



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**JOSÉ AIRTON BORGES CARNEIRO**

**PROPOSTA DE METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETO**  
**PARA APLICAÇÃO NA DISCIPLINA DE SUPERVISÃO E CONTROLE DE**  
**SISTEMAS ELÉTRICOS DE POTÊNCIA**

**FORTALEZA**

**2021**

JOSÉ AIRTON BORGES CARNEIRO

PROPOSTA DE METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETO PARA  
APLICAÇÃO NA DISCIPLINA DE SUPERVISÃO E CONTROLE DE SISTEMAS  
ELÉTRICOS DE POTÊNCIA

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Ceará como parte dos requisitos para a obtenção de Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Raimundo Furtado Sampaio.

Coorientador: Eng. Claudivan Domingos de Freitas.

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

C289p Carneiro, José Airton Borges.

Proposta de metodologia de aprendizagem baseada em projeto para aplicação na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência / José Airton Borges Carneiro. – 2021.  
100 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2021.

Orientação: Prof. Dr. Raimundo Furtado Sampaio.

Coorientação: Prof. Claudivan Domingos de Freitas.

1. Metodologia de Aprendizagem Baseada em Projeto. 2. Sistema SCADA. 3. ScadaBR. 4. Supervisão e Controle de Sistemas Elétricos. I. Título.

CDD 621.3

---

JOSÉ AIRTON BORGES CARNEIRO

PROPOSTA DE METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETO PARA  
APLICAÇÃO NA DISCIPLINA DE SUPERVISÃO E CONTROLE DE SISTEMAS  
ELÉTRICOS DE POTÊNCIA

Trabalho de Conclusão de Curso apresentado  
ao Departamento de Engenharia Elétrica da  
Universidade Federal do Ceará como parte dos  
requisitos para a obtenção de Graduação em  
Engenharia Elétrica.

Aprovada em: \_\_\_/\_\_\_/\_\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Raimundo Furtado Sampaio (Orientador)  
Universidade Federal do Ceará (UFC)

---

Eng. Claudivan Domingos de Freitas (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Profª. Ph.D. Ruth Pastôra Saraiva Leão  
Universidade Federal do Ceará (UFC)

Ao meu pai Wader Vieira,  
À minha avó Eliane Oliveira,  
À minha família e amigos.

## AGRADECIMENTOS

Ao meu avô José Borges, pois jamais teria conseguido chegar aqui sem seu apoio, esforço, ensinamentos e amor dados a mim desde que cheguei nesse mundo.

Ao meu pai Wader Vieira e à minha avó Eliane Oliveira por todo amor e dedicação. Mesmo não estando mais presentes fisicamente na Terra, sei que torcem por mim e sempre me desejam o melhor de onde quer que estejam.

À minha mãe Nadir Oliveira por sempre ter acreditado em mim, e por todo apoio e amor.

Aos meus avós Airton e Socorro, e meu tio Wátilla por sempre torcerem por mim e me amarem mesmo eu sendo distante.

Ao meu tio Neto e, especialmente, ao meu tio Nigel por ser até hoje um exemplo de pessoa que eu almejo ser um dia.

À AGUFC (Bruno Gomes, Caio “Prescott”, Leonardo Andrade, Matheus Alencar, Matheus Torres, Pedro Tino e Vanderson), grupo do qual faço parte, por terem sido meus parceiros e amigos durante toda minha jornada na UFC. Sem vocês, eu jamais teria conseguido me formar.

Aos meus amigos de faculdade José Rafael, Gabriel Marçal, Leonardo Milfont, Pedro Victor, Fernando Vasconcelos, Melissa Rebouças, Guilherme Bezerra, Davi Gomes, Daniel Akio, Vitor Arantes, Nathalia Drussila (e tantos outros que com certeza devo ter esquecido) por toda amizade, companheirismo e ajuda durante o curso.

Ao grupo Sonoplastas (Teuzim, Haroldo, Sacul, Tino, Marçal, Thiago, Danilo, João Lucas e Milfont) por terem me proporcionado ótimas resenhas ao longo da faculdade.

Ao grupo “TCC agora vai!” (Bruno Gomes, Matheus Torres, Guilherme Bezerra, Igor Prata e Bruno Vasconcelos) pelos incentivos mútuos e reuniões para elaboração do TCC.

À Tecsys Jr. por ter me dado grandes experiências profissionais e práticas no curso.

Ao meu amigo Edyvalberty Alenquer pela amizade e por ter me ajudado na confecção de alguns códigos nesse trabalho.

À minha amiga Márlya Evelyn por toda amizade e incentivo de toda manhã para eu fazer o TCC.

Aos meus supervisores de estágio na ARCE Filipe e Gleyson pela confiança. Além de, muito especialmente, meus supervisores Eugênio Bittencourt e Daniela Cambraia por toda confiança e ensinamentos dados a mim, além de serem um exemplo do que pretendo ser profissionalmente.

À toda a equipe da ARCE, em especial, Thais, Paula, Aurélia, Morgana, Aliene, Ingrid, Conceição, Bia, Rosa e Renata. Vocês tornavam minhas manhãs muito mais agradáveis.

À Carol, Carla, Cacilda e Carlos pelo apoio e por terem sido uma segunda família pra mim durante boa parte da faculdade.

Aos meu orientadores do TCC Prof. Dr. Raimundo e Eng. Claudivan por todo apoio, paciência e ajuda na elaboração e revisão do meu trabalho.

À Profª Ph.D. Ruth por ter aceitado fazer parte da banca, e pelas contribuições e sugestões.

“All you need is love.” (John Lennon/Paul McCartney).



## RESUMO

Nos últimos anos, ocorreram mudanças e evoluções significativas no setor elétrico, fazendo que os professores de Engenharia Elétrica sintam a necessidade de implementar novas metodologias de ensino. Nesse contexto, desenvolveu-se uma Metodologia de Aprendizagem Baseada em Projeto (ABP) para construção de um Sistema de Aquisição e Controle de Dados (SCADA), e apresentou-se uma aplicação dessa metodologia como uma forma de ensino prático na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência do curso de Engenharia Elétrica da Universidade Federal do Ceará (UFC) com uso do *software* ScadaBR. Ademais, o desenvolvimento de um sistema SCADA é uma ferramenta de ensino prático extremamente importante na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência. Entretanto, os *softwares* supervisórios são caros, e suas versões de demonstração são limitadas, o que dificulta a utilização dessas plataformas por alunos de graduação. Nesse contexto, o ScadaBR se mostra uma ferramenta aplicável no ensino de Sistema SCADA, visto que é um *software* livre, gratuito e de fácil uso, possuindo os recursos para utilização no desenvolvimento de um sistema supervisório. Neste trabalho, foi implementado um sistema supervisório aplicado a uma subestação, com a utilização do ScadaBR, do protocolo de comunicação *Modbus* e do simulador *Eclipse Modbus Simulator* seguindo-se a metodologia ABP desenvolvida. Nesse sistema supervisório, foram definidos os pontos analógicos e digitais a serem monitorados e/ou comandados, o tratamento lógico dos dados, e desenvolvidas telas de supervisão. Este trabalho permite mostrar que o sistema SCADA desenvolvido apresentou bom funcionamento, e que *software* ScadaBR apresenta recursos que possibilitam sua utilização em subestações e no ensino de Sistema SCADA. Além disso, o desenvolvimento da metodologia ABP se mostrou como uma ferramenta extremamente útil para o ensino prático de Sistema SCADA na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência, pois poderá estimular o engajamento e criatividade dos alunos.

**Palavras-chave:** Metodologia de Aprendizagem Baseada em Projeto. Sistema SCADA. ScadaBR. Supervisão e Controle de Sistemas Elétricos.

## ABSTRACT

In the last years, there have been significant changes and evolutions in the electric sector, making Electrical Engineering teachers feel the need to implement new teaching methodologies. In this context, a Project-Based Learning Methodology (PBL) was developed to build a Data Acquisition and Control System (SCADA), and an application of this methodology was presented as a form of practical teaching in the discipline of Supervision and Control of Electrical Power Systems in the Electrical Engineering course at the Federal University of Ceará (UFC) using the ScadaBR software. Furthermore, the development of a SCADA system is an extremely important practical teaching tool in the discipline of Supervision and Control of Electric Power Systems. However, supervisory software is expensive, and its demo versions are limited, making it difficult for undergraduate students to use these platforms. In this context, ScadaBR proves to be an applicable tool in the teaching of the SCADA system, since it is open-source, free and easy to use software, having the resources to be used in the development of a supervisory system. In this work, a supervisory system applied to a substation was implemented, using ScadaBR, the Modbus communication protocol and the Elipse Modbus Simulator, following the developed PBL methodology. In this supervisory system, the analog and digital points to be monitored and/or commanded, the logical treatment of the data, and supervision screens were developed. This work also show that the developed SCADA system presented good functioning, and that ScadaBR software has resources that allow its use in substations and in the teaching of SCADA System. In addition, the development of the PBL methodology proved to be an extremely useful tool for the practical teaching of the SCADA System in the discipline of Supervision and Control of Electric Power Systems, as it may stimulate students' engagement and creativity.

**Keywords:** Project-Based Learning. SCADA. ScadaBR. Supervision and Control of Electrical Systems.

## LISTA DE FIGURAS

Figura 1 - Hierarquia de um sistema de automação de subestação .....	19
Figura 2 - Relé de sobrecorrente eletromecânico de indução.....	21
Figura 3 - Relé digital.....	22
Figura 4 - Metodologia de aprendizagem baseada em projeto .....	27
Figura 5 - Metodologia de aprendizagem baseada em projeto desenvolvida.....	31
Figura 6 - Quadro geral do <i>Modbus</i> .....	34
Figura 7 - Transação <i>Modbus</i> sem erros .....	35
Figura 8 - Transação <i>Modbus</i> com erros .....	35
Figura 9 - Diagrama unifilar da subestação <i>Campus</i> do Pici .....	38
Figura 10 - Tela do programa <i>Elipse Modbus Simulator</i> .....	44
Figura 11 - Ajustes efetuados no simulador .....	44
Figura 12 - Simulador disponível para comunicação .....	45
Figura 13 - Servidor <i>Apache Tomcat</i> .....	46
Figura 14 - Tela inicial do <i>ScadaBR</i> .....	47
Figura 15 - Barra de ferramentas do <i>ScadaBR</i> ( <i>data source</i> em destaque).....	47
Figura 16 - Tela para visualização e adição do <i>data source</i> .....	47
Figura 17 - Ajustes das propriedades do <i>Modbus IP</i> do primeiro dispositivo no <i>ScadaBR</i> .....	48
Figura 18 - Ajustes das propriedades do <i>Modbus IP</i> do segundo dispositivo no <i>ScadaBR</i> .....	49
Figura 19 - Habilitação dos <i>data sources</i> .....	49
Figura 20 - Tela de <i>data source</i> (adição de <i>data points</i> ) .....	50
Figura 21 - Adição de <i>data points</i> .....	51
Figura 22 - Detalhes do <i>data point</i> .....	51
Figura 23 - Definição do <i>data point</i> Estado do Disjuntor (11T1) .....	52
Figura 24 - Definição do <i>data point</i> Estado do Disjuntor (12T1).....	53
Figura 25 - Definição do <i>data point</i> Ajuste 1 (11T1) .....	53
Figura 26 - Definição do <i>data point</i> Ajuste 3 (11T1) .....	54
Figura 27 - Definição do <i>data point</i> Ajuste 2 (11T1) .....	54
Figura 28 - Definição do <i>data point</i> Ajuste 1 (12T1).....	55
Figura 29 - Definição do <i>data point</i> Ajuste 2 (12T1).....	55
Figura 30 - Definição do <i>data point</i> Ajuste 3 (12T1).....	56
Figura 31 - Definição do <i>data point</i> Abertura do Disjuntor (11T1) .....	56
Figura 32 - Definição do <i>data point</i> Fechamento do Disjuntor (11T1).....	57

Figura 33 - Definição do <i>data point</i> Abertura do Disjuntor (12T1) .....	57
Figura 34 - Definição do <i>data point</i> Fechamento do Disjuntor (12T1) .....	58
Figura 35 - Definição do <i>data point</i> Nível de Tensão (01B1) .....	58
Figura 36 - Definição do <i>data point</i> Nível de Tensão (02B1) .....	59
Figura 37 - Habilitação dos <i>data points</i> .....	59
Figura 38 - Acesso ao código "dpid" do <i>data point</i> .....	60
Figura 39 - Exibição dos <i>data points</i> .....	60
Figura 40 - Tela de propriedades do <i>data point</i> .....	61
Figura 41 - Ícone para exibição da tela de <i>watch list</i> .....	61
Figura 42 - Tela <i>watch list</i> .....	62
Figura 43 - Barra de ferramentas da <i>watch list</i> .....	62
Figura 44 - Inserção dos <i>data points</i> na <i>watch list</i> .....	62
Figura 45 - <i>Watch list</i> Relé 1 .....	63
Figura 46 - <i>Watch list</i> Relé 2 .....	63
Figura 47 - Definição de valor para escrita .....	63
Figura 48 - Definição e geração do gráfico na <i>watch list</i> .....	64
Figura 49 - Barra de ferramentas do ScadaBR ( <i>scripting</i> ).....	64
Figura 50 - Tela de adição de um <i>script</i> .....	65
Figura 51 - Detectores de Eventos.....	66
Figura 52 - Barra de ferramentas do ScadaBR (tratador de eventos).....	67
Figura 53 - Tratadores de eventos.....	67
Figura 54 - Fluxograma para mudança de grupo de ajuste .....	68
Figura 55 - Definição do detector de evento para mudança de grupo de ajuste.....	68
Figura 56 - Tratador de evento para mudança de ajuste .....	69
Figura 57 - Fluxograma de abertura e fechamento do disjuntor .....	69
Figura 58 - Detectores de eventos associados ao <i>data point</i> Abertura do Disjuntor (11T1) ....	70
Figura 59 - Tratador de evento para abertura do disjuntor (estado do disjuntor) .....	71
Figura 60 - Tratador de evento para abertura do disjuntor (pulso do comando) .....	71
Figura 61 - Tratador de evento para fechamento do disjuntor.....	72
Figura 62 - Tratador de evento para fechamento do disjuntor.....	72
Figura 63 - Definição de alarme para abertura de disjuntor .....	73
Figura 64 - Alarmes gerados pelo ScadaBR.....	73
Figura 65 - Barra de ferramentas do ScadaBR (representações gráficas).....	74
Figura 66 - Representações gráficas .....	74

Figura 67 - Propriedades da tela.....	74
Figura 68 - Imagem de fundo (unifilar).....	75
Figura 69 - Definição do código HTML .....	76
Figura 70 - Definição do <i>script</i> para servidor .....	77
Figura 71 - Imagem de fundo (monitoramento de tensão) .....	77
Figura 72 - Definição do gráfico para o Nível de Tensão (01B1) .....	78
Figura 73 - Imagem de fundo (alarmes) .....	79
Figura 74 - Definição da Lista de Alarmes.....	80
Figura 75 - Resultados para Mudança de Grupo de Ajuste (11T1) .....	80
Figura 76 - Resultados para mudança de grupo de ajuste (12T1) .....	81
Figura 77 - Resultados de abertura e fechamento do disjuntor (Relé 1) .....	82
Figura 78 - Resultados de abertura e fechamento do disjuntor (Relé 2) .....	82
Figura 79 - Resultado da tela de supervisão (unifilar).....	83
Figura 80 - Tela de supervisão (unifilar) .....	84
Figura 81 - Resultado da tela de supervisão (monitoramento de tensão).....	85
Figura 82 - Resultado da tela de supervisão (alarmes).....	86

## LISTA DE TABELAS

Tabela 1 - Descrição dos produtos e prazos de cada etapa .....	30
Tabela 2 - Descrição dos pontos a serem avaliados de cada produto .....	31
Tabela 3 - Modelo de dados do protocolo <i>Modbus</i> .....	35

## LISTA DE ABREVIATURAS E SIGLAS

ABP	Aprendizagem Baseada em Projeto
CLP	Controlador Lógico Programável
COS	Centro de Operação do Sistema
COR	Centro de Operação Regional
DEC	Duração equivalente de interrupção por consumidor
DNP	<i>Distributed Network Protocol</i>
FEC	Frequência equivalente de interrupção por consumidor
FTP	<i>File Transfer Protocols</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEC	<i>International Electrotechnical Commission</i>
IED	<i>Intelligent Electronic Device</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IP	<i>Internet Protocol</i>
OSI	<i>Open System Interconnection</i>
PDU	Unidade de Dados de Protocolo
SAS	Sistema de Automação de Subestação
SCADA	<i>Supervisory Control and Data Acquisition</i>
SEP	Sistema Elétrico de Potência
TC	<i>Technical Committee</i>
TC	Transformador de Corrente
TCP	<i>Transmission Control Protocol</i>
TP	Transformador de Potencial
UFC	Universidade Federal do Ceará
UTR	Unidade Terminal Remota

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	14
<b>1.1</b>	<b>Problemática e justificativa</b> .....	14
<b>1.2</b>	<b>Objetivos</b> .....	15
<i>1.2.1</i>	<i>Objetivo geral</i> .....	15
<i>1.2.2</i>	<i>Objetivos específicos</i> .....	15
<b>1.3</b>	<b>Estrutura do trabalho</b> .....	16
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	17
<b>2.1</b>	<b>Introdução</b> .....	17
<b>2.2</b>	<b>Sistema elétrico de potência</b> .....	17
<i>2.2.1</i>	<i>Automação de subestações</i> .....	17
<i>2.2.1.1</i>	<i>Hierarquia em um sistema de automação de subestação</i> .....	18
<b>2.3</b>	<b>Proteção de SEP</b> .....	19
<i>2.3.1</i>	<i>Relé de proteção</i> .....	20
<i>2.3.1.1</i>	<i>Relé digital</i> .....	22
<b>2.4</b>	<b>Sistema de supervisão, aquisição e controle de dados (SCADA)</b> .....	23
<b>2.5</b>	<b>Protocolos de comunicação</b> .....	24
<i>2.5.1</i>	<i>DNP3 IP</i> .....	24
<i>2.5.2</i>	<i>IEC 61850</i> .....	25
<i>2.5.3</i>	<i>Modbus IP</i> .....	25
<b>2.6</b>	<b>Metodologia ativa de aprendizagem</b> .....	26
<i>2.6.1</i>	<i>Aprendizagem baseada em projeto</i> .....	26
<b>2.7</b>	<b>Considerações finais</b> .....	27
<b>3</b>	<b>METODOLOGIA PARA DESENVOLVIMENTO DE PRÁTICAS DE ENSINO</b> .....	29
<b>3.1</b>	<b>Introdução</b> .....	29
<b>3.2</b>	<b>Metodologia de aprendizagem baseada em projeto</b> .....	29
<i>3.2.1</i>	<i>Objetivos da aprendizagem</i> .....	29
<i>3.2.1.1</i>	<i>Objetivo geral</i> .....	29
<i>3.2.1.2</i>	<i>Objetivos específicos</i> .....	29
<i>3.2.2</i>	<i>Atividades do projeto</i> .....	30
<i>3.2.3</i>	<i>Produtos</i> .....	30
<i>3.2.4</i>	<i>Formas de avaliação</i> .....	30



3.2.5	<b>Metodologia desenvolvida</b> .....	31
3.3	<b>Ferramentas utilizadas para construção do sistema SCADA</b> .....	32
3.3.1	<b>Software supervisor ScadaBR - especificações</b> .....	32
3.3.1.1	<i>Tipos de dados suportados</i> .....	33
3.3.1.2	<i>Data source</i> .....	33
3.3.1.3	<i>Data points</i> .....	33
3.3.2	<b>Protocolo Modbus IP – especificações</b> .....	34
3.3.2.1	<i>Modelo de Dados do Protocolo Modbus</i> .....	35
3.3.2.2	<i>Funções do Modbus</i> .....	36
3.3.2.2.1	<i>0x01 read coils</i> .....	36
3.3.2.2.2	<i>0x02 read discrete input</i> .....	36
3.3.2.2.3	<i>0x03 read holding registers</i> .....	36
3.3.2.2.4	<i>0x04 read input registers</i> .....	37
3.3.2.2.5	<i>0x05 write single coil</i> .....	37
3.3.2.2.6	<i>0x06 write single register</i> .....	37
3.3.3	<b>Elipse simulator Modbus</b> .....	37
3.3.4	<b>Subestação Campus do Pici</b> .....	38
3.4	<b>Considerações finais</b> .....	39
4	<b>APLICAÇÃO DA METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETO UTILIZANDO O SCADABR</b> .....	40
4.1	<b>Introdução</b> .....	40
4.2	<b>Metodologia de aprendizagem baseada em projeto para supervisor de uma subestação</b> .....	40
4.2.1	<b>Etapa 1 - estudos preliminares</b> .....	40
4.2.1.1	<i>Objetivo geral</i> .....	40
4.2.1.2	<i>Objetivos específicos</i> .....	40
4.2.1.3	<i>Metas – produtos da etapa</i> .....	41
4.2.2	<b>Etapa 2 - integração do sistema SCADA com os IEDs</b> .....	41
4.2.2.1	<i>Objetivo geral</i> .....	41
4.2.2.2	<i>Objetivos específicos</i> .....	41
4.2.2.3	<i>Metas – produtos da etapa</i> .....	42
4.2.3	<b>Etapa 3 – elaboração das telas de supervisão</b> .....	42
4.2.3.1	<i>Objetivo geral</i> .....	42
4.2.3.2	<i>Objetivos específicos</i> .....	42

4.2.3.3	<i>Metas – produtos da etapa</i> .....	43
4.2.4	<b>Formas de avaliação</b> .....	43
4.3	<b>Desenvolvimento do sistema SCADA</b> .....	43
4.3.1	<b>Configurações de comunicação</b> .....	43
4.3.1.1	<i>Ajustes efetuados no Elipse Modbus Simulator</i> .....	43
4.3.1.2	<i>Ajustes efetuados no ScadaBR</i> .....	45
4.3.2	<b>Definições dos dados a serem lidos e comandados</b> .....	50
4.3.2.1	<i>Estado do disjuntor</i> .....	52
4.3.2.2	<i>Mudança de grupo de ajuste</i> .....	53
4.3.2.3	<i>Abertura e fechamento de disjuntor</i> .....	56
4.3.2.4	<i>Nível de tensão</i> .....	58
4.3.2.5	<i>Habilitação dos data points</i> .....	59
4.3.2.6	<i>Detalhes dos data points</i> .....	60
4.3.3	<b>Watch list</b> .....	61
4.3.4	<b>Desenvolvimento de scripts</b> .....	64
4.3.5	<b>Detectores e tratadores de eventos</b> .....	65
4.3.5.1	<i>Detectores de eventos</i> .....	65
4.3.5.2	<i>Tratadores de eventos</i> .....	66
4.3.5.3	<i>Mudança de ajuste</i> .....	67
4.3.5.4	<i>Abertura e fechamento do disjuntor</i> .....	69
4.3.5.5	<i>Alarme indicativo de abertura do disjuntor</i> .....	73
4.3.6	<b>Desenvolvimento das telas de supervisão</b> .....	73
4.3.6.1	<i>Tela de supervisão - unifilar</i> .....	75
4.3.6.2	<i>Tela de supervisão – monitoramento de tensão</i> .....	77
4.3.6.3	<i>Tela de supervisão – alarmes</i> .....	79
4.3.7	<b>Resultados</b> .....	80
4.3.7.1	<i>Lógica para mudança de grupo de ajuste</i> .....	80
4.3.7.2	<i>Lógica para abertura e fechamento dos disjuntores</i> .....	82
4.3.7.3	<i>Tela de supervisão - unifilar</i> .....	83
4.3.7.4	<i>Tela de supervisão - monitoramento de tensão</i> .....	84
4.3.7.5	<i>Tela de supervisão - alarmes</i> .....	85
4.4	<b>Considerações finais</b> .....	86
5	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> .....	88
5.1	<b>Trabalhos futuros</b> .....	88

<b>REFERÊNCIAS</b> .....	90
<b>APÊNDICE A – <i>SCRIPTS</i> PARA MUDANÇA DE AJUSTE</b> .....	92
<b>APÊNDICE B – CÓDIGOS DOS COMPONENTES “HTML” E “<i>SCRIPT</i> PARA SERVIDOR”</b> .....	93

# 1 INTRODUÇÃO

## 1.1 Problemática e justificativa

Nos últimos anos, o uso de Sistema de Controle e Aquisição de Dados (SCADA) com a utilização do *software* livre ScadaBR tem se aplicado a: automação industrial e máquinas em geral; *hobby* e experimentos; automação residencial; automação predial; climatização e refrigeração; e energia. (DE FARIA *et al.*, 2012, p. 04).

A maior utilização do ScadaBR diz respeito à automação, seja de sistemas industriais, residenciais ou prediais. Sua aplicabilidade para a área de Sistemas Elétricos de Potência (SEP) é relativamente baixa, se comparada à utilização para automação em geral, o que é bastante refletido pela escassez de publicação de materiais que utilizam o *software* em Proteção de Sistemas Elétricos de Potência (PSE). (DE FARIA *et al.*, 2012, p. 04).

Os sistemas SCADA são componentes importantes para SEP, visto serem eles responsáveis pelo monitoramento e supervisão de subestações e, portanto, facilitam o trabalho dos operadores. Entre as funções de um SCADA está a comunicação por meio de protocolos com Dispositivos Elétricos Inteligentes (IED), dentre os quais se pode citar os relés. Dessa forma, realizar a comunicação entre o ScadaBR e um relé de proteção torna-se um trabalho bastante efetivo como uma fonte de estudo para diversas pessoas que têm interesse em efetuar essa aplicação com *softwares* livres, pois:

No mercado atual, existem muitos *softwares* SCADA disponíveis em diversas línguas, com custo relativamente elevado, o que gera uma inviabilização de sua utilização para automação de sistemas de menor porte ou em instituições de ensino tecnológico. Contudo, do decorrer dos anos 90, surgiram *softwares* livres com algumas funcionalidades de SCADA, que acarretou no incentivo de desenvolvimento de novas tecnologias baseadas em código aberto. (DE FARIA *et al.*, 2012, p. 03)

Com as inovações e evoluções ocorridas no setor elétrico, a quantidade de informações que um aluno de Engenharia Elétrica deve absorver é bastante elevada. Além disso, em decorrência à era digital que o mundo experimenta, novas tecnologias vêm sendo aplicadas ao ensino, de forma a incentivar a busca de conhecimento pelos alunos. Nesse contexto, faz-se necessário que os professores adotem diferentes metodologias de ensino que estimulem a criatividade e proporcionem o aprendizado necessário para formar os estudantes. (CARNEIRO *et al.*, 2019)

“A Metodologia de Aprendizagem Baseada em Projetos, como uma forma de ensino centrada no estudante, tem contribuído para importantes desenvolvimentos na educação de engenharia nos últimos anos.” (SANTOS-MARTIN *et al.*, 2012, p. 126, tradução própria).

Diante disso, desenvolver um sistema supervisorio com o uso do *software* livre ScadaBR a fim de se elaborar uma Metodologia de Aprendizagem Baseada em Projetos é extremamente importante como uma ferramenta a ser aplicada no ensino prático das disciplinas de Supervisão e Controle de Sistemas Elétricos de Potência, sendo utilizada uma plataforma de *software* livre, inclusiva, acessível financeiramente e de fácil uso.

A disciplina de Supervisão e Controle de Sistemas Elétricos de Potência faz parte da grade curricular optativa ofertada no curso de Engenharia Elétrica da Universidade Federal do Ceará (UFC).

## **1.2 Objetivos**

### ***1.2.1 Objetivo geral***

O objetivo desse trabalho é apresentar a aplicação de uma Metodologia de Aprendizagem Baseada em Projetos para o desenvolvimento de um Sistema Supervisorio SCADA na Disciplina de Supervisão e Controle de Sistemas Elétricos de Potência.

### ***1.2.2 Objetivos específicos***

Entre os objetivos específicos desse trabalho estão:

- Utilizar o *software* ScadaBR para desenvolvimento de sistema supervisorio de uma subestação de energia elétrica;
- Utilizar o *software* Elipse Modbus Simulator para simular a operação de uma relé de proteção;
- Empregar o protocolo de comunicação Modbus IP para o interfaceamento entre o Elipse Modbus Simulator e o ScadaBR;
- Desenvolver telas de supervisão, controle e aquisição de dados na plataforma;
- Efetuar a definição dos dados disponibilizados pelo Elipse Modbus Simulator, interpretando-os como dados fornecidos por um relé de proteção;
- Realizar a leitura de dados que representam grandezas analógicas e digitais;
- Enviar comandos do ScadaBR para o simulador;

- Criar alarmes para eventos ocorridos na simulação do relé.

### 1.3 Estrutura do trabalho

O presente trabalho está dividido da seguinte forma:

O capítulo 2 aborda a fundamentação teórica utilizada como base para desenvolver este trabalho, sendo abordados os diversos conceitos acerca de SEP, PSE, Sistema SCADA, uma visão geral acerca de protocolos de comunicação, e conceitos sobre Metodologia Ativa de Aprendizagem e sobre Metodologia de Aprendizagem Baseada em Projeto.

O capítulo 3 apresenta a Metodologia de Aprendizagem Baseada em Projeto desenvolvida, e os recursos utilizados para a realização do trabalho, além de explicar o protocolo *Modbus* IP, que foi utilizado para realizar a comunicação entre o ScadaBR e o *Elipse Modbus Simulator*.

O capítulo 4 apresenta a aplicação da Metodologia de Aprendizagem Baseada em Projeto aplicada ao ScadaBR, para aplicação como ensino prático na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência. Além disso, descreve o desenvolvimento e os resultados deste trabalho, no que diz respeito a: configurações de comunicação, via protocolo *Modbus*, efetuadas no ScadaBR e no *Elipse Modbus Simulator*; definição e interpretação dos dados a serem lidos e comandados do *Elipse Modbus Simulator*; realização de tratamentos lógicos nos dados e geração de alarmes; e elaboração das telas supervisórias.

O capítulo 5 apresenta as conclusões acerca do desenvolvimento do trabalho, e as propostas de realizações futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Introdução

Neste capítulo são apresentados os conceitos essenciais para o entendimento do desenvolvimento deste trabalho, como: Sistema Elétrico de Potência (SEP); Proteção de Sistemas Elétricos de Potência e seus componentes; Sistema de Supervisão, Controle e Aquisição de Dados (SCADA); e são descritos os conceitos gerais dos protocolos de comunicação. Além disso, é dada uma explanação acerca dos conceitos de Metodologia Ativa de Aprendizagem e de Metodologia de Aprendizagem Baseada em Projetos.

### 2.2 Sistema elétrico de potência

O SEP é um conjunto de equipamentos e instalações elétricas destinados a gerar, transmitir e distribuir energia elétrica. (BICHELS, 2018, p. 24).

Leão (2015, p. 17) destaca que um SEP tem o objetivo de gerar, transmitir e distribuir energia elétrica de forma que sejam atendidos padrões de confiabilidade, disponibilidade, qualidade, segurança e custos, ocasionando o menor impacto ambiental possível e proporcionando uma maior segurança pessoal.

#### 2.2.1 Automação de subestações

Com as evoluções ocorridas nos SEPs, foram implementadas tecnologias de automação nas subestações. “O termo Automação Elétrica tem sido usado para designar sistemas digitais que são utilizados para supervisão, comando, controle e proteção dos vários componentes do sistema elétrico.” (JARDINI, 1999).

A função majoritária de um Sistema de Automação de Subestação (SAS) é promover uma funcionalidade de supervisão da subestação, monitoramento dos dados provenientes dos equipamentos e o envio de comandos de forma remota, aumentando, dessa forma, o nível de proteção e a confiabilidade na operação de uma subestação. Esse sistema consiste em armazenar os dados dos *Intelligent Electronic Devices* (IEDs) em uma ou mais mensagens para ser efetuada a transmissão para uma estação mestre, podendo incluir um protocolo de tradução para se efetuar a leitura de dados de diferentes IEDs. (ACKERMAN, 1999, p. 275).

De acordo com Silva (2002, p. 15), há alguns fatores que tornam a automação extremamente necessária para a operação de uma subestação:

- Melhoria da qualidade no fornecimento de energia elétrica, com redução da quantidade e do tempo de interrupções, por meio da supervisão direta e em tempo real do sistema elétrico;
- Melhoria da qualidade de informação para a operação local;
- Redução da incidência de ocorrências no sistema elétrico em razão de erros humanos de operação;
- Implementação de novas funções para controles sistêmicos e de comandos de equipamentos;
- Operação remota de SE com base no centro de operação do sistema, padronizando os relatórios;
- Redução do custo operacional, com a automação de tarefas e centralização de ações operativas;
- Redução dos investimentos com a modelagem da curva de carga que possibilita aproveitar melhor a capacidade dos equipamentos;
- Melhoria dos níveis de tensão, de segurança operacional e dos índices de “Duração equivalente de interrupção por consumidor” (DEC) e “Frequência equivalente de interrupção por consumidor” (FEC);
- Otimização do quadro de operadores.

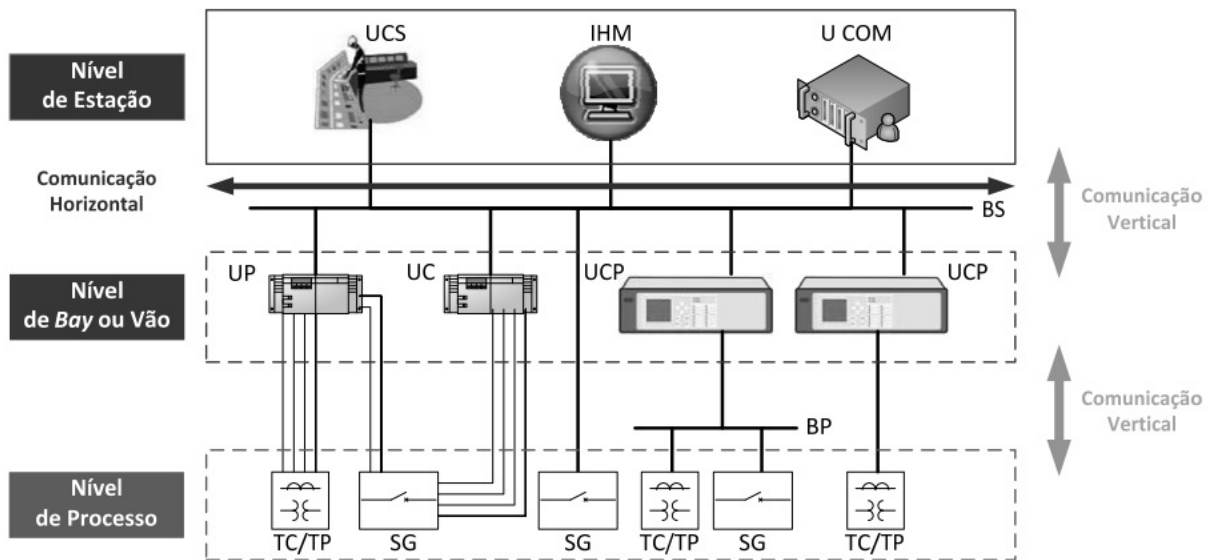
Diante disso, constata-se que a automatização de subestações é muito importante para trazer uma maior eficiência operacional em uma subestação, proporcionar uma maior proteção para os equipamentos da rede elétrica e para as pessoas e melhorar o fornecimento de energia elétrica.

#### *2.2.1.1 Hierarquia em um sistema de automação de subestação*

O SAS opera sob uma hierarquia dividida em três níveis, conforme mostrado na Figura 1.



Figura 1 - Hierarquia de um sistema de automação de subestação



Fonte: ALMEIDA (2011, p. 43)

O nível de processo é onde se localizam os equipamentos primários da rede elétrica, como disjuntores, transformadores de corrente e transformadores de potencial, dentre outros equipamentos. O nível de *bay* é onde se localizam os dispositivos inteligentes para proteção e controle da subestação, como relés de proteção. O nível de estação é onde se localizam as unidades de controle e o sistema SCADA.

O nível de processo se comunica com o nível de *bay* por meio de conexões de *input/output* (I/O), já a comunicação entre o nível de *bay* com o nível de estação ocorre por meio dos protocolos de comunicação.

### 2.3 Proteção de SEP

A operação do SEP é suscetível à ocorrência de falhas, as quais ocorrem frequentemente, o que ocasiona interrupções no fornecimento de energia elétrica, diminuindo, dessa forma, a qualidade do serviço prestado. Uma das falhas que ocorrem com mais frequência é o curto-circuito, o qual pode ocasionar danos aos equipamentos do SEP, danos às instalações elétricas dos consumidores. Além disso, existem outros tipos de anormalidades, como sobretensões, subtensões e sobrecargas. Essas falhas podem ser irrelevantes para o SEP ou podem ser desastrosas. (MAMEDE FILHO; MAMEDE, 2013).

Nesse contexto, observa-se a grande relevância de um sistema de proteção bem elaborado, pois ele garante a integridade dos equipamentos do SEP, garante a segurança das pessoas e aumenta o grau de confiabilidade do sistema elétrico.

Para Mamede Filho e Mamede (2013):

A principal função de um sistema de proteção é assegurar a desconexão de todo sistema elétrico submetido a qualquer anormalidade que o faça operar fora dos limites previstos ou de parte dele.

Em segundo lugar, o sistema de proteção tem a função de fornecer as informações necessárias aos responsáveis por sua operação, de modo a facilitar a identificação dos defeitos e a sua consequente recuperação. De modo geral, a proteção de um sistema de potência é projetada tomando como base os fusíveis e os relés incorporados necessariamente a um disjuntor, que é, na essência, a parte mecânica responsável pela desconexão do circuito afetado com a fonte supridora.

Para garantir uma proteção mais eficiente e eficaz, a proteção de um sistema elétrico deve possuir os seguintes requisitos:

- **Seletividade:** quando somente os dispositivos de proteção mais próximos à falta atuam;
- **Zonas de Atuação:** o sistema elétrico deve ser dividido em zonas, e para cada zona devem ser associados dispositivos de proteção, os quais devem ser ajustados de forma a garantir a seletividade;
- **Velocidade:** o sistema de proteção deve ser projetado de forma que sua atuação deve ser rápida, evitando que ocorram maiores danos ao sistema e às pessoas;
- **Sensibilidade:** os dispositivos devem efetuar a leitura dos valores da rede com precisão, de forma que atuem somente nos valores adequados;
- **Confiabilidade:** o sistema de proteção deve ser seguro e exato quanto as suas respectivas atuações;
- **Automação:** os dispositivos de proteção devem atuar, quando devido, de forma automática de acordo com as grandezas elétricas.

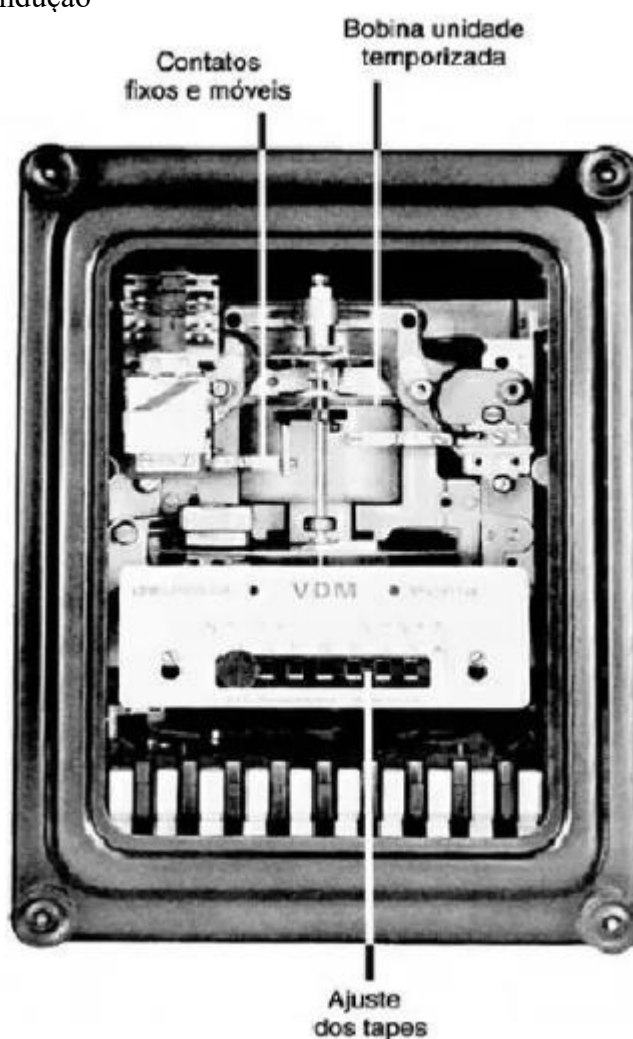
### ***2.3.1 Relé de proteção***

O relé é um dispositivo bastante essencial para a proteção de um SEP, visto que ele realiza o comando para a abertura ou fechamento do disjuntor. Ele pode fornecer proteção contra: sobrecarga, sobretensão, subtensão, curto-circuito, dentre outros tipos de faltas. Esse dispositivo pode oferecer proteção contra apenas um tipo de ocorrência na rede, ou vários tipos. Esse equipamento pode ser classificado quanto à sua tecnologia como: relés eletromecânicos

de indução, relés eletrônicos e relés digitais. Este último tipo dominou o mercado, sendo um dos tipos mais utilizados em subestações, ou outros sistemas elétricos (MAMEDE FILHO; MAMEDE, 2013).

O primeiro relé desenvolvido foi do tipo eletromecânico de indução, o qual está apresentado na Figura 2. Esse equipamento possuía bobinas, contatos fixos e móveis, molas e discos de indução. Além disso, esse tipo de relé é bastante robusto, possuindo vida-útil elevada. (MAMEDE FILHO, MAMEDE, 2013).

Figura 2 - Relé de sobrecorrente eletromecânico de indução



Fonte: Mamede; Mamede Filho (2013)

Os relés de proteção são associados aos dispositivos Transformador de Corrente (TC) e Transformador de Potencial (TP), os quais são componentes que reduzem as grandezas elétricas de forma que as estruturas físicas dos dispositivos de proteção não precisem ser tão robustas para suportarem, e, por conseguinte, lerem, valores elétricos elevados.

### 2.3.1.1 Relé digital

Um tipo de relé, que contribui para a modernização do SEP, é o relé digital. De acordo com Mamede e Mamede Filho (2013), o relé digital é um dispositivo microprocessado, com uma alta velocidade de processamento, e funcionam por meio de programas dedicados que processam as informações oriundas do TC e do TP. A Figura 3 representa o relé digital de sobrecorrente da fabricante PEXTRON, modelo URPE 7104.

Os relés digitais revolucionaram os esquemas de proteção, oferecendo vantagens impossíveis de serem obtidas dos seus antecessores. Além das funções de proteção propriamente ditas, os relés digitais realizam funções de comunicação, medidas elétricas, controle, sinalização remota, acesso remoto etc. (MAMEDE FILHO; MAMEDE, 2013).

Os ajustes dos relés digitais podem ser efetuados por meio de um painel frontal ou por meio de comunicação com um computador, por meio de protocolos de comunicação específicos. Essa comunicação pode ser efetuada por meio de comunicação serial ou por meio do IP, sendo utilizados cabos de fibra ótica.

Figura 3 - Relé digital



Fonte: Manual de Operação URPE 7104, PEXTRON.

## 2.4 Sistema de supervisão, aquisição e controle de dados (SCADA)

De forma a garantir que os operadores tenham maior controle e visibilidade sobre os sistemas elétricos, em especial subestações, são utilizados sistemas SCADA. Um sistema SCADA consiste em utilizar tecnologias de comunicação de forma a se monitorar, supervisionar ou, até mesmo, controlar um processo ou um sistema, por meio de uma interface que possibilite a realização de integração entre indivíduos capacitados e os equipamentos que estão inseridos nesse sistema. Essa interface é conhecida como Interface Homem-Máquina (IHM). O SCADA proporciona uma forma compacta e ampla de visualização do estado de todo o sistema e de todos os eventos que podem ocorrer neste. É bastante aplicado na área industrial e na área elétrica.

Por intermédio de um sistema SCADA, no caso de sua aplicação em sistema elétrico, é possível visualizar em um *software* os valores das grandezas elétricas, visualizar se os disjuntores estão abertos ou fechados, gerar alarmes para ocorrências de caráter urgente em uma subestação, dentre diversos outros estados e grandezas, o que facilita diversas operações em uma subestação, como: manobras e manutenção de equipamentos.

Segundo Silva (2002, p.18), a arquitetura básica de um sistema SCADA consiste em:

- Um sistema central de controle COS que deve contar com equipamentos e funcionalidade suficiente para assumir as funções referentes também a qualquer centro regional, em caso de contingências.
- Sistemas regionais COR, normalmente, um em cada uma das gerências, para assumir as funções de centros de operação de distribuição.

Silva (2002, p. 19) pontua que o COS deve ser responsável pelas seguintes funções:

- Supervisão e controle do sistema de alta tensão.
- Coordenação da operação com companhias interligadas.
- Controle de tensões.
- Controle de desligamento de cargas.

Para Liberato (2014, p. 15):

Os métodos atuais de supervisão, controle e proteção presentes nas subestações de energia são pensados para atender inúmeros requisitos operacionais do setor elétrico. Esta estrutura é composta por relés de proteção, controladores, redes de comunicação, *gateway* para prover a integração com o SCADA remoto e uma IHM. O objetivo é

facilitar as ações de operadores, de forma remota ou local, para realizarem atividades como: manobras, leituras instantâneas, registros históricos de grandezas, leitura e registros de lista de eventos, alarmes e outros. Facilita também o acesso aos dados pela equipe de engenharia, para alterações de ajustes de proteção e coleta de oscilografia.

## 2.5 Protocolos de comunicação

Para a realização de um sistema SCADA, faz-se necessário utilizar protocolos de comunicação, de forma que seja possível interligar o sistema supervisório com os dispositivos de uma subestação para serem efetuados comandos pelo sistema supervisório e que também seja possível visualizar o estado dos equipamentos da subestação – como o estado dos disjuntores – e verificar a ocorrência de eventos na rede – por exemplo, faltas no sistema.

De acordo com Liberato (2014, p. 16):

É um conjunto de regras pré-estabelecidas que regem e padronizam o tratamento e a formatação dos dados em um determinado sistema. [...]

Estas regras são de fundamental importância para garantir que em um projeto, em que são usados dispositivos de fabricantes diferentes possam interagir entre si de forma fácil sem a necessidade de conversões de protocolos. Este conceito de interoperabilidade nem sempre é possível.

Alguns dos principais protocolos de comunicação utilizados para se efetuar a comunicação em um sistema SCADA de um sistema elétrico são: DNP3, IEC 61850 e MODBUS.

### 2.5.1 DNP3 IP

Para Johnson (2000, p. 2, tradução própria):

O DNP3 é baseado no trabalho da International Electrotechnical Commission (IEC) Technical Committee 57 (TC-57) que resultou no protocolo IEC 60870-5, o qual é usado em muitas regiões do mundo, sendo predominante na Europa. Os protocolos DNP3 e IEC 60870-5 fazem parte da IEEE Standard 1379, que foi recentemente atualizada de uma Trial Recommended Practice para uma Full Recommend Practice.

O *Distributed Network Protocol* (DNP3) é um protocolo de comunicação que possui código aberto. O grupo de usuários do DNP3 – o *DNP3 Users Group* – é o responsável pela evolução do protocolo, visto que os usuários sugerem adicionais ou modificações para o protocolo, os quais são avaliados por um comitê desse grupo de usuários – o *DNP3 Users Group*

*Committee* – e o comitê efetua a alteração do protocolo de acordo com o direcionamento desse grupo de usuários (*Distributed Network Protocol*).

A conexão para ser realizada a comunicação pelo protocolo DNP3 se dá utilizando-se uma porta serial, ou uma conexão *ethernet*, conhecida como *DNP3 Transmission Control Protocol* (TCP)/IP.

### 2.5.2 IEC 61850

Segundo Souza (2016, p. 30):

Esta norma foi publicada em 2004, mas vem sendo desenvolvida desde a década de 1990 envolvendo grandes entidades de pesquisas mundiais, como o Electric Power Research Institute (EPRI), Engenharia Eletrotécnica Comitê (IEC), Centro de Pesquisas de Energia Elétrica (Cepel), só para citar alguns. A norma tem grande aceitação nas Américas, Europa e Ásia e já está se firmando como um padrão mundial, o que justifica uma real avaliação pelas empresas sobre a pertinência da sua utilização.

A Norma Internacional IEC 61850 possui o objetivo de proporcionar uma interoperabilidade entre equipamentos de fabricantes diferentes, de forma que não haja prejuízo na troca de dados e que seja permitida a operação do conjunto de equipamentos, independente do fabricante do IED. (LIBERATO, 2014, p. 17).

### 2.5.3 Modbus IP

De acordo com *Modbus*:

O protocolo *Modbus* é uma estrutura de mensagem desenvolvida pela empresa *Modicon* em 1979. É utilizado para estabelecer uma comunicação cliente-servidor entre dispositivos inteligentes. É, de fato, um protocolo de comunicação padrão, aberto e o mais amplamente utilizado na indústria. Ele foi implementado por centenas de fornecedores em milhares de dispositivos distintos para transferir entradas/saídas discretas e analógicas, e registradores de dados entre dispositivos de controle. É denominador comum entre diferentes fabricantes. Um relatório se referiu ao protocolo como “padrão, de fato, de integração entre diversos fornecedores”. Analistas da indústria reportaram mais de 7 milhões de nós *Modbus* apenas na América do Norte e na Europa.

O protocolo *Modbus* é utilizado em diversas aplicações cliente-servidor com a finalidade de programar e monitorar dispositivos. Esse protocolo realiza a comunicação entre IEDs, sensores e instrumentos; monitora um campo de dispositivos utilizando computadores e IHMs. O *Modbus* é bastante aplicado em Unidades Terminais Remotas (UTRs), sendo, dessa

forma, usado em um contexto de uma subestação, monitoramento de óleo e gás. É, portanto, aplicado nas áreas: industrial, civil, infraestrutura, transporte e energia. (*MODBUS*).

## **2.6 Metodologia ativa de aprendizagem**

A metodologia ativa de aprendizagem consiste em o aluno ir em busca de obter o conhecimento por meio da escrita, leitura, discussões, elaboração de um projeto ou solução de problemas. O professor, nesse caso, é um guia, ou orientador, para o processo de aprendizagem. (BARBOSA, MOURA, 2011, p. 111).

Segundo Barbosa e Moura (2011, p. 111), a estratégia utilizada para o uso da metodologia ativa deve fazer que os alunos utilizem suas funções mentais de pensar, raciocinar, observar, refletir, entender e combinar durante o processo de aprendizagem, ocasionando que os estudantes aprendam um maior volume de conteúdo, e retenham o conhecimento por mais tempo.

Valente, Almeida e Geraldini (2017, p. 464) pontuam que algumas das estratégias existentes para implementação da metodologia ativa de aprendizagem são:

- Aprendizagem baseada em projeto;
- Aprendizagem por meio de jogos;
- Método do caso ou discussão e solução de casos;
- Aprendizagem em equipe.

Dentre as estratégias mencionadas, escolheu-se a utilização da Aprendizagem Baseada em Projeto (ABP), por se enquadrar no contexto de ensino da disciplina de Supervisão e Controle de Sistemas Elétricos de Potência, do curso de Engenharia Elétrica da Universidade Federal do Ceará (UFC).

### **2.6.1 Aprendizagem baseada em projeto**

A Aprendizagem Baseado em Projeto (ABP) é uma metodologia ativa de aprendizagem, isto é, centrada no aluno. Consiste em propor o desenvolvimento de um projeto, com emprego de uma problemática determinada, de forma que haja uma orientação das atividades que devem ser desenvolvidas ao longo da metodologia de forma que o aluno obtenha o conhecimento necessário para o desenvolvimento do projeto. O projeto e a situação problema



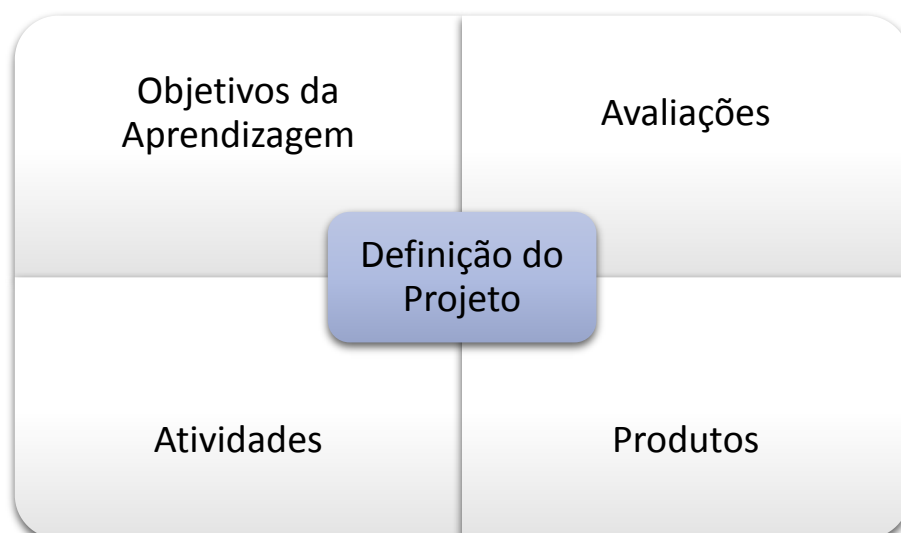
devem ser novos aos alunos, exigindo para elaboração um maior conhecimento do que o aluno possa oferecer. (SANTOS-MARTIN *et. al*, 2012, p. 126).

Vieira (2017, p. 5) pontua que a Metodologia de Aprendizagem Baseada em Projeto compõe uma alternativa de aprendizagem que proporciona uma maior interação do estudante com o problema, e, conseqüentemente, com novas aprendizagens.

Essa metodologia se mostra como uma ferramenta diferenciada para aplicação no ensino dos alunos, visto que os alunos utilizarão suas funções mentais para elaboração do projeto e obtenção do conhecimento.

A metodologia de ABP proposta nesse trabalho abrange quatro etapas bem definidas, as quais são: objetivos da aprendizagem, atividades, produtos e avaliação, conforme mostrado na Figura 4.

Figura 4 - Metodologia de aprendizagem baseada em projeto



Fonte: O próprio autor.

As quatro etapas de desenvolvimento da Metodologia de Aprendizagem Baseada em Projeto estão descritas com mais detalhes nos capítulos 3 e 4.

## 2.7 Considerações finais

Com base nas informações realizadas neste capítulo, optou-se por utilizar o *software* ScadaBR para elaboração do sistema SCADA e o protocolo *MODBUS* IP para desenvolvimento do trabalho.

O *software* livre ScadaBR foi escolhido por ser gratuito, e por ter se mostrado uma ferramenta de utilização simples, com interface amigável e com diversos recursos.

Dentre os protocolos de comunicação mencionados, foi escolhido o *MODBUS* IP em decorrência de este protocolo ser disponibilizado pelo *software* ScadaBR e ser amplamente utilizado na elaboração de sistemas com o referido programa, havendo bastante material na *internet* acerca desse uso.

Ademais, foram apresentados aspectos conceituais e os tipos de Metodologia Ativa de Aprendizagem, com maior ênfase na Metodologia de Aprendizagem Baseada em Projeto (ABP). Nos capítulos 3 e 4, será apresentada a ABP aplicada no ensino de sistema SCADA.

### **3 METODOLOGIA PARA DESENVOLVIMENTO DE PRÁTICAS DE ENSINO**

#### **3.1 Introdução**

Nesse capítulo é apresentada uma Metodologia de Aprendizagem Baseada em Projeto desenvolvida para elaboração de um sistema SCADA. Além disso, são listadas e descritas as ferramentas utilizadas para o desenvolvimento do sistema SCADA, que são: *software* ScadaBR, protocolo de comunicação, o simulador *Elipse Modbus Simulator* e o diagrama unifilar da subestação *Campus* do Pici, que serviu como base para elaboração para uma das telas do supervisório.

#### **3.2 Metodologia de aprendizagem baseada em projeto**

Nessa seção é apresentada a proposta de Metodologia de Aprendizagem Baseada em Projeto desenvolvida para a elaboração de um sistema supervisório SCADA.

##### **3.2.1 *Objetivos da aprendizagem***

###### **3.2.1.1 *Objetivo geral***

O desenvolvimento do projeto objetiva propiciar conhecimentos e habilidades aos alunos acerca de Sistema de Aquisição e Controle de Dados (SCADA) aplicado a subestação.

###### **3.2.1.2 *Objetivos específicos***

Os alunos devem desenvolver competências e habilidades para:

- Compreender o funcionamento de um Sistema SCADA;
- Realizar a integração entre o *software* SCADA e os *Intelligent Electronic Devices* (IEDs) por meio de protocolo de comunicação;
- Interpretar os dados utilizados no Sistema SCADA e fornecidos pelos dispositivos;
- Criar telas de supervisão, utilizando-se elementos como botões, gráficos e listas de seleção.

### 3.2.2 Atividades do projeto

O desenvolvimento da Metodologia de Aprendizagem Baseada em Projetos, proposta para aplicação na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência, abrange as seguintes etapas:

- Etapa 1 – Estudos preliminares;
- Etapa 2 – Integração do Sistema SCADA com os IEDs;
- Etapa 3 – Elaboração das telas de supervisão.

### 3.2.3 Produtos

Ao final de cada etapa, os alunos deverão apresentar produtos para o professor em um prazo definido. A etapas e a definição dos prazos.

Tabela 1 apresenta os produtos que devem ser entregues na finalização das etapas e a definição dos prazos.

Tabela 1 - Descrição dos produtos e prazos de cada etapa

<b>Etapa</b>	<b>Descrição</b>	<b>Entregas/Produtos</b>	<b>Prazo</b>
1	Estudos preliminares.	<ul style="list-style-type: none"> <li>• Apresentação de relatório ou de seminário sobre os estudos realizados.</li> </ul>	Primeira semana
2	Integração do Sistema SCADA com os IEDs.	<ul style="list-style-type: none"> <li>• Apresentação de relatório ou seminário acerca das atividades implementadas;</li> <li>• Apresentação de vídeo com a apresentação dos pontos analógicos e digitais configurados no SCADA.</li> </ul>	Segunda semana
3	Elaboração das telas de supervisão.	<ul style="list-style-type: none"> <li>• Apresentação de relatório sobre o desenvolvimento das telas.</li> <li>• Apresentação de vídeo demonstrando o funcionamento das telas.</li> </ul>	Terceira semana

Fonte: O próprio autor.

### 3.2.4 Formas de avaliação

A Tabela 2 apresenta propostas de formas de avaliação dos produtos descritos.

Tabela 2 - Descrição dos pontos a serem avaliados de cada produto

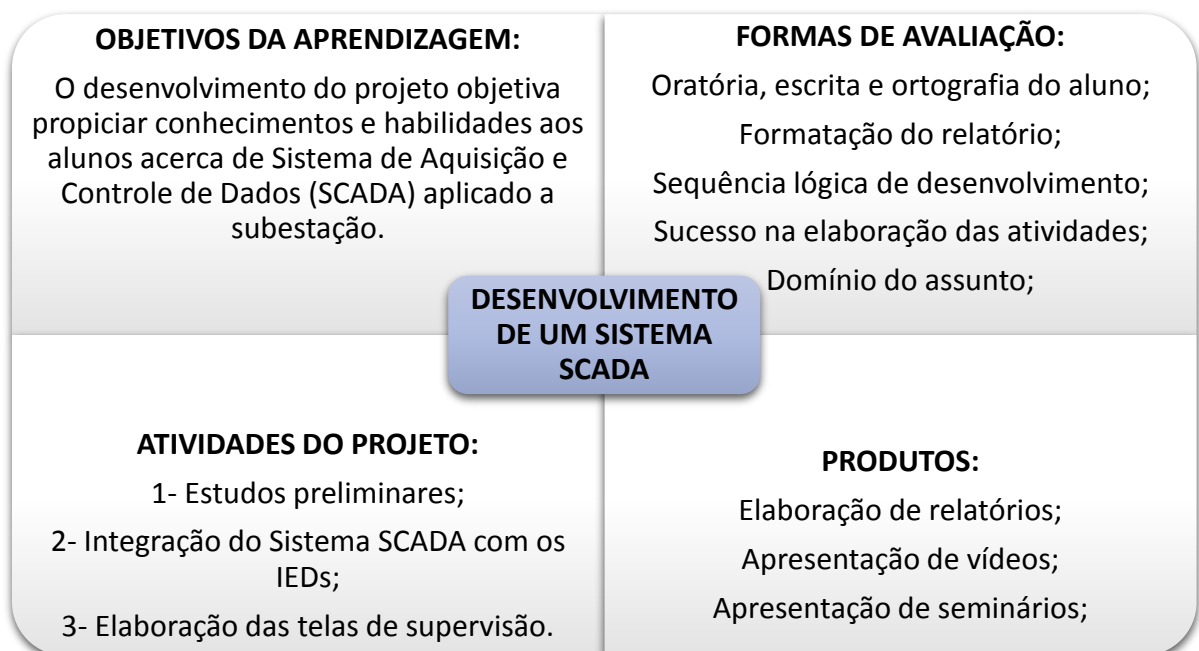
Produto	Pontos a serem avaliados
Apresentação de relatório	<ul style="list-style-type: none"> <li>• Escrita e ortografia do aluno;</li> <li>• Formatação do relatório;</li> <li>• Sequência lógica de desenvolvimento da escrita, contendo: introdução, desenvolvimento, metodologia e conclusões;</li> <li>• Domínio do assunto;</li> <li>• Sucesso na elaboração das atividades.</li> </ul>
Apresentação de vídeo ou de seminário	<ul style="list-style-type: none"> <li>• Oratória do aluno;</li> <li>• Apresentação das ideias em uma sequência lógica, contendo: introdução, desenvolvimento, metodologia e conclusões;</li> <li>• Domínio do assunto;</li> <li>• Sucesso na elaboração das atividades.</li> </ul>

Fonte: O próprio autor.

### 3.2.5 Metodologia desenvolvida

A Figura 5 apresenta o quadro geral da Metodologia de Aprendizagem Baseada em Projeto desenvolvida.

Figura 5 - Metodologia de aprendizagem baseada em projeto desenvolvida



Fonte: O próprio autor.

### 3.3 Ferramentas utilizadas para construção do sistema SCADA

#### 3.3.1 *Software supervisorio ScadaBR - especificações*

O ScadaBR é um *software* livre para desenvolvimento de sistema SCADA para as mais diversas aplicações. Além disso, o ScadaBR é gratuito e de código-fonte aberto.

Para ScadaBR (2021):

O ScadaBR pretende oferecer todas as funcionalidades de um sistema SCADA tradicional. Este tipo de software (*Supervisory Control and Data Acquisition*, na sigla em inglês) existe desde o final dos anos 60, e é a peça fundamental em qualquer tipo de aplicação computadorizada que envolva máquinas, controladores programáveis (CLP's), acionamentos eletrônicos e sensores.

O ScadaBR é um *software* multiplataforma baseado em Java, podendo ser utilizado em diversos sistemas operacionais, como *Windows*, *Linux* e *MAC*, necessitando de um servidor de aplicações para execução. Utiliza-se como servidor padrão o *Apache Tomcat*. (SCADABR, 2010, p. 6).

São diversas funcionalidades do ScadaBR (SCADABR, 2021), como:

- Ambiente de desenvolvimento para aplicações no sistema SCADA;
- Alarmes sonoros;
- Relatórios de variáveis, alarmes e eventos;
- Acesso em *browser*, o qual pode ser efetuado por meio de computadores, notebooks, ou, até mesmo, por meio de celulares;
- Visualização de dados em tempo real;
- Construção de telas gráficas (IHMs ou Sinópticos).
- Suporte a mais de 20 protocolos de comunicação, como: *MODBUS TCP/IP* e *Serial*, *DNP3*, *Serial ASCII*, *IEC101* e *Hypertext Transfer Protocol (HTTP)*.
- Detectores de eventos, tratadores de eventos e elaboração de *scripts* funcionando como um tratamento lógico para os dados;

O *software* pode ser utilizado em diversas aplicações, como:

- Automação industrial e residencial;
- Criação de sistema supervisorio para subestações, pequenas centrais hidrelétricas;
- Monitoramento de temperatura.

Devido ao fato de o *software* ser gratuito e de código-fonte aberto, o acesso e estudo do ScadaBR pode ser efetuado por diversos alunos de universidades que tenham interesse nessa área, tendo em vista que alguns programas que realizam as mesmas funções possuem um preço elevado. Nesse contexto, o ScadaBR estimula a criatividade e o desenvolvimento de raciocínio lógico dos alunos para a execução desses sistemas.

#### 3.3.1.1 *Tipos de dados suportados*

O ScadaBR suporta cinco tipos de dados:

- Valores binários (ou booleanos);
- Valores de estados múltiplos, que são números inteiros;
- Valores numéricos analógicos, que são números decimais representados no sistema como variáveis de ponto flutuante;
- Valores alfanuméricos, que são uma sequência de caracteres;
- Valores em imagens, isto é, representações binárias de dados de imagens.

#### 3.3.1.2 *Data source*

*Data source* é o dispositivo de onde se adquire os dados pelo protocolo de comunicação. No caso do protocolo *MODBUS* IP, o ScadaBR realiza a aquisição de dados por varredura (*polling*), ou seja, o *software* faz a requisição de dados em um dispositivo por vez em um intervalo de tempo definido.

#### 3.3.1.3 *Data points*

“Os valores de dados recebidos ou coletados por um *data source* são armazenados em *data points*.” (SCADABR, 2010, p.10).

Um *data point* é um histórico de valores específicos, podendo ser, por exemplo, a leitura de temperatura de um quarto ou umidade do mesmo quarto. Além disso, o *data point* pode ser uma variável de controle, como um botão liga/desliga de um equipamento. (SCADABR, 2010, p.11)

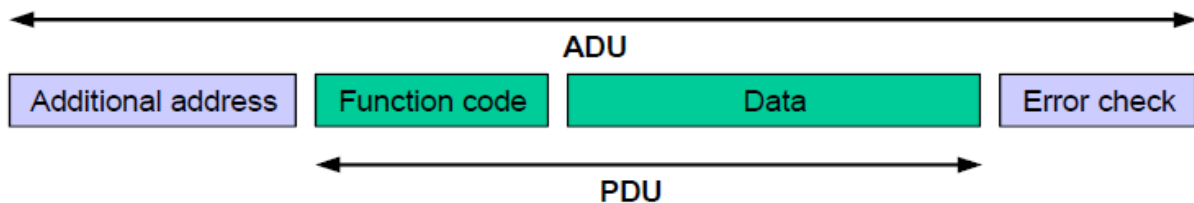
### 3.3.2 Protocolo Modbus IP – especificações

O protocolo *Modbus* está posicionado no nível 7 do Modelo *Open System Interconnection* (OSI), ou seja, na camada de aplicação, a qual realiza uma comunicação cliente/servidor entre dispositivos conectados em redes ou barramentos.

O protocolo *MODBUS* define uma simples unidade de dados de protocolo (PDU) independente das camadas de comunicação subjacentes. O mapeamento do protocolo *MODBUS* em barramentos específicos ou na rede pode introduzir alguns campos adicionais na unidade de dados da aplicação (ADU). (MODBUS, 2012, p. 03).

A Figura 6 ilustra a PDU e a ADU.

Figura 6 - Quadro geral do *Modbus*



Fonte: *Modbus* (2012, p. 03).

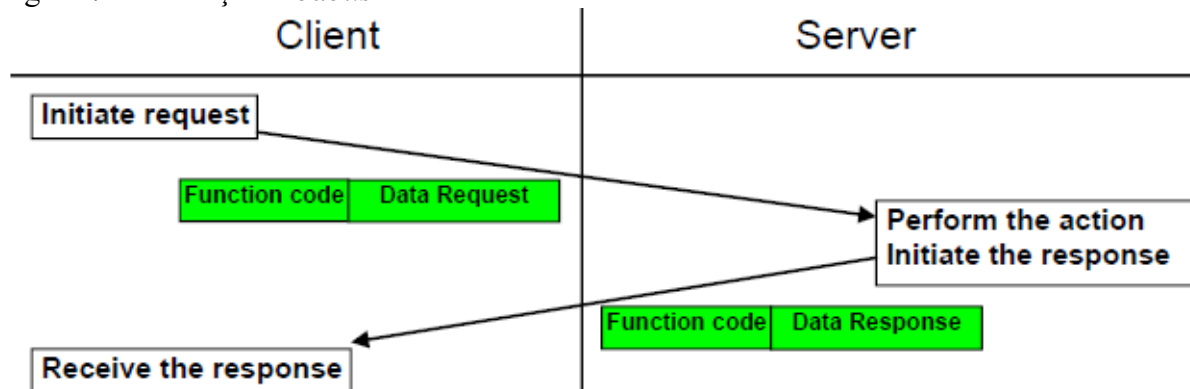
A ADU estabelece o formato de requisição iniciado por um cliente. O cliente que inicia a transação *Modbus* é responsável por construir a ADU. A PDU possui as seguintes definições:

- **Function Code:** quando enviada na requisição do cliente indica que ação o servidor deve tomar. Quando enviada na resposta do servidor ao cliente pode indicar a ocorrência de um erro, ou, caso contrário, é somente um espelho da função enviada pelo cliente.
- **Data:** no envio da mensagem do cliente para o servidor, indica informações adicionais para orientar a ação que o servidor deve tomar pela definição da *function code*, como o endereço dos dados a serem lidos ou alterados, neste caso, sendo informado, também, o valor de alteração. Na resposta do servidor ao cliente, esse campo contém o valor do dado solicitado caso não tenha ocorrido erro, ou, caso contrário, o *exception code* que indica a próxima ação que o servidor deve tomar.



A Figura 7 indica essa troca de mensagens cliente/servidor sem erros.

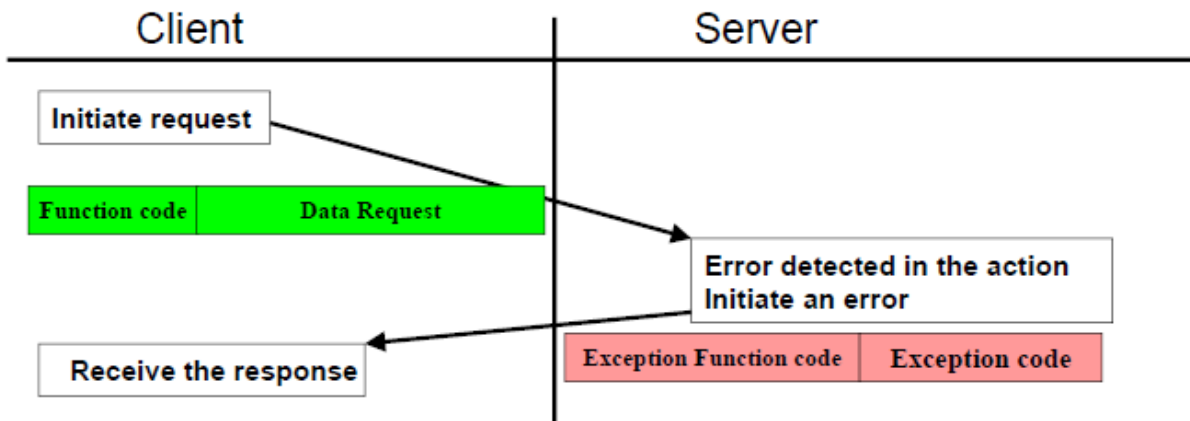
Figura 7 - Transação *Modbus* sem erros



Fonte: MODBUS (2012, p. 4).

A Figura 8 indica essa troca de mensagens cliente/servidor com erros.

Figura 8 - Transação *Modbus* com erros



Fonte: MODBUS (2012, p. 4).

### 3.3.2.1 Modelo de Dados do Protocolo Modbus

A Tabela 3 mostra os tipos de dados que são operados pelo protocolo *Modbus*, o tipo de objeto, tipos de operação e esclarecimentos adicionais.

Tabela 3 - Modelo de dados do protocolo *Modbus* (continua)

Tabela Primária	Tipo de objeto	Tipo	Comentários
Entradas Discretas	Único bit	Apenas leitura	Esse tipo de dado pode ser fornecido por um sistema E/S.
<i>Coils</i>	Único bit	Leitura-escrita	Esse tipo de dado pode ser alterado por um programa.

Tabela 3 – Modelo de dados do protocolo *Modbus* (conclusão)

<b>Registradores de Entrada</b>	<b>Palavra de 16 bits</b>	<b>Apenas leitura</b>	<b>Esse tipo de dado pode ser fornecido por um sistema E/S.</b>
Registradores <i> Holding</i>	Palavra de 16 bits	Leitura-escrita	Esse tipo de dado pode ser alterado por um programa.

Fonte: *Modbus* (2012, p. 6, tradução própria)

### 3.3.2.2 Funções do *Modbus*

A seguir, são apresentadas as principais funções do protocolo *Modbus* utilizadas no trabalho.

#### 3.3.2.2.1 *0x01 read coils*

A função *0x01* é utilizada para ler de 1 a 2000 contíguos estados de saída discreta de um dispositivo remoto. O pedido de dados realizado pela *Protocol Data Unit* (PDU) contém o código da função, o endereço de início e quantidade de saídas discretas. A resposta ao pedido de dados contém o código da função, o contador do byte e o estado da saída discreta.

#### 3.3.2.2.2 *0x02 read discrete input*

A função *0x02* é utilizada para ler de 1 a 2000 contíguos estados de entrada discreta de um dispositivo remoto. O pedido de dados realizado pela PDU contém o código da função, o endereço de início e a quantidade de entradas discretas. A resposta ao pedido de dados contém o código da função, o contador do byte e o estado da entrada discreta.

#### 3.3.2.2.3 *0x03 read holding registers*

A função *0x03* é utilizada para ler o conteúdo de um bloco contíguo de registradores do tipo *holding* em um dispositivo remoto. O pedido de dados realizado pela PDU contém o código da função, o endereço de início e a quantidade de registradores. A resposta ao pedido de dados contém o código da função, o contador do byte e o valor do registrador.

#### 3.3.2.2.4 *0x04 read input registers*

A função 0x04 é utilizada para ler de 1 a 125 contíguos registradores de entrada em um dispositivo remoto. O pedido de dados realizado pela PDU contém o código da função, o endereço de início e a quantidade de registradores. A resposta ao pedido de dados contém o código da função, o contador do byte e o valor do registrador de entrada.

#### 3.3.2.2.5 *0x05 write single coil*

A função 0x05 é utilizada para mudar o estado de uma única saída para ligado ou desligado. O pedido realizado de escrita pela PDU contém o código da função, o endereço da saída a ser comandada, e o valor 0xFF00 para comandar a saída para 1 ou o valor 0x0000 para comandar a saída para 0 o endereço de início e a quantidade de registradores. A resposta para esse pedido é uma repetição do pedido, porém retornando o valor já alterado da saída.

#### 3.3.2.2.6 *0x06 write single register*

A função 0x06 é utilizada para fazer a escrita de um dado em um registrador do tipo *holding* em um dispositivo remoto. O pedido realizado de escrita pela PDU contém o código da função, o endereço do registrador a ser alterado, e o valor de escrita, que pode ser de 0x0000 a 0xFFFF. A resposta para esse pedido é uma repetição do pedido, porém retornando o valor já alterado do respectivo registrador.

### 3.3.3 *Elipse simulator Modbus*

O *Elipse Simulator Modbus* é um *software* que simula dispositivos que se comunicam via protocolo *Modbus*.

De acordo com Elipse (2013), o *software* possui os seguintes recursos:

- Simulação de até 10 dispositivos simultaneamente;
- Compatibilidade com os protocolos *Modbus* RTU e *Modbus* TCP;
- Geração de valores aleatórios em um intervalo de tempo definido pelo usuário;

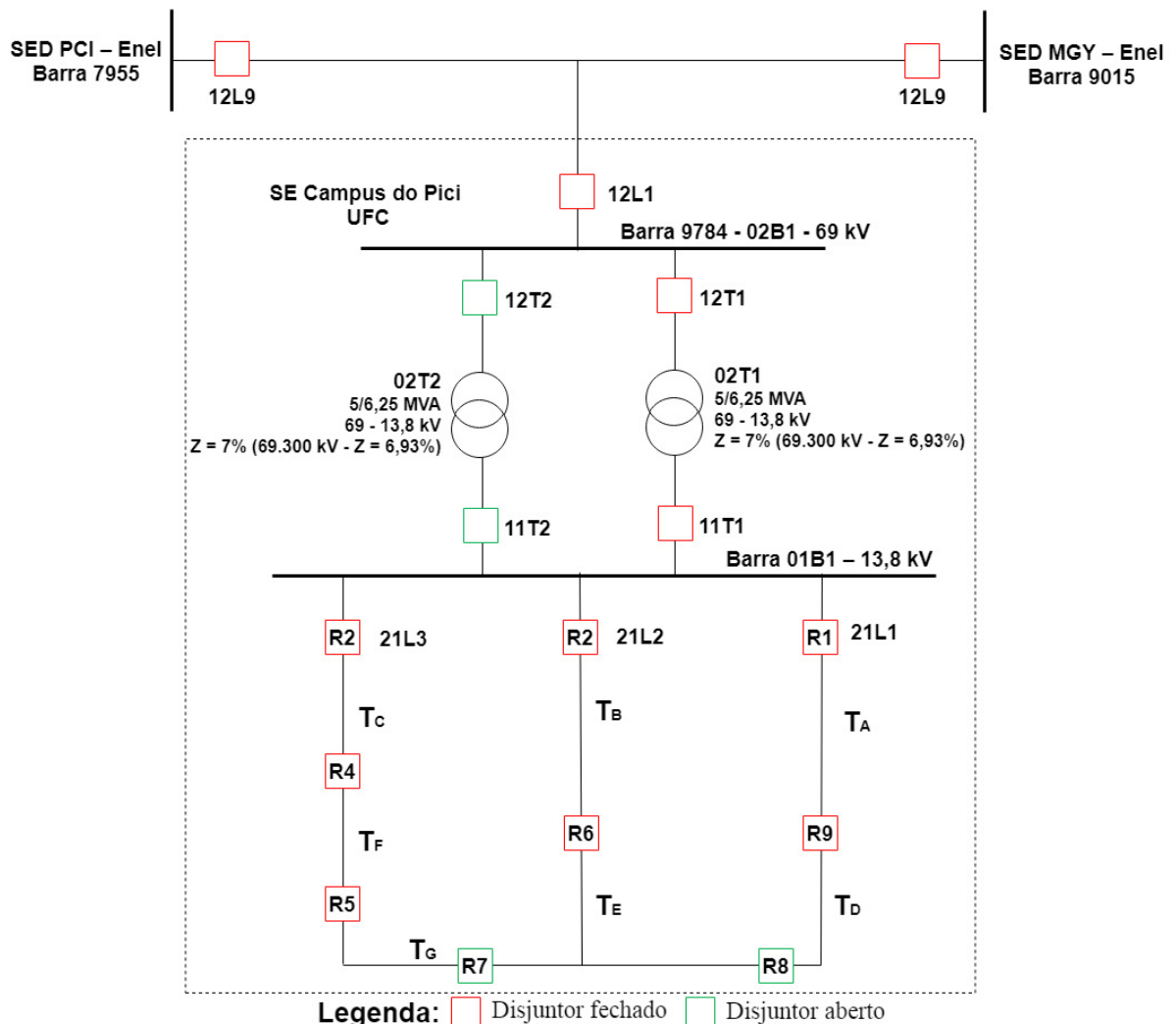
Além disso, o *Elipse Simulator Modbus* permite a definição de até 255 dados do tipo *coil* – permitindo a execução das funções 0x01 e 0x05 do protocolo *Modbus* – e de até 255

dados do tipo *register* – permitindo a execução das funções 0x03, 0x06 e 0x16 do protocolo *Modbus*. (Elipse, 2013). Nesse contexto, o *Elipse Simulator Modbus* será utilizado para simular a operação de um relé de proteção.

### 3.3.4 Subestação Campus do Pici

A Subestação *Campus* do Pici, cujo diagrama unifilar está apresentado na Figura 9, faz parte do sistema elétrico do *Campus* do Pici da Universidade Federal do Ceará (UFC).

Figura 9 - Diagrama unifilar da subestação *Campus* do Pici



Fonte: Vieira (2016, com adaptações).

De acordo com Lima (2019, p. 47), e conforme observação do diagrama unifilar apresentado, a Subestação *Campus* do Pici é composta por:

- Vão de entrada de linha em 69 kV;

- Uma barra simples (02B1) de 69 kV;
- Dois vãos de transformação, consistindo dos transformadores 02T1 (em operação) e 02T2 (reserva), ambos com potência de 5/6,25 MVA, os quais são protegidos pelos relés associados aos disjuntores 12T1 e 11T1 (no caso do transformador 12T1);
- Uma barra simples (01B1) de 13,8 kV;
- Três vãos de saída de alimentadores, os quais são protegidos pelos seus respectivos religadores.

O sistema de distribuição em 13,8 kV possui topologia radial com recurso, configurada pelos religadores R7 e R8, os quais são normalmente abertos. (LIMA, 2019, p. 48).

### 3.4 Considerações finais

Neste capítulo, é apresentada a proposta de uma Metodologia de Aprendizagem Baseada em Projeto aplicada na construção do sistema SCADA. A metodologia ABP proposta foi aplicada no desenvolvimento do sistema SCADA de uma subestação, utilizando o *software* ScadaBR, apresentado no capítulo 4. Vale destacar que a metodologia proposta é adequada para aplicação no desenvolvimento de qualquer sistema supervisão.

Além da metodologia ABP, foram apresentadas as ferramentas para elaboração do sistema supervisão, que são: o *software* ScadaBR, o protocolo de comunicação *Modbus*, o simulador *Elipse Modbus Simulator*. Além disso, foi efetuada uma breve descrição acerca do unifilar da Subestação *Campus* do Pici, o qual foi utilizado como base para elaboração das telas de supervisão.

Constata-se que o ScadaBR possui diversos recursos, mostrando-se bastante aplicável para o desenvolvimento de um sistema supervisão de uma subestação.

O simulador, embora possua limitações, mostrou-se bastante aplicável para o desenvolvimento deste trabalho, e pode ser devidamente utilizado como um ferramenta didática, de aplicação do protocolo *Modbus* e de testes para o sistema supervisão.

## **4 APLICAÇÃO DA METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETO UTILIZANDO O SCADABR**

### **4.1 Introdução**

Nesse capítulo é apresentada a Metodologia de Aprendizagem Baseada em Projeto proposta aplicada ao ScadaBR para utilização na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência. Além disso, é mostrado o desenvolvimento do sistema SCADA seguindo-se a referida metodologia, sendo descritos: as configurações realizadas para efetuar a comunicação de dados entre o ScadaBR e o simulador *Modbus*; as definições dos dados a serem escritos e lidos; o tratamento lógico desses dados e determinação dos alarmes; e o desenvolvimento das telas, abordando, inclusive, o desenvolvimento de códigos em *javascript* e *HyperText Markup Language* (HTML) para criação de botões de comando e botões de mudança de tela.

### **4.2 Metodologia de aprendizagem baseada em projeto para supervisório de uma subestação**

Nessa seção é apresentada a Metodologia de Aprendizagem Baseada em Projeto proposta na seção 3.6 aplicada ao ScadaBR como atividade prática da disciplina de Supervisão e Controle de Sistemas Elétricos de Potência.

#### ***4.2.1 Etapa 1 - estudos preliminares***

##### ***4.2.1.1 Objetivo geral***

Essa etapa objetiva o aluno conhecer e descrever o *software* ScadaBR e seus recursos, os principais protocolos de comunicação implementados no ScadaBR, os *Intelligent Eletronic Devices* (IEDs) utilizados e os dados proporcionados por este equipamento para comunicação.

##### ***4.2.1.2 Objetivos específicos***

Os objetivos específicos acerca dessa etapa consistem em o aluno aprender a:

- Conhecer os recursos disponibilizados pelo ScadaBR e os protocolos de comunicação suportados;

- Realizar o levantamento de dados disponibilizados pelos IEDs utilizados;
- Especificar os dados que serão utilizados para monitoramento e comando pelo ScadaBR;
- Definir o protocolo de comunicação a ser utilizado.

#### *4.2.1.3 Metas – produtos da etapa*

Nessa etapa, os alunos deverão:

- Apresentar a especificação dos pontos analógicos e digitais dos IEDs a estarem presentes no supervísório;
- Definir o protocolo de comunicação a ser utilizado;
- Sistema SCADA instalado.

Esses produtos devem ser entregues na forma de um relatório ou de apresentação de seminário acerca dos pontos mencionados.

### **4.2.2 Etapa 2 - integração do sistema SCADA com os IEDs**

#### *4.2.2.1 Objetivo geral*

Essa etapa objetiva o aluno compreender e efetuar a comunicação do ScadaBR com os IEDs utilizados, e compreender e implementar o tratamento lógico dos dados.

#### *4.2.2.2 Objetivos específicos*

Os objetivos específicos acerca dessa etapa consistem em o aluno aprender a:

- Efetuar as configurações necessárias no ScadaBR para a comunicação entre os IEDs e o *software* seja realizada;
- Elaborar o tratamento lógico dos dados;
- Implementar esse tratamento lógico por meio dos detectores de eventos, tratadores de eventos e dos *scripts* do ScadaBR;
- Elaborar as *watch lists* no ScadaBR.

### *4.2.2.3 Metas – produtos da etapa*

Ao final dessa etapa, os alunos devem apresentar os seguintes produtos:

- Pontos analógicos e digitais dos IEDs configurados no Sistema SCADA;
- Fluxograma e descrição das lógicas implementadas no SCADA;
- Códigos desenvolvidos para implementações de lógicas;
- Apresentação de gráficos para demonstração das lógicas implementadas para os dados;

Os pontos analógicos e digitais devem ser apresentados na forma de um vídeo mostrando a lista de dados a serem trabalhados, sendo aplicado comandos para as variáveis de leitura e escrita. Os fluxograma, códigos e gráficos devem ser apresentados na forma de relatório ou de seminário.

## ***4.2.3 Etapa 3 – elaboração das telas de supervisão***

### *4.2.3.1 Objetivo geral*

O objetivo dessa etapa é os alunos elaborarem as telas de supervisão utilizando os componentes disponibilizados pelo ScadaBR.

### *4.2.3.2 Objetivos específicos*

Os objetivos específicos acerca dessa etapa consistem em o aluno aprender a:

- Adicionar as telas de supervisão;
- Desenhar a imagem de fundo para a tela de supervisão;
- Compreender e utilizar os componentes “HTML” e “Script para servidor” para elaboração de códigos a fim de se confeccionar botões, listas de seleção e caixas de texto;
- Criar componentes gráficas nas telas de supervisão;
- Utilizar componentes de alarme nas telas;



#### *4.2.3.3 Metas – produtos da etapa*

Ao final dessa etapa, os alunos devem apresentar os seguintes produtos:

- Desenho da imagem de fundo das telas;
- Códigos desenvolvidos para confecção dos componentes utilizados nas telas – botões, listas de seleção e caixas de texto;
- Telas de supervisão e sistema supervisório concluídos.

Os desenhos e códigos devem ser apresentados na forma de relatório, com um detalhamento desse desenvolvimento. As telas de supervisão devem ser apresentadas na forma de um vídeo, sendo mostrado o funcionamento do sistema supervisório como um todo.

#### *4.2.4 Formas de avaliação*

As formas de avaliação devem ser definidas detalhadamente pelo professor que aplicará a metodologia, e deve levar em conta o dinamismo das aulas ministradas.

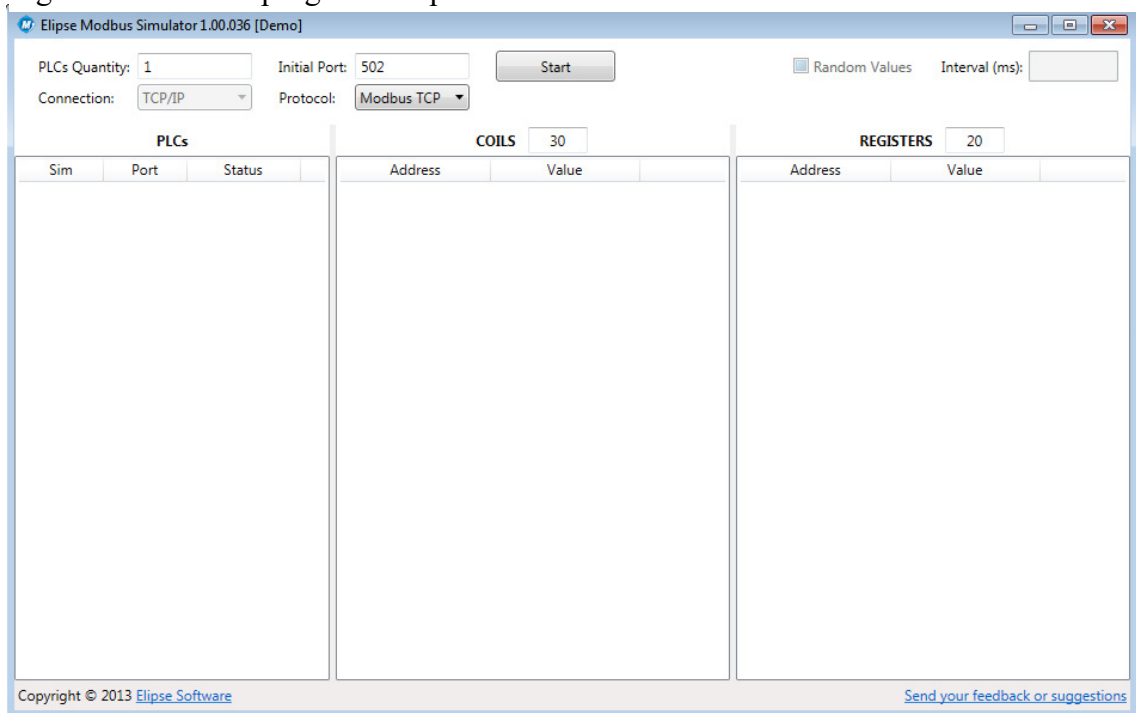
### **4.3 Desenvolvimento do sistema SCADA**

A seguir é apresentada as etapas de desenvolvimento do sistema supervisório de uma subestação de energia, utilizando-se o ScadaBR e seguindo-se a metodologia apresentada.

#### *4.3.1 Configurações de comunicação*

##### *4.3.1.1 Ajustes efetuados no Elipse Modbus Simulator*

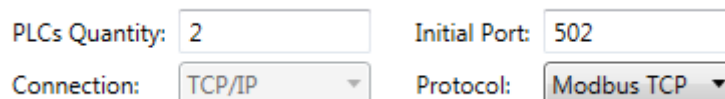
Inicialmente, será efetuada a configuração diretamente no simulador de protocolo *Modbus* da Elipse. Ao se abrir o programa, a tela mostrada na Figura 10 é exibida.

Figura 10 - Tela do programa Elipse *Modbus Simulator*

Fonte: Elipse, 2013.

Na parte superior da tela mostrada na Figura 10 é possível ajustar a quantidade de dispositivos simulados no campo “*PLCs Quantity*”, a porta inicial do protocolo de comunicação em “*Initial Port*” e se o *software* irá efetuar a simulação do protocolo MODBUS de forma serial (RTU) ou por conexão *ethernet* (TCP). Além disso, é possível gerar valores aleatórios com a caixa de seleção “*Random Values*”, e o intervalo de geração desses valores na caixa de texto “*Interval (ms)*”. Ademais, é possível escolher a quantidade de *coils* (grandezas binárias) e de registradores (grandezas analógicas) as quais servem para leitura e escrita, e o botão “*Start*” torna o *software* disponível para comunicação. Nesse contexto, os ajustes foram efetuados conforme a Figura 11.

Figura 11 - Ajustes efetuados no simulador

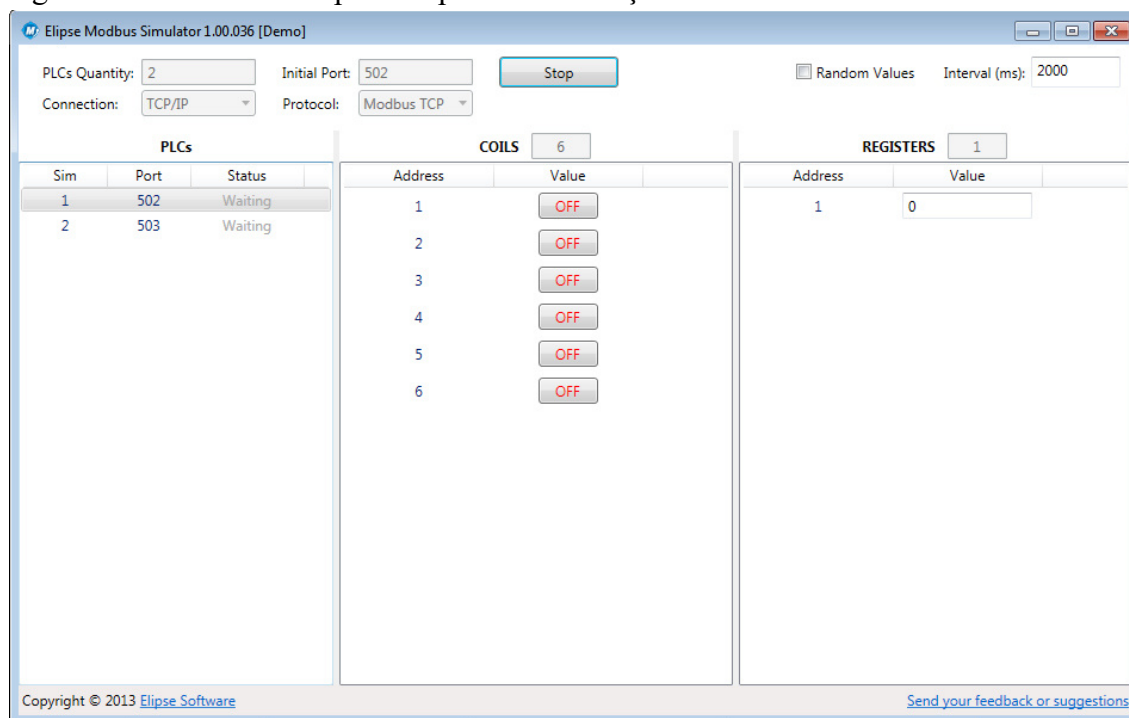


Fonte: Elipse, 2013.

Foram escolhidos dois dispositivos no *software*, de forma que seja simulada a atuação de dois relés distintos. Além disso, foi escolhida a forma TCP do protocolo *MODBUS* e a porta inicial foi estabelecida como a porta padrão 502. Outrossim, foram definidos 6 *coils* e 1 registrador para cada um dos dispositivos simulados, de forma que sejam possibilitados o

envio de comandos – abertura/fechamento de disjuntor e mudança de grupo de ajuste – e leitura de dados – estado do disjuntor e nível de tensão. Após os ajustes, clicou-se no botão “*Start*” e o simulador ficou disponível para comunicação conforme a Figura 12.

Figura 12 - Simulador disponível para comunicação



Fonte: Elipse, 2013.

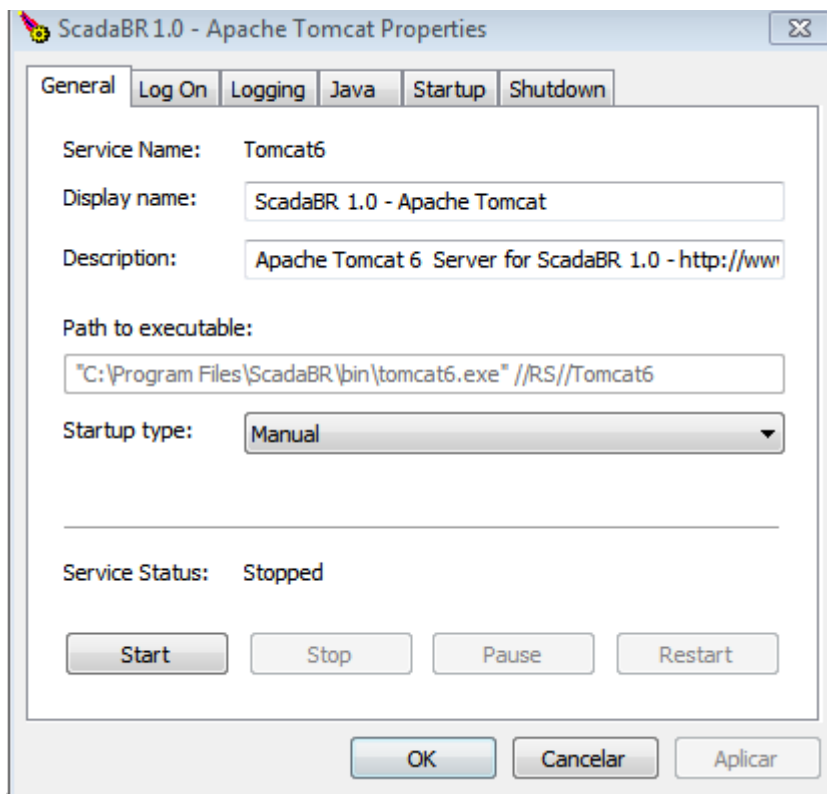
Na tela são exibidos os *coils* e registradores com seus endereços de referência por cada dispositivo, o qual pode ser selecionado na coluna PLCs, além de ser exibida a respectiva porta do IED simulado. Os *coils* podem ser alterados diretamente no simulador clicando-se nos botões “*OFF*”, e os registradores podem ser alterados inserindo-se um valor numérico na coluna *value*. Por padrão, ao se iniciar a simulação, o IP identificador do simulador é o *localhost* (127.0.0.1), visto que o programa está sendo executado no próprio computador que está o *software* ScadaBR.

#### 4.3.1.2 Ajustes efetuados no ScadaBR

Após os ajustes efetuados no simulador, fez-se necessário ajustar a comunicação no *software* ScadaBR. Antes de utilizar o ScadaBR, é necessário iniciar o servidor *Apache Tomcat*, executando o programa *Configure Tomcat* como administrador. Ao se executar o programa, a

tela conforme a Figura 13 é exibida. Clicando-se no botão “*Start*” o servidor é iniciado, e o ScadaBR pode ser aberto.

Figura 13 - Servidor *Apache Tomcat*

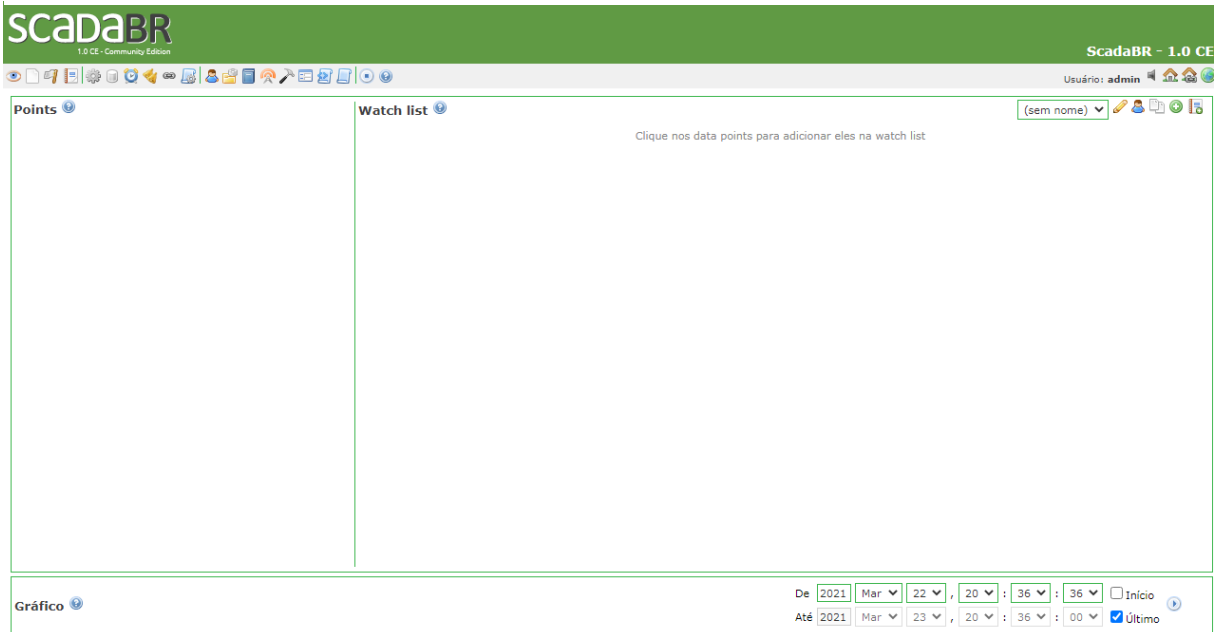


Fonte: *Apache Tomcat*.

Após iniciar o servidor, pode-se executar o ScadaBR, o qual irá solicitar o *login* do usuário na primeira tela. Por padrão, o usuário é *admin* e a senha é *admin*.

Em seguida, após o *login*, a tela conforme a Figura 14 é exibida.

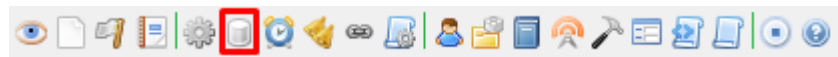
Figura 14 - Tela inicial do ScadaBR



Fonte: ScadaBR.

Na parte superior da tela, é exibida uma barra de ferramentas conforme a Figura 15.

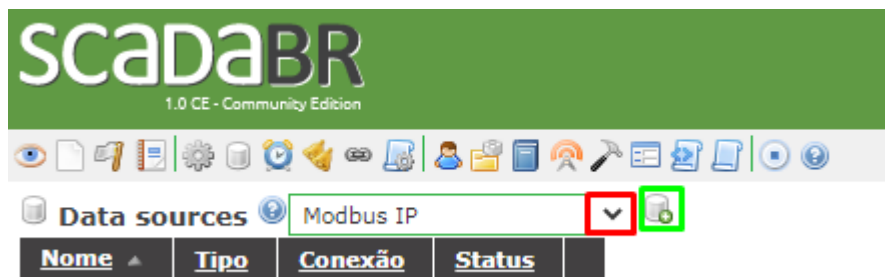
Figura 15 - Barra de ferramentas do ScadaBR (*data source* em destaque)



Fonte: ScadaBR

Ao clicar no ícone em destaque na Figura 15, é exibida a tela de *data source* conforme a Figura 16.

Figura 16 - Tela para visualização e adição do *data source*



Fonte: ScadaBR.

Na seta destacada em vermelho na Figura 16, selecionou-se o protocolo *Modbus IP*, e, em seguida, clicou-se no ícone destacado em verde na Figura 16 para configuração do

primeiro *data source*, o qual é o dispositivo simulado pelo *software* *Elipse Modbus Simulator* de porta *Modbus 502*.

Após isso, a tela de configuração das propriedades do *data source* foi exibida, e a configuração do primeiro dispositivo foi ajustada de acordo com a Figura 17. O dispositivo foi nomeado como Relé 1, e é a simulação do relé associado ao disjuntor 11T1 do diagrama unifilar.

Figura 17 - Ajustes das propriedades do *Modbus IP* do primeiro dispositivo no ScadaBR

Propriedades do modbus IP	
Nome	Relé 1
Export ID (XID)	DS_979868
Período de atualização	200 milissegundo(ms)
Quantificação	<input type="checkbox"/>
Timeout (ms)	500
Tentativas	2
Apenas quantidades contínuas	<input type="checkbox"/>
Criar pontos de monitor de escravo	<input type="checkbox"/>
Máxima contagem de leitura de bits	2000
Máxima contagem de leitura de registradores	125
Máxima contagem de escrita de registradores	120
Tipo de transporte	TCP
Host	localhost
Porta	502
Encapsulado	<input type="checkbox"/>
<b>Níveis de alarme de eventos</b>	
Exceção de data source	Urgente
Exceção de leitura de data point	Urgente
Exceção de escrita em data point	Urgente

Fonte: ScadaBR.

É importante ressaltar que os principais pontos de ajustes são o nome do dispositivo, o período de atualização, o *host* e a porta do dispositivo. Conforme mencionado anteriormente, o simulador, por *default*, possui o IP identificador *localhost* (127.0.0.1) e a porta do primeiro dispositivo, a qual é exibida no simulador, é a 502. O restante das configurações foram mantidas de acordo com a padronização do ScadaBR. Após efetuados os ajustes, clica-se no ícone do disquete no canto superior direito.

Em seguida, o mesmo procedimento foi repetido para se efetuar as definições do segundo *data source*, o qual é o dispositivo simulado pelo *software Elipse Modbus Simulator* de porta *Modbus* 503. Os ajustes estão exibidos na Figura 18.

Figura 18 - Ajustes das propriedades do *Modbus* IP do segundo dispositivo no ScadaBR

**Propriedades do modbus IP**

Nome: Relé 2

Export ID (XID): DS\_463261

Período de atualização: 200 milissegundo(ms)

Quantificação:

Timeout (ms): 500

Tentativas: 2

Apenas quantidades contínuas:

Criar pontos de monitor de escravo:

Máxima contagem de leitura de bits: 2000

Máxima contagem de leitura de registradores: 125

Máxima contagem de escrita de registradores: 120

Tipo de transporte: TCP

Host: localhost

Porta: 503

Encapsulado:

**Níveis de alarme de eventos**

Exceção de data source: Urgente

Exceção de leitura de data point: Urgente

Exceção de escrita em data point: Urgente

Fonte: ScadaBR.

Em seguida, deve-se abrir a tela de *data source* (Figura 16) para habilitar os dois dispositivos, clicando-se nos ícones em destaque na Figura 19.

Figura 19 - Habilitação dos *data sources*

Nome	Tipo	Conexão	Status	
Relé 1	Modbus IP	localhost:502		
Relé 2	Modbus IP	localhost:503		

Fonte: ScadaBR.

O ScadaBR gera para cada *data source* o *Export ID* (XID), o qual serve como identificador do dispositivo na elaboração dos códigos (*scripts*) lógicos em *javascript* e em HTML.

#### 4.3.2 Definições dos dados a serem lidos e comandados











Nos relés digitais de sobrecorrente, as ações de abertura/fechamento de disjuntor e mudança de grupo de ajuste são definidas por equações booleanas. As expressões lógicas tomam como base variáveis internas do relé que indicam a atuação de cada função de proteção do relé ou que são variáveis utilizadas para o envio de comandos remotos. Essas variáveis possuem um endereço de memória específico para acesso pelo *software* SCADA por meio dos protocolos de comunicação. Além disso, existem variáveis internas do relé para leitura de estado de disjuntor e de níveis de tensão, as quais possuem portas de entrada associadas.

Nesse contexto, definiu-se a utilização de 6 variáveis de natureza digital e 1 variável de natureza analógica, com as seguintes funções:

- 1 variável digital para simulação de leitura do estado do disjuntor, correspondendo ao endereço 1 dos *coils* no simulador;
- 3 variáveis digitais para simulação de envio de comando de mudança de 3 grupos de ajuste, correspondendo aos endereços 2, 3 e 4 dos *coils* no simulador;
- 1 variável digital para simulação de envio de comando de abertura de disjuntor, correspondendo ao endereço 5 dos *coils* no simulador;
- 1 variável digital para simulação de envio de comando de fechamento do disjuntor, correspondendo ao endereço 6 dos *coils* no simulador;
- 1 variável analógica para simulação de leitura de nível de tensão, correspondendo ao endereço 1 dos *registers* no simulador.

Diante dessas definições, é necessário adicionar os referidos dados para cada *data source* no ScadaBR. Esses dados são denominados de *data points*. Para adicioná-los, é necessário abrir a tela de *data source* e clicar no ícone em destaque, de acordo com a Figura 20.

Figura 20 - Tela de *data source* (adição de *data points*)

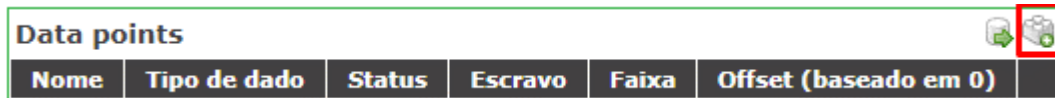
Nome ▲	Tipo	Conexão	Status	
Relé 1	Modbus IP	localhost:502		   
Relé 2	Modbus IP	localhost:503		   

Fonte: ScadaBR.



Ao se clicar no ícone em destaque, será exibida uma tela contendo o quadro de propriedades do respectivo *data source* (Figura 17 e Figura 18) e um quadro para adição de *data points* na parte inferior, de acordo com o apresentado na Figura 21.

Figura 21 - Adição de *data points*



Fonte: ScadaBR.

Ao se clicar no ícone em destaque na Figura 21, aparecerá um quadro para configuração dos *data points* do respectivo *data source* escolhido, conforme a Figura 22.

Figura 22 - Detalhes do *data point*

Fonte: ScadaBR.

É importante ressaltar que o campo “Faixa do registro” diz respeito à definição de 4 tipos de dados do protocolo *Modbus*:

- *Status do coil* – grandezas binárias que suportam leitura e escrita;
- Status de entrada – grandezas binárias que suportam apenas leitura;
- Registrador *holding* – grandezas analógicas que suportam leitura e escrita;
- Registrador de entrada - grandezas analógicas que suportam apenas leitura.

O Elipse *Modbus Simulator* disponibiliza apenas dados de leitura e escrita. Portanto, os dados devem ser configurados apenas como *status do coil* ou como registrador *holding*.

Além disso, os principais pontos de ajustes são o nome do *data point*, a faixa do registro, o *offset* (baseado em 0) – o qual diz respeito ao endereço do dado, e contagem iniciada em 0 –, o tipo de dados *Modbus* – o qual é escolhido apenas para registradores, para designar o tipo de grandeza analógica, em termos de quantidade de bytes, de sinal, de ser variável com ponto flutuante –, e a opção “Configurável” – a qual permite que o dado seja comandado pelo ScadaBR. O restante das configurações permanece com os ajustes padronizados do ScadaBR.

Para estabelecer o endereço do dado é necessário subtrair uma unidade do endereço fornecido pelo simulador, visto que sua contagem inicia em 1 e a contagem do ScadaBR inicia em 0.

#### 4.3.2.1 Estado do disjuntor

Apesar de ser um dado utilizado somente para leitura, essa variável será definida como um dado configurável, de forma que essa possa ser comandada para ser elaborada uma lógica no ScadaBR por meio de *scripts* em *javascript* e HTML, que simulem o comportamento do disjuntor ao receber um comando de abertura/fechamento.

Nesse contexto, para o *data source* Relé 1, os ajustes foram efetuados conforme a Figura 23. Para salvar os ajustes, clica-se no ícone de um disquete no canto superior direito.

Figura 23 - Definição do *data point* Estado do Disjuntor (11T1)

Detalhes do data point	
Nome	Estado do Disjuntor (11T1)
Export ID (XID)	DP_217924
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	0
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

De forma análoga, os ajustes dessa variável para o Relé 2 foram efetuados, conforme a Figura 24.

Figura 24 - Definição do *data point* Estado do Disjuntor (12T1)

The screenshot shows a configuration window titled 'Detalhes do data point' with a help icon and a save icon. The fields are as follows:

Nome	Estado do Disjuntor (12T1)
Export ID (XID)	DP_621788
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	0
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

#### 4.3.2.2 Mudança de grupo de ajuste

As 3 variáveis para mudança de grupo de ajuste do relé 1 foram definidas conforme a Figura 25, a Figura 27 e a Figura 26.

Figura 25 - Definição do *data point* Ajuste 1 (11T1)

The screenshot shows a configuration window titled 'Detalhes do data point' with a help icon and a save icon. The fields are as follows:

Nome	Ajuste 1 (11T1)
Export ID (XID)	DP_346567
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	1
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 27 - Definição do *data point* Ajuste 2 (11T1)

Detalhes do data point	
Nome	Ajuste 2 (11T1)
Export ID (XID)	DP_132017
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	2
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 26 - Definição do *data point* Ajuste 3 (11T1)

Detalhes do data point	
Nome	Ajuste 3 (11T1)
Export ID (XID)	DP_588958
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	3
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

De forma análoga, os ajustes foram efetuados para as variáveis de mudança de grupo de ajuste para o relé 2, conforme a Figura 28, a Figura 29 e a Figura 30.

Figura 28 - Definição do *data point* Ajuste 1 (12T1)

Detalhes do data point	
Nome	Ajuste 1 (12T1)
Export ID (XID)	DP_679190
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	1
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 29 - Definição do *data point* Ajuste 2 (12T1)

Detalhes do data point	
Nome	Ajuste 2 (12T1)
Export ID (XID)	DP_463074
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	2
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 30 - Definição do *data point* Ajuste 3 (12T1)

Detalhes do data point	
Nome	Ajuste 3 (12T1)
Export ID (XID)	DP_993657
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	3
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Essas variáveis de mudança de grupo de ajuste serão utilizadas na elaboração de *scripts* no ScadaBR para cada *data source* separadamente, visto que ao se comandar uma das três variáveis de mudança de ajuste para o valor binário 1, as outras duas devem ser forçadas para valor 0.

#### 4.3.2.3 Abertura e fechamento de disjuntor

A Figura 31 e a Figura 32 apresentam a definição, respectivamente, dos *data points* de abertura e fechamento de disjuntor para o Relé 1.

Figura 31 - Definição do *data point* Abertura do Disjuntor (11T1)

Detalhes do data point	
Nome	Abertura do Disjuntor (11T1)
Export ID (XID)	DP_332331
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	4
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 32 - Definição do *data point* Fechamento do Disjuntor (11T1)

Detalhes do data point	
Nome	Fechamento do Disjuntor (1
Export ID (XID)	DP_016874
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	5
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

De forma análoga para o Relé 2, na Figura 33 e na Figura 34 são apresentadas as definições dos *data points* para, respectivamente, abertura e fechamento do disjuntor.

Figura 33 - Definição do *data point* Abertura do Disjuntor (12T1)

Detalhes do data point	
Nome	Abertura do Disjuntor (12T1
Export ID (XID)	DP_076324
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	4
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Figura 34 - Definição do *data point* Fechamento do Disjuntor (12T1)

Detalhes do data point	
Nome	Fechamento do Disjuntor (1)
Export ID (XID)	DP_878330
Id do escravo	1
Faixa do registro	Status do coil
Tipo de dados modbus	Binário
Offset (baseado em 0)	5
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input checked="" type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

Por meio de detectores e tratadores de evento, as variáveis de abertura e fechamento serão configuradas para efetuar o comando em um pulso de 1 segundo, visto que após a abertura ou fechamento do disjuntor fisicamente, não é necessário que as variáveis permaneçam em 1.

#### 4.3.2.4 Nível de tensão

Nesse caso, o tipo de grandeza será analógico, e somente para leitura, o que é representado pelo tipo de dado: Registrador de Entrada. A Figura 35 e a Figura 36 mostram a definição desse dado para, respectivamente, o Relé 1 (ligado diretamente ao barramento 01B1) e para o Relé 2 (ligado diretamente ao barramento 02B1).

Figura 35 - Definição do *data point* Nível de Tensão (01B1)

Detalhes do data point	
Nome	Nível de Tensão (01B1)
Export ID (XID)	DP_235409
Id do escravo	1
Faixa do registro	Registrador holding
Tipo de dados modbus	Inteiro de 2 bytes sem sinal
Offset (baseado em 0)	0
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.



Figura 36 - Definição do *data point* Nível de Tensão (02B1)

Nome	Nível de Tensão (02B1)
Export ID (XID)	DP_382410
Id do escravo	1
Faixa do registro	Registrador holding
Tipo de dados modbus	Inteiro de 2 bytes sem sinal
Offset (baseado em 0)	0
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input type="checkbox"/>
Multiplicador	1
Aditivo	0

Fonte: ScadaBR.

#### 4.3.2.5 Habilitação dos *data points*

Após adicionados os *data points* para cada dispositivo, deve-se habilitá-los de forma que seja possibilitada a escrita e leitura desses dados. Conforme a Figura 37, na mesma tela em que se adiciona os *data points* para o Relé 1, além de ser possível visualizar os dados adicionados, é possível habilitá-los individualmente – clicando nos ícones com destaque em verde – ou em sua totalidade – clicando-se no ícone com destaque em vermelho. O mesmo procedimento deve ser efetuado para os dados adicionados no Relé 2.

Figura 37 - Habilitação dos *data points*

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
Abertura do Disjuntor (11T1)	Binário		1	Status do coil	4
Ajuste 1 (11T1)	Binário		1	Status do coil	1
Ajuste 2 (11T1)	Binário		1	Status do coil	2
Ajuste 3 (11T1)	Binário		1	Status do coil	3
Estado do Disjuntor (11T1)	Binário		1	Status do coil	0
Fechamento do Disjuntor (11T1)	Binário		1	Status do coil	5
Nível de Tensão (01B1)	Numérico		1	Registrador de entrada	0

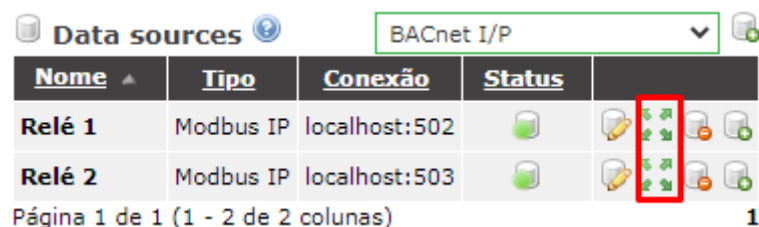
Fonte: ScadaBR.

#### 4.3.2.6 Detalhes dos data points

Para *data point* adicionado, o ScadaBR gera automaticamente o *Export ID* (XID), o qual pode ser observado na tela de configuração do *data point* (Figura 22). O *Export ID* serve como identificador do *data point*, podendo ser utilizado na elaboração dos *scripts* no *software*.

Além disso, outro identificador gerado pelo ScadaBR para o *data point* é o “*dpid*”, o qual é um número que fica disponível no *link* do *data point* no *browser* em que se está executando o ScadaBR. Para ter acesso a esse link, deve-se abrir a tela de *data source*, e clicar no ícone em destaque na Figura 38.

Figura 38 - Acesso ao código "dpid" do *data point*



Nome	Tipo	Conexão	Status	
Relé 1	Modbus IP	localhost:502	OK	[ícone de edição] [ícone de status] [ícone de dpid] [ícone de ajuda]
Relé 2	Modbus IP	localhost:503	OK	[ícone de edição] [ícone de status] [ícone de dpid] [ícone de ajuda]

Página 1 de 1 (1 - 2 de 2 colunas) 1

Fonte: ScadaBR.

Em seguida, são exibidos os *data points* do respectivo *data source*, conforme a Figura 39.

Figura 39 - Exibição dos *data points*



Nome	Tipo	Conexão	Status	
Relé 1	Modbus IP	localhost:502	OK	[ícone de edição] [ícone de status] [ícone de dpid] [ícone de ajuda]

Nome do data point	Descrição	Status	
Abertura do Disjuntor (11T1)	Binário / Id do escravo 1, offset 4	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Ajuste 1 (11T1)	Binário / Id do escravo 1, offset 1	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Ajuste 2 (11T1)	Binário / Id do escravo 1, offset 2	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Ajuste 3 (11T1)	Binário / Id do escravo 1, offset 3	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Estado do Disjuntor (11T1)	Binário / Id do escravo 1, offset 0	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Fechamento do Disjuntor (11T1)	Binário / Id do escravo 1, offset 5	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]
Nível de Tensão (01B1)	Numérico / Id do escravo 1, offset 0	OK	[ícone de status] [ícone de dpid] [ícone de ajuda]

Fonte: ScadaBR.

Ao se clicar no ícone com destaque em vermelho na Figura 39, é exibida a tela de propriedades do respectivo *data point* conforme a Figura 40.

Figura 40 - Tela de propriedades do *data point*

Fonte: ScadaBR.

O *link* que é exibido no *browser* ao se abrir essa tela é da seguinte forma: [http://localhost:8080/ScadaBR/data\\_point\\_edit.shtm?dpid=16](http://localhost:8080/ScadaBR/data_point_edit.shtm?dpid=16). Dessa forma, o “dpid” é o número que está no final do endereço após “dpid=”. Nessa tela, também é possível associar detectores de eventos ao respectivo *data point*.

### 4.3.3 Watch list

O ScadaBR permite o envio de comandos e a visualização em tempo real dos dados adicionados e habilitados no formato de lista por meio das *Watch List*. Além disso, é possível verificar o horário de atualização dos *data points*. No caso, serão criadas duas *watch lists*:

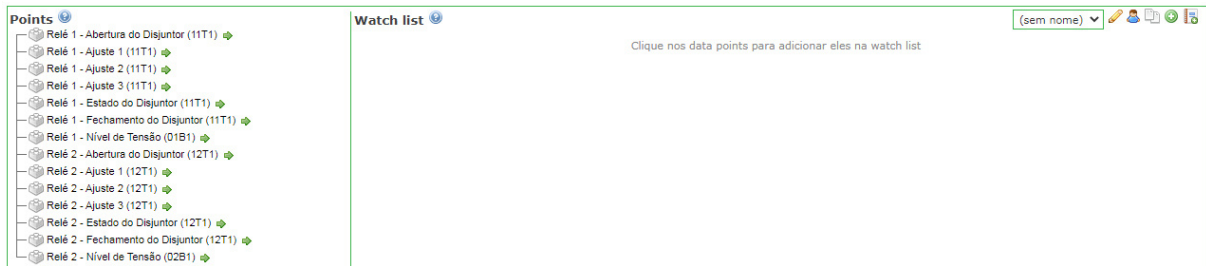
- *Watch list* referentes aos dados associados ao *data source* Relé 1;
- *Watch list* referentes aos dados associados ao *data source* Relé 2;

Para configurar essas listas, deve-se clicar no ícone da barra de ferramentas em destaque na Figura 41.

Figura 41 - Ícone para exibição da tela de *watch list*

Fonte: ScadaBR.

Em seguida, é exibida a tela conforme a Figura 42. Nessa tela, é exibida uma *watch list* vazia e um quadro contendo os *data points* adicionados no ScadaBR de todos os dispositivos, os quais podem ser adicionados à *watch list*.

Figura 42 - Tela *watch list*

Fonte: ScadaBR.

A Figura 43 mostra a barra de ferramentas da *watch list*, localizada no canto superior direito, a qual possibilita a seleção da *watch list* a ser visualizada – por meio da caixa de opções –, a edição do nome da *watch list* – clicando-se no ícone destacado em verde –, e a criação de uma nova *watch list* – clicando-se no ícone destacado em vermelho.

Figura 43 - Barra de ferramentas da *watch list*

Fonte: ScadaBR.

Diante disso, será renomeada a *watch list* vazia para “Relé 1”, e será criada uma nova *watch list* que será nomeada de “Relé 2”. Além disso, conforme mencionado, na tela de *watch list* é exibido um quadro com todos os *data points* adicionados no ScadaBR, conforme a Figura 44. Para adicioná-los, é necessário clicar na seta verde ao lado do nome do *data point*.

Figura 44 - Inserção dos *data points* na *watch list*

Fonte: ScadaBR.

Com isso, as variáveis correspondentes ao *data source* Relé 1 serão adicionadas na tela Relé 1, e as variáveis associadas ao *data source* Relé 2 serão adicionadas na tela Relé 2. A Figura 45 apresenta a *watch list* Relé 1, enquanto a Figura 48 apresenta a do Relé 2.

Figura 45 - *Watch list* Relé 1

Watch list		Relé 1				
Relé 1 - Estado do Disjuntor (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Abertura do Disjuntor (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Fechamento do Disjuntor (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Ajuste 1 (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Ajuste 2 (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Ajuste 3 (11T1)	0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 1 - Nível de Tensão (01B1)	0.0	18:34:11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: ScadaBR.

Figura 46 - *Watch list* Relé 2

Watch list		Relé 2				
Relé 2 - Estado do Disjuntor (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Abertura do Disjuntor (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Fechamento do Disjuntor (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Ajuste 1 (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Ajuste 2 (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Ajuste 3 (12T1)	0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Relé 2 - Nível de Tensão (02B1)	0.0	18:40:38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: ScadaBR.

Para se efetuar o comando para os dados, deve-se passar a seta do *mouse* no ícone de ferramenta localizado à esquerda do nome do *data point*. Com isso, surgirá uma caixa de texto para definição do valor a ser escrito na variável, conforme a Figura 47. Após a definição, deve-se clicar na opção “Definir”.

Figura 47 - Definição de valor para escrita

Definir valor para escrita:  Definir

Fonte: ScadaBR.

Além disso, na parte inferior da tela de *watchlist* é possível gerar gráficos em um intervalo de tempo definido. Para isso, deve-se selecionar os dados de interesse na visualização por meio das caixas de seleção, as quais estão localizadas ao lado do horário de atualização de cada *data point*. Em seguida, deve-se definir o horário de início da geração do gráfico – podendo

ser selecionada a opção “início” para gerar desde o início da definição e habilitação do data source –, e o horário final da geração do gráfico – podendo ser selecionada a opção “último” para o gráfico ser gerado até o instante em que se realiza o comando para geração do gráfico. A mostra onde deve ser realizada essas definições de horários, e ao clicar no botão em destaque, o gráfico é elaborado.

Figura 48 - Definição e geração do gráfico na *watch list*

De 2021 Mar 28 , 15 : 38 : 00  Início 

Até 2021 Mar 28 , 15 : 40 : 00  Último

Fonte: ScadaBR.

#### 4.3.4 Desenvolvimento de scripts

O ScadaBR possibilita a criação de códigos em *javascript*, os quais podem executar ações nos *data sources* – habilitando-os ou desabilitando-os – ou nos *data points* – podendo alterar seus valores. Os *scripts* podem ser executados por meio de botões adicionados nas telas do supervisor e de detectores e tratadores de eventos, os quais, com base na ocorrência de um evento com critérios definidos pelo usuário do ScadaBR, podem comandar a execução do *script* como uma forma de tratar esse evento.

No desenvolvimento do trabalho, foram criados 7 *scripts* sendo:

- 3 *scripts* de mudança de grupo de ajustes para cada ajuste referente ao *data source* Relé 1;
- 3 *scripts* de mudança de grupo de ajustes para cada ajuste referente ao *data source* Relé 2;

Para se adicionar um *script*, deve-se clicar no ícone com destaque em vermelho da barra de ferramentas do ScadaBR, conforme a Figura 49.

Figura 49 - Barra de ferramentas do ScadaBR (*scripting*)



Fonte: ScadaBR.

Em seguida, será aberta a tela para edição do script, conforme a Figura 50. Para se abrir o quadro “Detalhes do *script*” deve-se clicar no ícone com destaque em vermelho.

Figura 50 - Tela de adição de um *script*

The screenshot shows a web-based interface for adding a script. The main window is titled 'Scripts' and contains a sub-section 'Detalhes do Script'. The form has the following fields:

- Nome:** An empty text input field.
- Export ID (XID):** A text input field containing the value 'SC\_443122'.
- Pontos no contexto:** A dropdown menu showing 'Abertura do Disjuntor (11T1)' with a green plus icon to its right.
- Objetos no contexto:** A table with the following data:
 

Objeto	Var	Add
Comandos para Datasources	val_1	<input type="checkbox"/>
Comandos para Datapoints	val_2	<input type="checkbox"/>
- Script:** A large empty text area for entering the script code.

Below the 'Pontos no contexto' dropdown, there is a message: 'Nenhum ponto foi adicionado ao contexto do script'.

Fonte: ScadaBR.

Nessa tela, é possível configurar o nome do *script*, é gerado o identificador para o *script* *Export ID*, e é possível adicionar os pontos que estarão no contexto do *script* – essa opção é necessária apenas quando se deseja realizar a leitura da variável adicionada em algum ponto do *script*. Além disso, deve-se escolher se os comandos serão enviados para *data sources* ou para *data points*. Em seguida, há a caixa de texto, em que se elabora o código.

Os *scripts* elaborados para as mudanças de grupo de ajuste realizarão a seguinte tarefa: as variáveis respectivas aos dois ajustes que não são selecionados serão comandadas para 0 e o ajuste selecionado será comandado para 1. Nesse caso, como se deseja apenas realizar a escrita de dados, não serão adicionados pontos no contexto, e serão selecionados apenas comandos para *data points*. A única função em *javascript* utilizada será: “*void val\_2.writeDataPoint(String xid, String value)*”. No campo “*String XID*” coloca-se o *Export ID* do *data point*, e no campo “*String value*” coloca-se o valor de escrita desejado. Além disso, a denominação “*val\_2*” deve ser alterada para o nome atribuído na caixa de texto “*Var*”. Para execução desse *script* serão utilizados detectores e tratadores de eventos.

Os códigos dos *scripts* estão presentes no APÊNDICE A.

#### 4.3.5 Detectores e tratadores de eventos

##### 4.3.5.1 Detectores de eventos

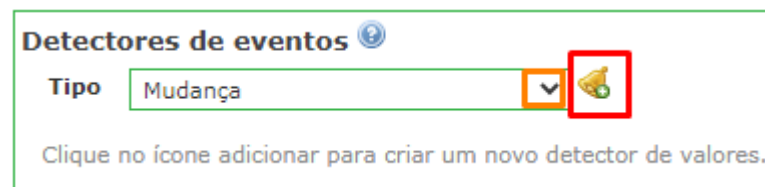
Os detectores de eventos têm a finalidade de constatar alterações, ou ausência de mudanças, nos valores dos *data points* com base nos seguintes critérios:

- **Mudança:** nesse caso, não há determinação do valor do *data point* para ocorrência do evento;
- **Estado:** nesse caso, há a determinação do valor do *data point* para ocorrência do evento e também do tempo de permanência no estado escolhido;
- **Contagem de mudança de estados:** nesse caso, determina-se a quantidade de mudanças no valor do *data point* para ocorrência do evento;
- **Sem mudanças:** nesse caso, determina-se a duração que a variável não sofre mudanças no seu valor;
- **Sem atualizações:** nesse caso, determina-se a duração que a variável não sofre atualizações, não precisando ocorrer mudança no valor.

Na detecção de um evento, pode-se gerar um alarme de nível: informação, urgente, crítico e *life safety*.

Para se adicionar um detector de evento para um *data point*, deve-se abrir a tela de propriedades do *data point* que se deseja adicionar a detecção de evento (Figura 40). Nessa tela, há um quadro “Detecção de Eventos”, conforme a Figura 51.

Figura 51 - Detectores de Eventos



Fonte: ScadaBR.

Na seta com destaque em laranja na Figura 51 é possível escolher o tipo de evento que se deseja adicionar, e no ícone com destaque em vermelho se adiciona o detector de evento. É possível adicionar mais de um detector de evento para o mesmo *data point*.

#### 4.3.5.2 Tratadores de eventos

Os tratadores de evento têm a finalidade de definir uma ação que deve ser tomada pelo ScadaBR na ocorrência de um evento, os quais podem ser do tipo:

- **E-mail:** é enviado e-mail de notificação, podendo ser especificado o endereço de envio;



- **Set point:** uma variável pode ser comandada com um valor em específico, podendo ser, inclusive, o *data point* que deu origem ao evento;
- **Script:** seleciona-se um *script* elaborado no ScadaBR para ser executado na ocorrência do evento.

Para criar um tratador de eventos, deve-se clicar no ícone da barra de ferramentas do ScadaBR destacado na Figura 52.

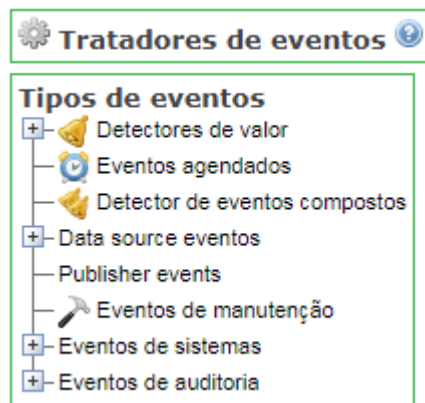
Figura 52 - Barra de ferramentas do ScadaBR (tratador de eventos)



Fonte: ScadaBR.

Em seguida, será exibida uma tela conforme a Figura 53.

Figura 53 - Tratadores de eventos



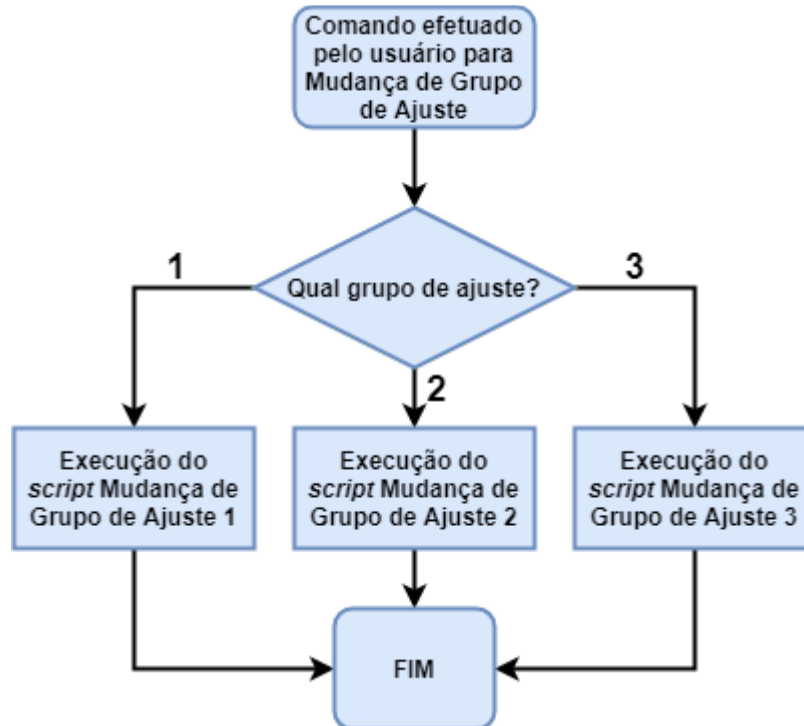
Fonte: ScadaBR.

Os detectores de eventos ficam disponíveis na opção “Detectores de valor”, sendo identificados pelo *data point* e pela definição do evento.

#### 4.3.5.3 Mudança de ajuste

Conforme mencionado anteriormente, foram elaborados *scripts* para os *data points* de mudança de ajuste. Para esses *scripts* serem executados, desenvolveu-se uma lógica conforme o fluxograma apresentado na Figura 54.

Figura 54 - Fluxograma para mudança de grupo de ajuste



Fonte: O próprio autor.

Para implementação desse fluxograma, inicialmente, será adicionado um detector de evento para cada *data point* de ajuste de cada relé. A Figura 55 apresenta essa definição para o *data point* Ajuste 1 (11T1).

Figura 55 - Definição do detector de evento para mudança de grupo de ajuste

Detectores de eventos	
Tipo	Estado
Tipo	Detector de estado
Export ID (XID)	PED_748964
Alias	
Nível de alarme	Nenhum alarme
Estado	Um
Duração	0 segundo(s)

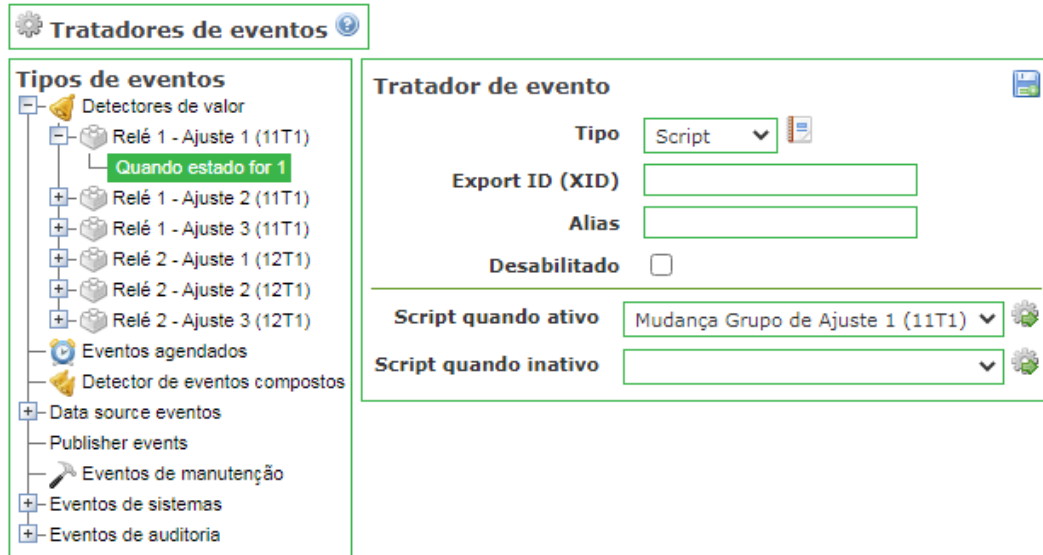
Fonte: ScadaBR.

Após a definição, é necessário clicar no botão “Salvar” localizado na parte inferior da tela de propriedades do *data point* (Figura 40).

Após efetuadas a definição dos detectores de eventos, deve-se efetuar a definição de um tratador de evento para cada *data point* de ajuste de cada relé. A Figura 56 apresenta essa

definição para o *data point* Ajuste 1 (11T1). Em seguida, deve-se salvar o tratador de evento clicando no ícone de disquete.

Figura 56 - Tratador de evento para mudança de ajuste

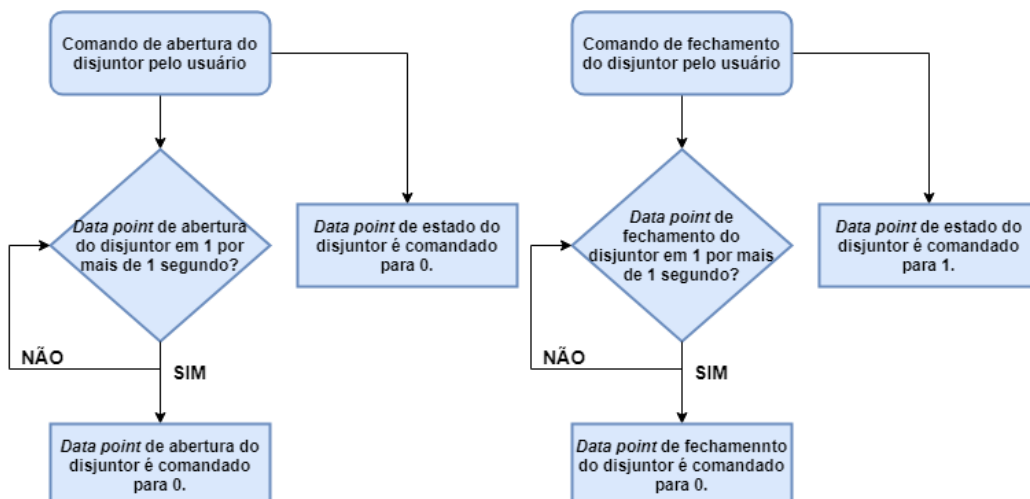


Fonte: ScadaBR.

#### 4.3.5.4 Abertura e fechamento do disjuntor

Para os *data points* de abertura e fechamento do disjuntor foram criados detectores e tratadores de eventos de forma que o comando efetuado tivesse a característica de um pulso de 1 segundo, e, também, de forma que o *data point* de estado do disjuntor fosse comandado para 0, caso o comando efetuado seja de abertura do disjuntor, ou para 1, caso o comando efetuado seja de fechamento do disjuntor. As definições dos detectores e tratadores de eventos seguiu a lógica apresentada na Figura 57.

Figura 57 - Fluxograma de abertura e fechamento do disjuntor



Fonte: ScadaBR.

Para implementação desse fluxograma, inicialmente, foram criados dois detectores de eventos para cada *data point* de abertura e de fechamento do disjuntor de cada *data source*. O primeiro detector de evento está associado à alteração do *data point* de estado do disjuntor, já o segundo está associado a determinar o comando com um pulso de um segundo. A Figura 58 apresenta as definições efetuadas para o *data point* Abertura do Disjuntor (11T1).

Figura 58 - Detectores de eventos associados ao *data point* Abertura do Disjuntor (11T1)

The screenshot shows a configuration window titled 'Detectores de eventos'. It contains two identical detector configurations. The first configuration has the following settings: Tipo: Estado; Export ID (XID): PED\_984160; Alias: (empty); Nível de alarme: Nenhum alarme; Estado: Um; Duração: 0 segundos. The second configuration has: Tipo: Detector de estado; Export ID (XID): PED\_981954; Alias: (empty); Nível de alarme: Nenhum alarme; Estado: Um; Duração: 1 segundos.

Fonte: ScadaBR.

Foram efetuadas as mesmas definições para os *data points* de abertura e fechamento de ambos os *data sources*. Vale ressaltar que, após a definição, é necessário clicar no botão “Salvar” localizado na parte inferior da tela de propriedades do *data point* (Figura 40).

Em seguida, foram definidos dois tratadores de eventos para cada *data point* de abertura e fechamento do disjuntor dos dois *data sources*, sendo um dos tratadores de eventos associados à mudança de estado do disjuntor, e o outro associado ao pulso do comando. Essas configurações são idênticas para os *data points* dos dois *data sources*.

A Figura 59 apresenta a configuração do tratador de evento associado à mudança de estado do disjuntor para o *data point* Abertura do Disjuntor (11T1). Nesse caso, o *data point* a ser alterado é o Estado do Disjuntor (11T1), sendo efetuado o comando desta variável para 0.

Figura 59 - Tratador de evento para abertura do disjuntor (estado do disjuntor)

The screenshot displays the ScadaBR configuration interface. On the left, a tree view under 'Tipos de eventos' shows 'Detectores de valor' expanded to 'Relé 1 - Abertura do Disjuntor (11T1)'. Under this, 'Quando estado for 1' is selected. The right panel, 'Tratador de evento', is configured as follows:

- Tipo:** Set point
- Export ID (XID):** (empty text box)
- Alias:** (empty text box)
- Desabilitado:**
- Target:** Relé 1 - Estado do Disjuntor (11T1)
- Ação ativa:** Setar para um valor estático
- Valor para setar:** 0
- Ação inativa:** Nenhuma

Fonte: ScadaBR.

A Figura 60 apresenta a configuração do tratador de evento associado ao pulso do comando para o *data point* Abertura do Disjuntor (11T1). Nesse caso, o *data point* a ser alterado é Abertura do Disjuntor (11T1), sendo efetuado o comando desta variável para 0.

Figura 60 - Tratador de evento para abertura do disjuntor (pulso do comando)

The screenshot displays the ScadaBR configuration interface. On the left, the tree view under 'Tipos de eventos' shows 'Detectores de valor' expanded to 'Relé 1 - Abertura do Disjuntor (11T1)'. Under this, 'Quando estado for 1' is selected, and 'Quando estado for 1 por 1 segundo(s)' is highlighted. The right panel, 'Tratador de evento', is configured as follows:

- Tipo:** Set point
- Export ID (XID):** (empty text box)
- Alias:** (empty text box)
- Desabilitado:**
- Target:** Relé 1 - Abertura do Disjuntor (11T1)
- Ação ativa:** Setar para um valor estático
- Valor para setar:** 0
- Ação inativa:** Nenhuma

Fonte: ScadaBR.

A Figura 61 apresenta a configuração do tratador de evento associado à mudança de estado do disjuntor para o *data point* Fechamento do Disjuntor (11T1). Nesse caso, o *data*

*point* a ser alterado é o Estado do Disjuntor (11T1), sendo efetuado o comando desta variável para 1.

Figura 61 - Tratador de evento para fechamento do disjuntor

The screenshot displays the 'Tipos de eventos' (Event Types) tree on the left and the 'Tratador de evento' (Event Processor) configuration panel on the right. In the tree, 'Relé 1 - Fechamento do Disjuntor (11T1)' is selected, with its sub-item 'Quando estado for 1' highlighted in green. The configuration panel shows the following settings:

- Tipo:** Set point
- Export ID (XID):** (empty text box)
- Alias:** (empty text box)
- Desabilitado:**
- Target:** Relé 1 - Estado do Disjuntor (11T1)
- Ação ativa:** Setar para um valor estático
- Valor para setar:** 1
- Ação inativa:** Nenhuma

Fonte: ScadaBR.

A Figura 62 apresenta a configuração do tratador de evento associado ao pulso do comando para o *data point* Fechamento do Disjuntor (11T1). Nesse caso, o *data point* a ser alterado é Fechamento do Disjuntor (11T1), sendo efetuado o comando desta variável para 0.

Figura 62 - Tratador de evento para fechamento do disjuntor

The screenshot displays the 'Tipos de eventos' (Event Types) tree on the left and the 'Tratador de evento' (Event Processor) configuration panel on the right. In the tree, 'Relé 1 - Fechamento do Disjuntor (11T1)' is selected, with its sub-item 'Quando estado for 1' and its sub-item 'Set point' highlighted in green. The configuration panel shows the following settings:

- Tipo:** Set point
- Export ID (XID):** (empty text box)
- Alias:** (empty text box)
- Desabilitado:**
- Target:** Relé 1 - Fechamento do Disjuntor (11T1)
- Ação ativa:** Setar para um valor estático
- Valor para setar:** 0
- Ação inativa:** Nenhuma

Fonte: ScadaBR.

#### 4.3.5.5 Alarme indicativo de abertura do disjuntor

Para os *data points* de estado do disjuntor de ambos os *data sources*, foram criados detectores de eventos para gerar um alarme de nível urgente, indicando a abertura do respectivo disjuntor. Nesse caso, deve-se detectar o estado de valor 0 desse *data point*. A Figura 63 mostra a definição do detector de eventos para o *data point* Estado do Disjuntor (11T1). Essa mesma definição foi efetuada para o *data point* Abertura do Disjuntor (12T1).

Figura 63 - Definição de alarme para abertura de disjuntor

The image shows a configuration window titled 'Detectores de eventos'. It contains several input fields and dropdown menus:

- Tipo:** Estado
- Tipo (ícone):** Detector de estado
- Export ID (XID):** PED\_354118
- Alias:** (empty field)
- Nível de alarme:** Urgente
- Estado:** Zero
- Duração:** 0 segundos

Fonte: ScadaBR.

Com isso, o ScadaBR gerará um alarme de nível urgente caso os estados dos disjuntores estejam em nível 0, ou seja, os disjuntores estarão abertos. A Figura 64 representa a lista de alarmes gerados em decorrência desses detectores de evento.

Figura 64 - Alarmes gerados pelo ScadaBR

Alarmes pendentes					Reconhecer todos ✓	Silenciar todos 📢
Id	Nível de alarme	Tempo	Mensagens	Tempo de inatividade		
39629	📢	22:53:58	Estado do Disjuntor (12T1) é 0 🗨️	Ativo 📢	✓	📢
39625	📢	22:53:52	Estado do Disjuntor (11T1) é 0 🗨️	Ativo 📢	✓	📢

Fonte: ScadaBR.

#### 4.3.6 Desenvolvimento das telas de supervisão

As telas de supervisão permitem uma melhor visualização dos dados a serem supervisionados ou comandados. No ScadaBR é possibilitada a criação de várias telas, sendo permitida a inclusão de uma imagem de fundo elaborada pelo desenvolvedor do supervisão.

No desenvolvimento do trabalho foram criadas três telas:

- **Unifilar:** nessa tela é possível visualizar o unifilar da subestação utilizado, enviar comandos de abertura e fechamento dos disjuntores, verificar o estado do disjuntor e efetuar mudança de grupo de ajuste;
- **Monitoramento de tensão:** nessa tela são visualizados dois gráficos referentes aos níveis de tensão simulados pelos dois *data sources*, além de ser mostrado o valor da tensão e especificado se o nível de tensão está adequado, precário ou crítico;
- **Alarmes:** nessa tela é possível visualizar a lista de alarmes gerados pelo ScadaBR.

Para se configurar as telas, deve-se clicar no ícone destacado em vermelho presente na barra de ferramentas do ScadaBR, conforme a Figura 65.

Figura 65 - Barra de ferramentas do ScadaBR (representações gráficas)



Fonte: ScadaBR.

Após isso, será aberta a tela de Representações Gráficas de acordo com a Figura 66.

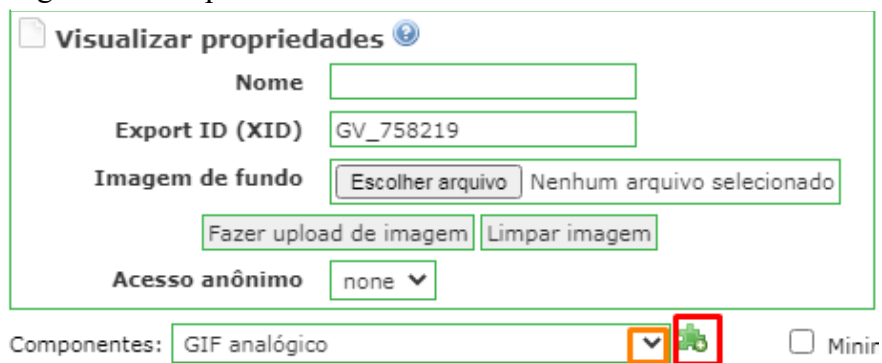
Figura 66 - Representações gráficas



Fonte: ScadaBR.

Clicando-se no item destacado em laranja na Figura 66, edita-se uma tela já criada. Clicando-se no item destacado em vermelho na Figura 66, acrescenta-se uma nova tela. Além disso, é possível visualizar a tela criada em tela cheia, clicando-se no botão "Full Screen". Ao acrescentar uma nova tela, são exibidas as opções de configurações conforme a Figura 67.

Figura 67 - Propriedades da tela



Fonte: ScadaBR.

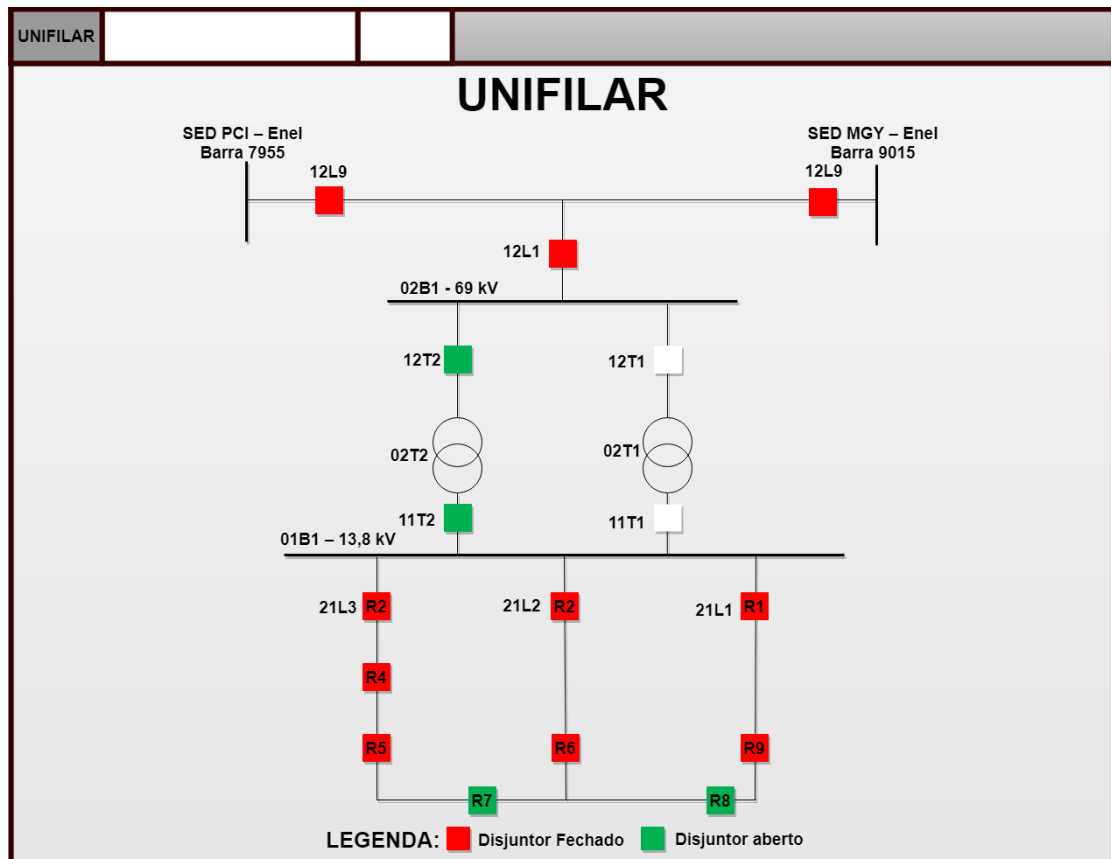


Nessas opções, é possível determinar um nome para a tela, escolher uma imagem de fundo e adicionar os componentes na tela – como botões, elementos HTML e gráficos. Para adicionar os componentes, seleciona-se o componente clicando na seta com destaque em laranja na Figura 67, e, em seguida, adiciona-se o componente selecionado clicando no ícone com destaque em vermelho na Figura 67. Cada tela criada possui um *link* diferente, o qual pode ser utilizado para criar os botões de mudança de tela.

#### 4.3.6.1 Tela de supervisão - unifilar

Para essa tela, foi elaborada a imagem de fundo apresentada na Figura 68.

Figura 68 - Imagem de fundo (unifilar)



Fonte: O próprio autor.

Para essa tela, adicionou-se os seguintes elementos:

- 2 botões para alternar entre a tela de monitoramento de tensão e a tela de alarmes;
- 2 listas de seleção, sendo uma para cada *data source*, para efetuar a mudança de ajustes;
- 2 ícones, sendo um para o estado do disjuntor de cada *data source*, os quais

indicam que o respectivo disjuntor (11T1 ou 12T1) estará aberto ou fechado;

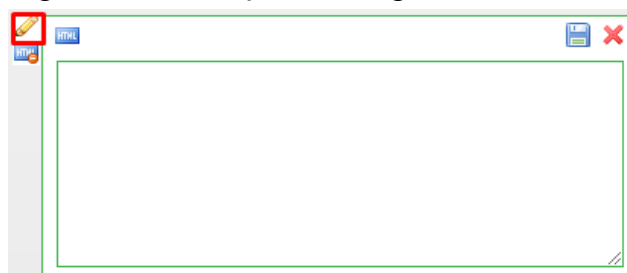
- 4 botões, sendo dois para abertura do disjuntor e dois para o fechamento do disjuntor respectivos a cada um dos *data sources*;

Para a criação do botão de mudança de tela, adicionou-se o componente “HTML” do ScadaBR, no qual se elaborou um código HTML para confecção do botão, que, quando clicado, efetua a mudança do endereço do *browser* para o endereço respectivo à tela escolhida.

Para a criação das listas de seleção, adicionou-se, também, o componente “HTML” do ScadaBR, no qual se elaborou um código HTML para confecção da lista, que, ao selecionar-se um dos três ajustes na lista, é executado o *script* para o respectivo ajuste selecionado.

Ao se adicionar o componente “HTML”, é exibido um menu de opções. Para se escrever o código, deve se clicar no botão com destaque em vermelho que será exibido uma caixa de texto para edição do código, conforme a Figura 69.

Figura 69 - Definição do código HTML



Fonte: ScadaBR.

Para a criação dos ícones de indicação do estado do disjuntor de cada *data source*, adicionou-se o componente “Script para servidor”, o qual se associa a um *data point* específico. Em seguida, elaborou-se um código em *javascript*, o qual varia o ícone conforme o valor da variável. As mesmas definições foram feitas para o *data point* Estado do Disjuntor (12T1).

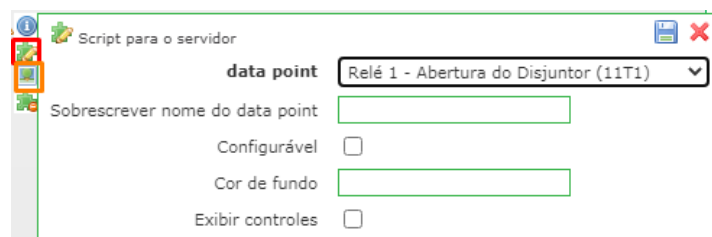
Para se adicionar os 4 botões de abertura e fechamento do disjuntor, adicionou-se o componente “Script para servidor” para cada um dos *data points*. No “Script para servidor” é definido um *data point* para se efetuar comandos e leitura, e é possível realizar somente a leitura de *data points* além do definido. Para esse componente, deve-se elaborar um código em *javascript* para criação alguns elementos – como: botão, *checkbox* e lista de seleção – e para a integração entre esses elementos e o *data point* definido.

Nesse caso, um código em *javascript* foi elaborado para a confecção de um botão, o qual, quando clicado, escreve o valor 1 no respectivo *data point* de abertura ou fechamento de disjuntor. Além disso, no código, o botão é desabilitado com base na leitura do *data point* de

estado do disjuntor – isto é, quando o disjuntor está aberto, o botão de abertura do disjuntor é desabilitado, e quando o disjuntor está fechado, o botão de fechamento do disjuntor é desabilitado. Outrossim, o botão também é desabilitado quando o *data point* de abertura ou de fechamento já está em 1.

Ao se adicionar o “*Script para servidor*”, é exibido um menu de opções. A Figura 70 mostra a definição do *data point* Abertura do Disjuntor (11T1), sendo efetuada clicando no ícone destacado em vermelho. O código é escrito clicando-se no ícone com destaque em laranja.

Figura 70 - Definição do *script* para servidor



Fonte: ScadaBR.

Os códigos mencionados estão disponíveis no APÊNDICE B.

#### 4.3.6.2 Tela de supervisão – monitoramento de tensão

Para essa tela, foi elaborada a imagem de fundo apresentada na Figura 71.

Figura 71 - Imagem de fundo (monitoramento de tensão)



Fonte: O próprio autor.

Para essa tela, foram adicionados os seguintes elementos:

- 2 gráficos de monitoramento em tempo real dos *data points* de nível de tensão de cada *data source*;
- 2 botões para alternar entre as telas “Unifilar” e “Alarmes”
- Para cada gráfico, foram adicionadas duas caixas de texto, sendo uma delas mostrando o valor da tensão e a outra exibindo se o nível de tensão está adequado, precário ou crítico.

Para se adicionar os gráficos, o componente “Gráfico” foi adicionado na tela. Nesse componente, deve-se selecionar os *data points* que se deseja monitorar – pode-se escolher até 10 *data points* -, definir o tamanho do gráfico e a duração de exibição dos dados. Nesse caso, foram criados dois gráficos para ser exibido cada nível de tensão individualmente.

A Figura 72 mostra a configuração do gráfico para o *data point* Nível de Tensão (01B1). A mesma definição foi efetuada para o *data point* Nível de Tensão (02B1).

Figura 72 - Definição do gráfico para o Nível de Tensão (01B1)

A imagem mostra uma janela de configuração de um gráfico. No topo, há ícones de ajuda, salvar e fechar. O formulário contém os seguintes campos:

- Nome:** Nível de Tensão (01B1)
- Largura:** 500
- Altura:** 400
- Duração:** 5 minutos (selecione "minuto(s)" no menu suspenso)
- data point 1:** Relé 1 - Nível de Tensão (01B1)

Fonte: ScadaBR.

Para a criação das caixas de texto, foram adicionados os componentes “Script de texto”, associando-os aos *data points* de nível de tensão de cada *data source*. Para a exibição do estado do nível da tensão, foi criado no “Script para servidor” um código *javascript* com 3 condições de intervalos. No caso, o Elipse *Modbus Simulator* gera valores entre 0 e 65535 (2 bytes), e, nesse contexto, atribuiu-se o seguinte: caso o valor esteja no intervalo [10000, 20000], será exibido que o nível de tensão está adequado; caso o valor esteja no intervalo [8000, 10000[ ou ]20000, 22000], será exibido que o nível de tensão está precário; e caso o valor esteja no intervalo [0, 8000[ ou ]22000, 65535] será exibido que o nível de tensão está crítico. Para exibição do valor da tensão, foi criado um outro “Script para servidor” com um código em *javascript* para exibição do valor do *data point*.

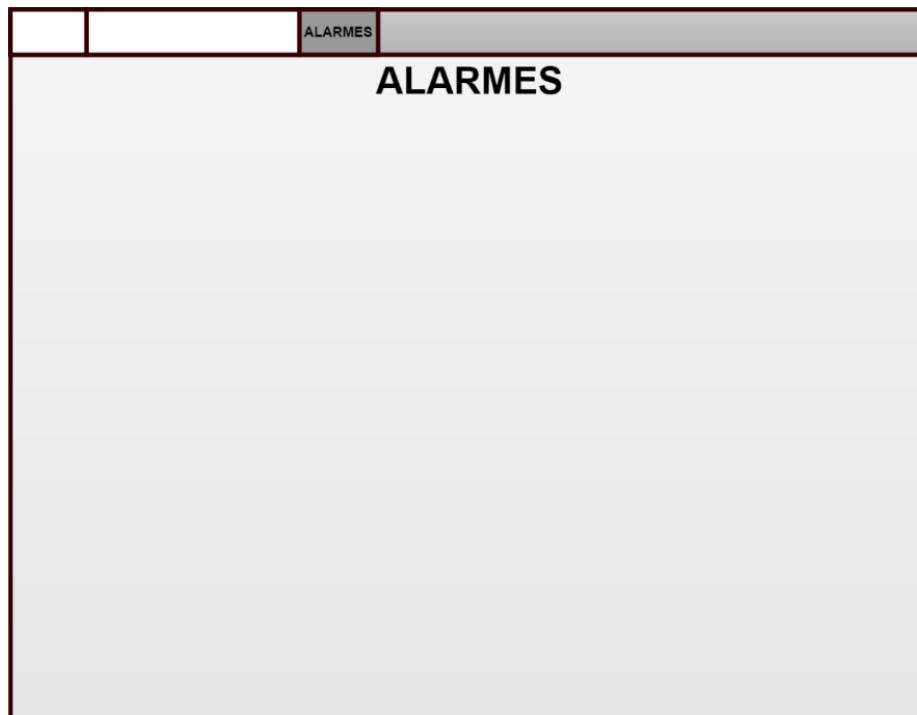
Para a geração de valores nos *data points* de nível de tensão, marcou-se a opção “*Random Values*” no *Eclipse Modbus Simulator*. Essa opção possibilita a geração de valores aleatórios pelo simulador e, conseqüentemente, uma melhor visualização dos gráficos para esses *data points*.

A criação de botões para mudança de tela se deu de forma análoga à criação na tela “Unifilar”. Os códigos mencionados estão presentes no APÊNDICE B.

#### 4.3.6.3 Tela de supervisão – alarmes

Para essa tela, foi elaborada a imagem de fundo apresentada na Figura 73.

Figura 73 - Imagem de fundo (alarmes)



Fonte: O próprio autor.

Para essa tela, foram adicionados os seguintes elementos:

- 1 lista contendo os 15 últimos alarmes gerados pelo ScadaBR;
- 2 botões para alternar entre as telas “Unifilar” e “Monitoramento de Tensão”

Para adição da lista de alarmes, adicionou-se o componente “Lista de Alarmes”, o qual foi configurado conforme a Figura 74.

Figura 74 - Definição da Lista de Alarmes

Fonte: ScadaBR.

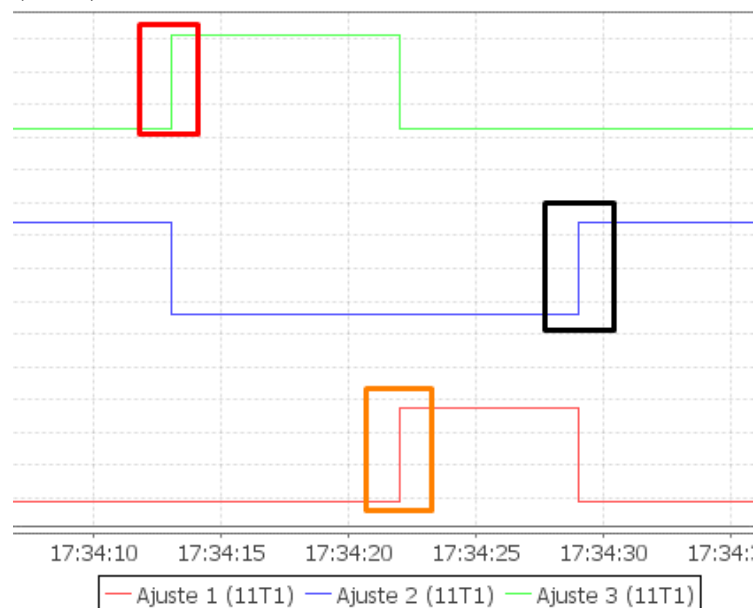
A criação de botões para mudança de tela se deu de forma análoga à criação na tela “Unifilar”. Os códigos estão presentes no APÊNDICE B.

#### 4.3.7 Resultados

##### 4.3.7.1 Lógica para mudança de grupo de ajuste

A Figura 75 apresenta os resultados obtidos para o *data source* Relé 1 acerca da lógica implementada para a mudança de grupo de ajuste. O gráfico foi obtido no ScadaBR.

Figura 75 - Resultados para Mudança de Grupo de Ajuste (11T1)



Fonte: O próprio autor.

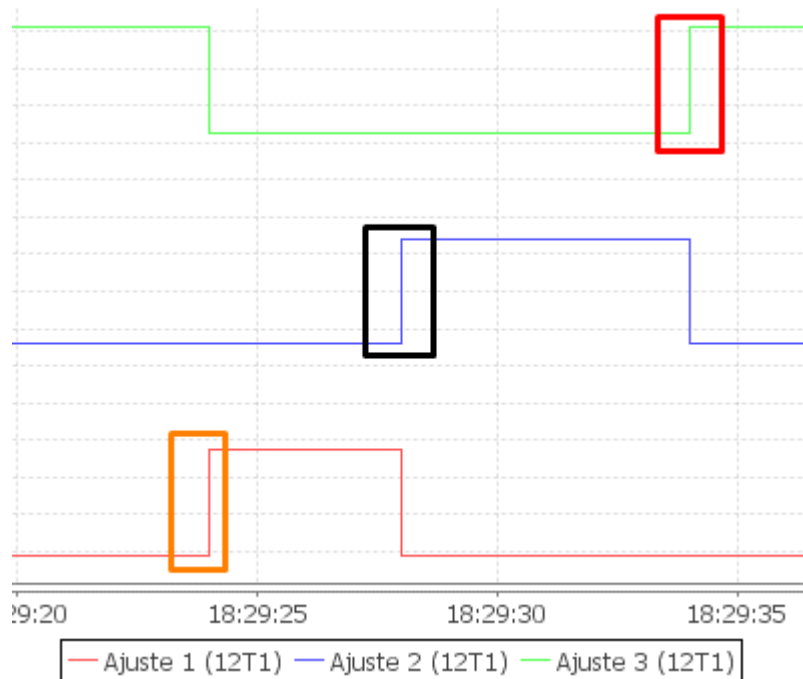
Os pontos em destaque na Figura 75 representam o envio de comandos pelo usuário, via *watchlist*, para mudança de grupo de ajuste, sendo o destaque vermelho a mudança para o

Grupo de Ajuste 3, o destaque laranja a mudança para o Grupo de Ajuste 1, e o destaque azul a mudança para o Grupo de Ajuste 2.

Como se observa na Figura 75, inicialmente, o relé estava configurado com o Ajuste 2, em seguida indo para o Ajuste 3, após isso, indo para o Ajuste 1 e retornando para o Ajuste 2, conforme os comandos enviados.

A Figura 76 apresenta os resultados obtidos para o *data source* Relé 2 acerca da lógica implementada para a mudança de grupo de ajuste. O gráfico foi obtido no ScadaBR.

Figura 76 - Resultados para mudança de grupo de ajuste (12T1)



Fonte: O próprio autor.

Os pontos em destaque representam o envio de comandos pelo usuário, via *watchlist*, para mudança de grupo de ajuste, sendo o destaque laranja a mudança para o Grupo de Ajuste 1, o destaque azul a mudança para o Grupo de Ajuste 2, e o destaque vermelho a mudança para o Grupo de Ajuste 3.

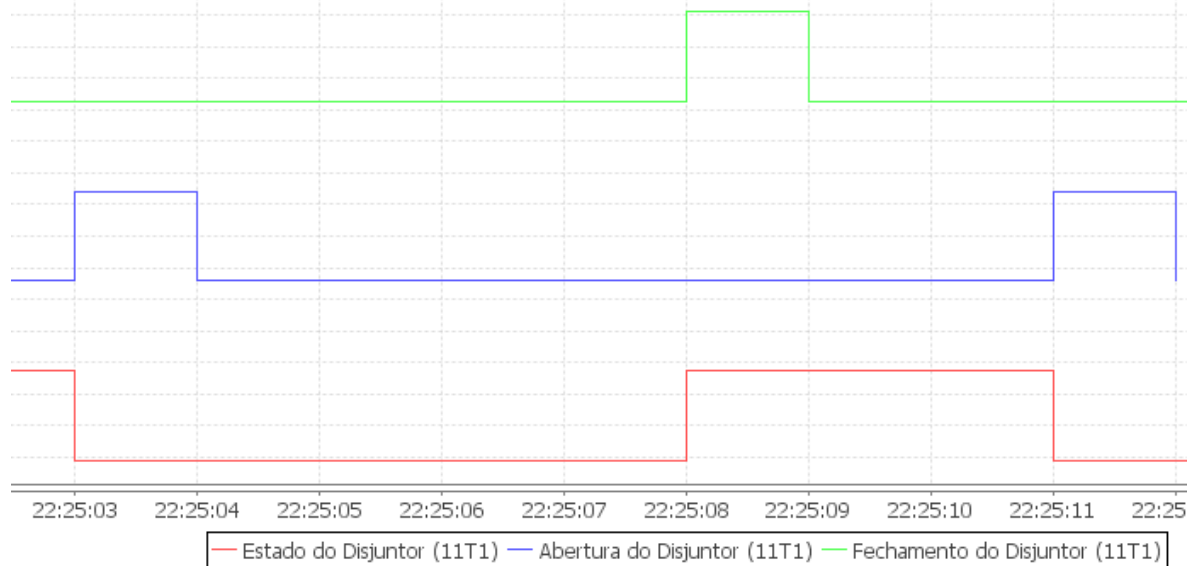
Como se observa na Figura 76, inicialmente o relé estava configurado com o Ajuste 3, em seguida indo para o Ajuste 1, após isso, indo para o Ajuste 2 e retornando para o Ajuste 3, conforme os comandos enviados.

Constatou-se que ao se comandar uma das variáveis de ajuste, as outras duas foram comandadas para 0 por meio da lógica elaborada utilizando-se os detectores e tratadores de eventos, e os *scripts*.

#### 4.3.7.2 Lógica para abertura e fechamento dos disjuntores

A Figura 77 apresenta os resultados obtidos para o *data source* Relé 1 acerca da lógica implementada de abertura e fechamento do disjuntor. O gráfico foi obtido no ScadaBR.

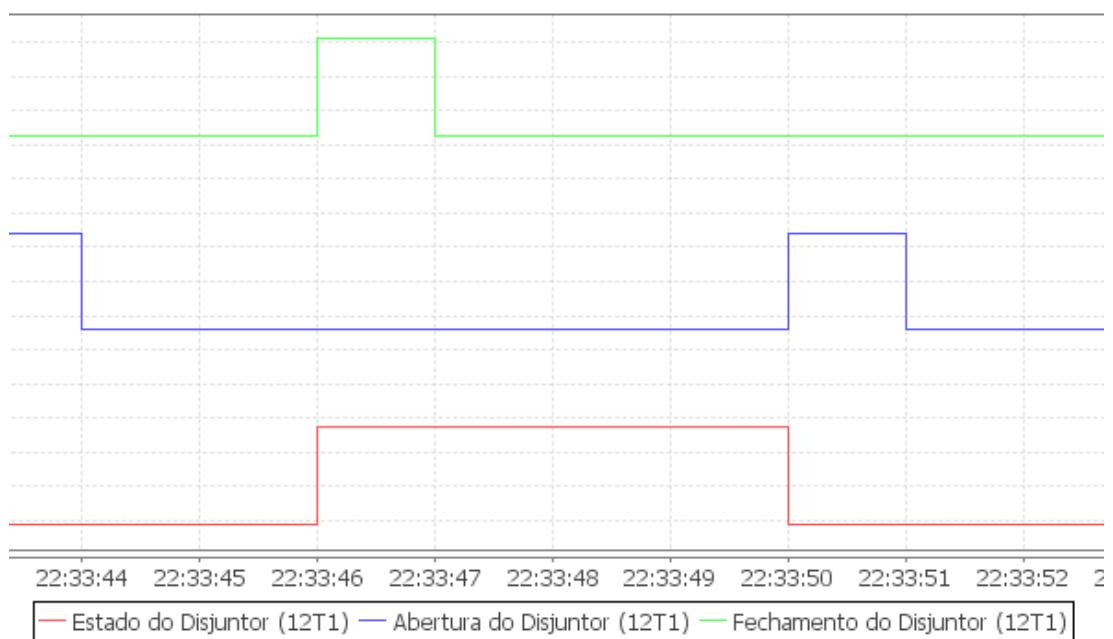
Figura 77 - Resultados de abertura e fechamento do disjuntor (Relé 1)



Fonte: ScadaBR.

A Figura 78 apresenta os resultados obtidos para o *data source* Relé 2 acerca da lógica implementada de abertura e fechamento do disjuntor. O gráfico foi obtido no ScadaBR.

Figura 78 - Resultados de abertura e fechamento do disjuntor (Relé 2)



Fonte: ScadaBR.

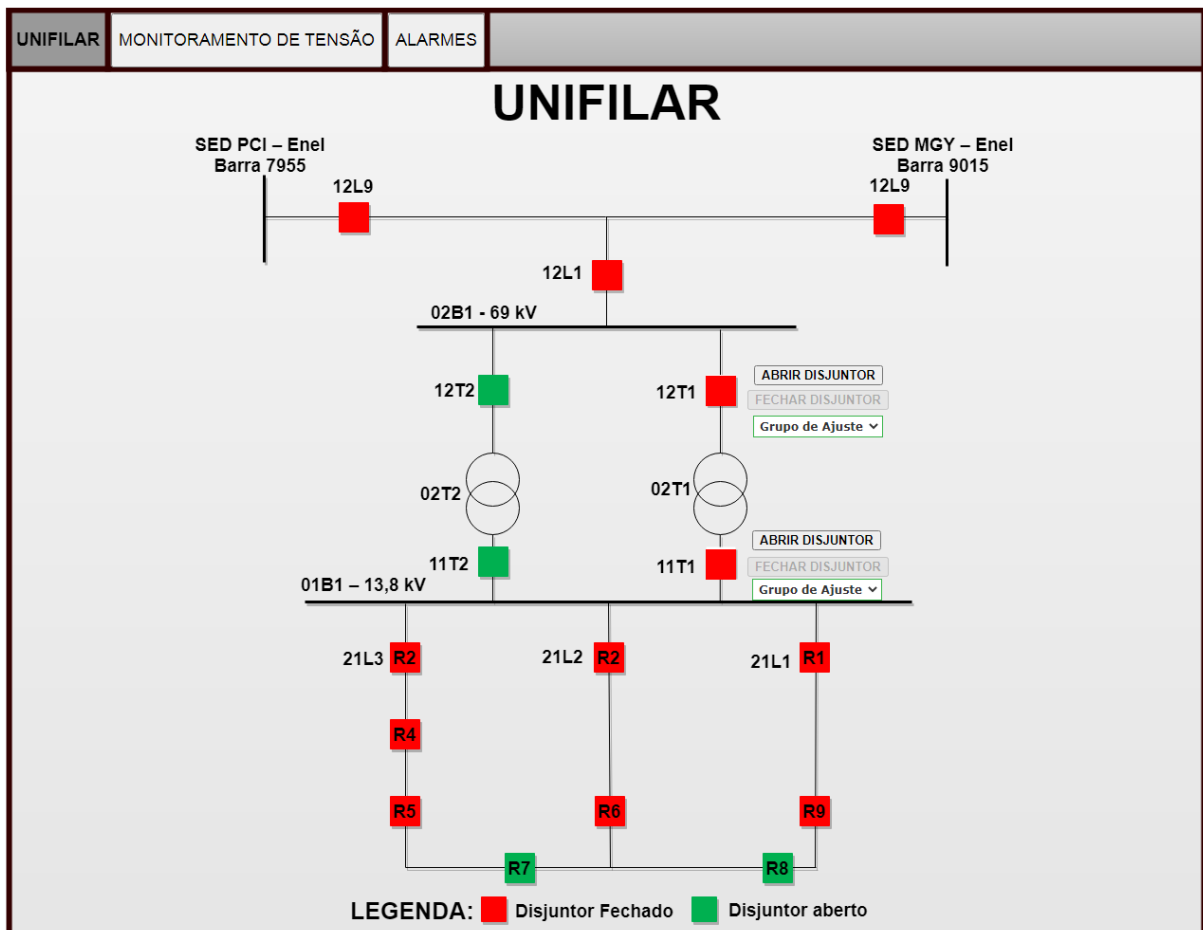


Conforme se observa, constata-se que os comandos de abertura e fechamento do disjuntor para os dois *data sources* se deram na forma de um pulso com duração de 1 segundo. Além disso, o *data point* de estado do disjuntor respondeu bem aos comandos de abertura e fechamento do disjuntor, simulando, assim, o comportamento de um disjuntor real.

#### 4.3.7.3 Tela de supervisão - unifilar

A Figura 79 apresenta o resultado da elaboração da tela “Unifilar”, supondo um funcionamento normal da Subestação *Campus* do Pici (11T1 e 12T1 fechados).

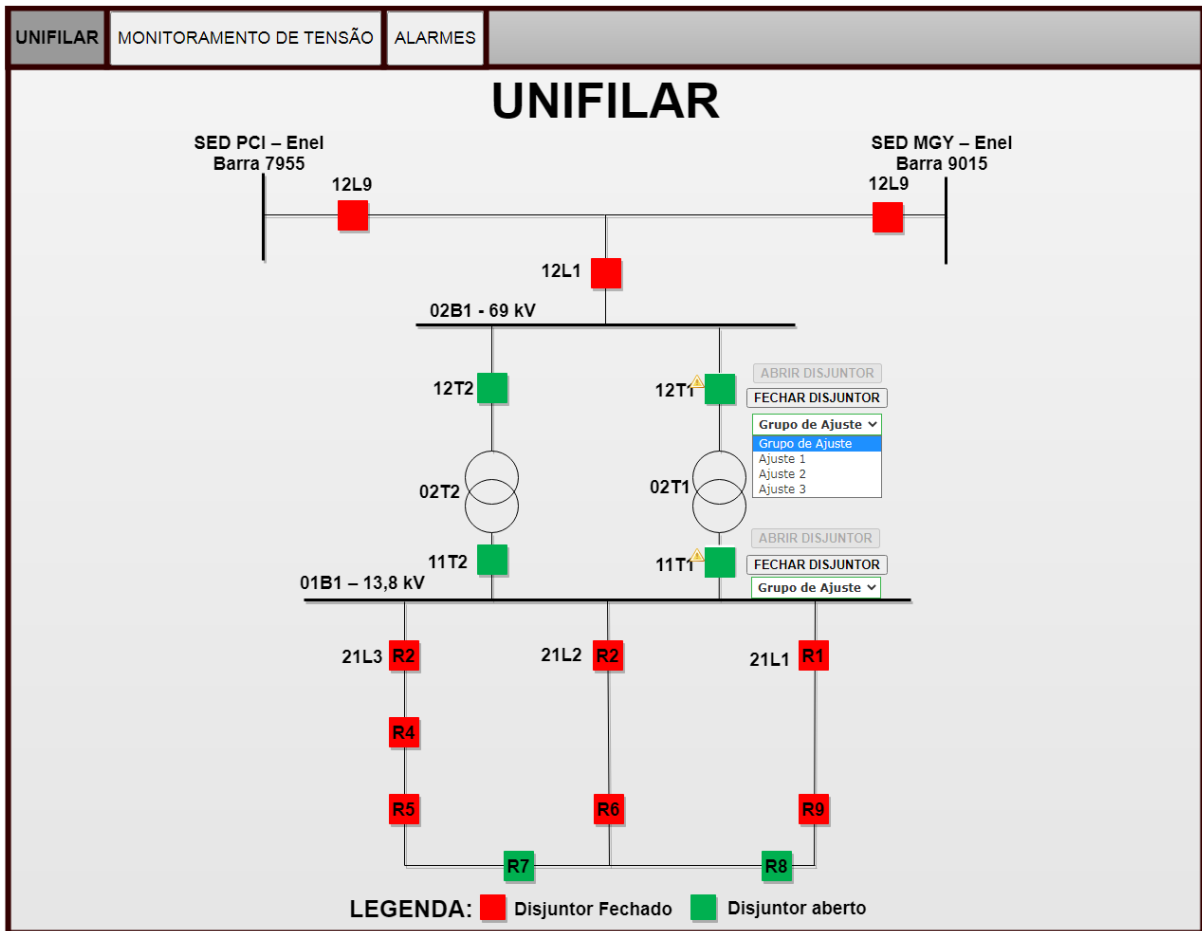
Figura 79 - Resultado da tela de supervisão (unifilar)



Fonte: O próprio autor.

Em seguida, enviou-se o comando para abertura dos disjuntores 11T1 e 12T1. Dessa forma, as telas se apresentaram conforme a Figura 80.

Figura 80 - Tela de supervisão (unifilar)



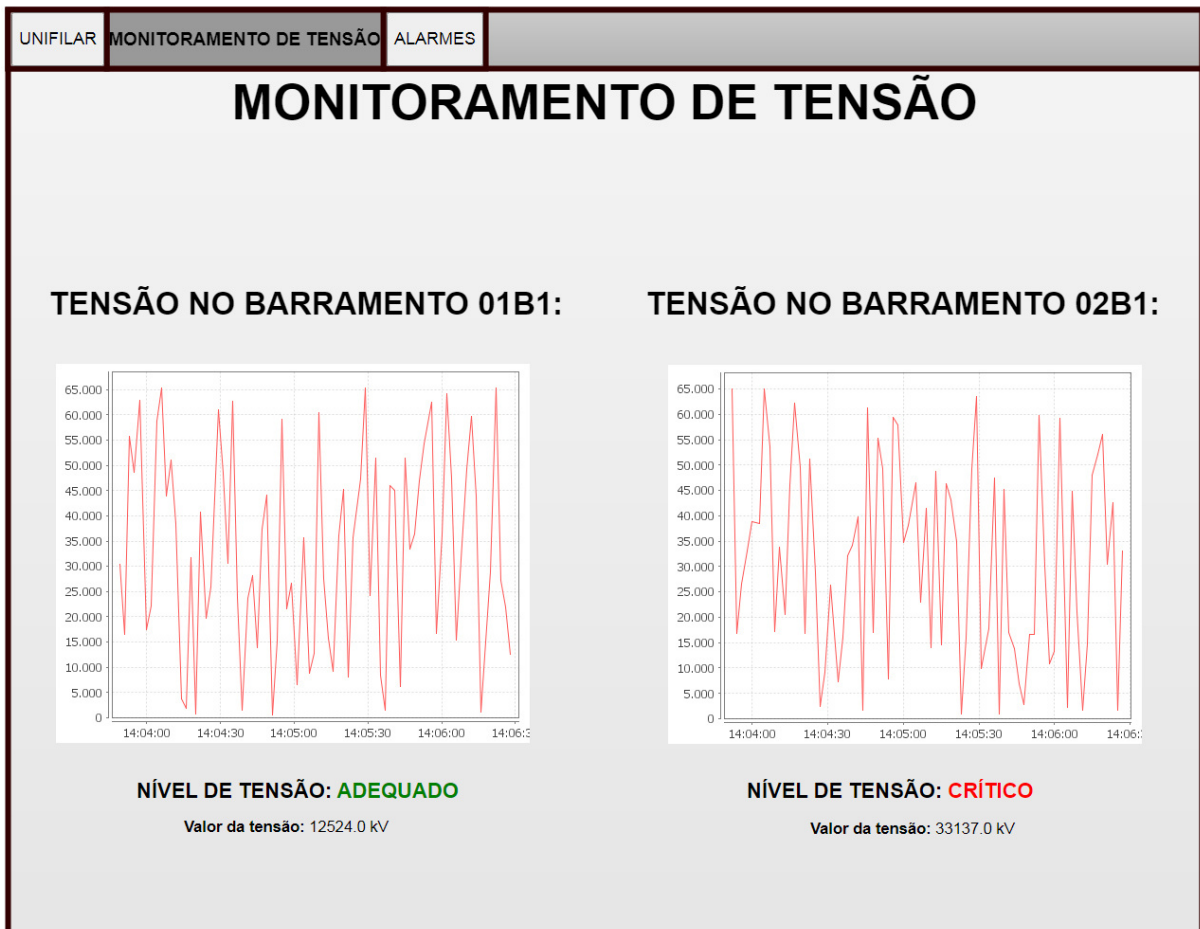
Fonte: O próprio autor.

Conforme se observa, os botões “ABRIR DISJUNTOR” foram desabilitados e os botões “FECHAR DISJUNTOR” foram habilitados. Verifica-se, também, que o estado da variável mudou, visto que o ícone de estado mudaram para a cor verde. Além disso, é mostrada a lista de seleção para mudança de grupo de ajuste. Dessa forma, o desenvolvimento dessa tela foi bem sucedido.

#### 4.3.7.4 Tela de supervisão - monitoramento de tensão

A Figura 81 apresenta o resultado da criação da tela de supervisão “Monitoramento de Tensão”.

Figura 81 - Resultado da tela de supervisão (monitoramento de tensão)



Fonte: O próprio autor.

Nessa tela, observa-se que a descrição do nível de tensão variou conforme o programado. No caso, o valor de 12524 foi classificado como “ADEQUADO” e o valor de 33137 foi classificado como “CRÍTICO”, conforme o esperado.

#### 4.3.7.5 Tela de supervisão - alarmes

A Figura 82 apresenta o resultado da criação da tela de supervisão “Monitoramento de Tensão”.

Figura 82 - Resultado da tela de supervisão (alarmes)

UNIFILAR	MONITORAMENTO DE TENSÃO	ALARMES	ALARMES		
			Nível de alarme	Tempo	Mensagens
			✘	14:06:41	Estado do Disjuntor (11T1) é 0
			✘	14:06:40	Estado do Disjuntor (12T1) é 0
			✘	14:06:38	Estado do Disjuntor (12T1) é 0
			✘	14:06:37	Estado do Disjuntor (11T1) é 0
			✘	14:06:35	Estado do Disjuntor (12T1) é 0
			✘	14:06:33	Estado do Disjuntor (12T1) é 0
			✘	14:06:32	Estado do Disjuntor (11T1) é 0
			✘	14:06:28	Estado do Disjuntor (11T1) é 0
			✘	14:06:26	Estado do Disjuntor (11T1) é 0
			✘	14:06:25	Estado do Disjuntor (12T1) é 0
			✘	14:06:23	Estado do Disjuntor (12T1) é 0
			✘	14:06:22	Estado do Disjuntor (11T1) é 0
			✘	14:06:19	Estado do Disjuntor (12T1) é 0
			✘	14:06:18	Estado do Disjuntor (11T1) é 0
			✘	14:06:15	Estado do Disjuntor (12T1) é 0

Fonte: O próprio autor.

Conforme se observa, nessa tela é possível visualizar os últimos 15 alarmes registrados pelo ScadaBR. Além disso, é possível constatar que o ScadaBR gerou alarmes quando as variáveis de estado do disjuntor, tanto para o disjuntor 11T1 quanto para o 12T1, estavam em estado 0.

#### 4.4 Considerações finais

Neste capítulo é apresentada a Metodologia de Aprendizagem Baseada em Projeto para aplicação no ensino do *software* ScadaBR na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência do curso de Engenharia Elétrica da Universidade Federal do Ceará (UFC).

O desenvolvimento do sistema SCADA em etapas conforme a metodologia proposta mostrou-se satisfatório, demonstrando um funcionamento adequado quanto ao envio

de comandos, leitura de dados, e demonstrando grande utilidade dos componentes fornecidos pelo ScadaBR para elaboração das telas.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de Metodologia de Aprendizagem Baseada em Projeto para construção de um sistema SCADA, e uma de aplicação dessa metodologia ABP com uso do *software* livre ScadaBR na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência do curso de Engenharia Elétrica da Universidade Federal do Ceará (UFC), com o intuito de contribuir positivamente para a formação dos alunos no que tange o ensino prático de Sistemas SCADA. Além disso, foi apresentado o desenvolvimento de um sistema supervísório seguindo-se a metodologia ABP proposta, com a utilização do ScadaBR, protocolo de comunicação *Modbus IP* e o *Elipse Modbus Simulator*.

No desenvolvimento do sistema supervísório, constatou-se que o ScadaBR é uma ferramenta que possui – como criação das telas de supervisão, monitoramento de dados, envio de comandos e criação de lógicas com os dados – para integração de dispositivos e equipamentos que simulam uma subestação. Além disso, o ScadaBR se mostrou uma ótima ferramenta didática, devido ao seu fácil uso e a ser uma plataforma livre e gratuita, podendo ser aplicado em disciplinas de automação.

O sistema supervísório demonstrou funcionamento conforme o esperado, e, pode ser replicado por alunos de curso de graduação de Engenharia Elétrica. Ademais, observa-se que para elaboração de um sistema supervísório, os alunos devem ter conhecimentos prévios da disciplina de Proteção de Sistemas Elétricos de Potência, de forma que seja possível os estudantes interpretar diagramas unifilares e compreenderem a operação dos relés de proteção em um sistema elétrico.

Portanto, o desenvolvimento de uma Metodologia de Aprendizagem Baseada em Projetos, com a utilização da plataforma livre ScadaBR, é extremamente eficiente como uma ferramenta de ensino prático na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência, pois pode estimular o engajamento e criatividade dos alunos. Ademais, essa metodologia também pode ser aplicada com a utilização de diferentes *softwares* supervísórios e diferentes dispositivos, permitindo, assim, que os alunos tenham contato com diversos programas SCADA.

### 5.1 Trabalhos futuros

Propõe-se como trabalhos futuros:

- Aplicação do trabalho a relés de sobrecorrente reais, utilizando dados de atuação

das funções de proteção no ScadaBR;

- Utilização de uma maleta de testes para se observar dados de grandeza analógica de forma a se obter uma onda de tensão ou de corrente no supervísório;
- Realizar uma avaliação da aplicação dessa metodologia na disciplina de Supervisão e Controle de Sistemas Elétricos de Potência, aplicando-se conceitos de *Business Intelligence*;
- Utilização de simuladores para outros protocolos de comunicação.

## REFERÊNCIAS

- ACKERMAN, W. J. **Substation automation and the EMS**. *In*: IEEE TRANSMISSION AND DISTRIBUTION CONFERENCE (CAT. NO. 99CH36333), 1999, New Orleans. New Orleans: IEEE, 1999. p. 274-279.
- ALMEIDA, E. M. **Norma IEC 61850 – Novo Padrão em Automação de Subestações**. 2011. 77 f. Monografia (Graduação em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2011.
- BARBOSA, E. F.; MOURA, D. G. **Metodologias Ativas de Ensino de Engenharia**. *In*: INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY EDUCATION, 13., 2014, Portugal. **Artigo...** Portugal: COPEC, 2014. p. 110-116.
- BICHELS, A. **Sistemas Elétricos de Potência: métodos de análise e solução**. Curitiba: EDUTFPR, 2018. 466 p.
- CARNEIRO, J. A. B. *et al.* Aplicação de Business Intelligence para Avaliação e Otimização da Disciplina de Proteção de Sistemas Elétricos. *In*: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 47., 2019, Fortaleza. **Artigo...**
- CURTIS, K. **A DNP3 Protocol Primer**. Disponível em: <<https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf>>. Acesso em: 12 fev. 2021.
- DE FARIA, M. H. M. *et al.* Estudo Comparativo Entre Ferramentas de Supervisão, Controle e Aquisição de Dados e a Importância Destas para o Ensino em Engenharia. *In*: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 40., 2012, Belém. **Artigo...**
- DISTRIBUTED NETWORK PROTOCOL. **Benefits Of DNP3**. Disponível em: <<https://www.dnp.org/About/Benefits-of-DNP3>>. Acesso em: 12 fev. 2021.
- DISTRIBUTED NETWORK PROTOCOL. **Features Of DNP3**. Disponível em: <<https://www.dnp.org/About/Features-of-DNP3>>. Acesso em: 12 fev. 2021.
- DISTRIBUTED NETWORK PROTOCOL. **Overview Of DNP3 Protocol**. Disponível em: <<https://www.dnp.org/About/Overview-of-DNP3-Protocol>>. Acesso em: 12 fev. 2021.
- JARDINI, J. A. **Sistemas Elétricos De Potência: Automação**. São Paulo: EPUSP, 1999.
- JOHNSON, D. **Is DNP 3.0 the Right Standard For You?** Disponível em: <<https://www.dnp.org/Portals/0/AboutUs/2000-06-UA-DNP.pdf>>. Acesso em: 12 fev. 2021.
- LEÃO, R. P. S. **GTD – Geração, Transmissão e Distribuição de Energia Elétrica**. Fortaleza, 2015. 47 p.
- LIBERATO, J. P. D. **Estudo e aplicação de um supervisório integrado a um banco de dados para a rede de distribuição do Campus do Pici**. 2014. 99 f. Monografia (Graduação em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2014.



LIMA, M. A. R. **Estudo dos Impactos da Inserção de Geração Distribuída no Sistema de Proteção da Rede Elétrica do Campus do Pici Utilizando o Software ATPDRAW**. 2019. 127 f. Monografia (Graduação em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2019.

MAMEDE FILHO, J.; MAMEDE, D. R. **Proteção de Sistemas Elétricos de Potência**. Rio de Janeiro: LTC, 2013.

MODBUS. **MODBUS Application Protocol Specification V1.1b3**. Disponível em: <[https://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)>. Acesso em: 12 fev. 2021.

MODBUS. **Modbus FAQ: About the Protocol**. Disponível em: <<https://modbus.org/faq.php>>. Acesso em: 02 mar. 2021.

PEXTRON. **Manual de Operação URPE 7104**, 2018.

SCADABR. **Principais Funcionalidades**. Disponível em: <<https://www.scadabr.com.br/>>. Acesso em: 03 mar. 2021.

SCADABR. **Aplicações do ScadaBR**. Disponível em: <<https://www.scadabr.com.br/>>. Acesso em: 03 mar. 2021.

SCADABR. **Manual do Software**, 2010.

SILVA, M. S. **Modelagem das Funções de uma Subestação Automatizada Empregando Modelos Orientados a Objeto**. 2002. 169 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2002.

SANTOS-MARTIN, D. *et al.* Problem-Based Learning in Wind Energy Using Virtual and Real Setups. **IEEE Transactions On Education**, v. 55, n. 1, p. 126-134, fev. 2012.

SOUZA, E. H. N. **Automação de Subestações com Protocolo IEC-61850: estudo de caso**. 2016. 50 f. Monografia (Especialização em Automação Industrial) – Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná, Curitiba, 2016.

VALENTE, V. A.; ALMEIDA, M. E. B.; GERALDINI, A. F. S. Metodologias ativas: das concepções às práticas em distintos níveis de ensino. **Rev. Diálogo Educ.**, Curitiba, v. 17, n. 52, p. 455-478, abr./jun. 2017.

VIEIRA, C. R. **Análise dos Impactos da Geração Solar Fotovoltaica no Sistema Elétrico de Distribuição de um Campus Universitário**. 2016. 148 f. Monografia (Graduação em Engenharia Elétrica) – Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2016.

VIEIRA, K. **A Utilização do PBL nos Cursos de Engenharia do Brasil: Uma Análise Bibliométrica**. 2015. 29 f. Trabalho de Conclusão de Curso apresentado como requisito parcial para titulação no Curso de Pós-Graduação Lato Sensu em Ciências e Tecnologia – Centro Tecnológico de Joinville, Universidade Federal de Santa Catarina, Joinville, 2017.

## APÊNDICE A – *SCRIPTS* PARA MUDANÇA DE AJUSTE

### 1 MUDANÇA DE GRUPO DE AJUSTE 1 (11T1)

```
void dp.writeDataPoint('DP_132017', 0);  
void dp.writeDataPoint('DP_588958', 0);  
void dp.writeDataPoint('DP_346567', 1);
```

### 2 MUDANÇA DE GRUPO DE AJUSTE 2 (11T1)

```
void dp.writeDataPoint('DP_346567', 0);  
void dp.writeDataPoint('DP_588958', 0);  
void dp.writeDataPoint('DP_132017', 1);
```

### 3 MUDANÇA DE GRUPO DE AJUSTE 3 (11T1)

```
void dp.writeDataPoint('DP_346567', 0);  
void dp.writeDataPoint('DP_132017', 0);  
void dp.writeDataPoint('DP_588958', 1);
```

### 4 MUDANÇA DE GRUPO DE AJUSTE 1 (12T1)

```
void dp.writeDataPoint('DP_463074', 0);  
void dp.writeDataPoint('DP_993657', 0);  
void dp.writeDataPoint('DP_679190', 1);
```

### 5 MUDANÇA DE GRUPO DE AJUSTE 1 (12T1)

```
void dp.writeDataPoint('DP_679190', 0);  
void dp.writeDataPoint('DP_993657', 0);  
void dp.writeDataPoint('DP_463074', 1);
```

### 6 MUDANÇA DE GRUPO DE AJUSTE 1 (12T1)

```
void dp.writeDataPoint('DP_679190', 0);  
void dp.writeDataPoint('DP_463074', 0);  
void dp.writeDataPoint('DP_993657', 1);
```

## APÊNDICE B – CÓDIGOS DOS COMPONENTES “HTML” E “SCRIPT PARA SERVIDOR”

### 1 COMPONENTE “HTML”

#### 1.1 Botões de Mudança de Tela – Unifilar

```
<button  
onClick="javascript:window.location.href=  
'http://localhost:8080/ScadaBR/views.shtm?viewId=3' ">  
<br>  
<font size=+1 face=Arial  
color=black>UNIFILAR</font>  
<br>  
<br>  
</button>
```

#### 1.2 Botões de Mudança de Tela – Monitoramento de Tensão

```
<button  
onClick="javascript:window.location.href=  
'http://localhost:8080/ScadaBR/views.shtm?viewId=5' ">  
<br>  
<font size=+1 face=Arial  
color=black>ALARMES</font>  
<br>  
<br>  
</button>
```

#### 1.3 Botões de Mudança de Tela – Alarmes

```
<button  
onClick="javascript:window.location.href=  
'http://localhost:8080/ScadaBR/views.shtm?viewId=5' ">  
<br>  
<font size=+1 face=Arial  
color=black>ALARMES</font>  
<br>  
<br>  
</button>
```

## 2 COMPONENTE “SCRIPT PARA SERVIDOR”

### 2.1 Ícone de Estado do Disjuntor (11T1 e 12T1)

Foi utilizado o mesmo código para os dois *scripts* dos dois disjuntores. Apenas na definição do *script* para servidor deve-se selecionar o *data point* adequado.

```
var s="";

if(value)
s+="

```

### 2.2 Botão de Abertura do Disjuntor (11T1 e 12T1)

Foi utilizado o mesmo código para os dois *scripts* dos botões respectivos aos dois disjuntores. Apenas na definição do *script* para servidor deve-se selecionar o respectivo *data point*.

```
var s = "";
var dpDao = new com.serotonin.mango.db.dao.PointValueDao();
var dpVo = dpDao.getLatestPointValue('8');

if (dpVo.value=="true")
{
if (value)
s+="

```

### 2.3 Botão de Fechamento do Disjuntor (11T1 e 12T1)

Foi utilizado o mesmo código para os dois *scripts* dos botões respectivos aos dois disjuntores. Apenas na definição do *script* para servidor deve-se selecionar o respectivo *data point*.

```
var s = "";
var dpDao = new com.serotonin.mango.db.dao.PointValueDao();
var dpVo = dpDao.getLatestPointValue('8');

if (dpVo.value=="false")
{
if (value)
    s+="

```

### 2.4 Caixa de Texto para Nível de Tensão (11T1 e 12T2)

Foi utilizado o mesmo código para os dois *scripts* das caixas de texto respectivas aos dois relés. Apenas na definição do *script* para servidor deve-se selecionar o respectivo *data point*.

```
var texto = "";

if(value<=20000&& value>=10000)
{
texto="<h1 style='font-family:arial;'>NÍVEL DE TENSÃO: <span
style='color:green;'>ADEQUADO</span></h1>";
}
else if ((value<10000 && value>=8000)|| (value<=22000 && value>20000))
{
texto="<h1 style='font-family:arial;'>NÍVEL DE TENSÃO: <span
style='color:#FF8000;'>PRECÁRIO</span></h1>";
}
else
{
```

```
texto="<h1 style='font-family:arial;'>NÍVEL DE TENSÃO: <span  
style='color:red;'>CRÍTICO</span></h1>";  
}  
  
return texto;
```

## 2.5 Caixa de Texto para Valor de Tensão (11T1 e 12T2)

Foi utilizado o mesmo código para os dois *scripts* das caixas de texto respectivas aos dois relés. Apenas na definição do *script* para servidor deve-se seleccionar o respectivo *data point*.

```
var texto = "<h2 style='font-family:arial;'>Valor da tensão: <span style='font-  
weight:normal;'>"+ renderedText + " kV</span></h2>";  
  
return texto
```