora (2020).

Uma vez que o SIMEA deve simular as concentradoras de dados, o fluxo de dados entre o SCC e o simulador deveria ser tal qual mostrado na Figura 29. Porém, como o SIMEA também permite que sejam criadas simulações em horários pré-determinados pelo usuário, ele também deve ser capaz de enviar comandos ao SCC. Por exemplo, caso o usuário configure uma simulação em que ele deseje comandar todos os disjuntores de determinada estação para o estado de fechado, o SIMEA deve enviar um comando ao SCC e, em seguida, configurar os disjuntores para o estado de fechado (caso não haja nenhum intertravamento ativo). Portanto, a lógica de comandos dos equipamentos foi desenvolvida para contemplar tanto os comandos enviados do SCC ao SIMEA, quanto os comandos enviados do SIMEA ao SCC. A Figura 31 ilustra o fluxo de dados entre o SCC e o SIMEA.

Figura 31 – Fluxo de dados entre o SCC e o SIMEA.



Fonte: Elaborado pela autora (2020).

Para explicar a lógica de comando desenvolvida, será utilizado o equipamento Botoeira Emergência Geral como exemplo. Porém, é válido ressaltar que a lógica é semelhante para o restante dos equipamentos.

Para enviar comandos do SIMEA ao SCC, pode-se configurar uma sequência de eventos de uma simulação ou atuar nas propriedades de comando diretamente pelo Placeholder. A Figura 32 ilustra o *Placeholder* populado com as propriedades do equipamento Botoeira Emergência Geral. A caixa de seleção do estado **Automático** está selecionada, indicando que o equipamento está no modo automático. Quando o usuário clicar na caixa de seleção **Comando Manual**, o SIMEA deverá deselegionar a caixa de seleção referente ao estado automático e selecionar a referente ao estado manual.

Figura 32 – *Placeholder* com as propriedades do equipamento Botoeira Emergência Geral.

The image shows a configuration window for the 'Botoeira Emergência Geral' equipment. It features a dark teal background with white text and controls. At the top, there are two buttons labeled 'LTR' and 'SEA'. Below them is a larger button labeled 'SVP'. The equipment name 'Botoeira Emergência Geral' is displayed in a white box. The configuration is divided into three main sections: 'COMPORTAMENTO' (Behavior), 'COMANDOS' (Commands), and 'ESTADOS' (States). In the 'COMPORTAMENTO' section, there is a checkbox for 'Falha de Comando' which is unchecked, and a spinner control for 'Atraso dos comandos (s)' set to the value '0'. In the 'COMANDOS' section, there are two checkboxes: 'Comando Automático' (unchecked) and 'Comando Manual' (unchecked). In the 'ESTADOS' section, there are two checkboxes: 'Automático' (checked) and 'Manual' (unchecked).

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Após o envio do comando pelo operador do SIMEA, o registro de eventos do SCC deve registrar os eventos de Comando Manual Enviado e Comando Manual Iniciado. Caso não haja nenhum intertravamento ativo, o SIMEA deve, automaticamente, configurar o equipamento para o estado Manual, normalizar o estado Automático e gerar um evento de Comando Manual Executado, conforme mostrado na Figura 33.

Figura 33 – Registro de eventos do SCC após o comando Manual ser enviado ao equipamento.

Data Registro	Estação	Sistema	Subsistema	Equipamento	Operador ▾	Rec	Mensagem
14/11/2020 16:37:57	LTR	SEA	SVP	Botoeira de Emergência Geral	System		Automático normalizado
14/11/2020 16:37:57	LTR	SEA	SVP	Botoeira de Emergência Geral	System		Comando manual executado
14/11/2020 16:37:57	LTR	SEA	SVP	Botoeira de Emergência Geral	System		Manual
14/11/2020 16:37:56	LTR	SEA	SVP	Botoeira de Emergência Geral	System		Comando manual iniciado
14/11/2020 16:37:56	LTR	SEA	SVP	Botoeira de Emergência Geral	SIMEA	X	Comando manual enviado

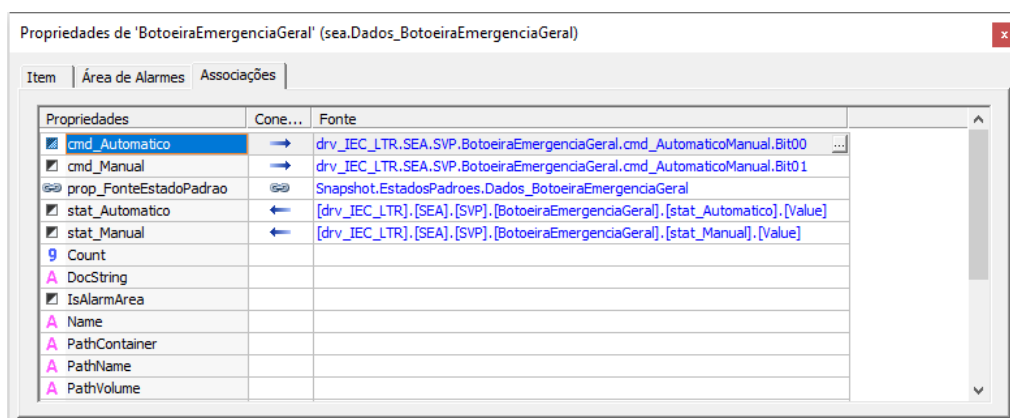
Fonte: Simulador de Equipamentos e Auxiliares (2020).

Na classe do equipamento Botoeira Emergência foram criados dois *scripts* do tipo *On Property Change* associados com as propriedades `cmd_Automático` e `cmd_Manual`. Como mencionado anteriormente na Subseção 4.3.2, esse tipo de *script* é disparado sempre que a propriedade associada a ele tiver seu valor alterado. Sendo assim, quando o usuário clicar na caixa de seleção Comando Manual, o *script* associado com a propriedade `cmd_Manual` será executado. De forma semelhante, quando o usuário clicar na caixa de seleção Comando Automático, o *script* associado com a propriedade `cmd_Automático` será executado.

Um *script* do tipo *On Property Change* foi criado na propriedade `cmd_Manual`. Assim, caso não exista nenhum intertravamento ativo, o SIMEA irá, por meio do domínio remoto, configurar as propriedades `cmd_Manual` e `cmd_Automático` do SCC para *true* e *false*, respectivamente.

A Figura 34 ilustra como são feitas as associações no objeto de dados do equipamento Botoeira Emergência Geral no SCC. Note que os pontos de comando possuem uma associação reversa com as *tags* de *driver* e, conseqüentemente, quando o SIMEA atua nas propriedades `cmd_Manual` e `cmd_Automático`, as propriedades de *driver* também são atuadas.

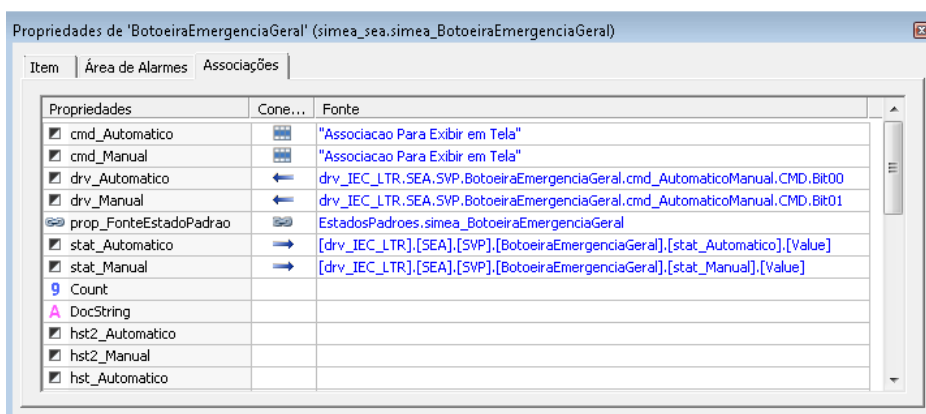
Figura 34 – Associações no objeto de dados do equipamento Botoeira Emergência Geral no SCC.



Fonte: Sistema de Controle Centralizado (2020).

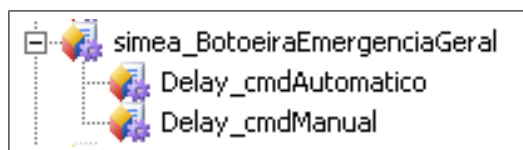
A Figura 35, por sua vez, ilustra como são realizadas as associações no objeto de dados do equipamento Botoeira Emergência Geral no SIMEA. Note que as propriedades com o prefixo **drv** possuem uma associação simples com as *tags* de *driver*. Sendo assim, quando o SCC atua nessas *tags* de *driver* as propriedades do SIMEA também são atuadas.

Figura 35 – Associações no objeto de dados do equipamento Botoeira Emergência Geral no SIMEA.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

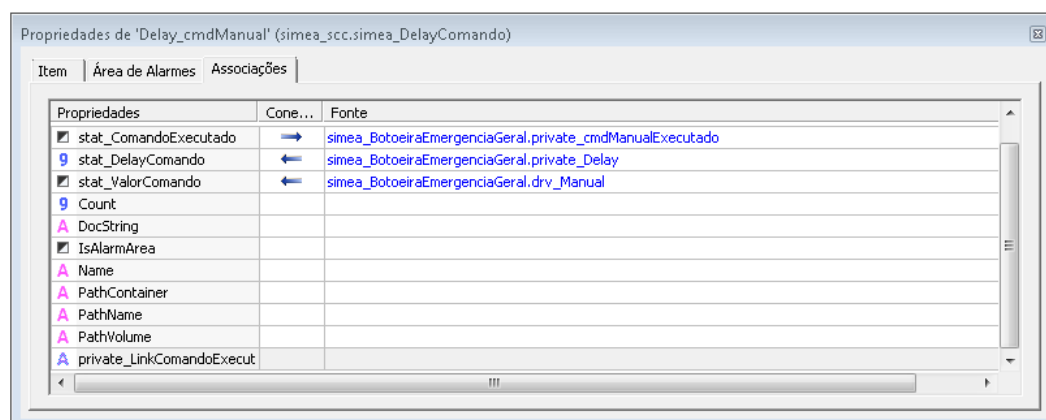
Todos os *XObjects* dos equipamentos comandáveis possuem uma instância do objeto *Delay* correspondente a cada um dos comandos do equipamento. Por exemplo, a Botoeira Emergência Geral dispõe de comandos de manual e automático, portanto a classe do equipamento deve possuir um *delay* para o comando de manual e outro para o comando de automático, conforme pode ser visualizado na Figura 36.

Figura 36 – Instâncias do *XObject Delay* na classe da Botoeira Emergência Geral.

Fonte: Simulador de Equipamentos e Auxiliares (2020).

Os objetos do tipo *Delay* permitem que o tempo de atraso dos comandos seja configurável. Assim, caso o usuário configure um tempo de atraso de 5 segundos, o SIMEA executará o comando somente quando esse período de tempo for atingido.

A Figura 37 ilustra as associações para a instância do objeto *Delay* do comando manual. Note que a propriedade `stat_ValorComando` possui uma associação simples com a propriedade `drv_Manual`, que por sua vez está associada com a *tag* de *driver* correspondente (conforme mostrado anteriormente na Figura 35). Assim, uma vez que a *tag* de *driver* for atuada, a atuação será transmitida para as propriedade `drv_Manual` e `stat_ValorComando`, respectivamente.

Figura 37 – Associações no objeto de dados do objeto *Delay* para o comando manual da Botoeira de Emergência Geral.

Fonte: Simulador de Equipamentos e Auxiliares (2020).

A propriedade `stat_ValorComando` contém um *script* do tipo *On Property Change* associado a ela. Dessa forma, quando o valor lógico da propriedade for verdadeiro, o *script* irá atuar na propriedade `stat_ComandoExecutado`, assim que o tempo de atraso configurado para os comandos for atingido.

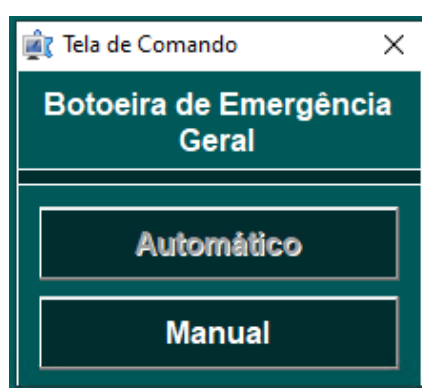
A propriedade `stat_ComandoExecutado`, por sua vez, possui uma associação reversa com a propriedade `private_cmdManualExecutado` (conforme mostrado anteri-

ormente na Figura 37). Logo, quando o valor lógico de `stat_ComandoExecutado` for verdadeiro, a propriedade `private_cmdManualExecutado` será atuada.

Assim, caso o equipamento não esteja em falha de comando, o *script* do tipo *On Property Change* existente na propriedade `private_cmdManualExecutado` irá configurar o estado manual da Botoeira Emergência Geral para verdadeiro e o estado automático para falso. Conseqüentemente, essa mudança de estados será transmitida ao SCC que, por sua vez, irá registrar o evento de comando executado no registro de eventos.

A lógica dos comandos é semelhante quando os comandos são enviados do SCC ao SIMEA. Na Figura 38 pode ser visualizada a modal de comando para a Botoeira de Emergência Geral. O botão Automático está intertravado indicando que o equipamento está no modo automático. Caso o usuário clique no botão Manual e não exista nenhum intertravamento ativo, o SIMEA irá configurar o equipamento para o modo manual seguindo a mesma metodologia descrita anteriormente.

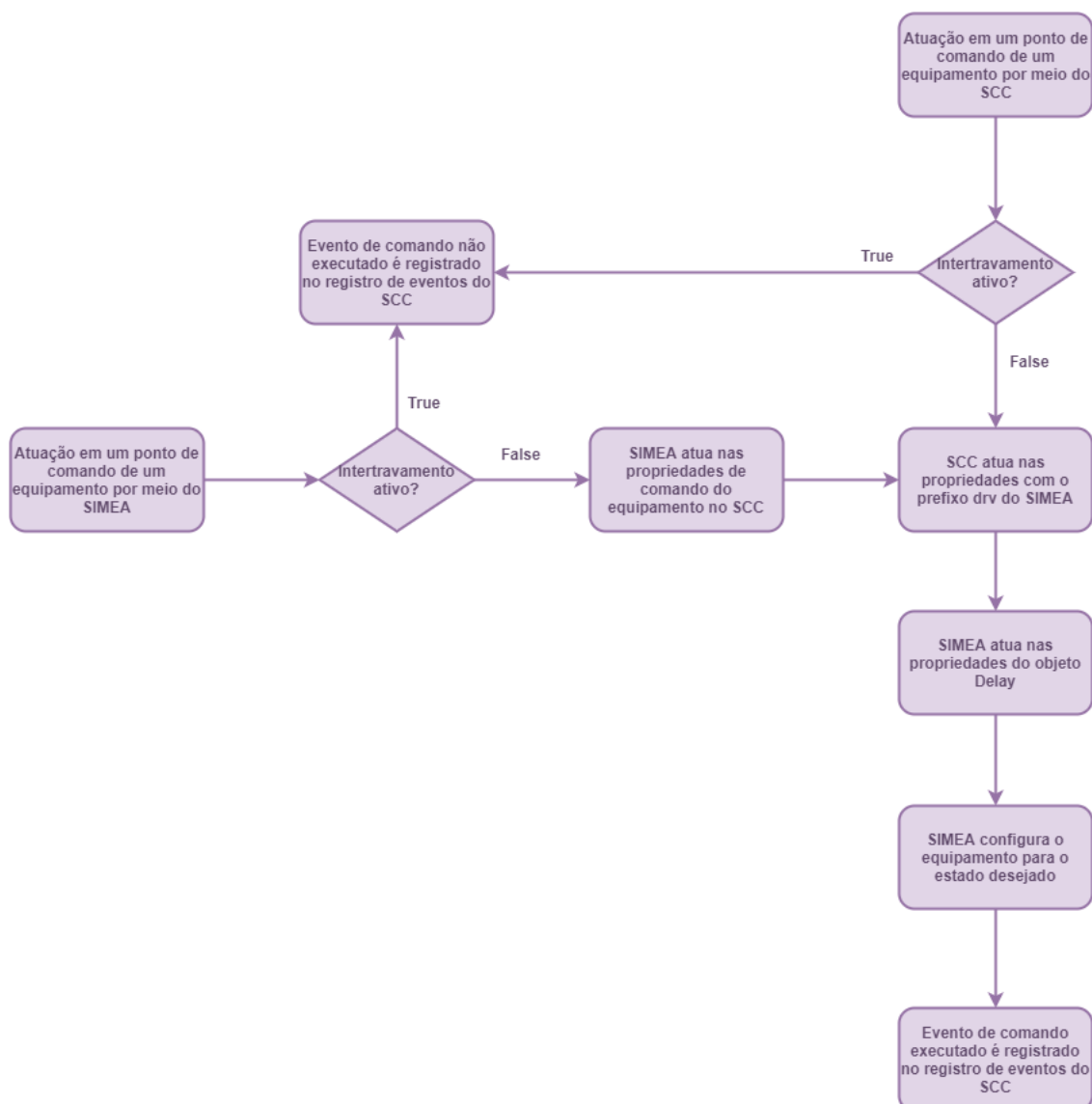
Figura 38 – Tela modal de comando da Botoeira de Emergência Geral no SCC.



Fonte: Simulador de Equipamentos e Auxiliares (2020).

O fluxograma da Figura 39 ilustra, de maneira geral, o funcionamento descrito para a lógica dos comandos no SIMEA.

Figura 39 – Fluxo dos comandos no SIMEA.



Fonte: Elaborado pela autora (2020).

Neste capítulo foi apresentada a metodologia de projeto utilizada, bem como a contribuição da aluna ao projeto, evidenciando o desenvolvimento necessário para atingir os objetivos específicos elencados na Seção 1.2, tais como a modelagem dos equipamentos e instanciação das estações, bem como as lógicas desenvolvidas para o funcionamento correto dos equipamentos. É válido ressaltar que os códigos desenvolvidos não foram apresentados no documento uma vez que possuem informações sigilosas. No próximo capítulo serão apresentados os resultados do Teste de Aceitação em Fábrica (TAF) do simulador.



## 5 TESTE DE ACEITAÇÃO EM FÁBRICA E RESULTADOS

Neste capítulo é apresentada a metodologia utilizada no Teste de Aceitação em Fábrica do simulador, bem como os resultados obtidos ao final do projeto.

### 5.1 TESTE DE ACEITAÇÃO EM FÁBRICA (TAF)

O Teste de Aceitação em Fábrica (TAF) é o processo de validação para confirmação da integridade e documentação do sistema de acordo com o que foi encomendado. Serve também para validar a funcionalidade e performance do sistema. Antes do início do TAF, o sistema é construído, programado e testado de acordo com os requisitos específicos e toda documentação do sistema é finalizada. O TAF é, acima de tudo, uma validação de que o sistema está de acordo com o encomendado, que funciona de acordo com as especificações e que está pronto para ser recebido pelo contratante (BARBOSA, 2020).

Realizar o TAF é de extrema importância, visto que permite a detecção de erros de funcionalidade e operação do sistema. O TAF é realizado por um membro da equipe de desenvolvimento e supervisionado por um representante do cliente. Uma prática comum na empresa é a de realizar o Teste de Aceitação em Fábrica Interno, ou seja, o sistema é testado internamente por membros da equipe antes de ser testado em produção. Tal fato resulta em um ganho de performance para o sistema, uma vez que reduz o número de erros detectados em campo.



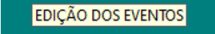


Para realizar o TAF do SIMEA é necessário seguir o Procedimento de Teste (PT) do simulador. O procedimento é formado por folhas de teste independentes compostas por sequência de casos de teste. Se o sistema apresentou o comportamento descrito na PT após a execução de cada caso de teste, é dito que o caso de teste foi aprovado. Do contrário, o caso de teste é reprovado e uma pendência é criada. É válido ressaltar que a reprovação de um caso de teste não impede a realização e a avaliação de outro. O Procedimento de Teste é composto por 16 folhas de teste ao todo, sendo elas:

1. Permissão e administração de usuários;
2. Navegação entre telas;
3. Criação de cenários;
4. Criação de sequência de eventos;
5. Criação de simulações;
6. Menu esquerdo – Seleção de equipamentos;
7. Menu esquerdo – Edição de cenários;

8. Menu esquerdo – Edição de sequências de eventos;
9. Menu esquerdo – Gerenciamento da simulação;
10. Menu esquerdo – Edição de simulações;
11. Edição de cenários;
12. Edição de sequências de eventos;
13. Edição de simulações;
14. Execução da simulação;
15. Comportamento dos objetos;
16. Comunicação entre SCC de treinamento e SIMEA.

A Figura 40 ilustra um exemplo do roteiro de teste para os casos de testes relacionados à navegação entre telas. Para cada etapa existe uma descrição, um resultado esperado e uma representação gráfica. Além do mais, cada caso de teste dispõe de uma coluna para indicar se o teste passou ou falhou, bem como de um espaço dedicado para observações.

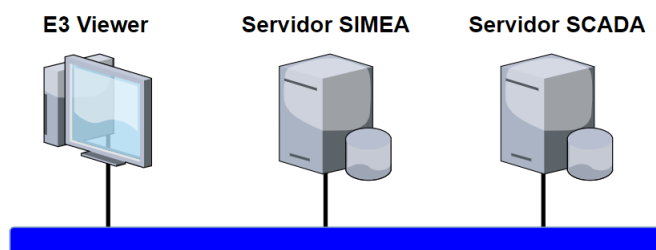
Figura 40 – Exemplos de casos de testes relacionados à navegação entre telas.

Etapa	Descrição	Resultado Esperado	Representação Gráfica	Passou	Falhou	Obs
2.1	Realize login com o usuário "InstrutorAvancado".	Exibição de nova tela com seis opções disponíveis no menu superior: "modelagem do ambiente", "edição dos cenários", "edição das sequências de eventos", "edição das simulações", "gerenciamento da simulação" e "logout". Inicia-se selecionado o botão "edição dos cenários".				
2.2	Posicione o cursor sobre o botão "edição dos eventos".	O botão muda imediatamente para a cor branca e o cursor é alterado para o formato de mão.				
2.3	Mantenha o cursor parado sobre o botão "edição dos eventos".	Exibição do tooltip "EDIÇÃO DOS EVENTOS" próximo ao botão.				
2.4	Remova o cursor de cima do botão "edição dos eventos".	O botão retorna para a cor original e o cursor para o formato normal.				
2.5	Clique sobre o botão "edição dos eventos".	O botão selecionado permanece na cor branca e a tela "EDIÇÃO DAS SEQUÊNCIAS DE EVENTOS" é exibida.				

Fonte: Simulador de Equipamentos e Auxiliares (2020).

A Figura 41 ilustra a arquitetura de rede que é utilizada no TAF, sendo que essa possui configurações semelhantes à configuração final de campo.

Figura 41 – Arquitetura de rede para o TAF.



Fonte: Elaborado pela autora (2020).

A primeira rodada de testes contemplam todos os casos de testes do PT. Na segunda rodada são testados somente aqueles que falharam na primeira e assim por diante. O resultado quantitativo da primeira rodada do TAF do SIMEA é apresentado na Tabela 1, mostrando o total de testes realizados, aprovados e também a quantidade de testes que falharam.

Tabela 1 – Resultados quantitativos da primeira rodada de teste.

<b>Testes</b>	<b>Quantidade</b>
Testes realizados	344
Testes aprovados	324
Testes reprovados	20

Fonte – Elaborado pela autora (2020).

O Quadro 4 ilustra algumas das pendências encontradas ao longo do TAF do simulador.

Quadro 4 – Exemplos de pendências geradas ao longo do TAF.

Caso de Teste	Descrição
8.7	Ao escolher o equipamento CPR - SCE - SBT - Medidor Quadro PQ11, no menu lateral esquerdo da tela de Edição dos Cenários, surgiu uma <i>pop up</i> de erro.
10.12	Comandos de abrir/fechar dos Disjuntores Tramo Comando não estão sendo executados.
13.5	Ao enviar comando de abrir para a seccionadora de Via estão sendo registrados eventos errados no registro de eventos do SCC.
14.8	O cursor não foi alterado para o formato de mão e o botão não foi destacado. Não é possível clicar sobre o botão.
15.15	Ao configurar uma simulação em que os primeiros eventos não possuem atraso, ou seja, são realizados no mesmo momento do Instante Inicial, o botão Avançar da Tela Simulação Resumo não funciona. Ele não avança para os próximos eventos nem executa os iniciais.
15.52	Ao selecionar/deselecionar a caixa de seleção do estado Automático do Quadro Elétrico QGD é registrado evento de comando automático executado normalizado e comando automático executado no registro de eventos.
15.52	As tags de comandos de bombas das estações CPR, CPL, VBE, GGR e LTR possuem um comportamento diferente das estações antigas e a lógica de comando atual não funciona. É preciso revisar a lógica.
15.52	Equipamento Sistemas Bombas de CPR, CPL, VBE, GGR e LTR possuem pontos de comando. É preciso implementar a lógica.

Fonte – Elaborado pela autora (2020).

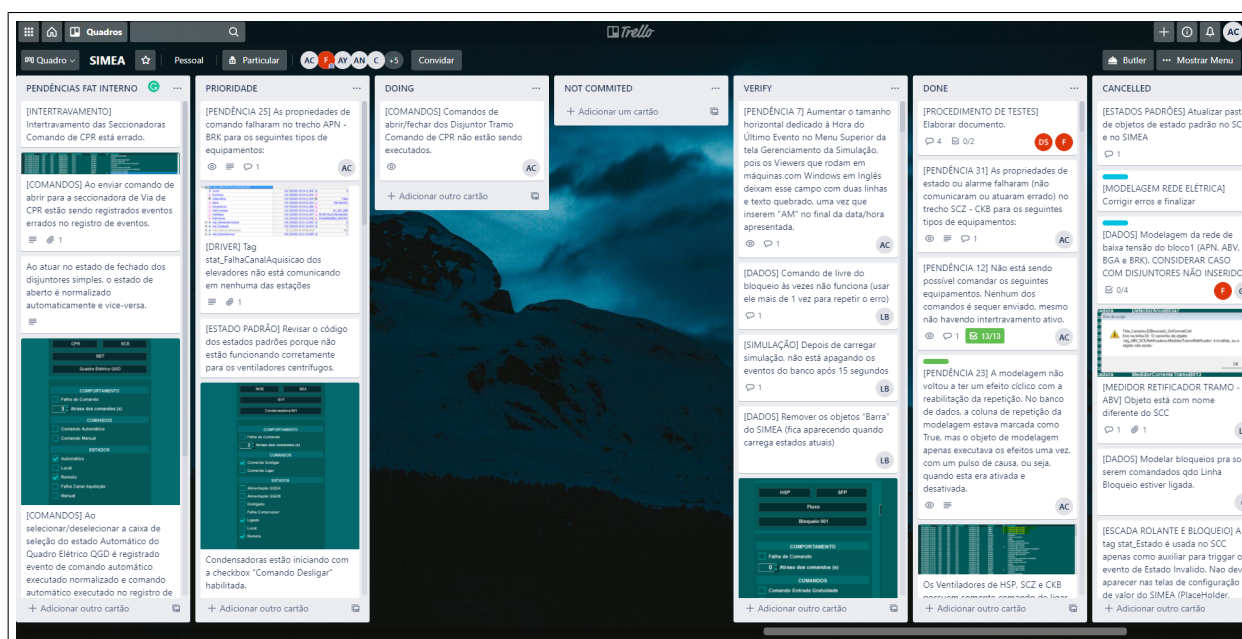
Ao final do TAF, as pendências foram listadas em forma de cartões no aplicativo de gerenciamento Trello <sup>1</sup> e, então, resolvidas. A Figura 42 ilustra um *screenshot* do quadro do SIMEA no Trello. Note que os cartões ficam organizados em forma de listas, sendo que cada lista possui uma função diferente. Por exemplo, as pendências gerados no teste interno são adicionadas na lista **Pendências FAT Interno**, enquanto que aquelas consideradas prioritárias, na lista **Prioridades**. Quando um membro da equipe está resolvendo uma pendência, o indicado é que ele adicione o cartão na lista **Doing** e, assim que terminar de resolvê-la, adicionar o cartão na lista **Verify**. O ideal é que os cartões da lista **Verify** sejam testados novamente por outro membro da equipe e, em seguida, adicionados à lista **Done**. Caso alguma pendência seja cancelada por

<sup>1</sup> Plataforma de colaboração Trello: [www.trello.com](http://www.trello.com)

algum motivo, o cartão deve ser adicionado na lista **Cancelled**.

Seguir as etapas descritas anteriormente é muito importante ao corrigir as pendências do sistema porque, uma vez que os cartões são testados e verificados por membros diferentes da equipe, as chances de erros adicionais são minimizadas.

Figura 42 – Trello do SIMEA.



Fonte: Arquivo Pessoal (2020).

## 5.2 RESULTADOS

O SIMEA contempla, agora, todas as 17 estações da Linha 5 do metrô de São Paulo. A comunicação com o SCC está funcionando conforme o esperado, a lógica dos alarmes e dos comandos dos equipamentos foi implementada e o Teste de Aceitação em Fábrica, realizado. As pendências encontradas ao longo do TAF do simulador foram adicionadas ao Trello do sistema e, em seguida, corrigidas. Os resultados obtidos ao final do Teste de Aceitação em Fábrica do simulador apontam que o desempenho do sistema foi satisfatório, uma vez que ele é capaz de replicar os sinais provenientes do campo de forma idêntica ao modelo real, respeitando as lógicas básicas de funcionamento do SCC. Sendo assim, pode-se afirmar que o objetivo geral e os específicos deste projeto de fim de curso foram alcançados.

## 6 CONCLUSÃO

Neste projeto de fim de curso foi apresentado o Simulador de Equipamentos e Auxiliares (SIMEA) do Sistema de Controle Centralizado (SCC) da Linha 5 do metrô de São Paulo. O SIMEA é responsável por simular as concentradoras de dados existentes em campo, sendo a ferramenta utilizada para simulação, treinamento e teste de estratégias operacionais dos sistemas SCE, SEA e SFP. Sendo assim, o simulador deve se comportar de forma similar ao campo, replicando lógicas de funcionamento e sendo capaz de reproduzir cenários programados e/ou acontecimentos armazenados na base histórica do SCC, possibilitando a análise de sequências de eventos da operação comercial.

O objetivo principal deste projeto de fim de curso era o de finalizar o desenvolvimento do SIMEA. No Capítulo 2, conceitos básicos de sistemas SCADA foram apresentados, juntamente com uma visão geral de funcionamento do software Elipse E3. As principais características do sistema supervisorio da Linha 5 do metrô de São Paulo também foram descritas nesse capítulo, sendo que uma breve descrição sobre cada um dos três grandes sistemas que compõe o SCC foi feita.

No Capítulo 3, as principais funcionalidades do SIMEA foram apresentadas. A comunicação entre o SIMEA e o SCC é feita por meio do protocolo de comunicação IEC104, sendo que o SCC atua como mestre e o SIMEA como escravo. Além do mais, os conceitos de cenários e eventos foram definidos neste capítulo, juntamente com a configuração de domínios remotos e as principais telas do simulador.

No Capítulo 4 foi apresentada a metodologia de projeto adotada e também a contribuição da aluna no projeto. Como mencionado anteriormente neste documento, embora muitas das funcionalidades do sistema já estivessem desenvolvidas, o SIMEA não contemplava todas as estações da Linha 5. Logo, fez-se necessário realizar um levantamento para saber quais estações precisavam ser instanciadas. Também houve a necessidade de verificar quais equipamentos não estavam modelados no simulador, uma vez que as estações possuem muitas divergências entre si e, conseqüentemente, possuem tipos de equipamentos diferentes.

O procedimento de instanciação das estações no SIMEA é altamente manual, visto que para cada uma das *tags* de *driver* existentes é necessário criar um *link* com a propriedade da instância correspondente. Portanto, foi desenvolvido um código capaz de automatizar o processo, permitindo a criação de mais de 3000 instâncias de objetos de forma automática e resultando em ganho de tempo e performance para o projeto. Além do mais, para garantir o correto funcionamento do SIMEA, lógicas internas foram desenvolvidas, tais como a configuração automática dos estados padrões, lógicas dos alarmes e lógicas dos comandos dos equipamentos.

A etapa final do projeto é descrita no Capítulo 5, onde foi apresentada a me-

metodologia utilizada no Teste de Aceitação em Fábrica do simulador, juntamente com os resultados obtidos. Portanto, pelo conteúdo exposto ao longo deste documento e os resultados obtidos com o Teste de Aceitação em Fábrica, pode-se concluir que o objetivo geral e os específicos foram atingidos. Ainda, é válido ressaltar o quanto o SIMEA agrega valor frente ao metrô de São Paulo, uma vez que o sistema desenvolvido permite:

- **Simular:** o operador pode, por exemplo, realizar o *playback* de uma situação de incidente na Linha 5, identificando a causa raiz de um problema, erros no processo e até mesmo falhas de operação humana;
- **Testar:** quando o SCC for modificado (devido a troca de algum equipamento, por exemplo), o SIMEA permitirá que testes sejam realizados antes de subir o sistema para a produção;
- **Treinar:** os operadores terão a possibilidade de utilizar o SIMEA como um ambiente de treinamento, sendo que um instrutor avaliará o desempenho individual de cada um. Assim, o número de acidentes pode ser minimizado, uma vez que o operador já terá conhecimentos suficientes para operar o SCC com segurança.

Como mencionado anteriormente no Capítulo 2, o SCC é responsável por monitorar, além das 17 estações de metrô, dois pátios e duas subestações de energia. Logo, a perspectiva de trabalho futuro é fazer com que o SIMEA contemple também as subestações de energia primária de Bandeirantes e Guido Caloi, bem como os pátios Capão Redondo e Guido Caloi.

Ainda, é válido ressaltar que os conhecimentos adquiridos nas disciplinas de Informática Industrial, Redes de Computadores para Automação e Informática para Automação foram essenciais no desenvolvimento do projeto de fim de curso.

Por fim, destaca-se que o período de estágio foi muito gratificante pois, além de proporcionar conhecimentos valiosos para a formação acadêmica da aluna, possibilitou o desenvolvimento e melhoria de habilidades pessoais, tais como proatividade e trabalho em equipe. Acima de tudo, ter a oportunidade de estagiar em uma empresa que incentiva o bom relacionamento entre os colaboradores, que valoriza as pessoas e que proporciona um ambiente de trabalho amigável é indescritível.

## REFERÊNCIAS

ABBAS H.; SHAHEEN, S. Simple, Flexible, and Interoperable SCADA System Based on Agent Technology, 2015.

BARBOSA, M. **Teste de Aceitação em Fábrica**. 2020. Disponível em: <https://public-library.safetyculture.io/products/teste-de-aceitacao-de-fabrica-taf>.

COELHO, M. S. **Apostila de Sistemas Supervisórios. Curso superior de tecnologia em automação e controle de processos industriais contínuos**. [S.l.]: Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Cubatão, 2010.

CONSTAIN, N.B.P. **Integração de sistemas SCADA com a implementação de controle supervisão em CLP para sistemas de manufatura**. 2011. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/95357/296291.pdf?sequence=1&isAllowed=y>.

CPTM, Companhia Paulista de Trens Metropolitanos. **Mapa das Estações**. 2019. Disponível em: <https://www.metrocptm.com.br/diagrama-linha5-abr19/>.

ELIPSE. **Elipse E3: Uma visão geral**. 2019. Disponível em: <https://kb.elipse.com.br/elipse-e3-uma-visao-geral/>.

ELIPSE. **Manual do Usuário do E3**. 2020. Disponível em: <https://www.elipse.com.br/downloads/?cat=19&key=manual&language=ptbr#header-main/>.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 60870-5-104: Telecontrol equipment and systems - Part 5-104: Transmission protocols**. [S.l.], 2006.

MORAES C.C.; CASTRUCCI, L.C. **Engenharia de Automação Industrial**. [S.l.]: Editora LTC, 2006.

ROSÁRIO, João Mauricio. **Automação Industrial**. [S.l.]: Baraúna, 2009. (1ª Edição).

SALVADOR, M. **Como o Elipse E3 funciona**. 2019. Disponível em: <https://kb.elipse.com.br/como-o-elipse-e3-funciona/>.



SANTELLA, José. **Metodologia de Projetos**. 2019. Disponível em:  
<https://www.euax.com.br/2020/07/metodologia-de-projetos/>.

SOUZA, R.B. **Uma Arquitetura para Sistema Supervisórios Industriais e sua Aplicação em Processos de Elevação Artificial de Petróleo**. 2005. Disponível em:  
<https://repositorio.ufrn.br/jspui/bitstream/123456789/15444/1/RodrigoBS.pdf>.