



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RHÚLIO VICTOR LUZ CARVALHO SOUSA

**DESENVOLVIMENTO DE UM SISTEMA DE SUPERVISÃO E AQUISIÇÃO DE
DADOS PARA MÚLTIPLOS PROJETOS COM VISUALIZAÇÃO WEB**

TERESINA

2019

RHÚLIO VICTOR LUZ CARVALHO SOUSA

DESENVOLVIMENTO DE UM SISTEMA DE SUPERVISÃO E AQUISIÇÃO DE DADOS
PARA MÚLTIPLOS PROJETOS COM VISUALIZAÇÃO WEB

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Piauí, como requisito parcial à obtenção do
grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. José Maria Pires
de Menezes Júnior

Co-Orientador: Prof. Dr. Otacílio da
Mota Almeida

TERESINA

2019

RHÚLIO VICTOR LUZ CARVALHO SOUSA

DESENVOLVIMENTO DE UM SISTEMA DE SUPERVISÃO E AQUISIÇÃO DE DADOS
PARA MÚLTIPLOS PROJETOS COM VISUALIZAÇÃO WEB

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia Elétrica do
Centro de Tecnologia da Universidade Federal
do Piauí, como requisito parcial à obtenção do
grau de bacharel em Engenharia Elétrica.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. José Maria Pires de Menezes
Júnior (Orientador)
Universidade Federal do Piauí (UFPI)

Prof. Dr. José Medeiros De Araújo Júnior
Universidade Federal do Piauí (UFPI)

Prof. PhD. Antônio Airton Carneiro de Freitas
Universidade Federal do Piauí (UFPI)

Este trabalho é dedicado ao homem, que apesar de não ter sido engenheiro, possui genialidade, experiência e talento superiores ao que fosse, e ainda assim, trilhou por vários caminhos que tornaram possível a formação de um e a concretização de um sonho. Meu pai.

AGRADECIMENTOS

Ao meu pai Joaquim Francisco de Sousa pelos ensinamentos e exemplos de que, respeito e benfeitorias são proporcionais ao caráter, honestidade, responsabilidade e compromisso de um homem, bases que tornam possível o seu crescimento.

À minha mãe Maria Josenildes Luz Carvalho pelo dom da vida e a capacidade de ser justo em detrimento de benefícios pessoais, de poder replicar, mesmo que infinitesimalmente, a humildade e compaixão com os meus semelhantes.

Ao orientador José Maria Pires de Menezes Júnior por toda a evolução que obtive ao longo do curso devido seu apoio em todas as etapas de meus trabalhos, propiciando que eu obtivesse sempre os melhores resultados possíveis.

Às minhas irmãs e minha família, em especial aos meus falecidos avós maternos, por serem o todo da parte que sou, demonstrando que mesmo com a distância, os laços familiares são prioridade e o fator mais importante da vida em que estamos.

À minha namorada Jéssica, pelo amor, companheirismo e apoio nas horas que mais precisei, por sua presença me lembrar a todo momento de que as dificuldades não passam de autoflagelações e abstrações da mente e que o mundo é pequeno quando se tem a coragem necessária para vivê-lo por completo.

Aos amigos de infância e de todo o ensino escolar,
por mostrarem que família independe de sangue e que da infância nascem amizades.

A todos meus amigos que tive a honra de conhecer na graduação, que me fizeram seguir em frente e acreditar que mesmo em situações ruins, existem caminhos e possibilidades para a resolução de todas as adversidades por pior que sejam e que, por maior que seja o objetivo, ele pode ser alcançado.

Aos professores do Departamento de Engenharia Elétrica da Universidade Federal do Piauí que somaram de forma positiva com a minha formação e tornaram possível a compreensão da imensidão da engenharia, e também aqueles, que através exemplos, possibilitaram a moldura do profissional que não devo ser.

"Faça as coisas o mais simples que você puder,
porém não se restrinja às mais simples."

(Albert Einstein)

RESUMO

O resumo do trabalho todinho vai aqui.

Palavras-chave: Palavras. Chave. Aqui.

ABSTRACT

Tradução do resumo aqui.

Keywords: Key. Words. Here.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de IHM da fabricante Branqs.	18
Figura 2 – Exemplo de CLP de modelo PLC300 da fabricante WEG.	19
Figura 3 – Exemplo de UTR de modelo RUW-03 da fabricante WEG.	20
Figura 4 – Pinagem do conector RS485 utilizado no protocolo Modbus Serial.	21
Figura 5 – Pinagem do conector RJ45 utilizado no protocolo Modbus Ethernet.	22
Figura 6 – Diagrama do funcionamento de um sistema que utilize OPC Classic.	24
Figura 7 – Diagrama do funcionamento de um sistema que utilize OPC-UA.	25
Figura 8 – Representação do funcionamento de uma API REST.	27
Figura 9 – Exemplo de informações organizadas no formato XML.	28
Figura 10 – Exemplo de informações organizadas no formato JSON.	29
Figura 11 – Representação do funcionamento do MQTT.	30
Figura 12 – Arquitetura física básica do SCADA.	32
Figura 13 – Família de Dispositivos com Comunicação Integrada da fabricante WAGO.	33
Figura 14 – Arquitetura de um SCADA WEB.	34
Figura 15 – Demonstração da tela de processo do <i>software</i> Elipse E3.	36
Figura 16 – Demonstração da tela de processo do <i>software</i> InduSoft Web Studio®.	37
Figura 17 – Demonstração de telas, incluindo a de processo, do <i>software</i> ScadaBR.	38
Figura 18 – Demonstração da tela de processo do <i>software</i> TANGO Controls.	39
Figura 19 – Demonstração da tela de processo do <i>software</i> Rapid SCADA.	40
Figura 20 – Diagrama das etapas do servidor para manipulação de dados.	42
Figura 21 – Diagrama de validação das informações recebidas.	43
Figura 22 – Diagrama de autenticação do usuário.	43
Figura 23 – Diagrama sobre a identificação do tipo das informações.	43
Figura 24 – Banco de Dados.	44
Figura 25 – Tela de autenticação da Interface Web.	47
Figura 26 – Tela de cadastro para novos usuários.	48
Figura 27 – Tela inicial do sistema após autenticação.	48
Figura 28 – Apresentação Geral de todos os projetos cadastrados.	49
Figura 29 – Página de cadastro de novos Projetos.	49
Figura 30 – Página de cadastro de novas Variáveis.	49
Figura 31 – Página de cadastro de novos Objetos.	50

Figura 32 – Objeto recém-criado.	50
Figura 33 – Associação de novo cliente ao projeto.	50
Figura 34 – Visão geral dos clientes cadastrados no projeto.	51
Figura 35 – Variáveis utilizadas para o projeto.	52
Figura 36 – Últimas informações enviadas.	52
Figura 37 – Variáveis utilizadas no projeto.	53
Figura 38 – Botão para ação no processo além de últimas informações enviadas.	53
Figura 39 – Quantidade de hóspedes conectados no roteador da recepção durante 24 horas.	53
Figura 40 – Uso máximo de downlink pelo roteador da recepção durante 24 horas.	54
Figura 41 – Quantidade de hóspedes conectados no roteador "107" durante 24 horas.	54
Figura 42 – Uso máximo de downlink pelo roteador "107" durante 24 horas.	54

LISTA DE TABELAS

Tabela 2 – Representação do pacote no modo RTU.	21
Tabela 3 – Representação do pacote no modo ASCII.	21
Tabela 4 – Pinagem do conector RS485 utilizado no protocolo Modbus Serial.	22
Tabela 5 – Pinagem do conector RJ45 utilizado no protocolo Modbus Ethernet.	23
Tabela 6 – Comparativo entre os métodos de chamada.	28

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ASCII	American Standard Code for Information Interchange
CLP	Controlador Lógico Programável
COM/DCOM	<i>Distributed Component Object Model</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
IHM	Interface Humano-Máquina
JSON	<i>JavaScript Object Notation</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MQTT	<i>Message Queue Telemetry Transport</i>
OASIS	<i>Organization for the Advancement of Structured Information Standards</i>
OLE	<i>Object Linking and Embedding</i>
OPC	<i>Open Platform Communications</i>
OPC-UA	<i>Object Linking and Embedding for Process Control - Unified Architecture</i>
REST	<i>Representational State Transfer</i>
RTU	Remote Terminal Unit
SCADA	<i>Supervisory Control and Data Acquisition</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
TLS	<i>Transport Layer Security</i>
UAD	Unidade de Aquisição de Dados
UADC	Unidade de Aquisição de Dados e Controle
UD	Unidade Dedicada
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
UTR	Unidade Terminal Remota
W3C	<i>World Wide Web Consortium</i>
WEB	<i>World Wide Web</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	15
2	DISPOSITIVOS E PROTOCOLOS	17
2.1	Automação Industrial	17
2.1.1	<i>Interface Humano-Máquina</i>	17
2.1.2	<i>Unidade de Aquisição de Dados</i>	17
2.1.2.1	<i>Unidade Dedicada</i>	18
2.1.2.2	<i>Unidade de Aquisição de Dados e Controle</i>	18
2.1.2.2.1	<i>Controlador Lógico Programável</i>	18
2.1.2.2.2	<i>Unidade Terminal Remota</i>	19
2.2	Protocolos de Comunicação	20
2.2.1	<i>Modbus</i>	20
2.2.1.1	<i>Modbus Serial</i>	20
2.2.1.2	<i>Modbus TCP/IP</i>	21
2.2.2	<i>OPC</i>	22
2.2.2.1	<i>OPC Classic</i>	23
2.2.2.2	<i>OPC-UA</i>	24
2.2.3	<i>HTTP</i>	25
2.2.3.1	<i>HTTPS</i>	26
2.2.3.2	<i>REST</i>	26
2.2.3.3	<i>Métodos de Chamada</i>	27
2.2.3.4	<i>Formatos de Conteúdo</i>	27
2.2.4	<i>MQTT</i>	29
3	SISTEMAS SCADA	31
3.1	SCADA WEB	31
3.2	Sistemas SCADA disponíveis no mercado	35
3.2.1	<i>Sistemas proprietários</i>	35
3.2.1.1	<i>Elipse E3</i>	35
3.2.1.2	<i>InduSoft Web Studio®</i>	36
3.2.2	<i>Sistemas de código aberto</i>	37
3.2.2.1	<i>ScadaBR</i>	37

3.2.2.2	<i>TANGO Controls</i>	39
3.2.2.3	<i>Rapid SCADA</i>	40
3.3	Bancos de Dados	41
3.3.1	<i>Relacionais</i>	41
3.3.2	<i>Não-Relacionais</i>	41
4	SISTEMA DESENVOLVIDO	42
4.1	Aquisição de Dados	42
4.1.1	<i>Métodos de Aquisição</i>	42
4.1.1.1	<i>HTTP</i>	42
4.1.1.2	<i>MQTT</i>	43
4.1.2	<i>Armazenamento dos Dados</i>	43
4.1.2.1	<i>Banco de Dados</i>	44
4.1.2.2	<i>Tipos de Variáveis</i>	44
4.1.2.3	<i>Tempo</i>	44
4.2	Painel de Controle	44
4.2.1	<i>Segurança</i>	44
4.2.1.1	<i>Proteções</i>	45
4.2.1.2	<i>Criptografia</i>	45
4.2.1.3	<i>Controle de Acesso</i>	45
4.2.2	<i>Projetos</i>	45
4.2.2.1	<i>Objetos</i>	45
4.2.2.2	<i>Clientes Associados</i>	45
4.2.3	<i>Clientes</i>	45
4.2.4	<i>Domínios</i>	45
4.3	Recursos Computacionais	45
4.3.1	<i>Armazenamento</i>	46
4.3.2	<i>Processamento</i>	46
4.4	K-NN	46
4.4.1	<i>Classificação</i>	46
4.4.2	<i>Teste</i>	46
4.5	Plataforma Estudantil	46
4.6	Drivers de Comunicação	46

5	INTERFACE WEB	47
6	RESULTADOS	52
6.1	Temperatura e Umidade	52
6.2	Qualidade Sinal - WiFi	52
6.3	Demanda de Roteadores	52
7	CONCLUSÕES E TRABALHOS FUTUROS	55
	REFERÊNCIAS	56
	APÊNDICES	58
	APÊNDICE A – TÍTULO	58
	ANEXOS	58
	ANEXO A – Exemplo de Anexo	59

1 INTRODUÇÃO

A indústria busca frequentemente métodos que possam agilizar e aumentar a eficiência nos seus processos, alguns trabalhos implementam automação para isto, como o (CASSANDRA AMARAL; DANIELLE, 2014), que em um sistema de abastecimento de água, consegue a diminuição no desgaste de motores, menor consumo de energia e um maior controle do processo com supervisão em tempo real através de uma análise detalhada sobre ele. Para que isso aconteça, fazem-se necessárias ações e ferramentas específicas para conduzir uma mudança de forma significativa. Uma gestão interna, confiável e integrada, aumenta a produtividade e diminui o tempo de atuação em determinadas tarefas, (LEE SEUNG-WOO; NAM, 2012) demonstra que com o auxílio de uma Interface em Tempo Real para o controle de um equipamento de produção, é possível o aumento da produtividade além de prever a capacidade produtiva distribuindo eficientemente os recursos de produção. Existe uma infinidade de fatores à serem considerados para a manutenção de um processo, tais como: materiais, equipamentos e qualificação de colaboradores através de procedimentos que sejam capazes de oferecer autonomia e continuidade no serviço.

Com o surgimento de computadores, conectividade e a inclusão de máquinas automáticas no ambiente de trabalho, mediante a terceira revolução industrial, iniciou-se uma necessidade de controle de todas as etapas do processo produtivo, não somente sobre a atuação dos profissionais, mas também sobre as informações específicas de partes do processo. Tornaram-se indispensáveis nestes casos, o uso de tecnologias para que a possibilidade de tomada de decisões, que antes não seriam possíveis devido uma infinidade de informações terem que ser analisadas de forma manual, ou simplesmente não poderem ser adquiridas, sejam agora facilmente implementadas. Uma indústria em que todas as partes do processo são conectadas à rede, denominada "Indústria 4.0", implica em uma quarta revolução industrial, em que seu histórico, desde a evolução da máquina à vapor ao uso de motores movidos à eletricidade e, em seguida, o uso da eletricidade para automatização do processo produtivo através da eletrônica, dá um passo mais adiante agora, a coleta extensiva destas informações que abre possibilidades para manutenções de diagnóstico e/ou preditivas que evitariam eventuais paradas, ou outras tarefas que resultem em ineficiência nas tarefas humanas ou em uso excessivo de recursos.

Comunicação hoje, é algo de vital importância e, com a transição para a Indústria 4.0, serão necessários mecanismos que possam gerir e compatibilizar toda a informação, recursos computacionais que façam o processamento e canais que disponibilizem de forma remota estes dados.

Debater sobre este problema citando um caso que essa dificuldade causou, algum acidente, algum desperdício de dinheiro, ineficiência, dados concretos.

Existem muitos dados, sensores, processos mais complexos a cada dia, dificuldade de mudança ou upgrade no hardware local para acompanhar isso e mostrar porque o processamento remoto é a solução.

2 DISPOSITIVOS E PROTOCOLOS

Nesta seção, serão introduzidas tecnologias existentes para supervisão de processos e aquisição de dados e os protocolos mais utilizados para isto. Vantagens e desvantagens serão comentadas para justificar a escolha do método deste trabalho.

2.1 Automação Industrial

Para o aumento de produtividade de processos, a indústria utiliza diversas técnicas com o objetivo de introduzir máquinas eletromecânicas para a realização de tarefas que demandariam enorme esforço muscular e mental humanos. Além de oferecer um menor custo devido o aumento da capacidade de produção, essas técnicas acabam também por uniformizar o produto final e aumentar sua qualidade. Alguns conceitos utilizados na Automação Industrial e dispositivos empregados serão melhor descritos nesta seção.

2.1.1 Interface Humano-Máquina

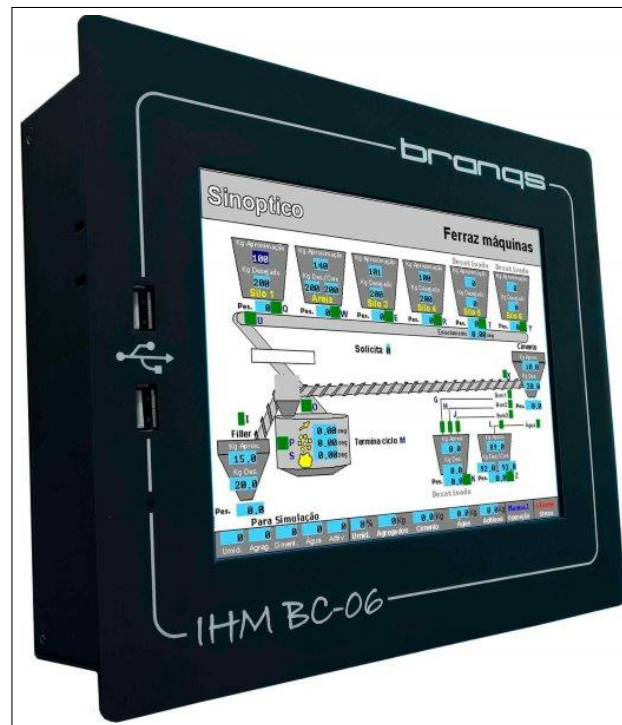
A Interface Humano-Máquina (IHM) é uma ferramenta capaz de oferecer um aspecto visual de um ou mais processos à ela associados e, por meio de telas, fornece informações detalhadas sobre ele(s). Pode possuir teclado ou outras ferramentas para a interação do usuário com o processo final através de programas instalados no(s) dispositivo(s) (FILHO, 2017). A Figura 1 traz um exemplo de uma IHM desenvolvida pela fabricante Branqs, que possui tela de 15 polegadas resistiva e colorida, entradas e saídas digitais integradas, além de outras funções que podem ser utilizadas para fornecer ao operador monitoramento e controle locais do processo em que esteja instalada (BRANQS, 2019).

2.1.2 Unidade de Aquisição de Dados

Unidade de Aquisição de Dados (UAD) são dispositivos que recebem informações relativas ao processo em que estão inseridas e as transferem à um controlador de processo ou diretamente ao sistema de supervisão e controle, onde serão processadas e organizadas para exibição (FILHO, 2017). Dividem-se em duas categorias mais específicas:

- Unidade Dedicada (UD)
- Unidade de Aquisição de Dados e Controle (UADC)

Figura 1 – Exemplo de IHM da fabricante Branqs.



Fonte – (BRANQS, 2019)

2.1.2.1 Unidade Dedicada

É um dispositivo inserido dentro do processo em que se mantenha apenas uma função dedicada (FILHO, 2017), como exemplos: relés digitais, intertravamento, etc.

2.1.2.2 Unidade de Aquisição de Dados e Controle

Tem a função de adquirir dados e controlar ações nos equipamentos respectivos, são compostos por cartões de eletrônicos associados cada um à uma função específica, unidades lógicas, memórias, entradas e saídas de dados digitais ou analógicos (FILHO, 2017). Dentre elas, as mais comuns são:

- Controlador Lógico Programável
- Unidade Terminal Remota

2.1.2.2.1 Controlador Lógico Programável

Controlador Lógico Programável (CLP) é a UADC mais utilizada para controle de equipamentos através de programas desenvolvidos externamente pelo utilizador e nele gravados, simulando à nível de *software* e substituindo: chaves, contadores, temporizadores, relés e outros

dispositivos. Permitem a inclusão de cartões eletrônicos para a realização de diferentes tarefas específicas. Possui IHM, onde o utilizador pode alterar a programação ou executar tarefas configuradas no CLP (FILHO, 2017). A Figura 2 mostra um CLP da fabricante WEG, de modelo PLC300 que possui todas as características aqui descritas.

Figura 2 – Exemplo de CLP de modelo PLC300 da fabricante WEG.



Fonte – (WEG, 2019b).

2.1.2.2.2 Unidade Terminal Remota

Unidade Terminal Remota (UTR) é uma UADC responsável por coletar informações e executar comandos de equipamentos do processo, sejam eles digitais ou analógicos. Possuem capacidade de executar programas em modo local independente do sistema de supervisão, ao mesmo tempo que possui capacidade de integração com o mesmo. Os comandos locais para equipamentos são feitos através de relés de maneira similar ao que ocorre no CLP, por rotinas específicas armazenadas em programas gravados na própria UTR (FILHO, 2017). A Figura 3 mostra um UTR da fabricante WEG, de modelo RUW-03 que possui todas as características aqui descritas.

Figura 3 – Exemplo de UTR de modelo RUW-03 da fabricante WEG.



Fonte – (WEG, 2019a)

2.2 Protocolos de Comunicação

A comunicação entre os dispositivos citados na seção anterior e outros, como: sensores, válvulas e atuadores em geral, é essencial para o funcionamento conjunto e ordenado dos mesmos. Desta forma, várias opções foram desenvolvidas ao longo do tempo para tornar a comunicação mais confiável, alguns dos protocolos mais utilizados atualmente são descritos nesta seção.

2.2.1 Modbus

Modbus é um protocolo de comunicação de dados voltado à automação industrial. Desenvolvido em 1979, pela *Modicon*, é até hoje utilizado vastamente na indústria em CLPs para comandos e aquisição de informações. Podem ser utilizados os padrões: RS-232, RS-485 ou Ethernet para a camada física de ligação, através de sinais discretos ou analógicos. É geralmente utilizado no tipo mestre-escravo, onde os escravos só enviam comunicação quando solicitadas pelo mestre (ORGANIZATION, 2019).

2.2.1.1 Modbus Serial

Em redes baseadas em RS-232 e RS-485, a comunicação do Modbus é feita de forma serial através de dois modos distintos: Remote Terminal Unit (RTU) e American Standard Code for Information Interchange (ASCII). A Figura 4 e Tabela 4 representam os pinos da estrutura física RS-485 utilizado pelo Modbus Serial (ORGANIZATION, 2019).

No RTU, para cada byte transmitido, são codificados 2 caracteres. Os números variam entre -32768 e 32767, o tamanho da palavra RTU é de 8 bits.

Tabela 2 – Representação do pacote no modo RTU.

Endereço Escravo	Código Função	Dados	CRC
1 byte	1 byte	0 a 252 bytes	2 bytes (CRC-16)

Fonte – (ORGANIZATION, 2019).

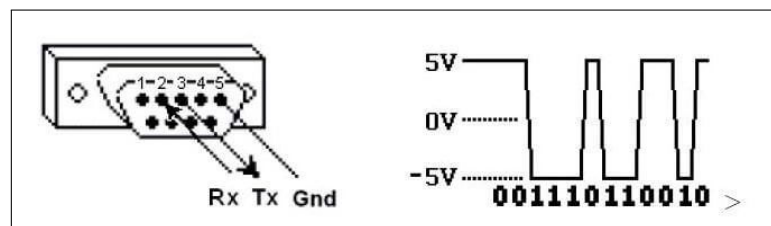
No ASCII, os dados são codificados com base na tabela ASCII, em que cada byte é transmitido através de dois caracteres. O tamanho da palavra ASCII é de 7 bits, utilizando-se caracteres de intervalos 0-9 ou A-F e entre duas mensagens, 3-5 caracteres.

Tabela 3 – Representação do pacote no modo ASCII.

Início Final	Endereço	Função	Dados	LRC
":"(ASCII 0x3Ah) CR+LF (ASCII 0x0Dh + 0x0Ah)	2 caracteres	2 caracteres	0 a 2 x 252 caracteres	2 caracteres

Fonte – (ORGANIZATION, 2019).

Figura 4 – Pinagem do conector RS485 utilizado no protocolo Modbus Serial.



Fonte – se.com - Acessado em: 28/03/2019

2.2.1.2 Modbus TCP/IP

As redes baseadas em Ethernet, sob o protocolo *Transmission Control Protocol / Internet Protocol* (TCP/IP), se tornaram o método de transporte comum na Internet. O TCP/IP é um conjunto de protocolos em camadas, que oferece confiabilidade no transporte de dados entre máquinas, e devido à isso, este padrão torna-se uma opção ideal para sistemas empresariais corporativos. O Modbus TCP/IP tornou-se muito utilizado devido sua simplicidade e baixo custo, demandando *hardwares* mínimos para ser utilizado. A maioria dos dispositivos Modbus

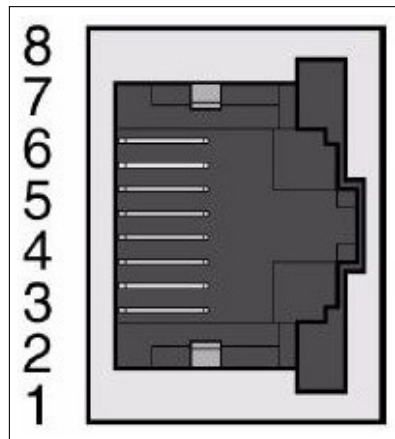
Tabela 4 – Pinagem do conector RS485 utilizado no protocolo Modbus Serial.

Pino	Sinal
1	Não conectado
2	RX - Recepção
3	TX - Envio
4	Não conectado
5	Gnd - Terra
6	Não conectado
7	Não conectado
8	Não conectado
9	Não conectado

Fonte – Adaptado de se.com - Acessado em: 28/03/2019

atualmente presentes no mercado, suportam o padrão TCP/IP, aumentando a cada ano a disponibilidade. Há também a possibilidade de conversão entre TCP/IP e Serial, onde é possível garantir a retrocompatibilidade entre dispositivos. A Figura 5 e Tabela 6 trazem a representação do conector utilizado no Modbus TCP/IP (ORGANIZATION, 2019).

Figura 5 – Pinagem do conector RJ45 utilizado no protocolo Modbus Ethernet.



Fonte – Adaptado de se.com - Acessado em: 28/03/2019

2.2.2 OPC

Open Platform Communications (OPC), inicialmente chamado *Object Linking and Embedding for Process Control*, desenvolvido pela *OPC Foundation* em 1996 e gerenciado por esta desde então, é o padrão de interoperabilidade para o transporte seguro e confiável de informações no espaço industrial, ele é independente de plataforma e garante o fluxo contínuo de informações entre dispositivos de vários fornecedores. É uma série de especificações desen-

Tabela 5 – Pinagem do conector RJ45 utilizado no protocolo Modbus Ethernet.

Pino	Sinal
1	CAN_H
2	CAN_L
3	CAN_GND
4	D1 - RS485 (Modbus)
5	D0 - RS485 (Modbus)
6	Não conectado
7	VP - Reservado ao conversor RS232/RS485
8	Comum

Fonte – Adaptado de se.com - Acessado em: 28/03/2019

volvidas por fornecedores do setor, usuários e desenvolvedores. Essas especificações definem a interface entre Clientes e Servidores, bem como Servidores e Servidores, incluindo acesso a dados em tempo real, monitoramento de alarmes e eventos, acesso a dados históricos e outros aplicativos. (FOUNDATION, 2019a)

Seu propósito inicial era agregar vários outros tipos de protocolos distintos de CLPs, proprietários ou não, como: Modbus (seção 2.2.1), Profibus, etc, de forma simplificada, para que IHMs ou SCADAs (seção 3) pudessem solicitar informações através de comunicação genérica, permitindo então, que os usuários implementassem sistemas usando os melhores produtos, interagindo perfeitamente via OPC.

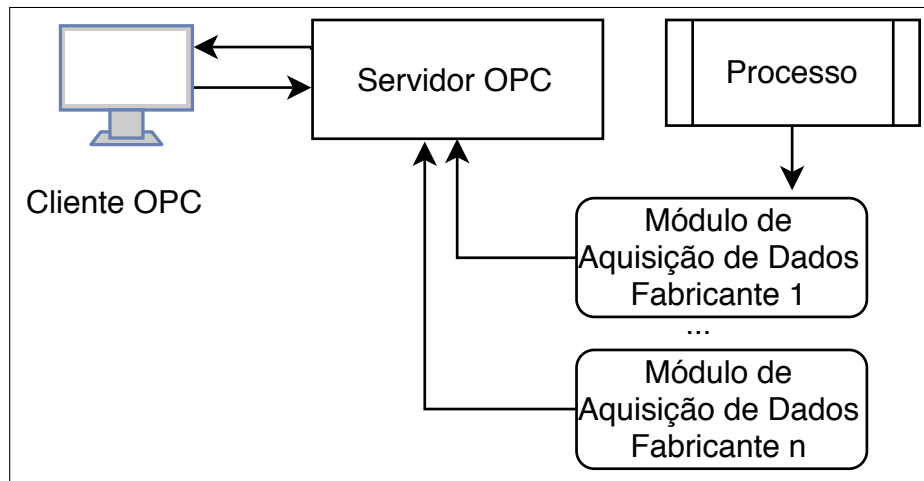
2.2.2.1 OPC Classic

Inicialmente, o padrão OPC era restrito e baseado na plataforma *Windows*, utilizando *Distributed Component Object Model* (COM/DCOM) para a comunicação entre *softwares*. Como visto na sigla original do OPC, ele era suportado por *Object Linking and Embedding* (OLE) voltado à Controle de Processo, essas especificações, agora conhecidas como *OPC Classic*, tiveram ampla adoção em vários setores (FOUNDATION, 2019b). Na Figura 6 é representado funcionamento de um sistema que utilize *OPC Classic*, onde o Servidor OPC recebe informações de inúmeros módulos de aquisição de dados, de diferentes fabricantes e diferentes protocolos e permite a interoperabilidade deles com o Cliente OPC, de forma genérica.

Existem três definições principais:

- Acesso a Dados - *OPC Data Access (DA)*: onde ocorrem troca de dados, incluindo valores, tempo e informações de qualidade.
- Alarmes e Eventos - *OPC Alarms & Events (AE)*: para troca de mensagens de

Figura 6 – Diagrama do funcionamento de um sistema que utilize OPC Classic.



Fonte – O autor

alarmes e tipos de eventos, estados de variáveis e gerenciamento de estados.

- Acesso a Dados Históricos - *OPC Historical Data Access (HDA)*: define os métodos de consulta e quais análises podem ser aplicadas a dados históricos, com registro de data e hora.

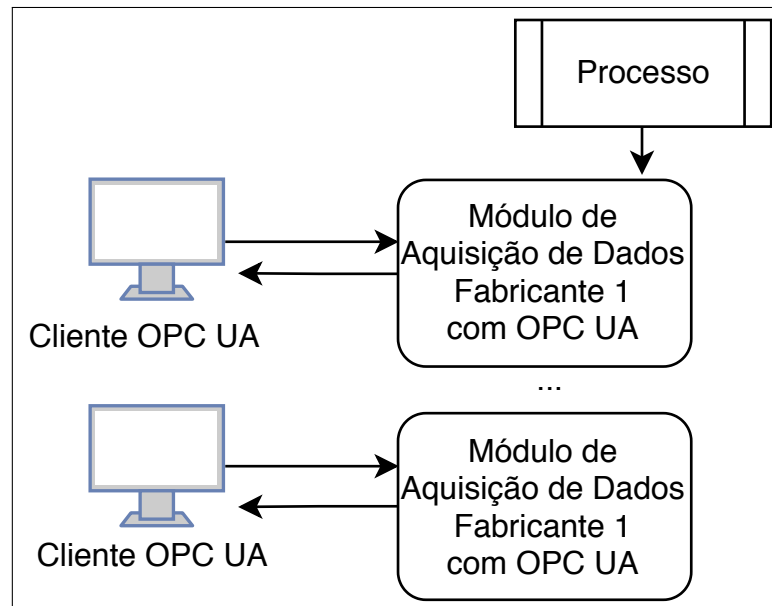
2.2.2.2 OPC-UA

Com a introdução de arquiteturas orientadas a serviços em sistemas de manufatura, surgiram novos desafios em segurança e modelagem de dados, a *OPC Foundation* desenvolveu as especificações do *Object Linking and Embedding for Process Control - Unified Architecture* (OPC-UA) em 2008, sendo uma arquitetura orientada a serviços independentes, aberta e escalável.

O OPC-UA integra todas as funcionalidades do *OPC Classic*, além de outras melhorias, como:

- Segurança: criptografia de 128 ou 256 bits, verificação de erros para que a mensagem recebida seja exatamente a mensagem enviada, autenticação através de certificados e níveis de permissão.
- Extensível: é possível adicionar novos recursos mantendo a compatibilidade com aplicações já existentes.
- Descoberta: permite a busca por servidores OPC na rede ou em computadores.
- Hierarquia: todos os dados são dispostos de forma hierárquica, permitindo informações simples e complexas na mesma estrutura.

Figura 7 – Diagrama do funcionamento de um sistema que utilize OPC-UA.



Fonte – O autor

- Auditoria: os dados a serem lidos/escritos possuem permissões de acesso tais como registros sobre sua utilização.
- Independência de plataforma: funciona em computadores tradicionais e servidores em nuvem, seja o sistema operacional *Linux*, *Windows* ou outros, CLPs, micro-controladores, etc.

2.2.3 HTTP

Hypertext Transfer Protocol (HTTP), coordenado pela *World Wide Web Consortium* (W3C), é um protocolo de comunicação à nível de aplicação para distribuição de informação de hipermídia, é base para comunicação pela *World Wide Web* (WEB) desde 1990. Inicialmente, em sua versão HTTP/0.9, era um simples protocolo de transferência de dados não tratados através da Internet e em sua versão atual HTTP/1.1, lançado em 1999, foram implementadas outras funcionalidades como a possibilidade de troca de mensagens no formato *Multipurpose Internet Mail Extensions* (MIME), que carregam consigo metainformações sobre a requisição ou resposta e o corpo das informações transferidas (SOCIETY, 1999a).

A transferência de informação acontece através de *sockets* sob o protocolo TCP/IP, onde com a arquitetura cliente-servidor, o cliente envia uma requisição ao servidor, com o padrão MIME e localizado através endereços, como o *Uniform Resource Identifier* (URI), que identifica a informação acessada e *Uniform Resource Locator* (URL), que determina a localização desta

informação, a conexão é completada e o servidor retorna o *status* de acordo com o sucesso ou não da requisição e possíveis conteúdos também em formato MIME caso sejam necessários, encerrando assim a conexão.

2.2.3.1 *HTTPS*

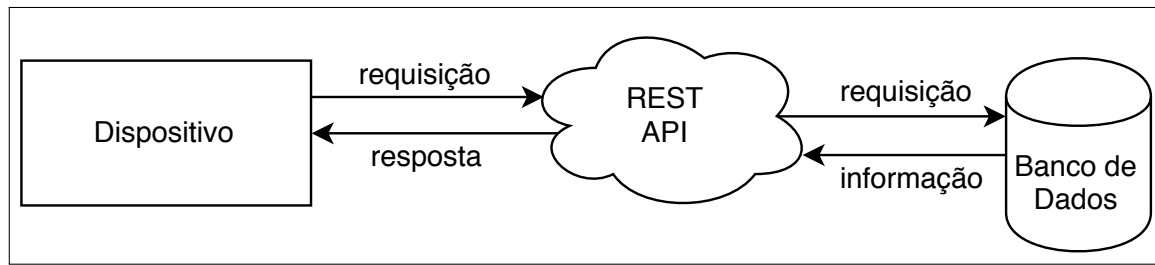
O *Hyper Text Transfer Protocol Secure* (HTTPS) é uma derivação do protocolo de comunicação HTTP para mensagens seguras, projetado como uma camada de segurança utilizando o protocolo *Transport Layer Security* (TLS). Além de fornecer uma variedade de mecanismos de segurança para clientes e servidores, não são necessárias chaves públicas do lado cliente, suporta criptografia ponta a ponta e torna possível verificar a autenticidade do servidor através de certificados digitais. Seu uso é recomendado em redes inseguras, evitando a clonagem das informações trafegadas que poderia acontecer na ausência de criptografia (SOCIETY, 1999b). Segundo (HELME, 2019), em fevereiro de 2019, mais de 58.44% dos 1 milhões *websites* mais visitados da *internet* já utilizavam o HTTPS.

2.2.3.2 *REST*

O *Representational State Transfer* (REST) é um estilo de arquitetura para *WEB Service*, uma solução padronizada pela W3C e *Organization for the Advancement of Structured Information Standards* (OASIS) que busca fornecer interoperabilidade entre dispositivos e aplicações pela internet utilizando diferentes tipos de linguagens, o que a torna compatível com a maioria das aplicações já existentes (W3C, 2004).

O envio e recebimento das mensagens é realizada de forma simplificada através dos protocolos HTTP ou HTTPS utilizando os formatos: *eXtensible Markup Language* (XML), *JavaScript Object Notation* (JSON) ou *Hypertext Markup Language* (HTML) e métodos de chamada bem definidos: GET, POST, PUT, PATCH e DELETE. Comumente utilizado por empresas no o desenvolvimento de *Application Programming Interface* (API) para acesso a informações específicas sobre serviços, aplicações, faturas, etc. A Figura 11 traz uma ideia geral sobre o funcionamento de uma API que utiliza a arquitetura REST para troca dos dados.

Figura 8 – Representação do funcionamento de uma API REST.



Fonte – O autor

2.2.3.3 Métodos de Chamada

Quando uma nova requisição é feita, é necessário definir o método que será utilizado. Os métodos de chamada são padronizados para o protocolo HTTP e são conhecidos como verbos, pois identificam a ação que será executada pela requisição. Os mais comuns são:

- GET: Apenas recebe informações, as requisições devem ser seguras e idempotentes, ou seja, independente de quantas vezes ela seja repetida, com os mesmos parâmetros, o resultado sempre deve ser o mesmo. Podem haver solicitações parciais ou condicionais.
- POST: Envia e recebe informações, é utilizado na criação de novos "objetos" (elementos da aplicação), mas também é comum o uso para atualização destes.
- PUT: Envia e recebe informações, é utilizado na atualização de "objetos" já existentes, na falta do envio de algumas informações necessárias, estas são consideradas nulas ou vazias. Assim como o GET, o PUT é idempotente.
- PATCH: Envia e recebe de informações, similar ao PUT, é utilizado na atualização de "objetos" já existentes, porém, apenas os campos especificados.
- DELETE: Envia e recebe de informações, é utilizado na exclusão de "objetos", podendo ser imediato ou não.

2.2.3.4 Formatos de Conteúdo

São tipos de linguagem de marcação para necessidades especiais com a finalidade de transferência de informações pela *internet*. Os mais comuns são:

- HTML (texto puro).
- XML: É baseado em texto simples, de simples leitura, pode representar listas,

Tabela 6 – Comparativo entre os métodos de chamada.

Método	Descrição	Seguro	Idempotente
GET	Recebe informações	sim	sim
POST	Cria objetos	não	não
PUT	Atualiza objetos, na falta de informações, considera como nulas	não	sim
PATCH	Atualiza objetos, alterando apenas as informações enviadas	não	não
DELETE	Exclui objetos, imediatamente ou não	não	sim

Fonte – O autor

registros e árvores. Seu próprio formato descreve sua estrutura, campos e valores, os dados são organizados de forma hierárquica e é editável em qualquer ambiente (SOCIETY, 2001). A Figura 9 traz um exemplo de utilização do formato XML.

Figura 9 – Exemplo de informações organizadas no formato XML.

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <sensores>
3    <temperatura>
4      <dados>
5        <hora>2019-04-15 00:30:00</hora>
6        <valor>25</valor>
7      </dados>
8      <dados>
9        <hora>2019-04-15 00:31:00</hora>
10       <valor>25</valor>
11     </dados>
12   </temperatura>
13   <umidade>
14     <dados>
15       <hora>2019-04-15 00:30:00</hora>
16       <valor>80</valor>
17     </dados>
18     <dados>
19       <hora>2019-04-15 00:31:00</hora>
20       <valor>80</valor>
21     </dados>
22   </umidade>
23 </sensores>

```

Fonte – O autor

- JSON: É um formato leve de informações, de simples leitura e análise. Assim como o XML é hierárquico, em pares, ou seja, para cada rótulo, há um valor associado ou sub-conjunto destes. A Figura 10 traz um exemplo de utilização do formato JSON (JSON, 2019).

Figura 10 – Exemplo de informações organizadas no formato JSON.

```

1  {
2    "sensores": {
3      "temperatura": {
4        "dados": [
5          {
6            "hora": "2019-04-15 00:30:00",
7            "valor": "25"
8          },
9          {
10           "hora": "2019-04-15 00:31:00",
11           "valor": "25"
12         }
13       ]
14     },
15     "umidade": {
16       "dados": [
17         {
18           "hora": "2019-04-15 00:30:00",
19           "valor": "80"
20         },
21         {
22           "hora": "2019-04-15 00:31:00",
23           "valor": "80"
24         }
25       ]
26     }
27   }
28 }

```

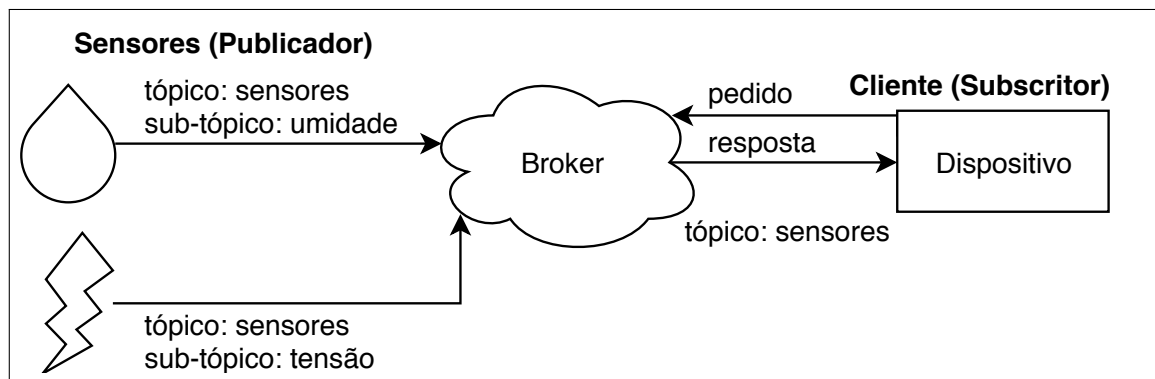
Fonte – O autor

2.2.4 MQTT

O *Message Queue Telemetry Transport* (MQTT) foi desenvolvido por Dr. Andy Stanford-Clark, da IBM, e Arlen Nipper, da Arcom no ano de 1999. É um protocolo de mensagens extremamente simples e leve, projetado para ser utilizado em dispositivos que tenham restrição de largura de banda, alta latência ou baixa confiabilidade. Baseia-se na topologia publicador/assinatura, onde as mensagens são enviadas com identificação através de tópicos (*topics*) ou sub-tópicos, o que permite uma única mensagem ser destinada à múltiplos receptores com apenas um envio, ou da mesma forma, receber informações agrupadas de vários sub-tópicos. O elemento responsável pelo envio e recebimento de mensagens é denominado *broker*, que funciona como uma central, intermediando as informações enviadas pelos dispositivos e aplicações da rede (MQTT, 2019). A Figura 11 apresenta um exemplo de utilização ao qual um subscritor (possível dispositivo associado à rede) recebe informações de dois sensores utilizando um único tópico.

A autenticação é feita através de usuário e senha, com a possibilidade de conexão criptografada e a escolha de três níveis de serviço (prioridades na transmissão) que dependerá do projeto em questão, qualidade de conexão do dispositivo, entre outros, sendo elas:

Figura 11 – Representação do funcionamento do MQTT.



Fonte – O autor

- **Nível 0:** Não é feita quaisquer confirmações sobre a entrega da informação, de forma que a mensagem é descartada após o envio.
- **Nível 1:** São feitas várias tentativas de entrega até que se obtenha confirmação no recebimento, mesmo que isso implique no recebimento em duplicidade.
- **Nível 2:** Há garantia de que a mensagem só será entregue uma vez, havendo tanto a confirmação de entrega da mensagem como a confirmação da confirmação de entrega.

3 SISTEMAS SCADA

Sistemas de Supervisão e Aquisição de Dados, do inglês, *Supervisory Control and Data Acquisition* (SCADA), consistem basicamente de *softwares* que monitoram e operam partes de um ou mais processos, através de unidades de aquisição de dados, como o CLP, que por sua vez, conectado fisicamente ao servidor SCADA e aos atuadores, utiliza protocolos de comunicação como os citados na seção 2.1 para obtenção e armazenamento destas informações. Com o domínio sobre as informações do processo, esta ferramenta é capaz de apresentar através de uma IHM e de forma simplificada, valores e estados gerais do processo que se deseja atuar, desta forma, obtêm-se um maior controle sobre a tarefa, pois é possível centralizar a leitura de todos os sensores atuantes, a categorização e histórico destes dados, além da priorização de pendências do processo neste único sistema, reduzindo assim a necessidade um maior número de trabalhadores especializados que desempenhem a mesma função. (DANEELS; SALTER, 1999)

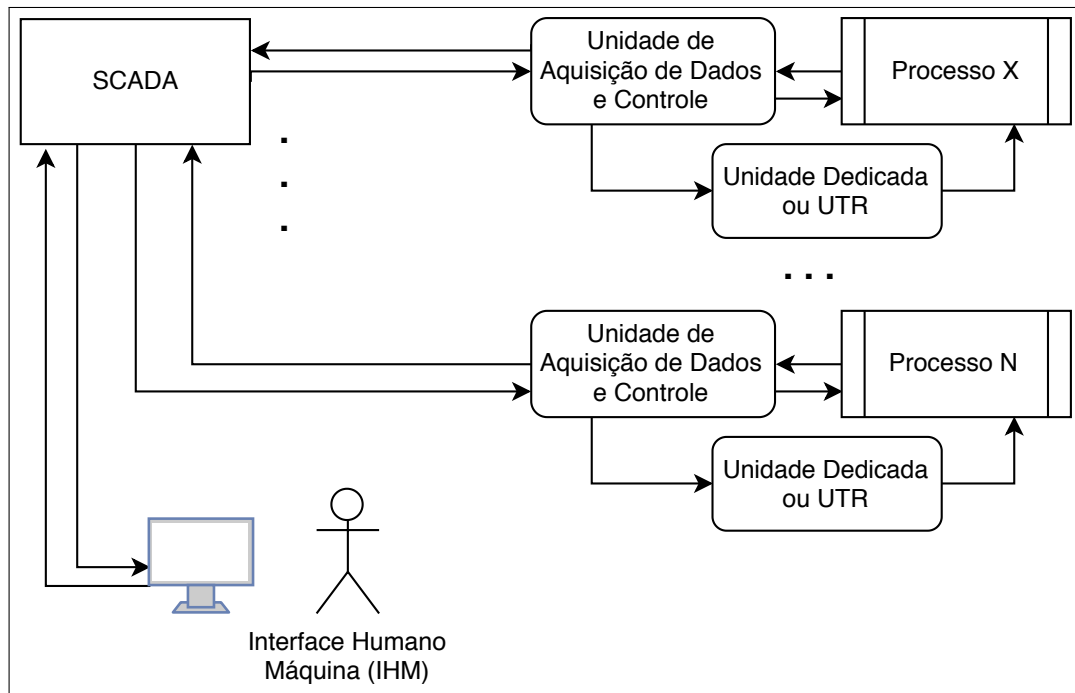
O SCADA apresenta uma série de vantagens, dentre elas: (i) redução de custos, devido a possibilidade de geração de relatórios detalhados úteis ao planejamento estratégico, evidenciando possíveis vícios do processo produtivo, (ii) maior desempenho na produção, por determinar os valores ótimos de trabalho, (iii) confiabilidade e continuidade, devido a existência de alarmes críticos, ou seja, notificações visuais ao operador, quando alguma variável ou condição do processo esteja em desacordo com o padrão de operação, desta forma, possíveis problemas que ocasionariam uma maior parada na produção, são mitigados com intervenções de forma quase imediata pelo operador caso sejam necessárias, trazendo assim vantagem competitiva.

Todas as informações do processo podem coletadas e armazenadas em tempo real em um banco de dados, podendo serem implementadas no sistema de gestão empresarial da empresa ou utilizadas para cálculos mais complexos, sendo o último realizado por outras máquinas para garantir a que o SCADA não tenha seu desempenho prejudicado. Uma representação básica do sistema SCADA é ilustrada na Figura 12, ao qual todas as informações do processo são centralizadas e exibidas de forma simplificada ao colaborador.

3.1 SCADA WEB

Rede Mundial de Computadores (WEB), é como se designa o sistema de hiperligações e marcação de texto que permitem a disponibilidade de conteúdo através da *internet*, como: páginas de texto, documentos, músicas, entre outros (W3C, 2004). O SCADA WEB é uma

Figura 12 – Arquitetura física básica do SCADA.



Fonte – O autor

versão elaborada do SCADA convencional, onde os dados são transferidos para servidores na *internet* e posteriormente processados, integrados às demais plataformas e/ou vistos em páginas WEB. A transição do SCADA para SCADA WEB ocorre principalmente devido à superação de uma baixa largura de banda e restrições de comunicação como ocorria antigamente. Os avanços tecnológicos possibilitaram a rápida expansão dos canais de dados através da *internet*, onde até mesmo a transmissão de informações em tempo real, não é mais um fator limitante. (OSMIC NEDIM; VELAGI, 2017)

Sistemas SCADA com base na *internet* podem se tornar uma parte importante do funcionamento de sistemas de controle, onde o XML e outros formatos disponíveis, podem oferecer possibilidade de resolução de problemas de incompatibilidade que existiam no SCADA convencional, onde as fontes de comunicação com o processo são físicas e tornam necessários protocolos de comunicação específicos ou adaptações como o OPC. Esta transição também ocorre com os dispositivos utilizados, como CLPs e outros, que passam à contar com comunicação TCP/IP integrada, alguns deles já com possibilidade de conexão *Wi-Fi*, permitindo a utilização de protocolos como o HTTP e MQTT, nativos da WEB, diretamente do dispositivo, removendo grande parte da camada física. A Figura 13 traz um exemplo da fabricante WAGO, com uma família de dispositivos que possuem conexão direta à internet, além de recursos de criptografia.

Figura 13 – Família de Dispositivos com Comunicação Integrada da fabricante WAGO.



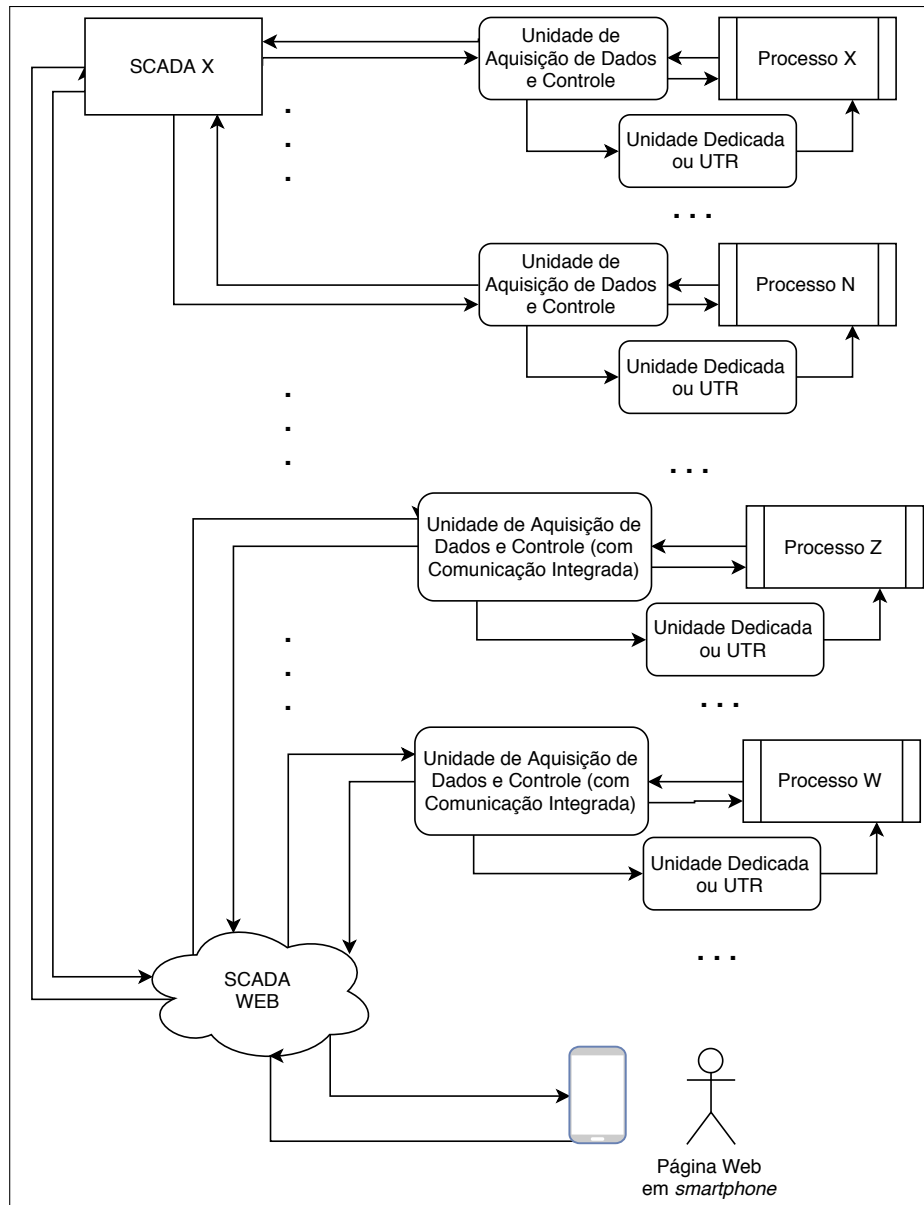
Fonte – (WAGO, 2019)

Independente de como são obtidas as informações, a visualização se dá de forma mais familiar ao usuário, onde antes seria feita através de uma IHM física, este ato se reduz à um simples *smartphone* ou página no navegador simulando esta interface, sem ser necessária a instalação de qualquer *software* adicional. A convergência das novas tecnologias, afeta drasticamente a supervisão destas informações, possibilitando controle distribuído e a possibilidade de armazenamento destas informações em qualquer lugar do mundo, com uma ampla capacidade de recursos (LIPNICKAS ARUNAS; RUTKAUSKAS, 2009).

A implementação de um SCADA WEB, não só abre possibilidade de armazenamento de dados em várias localizações, como também eleva a capacidade de recursos computacionais à um nível muito superior, devido à possibilidade de utilização de servidores em nuvem - interligação de vários servidores através da internet formando um núcleo único de processamento - é possível controlar além de grupos de processos, grupos de plantas em único sistema. Segundo (KARNOUSKOS; COLOMBO, 2011), a nova geração do SCADA pode modificar significamente a forma de projetar e implementar os processos industriais no futuro, onde o sistema terá que lidar com uma quantidade muito superior de dados distribuídos e informações em tempo real para tomar as decisões baseadas neste dados com informações internas e externas. A IHM fica não mais limitada à um local físico, mas acessível através de todos os computadores, *smartphones* e *tablets* conectados à *internet*, permitindo a colaboração simultânea na supervisão do processo. Com o uso de várias plantas simultaneamente, há a possibilidade de implementação de uma rotina por prioridades, dependendo das necessidades dinâmicas de cada cliente. Poucas são as desvantagens de um sistema SCADA WEB, uma delas, seria a perda de robustez do sistema devido as informações e controle serem feitos à distância através da rede, tendo que serem

considerados atraso em transporte que apesar de ser irrisório para a maioria das aplicações, podem chegar a ser um problema para outras. Uma representação básica do sistema SCADA WEB é ilustrada na Figura 14, onde as informações de um grupo de plantas são centralizadas e exibidas de forma simplificada ao colaborador.

Figura 14 – Arquitetura de um SCADA WEB.



Fonte – O autor

3.2 Sistemas SCADA disponíveis no mercado

3.2.1 Sistemas proprietários

3.2.1.1 Elipse E3

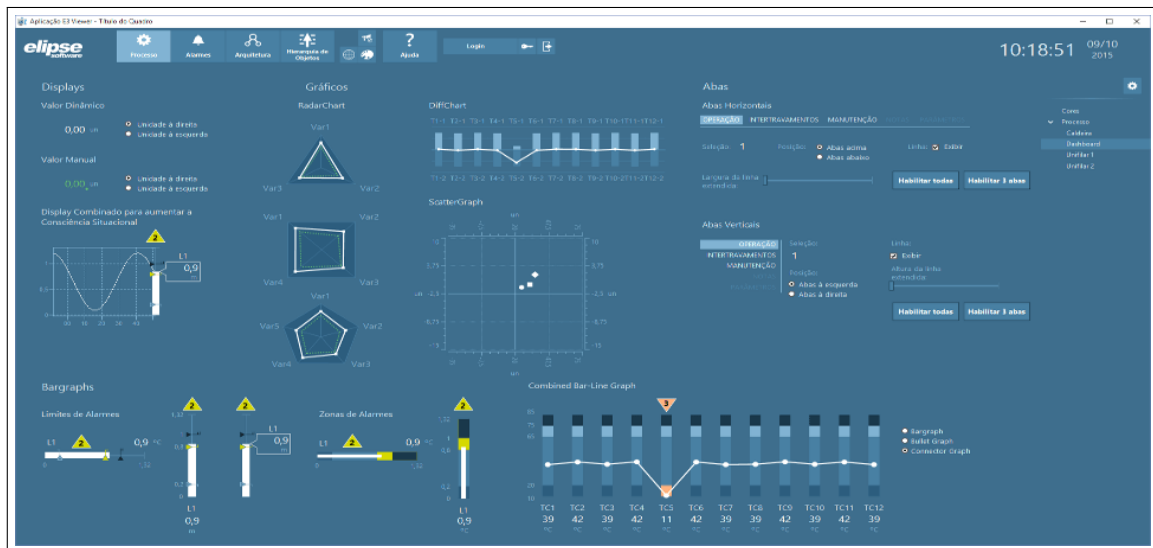
O Elipse E3 (ELIPSE, 2019), desenvolvido pela empresa Elipse Software, representa a terceira geração do SCADA. Utiliza o conceito de múltiplas camadas, onde incluem: servidores, regras de aplicação ou de negócio e estações clientes. O sistema é composto por 3 aplicações:

- *E3Server*: é o servidor das aplicações, em que se gerencia todos os processos de execução do *software* e processa a comunicação entre eles. Suas ações são basicamente: envio das informações gráficas e dados para o cliente, gerenciamentos dos processos de E/S e comunicação com os diversos pontos de aquisição, controle da cópia de produtos, cliente e servidor OPC e sincronismo de alarmes e bases de dados. Permite também a distribuição deste serviço entre várias máquinas de acordo com a necessidade, com objetivo de manter a continuidade em uma eventual falha.
- *E3Viewer*: responsável pela interface de operação e visualização da aplicação que se encontra no *E3Server*, com operação local ou via *intranet/internet*, pode ser acessado por diversas plataformas como: Mac OS, Linux, Windows CE ou ainda, há a possibilidade de utilização do *E3WebServer* para gerenciamento adicional do acesso via Internet.
- *E3Studio*: ferramenta para configuração do sistema, servindo como plataforma universal do desenvolvimento. A configuração e execução compartilham da mesma base de dados, de forma que as edições das aplicações podem ser enviadas em *runtime*, sem ser necessário a parada da aplicação, independente de ser feita local ou remotamente. É possível a edição de mais de um aplicativo ao mesmo tempo ou a edição ser feita por mais de uma pessoa devido compartilharem o mesmo servidor. Possui ferramentas, como: editores de telas, relatórios e *scripts*.

Outras informações importantes:

- Possui drivers para comunicação com mais de 300 tipos de dispositivos e sistemas, sejam eles proprietários ou OPC, além de produzir drivers sob encomenda.
- Possui interfaces específicas para Access (.MDB), SQL Server/MSDE, Oracle ou

Figura 15 – Demonstração da tela de processo do *software* Elipse E3.



Fonte – (ELIPSE, 2019).

acesso genérico através de padrões ADO e ODBC, faz acesso à base de dados corporativas fazendo o interfaceamento entre o processo e sistemas administrativos, de produção, manutenção e gestão.

3.2.1.2 InduSoft Web Studio®

O InduSoft Web Studio® (INDUSOFT, 2019), desenvolvido pela empresa InduSoft, fornece componentes básicos de automação para o desenvolvimento de IHMs, sistemas SCADAs e soluções de instrumentação embarcada. O sistema é composto por 2 aplicações:

- *Server*: é o servidor das aplicações, em que se gerencia todos os processos de execução do *software* e processa a comunicação entre eles. Suas ações são basicamente: envio das informações gráficas e dados para o cliente, gerenciamentos dos processos de entrada e saída e comunicação com os diversos pontos de aquisição, controle da cópia de produtos, servidor OPC e sincronismo de alarmes e bases de dados.
- *IoTViewer*: responsável pela interface de operação e visualização da aplicação que se encontra no *Server*, com operação local ou via *intranet/internet*.

Outras informações importantes:

- A aplicação *Server* suporta as plataformas *Microsoft*, como: *Windows CE*, *Mobile*, *XP Embedded* e *Server*, enquanto a aplicação cliente, o *IoTView*, pode também suportar plataformas, como: *Linux* e *VXWorks*.

Figura 16 – Demonstração da tela de processo do *software* InduSoft Web Studio®.



Fonte – (INDUSOFT, 2019).

- Permite visualização de processo através de Navegador WEB, podendo ser acessado através de celulares ou computadores de mesa, seja em rede local ou pela *internet*.
- Possui suporte para CLP ou controlador e drivers para comunicação com mais de 200 tipos de dispositivos e sistemas, sejam eles proprietários ou OPC, além de comunicação por TCP/IP.
- Alarmes podem ser enviados via *e-mail*, celulares ou através do próprio navegador.
- Permite acesso à base de dados corporativas fazendo o interfaceamento entre o processo e sistemas administrativos, de produção, manutenção e gestão.

3.2.2 Sistemas de código aberto

3.2.2.1 ScadaBR

O ScadaBR (SCADABR, 2019) é um *software* livre e de código-fonte aberto. Abrange profissionais de automação, universidades, escolas técnicas e empresas de todos os

portes. O projeto foi iniciado em 2006, por iniciativa da empresa MCA Sistemas com sede em Florianópolis - SC, que com o auxílio de outras empresas, a fundação CERTI e a Universidade Federal de Santa Catarina - UFSC, desenvolveram o sistema de forma completa em português baseado no *software* Mango. Possui apoio da FINEP, SEBRAE e CNPq, que também, financiaram a iniciativa durante 2 anos.

Figura 17 – Demonstração de telas, incluindo a de processo, do *software* ScadaBR.



Fonte – (SCADABR, 2019).

Outras informações importantes:

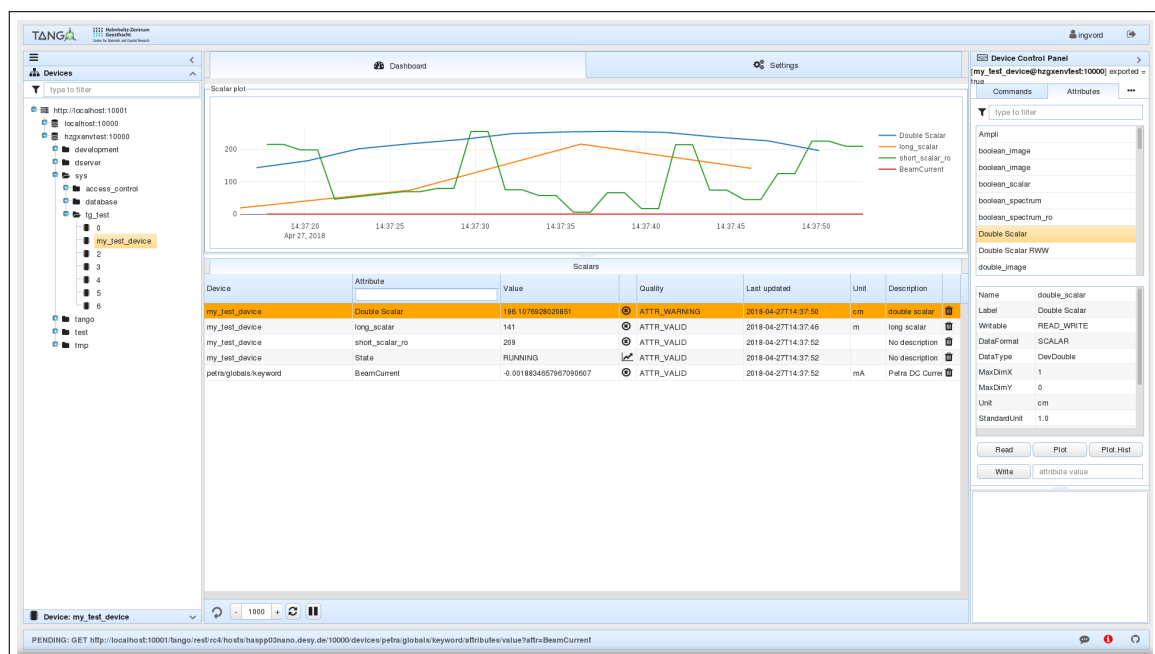
- A aplicação *Server* suporta diferentes plataformas, como: *Windows* 32/64 bits e *Linux*.
- Permite visualização de processo através de Navegador WEB, podendo ser acessado através de celulares ou computadores de mesa, seja em rede local ou pela *internet*.
- Possui mais de 20 protocolos de comunicação, como: Modbus TCP/IP e Serial, HTTP, etc.
- Customização de *scripts* para controle, automação, etc.
- Possibilidade de cálculos com funções matemáticas, estatísticas etc, com as variáveis do processo.

- Níveis de permissão de usuários, com controle de acesso.

3.2.2.2 TANGO Controls

O TANGO Controls (CONTROLS, 2019) é um *software* livre e de código-fonte aberto. Foi desenvolvido pelo *European Synchrotron Radiation Facility* em Genebra, França e seu desenvolvimento já supera 20 anos de duração. Foi desenvolvido principalmente para necessidades de instalações de pesquisas, com o conceito agregado de ser criado um novo *framework*. Pode ser executado de forma autônoma ou distribuída, local ou remota.

Figura 18 – Demonstração da tela de processo do *software* TANGO Controls.



Fonte – (CONTROLS, 2019).

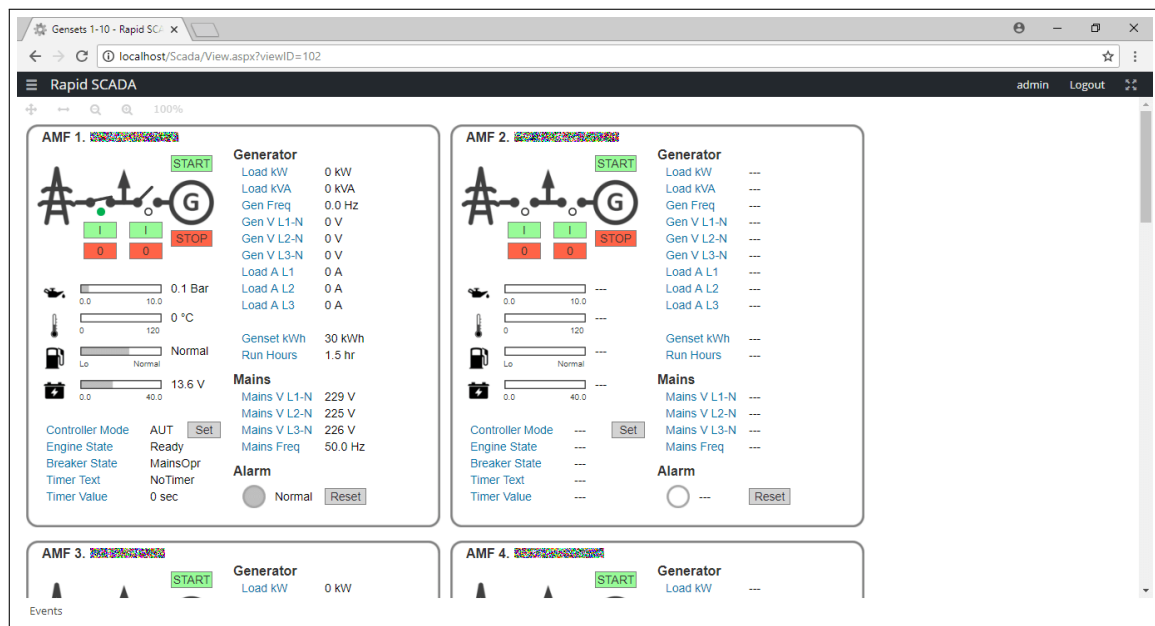
Outras informações importantes:

- Permite visualização de processo através de Navegador WEB, podendo ser acessado através de celulares ou computadores de mesa, seja em rede local ou pela *internet*.
- Possui diversos *drivers* de comunicação, disponibilizados de forma gratuita.
- Permite a adição de funções analíticas para a tomada de decisões.
- Encontra-se em fase de transição para atender a demanda *IoT* industrial.

3.2.2.3 Rapid SCADA

O Rapid SCADA (SOFTWARE, 2019) é um *software* livre e de código-fonte aberto, desenvolvido pela empresa russa *Rapid Software*.

Figura 19 – Demonstração da tela de processo do *software* Rapid SCADA.



Fonte – (SOFTWARE, 2019).

Outras informações importantes:

- É suportado por plataformas *Windows* e *Linux*.
- O sistema possui uma interface de administração no modelo cliente/servidor e outra de monitoramento no formato WEB.
- Possui *drivers* de comunicação, disponibilizados de forma gratuita, como: Modbus, OPC, MQTT, etc.
- Níveis de permissão de usuários, com controle de acesso.
- Alarmes de fogo e segurança, com avisos via interface.
- A empresa cobra por serviços de treinamento e suporte na ferramenta, além de comercializar módulos de software adicionais.

3.3 Bancos de Dados

3.3.1 *Relacionais*

Oracle, MySQL, PostgreSQL

3.3.2 *Não-Relacionais*

(CATTELL, 2011)

MongoDB, CouchDB e Cassandra

4 SISTEMA DESENVOLVIDO

texto texto texto

4.1 Aquisição de Dados

oi

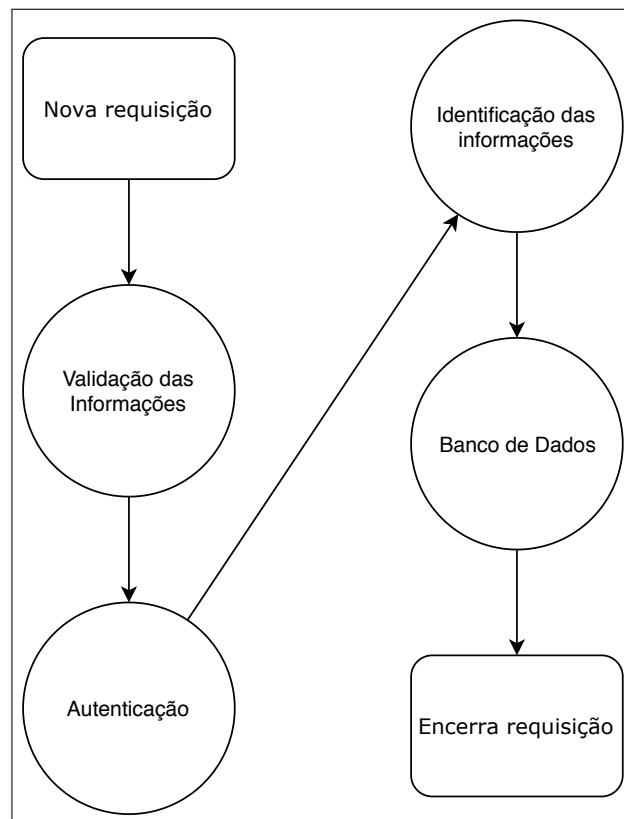
4.1.1 Métodos de Aquisição

oi

4.1.1.1 HTTP

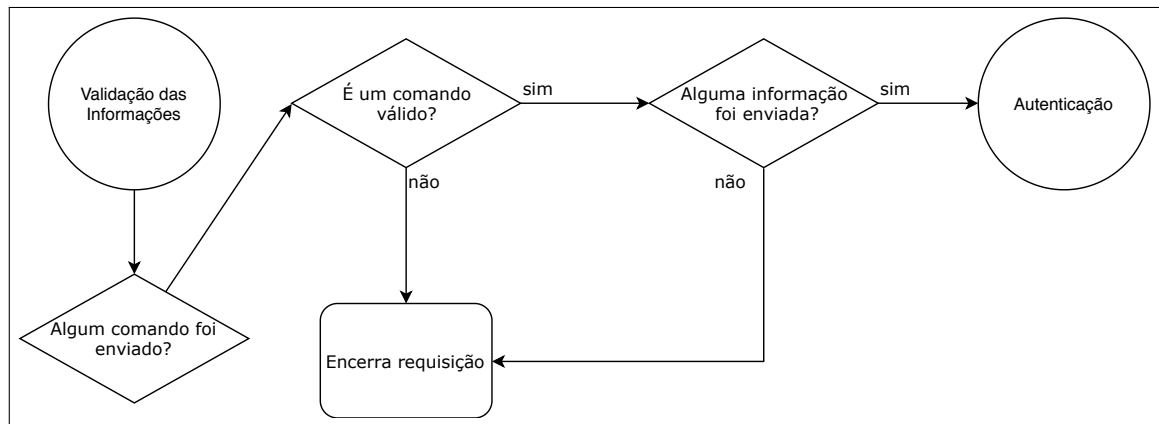
oi

Figura 20 – Diagrama das etapas do servidor para manipulação de dados.



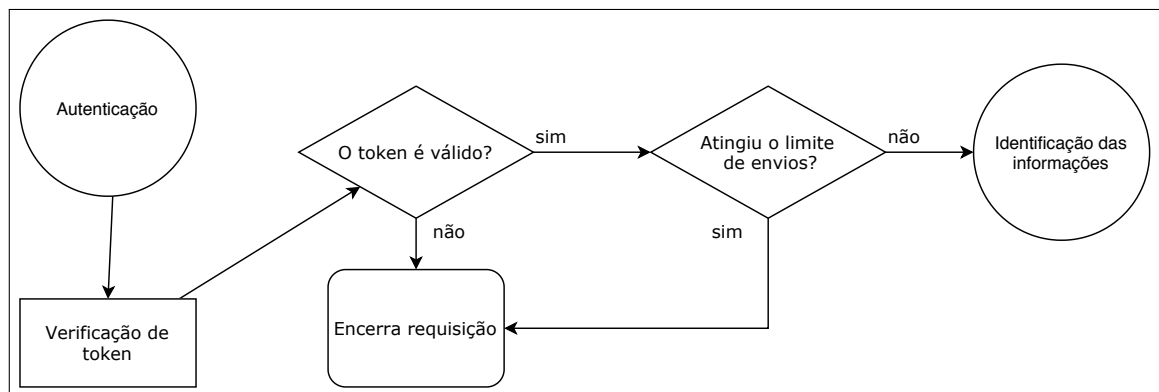
Fonte – O autor

Figura 21 – Diagrama de validação das informações recebidas.



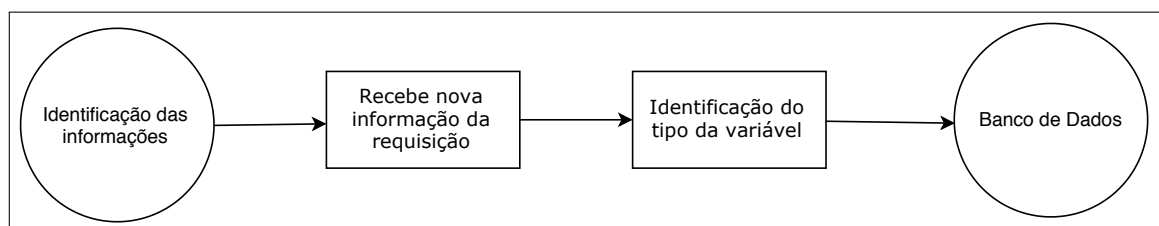
Fonte – O autor

Figura 22 – Diagrama de autenticação do usuário.



Fonte – O autor

Figura 23 – Diagrama sobre a identificação do tipo das informações.



Fonte – O autor

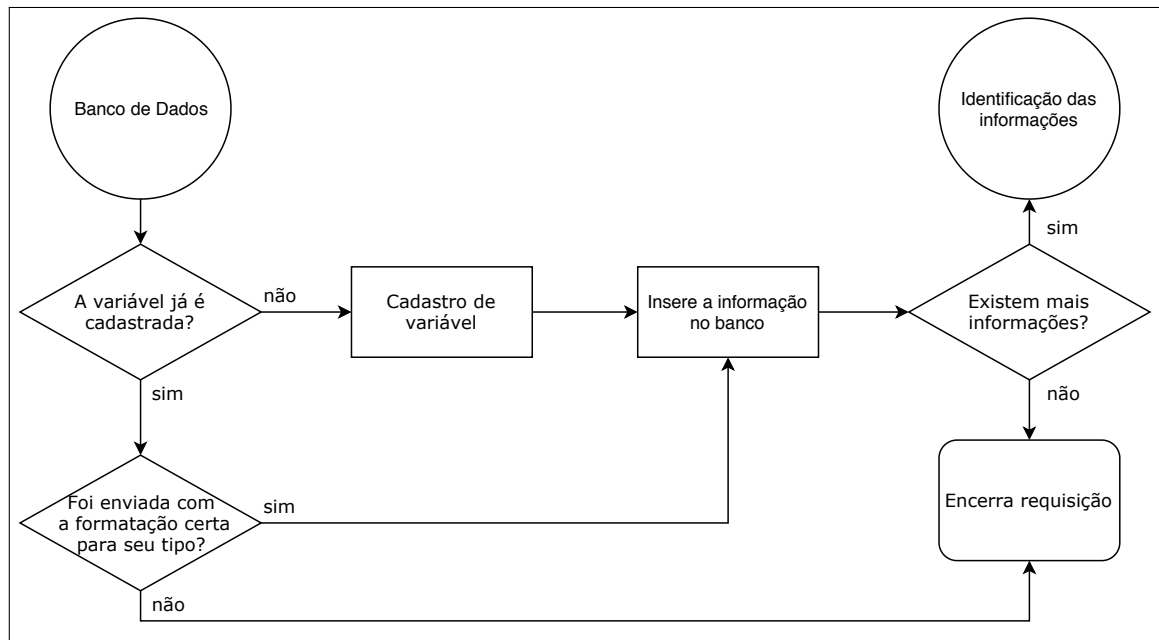
4.1.1.2 MQTT

oi

4.1.2 Armazenamento dos Dados

oi

Figura 24 – Banco de Dados.



Fonte – O autor

4.1.2.1 Banco de Dados

oi

4.1.2.2 Tipos de Variáveis

oi

4.1.2.3 Tempo

oi

4.2 Painel de Controle

oi

4.2.1 Segurança

oi

4.2.1.1 Proteções

oi

4.2.1.2 Criptografia

oi

4.2.1.3 Controle de Acesso

oi

4.2.2 Projetos

oi

4.2.2.1 Objetos

oi

4.2.2.2 Clientes Associados

oi

4.2.3 Clientes

oi

4.2.4 Domínios

oi

4.3 Recursos Computacionais

oi

4.3.1 Armazenamento

oi

4.3.2 Processamento

oi

4.4 K-NN

4.4.1 Classificação

oi

4.4.2 Teste

oi

4.5 Plataforma Estudantil

oi

4.6 Drivers de Comunicação

oi

5 INTERFACE WEB

oi oi oi oi oi

Figura 25 – Tela de autenticação da Interface Web.



A imagem mostra a tela de autenticação da Interface Web. No topo, há um ícone de três pessoas com engrenagens e o texto "rscada". Abaixo, a pergunta "Quais seus dados?". Em seguida, dois campos de entrada: "Usuário ou E-mail" e "Senha". Abaixo dos campos, um botão azul com o texto "Entrar no Sistema". Na base, dois links: "Criar uma conta" e "Esqueceu a senha?".

Fonte – O autor

oi oi

oi

oi oi

Figura 26 – Tela de cadastro para novos usuários.

rscada

Criar nova conta

Nome

E-mail

Usuário

Senha

Confirmar Senha

☐ Aceitos os termos e condições!

Criar nova conta

Já é registrado? [Entrar no Sistema.](#)

Fonte – O autor

Figura 27 – Tela inicial do sistema após autenticação.

rscada

MENU

- Minha Conta
- Meus Projetos**
- Meus Clientes
- Meus Domínios
- Minhas Faturas
- Meus Chamados
- Alarmes
- Outras Informações
- Sair do Sistema

Meus Projetos

Novo Projeto

Nome	Clientes	Opções
Qualidade Wi-Fi	2 cliente(s)	Gerenciar Excluir
Temperatura e Umidade	1 cliente(s)	Gerenciar Excluir
OBD II	1 cliente(s)	Gerenciar Excluir
Roteadores	1 cliente(s)	Gerenciar Excluir

Fonte – O autor

Figura 28 – Apresentação Geral de todos os projetos cadastrados.

Meus Projetos		
<div> <div></div> <div>Novo Projeto</div> </div>		
Nome	Clientes	Opções
Qualidade Wi-Fi	2 cliente(s)	<div>Gerenciar</div> <div>Excluir</div>
Temperatura e Umidade	1 cliente(s)	<div>Gerenciar</div> <div>Excluir</div>
OBD II	1 cliente(s)	<div>Gerenciar</div> <div>Excluir</div>
Roteadores	1 cliente(s)	<div>Gerenciar</div> <div>Excluir</div>

Fonte – O autor

Figura 29 – Página de cadastro de novos Projetos.

Meus Projetos

Novo Projeto

Voltar aos Projetos

Nome do Projeto:

Digite o nome do projeto (ex.: Controle de Nivel)

Salvar Alterações!

Fonte – O autor

Figura 30 – Página de cadastro de novas Variáveis.

Meus Projetos

+ Nova Variável

Voltar ao Projeto

Tipo de Variável:

Numérico

Variável:

Digite a variável desejada (ex.: vel)

Nome da Variável:

Digite o nome da variável (ex.: Velocidade)

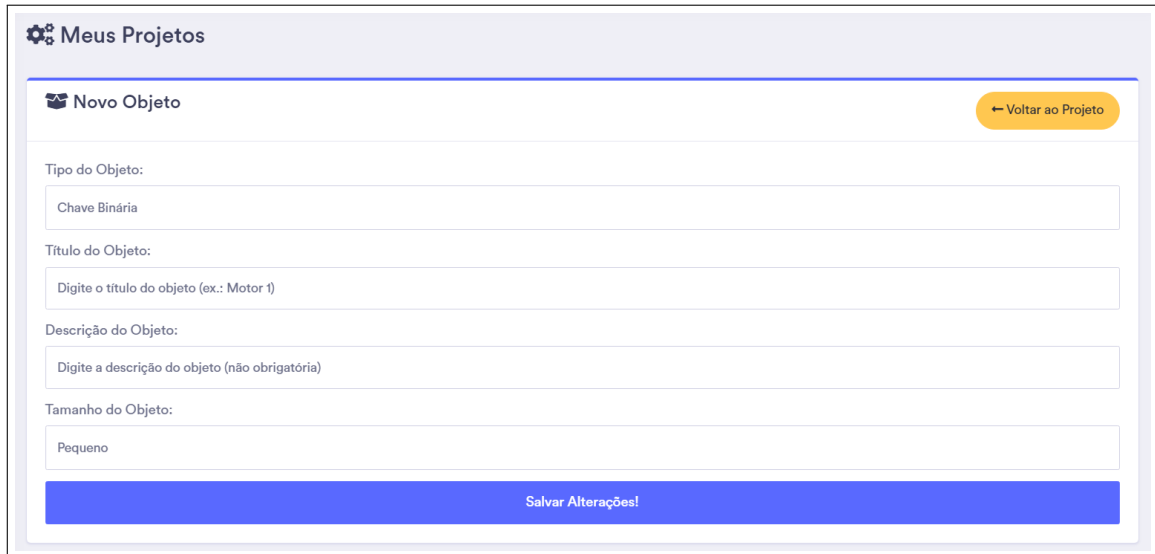
Unidade:

Digite a unidade (ex.: km/h) (não obrigatória)

Salvar Alterações!

Fonte – O autor

Figura 31 – Página de cadastro de novos Objetos.



Meus Projetos

Novo Objeto [← Voltar ao Projeto](#)

Tipo do Objeto:

Título do Objeto:

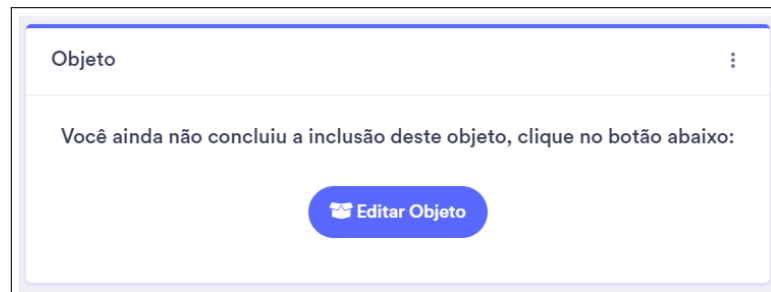
Descrição do Objeto:

Tamanho do Objeto:

[Salvar Alterações!](#)

Fonte – O autor

Figura 32 – Objeto recém-criado.



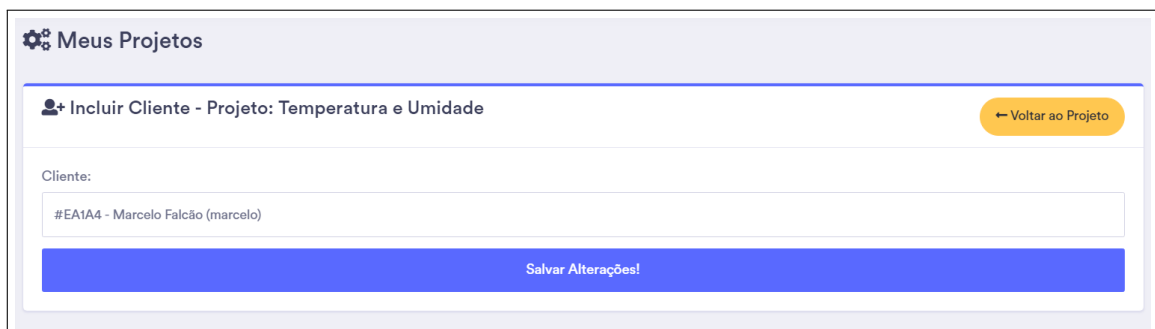
Objeto

Você ainda não concluiu a inclusão deste objeto, clique no botão abaixo:

[✎ Editar Objeto](#)

Fonte – O autor

Figura 33 – Associação de novo cliente ao projeto.



Meus Projetos

Incluir Cliente - Projeto: Temperatura e Umidade [← Voltar ao Projeto](#)

Cliente:

[Salvar Alterações!](#)

Fonte – O autor

Figura 34 – Visão geral dos clientes cadastrados no projeto.

Meus Projetos				
Clientes Incluídos - Projeto: Temperatura e Umidade				Incluir Cliente Voltar ao Projeto
Nome	Token	Nível	Registros	Opções
#EA1A4 - Marcelo Falcão	53027A	0	11772783	Editar Excluir

Fonte – O autor

6 RESULTADOS

6.1 Temperatura e Umidade

oi

Figura 35 – Variáveis utilizadas para o projeto.



The screenshot shows a web interface titled 'Meus Projetos'. Below the title is a section 'Gerenciar Variáveis - Projeto: Temperatura e Umidade'. There are two buttons: '+ Nova Variável' (green) and '← Voltar ao Projeto' (orange). Below these is a table with the following data:

Nome	# Variável	Tipo	Registros	Opções
Latência	latencia	Númerica	3922889	Gerenciar, Editar, Excluir
Temperatura	temperatura	Númerica	3926014	Gerenciar, Editar, Excluir
Umidade	umidade	Númerica	3929830	Gerenciar, Editar, Excluir

Fonte – O autor

oi

Figura 36 – Últimas informações enviadas.



The screenshot shows three cards representing the latest data sent. Each card displays a value, a percentage change, and the time 'Agora.' (Now).

Temperatura	Umidade	Latência
30,80 °C	73,00 %	127,00 ms
0,0%	0,0%	0,0%
Agora.	Agora.	Agora.

Fonte – O autor

6.2 Qualidade Sinal - WiFi

oi

oi

6.3 Demanda de Roteadores

oi

Figura 37 – Variáveis utilizadas no projeto.

Gerenciar Variáveis - Projeto: Qualidade Wi-Fi				
			+ Nova Variável	← Voltar ao Projeto
Nome	# Variável	Tipo	Registros	Opções
Latência	latencia	Númerica	4016223	Gerenciar Editar Excluir
Monitoramento	monitoramento	Binária (booleano)	0	Gerenciar Editar Excluir
Nível de Sinal	sinal	Númerica	4014618	Gerenciar Editar Excluir

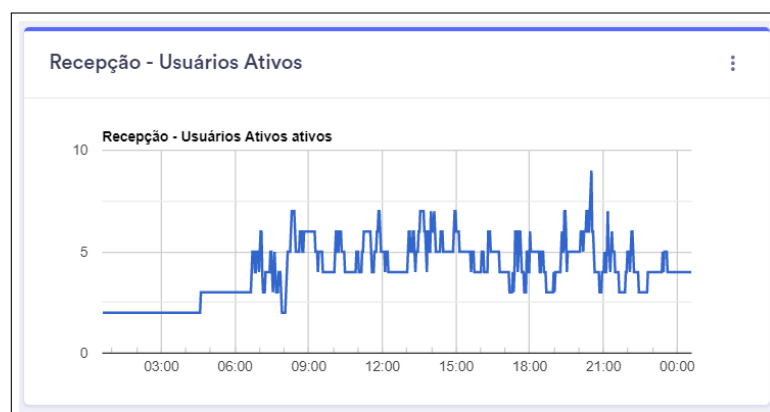
Fonte – O autor

Figura 38 – Botão para ação no processo além de últimas informações enviadas.



Fonte – O autor

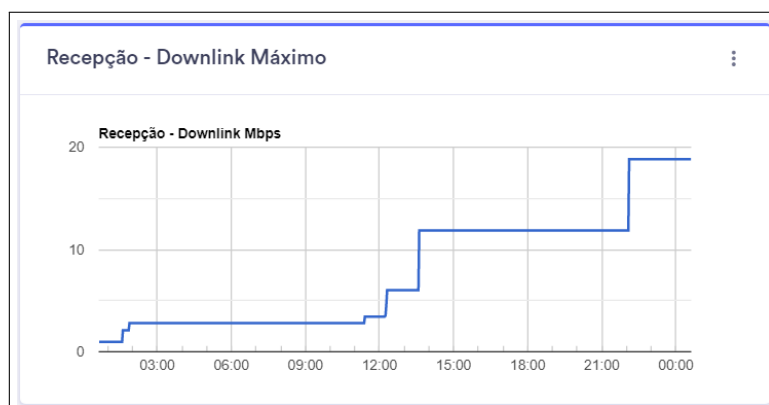
Figura 39 – Quantidade de hóspedes conectados no roteador da recepção durante 24 horas.



Fonte – O autor

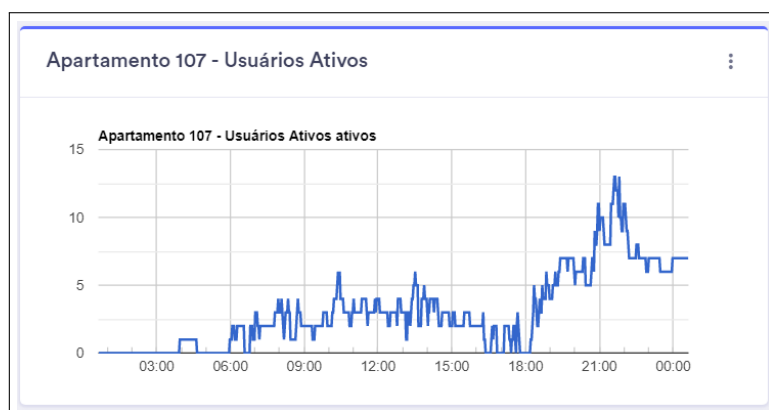
oi
oi
oi

Figura 40 – Uso máximo de downlink pelo roteador da recepção durante 24 horas.



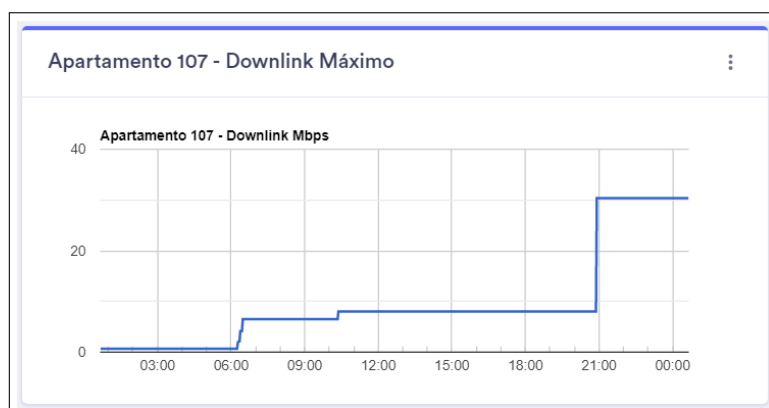
Fonte – O autor

Figura 41 – Quantidade de hóspedes conectados no roteador "107" durante 24 horas.



Fonte – O autor

Figura 42 – Uso máximo de downlink pelo roteador "107" durante 24 horas.



Fonte – O autor

7 CONCLUSÕES E TRABALHOS FUTUROS

Parte final do texto na qual se apresentam as conclusões apoiadas no desenvolvimento do assunto. É a recapitulação sintética dos resultados obtidos. Pode apresentar recomendações e sugestões para pesquisas futuras.

REFERÊNCIAS

- BRANQS. **ihm touch screen**. 2019. Disponível em: <<https://www.branqs.com.br/ihm-touch-screen>>. Acesso em: 25 fev. 2019.
- CASSANDRA AMARAL, N. F. V. K. O. M.; DANIELLE, C. S. S. Automação como ferramenta de análise de eficiência energética. **Anais do XX Congresso Brasileiro de Automática**, Belo Horizonte, MG, p. 699–706, 2014.
- CATTELL, R. Scalable sql and nosql data stores. **Acm Sigmod Record**, ACM, v. 39, n. 4, p. 12–27, 2011.
- CONTROLS, T. **TANGO Controls**. 2019. Disponível em: <<https://www.tango-controls.org/>>. Acesso em: 28 mar. 2019.
- DANEELS, A.; SALTER, W. What is scada? **International Conference on Accelerator and Large Experimental Physics Control Systems**, Trieste, Italy, 1999.
- ELIPSE. **Elipse E3: uma visão geral**. 2019. Disponível em: <<http://kb.elipse.com.br/pt-br/questions/40/Elipse+E3%3A+uma+vis%C3%A3o+geral>>. Acesso em: 28 mar. 2019.
- FILHO, J. M. **Instalações Elétricas Industriais**. [S.l.]: Addison-Wesley Reading, Massachusetts, 2017. (9).
- FOUNDATION, O. **What is OPC?** 2019. Disponível em: <<https://opcfoundation.org/about/what-is-opc/>>. Acesso em: 28 mar. 2019.
- FOUNDATION, O. **What is OPC Classic?** 2019. Disponível em: <<https://opcfoundation.org/faq/what-is-opc-classic/>>. Acesso em: 28 mar. 2019.
- HELME, S. **Alexa Top 1 Million Analysis - February 2019**. 2019. Disponível em: <<https://scotthelme.co.uk/alexa-top-1-million-analysis-february-2019/>>. Acesso em: 28 mar. 2019.
- INDUSOFT. **InduSoft Web Studio HMI SCADA Development Software**. 2019. Disponível em: <<http://www.indusoft.com/>>. Acesso em: 28 mar. 2019.
- JSON. **Introdução ao JSON**. 2019. Disponível em: <<http://json.org/json-pt.html>>. Acesso em: 28 mar. 2019.
- KARNOUSKOS, S.; COLOMBO, A. W. Architecting the next generation of service-based scada/dcs system of systems. **37th Annual Conference of the IEEE Industrial Electronics Society**, Melbourne, VIC, Australia, p. 359–364, 2011.
- LEE SEUNG-WOO; NAM, S.-J. L. J.-K. A real-time equipment interface for controlling production equipment. **12th International Conference on Control, Automation and Systems**, ICC, Jeju Island, Korea, p. 1173–1177, 2012.
- LIPNICKAS ARUNAS; RUTKAUSKAS, R. C.-R. Interoperability of scada system applications with web services. **IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications**, Rende (Cosenza), Italy, p. 196–200, 2009.
- MQTT. 2019. Disponível em: <<http://www.mqtt.org>>. Acesso em: 25 fev. 2019.

ORGANIZATION, M. **Modbus FAQ**. 2019. Disponível em: <<http://www.modbus.org/faq.php>>. Acesso em: 28 mar. 2019.

OSMIC NEDIM; VELAGI, J. Design of a simple service oriented supervisory control and data acquisition system. **59th International Symposium ELMAR**, Zadar, Croatia, p. 245–248, 2017.

SCADABR. **ScadaBR**. 2019. Disponível em: <<http://www.scadabr.com.br/>>. Acesso em: 28 mar. 2019.

SOCIETY, T. I. **Hypertext Transfer Protocol – HTTP/1.1**. 1999. Disponível em: <<https://tools.ietf.org/html/rfc2616>>. Acesso em: 28 mar. 2019.

SOCIETY, T. I. **The Secure HyperText Transfer Protocol**. 1999. Disponível em: <<https://tools.ietf.org/html/rfc2660>>. Acesso em: 28 mar. 2019.

SOCIETY, T. I. **XML Media Types**. 2001. Disponível em: <<https://tools.ietf.org/html/rfc3023#section-3.2>>. Acesso em: 28 mar. 2019.

SOFTWARE, R. **Rapid SCADA**. 2019. Disponível em: <<https://rapidscada.org/>>. Acesso em: 28 mar. 2019.

W3C. **Web Services Architecture**. 2004. Disponível em: <<https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>. Acesso em: 12 mar. 2019.

WAGO. **Um plugin torna os controladores WAGO prontos para IoT**. 2019. Disponível em: <<https://www.wago.com/br/controlador-clp-iot-com-mqtt>>. Acesso em: 25 fev. 2019.

WEG. **CANopen - RUW03**. 2019. Disponível em: <<https://static.weg.net/medias/downloadcenter/ha2/hb7/WEG-ruw03-manual-da-unidade-remota-canopen-10003264976-manual-portugues-br.pdf>>. Acesso em: 28 mar. 2019.

WEG. **Controlador Lógico Programável PLC300**. 2019. Disponível em: <https://www.weg.net/catalog/weg/BR/pt/Automa%C3%A7%C3%A3o-e-Controle-Industrial/Controle-de-Processos/Controladores-L%C3%B3gicos-Program%C3%A1veis/Controlador-L%C3%B3gico-Program%C3%A1vel-PLC300/Controlador-L%C3%B3gico-Program%C3%A1vel-PLC300/p/MKT_WDC_BRAZIL_PROGRAMMABLE_LOGIC_CONTROLLER_PLC_PLC300>. Acesso em: 28 mar. 2019.

APÊNDICE A – TÍTULO

oi

ANEXO A – EXEMPLO DE ANEXO

oi