

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE E INOVAÇÃO

MATHEUS DOS SANTOS VASCONCELLOS

**ANÁLISE E COLETA DE DADOS
UTILIZANDO INTERNET DAS COISAS
PARA INTEGRAÇÃO DE APLICAÇÕES
DISTRIBUÍDAS**

Monografia de Conclusão de Curso apresentada
como requisito parcial para a obtenção do grau
de Especialista em Engenharia de Software e
Inovação

Orientador: Profa. Dra. Renata de Matos Galante

Porto Alegre
2021

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

dos Santos Vasconcellos, Matheus

ANÁLISE E COLETA DE DADOS UTILIZANDO INTERNET DAS COISAS PARA INTEGRAÇÃO DE APLICAÇÕES DISTRIBUÍDAS / Matheus dos Santos Vasconcellos. – Porto Alegre: PPGC da UFRGS, 2021.

74 f.: il.

Monografia (especialização) – Universidade Federal do Rio Grande do Sul. Curso de Especialização em Engenharia de Software e Inovação, Porto Alegre, BR-RS, 2021. Orientador: Renata de Matos Galante.

1. Indústria 4.0. 2. Internet das Coisas. 3. Dispositivos ciberfísicos. 4. Broker. 5. Publish/Subscribe. 6. Internet. I. de Matos Galante, Renata. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitor de Pós-Graduação: Prof. Júlio Otávio Jardim Barcellos

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenadora do Curso: Prof^a. Karin Becker

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“ Deus nos conceda, a cada dia, uma página de vida nova no livro do tempo.

Aquilo que colocarmos nela, corre por nossa conta.”

— XICO XAVIER

AGRADECIMENTOS

A Deus por permitir que os estudos façam parte da minha vida, complementando, da melhor forma possível, a minha jornada pessoal, acadêmica e profissional.

A minha família, especialmente os meus queridos irmãos por sempre acreditarem no meu potencial me dando forças para seguir em frente, mesmo diante das dificuldades que fomos submetidos, a minha amada mãe por todo carinho e apoio nas horas mais difíceis da minha vida e, ao meu maravilhoso pai, uma estrela que brilha lá no céu e que sempre guia o meu caminho em direção do bem e na conquista dos meus objetivos.

A família que a vida me trouxe, especialmente a Stephanie Henz Marques, minha namorada e parceira que Deus colocou no meu caminho e a Ivone Henz Marques, minha querida sogra e segunda mãe, pelos momentos de ausência da minha parte em meio aos estudos e pesquisas realizadas.

A minha orientadora Profa. Dra. Renata de Matos Galante, pelas valiosas reuniões e pela confiança em mim depositada para seguir com a pesquisa sobre um tema tão relevante atualmente.

Ao Grupo Tecnova Energia, empresa a qual faço parte atuando junto ao núcleo de Sistemas de Proteção, Controle e Supervisão (SPCS), especialmente ao Diretor e Presidente Eng. Carlos Frederico Behrends por acreditar no meu potencial, permitindo que eu me desenvolva como profissional agregando conhecimento em tecnologia aos projetos.

RESUMO

Com a evolução e o aumento na conectividade entre aplicações de *softwares*, *hardwares* e dispositivos inteligentes motivados pelo contexto da transformação digital, é importante que as organizações, incluindo fornecedores e consumidores, percebam as mudanças que o avanço tecnológico proporciona. Isto torna possível a estruturação de modelos baseados em tecnologias como a Indústria 4.0, Internet das Coisas e o desenvolvimento de aplicações de baixo custo através de ferramentas de código aberto juntamente com protocolos de comunicação para redes com baixa largura de banda e alta latência, solucionando problemas relacionados à falta de monitoramento de ambientes descentralizados, como é o caso das plantas fotovoltaicas, utilizadas como fonte de energia renovável para geração distribuída são geralmente executadas em localidades onde o acesso à internet é inseguro ou limitado. Contudo, apesar das soluções de tecnologia de hardwares constituídas por dispositivos ciber-físicos oferecidas pelos grandes *players* do setor industrial, como: *Siemens*, *Schneider*, *General Electric*, entre outros, consolidadas no setor, é necessário um alto investimento inicial. Neste trabalho, é proposto um modelo de integração de dados que integra sistemas de monitoramento presentes em domínios distribuídos a sistemas colaborativos centralizados, permitindo o gerenciamento de informações sobre o processo de geração de energia solar. Por isto, o modelo de integração de dados proposto é baseado nos conceitos de Internet das Coisas e uma arquitetura de comunicação conforme o padrão *Publish/Subscribe*, utilizando dispositivos ciber-físicos como *Gateway* publicando mensagens através da internet no *Broker*, permitindo a telemetria dos dados gerados por plantas fotovoltaicas a partir de um sistema colaborativo de código aberto. Para avaliar o modelo proposto, foi feito um teste de performance do servidor *Mosquitto Broker* utilizando a ferramenta *Apache JMeter*. Os resultados indicaram que é possível conectar grupos de até 1000 conexões simulando os domínios distribuídos usando o modelo de integração de dados utilizando o software *Mosquitto* como servidor e *Broker* para o gerenciamento das mensagens, sendo que os planos de teste com até 100 clientes assinantes apresentaram resultados totalmente satisfatório e parcialmente para grupos de 1000 assinantes inscritos no mesmo tópico de publicação de dados.

Palavras-chave: Indústria 4.0. Internet das Coisas. Dispositivos ciber-físicos. Broker. Publish/Subscribe. Internet.

ABSTRACT

With the evolution and increase in connectivity between software applications, hardware and intelligent devices motivated by the context of digital transformation, it is important that organizations, including suppliers and consumers, realize the changes that technological advance offers. This makes possible the structuring of models based on technologies such as Industry 4.0, Internet of Things and developing low cost applications through open source tools with communication connections for networks with low bandwidth and high latency, solving fault related problems monitoring of decentralized environments, such as photovoltaic plants, used as a source of renewable energy for distributed generation, are generally performed in locations where internet access is insecure or limited. However, despite the hardware technology solutions constituted by cyber-expanded devices by the big players in the industrial sector, such as: Siemens, Schneider, General Electric, among others, consolidated in the sector, a high initial investment is needed. In this work, a data integration model is proposed that integrates monitoring systems present in distributed domains to centralized collaborative systems, allowing the management of information about the solar energy generation process. Therefore, the proposed data integration model is based on the concepts of Internet of Things and a communication architecture conforming to or standard Publish / Subscribe, using cyber-physical devices such as Gateway publishing messages through the internet on the Broker, allowing telemetry of the data generated by photovoltaic plants from the open source collaborative system. To evaluate the proposed model, a performance test of the Mosquitto Broker server was performed using the Apache JMeter tool. The results indicated that it is possible to connect groups of up to 1000 connected, simulating the domains using the data integration model using the Mosquitto software as a server and Broker for message management, and the test plans with up to 100 subscriber clients dissipated fully satisfactory and partially for groups of 1000 subscribers subscribed to the same data publishing topic.

Keywords: Industry 4.0, Internet of Things, Ciber Physical Device, Broker.

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ADS	Autonomous Decentralized System Autonomous
CPS	Cyber-Physical Systems
CPU	Central Processing Unit Correspondence
SMP	Symmetric Multi-Processor
NUMA	Non-Uniform Memory Access
SIMD	Single Instruction Multiple Data
SPMD	Single Program Multiple Data
OT	Operational Technology
OASIS	Organization for the Advancement of Structured Information Standards
IEC	International Electrotechnical Commission
LAN	Local Area Network
IT	Information Technology
IP	Internet Protocol
IoT	Internet of Things
ISO	International Organization for Standardization
MQTT	Message Queue Telemetry Transport
QoS	Quality of Service
TCP	Transmission Control Protocol
TLS	Transport Layer Security
SCADA	Supervisory Control and Data Acquisition
SSL	Secure Sockets Layer
UML	Unified Modeling Language

LISTA DE FIGURAS

Figura 2.1	Significado de Coisa no contexto do padrão IEEE Std 2413™-2019	16
Figura 2.2	Evolução IoT	17
Figura 2.3	Aplicações IoT.....	19
Figura 2.4	Fases da Revolução Industrial	20
Figura 2.5	Princípios da Indústria 4.0	22
Figura 2.6	Pilares da Indústria 4.0	23
Figura 2.7	Arquitetura ADS.....	24
Figura 2.8	Arquitetura ADS.....	24
Figura 2.9	Modelo Conceitual de Integração	25
Figura 2.10	3C do CPS	25
Figura 2.11	Arquitetura Sistemas Ciber-físicos	26
Figura 2.12	Overview do Protocolo de Dados	26
Figura 2.13	Estrutura de Controle de Pacote	28
Figura 2.14	Formato do Cabeçalho Fixo.....	28
Figura 2.15	Conexão de Rede MQTT.....	30
Figura 2.16	Plataformas de Instalação	32
Figura 2.17	Nós de função	33
Figura 4.1	Arquitetura do Modelo de Integração.....	39
Figura 4.2	Sistema Colaborativo entre Domínios	39
Figura 4.3	Arquitetura <i>Publish/Subscribe</i>	41
Figura 4.4	Modelo Conceitual Domínio Local	42
Figura 4.5	Diagrama de Classe do Sistema Autônomo Descentralizado.....	43
Figura 4.6	Modelo de Estrutura de Sistemas	45
Figura 5.1	Ambiente de Execução	47
Figura 5.2	Estrutura de Programas.....	49
Figura 5.3	Diretório de Bibliotecas Plataforma de Automação	49
Figura 5.4	Declaração de Variáveis Programa de Publicação de Dados.....	50
Figura 5.5	Estrutura da Classe das Coleções	51
Figura 5.6	Inicialização do Broker.....	52
Figura 5.7	Configuração de Conexão.....	52
Figura 5.8	Assinatura do Cliente MQTT	53
Figura 5.9	Publicação de Dados MQTT	53
Figura 5.10	Interação entre Domínios.....	54
Figura 5.11	Inicialização Node-RED.....	54
Figura 5.12	Fluxo Básico Sistema Colaborativo.....	54
Figura 5.13	Nó de Entrada para Conexões MQTT	55
Figura 5.14	Configuração do Nó de Entrada MQTT	55
Figura 5.15	Nó Parser JSON - Objetos	56
Figura 5.16	Configuração do Nó de Entrada MQTT	56
Figura 5.17	Payload Antes e Depois do Parser.....	57
Figura 5.18	Nó de Função.....	57
Figura 5.19	Nó de Função.....	57
Figura 5.20	Conjunto de Nós UI.....	58
Figura 5.21	Template de UI	59
Figura 5.22	Interface Web Browser Plataforma de Colaboração	59
Figura 5.23	Interface Web Mobile Plataforma de Colaboração.....	60

Figura 5.24 Fluxo de Programação Básico	60
Figura 5.25 Fluxo de Avaliação do Modelo de Integração	61
Figura 5.26 Configuração MQTT Connect	61
Figura 5.27 Configuração MQTT Sub	62
Figura 5.28 Propriedades Grupo de Usuário Plano de Teste 1	63
Figura 5.29 Propriedades Grupo de Usuário Plano de Teste 2	64
Figura 5.30 Propriedades Grupo de Usuário Plano de Teste 3	64
Figura 5.31 Propriedades Grupo de Usuário Plano de Teste 4	65
Figura 5.32 Gráfico Plano de Teste 1	66
Figura 5.33 Gráfico Plano de Teste 2	67
Figura 5.34 Gráfico Plano de Teste 3	68
Figura 5.35 Gráfico Plano de Teste 4	69

LISTA DE TABELAS

Tabela 2.1 Principais Conceitos do MQTT.....	27
Tabela 2.2 Estrutura de Controle de Pacotes MQTT	28
Tabela 2.3 Identificação de Bits	29
Tabela 2.4 Definições de QoS	30
Tabela 5.1 Variáveis Simuladas.....	58
Tabela 5.2 Metodologia GQM ao modelo de integração proposto:	62
Tabela 5.3 Resumo Desempenho Servidor Broker - Plano de Teste 1.....	66
Tabela 5.4 Resumo Desempenho Servidor Broker - Plano de Teste 2.....	67
Tabela 5.5 Resumo Desempenho Servidor Broker - Plano de Teste 3.....	68
Tabela 5.6 Resumo Desempenho Servidor Broker - Plano de Teste 4.....	69
Tabela 5.7 Resumo dos Resultados sobre o Desempenho do Servidor Mosquitto Broker	69

SUMÁRIO

1 INTRODUÇÃO	13
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Internet das Coisas	16
2.2 Indústria 4.0	20
2.3 Conceitos e Tecnologias Utilizadas	24
2.3.1 Sistemas Autônomos Descentralizados	24
2.3.2 Sistemas Ciber-físicos.....	25
2.3.3 Protocolo MQTT.....	27
2.3.3.1 Principais Conceitos.....	27
2.3.3.2 Estrutura do Protocolo MQTT	28
2.3.3.3 Conexão de Rede	29
2.3.3.4 Qualidade de Serviço	30
2.3.3.5 Segurança.....	31
2.3.4 Broker Mosquitto	31
2.3.5 Node-RED.....	32
2.3.5.1 Características	32
2.3.5.2 Funcionalidades	32
3 TRABALHOS RELACIONADOS	34
3.1 Arquitetura ATIoT: Integrando Tecnologias Assistivas com Internet das Coisas	34
3.1.1 Fundamentação	34
3.1.2 Vantagens e Desvantagens	35
3.1.3 Críticas e Diferenças	35
3.2 Uma Abordagem para a Integração de Sistemas Industriais Aplicando o Conceito de Internet das Coisas e de Modelos Semânticos no Contexto da Indústria 4.0	36
3.2.1 Fundamentação	36
3.2.2 Vantagens e Desvantagens	36
3.2.3 Críticas e Diferenças	37
3.2.4 Comparação entre Modelos de Integração Relacionados e Proposto	37
4 PROPOSTA DE UM MODELO DE INTEGRAÇÃO PARA ANÁLISE E COLETA DE DADOS	38
4.1 Planejamento	38
4.2 Visão Geral	38
4.3 Características sobre o Domínio da Aplicação	39
4.3.1 Domínio IoT.....	40
4.3.2 Domínio Distribuído	40
4.3.3 Relacionamento entre Domínios.....	40
4.4 Modelagem da Aplicação	41
4.4.1 Modelo Conceitual da Planta Física	41
4.4.2 Modelo de Domínio Local	42
4.4.3 Modelo de Sistema Autônomo Descentralizado.....	44
4.4.4 Modelo de Dispositivo Ciber-físico	44
5 IMPLEMENTAÇÃO E AVALIAÇÃO DA APLICAÇÃO	46
5.1 Execução do Modelo	46
5.1.1 Hardwares	47
5.1.2 Softwares.....	48
5.1.3 Plataforma de Automação.....	48

5.1.4	Broker MQTT	51
5.1.5	Sistema Colaborativo	53
5.1.5.1	Configurações Iniciais.....	54
5.1.5.2	Fluxo de Programação	56
5.1.5.3	Interface de Visualização	58
5.1.6	Plano de Testes e Análise de Resultados	61
5.1.6.1	Plano de Teste 1	63
5.1.6.2	Plano de Teste 2	63
5.1.6.3	Plano de Teste 3	64
5.1.6.4	Plano de Teste 4	65
5.1.6.5	Análise dos Resultados	65
5.1.6.6	Limitações.....	70
6	CONCLUSÃO	71
	REFERÊNCIAS.....	73

1 INTRODUÇÃO

O contexto da pesquisa realizada neste trabalho aborda sobre os conceitos e os fundamentos teóricos que envolvem as etapas de estudo, mapeamento, desenvolvimento e implementação de modelos de baixo custo para integração de dados baseados em tecnologias digitais, incorporando hardwares dedicados e softwares de código aberto, que permitem a convergência e a interoperabilidade de elementos provocando adaptações e possíveis melhorias na cadeia de valor das organizações a partir da análise orientada a dados (MEHMOOD et al., 2019). Desta forma, o relacionamento entre os ambientes físicos e digitais se torna viável pelo uso de *gateways* como, por exemplo, sistemas ciber-físicos que incorporam recursos avançados em tecnologias de operação, informação e comunicação interagindo com sistemas autônomos descentralizados, também classificados como entidades com elementos capazes de realizar a leitura de variáveis de processo em tempo real, assim como executar controles a distância entre domínios utilizando o protocolo de comunicação MQTT executado sobre TCP/IP pela infraestrutura de internet com arquitetura *Publish/Subscribe* no padrão cliente-servidor impactando de forma positiva nos tradicionais modelos de supervisão e controle de processos operacionais (OASIS, 2019).

A motivação deste trabalho é baseada na constante necessidade de integração entre dispositivos à sistemas distribuídos, normalmente utilizados como meio de interface entre os processos operacionais e as plataformas de gerenciamento de informações colaborando entre si, sejam elas instaladas em infraestruturas locais (*on-premises*) ou em ambientes virtuais hospedados na nuvem (*cloud computing*). Essa necessidade exige uma nova abordagem para especificação e para o desenvolvimento de soluções tecnológicas que impulsionam a inovação dos setores emergentes. Assim, praticando tecnologias que suportem a digitalização dos dados através de ferramentas e plataformas de código aberto, como é o caso do conceito da Internet das Coisas (*IoT*), *Big Data* e Indústria 4.0. Já que os sistemas integrados que são oferecidos pelos principais fornecedores como, por exemplo, AWS, Google, IBM, Microsoft, entre outros, exigem um investimento inicial com custos elevados para implantação e manutenção de modelos de integração, pois exigem uma dependência que poderia dificultar na escalabilidade e na evolução das aplicações de gerenciamento de dados na medida que o número de elementos conectados amplie, exigindo assim uma taxa de transmissão de dados proporcional.

Nesse contexto, a dificuldade de integração entre Tecnologia de Informação (IT) e Tecnologia de Operação (OT) continua sendo um paradigma presente nos mais vari-

ados setores da economia impactando na convergência e na interoperabilidade entre os equipamentos e sistemas, sendo um grande desafio para adoção de novas práticas e pela busca de novos modelos de negócio. De acordo com (LI et al., 2020), um dos principais afetados é o setor industrial devido a falta de acoplamento entre os protocolos industriais utilizados pelos fabricantes de dispositivos presentes na indústria com sistemas de gerenciamento utilizados pelas áreas de tecnologia de informação das organizações. Outra grande influência, ocorre devido à dificuldade na tomada de decisões estratégicas por parte de líderes e gestores que deixam de aprovar planos de investimento em setores de pesquisa e desenvolvimento impactando diretamente na inovação (GARIMELLA, 2018).

Alguns trabalhos da literatura contribuem para o estudo apresentando soluções e ambientes baseados em modelos de arquiteturas IoT para viabilizar a interoperabilidade e a convergência entre os agentes de dados do sistema. Segundo (BLAZY et al., 2021) o modelo de comunicação *Publish/Subscribe* permite a comunicação entre os agentes de dados no sistema com a implementação do protocolo MQTT (*Message Queue Telemetry Transport*). Ainda, conforme (MEHMOOD et al., 2019) os dispositivos de baixo custo possuem a capacidade de processar os dados centralizando as informações em servidores em nuvem.

O modelo de abordagem de desenvolvimento adotado neste trabalho segue critérios e técnicas que estão presentes na literatura, porém, contribuindo para agilidade na verificação e validação de projetos que integrem aplicações IoT, devido a sua facilidade de implementação na criação de ambientes e a possibilidade de instalação em sistemas operacionais tradicionais, com capacidade de instalação em ambientes locais ou embarcados (Windows, Linux e Raspberry Pi), quando comparado a outras tecnologias, como é o caso do *middleware FIWARE* que requer um sistema operacional dedicado no modelo proposto por Steinmetz (2018). Além disso, a modelagem de uma arquitetura por meio de um fluxo orientado a eventos torna intuitiva a interpretação sobre o funcionamento da aplicação a ser desenvolvida.

O objetivo principal deste trabalho é especificar um modelo de integração de dados orientado a eventos, utilizando uma abordagem baseada em conceitos de Internet das Coisas (IoT). O modelo permitirá a coleta de dados de um conjunto de elementos interconectados a um sistema distribuído. Por meio de uma arquitetura orientada a eventos, dispositivos em rede poderão publicar e/ou subscrever em tópicos de interesse, através de um servidor intermediário (*Broker*) transmitindo dados via protocolo MQTT (*Message Queue Telemetry Transport*), possibilitando a centralização das informações em um

único domínio. Por fim, o modelo busca contribuir para o gerenciamento de dados a partir de uma integração entre dispositivos e sistemas utilizando ferramentas de código aberto para domínios distribuídos, beneficiando empresas que buscam um baixo investimento inicial para inovação de seus métodos de controle e supervisão de dados.

Os objetivos específicos deste trabalho são:

- Criar um modelo de arquitetura orientada a eventos com base na literatura que viabilize o gerenciamento de dados entre dispositivos conectados à internet;
- Criar um fluxo que sirva como gateway de dados permitindo uma integração entre aplicações de diferentes domínios;
- Propor uma interface de usuário para interpretação e gerenciamento dos dados coletados.

O restante deste trabalho proposta está organizado da seguinte forma: o segundo capítulo apresenta a Fundamentação Teórica utilizada para embasamento sobre o tema da pesquisa. O terceiro capítulo contempla os Trabalhos Relacionados com o tema da pesquisa, contemplando, de forma resumida, fundamentos, vantagens, desvantagens e críticas aos modelos. O quarto capítulo descreve a Proposta de um Modelo de Integração para Análise e Coleta de Dados que inclui as atividades de mapeamento, estruturação de diagramas e modelos conceituais sobre os domínios da aplicação. Em seguida, o quinto capítulo aborda a Implementação e Avaliação da Aplicação incluindo os métodos, técnicas, ferramentas utilizadas, procedimentos realizados, assim como uma avaliação dos resultados obtidos após a realização de testes para verificação e validação do modelo. Ao final, o sexto capítulo do trabalho é relacionado a Conclusão, onde ressalta a contribuição na importância do modelo e discutindo as implicações e apresentando possibilidades futuras de trabalhos e temas relevantes para inovação e aperfeiçoamento do modelo sugerido.

2 FUNDAMENTAÇÃO TEÓRICA

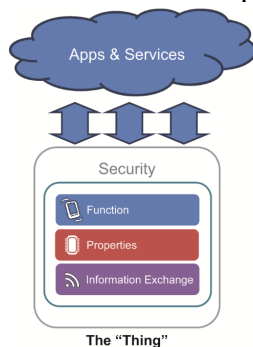
Este capítulo aborda a fundamentação teórica da pesquisa contemplando o embasamento teórico de sustentação do trabalho sobre os principais temas da pesquisa. O conteúdo está distribuído em tópicos que estão relacionados com os conceitos e as tecnologias aplicadas ao modelo de integração proposto para um ambiente de monitoramento e supervisão de dados. Isto inclui seções que contextualizam sobre Internet das Coisas, Indústria 4.0, Sistemas Autônomos Descentralizados (ADS), Sistemas Ciber-físicos (CPS), Protocolo MQTT, *Broker* Mosquitto e Node-RED.

2.1 Internet das Coisas

Uma das principais abordagens do modelo de integração é o conceito de IoT (em inglês, *Internet of Things*), que pode ser traduzido como Internet das Coisas. Hoje, considerada um dos principais pilares da Indústria 4.0, o conceito de IoT trata-se de uma tecnologia que permite a integração entre dispositivos ou objetos, que podem ser interpretados como “coisas” utilizando a internet como camada de rede e meio de transmissão de dados compartilhando informação entre sistemas.

De acordo com (SA et al., 2019), a definição sobre IoT baseia-se em um sistema formado por entidades¹ incluindo sistemas ciber-físicos com recursos avançados para troca de informação que interagem com o mundo físico, assim, processando dados através da leitura de sensores e atuação em sistemas físicos. E, por este motivo, a importância em utilizar os conceitos de IoT como base do modelo de integração proposto neste trabalho.

Figura 2.1: Significado de Coisa no contexto do padrão IEEE Std 2413™-2019



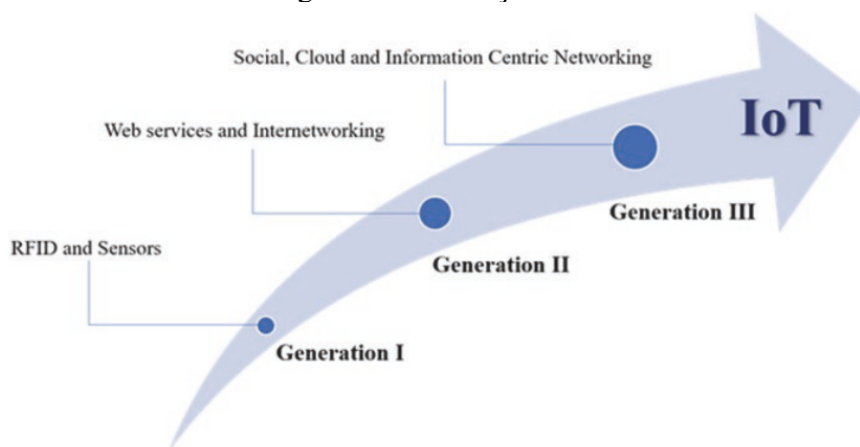
Fonte: (SA et al., 2019)

¹Entidade pode ser definida como algo em particular, incluindo: pessoas; lugares; processos; objetos; concepções; associação ou evento.

Segundo (STEINMETZ, 2018), o termo IoT foi apresentado pela primeira vez em 1999 utilizado na cadeia de suprimentos quando ganhou maior destaque pela utilização combinada com a tecnologia de rádio frequência para identificação de objetos surgindo, então, o RFID (em inglês, *Radio Frequency Identification*). Contudo, (LEA, 2018) complementa que a origem histórica da tecnologia RFID foi a partir de um projeto britânico para detecção de aeronaves de caça durante a Segunda Guerra Mundial.

Desde então, a tecnologia IoT vive em constante evolução devido ao número de dispositivos conectados em larga escala à internet, assim como sua implementação, o que contribui para a interoperabilidade entre sistemas e aplicações distribuídas. Além de integrar informações do mundo físico ao virtual, por meio da digitalização de dados pelo uso combinado com sistemas ciber-físicos (CPS) associados à sistemas autônomos descentralizados (ADS), a tecnologia IoT possui papel fundamental na estruturação de ambientes inteligentes com a sua aplicação aliada a outras tecnologias inovadoras promovidas pela mobilização da quarta revolução industrial, também conhecida como a era da conectividade.

Figura 2.2: Evolução IoT



Fonte: (SA et al., 2019)

As principais características que estão associadas aos ambientes de aplicações IoT, considerando a definição da tecnologia utilizada em domínios, sistemas ou componentes são citadas por (FIROUZI et al., 2020), sendo resumidas por:

- **Coisas:** Refere-se a objetos ou dispositivos com elementos que possuem capacidade de interagir com sistemas ou pessoas usando a internet;
- **Conectividade:** Refere-se a interconexão em rede utilizando protocolos de comunicação;
- **Dados:** refere-se ao conjunto de dados transportados em forma digital;

- **Inteligência:** Refere-se aos potenciais extraídos a partir da troca e do volume de informações enviadas pela rede;
- **Ação:** refere-se a tomada de decisões estratégicas das partes interessadas e também por decorrência da inteligência de acordo com a maturidade de cada domínio;
- **Ecossistema:** Refere-se a todo o processo, desde a coleta de dados até a tomada de decisão, incluindo dispositivos, sistemas, processos, plataformas e tecnologias associadas ao domínio;
- **Heterogeneidade:** refere-se a interoperabilidade dos dispositivos que possuem capacidade de funcionamento com plataformas e redes distintas;
- **Mudanças Dinâmicas:** Refere-se ao estado operacional dos dispositivos, com telemetria dos dados dinâmica;
- **Larga Escala:** refere-se a possibilidade de aumentar o número de dispositivos interconectados e a capacidade de aumento na geração de dados do sistema;
- **Segurança e Privacidade:** Refere-se a segurança e a privacidade das informações relativas ao processo, devendo ser definida de acordo com a relevância e a sensibilidade da informação trocada pela rede de dados no ecossistema IoT.

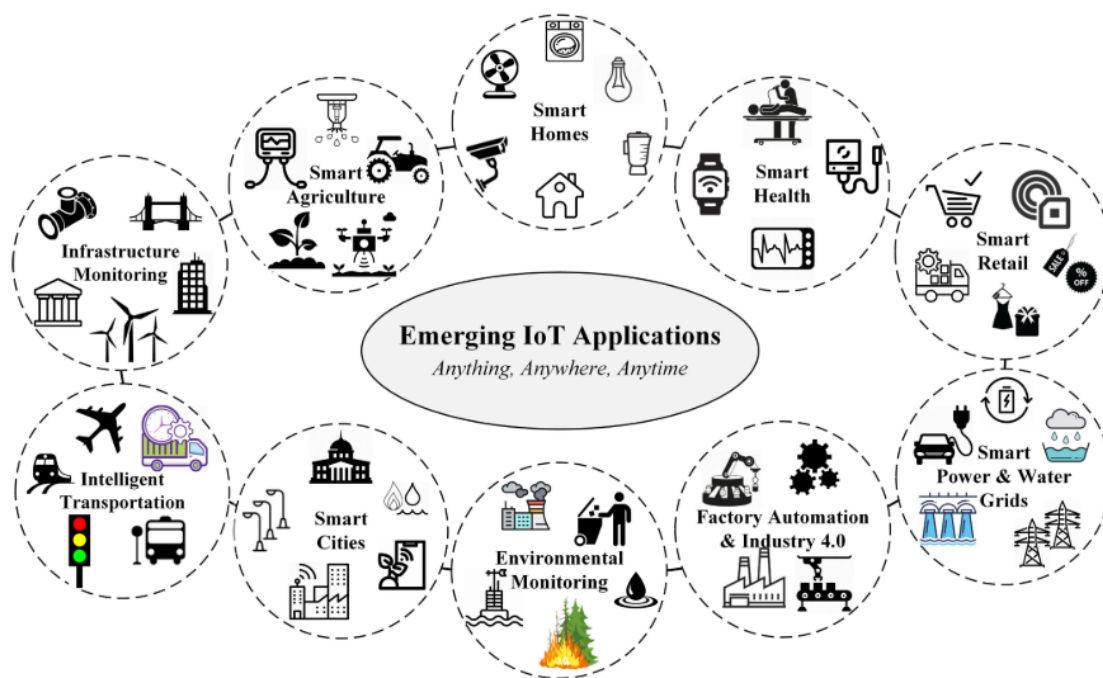
De acordo com (SWAMY; KOTA, 2020), os dados obtidos por uma pesquisa realizada pela Forbes mencionaram que a estimativa era que cerca de 75,44 bilhões de objetos estariam conectados à internet até 2025, representando um crescimento na economia entre 2,7 a 6,2 trilhões de dólares. Entretanto, conforme (JUNIOR, 2019), a expectativa global publicada no relatório do GCI (em inglês, *Global Connectivity Index*) ou Índice de Conectividade Global, é que em 2025 o número de dispositivos conectados pode chegar a cerca de 100 bilhões. Logo, o impacto que o decreto do *lockdown* causou aos diversos setores da indústria em virtude da pandemia de COVID-19, refletiu diretamente na economia mundial das organizações, podendo elevar o número ainda mais para as organizações que inovarem seus processos operacionais e estratégicos.

Entre as áreas emergentes para aplicação da tecnologia IoT, estão:

- Saúde;
- Indústria;
- Energia;
- Transporte;
- Varejo;
- Agricultura;

- Cidades e;
- Monitoramento de Infraestruturas.

Figura 2.3: Aplicações IoT



Fonte: (SWAMY; KOTA, 2020)

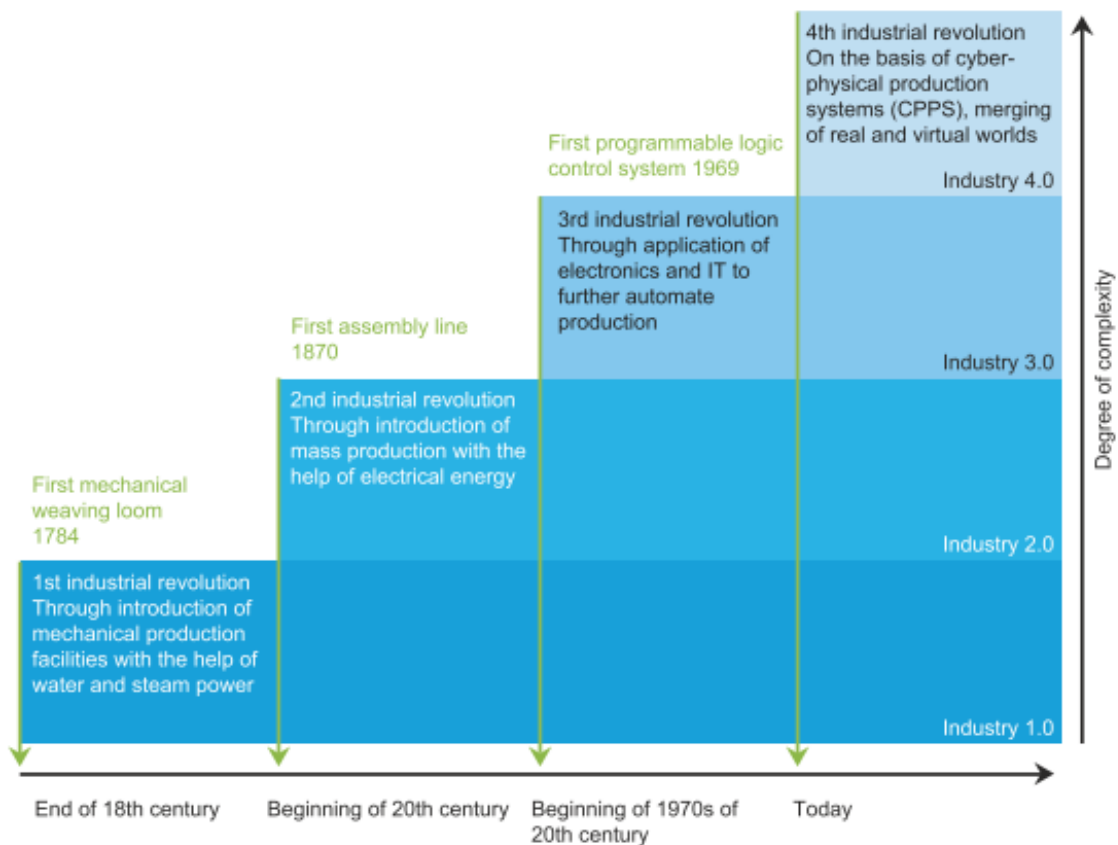
Na Figura 2.3 é possível identificar os principais domínios de aplicação que são diretamente impactados pelo uso da tecnologia IoT. Nesse cenário, o domínio de aplicação do modelo de integração, proposto na pesquisa, pode ser relacionado com algumas áreas, incluindo Energia Inteligente (em inglês, *Smart Power*) e ambientes de monitoramento (em inglês, *Environmental Monitoring*), conforme os seguintes contextos:

- *Smart Power*: associado ao processo de geração distribuída, com o uso de módulos de geração de energia fotovoltaica e com os seus sinais centralizados pelo concentrador dos inversores e digitalizados pelo dispositivo CPS, como *Gateway MQTT* do processo físico;
- *Environmental Monitoring*: associado ao processo de monitoramento através do sistema colaborativo que subscreve-se no tópico de interesse e permite a visualização das informações do processo físico.

2.2 Indústria 4.0

Atualmente, o período que vivemos está sendo considerado como a era digital. É nesse contexto que o conceito de Indústria 4.0 está diretamente relacionado ao movimento de transformação digital, sendo motivado pela mobilização da Quarta Revolução Industrial, onde os ambientes de manufatura investem em dispositivos de alta tecnologia para execução de seus processos produtivos, com a mínima interação possível de seres humanos, desde a etapa inicial até a entrega do produto, tornando-os inteligentes e auto-adaptáveis a diferentes demandas e quantidades (NAYYAR; KUMAR, 2020). Sendo assim, o modelo de integração, proposto no trabalho de pesquisa, está essencialmente relacionado com as tecnologias que são promovidas pela Indústria 4.0, como é o caso da tecnologia IoT (em inglês, *Internet of Things* ou Internet das Coisas), é considerada como um dos principais pilares dessa mudança, utilizado para integração entre sistemas ciber-físicos (em inglês, *Cyber-Physical Systems* ou Sistemas Ciber-físicos), tornando possível a digitalização dos dados provenientes de processos produtivos em uma planta física operacional.

Figura 2.4: Fases da Revolução Industrial



Fonte: (Landeta Echeberria, 2020)

Dessa forma, todas as fases estão associadas ao avanço tecnológico do setor industrial, incluindo o aumento de produtividade, a redução de custos com matéria-prima e o avanço no desenvolvimento de novos modelos no processo de manufatura do setor fabril.

Entretanto, um fator comum e relevante entre cada revolução está vinculado ao quão inovador foram as ideias, criações e avanços realizados no decorrer dos anos, conforme (Landeta Echeberria, 2020). Resumidos a seguir:

- **Primeira Revolução Industrial (1784):** A primeira fase é conhecida por representar a transição entre o trabalho que antes era realizado de forma manual para a utilização de máquinas mecânicas, geralmente movidas a água ou vapor, tornando a produção mecânica um marco fundamental para as indústrias têxteis e ferroviárias da época;
- **Segunda Revolução Industrial (1870):** Já a segunda fase está ligada ao uso da eletricidade como fonte de energia de ambientes produtivos, novamente acelerando os processos de manufatura do setor industrial, com foco na produção em massa e redução de custos;
- **Terceira Revolução Industrial (1969):** Na terceira fase surge a introdução da automação de processos, com a utilização de controladores lógicos programáveis, computadores e softwares em ambientes de rede;
- **Quarta Revolução Industrial (2011):** Agora, na quarta fase inicia-se então a transformação digital dos processos de manufatura, minimizando ainda mais a intervenção humana e oferecendo novos conceitos, como: IoT; Computação em Nuvem; *Machine Learning*; Virtualização; entre outros, sendo um novo marco para o avanço tecnológico nos mais variados setores.

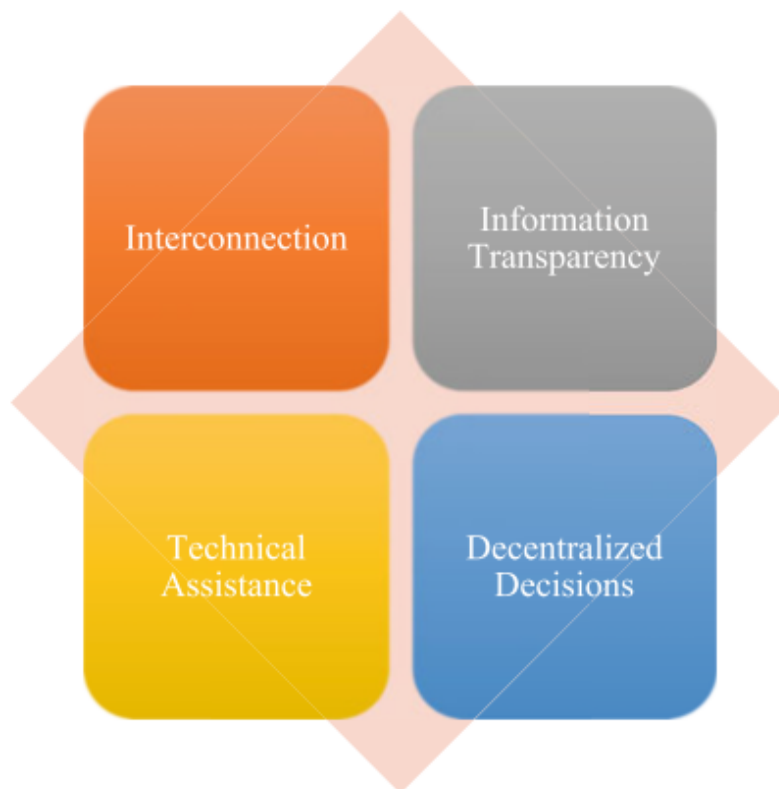
Quanto à origem do nome "Indústria 4.0", segundo (FIROUZI et al., 2020), surgiu em 2011 na feira de *Hannover* em decorrência de um projeto de pesquisa do governo da Alemanha para a indústria de manufatura. Da mesma forma, (Landeta Echeberria, 2020) complementa que após estudos analíticos acerca do tema realizada pelo Departamento de Política do Parlamento Europeu, considera-se que o conceito de Indústria 4.0, também conhecido como Manufatura Avançada, é inovador e disruptivo proporcionando ganhos relevantes na produtividade e, conseqüentemente, a obtenção por melhores resultados.

Historicamente, como forma de incentivo à implementação, (MATT; MODRÁK, 2021) ressalta sobre os primeiros planos nacionais que foram oferecidos pela Europa em

2015, por intermédio de suporte financeiro e redução de impostos para implementação do conceito de Indústria 4.0 promovendo o avanço tecnológico no setor.

Então, para melhor interpretação sobre o conceito da Indústria 4.0, (NAYYAR; KUMAR, 2020) apresenta os princípios fundamentais sobre esta mudança no processo produtivo, conforme mostra a Figura - Princípios da Indústria 4.0.

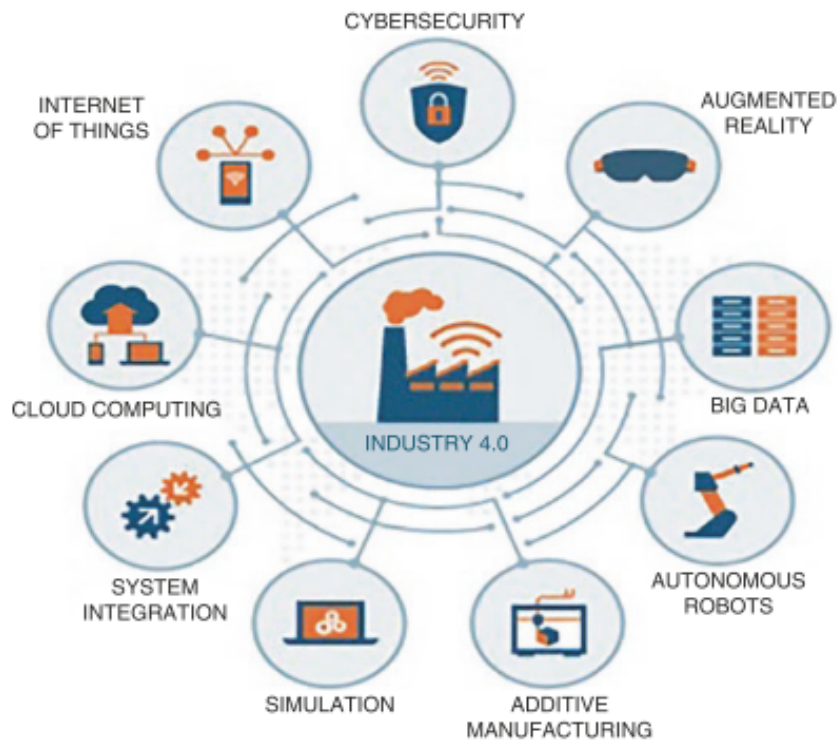
Figura 2.5: Princípios da Indústria 4.0



Fonte: (NAYYAR; KUMAR, 2020)

- **Interconexão:** Princípio essencial para transmissão de mensagens entre componentes, sistemas e domínios, auxiliando a integração dos dados;
- **Transparência de Informação:** Facilidade na visualização dos dados, auxiliando na melhoria dos processos;
- **Assistência Técnica:** Sistemas de assistência orientado aos dados que são coletados e analisados, auxiliando no suporte e na manutenção dos processos de fabricação;
- **Decisões Descentralizadas:** Capacidade de tomada de decisões de forma autônoma, exceto em situações específicas, onde seja necessário intervenção do nível estratégico.

Figura 2.6: Pilares da Indústria 4.0



Fonte: (LEA, 2018)

A partir daí é possível entender sobre os principais pilares que formam o conceito da Indústria 4.0, conforme mostra a Figura 2.6. No caso de IoT e Integração de Sistemas. Isto é possível devido aos sistemas ciber-físicos (em inglês, *Cyber-Physical Systems*) estarem presentes nos ambientes físicos fortalecendo o conceito de Fábricas Inteligentes (em inglês, *Smart Factory*) com capacidade de interagir com máquinas em tempo real, entre dispositivos inteligentes integrados e elementos físicos à sistemas computacionais, envolvendo o desenvolvimento da manufatura industrial em ambientes produtivos inteligentes (Landeta Echeberria, 2020).

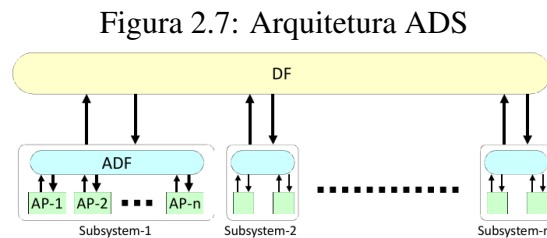
Assim, (NAYYAR; KUMAR, 2020) descreve algumas das tecnologias complementares que são consideradas fundamentais para implantação do conceito da Indústria 4.0, sendo:

- *Cloud Computing*;
- *Big Data Analytics*;
- *Industrial Internet of Things (IIoT)*;
- *Cyber-Physical Systems (CPS)*.

2.3 Conceitos e Tecnologias Utilizadas

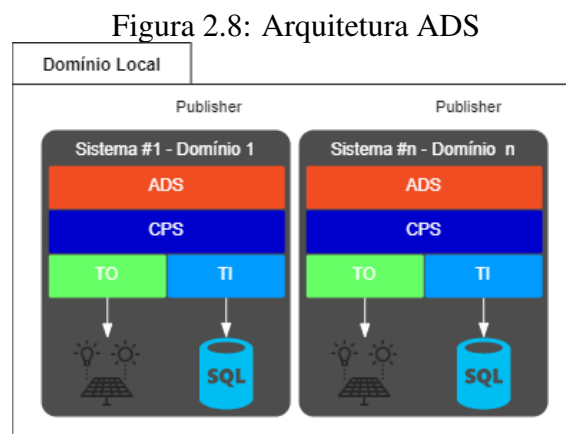
2.3.1 Sistemas Autônomos Descentralizados

No modelo de integração proposto neste trabalho o termo ADS (em inglês, *Autonomous Decentralized System*) é utilizado como abstração do conjunto de sistemas que compõem o domínio local, nesse caso uma planta fotovoltaica constituída por módulos, entidades e elementos que fazem parte do processo operacional. Da mesma forma, (KUNIFUJI, 2013) define que ADS é o resultado da integração de subsistemas, facilitando o uso de modelos de aplicação de softwares de manutenção.



Fonte: (KUNIFUJI, 2013)

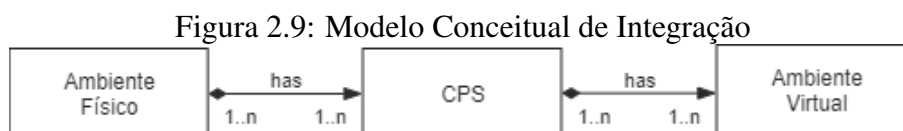
Além de tudo, a concepção de ADS pode agregar recursos de *OT* (em inglês, *Operational Technology*), na qual hardwares são constituídos por dispositivos com capacidade de processamento, armazenamento, comunicação, além de componentes de entrada e saída servindo de interface entre os processos operacionais e os sistemas de supervisão e aquisição de dados (SCADA, em inglês *Supervisory Control and Data Acquisition*). Da mesma forma, recursos de *IT* (em inglês, *Information Technology*) são utilizados para disponibilizar os dados coletados às camadas superiores de softwares atuando na dimensão tática e auxiliando na tomada de decisões estratégicas.



Fonte: Adaptado de (SA et al., 2019)

2.3.2 Sistemas Ciber-físicos

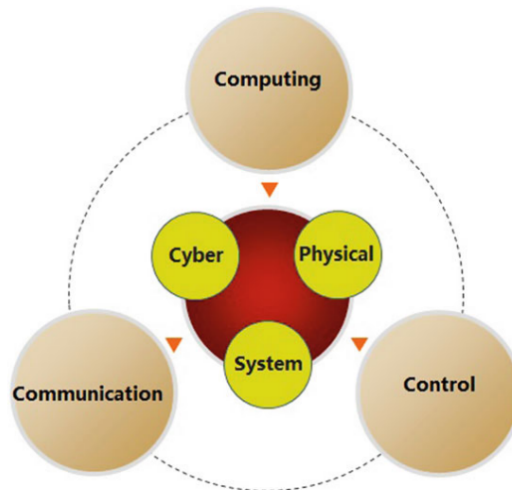
Exercendo um papel fundamental na colaboração entre os domínios da aplicação, permitindo a interação com o sistema colaborativo definidos no modelo de integração sugerido na pesquisa, os sistemas ciber-físicos também são conhecidos como produção física cibernética, sendo uma das principais entidades responsáveis pela virtualização das informações dos ambientes físicos ao ambiente virtual, conforme ilustra a Figura 2.9.



Fonte: Autor

Outro ponto importante diz respeito à implementação quando combinada com outras tecnologias como, por exemplo, a tecnologia IoT e Integração de Sistemas pode integrar ambientes usando a infraestrutura web para transmissão de dados sobre o protocolo de comunicação TCP/IP.

Figura 2.10: 3C do CPS

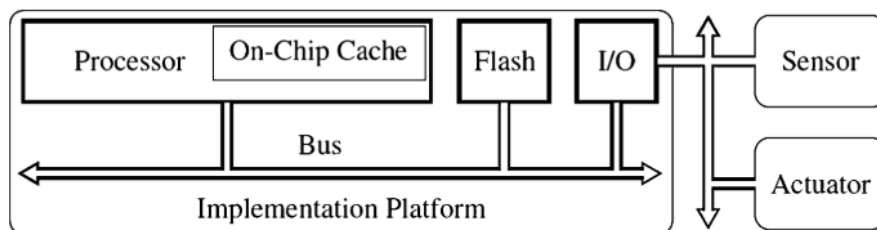


Fonte: (WANG; GAO, 2020)

Além de que, por possuírem elementos computacionais, interfaces de comunicação e unidades de controle, como mostra a Figura 2.10, podemos considerar os sistemas ciber-físicos como dispositivos inteligentes cibernéticos que permitem a estruturação de ambientes flexíveis colaborando para funcionamento em diferentes aplicações, onde podem existir múltiplos fabricantes do mesmo jeito que podem haver uma variedade de protocolos e interfaces de comunicação entre os elementos físicos e sistemas de softwares em rede.

Existem algumas discussões em relação a definição dos sistemas ciber-físicos em razão das mais variadas aplicações, mas (KRAVETS; BOLSHAKOV, 2021) define que CPS é um conjunto de elementos formado por componentes físicos, sistemas de medição, sistemas computacionais e interfaces de comunicação que embarcam algoritmos de controle. Já (MATT; MODRÁK, 2021) ressalta que os sistemas ciber-físicos (CPS) representam muito para a tecnologia da Indústria 4.0, pois são fundamentais no caminho da mudança do setor de manufatura, possibilitando a estruturação de processos inteligentes de acordo com novos modelos de negócios.

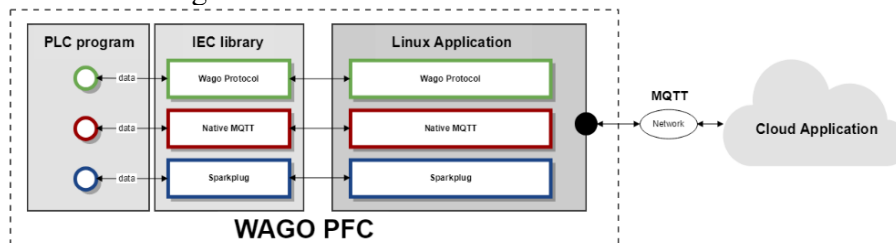
Figura 2.11: Arquitetura Sistemas Ciber-físicos



Fonte: (FIROUZI et al., 2020)

No contexto da pesquisa, a tecnologia CPS está diretamente associada ao sistema de automação de uma planta fotovoltaica, onde as tarefas de leitura e atuação são executadas por programas embarcados no Controlador Lógico Programável (CLP), servindo de *Gateway* IoT, onde o agente envia os dados ao *Broker*, sendo responsável pela interação entre o concentrador de dados dos inversores locais e o sistema de colaboração.

Figura 2.12: Overview do Protocolo de Dados



Fonte: (KG, 2020)

Por fim, na Figura 2.12, é possível interpretar, em forma de diagrama de blocos, as etapas que são executadas e que correspondem ao dispositivo ciber-físico utilizado. Primeiramente, um programa é desenvolvido por especialistas utilizando recursos de bibliotecas instaladas em aplicações embarcadas.

2.3.3 Protocolo MQTT

Utilizado para comunicação entre os domínios da aplicação, mapeados no Capítulo 4 - Proposta de um Modelo de Integração para Análise e Coleta de Dados, o protocolo MQTT (em inglês, *Message Queuing Telemetry Transport*) foi desenvolvido em 1999 por Andy Stanford (IBM) e Arlen Nipper (Arcom) para telemetria de oleodutos via satélite, sendo liberado para uso gratuito em 2010 e tornando-se um padrão OASIS (em inglês, *Organization for the Advancement of Structured Information Standards* em 29/10/2014 (HIVEMQ, 2020).

Segundo o padrão (OASIS, 2015), o MQTT (em inglês, *Message Queuing Telemetry Transport*) é um protocolo de comunicação utilizado para telemetria de mensagens com arquitetura do tipo Publicação/Assinatura (em inglês, *Publish/Subscribe* ou *Pub/Sub*) baseado no modelo de conexão Cliente/Servidor, onde os clientes são denominados como *Publisher* ou *Subscriber* e o servidor de *Broker*. Sobretudo, (KAWAGUCHI; BANDAI, 2019) ressalta sobre a utilização promissora do protocolo MQTT em aplicações IoT, por possuir características que favorecem a sua implementação devido a facilidade, largura de banda e o suporte de conexões sem perdas, executado amplamente sobre o protocolo de rede TCP/IP ou protocolos bidirecionais.

2.3.3.1 Principais Conceitos

Os principais conceitos sobre o protocolo MQTT são apresentados conforme a Tabela 2.1, onde é possível identificar os papéis e responsabilidades de cada elemento dentro da estrutura de comunicação oferecida pelo modelo *Pub/Sub*.

Tabela 2.1: Principais Conceitos do MQTT

Nome	Descrição
<i>Publisher</i>	Cliente responsável por transmitir os pacotes de dados
<i>Subscriber</i>	Cliente responsável por receber os pacotes de dados
<i>Broker</i>	Servidor responsável por receber, filtrar e distribuir os pacotes de dados publicados aos clientes
<i>Message</i>	O pacote de dados que trafega pela rede
<i>Topic</i>	A etiqueta de dados anexada a mensagem, acessível aos clientes por meio de comparação
<i>Publish</i>	Ação de enviar uma mensagem a um tópico de interesse no Broker
<i>Subscribe</i>	Ação de inscrição a um tópico de interesse no Broker
<i>Unsubscribe</i>	Ação de solicitar o cancelamento na inscrição de um tópico de interesse para não receber mensagens
<i>Payload</i>	Conteúdo de dados enviados na mensagem
<i>QoS</i>	A qualidade de serviço, onde o Broker deve receber e/ou enviar os dados conforme os parâmetros: 0 = No máximo uma vez (em inglês, At most once) 1 = Pelo menos uma vez (em inglês, At least once) 2 = Exatamente uma vez (em inglês, Exactly once)
<i>Last Will / Will Message</i>	Mensagem a ser enviada pelo Broker caso haja uma desconexão do cliente
<i>Clean</i>	Indica a persistência da sessão, relacionada aos nomes dos tópicos inscritos e ao recebimento de mensagens

Fonte: Adaptado de (COISAS, 2019)

2.3.3.2 Estrutura do Protocolo MQTT

Na estrutura de mensagem do protocolo MQTT, conforme mostra a Figura 2.13, é possível notar que os bytes 1 e 2 são exclusivos para formatação do cabeçalho e os restantes são dedicados ao conteúdo (*payload*) das mensagens que são limitadas em 256 MB de dados (FIROUZI et al., 2020).

Figura 2.13: Estrutura de Controle de Pacote

Fixed header, present in all MQTT Control Packets
Variable header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

Fonte: (OASIS, 2015)

A seguir, a Figura 2.14 mostra que o byte 1 forma o cabeçalho fixo das mensagens. Entretanto, o byte 2 fica disponível para ser utilizado no restante do cabeçalho de acordo com alguns tipos de controle de pacote.

Figura 2.14: Formato do Cabeçalho Fixo

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2...	Remaining Length							

Fonte: (OASIS, 2015)

Deste modo, a Figura 2.14 indica que os bits (7-4) são responsáveis pelos tipos de controle de pacote, conforme mostra a Tabela 2.2.

Tabela 2.2: Estrutura de Controle de Pacotes MQTT

Nome	Valor	Associação	Descrição
<i>Reserved</i>	0	-	Reservado
<i>CONNECT</i>	1	Cliente-Servidor	O cliente solicita uma conexão a um servidor
<i>CONNACK</i>	2	Servidor-Cliente	Confirmação de solicitação de conexão
<i>PUBLISH</i>	3	Bidirecional	Publicação de mensagem
<i>PUBACK</i>	4	Bidirecional	Confirmação de mensagem publicada
<i>PUBREC</i>	5	Bidirecional	Publicação recebida (QoS=2, parte 1)
<i>PUBREL</i>	6	Bidirecional	Publicação recebida (QoS=2, parte 2)
<i>PUBCOMP</i>	7	Bidirecional	Publicação recebida (QoS=2, parte 3)
<i>SUBSCRIBE</i>	8	Cliente-Servidor	Inscrição nos tópicos
<i>SUBACK</i>	9	Servidor-Cliente	Confirmação de inscrição
<i>UNSUBSCRIBE</i>	10	Cliente-Servidor	Cancelamento de inscrição em tópicos
<i>UNSUBACK</i>	11	Servidor-Cliente	Confirmação de cancelamento de inscrição
<i>PINGREQ</i>	12	Cliente-Servidor	Solicitação de PING
<i>PINGRESP</i>	13	Servidor-Cliente	Resposta de PING
<i>DISCONNECT</i>	14	Cliente-Servidor	Notificação de desconexão
<i>Reserved</i>	15	-	Reservado

Fonte: Adaptado de (OASIS, 2015)

Já a Tabela 2.3 mostra que os bits (3-0) do byte 1, apresentados na a Figura 2.14 são exclusivos para configuração dos parâmetros específicos, sinalizados em cada tipo de controle.

Tabela 2.3: Identificação de Bits

Pacote de Controle	Sinalizadores de Cabeçalho	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reservado	0	0	0	0
CONNACK	Reservado	0	0	0	0
PUBLISH	Usado no MQTT 3.1.1	DUP	QoS		RETAIN
PUBACK	Reservado	0	0	0	0
PUBREC	Reservado	0	0	0	0
PUBREL	Reservado	0	0	1	0
PUBCOMP	Reservado	0	0	0	0
SUBSCRIBE	Reservado	0	0	1	0
SUBACK	Reservado	0	0	0	0
UNSUBSCRIBE	Reservado	0	0	1	0
UNSUBACK	Reservado	0	0	0	0
PINGREQ	Reservado	0	0	0	0
PINGRESP	Reservado	0	0	0	0
DISCONNECT	Reservado	0	0	0	0

Fonte: Adaptado de (OASIS, 2015)

Onde no Pacote de Controle **PUBLISH**, usando o MQTT 3.1.1, encontramos nos bits (3-0) os seguintes parâmetros DUP², QoS³ e RETAIN⁴, respectivamente.

2.3.3.3 Conexão de Rede

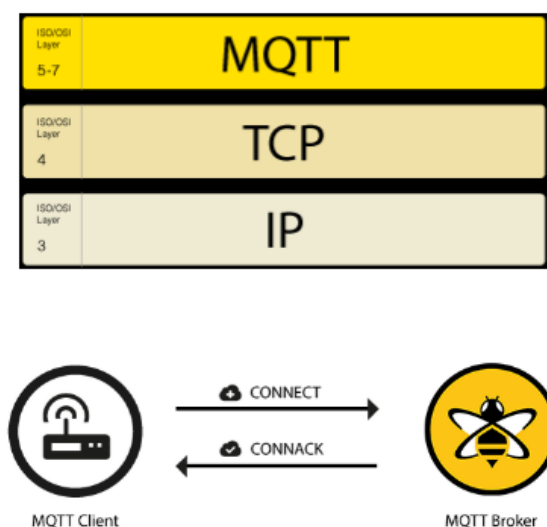
A conexão entre os clientes, Assinantes e/ou Publicadores de dados, e o servidor *Broker* é baseada no modelo TCP/IP, fazendo referência as camadas 3 (Internet), 4 (Transporte) e 5-7 (Sessão, Apresentação e Aplicação) do modelo ISO/OSI , usando os protocolos de comunicação IP, TCP e MQTT.

²Entrega duplicada de um pacote de controle da ação Publicar.

³Qualidade de Serviço da ação Publicar

⁴Sinalizador de retenção da ação Publicar

Figura 2.15: Conexão de Rede MQTT



Fonte: (HIVEMQ, 2020)

A Figura 2.15 mostra um exemplo de conexão onde um cliente MQTT solicita uma conexão com o servidor *Broker*. Assim que o controle de pacote *CONNECT* é transmitido, o cliente recebe uma confirmação de solicitação através do controle *CONNACK*, ambos já descritos na Tabela 2.2.

2.3.3.4 Qualidade de Serviço

A entrega de mensagens é realizada conforme a definição dos níveis de qualidade de serviço (QoS), com simetria do protocolo de entrega os agentes de dados podem ser Remetentes ou Destinatários. Assim, o tratamento de envio é individual preocupando-se exclusivamente com a entrega da mensagem de um Remetente até um Destinatário (OASIS, 2015).

Tabela 2.4: Definições de QoS

Valor de QoS	Bit 2	Bit 1	Descrição
0	0	0	Entrega somente uma vez
1	0	1	Entrega pelo menos uma vez
2	1	0	Entrega exatamente uma vez
-	1	1	Reservado - não deve ser usado

Fonte: Adaptado de (OASIS, 2015)

Os níveis de qualidade de serviço apresentados na Tabela 2.4, estabelecidos para o MQTT 3.1.1, são mencionados por (SWAMY; KOTA, 2020):

- QoS 0 (*At most once delivery*): Entrega somente uma vez;
- QoS 1 (*At least once delivery*): Entrega pelo menos uma vez;

- QoS 2 (*Exactly once delivery*): Entrega exatamente uma vez.

2.3.3.5 Segurança

Não há uma exigência normativa estabelecida pelo padrão OASIS, mas existe uma recomendação para que servidores SSL/TLS utilizem a porta TCP 8883 para conexões seguras criptografadas (OASIS, 2015). Visto que a porta TCP padrão é 1883 utilizada para conexões não criptografadas.

2.3.4 Broker Mosquitto

O papel do servidor *Broker* no modelo de integração proposto é gerenciar as conexões dos clientes Assinantes e Publicadores responsáveis pelo envio de mensagens ao tópico de interesse. Além disso, o servidor *Broker* também desempenha as funções de receber, filtrar e entregar as mensagens aos respectivos destinatários dentro da arquitetura de comunicação *Pub/Sub* (HIVEMQ, 2020).

Desenvolvido pela *Eclipse Foudation* o Mosquitto é um software utilizado na grande maioria das aplicações IoT. Por ser desenvolvida em linguagem C, possui vantagens por ser leve, não exigindo muitos recursos para seu processamento, podendo ser instalado e executado em ambientes locais ou remotos, pois tem a capacidade de rodar em alguns sistemas operacionais, entre os principais estão:

- *Linux*;
- *Mac*;
- *Windows* e;
- *Raspberry Pi*.

Neste trabalho foi utilizado como servidor *Broker* a versão 1.6.10 do software Mosquitto.

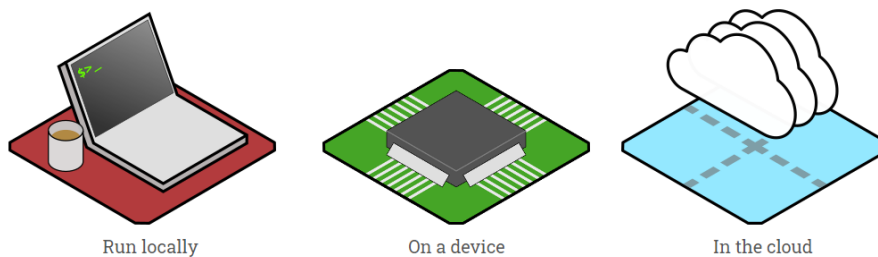
2.3.5 Node-RED

Desenvolvido originalmente em 2013 por Nick O’Leary e Dave Conway-Jones, ambos da IBM a plataforma que atualmente é mantida pela *JS Foundation*, tornou-se de código aberto em setembro de 2013 (NODE-RED, 2021). No modelo de integração proposto na pesquisa o software irá ser utilizado para subscrição no tópico onde serão publicadas as mensagens provenientes do CPS (*Topic:"1.5.0/CID/TagValues"*), servido de sistema colaborativo

2.3.5.1 Características

Considerado como uma plataforma multifuncional para diversos contextos, incluindo aplicações IoT, o Node-RED é um software de código aberto, com programação orientada a fluxos baseados em *browser* e nós funcionais (ONUKEI; OLIVEIRA, 2017).

Figura 2.16: Plataformas de Instalação



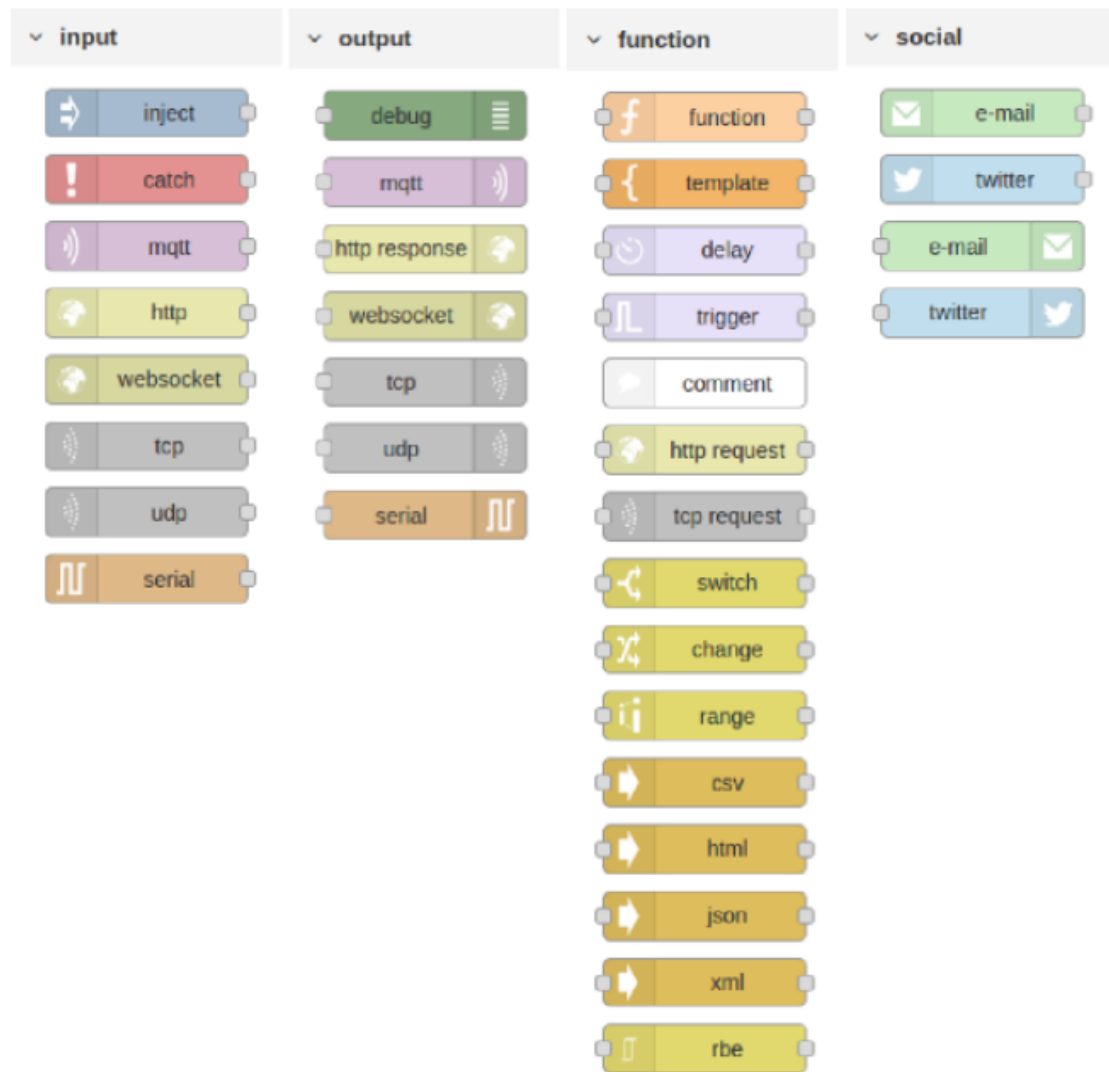
Fonte: (NODE-RED, 2021)

Segundo (HEYVAERT et al., 2019) o Node-RED pode ser implementado em ambiente locais, como um adaptador de dispositivo, ou como um serviço Web. E, de acordo com a Figura 2.16, também é possível executar o software embarcado em hardwares, assim como em nuvem.

2.3.5.2 Funcionalidades

A ferramenta possui um catálogo variado de funcionalidades permitindo maior flexibilidade na estruturação de fluxos que, por sua vez, são orientados a eventos, assim como uma comunidade *online* ativa que contribui no desenvolvimento de novas funcionalidades por meio de fóruns e repositórios o compartilhamento de projetos, compreendendo: coleções; nós com funcionalidades específicas e fluxos para uma variedade de aplicações, garantindo a evolução do software.

Figura 2.17: Nós de função



Fonte: (ONUKEI; OLIVEIRA, 2017)

3 TRABALHOS RELACIONADOS

Este capítulo descreve os trabalhos que estão relacionados com o tema da pesquisa que envolve o modelo de integração sugerido neste trabalho, abrangendo os contextos e fundamentos mais relevantes e compatíveis com as tecnologias e os conceitos utilizados nesta monografia. A distribuição do conteúdo está dividida por seções que estão relacionadas aos trabalhos pesquisados e subseções, contendo críticas, vantagens, desvantagens e diferenças relacionadas ao assunto.

3.1 Arquitetura ATIoT: Integrando Tecnologias Assistivas com Internet das Coisas

No trabalho de (JUNIOR, 2019) o principal objetivo foi apresentar uma proposta de arquitetura com a finalidade de integrar a Tecnologia Assistiva empregando os conceitos oferecidos pela tecnologia de Internet das Coisas, a fim de viabilizar a inclusão da acessibilidade do grupo de pessoas portadoras de deficiência ao mundo da tecnologia de informação e comunicação (TIC), onde hardwares e softwares podem trazer diversos benefícios à sociedade como um todo. Desta forma, o aspecto que se assemelha ao modelo de integração sugerido nesta pesquisa diz respeito à capacidade de integração entre sistemas abordando recursos tecnológicos avançados para facilitar a interoperabilidade e convergência entre sistemas e aplicações.

3.1.1 Fundamentação

O termo Tecnologia Assistiva (TA) originou-se de “Ajudas Técnicas”, com algumas referências importantes como, por exemplo, os decretos: 3.298 de 1999 e 5.296 de 2004, ambos visando à inclusão social de pessoas com limitações funcionais motoras, sensoriais ou mentais. Considerada como uma disciplina que reuni um grupo de profissionais de diferentes áreas de atuação com ênfase em pesquisas sobre o desempenho humano minimizando o impacto das limitações com atividades convencionais, assim como o autocuidado e a performance na execução de atividades profissionais, a definição de dispositivo de TA pode ser considerada como qualquer objeto, adequação de objeto ou produto que seja utilizado para potencializar a capacidade funcional desse grupo de pessoas.

3.1.2 Vantagens e Desvantagens

Primeiramente, uma das principais vantagens apresentadas no trabalho diz respeito ao benefício da TA ao ser humano possibilitando a inclusão digital à sociedade por meio de aumento de capacidades funcionais com uso da tecnologia. E, na dimensão tecnológica, temos diversos ganhos, um deles está relacionado ao ao padrão *publisher/subscriber*, onde o modelo de comunicação se baseia na distribuição de dados centralizada em um servidor intermediário chamado de *Broker*, onde existem conexões de clientes independentes e a possibilidade de trabalho com redes de largura de banda limitadas. Além disso, dispositivos fabricados ou adaptados com o conceito de TA podem ser executados em ambientes cloud, sendo executados como SaaS (em inglês, *software as a service*), melhorando assim aspectos comuns e fundamentais para o sistema:

1. Acessibilidade;
2. Confiabilidade;
3. Escalabilidade;
4. Maximização de recursos e;
5. Segurança.

3.1.3 Críticas e Diferenças

Por outro lado, algumas desvantagens podem estar relacionadas na infraestrutura de comunicação, pois quando aplicações ou serviços são instalados em servidores *on-premise*, por exemplo, alguns gastos podem impactar no desempenho financeiro do sistema. Em contrapartida, serviços hospedados em nuvem, apesar de oferecerem todos os benefícios já mencionados, também podem gerar custos elevados durante as fases iniciais para implantação do projeto.

3.2 Uma Abordagem para a Integração de Sistemas Industriais Aplicando o Conceito de Internet das Coisas e de Modelos Semânticos no Contexto da Indústria 4.0

Já o trabalho de (STEINMETZ, 2018) aborda sobre o uso de modelos semânticos baseados em conceitos da Indústria 4.0, como é o caso da Internet das Coisas (IoT), considerada um dos principais pilares da quarta revolução industrial, a fim de padronizar a representação das mensagens que são trocadas entre dispositivos facilitando a interoperabilidade entre sistemas. Além do mais, a proposta ainda permite a geração automática de interfaces de comunicação e códigos fonte de acordo com a ontologia modelada.

3.2.1 Fundamentação

Em síntese, a estruturação do modelo proposto pelo autor faz uso de ontologias baseadas em Web Semântica usando para construção do trabalho a linguagem *OWL* (em inglês, *Web Ontology Language*) onde existem elementos básicos como classes, indivíduos e propriedades. A relação conceitual pode ser comparada ao conceito de programação orientada a objetos devido a suas similaridades onde a ferramenta mais conhecida para modelagem de ontologias é a ferramenta Protégé, de código aberto e bastante difundida permitindo a visualização e edição de modelos. Outro item fundamental que foi agregado ao modelo, além de ser muito útil no emprego da Internet das Coisas, são os *Middlewares* que podem ser resumidos como um software intermediário entre as aplicações e o sistema operacional (OS). O trabalho ressaltou o uso da ferramenta *FIWARE* contribuindo para a modularização do projeto e a integração entre as aplicações através do *Context Broker*, módulo centralizador, que gerencia as informações de acordo com o número de consumidores.

3.2.2 Vantagens e Desvantagens

As aplicações baseadas em modelos semânticos apresentam vantagens quando o modelo sobre as características do processo em questão tiver sido fortemente mapeado por especialistas do processo. Além disso, a otimização do desenvolvimento do código fonte da aplicação também ganha destaque no modelo proposto pelo autor.

Contudo, alguns requisitos da ferramenta *FIWARE* podem ser os requisitos básicos para funcionamento em múltiplas plataformas, até mesmo em dispositivos embarcados, um conceito básico necessário para atender a demanda de aplicações distribuídas. Nesse contexto, a falta de infraestrutura de rede em ambientes fabris também pode dificultar o funcionamento do modelo sugerido.

3.2.3 Críticas e Diferenças

Apesar da proposta do trabalho apresentar uma arquitetura de sistema baseada em modelos semânticos integrados com os conceitos da Indústria 4.0, usando a tecnologia IoT e *Middlewares* para viabilizar o relacionamento entre as principais entidades entre camadas da aplicação, faltou uma análise de desempenho por parte do autor de tal forma que o sistema fosse submetido a um cenário de extrapolação. Desta forma, seria possível detectar possíveis problemas relacionados ao modelo. Por fim, as principais diferenças entre os aspectos abordados pelo autor em relação ao tema da pesquisa do modelo de integração proposto na pesquisa deste trabalho diz respeito ao emprego de ontologias e *Middlewares*.

3.2.4 Comparação entre Modelos de Integração Relacionados e Proposto

Desta forma, comparando o modelo proposto neste trabalho com os trabalhos relacionados ambos os modelos permitem a integração de dados por meio da tecnologia de IoT e pela comunicação baseada no protocolo MQTT, integrando: coisas; dispositivos; equipamentos; sistemas e domínios. Assim, de modo geral, os conceitos abordados pelos autores também estão associados ao uso de tecnologias baseadas na Indústria 4.0 e IoT.

Para concluir, as principais diferenças que envolvem os modelos sugeridos por (JUNIOR, 2019) e (STEINMETZ, 2018) em comparação ao modelo proposto neste trabalho, estão relacionadas as características das entidades que fazem parte dos domínios de cada aplicação escolhidos pelos autores. Além do mais, o uso de ontologias, assim como da computação em nuvem, mencionados nos trabalhos também podem se aplicar ao modelo de integração de dados proposto.

4 PROPOSTA DE UM MODELO DE INTEGRAÇÃO PARA ANÁLISE E COLETA DE DADOS

Este capítulo descreve a respeito das atividades que foram realizadas para definição do modelo de integração proposto, sendo distribuído em três principais seções:

- Planejamento;
- Visão Geral;
- Características sobre o Domínio da Aplicação e;
- Modelagem da Aplicação.

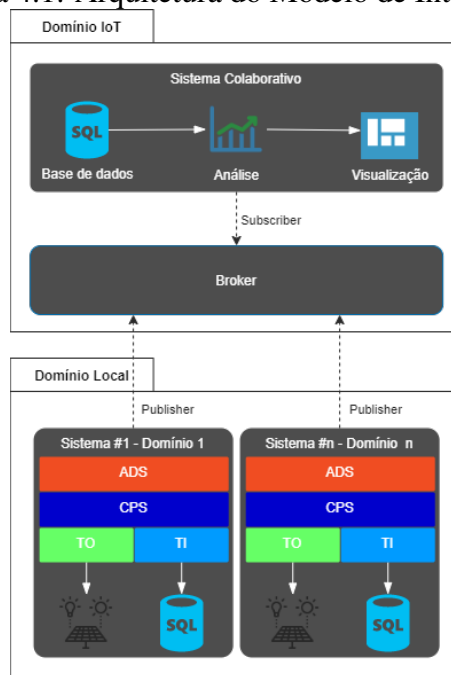
4.1 Planejamento

Inicialmente, foi necessário a realização de um planejamento para iniciar as atividades que auxiliaram na definição sobre os domínios da aplicação baseados em informações históricas de projetos executados anteriormente. Assim, foi possível identificar as classes, entidades, objetos e elementos envolvidos que fazem parte do processo de geração distribuída de uma planta solar e associá-los ao modelo de integração proposto. Desta forma, foi possível realizar o levantamento das características dos sistemas e os tipos de relacionamento entre as classes envolvidas durante a modelagem da aplicação por meio de modelos conceituais e diagramas de classes UML.

4.2 Visão Geral

A implementação do experimento busca a estruturação de um modelo de integração de dados de baixo custo, permitindo a interação entre os domínios da aplicação baseada em dados e no histórico de projetos de geração solar executados na prática. No modelo de integração proposto os dados serão agrupados em coleções orientadas a tópicos, onde as variáveis do processo foram mapeadas e classificadas de acordo com a sua relevância dentro de cada módulo associado ao domínio da aplicação. As informações são publicadas em um servidor intermediário utilizando a infraestrutura web para telemetria.

Figura 4.1: Arquitetura do Modelo de Integração

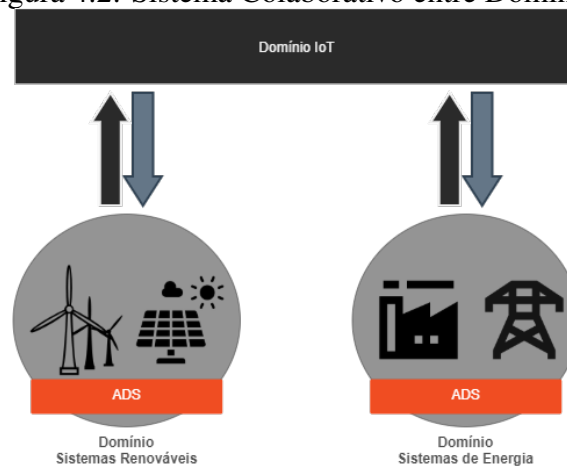


Fonte: Adaptado de (SA et al., 2019)

4.3 Características sobre o Domínio da Aplicação

O domínio da aplicação escolhido para implementação do modelo de integração proposto foi definido a partir da abstração de domínios IoT e Sistemas Renováveis ou de Energia, conforme mostra a Figura 4.2, formando assim um sistema colaborativo para o ambiente de monitoramento do processo de interesse.

Figura 4.2: Sistema Colaborativo entre Domínios



Fonte: Autor

Onde:

- *Domínio IoT*: Refere-se às características de um ambiente baseado nos conceitos de Internet das Coisas, colaborando com sistemas, componentes e elementos físicos;
- *Domínio Sistema Renovável*: Refere-se às características de um processo de geração distribuída baseado em sistemas fotovoltaicos para geração de energia renovável, com a colaboração entre módulos, equipamentos e dispositivos de supervisão e controle;
- *Domínio Sistema de Energia*: Refere-se às características de um processo baseado em subestações de energia para distribuição de energia, com a colaboração entre módulos, equipamentos, dispositivos de supervisão e controle.

4.3.1 Domínio IoT

O ambiente de colaboração entre domínios é formado pelo relacionamento e composição de entidades com recursos funcionais de coleta, análise ou gerenciamento de dados enviados pelos sistemas autônomos distribuídos.

4.3.2 Domínio Distribuído

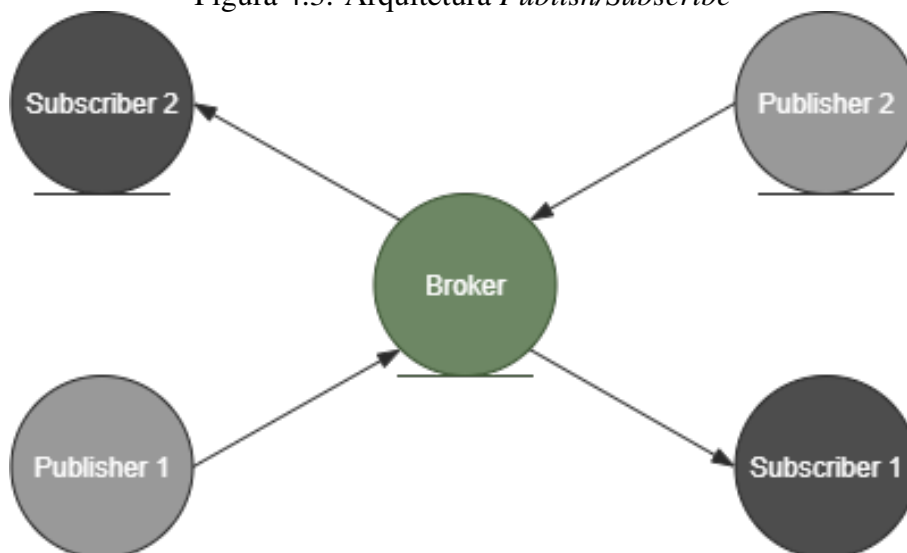
Nos domínios locais, sistemas autônomos descentralizados (ADS) representam o conjunto de componentes físicos instalados nas plantas que possuem elementos de comunicação em rede, componentes de tecnologia operacional (TO) e sistemas ciber-físicos (CPS) com a capacidade de integração entre domínios através da digitalização dos dados coletados por meio da tecnologia de informação (TI) utilizando os conceitos de IoT.

4.3.3 Relacionamento entre Domínios

O relacionamento de comunicação entre o sistema colaborativo e os domínios locais está amplamente baseado no uso de protocolos de comunicação de IoT e Internet. O protocolo MQTT é responsável pelo transporte das mensagens que carregam o conteúdo de carga útil ou *payload*, termo em inglês conhecido na área de comunicação para transmissão de dados, executado sobre o modelo TCP/IP. A informação é publicada em forma de tópico pelo CPS no *Broker*, agente intermediário localizado no domínio IoT, servindo

conteúdo aos agentes assinantes dos tópicos publicados, conforme propõe o modelo de arquitetura *Publish/Subscribe*.

Figura 4.3: Arquitetura *Publish/Subscribe*



Fonte: Autor

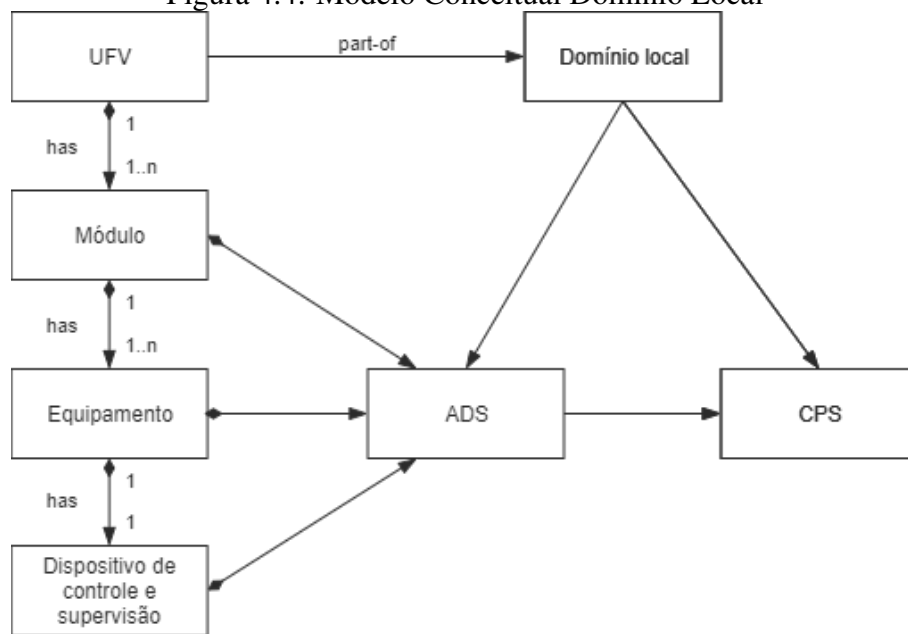
4.4 Modelagem da Aplicação

Primeiramente, no processo de modelagem, a arquitetura do modelo de integração foi definida pelo mapeamento dos domínios, entidades e dos principais elementos da aplicação a partir do histórico de projetos reais que foram executados na prática, conforme descrito anteriormente na seção 4.2. Utilizando a notação *UML* como linguagem padrão para modelagem, acrônimo em inglês de *Unified Modeling Language* que traduzindo significa Linguagem de Modelagem Unificada, foi possível representar a estrutura do projeto em forma de modelos conceituais, diagramas de classes e sequenciais.

4.4.1 Modelo Conceitual da Planta Física

O modelo de domínio apresentado na Figura 4.4 mostra a hierarquia de entidades e seus tipos de relacionamento através de classes, onde a classe UFV é uma superclasse composta pelas subclasses Módulo, Equipamento e Dispositivo de controle e supervisão.

Figura 4.4: Modelo Conceitual Domínio Local



Fonte: Autor

Onde:

- **UFV**: Refere-se ao ambiente físico para microgeração por meio de uma usina fotovoltaica, constituída por módulos, equipamentos e dispositivos;
- **Módulo**: Refere-se à segregação de sistemas de acordo com sua característica de aplicação;
- **Equipamento**: Representa os principais equipamentos presentes no ambiente de microgeração fotovoltaica;
- **ADS**: Sistema autônomo descentralizado representa o conjunto de dispositivos conectados em rede;
- **CPS**: Sistema ciber-físico é definido por dispositivos que associam componentes com elementos computacionais, sistemas operacionais embarcados, interfaces de comunicação e dispositivos físicos, com capacidade de integrar digitalizando processos operacionais, incluindo a leitura de entradas de sensores e controle de saídas ou atuações gerenciadas por sistemas de automação.

4.4.2 Modelo de Domínio Local

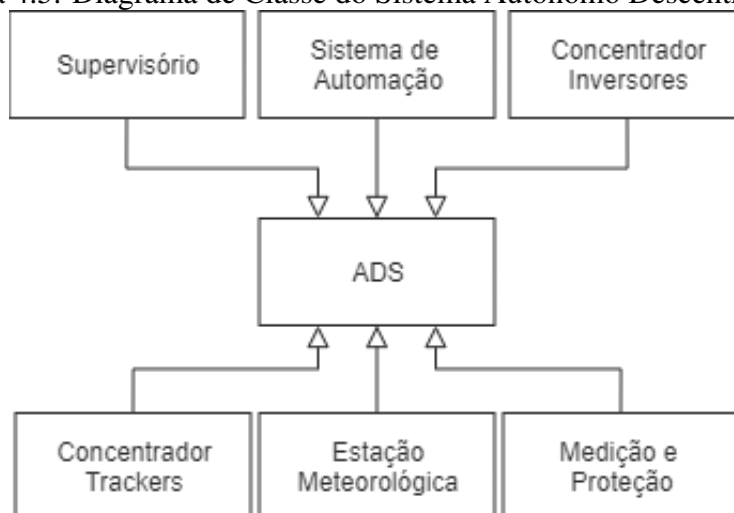
Na realização deste trabalho, a definição do escopo de configuração sobre o domínio local foi baseada a partir do histórico e também no andamento de projetos executados

no ambiente profissional do autor, utilizando informações disponibilizadas por plantas de geração solar reais.

Em síntese, UFV é uma superclasse e faz parte da classe domínio local que representa, em nível conceitual, o modelo de domínio de projeto independente de paradigma e tecnologia. Além disso, as três subclasses: módulo; equipamento e dispositivo de controle e supervisão compõem a classe ADS que está associada à classe domínio local, assim como a classe CPS.

Desta forma, a classe ADS é vinculada ao conjunto dos equipamentos que são essenciais no ambiente de monitoramento e controle dentro da UFV, suportando a subclasse CPS que refere-se à principal entidade mapeada dentro do modelo de integração proposto, detalhada a seguir.

Figura 4.5: Diagrama de Classe do Sistema Autônomo Descentralizado



Fonte: Autor

Onde:

- **Supervisório:** Refere-se ao Sistema de Supervisão, Controle e Aquisição de Dados constituído por uma plataforma de software dedicada, executada em dispositivos computacionais que possuem recursos funcionais para processamento e armazenamento de dados do processo;
- **Sistema de Automação:** Refere-se ao sistema de automação, formado por hardwares dedicados, como controlador lógico programável (CLP), com capacidade de processamento de lógicas de operação, interfaces de comunicação em rede ethernet, assim como acopladores remotos descentralizados para leitura e atuação de elementos físicos que fazem parte do processo;
- **Concentrador Inversores:** Refere-se ao hardware utilizado para concentração de

dados dos inversores de frequência, formado por componentes computacionais, interfaces de comunicação em rede ethernet, com capacidade de coleta de dados de um conjunto de inversores de frequência utilizando o protocolo de comunicação Modbus TCP;

- **Estação Meteorológica:** Refere-se ao hardware utilizado para leitura de instrumentos típicos, tais como: termômetros, anemômetros, pluviômetros e piranômetros;
- **Medição e Proteção:** Refere-se ao hardware utilizado para proteção elétrica do ponto de conexão com a rede básica da concessionária de energia, também conhecido como IED (em inglês, *Intelligent Electronic Device*) que significa Dispositivo Eletrônico Inteligente é formado por componentes computacionais, interfaces de comunicação em rede ethernet, com capacidade de leitura de grandezas elétricas, leituras discretas, funções de proteções elétricas utilizando o protocolo de comunicação Modbus TCP.

4.4.3 Modelo de Sistema Autônomo Descentralizado

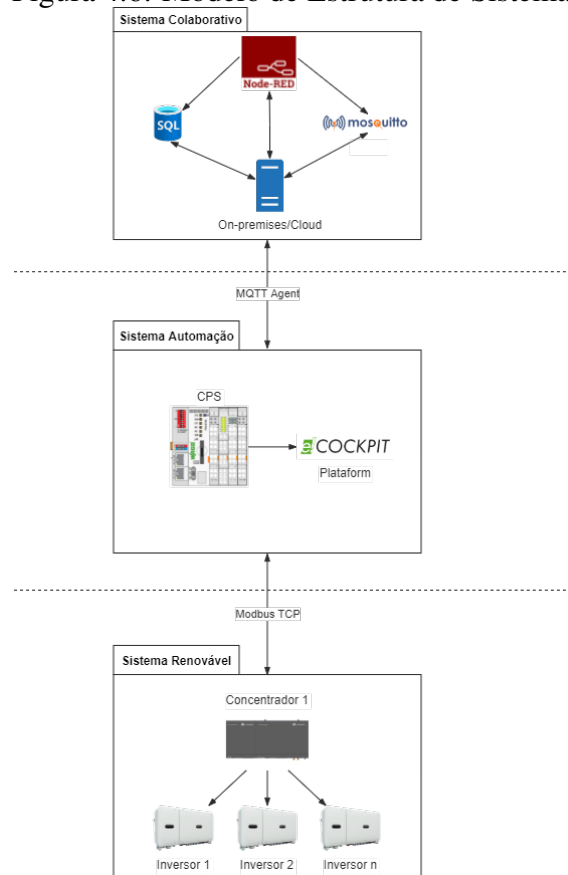
A classe ADS é composta pelo conjunto dos principais sistemas de supervisão e controle que fazem parte do ambiente de monitoramento da planta. Tipicamente, a classe denominada como Sistemas de Automação é formada por dispositivos com tecnologia de automação industrial que interagem com os módulos presentes no domínio.

4.4.4 Modelo de Dispositivo Ciber-físico

Sobretudo, o principal elemento dentro da implementação do modelo de integração proposto foi o Controlador Lógico Programável (CLP), fabricado pela empresa alemã Wago, o modelo PFC100 desempenhou o papel essencial dentro do projeto atuando como dispositivo ciber-físico (CPS) que serviu de *gateway* MQTT na publicação do conjunto de dados da planta física. Isto ocorreu devido aos seus recursos avançados que ficam embarcados no dispositivo, principalmente o sistema operacional *Linux Real Time* que atuou como um sistema multitarefa e também conhecido como *daemon running*, responsável pelo relacionamento no fluxo de dados entre os domínios ativos Outra funcionalidade importante foi o CODESYS *runtime*, sistema de código aberto para controladores programáveis baseado na norma IEC 61131-3, que juntamente com os blocos de função da bibli-

oteca *WagoAppCloud* disponibilizada pelo fabricante possibilitaram o desenvolvimento de um programa para publicação das informações coletadas pelo sistema de automação.

Figura 4.6: Modelo de Estrutura de Sistemas



Fonte: Autor

Finalmente, o modelo de estrutura de sistema se tornou factível na implementação do sistema de integração devido ao concentrador de dados dos inversores permitir múltiplas conexões utilizando o protocolo Modbus TCP, conforme ilustra a Figura 4.6. Em seguida, o CLP atuou como entidade dispositivo ciber-físico central na digitalização das informações relacionadas a planta, bem como o dispositivo concentrador de dados dos inversores que faz parte do módulo equipamento inversor e também da classe dispositivos de supervisão e controle que está diretamente relacionada a classe ADS, como já foi mencionado no tópico anterior. Assim, o modelo de integração de dados que foi estruturado a partir das classes que fazem parte de domínios baseados em plantas solares também permite a experimentação de forma prática, além de servir para outros sistemas de energia, como por exemplo, subestações de energia.

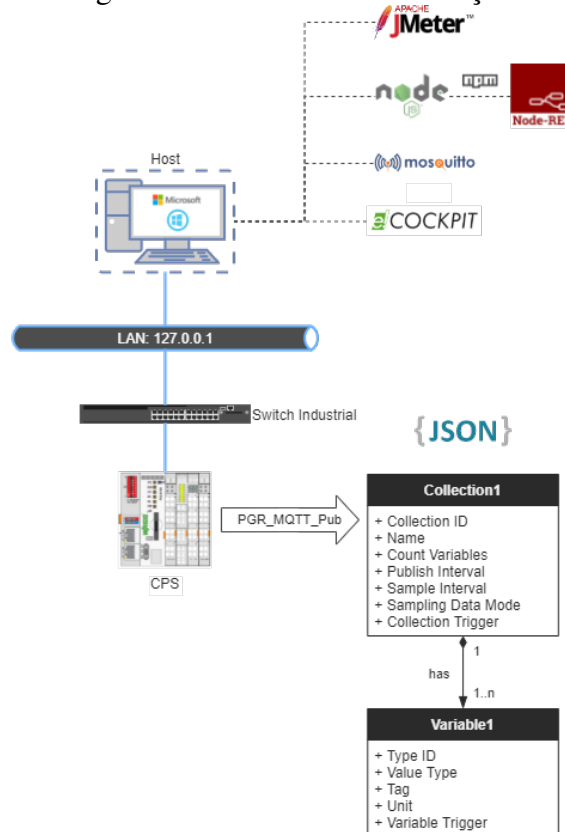
5 IMPLEMENTAÇÃO E AVALIAÇÃO DA APLICAÇÃO

Neste capítulo as atividades realizadas para implementação e avaliação experimental do modelo de integração sugerido são detalhadas, abordando sobre os materiais e os métodos utilizados para avaliação sobre o desempenho do modelo experimental, assim como as principais configurações e características técnicas para execução do sistema

5.1 Execução do Modelo

Após ao mapeamento e a modelagem sobre o domínio da aplicação, a etapa de execução da aplicação aborda os detalhes das atividades que foram realizadas para funcionamento do sistema, com objetivo de reproduzir o cenário de implementação do modelo de integração proposto, descrevendo o passo a passo dos principais processos para configuração de hardwares, plataformas e ferramentas que auxiliaram no desenvolvimento do modelo. Entretanto, em virtude da falta de acesso ao ambiente real de domínio local, os valores utilizados para execução deste experimento foram utilizando um ambiente de simulação, adotando valores fictícios durante a execução do programa de publicação no CLP, onde foram criadas variáveis auxiliares servindo como fonte de dados das mensagens que foram publicadas em forma JSON.

Figura 5.1: Ambiente de Execução



Fonte: Autor

O ambiente de execução utilizado durante a implementação do modelo de integração proposto foi prototipado em bancada, conforme mostra a Figura 5.1, utilizando os materiais e softwares listados nas subseções a seguir:

5.1.1 Hardwares

- Uma estação de trabalho, formada por notebook de uso profissional, marca Dell, modelo Inspiron 5570, com sistema operacional Windows 10, processador Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1800 Mhz, 4 Núcleo(s), 8 Processador(es) Lógico(s), 16GB de memória física (RAM), duas unidades de disco: unidade C, marca Western Digital, modelo Modelo WDC WD10SPZX-75Z10T1 - 1TB 5.4K RPM SATA 7mm 2.5" Hard Drive e unidade D, SSD WD Green 240GB M.2 2280 SATA III 6Gb/s, WDS240G2G0B-00EPW0 - Western Digital;
- Um controlador lógico programável compacto, com duas interfaces de comunicação Ethernet (RJ45), sistema operacional Real-time Linux 3.18 (com patch de preempção RT), CPU Cortex A8; 600 MHz - ref. 750-8100.

5.1.2 Softwares

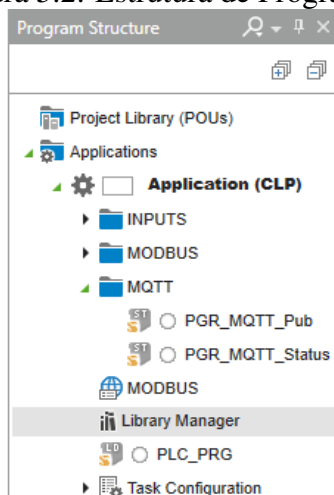
- Node-RED para windows 64 bit, versão 1.1.3, orientado a fluxos e eventos;
- Node-js para windows 64 bit, versão 12.18.4, ambiente de execução javascript;
- Plataforma de automação e!COCKPIT, utilizado em ambiente de programação integrado para configuração, programação e simulação de sistemas de automação;
- Eclipse mosquitto, versão 1.6.10 para windows 64 bit, utilizado como broker MQTT para gerenciamento de mensagens.

5.1.3 Plataforma de Automação

A plataforma de automação chamada e!COCKPIT foi o ambiente utilizado para o desenvolvimento dos programas de automação que ficaram embarcados no CLP, responsáveis pela estrutura de programas que executam as tarefas de leitura e publicação das mensagens disponíveis no concentrador de dados dos inversores que foram simuladas durante os testes. E, quanto ao modelo de integração, o projeto de automação se resume em dois principais programas:

1. **Principal:** o programa <PLC PGR> foi configurado para executar, de forma cíclica, a cada 50 ms, a tarefa de aquisição dos sinais provenientes do concentrador de dados dos inversores e chamar os programas auxiliares, via protocolo de comunicação Modbus TCP;
2. **Publicação:** o programa <PGR MQTT Pub> foi configurado para executar, de forma cíclica, a cada 200 ms, a tarefa de publicar as informações no *Broker*, em forma de coleção de objetos utilizando o formato padrão JSON no envio das mensagens.

Figura 5.2: Estrutura de Programas

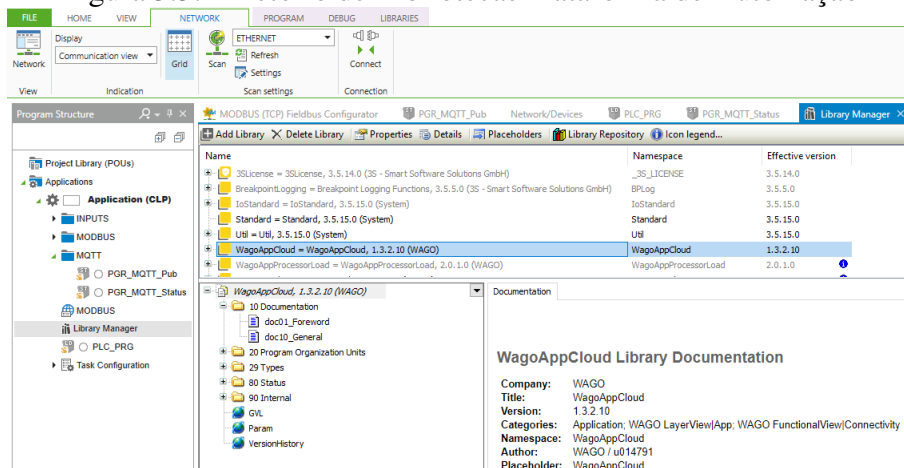


Fonte: Autor

Já a estrutura dos programas, apresentada na Figura 5.2, mostra como ficou a organização da hierarquia de todos os programas que foram desenvolvidos e que fizeram parte do sistema de automação executados pelo CLP. Os programas foram testados em ambiente de simulação, servindo como dispositivo ciber-físico e *gateway* MQTT.

Para criação do programa de publicação dos dados <PGR MQTT Pub> foi necessário adicionar no projeto a biblioteca *WagoAppCloud*, versão 1.3.2.10, disponível no repositório de bibliotecas do programa e!COCKPIT, conforme a Figura 5.3:

Figura 5.3: Diretório de Bibliotecas Plataforma de Automação



Fonte: Autor

Além disso, as variáveis do programa foram declaradas e associadas as funcionalidades da biblioteca *WagoAppCloud*, permitindo a estruturação da base em forma de coleções de dados, com atributos dedicados que foram enviados utilizando o protocolo MQTT, de forma cíclica, ao servidor intermediário *Mosquitto Broker*.

Figura 5.4: Declaração de Variáveis Programa de Publicação de Dados

```

1  PROGRAM PGR_MQTT_Pub
2
3  VAR RETAIN
4      /* Remanent variables for the sample and publish intervals */
5
6      /* Configure the sample and publish interval for collection 0 */
7      tSampleInterval0 : TIME := T#2S;
8      tPublishInterval0 : TIME := T#5S;
9      /* Configure the sample and publish interval for collection 1 */
10     tSampleIntervall : TIME := T#4S;
11     tPublishIntervall : TIME := T#10S;
12
13 END_VAR
14
15 VAR
16     /*Número de Coleções Criadas
17     aCollections: ARRAY[0..1] OF WagoAppCloud.typCollection; //de acordo com o número de coleções
18     //0 -> SL1
19     //1 -> SL2
20
21     /*Variável de cada Coleção
22     aTagConfiguration_1: ARRAY[0..1] OF WagoAppCloud.typVariableDescription; //de acordo com o número de elementos dentro da coleção
23     aTagConfiguration_2: ARRAY[0..1] OF WagoAppCloud.typVariableDescription;
24
25     /*Login para Envio das informações
26     oFbCollectionLogger : WagoAppCloud.FbCollectionLogger;
27
28     /*Simulações
29     rVar00: REAL := 0;
30     rVar01: REAL := 0;
31     rVar02: REAL := 0;
32     rVar03: REAL := 0;
33     rVar04: REAL := 0;
34     rVar05: REAL := 0;
35     rVar06: REAL := 0;
36     rVar07: REAL := 0;
37     rVar08: REAL := 0;
38     rVar09: REAL := 0;
39     rVar10: REAL := 0;
40
41 END_VAR

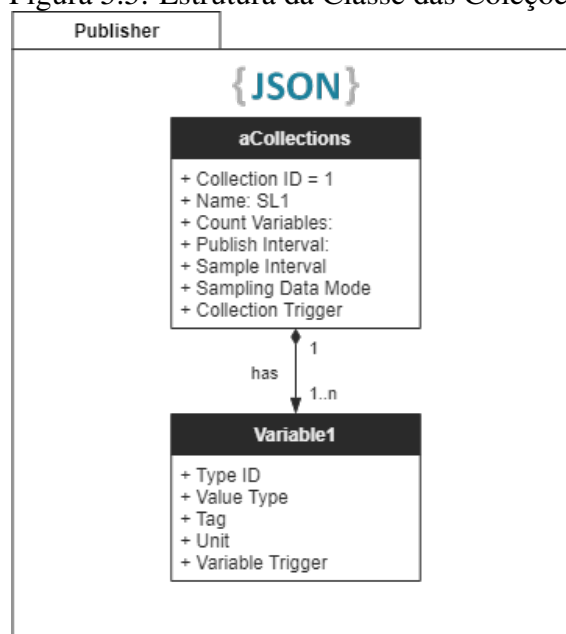
```

Fonte: Autor

Nesta condição, os concentradores de dados dos inversores foram representados pela variável do tipo array *<aCollections>*, que possibilitou a estruturação das coleções de dados durante a publicação das mensagens no *Broker*. O tempo de amostragem foi definido através da variável de tempo *<tSampleInterval>*, com aquisição a cada 2 segundos na coleção 1 e 4 segundos na coleção 2, com intervalo entre cada publicação configurados na variável retentiva *<tPublishInterval>*, sendo de 5 e 10 segundos para as coleções 1 e 2, respectivamente.

Depois, a fim de facilitar o desenvolvimento do programa em coleções separadas, foram declaradas as variáveis *<aTagConfiguration1>* e *<aTagConfiguration2>* do tipo array, que carregaram atributos relacionados aos sinais pertinentes ao processo físico, por meio de variáveis auxiliares que permitiram a simulação. Cabe ressaltar que no ambiente real seria necessário mapear os endereços lógicos no mapa de memória Modbus de cada equipamento, geralmente são fornecidos uma documentação de referência pelos fabricantes dos dispositivos de controle e supervisão.

Figura 5.5: Estrutura da Classe das Coleções



Fonte: Autor

Na Figura 5.5, mostra, em forma de estrutura de classe, como ficaram as coleções e os principais atributos que foram configurados no ambiente de simulação. Por fim, depois da estruturação do código, a variável `<oFbCollectionLogger>` foi declarada e inserida no final do programa de publicação para executar o login das coleções ao *Broker* MQTT, as propriedades de endereço e quantidade de coleções foram configuradas de acordo com o projeto, resultando na linha de código `<oFbCollectionLogger(pCollections := ADR(aCollections), dwCollectionsCount := n>`, onde “n” representa a quantidade de coleções do programa. O passo seguinte foi a publicação do conteúdo no tópico `<1.5.0/CID/TagValues>` em formato JSON, foi declarada na estrutura de variáveis do programa.

5.1.4 Broker MQTT

Em seguida, após a etapa de instalação da versão 1.6.10 no OS da estação de trabalho, o software Mosquitto desempenhou o papel de *Broker* MQTT da aplicação realizando o gerenciamento das mensagens publicadas e assinadas no tópico `<1.5.0/CID/TagValues>`. Porém, a inicialização do *Broker* precisou ser realizada pelo interpretador de linha de comando, mais conhecido como **Prompt de Comando**, sendo necessário acessar o diretório: `"C:\Program Files\mosquitto"`, usando o comando `mosquitto -v`, resultando na inicialização do sistema, com acesso pela porta TCP 1883, conforme mostra a Figura 5.6.

Figura 5.6: Inicialização do Broker

```

C:\Windows\System32\cmd.exe - mosquito -v
Microsoft Windows [versão 10.0.19042.1083]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Program Files\mosquitto>mosquitto -v
1626298368: mosquitto version 1.6.10 starting
1626298368: Using default config.
1626298368: Opening ipv6 listen socket on port 1883.
1626298368: Opening ipv4 listen socket on port 1883.

```

Fonte: Autor

Também foi necessário criar regras de segurança no *firewall* do OS, incluindo regras de entrada e saída na camada de rede para conexão à porta TCP padrão 1883, com objetivo de permitir o perfeito funcionamento da conexão entre clientes ao serviço de *Broker* executado pelo software Mosquitto. Assim, o acesso foi realizado através do navegador a página do servidor web do CLP, responsável pela publicação dos dados simulados, efetuando o login para habilitar e inserir os parâmetros de configuração na conexão 1.

Os parâmetros que foram utilizados são apresentados na Figura 5.7, considerados no ambiente de simulação os seguintes valores:

Figura 5.7: Configuração de Conexão

The screenshot shows the 'Configuration of Connection 1' window in the WAGO software. The left sidebar lists various system components, with 'Cloud Connectivity' selected. The main area displays the configuration for 'Connection 1'. Key settings include:

- Enabled:**
- Cloud platform:** MQTT AnyCloud
- Hostname:** 192.168.108.95
- Port number:** 1883
- Client ID:** UFV
- Clean session:**
- TLS:**
- Last Will:**
- User:** (empty field)
- Password:** (empty field)
- CA file:** /etc/ssl/certs/ca-certificates.crt
- Certification file:** (empty field)
- Key file:** (empty field)
- Data protocol:** WAGO Protocol 1.5
- Use compression:**
- Cache mode:** RAM (volatile)
- Device info:**
- Device status:**
- Standard commands:**

Fonte: Autor

- *Cloud platform:* MQTT AnyCloud
- *Hostname:* endereço IP da rede
- *Port number:* 1883
- *Client ID:* UFV

- *Data protocol*: WAGO Protocol 1.5

Enfim, com todos os parâmetros já configurados, o dispositivo foi reiniciado conectando-se ao *Broker* como um novo cliente UFV, como mostra a Figura 5.8, através do endereço de IP 192.168.108.17, configurado nos dados da rede no CLP.

Figura 5.8: Assinatura do Cliente MQTT

```
C:\Windows\System32\cmd.exe - mosquitto -v
Microsoft Windows [versão 10.0.19042.1083]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Program Files\mosquitto>mosquitto -v
1626306884: mosquitto version 1.6.10 starting
1626306884: Using default config.
1626306884: Opening ipv6 listen socket on port 1883.
1626306884: Opening ipv4 listen socket on port 1883.
1626306929: New connection from 192.168.108.17 on port 1883.
1626306929: New client connected from 192.168.108.17 as UFV (p2, c0, k120).
```

Fonte: Autor

Ao final, conforme a Figura 5.9, todas as informações relativas ao programa de publicação iniciaram a transmissão dos dados no tópico <1.5.0/UFV/TagValues>, enviando as mensagens com os conteúdos das coleções de origem em formato JSON.

Figura 5.9: Publicação de Dados MQTT

```
C:\Windows\System32\cmd.exe - mosquitto -v
1626306929: No will message specified.
1626306929: Sending CONNACK to UFV (0, 0)
1626306929: Received PUBLISH from UFV (d0, q1, r0, m2, '1.5.0/UFV/DeviceInfo', ... (668 bytes))
1626306929: Sending PUBACK to UFV (m2, rc0)
1626306929: Received SUBSCRIBE from UFV
1626306929: 1.5.0/UFV/Commands (QoS 1)
1626306929: UFV 1 1.5.0/UFV/Commands
1626306929: Sending SUBACK to UFV
1626306929: Received PUBLISH from UFV (d0, q1, r0, m4, '1.5.0/UFV/DeviceState', ... (393 bytes))
1626306929: Sending PUBACK to UFV (m4, rc0)
1626306929: Received PUBLISH from UFV (d0, q1, r0, m5, '1.5.0/UFV/DeviceState', ... (385 bytes))
1626306929: Sending PUBACK to UFV (m5, rc0)
1626306948: Received PUBLISH from UFV (d0, q1, r0, m6, '1.5.0/UFV/TagConfiguration', ... (692 bytes))
1626306948: Sending PUBACK to UFV (m6, rc0)
1626306948: Received PUBLISH from UFV (d0, q1, r0, m7, '1.5.0/UFV/TagValues', ... (211 bytes))
1626306948: Sending PUBACK to UFV (m7, rc0)
```

Fonte: Autor

5.1.5 Sistema Colaborativo

Na implementação do modelo de integração proposto, a ferramenta Node-RED foi utilizada como sistema colaborativo entre os domínios da aplicação, funcionando como cliente assinante do tópico <1.5.0/CID/TagValues> que foi publicado pelo CPS que simulou a conexão de um cliente na arquitetura *Pub/Sub*. Desta forma, após o tratamento dos dados através de nós funcionais, o modelo possibilitou o monitoramento das mensagens que foram publicadas através no navegador web.

Figura 5.10: Interação entre Domínios



Fonte: Autor

5.1.5.1 Configurações Iniciais

Após o processo de instalação da plataforma, a ferramenta precisou ser inicializada usando o comando de inicialização no interpretador de linha de comando do sistema operacional da estação de trabalho. Na Figura 5.11, mostra o exemplo utilizando o comando `<node-red>` no sistema operacional da *Microsoft Windows 10*, instalado na estação de trabalho.

Figura 5.11: Inicialização Node-RED

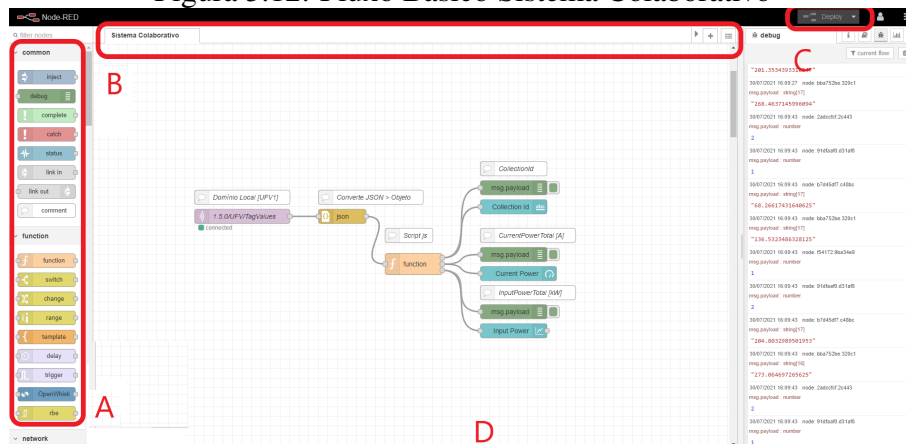
```

node-red
C:\WINDOWS\system32>node-red
30 Jul 15:43:25 - [info]
Welcome to Node-RED
=====
30 Jul 15:43:25 - [info] Node-RED version: v1.1.3
30 Jul 15:43:25 - [info] Node.js version: v12.18.4
30 Jul 15:43:25 - [info] Windows_NT 10.0.19042 x64 LE
30 Jul 15:43:27 - [info] Loading palette nodes
  
```

Fonte: Autor

Feito isto, o Node-RED pode ser acessado utilizando o navegador web, através da seguinte url: `<http://localhost:1880/>`. Desta forma, foi possível abrir a página inicial e iniciar o desenvolvimento do fluxo básico do sistema colaborativo, conforme mostra na Figura 5.12.

Figura 5.12: Fluxo Básico Sistema Colaborativo



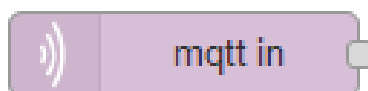
Fonte: Autor

Os detalhes básicos sobre o layout de navegação e interação de usuário com a fer-

ramenta Node-RED está dividido em quatro principais áreas, como mostra na Figura 5.12:

- A) Nós de função;
- B) Fluxos de programação;
- C) Botão para *Deploy* do projeto;
- D) Tela de conteúdo para programação utilizando os nós.

Figura 5.13: Nó de Entrada para Conexões MQTT



Fonte: Autor

Figura 5.14: Configuração do Nó de Entrada MQTT

The image shows a configuration dialog box titled 'Edit mqtt in node'. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' section with a gear icon and three utility icons (refresh, save, close). The properties are: 'Server' with a dropdown menu showing 'NodeRedClient@localhost:1883' and an edit icon; 'Topic' with a text input field containing '1.5.0/UFV/TagValues'; 'QoS' with a dropdown menu showing '1'; 'Output' with a dropdown menu showing 'auto-detect (string or buffer)'; and 'Name' with a text input field containing '1.5.0/UFV/TagValues'.

Fonte: Autor

O próximo passo foi realizar a configuração das propriedades de conexão com o servidor *Broker*, por intermédio do nó *MQTT in*, como mostram as Figuras 5.13 e 5.14.

Quanto a conexão com o servidor *Broker*, foram realizadas ordem a seguir:

1. Configurar o servidor *Broker*;
2. Inserir o tópico de interesse;
3. Selecionar o QoS.

5.1.5.2 Fluxo de Programação

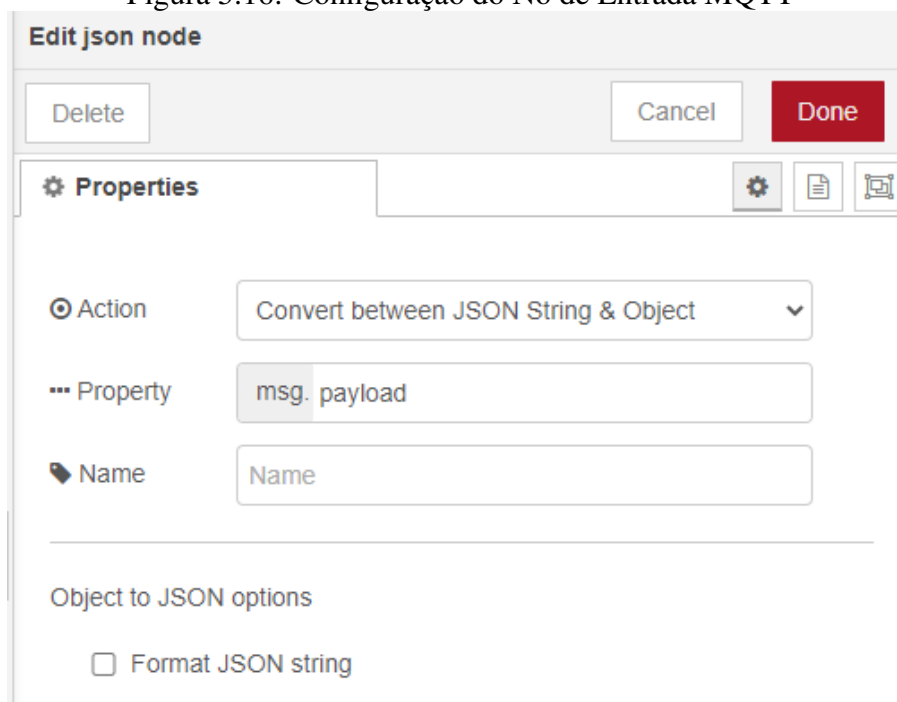
Deste modo, foi possível criar o fluxo de programação baseado na arquitetura *Pub-Sub*, por intermédio da subscrição no tópico onde as mensagens foram publicadas pelo CPS. Porém, foi necessário realizar um tratamento do *payload* porque o formato das informações publicadas estavam no padrão JSON, sendo necessário utilizar técnicas para converter o formato original de *string* para objetos.

Figura 5.15: Nó Parser JSON - Objetos



Fonte: Autor

Figura 5.16: Configuração do Nó de Entrada MQTT



Fonte: Autor

As Figuras 5.15 e 5.16 exibem o nó, assim como as propriedades que foram necessárias para análise e conversão do *payload* das mensagens para objetos, permitindo seguir no fluxo de programação.

Figura 5.17: Payload Antes e Depois do Parser

```

30/07/2021 16:09:06 node: c067e8e6.85cbd8
1.5.0/UFV/TagValues : msg.payload : string[205]
{"MessageType":"TagValues","WagoProtocol":"1.5.0","CollectionId":1,"TagData":
[{"Time":"2021-07-30T19:20:07.559+00:00","Values":
{"CurrentPowerTotal":68.26617431640625,"InputPowerTotal":136.5323486328125}}]}
30/07/2021 16:09:06 node: 5735a703.7872b8
1.5.0/UFV/TagValues : msg.payload : Object
▼ object
  MessageType: "TagValues"
  WagoProtocol: "1.5.0"
  CollectionId: 1
  ▼ TagData: array[1]
    ▼ 0: object
      Time: "2021-07-30T19:20:07.559+00:00"
      ▼ Values: object
        CurrentPowerTotal: 68.26617431640625
        InputPowerTotal: 136.5323486328125
  
```

Fonte: Autor

O resultado pode ser visto na Figura 5.17, onde:

1. *Payload* em formato JSON String;
2. *Payload* em formato de Objetos.

Figura 5.18: Nó de Função



Fonte: Autor

Figura 5.19: Nó de Função

```

Edit function node
Delete
Properties
Name
Setup Function Close
1 var obj = msg.payload;
2 Id = {"payload":obj.CollectionId};
3 Icc = {"payload":obj.TagData[0].Values.CurrentPowerTotal.toString().slice()};
4 Pi = {"payload":obj.TagData[0].Values.InputPowerTotal.toString().slice()};
5 return [Id, Icc, Pi];
  
```

Fonte: Autor

E quanto ao tratamento dos dados, a fim de selecionar apenas as variáveis simuladas na plataforma de automação, conforme já mencionado no item 5.1.3, foi utilizado

como controle o nó de função permitindo o desenvolvimento de *script* utilizando a linguagem de programação *javascript*, conforme mostram as Figuras 5.18 e 5.19. Os métodos `<toString()>` e `<slice()>` auxiliaram na seleção dos valores enviados em virtude da coleção de objetos do formato JSON.

5.1.5.3 Interface de Visualização

Enfim, finalmente foi possível concluir o fluxo de programação denominado como "Sistema Colaborativo" utilizando o conjunto de nós de controle para representação da UI (em inglês, *User Interface* ou Interface de Usuário). Nesta etapa os nós utilizados são para amostragem básica dos dados que foram obtidos mediante a simulação de valores para as variáveis do modelo de integração proposto, como mostra a Tabela 5.1.

Tabela 5.1: Variáveis Simuladas

Tag	Descrição
<i>CollectionId</i>	Número de identificação da coleção de objetos
<i>InputPower</i>	Potência de Entrada Total gerada pelo módulo dos inversores
<i>CurrentPower</i>	Corrente Contínua Total gerada pelo módulo dos inversores

Fonte: Autor

Lembrando que os valores obtidos foram para exemplificar o contexto de aplicação do modelo de integração proposto. Embora, as etapas sejam as mesmas para o ambiente real de execução.

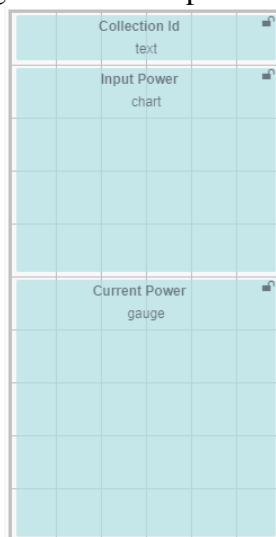
Figura 5.20: Conjunto de Nós UI



Fonte: Autor

O conjunto de nós de UI utilizados podem ser visualizados na Figura 5.20. Optou-se por apresentar os dados de forma simplificada, mas os recursos oferecidos pela ferramenta proporciona uma ampla flexibilidade que pode ser explorada em trabalhos futuros.

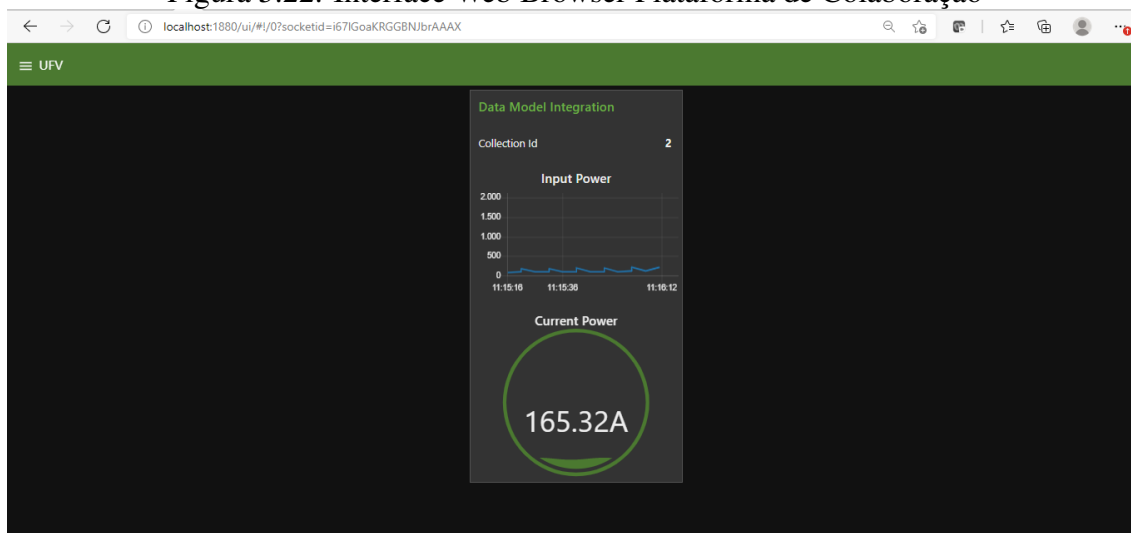
Figura 5.21: Template de UI



Fonte: Autor

Uma vantagem na etapa de criação do fluxo de programação utilizando o conjunto de nós de interface é que a ferramenta gera um template de UI automaticamente, de acordo com a Figura 5.21, o que facilita o trabalho do time de desenvolvimento e otimiza o tempo para ajustes no *front-end* da aplicação. Além disso, o sistema é responsivo, adaptando-se a qualquer tamanho de tela para visualização final.

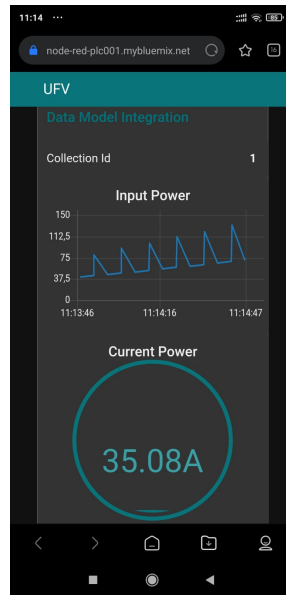
Figura 5.22: Interface Web Browser Plataforma de Colaboração



Fonte: Autor

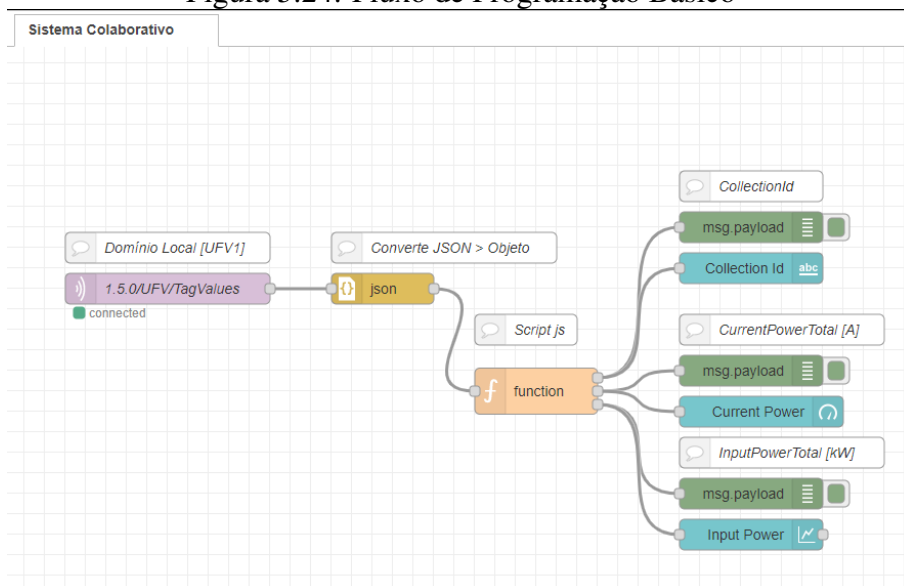
Na Figura 5.22 é possível visualizar as informações que foram simuladas. Este acesso foi realizado na máquina local dentro do ambiente de execução, através da URL: <http://localhost:1880/flow/f47a7bb6.527748>.

Figura 5.23: Interface Web Mobile Plataforma de Colaboração



Fonte: Autor

Figura 5.24: Fluxo de Programação Básico

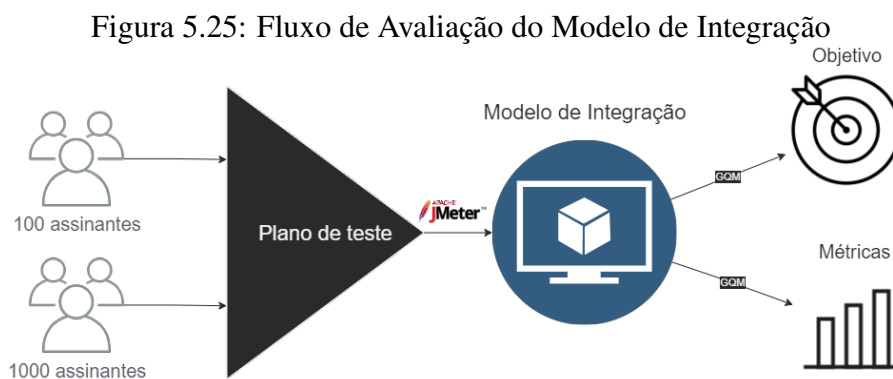


Fonte: Autor

Por último, a Figura 5.23 exibe a interface de usuário através do acesso de um *smartphone*, usando a plataforma Node-RED por meio de serviço disponível na nuvem da IBM. Dessa forma, foi possível importar o fluxo de programação básico desenvolvido em ambiente local, apresentado na Figura 5.24, com objetivo de permitir o acesso pela internet pelo link: <<https://node-red-plc001.mybluemix.net/ui/>>.

5.1.6 Plano de Testes e Análise de Resultados

Para avaliação do modelo de integração proposto, foram estruturados quatro planos de testes sobre o desempenho do servidor Mosquitto *Broker* MQTT utilizado, por intermédio da ferramenta *Apache JMeter*¹, juntamente com um *plugin* MQTT², conforme mostra a Figura 5.25. O principal objetivo foi prever o comportamento do modelo *Pub/Sub* para uma possível ampliação no número de domínios descentralizados, onde a coleta de dados seria centralizada a partir das mensagens publicadas por dispositivos ciber-físicos (CPS) nos tópicos gerenciados pelo *Broker*, que representariam clientes assinantes na infraestrutura que propõe o modelo *Pub-Sub*.



Fonte: Autor

As configurações para conexão com o servidor Mosquitto *Broker* MQTT foram realizadas de acordo com a Figura 5.26.

Figura 5.26: Configuração MQTT Connect

Fonte: Autor

¹ Software para teste de performance - <<https://jmeter.apache.org/>> - acessado em 13/07/2021

² Plugin xmeter-net/mqtt-jmeter - <<https://github.com/xmeter-net/mqtt-jmeter>> - acessado em: 13/07/2021

Figura 5.27: Configuração MQTT Sub

Fonte: Autor

Da mesma forma, o valor de QoS 1, assim como o tópic de subscrição dos dados e o parâmetro de recebimento de pelo menos uma resposta foram definidos de acordo com a Figura 5.27.

Em síntese, a sequência de etapas em relação a configuração dos planos de testes, realizados na ferramenta JMeter, foram as seguintes:

1. Criar um plano de teste;
2. Criar um grupo de usuários;
3. Configurar as propriedades do grupo de usuários;
4. Criar um controle para que o teste fosse realizado apenas uma vez;
5. Inserir ações para Conexões com o servidor *Broker* MQTT;
6. Inserir ações para Subscrição no tópic de interesse;
7. Inserir ações para Desconexão com o servidor *Broker* MQTT;
8. Inserir ações para Ouvintes, permitindo a coleta dos resultados.

Para elaboração deste plano de medição, foi adotado o contexto de abordagem baseado no método GQM (em inglês, *Goal, Question and Metric*), amplamente utilizado para mensurar o desenvolvimento de software. Há o interesse em assegurar a confiabilidade no funcionamento do modelo de integração proposto, classificando os resultados obtidos nos testes em três classes: totalmente satisfatórios, parcialmente satisfatório ou insatisfatórios de acordo com o objetivo, questões e métricas definidos na Tabela 5.2:

Tabela 5.2: Metodologia GQM ao modelo de integração proposto:

Objetivo:	Objeto de Estudo:	<i>Teste de performance do servidor broker</i>
	Propósito:	<i>Avaliar</i>
	Enfoque:	<i>Confiabilidade</i>
	Ponto de vista:	<i>Desenvolvedor</i>
	Contexto:	<i>Modelo de integração proposto no trabalho</i>
Questão:	Q1:	<i>Qual percentagem de erro no processo de conexão, subscrição no tópic de interesse e desconexão entre clientes assinantes e o servidor broker?</i>
	Q2:	<i>Qual é o tempo médio para conexão entre clientes assinantes com o servidor broker?</i>
	Q3:	<i>Qual é a quantidade de dados transferidos entre clientes assinantes com o servidor broker?</i>
Métrica:	M1:	<i>Percentual de erros no plano de teste (%)</i>
	M2:	<i>Tempo médio de conexão MQTT (ms)</i>
	M3:	<i>Throughput (kbps)</i>

Fonte: Autor

O objetivo do plano de medição é: Analisar o teste de performance do servidor *Broker* com a finalidade de avaliar com respeito a confiabilidade do ponto de vista de desenvolvedor no seguinte contexto modelo de integração proposto no trabalho.

Quanto a classificação dos resultados de acordo com a métricas, foi utilizado:

- **Totalmente Satisfatório:** $M1 \leq 10\%$; $M2 < 3000$ ms e $M3 > 33$ kbps;
- **Parcialmente Satisfatório:** $15\% \leq M1 < 20\%$; $M2 > 3000$ ms e $M3 < 20$ kbps;
- **Insatisfatório:** $M1 \geq 20\%$; $M2 > 5000$ ms e $M3 < 20$ kbps.

5.1.6.1 Plano de Teste 1

A cobertura deste plano de teste 1 contemplou um grupo de 100 usuários que representaram os acessos de clientes assinantes simulando a conexão de novos domínios distribuídos, a inscrição no tópico $\langle 1.5.0/CID/TagValues \rangle$ e a desconexão do servidor Mosquitto *Broker* MQTT, conforme mostra a Figura 5.28.

Figura 5.28: Propriedades Grupo de Usuário Plano de Teste 1

Fonte: Autor

Propriedades configuradas:

Usuários : 100

Rampa de subida : 1 segundo

Interação de usuário : 1 por vez

5.1.6.2 Plano de Teste 2

Já na cobertura deste plano de teste 2 também contemplou um grupo de 100, no entanto, com adequações nas propriedades, conforme mostra a Figura 5.29.

Figura 5.29: Propriedades Grupo de Usuário Plano de Teste 2

The screenshot shows the 'Thread Group' configuration window. The 'Name' field is '100 Subscribers - Test Plan 2'. The 'Comments' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users): 100', 'Ramp-up period (seconds): 1', 'Loop Count: Infinite', ' Same user on each iteration', ' Delay Thread creation until needed', ' Specify Thread lifetime', 'Duration (seconds): 5', and 'Startup delay (seconds): 2'.

Fonte: Autor

Propriedades configuradas:

Usuários : 100

Rampa de subida : 1 segundo

Interação de usuário : Infinitas

Ciclo de duração : 5 segundos

Delay : 2 segundos

5.1.6.3 Plano de Teste 3

Em seguida, a fim de avaliar um número maior de usuários, na cobertura deste plano de teste 3 aumentou-se para um grupo de 1000 clientes, no entanto, com adequações nas propriedades, conforme mostra a Figura 5.30.

Figura 5.30: Propriedades Grupo de Usuário Plano de Teste 3

The screenshot shows the 'Thread Group' configuration window. The 'Name' field is '1000 Subscribers - Test Plan 3'. The 'Comments' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users): 1000', 'Ramp-up period (seconds): 2', 'Loop Count: Infinite 1', ' Same user on each iteration', ' Delay Thread creation until needed', ' Specify Thread lifetime', 'Duration (seconds): 10', and 'Startup delay (seconds): 2'.

Fonte: Autor

Propriedades configuradas:

Usuários : 1000

Rampa de subida : 2 segundos

Interação de usuário : 1 por vez

5.1.6.4 Plano de Teste 4

Por último, repeti-se o número de 1000 usuários e propriedades similares ao plano de teste 2 5.1.6.2, mas com alterações conforme mostra a Figura 5.31.

Figura 5.31: Propriedades Grupo de Usuário Plano de Teste 4

Fonte: Autor

Propriedades configuradas:

Usuários : 1000

Rampa de subida : 2 segundos

Interação de usuário : Infinitas

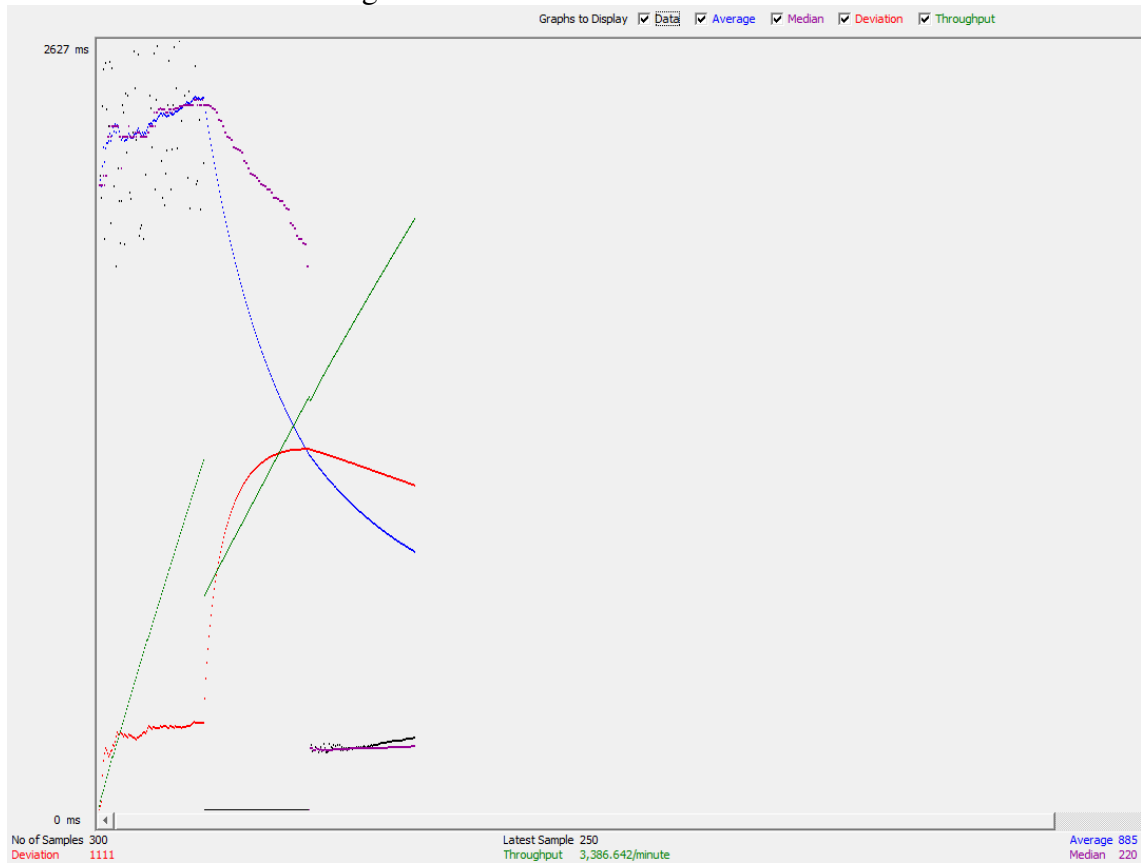
Ciclo de duração : 5 segundos

Delay : 4 segundos

5.1.6.5 Análise dos Resultados

Por fim, os resultados obtidos durante a execução dos testes de performance simulando a conexão, inscrição e a desconexão de grupos de usuários clientes com o servidor Mosquitto *Broker* MQTT, com QoS 1 (pelo menos uma entrega), podem ser vistos nas Figuras: 5.32, 5.33, 5.34 e 5.35, assim como resumidos em Tabelas: 5.3, 5.4, 5.5 e , 5.6.

Figura 5.32: Gráfico Plano de Teste 1



Fonte: Autor

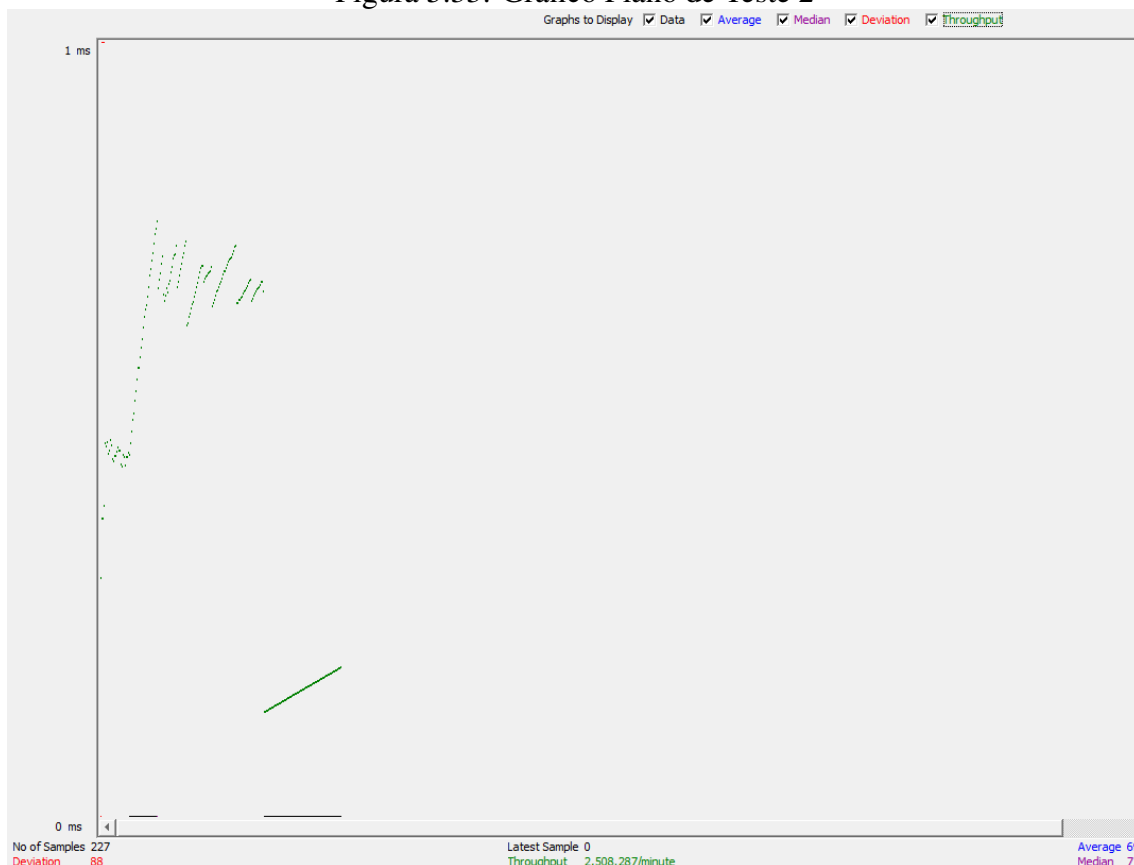
Tabela 5.3: Resumo Desempenho Servidor Broker - Plano de Teste 1

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput/sec	Received KB/sec	Sent KB/sec
100 Subscribers - Test Plan 1:MQTT Connect	100	2433	2410	2853	2921	2969	1858	2974	0.0	33.42	0.35	0.0
100 Subscribers - Test Plan 1:MQTT Sub Sampler	100	0	0	0	0	0	0	0	0.0	649.35	293.60	0.0
100 Subscribers - Test Plan 1:MQTT Disconnect	100	223	220	244	245	249	199	250	0.0	248.13	2.66	0.0
TOTAL	300	885	220	2627	2790	2960	0	2974	0.0	56.44	8.91	0.0

Fonte: Autor

Com resultados satisfatórios na cobertura do plano de teste 1, descrito no item 5.1.6.1, foi possível perceber que o valor de *throughput* ficou em torno de 3,386.642/min para todas as ações previstas. Além disso, os tempos mínimo e máximo para conexão ficaram em torno de 1858 e 2974 ms, viabilizando o uso do *broker* para grupos de até 100 usuários clientes conectados.

Figura 5.33: Gráfico Plano de Teste 2



Fonte: Autor

Tabela 5.4: Resumo Desempenho Servidor Broker - Plano de Teste 2

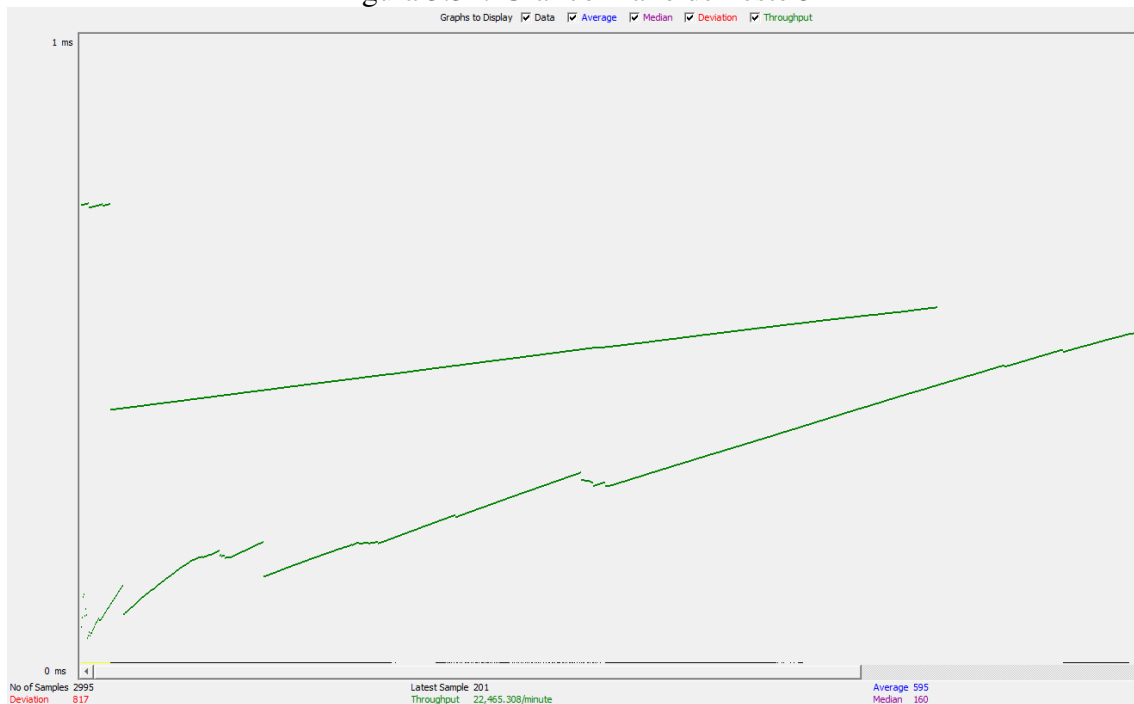
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput/sec	Received KB/sec	Sent KB/sec
100 Subscribers - Test Plan 2:MQTT Connect	100	117	111	253	295	331	4	346	0.0	95.23	1.02	0.0
100 Subscribers - Test Plan 2:MQTT Sub Sampler	100	0	0	0	0	0	0	0	0.0	19.40	8.11	0.0
100 Subscribers - Test Plan 2:MQTT DisConnect	27	145	142	182	187	193	93	193	0.0	118.94	1.27	0.0
TOTAL	227	69	7	187	249	318	0	346	0.0	41.80	7.95	0.0

Fonte: Autor

Na cobertura do plano de teste 2, descrito no item 5.1.6.2, os resultados também foram satisfatórios onde o valor de *throughput* diminuiu para 2,508.287/min para todas as ações previstas. Da mesma forma, os tempos de conexão reduziram para 4 ms de tempo mínimo e 346 ms para o maior tempo de conexão com o cliente, viabilizando o uso do *broker* para grupos de até 100 usuários clientes conectados.

Entretanto, as ações de desconexão de cerca de 73 clientes não se comportaram de acordo com o esperado, pois não conseguiram completar a desconexão com o servidor Mosquitto *Broker* MQTT. Mesmo assim, não foram reportados erros durante o teste o que tornou satisfatório o desempenho do serviço de mensagens para o cenário previsto.

Figura 5.34: Gráfico Plano de Teste 3



Fonte: Autor

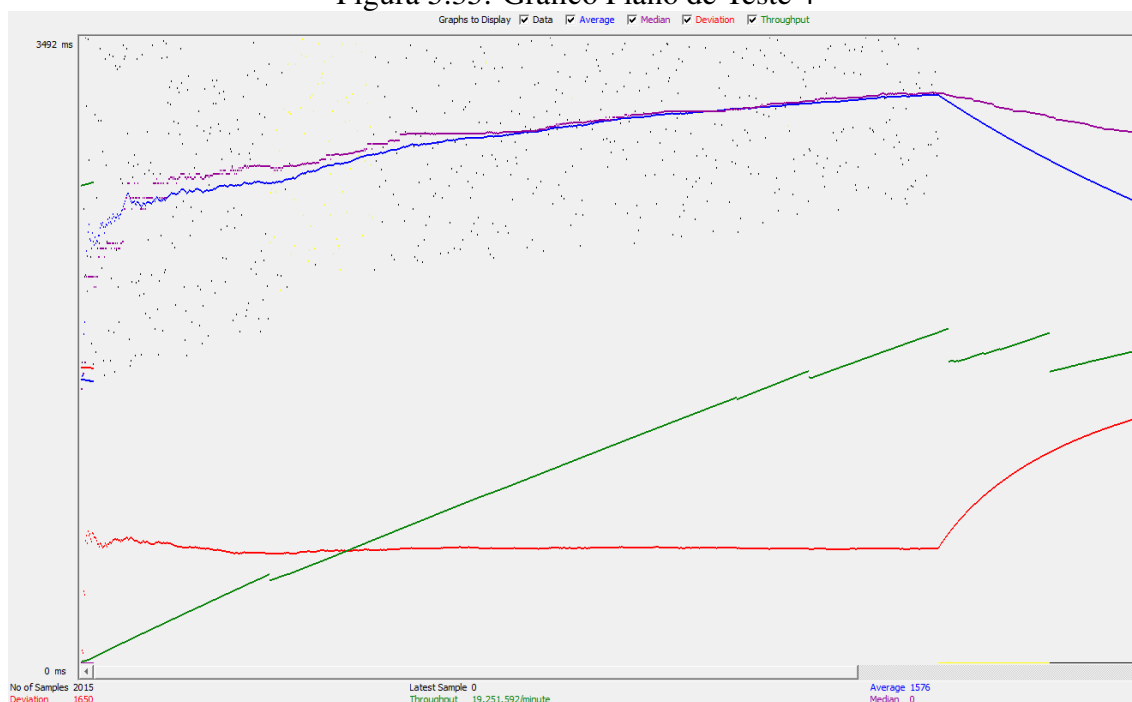
Tabela 5.5: Resumo Desempenho Servidor Broker - Plano de Teste 3

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput/sec	Received KB/sec	Sent KB/sec
1000 Subscribers - Test Plan 3:MQTT Connect	1000	1638	1757	2323	2358	2543	26	2550	0.188	262.74	4.94	0.0
1000 Subscribers - Test Plan 3:MQTT Sub Sampler	998	0	0	0	0	0	0	2	0.18	186.08	65.27	0.0
1000 Subscribers - Test Plan 3:MQTT DisConnect	997	146	161	236	244	267	0	293	0.18	211.18	2.96	0.0
TOTAL	2995	595	160	2121	2298	2421	0	2550	0.18	374.42	47.87	0.0

Fonte: Autor

No caso dos resultados na cobertura do plano de teste 3, conforme foi descrito no item 5.1.6.3, o resultado foi parcialmente satisfatório devido ao erro de 18% em todas as três ações para conexão, subscrição e desconexão. Ainda assim, o valor do *throughput* representou cerca de 262,7/seg para um grupo total de 1000 usuários, com 26 e 2550 ms para os tempos mínimo e máximo de conexão. E quanto a subscrição no tópico cerca de 998 conseguiram executar, o que representou um recebimento de 65,28 KB/sec viabilizando o uso do *broker* para grupos de até 1000 usuários clientes conectados.

Figura 5.35: Gráfico Plano de Teste 4



Fonte: Autor

Tabela 5.6: Resumo Desempenho Servidor Broker - Plano de Teste 4

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput/sec	Received KB/sec	Sent KB/sec
1000 Subscribers - Test Plan 4:MQTT Sub Sampler	1000	0	0	0	0	0	0	1	0.115	435.91	177.12	0.0
1000 Subscribers - Test Plan 4:MQTT Disconnect	15	0	0	0	0	8	0	8	1.0	145.63	4.12	0.0
1000 Subscribers - Test Plan 4:MQTT Connect	1000	3175	3185	4048	4165	4309	1531	4516	0.11	220.75	3.46	0.0
TOTAL	2015	1576	0	3789	4047	4258	0	4516	0.12	320.85	67.26	0.0

Fonte: Autor

A cobertura do plano de teste 4, conforme foi descrito no item 5.1.6.4, também apresentou um resultado satisfatório, entretanto, de forma parcialmente mediante ao erro global para as ações de conexão, subscrição e desconexão, ficando em torno de 12%. Mesmo assim, o valor do *throughput* nas ações de conexão e subscrição representaram cerca de 220,8/seg e 435,9/sec, respectivamente, para um grupo total de 1000 usuários, com tempos mínimo e máximo entre 1531 ms e 4516 ms para estabelecer conexão entre clientes e servidor.

Tabela 5.7: Resumo dos Resultados sobre o Desempenho do Servidor Mosquito Broker

Test Plan	Users	Ramp-up period (sec)	Interaction	Specify Thread lifetime	Duration (sec)	Startup delay (sec)	Result	Error %
1	100	1	1	False	0	0	Totalmente Satisfatório	0
2	100	1	Infinite	True	10	2	Totalmente Satisfatório	0
3	1000	1	1	False	0	0	Parcialmente Satisfatório	18,86
4	1000	1	Infinite	True	5	1	Parcialmente Satisfatório	12,16

Fonte: Autor

Em resumo, a Tabela 5.7 indica os principais parâmetros que foram configurados em cada grupo de usuários incluídos nos planos de testes realizados. Além disso, também

são mostrados a relação de classificação dos resultados, assim como a porcentagem de erro para cada cenário.

5.1.6.6 Limitações

Em especial, cabe ressaltar sobre o aspecto que algumas limitações implicam na aplicação da metodologia abordada para execução dos planos testes que foram apresentados na seção 5.1.6.5 - Análise dos Resultados. Logo, os resultados também podem sofrer influências em decorrências das limitações descritas a seguir:

1. Infraestrutura de rede em ambiente experimental, como mostra a Figura 5.1;
2. Capacidade computacional oferecida pela estação de trabalho, conforme especificação do item 5.1.1 - Hardwares;
3. Valores utilizados nos parâmetros que foram adotados para os grupos de clientes assinantes, configurados de acordo com cada plano de teste.

6 CONCLUSÃO

Neste trabalho, foi pesquisado sobre os principais temas que envolvem a utilização de tecnologias digitais para integração de dados, que foram abordados de forma detalhada no Capítulo 2 com a estruturação do modelo baseado em contextos e fundamentos teóricos mais relevantes sobre a evolução das organizações para convergência e interoperabilidade entre sistemas e aplicações.

Quanto ao conteúdo que foi apresentado no Capítulo 3, onde contempla os principais trabalhos pesquisados que são relacionados com a implementação de modelos semelhantes para integração de sistemas a partir da tecnologia de IoT, bem como no uso do protocolo MQTT em padrões de comunicação *Pub/Sub*, apresentando seções com a fundamentação de cada trabalho relacionado, assim como as principais vantagens e desvantagens em relação às arquiteturas que foram propostas pelos autores e, por fim, uma análise do ponto de vista crítico em relação a cada sistema sugerido.

Em relação ao Capítulo 4, o mesmo faz referência às etapas de modelagem sobre os domínios da aplicação. Isto envolve a composição das classes que foram identificadas e estruturadas no modelo de integração de dados, onde são tipicamente formadas por domínios locais associados a fontes renováveis de energia por meio de usinas fotovoltaicas (UFV), além de sistemas autônomos descentralizados (ADS) constituídos por sistemas ciber-físicos (CPS) chegando até a camada de interação com o processo físico efetuado por concentradores dedicados através das tecnologias operacionais (OT) geralmente utilizadas.

No Capítulo 3, foi possível experimentar o modelo proposto neste trabalho em ambiente de execução simulado em bancada. O uso de recursos em hardware, como, por exemplo, *switch* industrial e controlador lógico programável (CLP), assim como softwares livres e ferramentas necessárias para avaliação da arquitetura conforme o padrão *Pub/Sub*, sendo essenciais para as etapas de desenvolvimento, prototipagem, validação e testes no funcionamento dos programas embarcados nos dispositivos e no uso do Node-RED como plataforma de colaboração que apresentou vantagens em relação à programação orientada a fluxos e eventos.

A principal contribuição do trabalho diz respeito ao desenvolvimento de um modelo de integração de dados que permite o monitoramento de sistemas descentralizados utilizando os conceitos da Indústria 4.0 para comunicação entre domínios distribuídos. O modelo pode ser utilizado como base para aplicações através da tecnologia de IoT conec-

tando elementos ciber-físicos locais à plataformas de colaboração através do protocolo MQTT, onde os dados poderão ser interpretados para melhorar a tomada de decisões estratégicas em relação ao processo operacional. Contudo, embora o modelo de integração apresente um baixo custo quando comparado a sistemas legados oferecidos pelos líderes de mercado no seguimento, incluindo: IBM; Microsoft, AWS, entre outras, é importante avaliar os custos de manutenção e os recursos utilizados na implantação.

Para se obter mais assertividade na avaliação sobre o desempenho para implementação do modelo de integração de dados sugerido em ambientes operacionais que estejam em execução, é importante efetuar novos testes com uso de infraestrutura de rede onde seja possível ter acesso à internet. Além disso, os resultados que foram apresentados no item 5.1.6.5 foram em cenários hipotéticos, onde os tempos configurados podem sofrer impactos positivos ou negativos.

Por fim, como possibilidades de trabalhos futuros ao modelo de integração de dados, destacando os seguintes temas:

1. Estudo de comparação de armazenamento das informações em bancos de dados base relacionais ou não-relacionais;
2. Aperfeiçoamento de algoritmos para tratamento e manipulação dos dados;
3. Viabilidade de aplicação de sistema de recomendação baseado em aprendizado de máquina para fontes de energia renováveis.

REFERÊNCIAS

- BLAZY, O. et al. An IoT Attribute-Based Security Framework for Topic-based Publish/Subscribe Systems. **IEEE Access**, p. 1–13, 2021. ISSN 21693536.
- COISAS, C. I. e. **Protocolo MQTT**. 2019. Accessed: 2021-07-28. Available from Internet: <https://youtube.com/playlist?list=PLMmiQibT0iTblNNF_y6_xZfvid5LcWK6_>.
- FIROUZI, F. et al. **Intelligent Internet of Things**. [S.l.: s.n.], 2020. ISBN 9783030303662.
- GARIMELLA, P. K. It-ot integration challenges in utilities. In: **2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)**. [S.l.: s.n.], 2018. p. 199–204.
- HEYVAERT, P. et al. **SAD Generator: Eating Our Own Dog Food to Generate KGs and Websites for Academic Events**. [S.l.: s.n.], 2019. 95–99 p. ISSN 16113349. ISBN 9783030323264.
- HIVEMQ. MQTT & MQTT 5 Essentials. In: . 1. ed. [s.n.], 2020. p. 1–72. Available from Internet: <<https://www.hivemq.com/mqtt-essentials/>>.
- JUNIOR, C. S. S. G. ARQUITETURA ATIIOT: INTEGRANDO TECNOLOGIAS ASSISTIVAS COM INTERNET DAS COISAS. 2019.
- KAWAGUCHI, R.; BANDAI, M. A distributed mqtt broker system for location-based iot applications. In: **2019 IEEE International Conference on Consumer Electronics (ICCE)**. [S.l.: s.n.], 2019. p. 1–4.
- KG, W. K. G. . C. **WAGO-I / O-SYSTEM 750 WagoAppCloud**. Minden, Germany: [s.n.], 2020. 63 p. Available from Internet: <<https://www.wago.com/global/d/15719>>.
- KRAVETS, A. G.; BOLSHAKOV, A. A. **Cyber-Physical Systems : Design and Application**. [S.l.: s.n.], 2021. 1–440 p. ISBN 9783030660802.
- KUNIFUJI, T. A proposal of autonomous decentralized software architecture to realize gradual and short-term extension in safety related system. **Proceedings - 2013 11th International Symposium on Autonomous Decentralized Systems, ISADS 2013**, IEEE, 2013.
- Landeta Echeberria, A. **A Digital Framework for Industry 4.0**. [S.l.: s.n.], 2020. ISBN 9783030600488.
- LEA, P. **Internet of Things for Architects**. [S.l.: s.n.], 2018. 515 p. ISBN 9781788470599.
- LI, J. et al. Study on the Reference Architecture and Assessment Framework of Industrial Internet Platform. **IEEE Access**, v. 8, n. Iic, p. 164950–164971, 2020. ISSN 2169-3536.
- MATT, D. T.; MODRÁK, V. **Implementing Industry 4.0 in SMEs**. [S.l.: s.n.], 2021. ISBN 9783030705152.

MEHMOOD, M. U. et al. A Low Cost Internet of Things (LCIoT) Based System for Monitoring and Control of UPS System using Node-Red, CloudMQTT and IBM Bluemix. **1st International Conference on Electrical, Communication and Computer Engineering, ICECCE 2019**, IEEE, n. July, p. 24–25, 2019.

NAYYAR, A.; KUMAR, A. **A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development**. [s.n.], 2020. 1–213 p. ISBN 9783030145446. Available from Internet: <https://books.google.com.mx/books?id=nr_ADwAAQBAJ>.

NODE-RED. **História sobre Node-RED**. 2021. Accessed: 30-07-2021. Available from Internet: <<https://nodered.org/about/>>.

OASIS. **MQTT Version 3.1.1 Plus Errata 01**. 2015. 1–81 p. Accessed: 28-07-2021. Available from Internet: <<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>>.

OASIS. MQTT Version 5.0 OASIS Standard. n. March, 2019. Available from Internet: <<https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>>.

ONUKE, J.; OLIVEIRA, A. J. de. O uso do role-play no ensino de negociações internacionais. **Revista de Graduação USP**, v. 2, n. 1, p. 45, 2017. ISSN 2525-376X.

SA, I. et al. **IEEE Standard for an Architectural**. [S.l.: s.n.], 2019. ISBN 9781504458863.

STEINMETZ, C. Uma abordagem para a integração de sistemas industriais aplicando o conceito de internet das coisas e de modelos semanticos no contexto da industria 4.0. **Universidade Federal Do Rio Grande Do Sul Instituto De Informática Programa De Pós-Graduação Em Computação Charles**, 2018. Available from Internet: <<https://lume.ufrgs.br/handle/10183/180113>>.

SWAMY, S. N.; KOTA, S. R. An empirical study on system level aspects of Internet of Things (IoT). **IEEE Access**, v. 8, p. 188082–188134, 2020. ISSN 21693536.

WANG, X.; GAO, L. **When 5G Meets**. [S.l.: s.n.], 2020. 43–74 p. ISBN 9789811567315.