
Estudo da aplicação de um sistema IoT baseado no protocolo de comunicação MQTT a área da robótica industrial

Daniel Silva de Souza



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Uberlândia
2018

Daniel Silva de Souza

**Estudo da aplicação de um sistema IoT baseado
no protocolo de comunicação MQTT a área da
robótica industrial**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciências.

Área de concentração: Engenharia Elétrica de Controle e Automação

Orientador: Fábio Vincenzi Romualdo da Silva

Coorientador: Márcio José da Cunha

Uberlândia

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

S729e
2019 Souza, Daniel Silva de, 1990-
Estudo da aplicação de um sistema IoT baseado no protocolo de
comunicação MQTT a área da robótica industrial [recurso eletrônico] /
Daniel Silva de Souza. - 2019.

Orientador: Fábio Vincenzi Romualdo da Silva.

Coorientador: Márcio José da Cunha.

Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.

Modo de acesso: Internet.

Disponível em: <http://dx.doi.org/10.14393/ufu.di.2019.316>

Inclui bibliografia.

Inclui ilustrações.

1. Engenharia elétrica. 2. Robótica. 3. Internet das coisas. I. Silva,
Fábio Vincenzi Romualdo da, 1974- (Orient.). II. Cunha, Márcio José da,
1977- (Coorient.). III. Universidade Federal de Uberlândia. Programa de
Pós-Graduação em Engenharia Elétrica. IV. Título.

CDU: 621.3

Maria Salete de Freitas Pinheiro - CRB6/1262

*Dedico este trabalho primeiramente a Deus, autor do meu destino, meu guia, socorro
presente na hora da angústia, ao meu pai Divino Ribeiro de Souza, minha mãe Ivone
Aparecida Silva Ribeiro de Souza e ao meu irmão Matheus Silva de Souza.*

Agradecimentos

Agradeço a Deus pelo dom da vida, por ter me dado saúde e inteligência para superar todas as dificuldades e conseguir chegar onde hoje estou, aos meus pais e familiares pelo apoio e carinho incondicionais. Ao professor Márcio José Cunha, pela confiança, pelas sábias orientações e pela amizade. E a todas as pessoas que contribuíram para a realização deste trabalho, o meu muito obrigado.

*Não fui eu que ordenei a você? Seja forte e corajoso! Não se apavore nem desanime,
pois o Senhor, o seu Deus, estará com você por onde você andar".*

Josué 1:9

Resumo

A Internet das Coisas é formada por uma rede de objetos que possuem tecnologia embarcada, sensores e capacidade de conexão, esses objetos são capazes de coletar e transmitir dados. Os objetos da Internet das Coisas podem ser desde uma geladeira, até carros, máquinas, computadores e *smartphones*. A tendência é que cada vez mais produtos e dispositivos sejam ligados à rede, desde que forneçam serviços que agreguem aos usuários e à sociedade. As aplicações são imensas, indo desde saúde, transporte e bem-estar à agricultura, pecuária, indústria e economia. O resultado disso é um planeta mais arguto e responsivo. Esses objetos inteligentes oferecem maiores oportunidades de interação entre os fatores que englobam um processo como um todo e economia de recursos, pois possibilita redução de consumo de energia e ajuda na prevenção de possíveis erros e falhas devido ao acesso em tempo real as informações. Buscando se adequar ao novo paradigma de conectividade e compartilhamento de recursos da Internet das Coisas, o setor Industrial já realiza diversas pesquisas com o intuito de desenvolver novas tecnologias que possam ser utilizadas no setor industrial. Sendo assim, o trabalho proposto tem como objetivo analisar a eficiência de um sistema IoT baseado no protocolo de comunicação MQTT em um cenário constituído de braço robótico, sensores e atuadores. Nesse sentido são apresentados ferramentas e softwares que apresentam certo grau de automação no processo de desenvolvimento de aplicações *Internet of Things* (IoT) na área da robótica industrial. Os resultados obtidos mostraram que para o sistema proposta, os parâmetros de performance analisados da rede foram satisfatórios, sendo capazes de serem aplicados ao ambiente industrial.

Palavras-chave: IoT, MQTT, Robótica, Indústria.

Abstract

The Internet of Things has emerged from advances in various areas such as embedded systems, microelectronics, communication and sensing, and can be understood as a network of objects that have embedded technology, sensors, and network connection, and can collect and transmit data. These objects can be clocks, refrigerators, cars, machines, computers, smartphones etc. Any gadget you can imagine can theoretically enter the Internet world of Things. The trend is for more and more products and devices to be connected to the network, if they provide uses that aggregate to users and society. The applications are immense, ranging from health, transportation and well-being to agriculture, livestock, industry and more. The result of this is a sharper and more responsive planet. These intelligent objects provide greater opportunities for interaction between the factors that encompass a process and resource savings, as it enables a reduction of energy consumption and helps to prevent possible errors and failures due to real-time access information. Seeking to adapt to the new paradigm of connectivity and sharing resources of Internet of Things, the Industrial sector already carries out several researches with the intention of developing new technologies that can be used in the industrial sector. Thus, the proposed work aims to analyze if the efficiency of an IoT system based on the MQTT communication protocol is satisfactory in a scenario consisting of a robotic arm, sensors and actuators. In this sense are presented tools and software that present some degree of automation in the process of developing applications IoT in the area of industrial robotics.

Keywords: IoT, MQTT, Robotic, Industry.

Lista de ilustrações

Figura 1 – Blocos Básicos IoT	17
Figura 2 – Arquitetura <i>IoT</i> de 3 camadas	20
Figura 3 – Arquitetura IoT de 5 camadas	21
Figura 4 – Arquitetura <i>Message Queuing Telemetry Transport</i> (MQTT)	22
Figura 6 – Cadeia Cinemática Aberta	26
Figura 7 – Cadeia Cinemática Fechada	27
Figura 8 – Diagrama da Aplicação IoT	30
Figura 9 – Aplicação a nível global e regional	31
Figura 10 – Módulos dos Sistemas de Prognóstico PHM	32
Figura 11 – <i>Framework</i> Proposto	34
Figura 12 – Representação Gráfica da Aplicação	36
Figura 13 – Representação Gráfica da Aplicação	37
Figura 14 – Algoritmo Utilizado para Medição da Latência	38
Figura 15 – Arquitetura Utilizada para Avaliação dos Protocolos	40
Figura 16 – Braço Robótico	44
Figura 17 – Ambiente ferramenta <i>Node-Red</i>	45
Figura 18 – Rede IoT utilizada	46
Figura 19 – Rede IoT do caso 1	47
Figura 20 – Rede IoT do caso 2	48
Figura 21 – Aplicação Web	49
Figura 22 – Taxa de Pacotes Transmitidos	53
Figura 23 – Latência	55

Lista de tabelas

Tabela 1 – Exemplo da tabela gerada pelo <i>subscriber</i>	50
----------------------------------------------------------------------	----

Lista de siglas

APIs *Application Programming Interface*

CSV *Comma-Separated Values*

CoAP *Constrained Application Protocol*

DSET *Dempster-Shafer Evidence Theory*

GPS *Global Positioning System*

HTTP *Hypertext Transfer Protocol*

IoT *Internet of Things*

IIoT *Industrial Internet of Things*

IP *Internet Protocol*

MQTT *Message Queuing Telemetry Transport*

NFC *Near Field Communication*

NTP *Network Time Protocol*

NB-IoT *Narrowband-IoT*

OWL *Web Ontology Language*

PHM *Prognostic Health Management*

Qos *Quality of Service*

RDF *Resource Description Framework*

RFID *Radio Frequency Identification*

RTE *Real Time Ethernet*

REST *Representational State Transfer*

RUL *Remaining Useful Life*

RTT *Round Trip Time*

SSL/TLS *Transport Layer Security/ Secure Socket Layer*

SCADA *Sistemas de Supervisão e Aquisição de Dados*

TCP *Protocolo de Controle de Transmissão*

timestamp número do tipo de dado flutuante denotando a hora ou data que certo evento ocorreu

TTP *Taxa de Transmissão de Pacotes*

UDP *User Datagram Protocol*

URL *Uniform Resource Locator*

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	14
1.2	Organização da Dissertação	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Internet das Coisas	16
2.2	Blocos Básicos da IoT	17
2.2.1	Características da IoT	18
2.3	Arquiteturas IoT	19
2.4	MQTT	21
2.5	Manipulador Robótico	24
2.6	Considerações Finais	27
3	ESTADO DA ARTE	28
3.1	Internet das Coisas e a Indústria	28
3.2	Indústria, Internet das Coisas e MQTT	29
3.3	Considerações Finais	41
4	SOLUÇÃO PROPOSTA E METODOLOGIA	43
4.1	Materiais	43
4.2	<i>Node-Red</i>	44
4.3	Metodologia	46
4.3.1	Parâmetros de Performance	51
4.4	Considerações Finais	52
5	RESULTADOS	53
5.1	Taxa de Transmissão de pacotes do publisher P1	53
5.2	Latência Média da Rede	54

6	CONCLUSÃO	57
	REFERÊNCIAS	59

Introdução

A Internet das Coisas (IoT) pode ser definida como a Internet onde o número de dispositivos conectados superou o número de pessoas, os dispositivos do ambiente da IoT podem ser sensores, atuadores, veículos, *smartphones*, eletrodomésticos, brinquedos, câmeras, equipamentos médicos, sistemas industriais, prédios e etc (GOMATHI et al., 2018). Esses dispositivos possuem endereços únicos e são capazes de interagir e cooperar entre si para a criação de novos serviços sem a necessidade de supervisão humana (VERMESAN; FRIESS, 2013). Esses objetos possuem as habilidades de coleta, processamento, análise de dados e de envio e recebimento de informações de outros objetos que façam parte da rede, essas capacidades proveem uma forma de os dispositivos interagirem e aprenderem com os outros dispositivos conectados à rede. Diversos trabalhos já são realizados visando a viabilização da aplicação da Internet das Coisas nos mais diversos setores da economia, temos estudos sendo realizados nos setores automotivo (MENDIRATTA; DEY; SONA, 2017), de agricultura de precisão (DOLCI, 2017), hospitalar (STRADOLINI et al., 2018), de rastreamento veicular (HUSNI et al., 2016), agropecuária (MAINA, 2017), segurança veicular (RUENGITTINUN; PAISALWONGCHAROEN; WATCHARAJINDASAKUL, 2017). Seguindo essa tendência o setor Industrial é outro setor da economia que busca utilizar as vantagens de acesso e controle remoto de dispositivos advindas da Internet das Coisas a sua estrutura.

Já existem trabalhos que exploram a utilização de diversas tecnologias de infraestrutura IoT, como *ZigBee*, MQTT, *Constrained Application Protocol* (CoAP) no ambiente industrial, entretanto, são poucos os trabalhos que tratam da aplicação da tecnologia MQTT aos sistemas robóticos industriais dentro do cenário da Internet das Coisas.

1.1 Objetivos

O objetivo desse trabalho é avaliar a eficiência de uma rede IoT aplicada ao monitoramento e controle de plantas robóticas. Nesse trabalho foi utilizado um sistema robótico, em escala laboratorial, constituído por um braço, sensores e atuadores. Dessa forma,

pretende-se avaliar:

1. Eficiência do sistema IoT em um plantas robóticas;
2. Benefícios do sistema proposto aplicado a robótica;
3. Análise dos parâmetros de performance da solução proposta.

1.2 Organização da Dissertação

Esta dissertação está organizada em 6 capítulos, o conteúdo dos capítulos são descritos a seguir:

1. **capítulo 1:** introdução;
2. **capítulo 2:** fundamentação teórica;
3. **capítulo 3:** estado da arte em relação ao MQTT;
4. **capítulo 4:** solução implementada;
5. **capítulo 5:** resultados;
6. **capítulo 6:** conclusão, considerações finais e perspectivas de trabalhos futuros.

Fundamentação Teórica

Neste capítulo são apresentados conceitos importantes para o bom entendimento do trabalho proposto, são abordados os conceitos de IoT, Redes Industriais, Robótica e MQTT.

2.1 Internet das Coisas

A Internet das Coisas pode ser entendida como um conceito tecnológico onde todos os objetos conectados agem de modo inteligente e sensorial (SINGH; SINGH, 2015). Estas habilidades geram um grande número de oportunidades tanto no âmbito acadêmico como industrial (SANTOS et al., 2016), pois a capacidade de conexão com a internet, torna possível o controle remoto desses dispositivos e também o seu acesso como provedores de serviços. Os objetos integrantes da Internet das Coisas são computadores, TVs, veículos, *smartphones*, eletrodomésticos, câmeras, equipamentos médicos, sistemas industriais, pessoas, animais e prédios, usando os seus recursos de capacidade computacional, de comunicação, de conectividade à Internet e sensoriamento para detectar o seu contexto, controlá-lo, viabilizar troca de informações com outros dispositivos, acessar serviços da Internet e interagir com pessoas.

A Internet das Coisas muda a forma dos modelos de negócios e a maneira de interação da sociedade com o meio a sua volta, pois os objetos físicos passam a perceber estímulos do ambiente, produzir informações e interagir com outros objetos e pessoas (LACERDA; LIMA-MARQUES, 2015). Essa nova forma de interação propiciada pela IoT representa uma solução em potencial para melhoria da vida das pessoas, visto que além de facilitar as trocas de dados entre máquinas, facilitando o acesso à informações, existe ainda a possibilidade de economia de energia, segurança, saúde, educação e outros aspectos do cotidiano. Um exemplo disso é o *smartwatch*, que monitora a saúde e ainda está conectado à nuvem. Em indústrias e em empresas que se relacionam diretamente com o cliente final, sistemas embarcados intercomunicantes tem o poder de aumentar a produtividade, criar novas estratégias de produção e conhecer melhor o mercado. A IoT representa a evolução

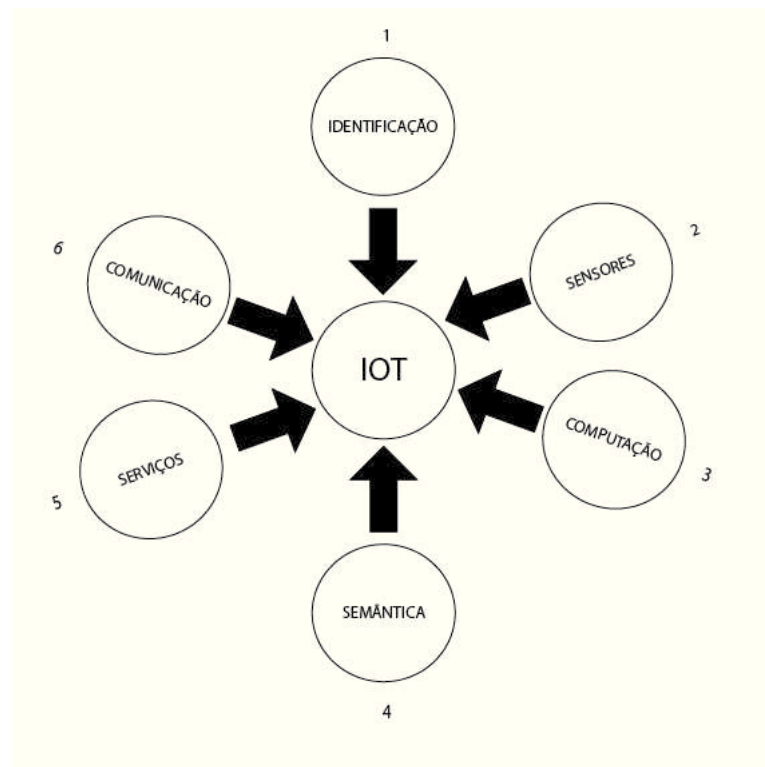
da internet, pois eles dá um grande passo na capacidade de coleta, análise e distribuição de dados que podem ser transformados em informações e conhecimento (EVANS, 2011).

No entanto, a Internet das Coisas tem um grande caminho a percorrer, pois atualmente a falta de padronização entre as tecnologias IoT existentes é um empecilho ao desenvolvimento da Internet das Coisas, pois a falta de padronização entre as tecnologias utilizadas impede que os diversos dispositivos conectados à internet possam interagir e trocar informações.

2.2 Blocos Básicos da IoT

Os blocos básicos de construção da Internet das Coisas são as tecnologias utilizadas que viabilizam todo o funcionamento da rede. A Figura 1 a seguir ilustra os blocos básicos constituintes da IoT.

Figura 1 – Blocos Básicos IoT



Fonte: Adaptado de (SANTOS et al., 2016)

Cada bloco tem a seguinte função:

1. **Identificação:** é um dos blocos mais fundamentais, pois ele é essencial para identificar os objetos unicamente para conectá-los à Internet. Tecnologias como a *Radio Frequency Identification* (RFID), *Near Field Communication* (NFC) e endereçamento *Internet Protocol* (IP) podem ser empregadas (SANTOS et al., 2016);

2. **Sensores:** obtém as informações sobre o contexto em que o objeto se encontra e em seguida armazena/encaminha esses dados para centros de armazenamento. Atuadores podem manipular o ambiente ou reagir de acordo com os dados lidos (SANTOS et al., 2016);
3. **Computação:** é responsável por executar os algoritmos locais nos objetos inteligentes (SANTOS et al., 2016);
4. **Semântica:** faz referência à habilidade de extração de conhecimento dos objetos IoT, ou seja, faz referência a capacidade de processar o conteúdo da informação existente com o objetivo de prover determinado serviço ao invés de apenas apresentá-la aos humanos. Com esse intuito podem ser utilizadas as linguagens *Resource Description Framework* (RDF), *Web Ontology Language* (OWL) entre outras (SANTOS et al., 2016);
5. **Serviços:** O bloco de serviços da IoT é responsável por prover diversas classes de serviços, dentre elas, destacam-se os serviços de identificação que são responsáveis por mapear entidades físicas, de interesse do usuário, em Entidades Virtuais como, por exemplo, a temperatura de um local físico em seu valor com as coordenadas geográficas do sensor e instante da coleta; os serviços de agregação de dados coletam e resumizam dados homogêneos ou heterogêneos obtidos dos objetos inteligentes; os serviços de colaboração e inteligência agem sobre os serviços de agregação de dados para tomar decisões e reagir de modo adequado a um determinado cenário; e serviços de ubiquidade que visam prover serviços de colaboração e inteligência em qualquer lugar e momento em que eles sejam necessários (SANTOS et al., 2016);
6. **Comunicação:** faz referência as técnicas utilizadas para conectar os objetos físicos. As tecnologias utilizadas na comunicação representam um fator crítico uma vez que representam parte do consumo de energia dos dispositivos. Algumas tecnologias utilizadas são *Wifi*, *Bluetooth*, *IEEE 802.15.4* e *RFID* (SANTOS et al., 2016).

2.2.1 Características da IoT

As principais características da tecnologia IoT são: interconectividade, serviços, heterogeneidade, mudanças dinâmicas, escalabilidade, segurança e conectividade (PATEL; PATEL, 2016).

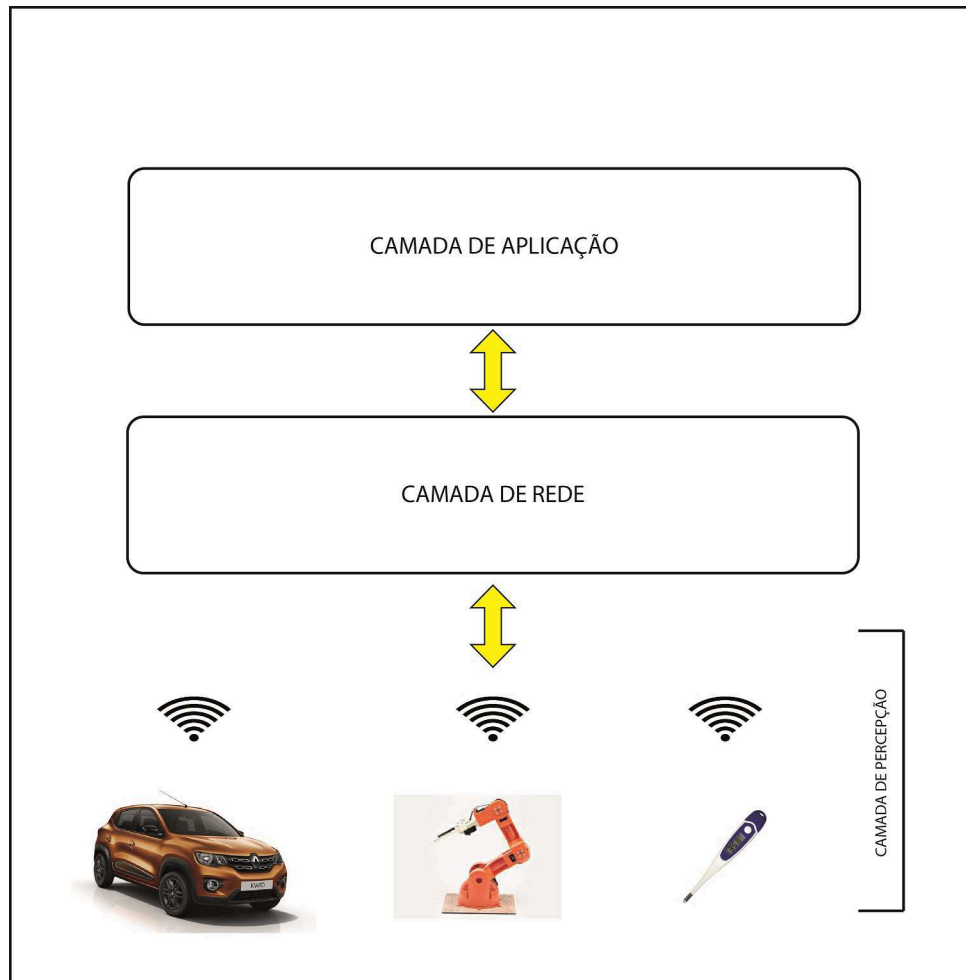
1. **Interconectividade:** todos os dispositivos podem se conectar (PATEL; PATEL, 2016);
2. **Serviços:** está relacionada com a segurança e consistência semântica entre os objetos físico e virtuais (PATEL; PATEL, 2016);

3. **Heterogeneidade:** está relacionada ao fato de os diversos dispositivos heterogêneos serem capazes de se comunicarem por intermédio das redes de comunicações (PATEL; PATEL, 2016);
4. **Mudança dinâmica:** está ligado ao fato dos dispositivos mudarem de localização geográfica e está ligado ao fato da quantidade de dispositivos na rede variar dinamicamente (PATEL; PATEL, 2016);
5. **Escalabilidade:** quanto maior o número de dispositivos conectados à internet, mais complexo se torna o gerenciamento da rede (PATEL; PATEL, 2016);
6. **Segurança:** está relacionado à segurança com que os dados são transmitidos pela rede (PATEL; PATEL, 2016);
7. **Conectividade:** está relacionada à capacidade de um objeto de se conectar à rede (PATEL; PATEL, 2016).

2.3 Arquiteturas IoT

As arquiteturas IoT mais utilizadas são a de três camadas e de cinco camadas (ALQASEEMI et al., 2017). A arquitetura faz referência à prática de se dividir um protocolo em camadas, cada uma dessas camadas é responsável por realizar uma tarefa específica. A utilização de arquiteturas facilita o entendimento, manutenção e concepção de sistemas.

A arquitetura de 3 camadas (Figura 2) é formada pelas camadas de percepção, rede e aplicação.

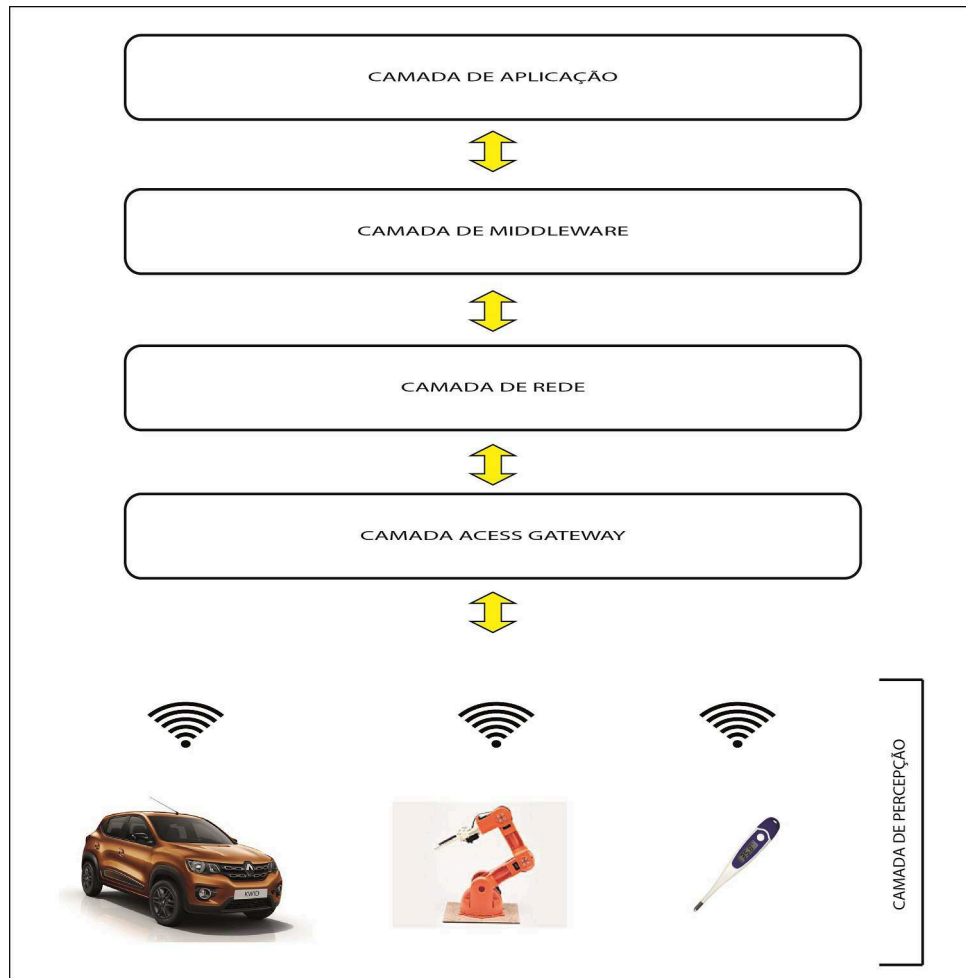
Figura 2 – Arquitetura *IoT* de 3 camadas

Fonte: Adaptado de (AL-QASEEMI et al., 2017)

A camada de percepção é considerada a camada física, nela estão os dispositivos físicos e a camada de sensores proveniente dos dispositivos embarcados. Na camada de percepção, os sensores dos dispositivos físicos coletam dados do ambiente onde estão situados e enviam esses dados para a camada de rede. A camada de rede é responsável por coletar as informações provenientes da camada de percepção e enviá-las para a camada de aplicação, essa camada possui tecnologias de comunicação *wireless* e *wired*. Na camada de aplicação os dados recebidos são processados e analisados, eles podem ser utilizados por serviços ou para tomada de decisões. As decisões tomadas na camada de aplicação são mandadas de volta para a camada de percepção por meio da camada de rede para que a ação necessária seja realizada.

Já na arquitetura de 5 camadas (Figura 3) existem duas camadas adicionais, as camadas *access gateway* e *middleware*.

Figura 3 – Arquitetura IoT de 5 camadas



Fonte: Adaptado de (AL-QASEEMI et al., 2017)

A camada *access gateway* é responsável por gerenciar as comunicações do ambiente IoT e também por realizar a troca de mensagens entre os dispositivos IoT. A camada de *middleware* é responsável por implementar a comunicação entre os *hardware devices* (notebook, *smartphone*, computador e etc) e as aplicações.

2.4 MQTT

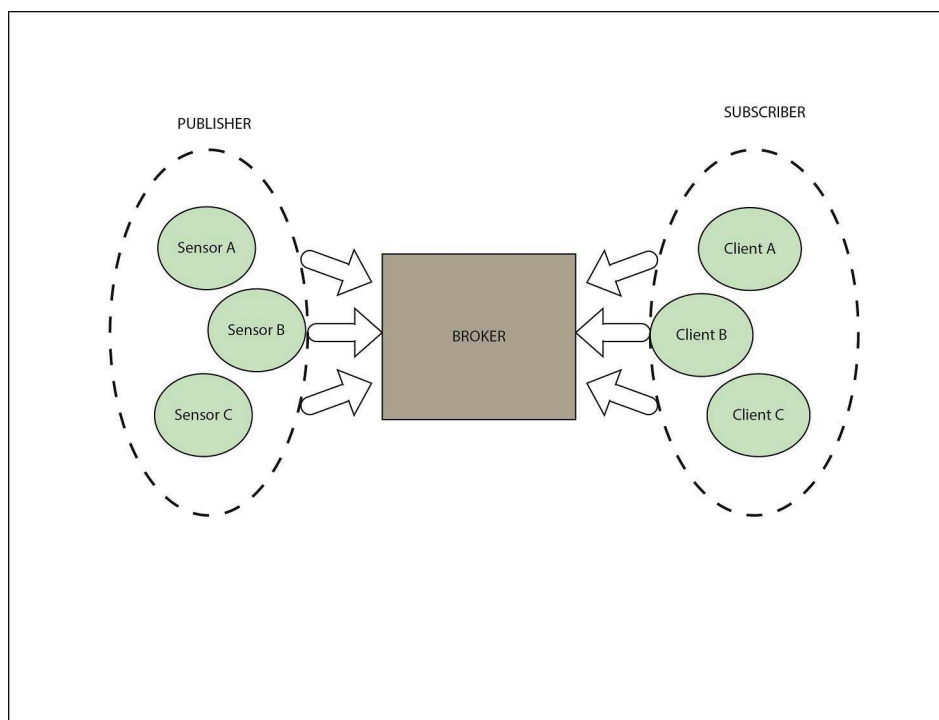
Criado pela IBM na década de 90 com foco em sistemas de supervisão e aquisição de dados, o protocolo evoluiu e encontrou seu espaço nesse amplo mercado de Internet das Coisas. O protocolo MQTT foi concebido para conectar dispositivos, redes, aplicações, serviços e *middlewares*. O protocolo foi concebido visando utilizar a infraestrutura e realizar integração com os protocolos TCP e IP. Além disso, o MQTT foi projetado para aplicações que utilizam pouca banda de rede, com requisitos de hardware extremamente simples e leve.

O MQTT está na mesma camada OSI que o *Hypertext Transfer Protocol* (HTTP), porém a maior diferença entre eles é o tamanho do *payload*. No HTTP o *payload* é maior, o que inviabiliza o uso em conexões de baixa qualidade. Além disso o MQTT possui maior segurança, apresenta mais níveis de serviço, é menos complexo e permite uma comunicação de 1 para N se comparado ao HTTP que também é um protocolo utilizado no universo de Internet das Coisas.

No MQTT o esquema de troca de mensagens é fundamentado no modelo *publish/subscriber* (YACCHIREMA; PALAU; ESTEVE, 2016). O modelo *publish/subscriber* faz com que a comunicação entre as partes seja assíncrona, ele desacopla o emissor e o receptor da mensagem tanto no espaço quanto no tempo. Na arquitetura *publish/subscriber* do protocolo MQTT a identificação das mensagens se dá por meio de tópicos (*topics*). O tópico lembra o conceito de *Uniform Resource Locator* (URL) onde os níveis são separados por barras ("/") (CUNHA et al., 2016). Em aplicações envolvendo a Internet das Coisas, o MQTT é um dos protocolos mais utilizados devido a sua definição de qualidade de serviço, especificações de segurança, implementação simples e garantia de utilização da banda de uso moderada (LAMPKIN et al., 2012).

A Figura 4 elucida o modo de funcionamento do protocolo MQTT.

Figura 4 – Arquitetura MQTT



Fonte: Adaptado de (CUNHA et al., 2016)

No padrão *publish/subscriber*, quando o cliente ou elemento da rede deseja receber uma determinada informação, ele dá um *subscribe* no tópico de interesse através de uma requisição para um outro elemento da rede que é conhecido como *Broker* que atua como

um intermediário no processo de comunicação. Clientes que desejam publicar algum tipo de informação o fazem através do *Broker* também, enviando-lhe as informações que possui e este fica responsável por rotear a mensagem até o destinatário.

Clientes podem ser tanto publicadores (*publishers*) como assinantes (*subscribers*), ou seja, clientes assinam tópicos para enviar e receber mensagens. É importante frisar que o fluxo de comunicação se dá majoritariamente de quem publica informações como sensores por exemplo, para quem assina e deseja receber aquela informação que são os mais variados clientes como por exemplo aplicações rodando na nuvem ou um simples *Raspberry*. O cliente é dito um cliente MQTT quando um dispositivo é capaz de rodar as bibliotecas e módulos MQTT disponíveis para os mais diversos ambientes e linguagens de programação. Um exemplo de cliente MQTT seria um *Raspberry* rodando um código em *python* utilizando as bibliotecas *Eclipse Paho MQTT Python client* ou um ESP32 utilizando um *client* para MQTT (SAHA et al., 2018).

A conexão dos clientes MQTT ao *message Broker* acontece via Protocolo de Controle de Transmissão (TCP). Na conexão se estabelece o *login* (usuário e senha) e o método de criptografia *Transport Layer Security/ Secure Socket Layer* (SSL/TLS). No processo de conexão entre os elementos também se estabelece o nível de *Quality of Service* (Qos) que se deseja para o elemento conectado ao *Broker*. Este Qos defini a garantia da entrega da mensagem e possui três níveis. Sendo eles:

1. **Qos 0 (*at most once*)**: é o que possui o menor esforço, onde não são exigidas confirmações no momento em que a mensagem é entregue (CUNHA et al., 2016);
2. **Qos 1 (*at least once*)**: neste nível existe a confirmação de entrega de uma mensagem, porém várias mensagens iguais são geradas, mas apenas uma terá o reconhecimento de chegada (CUNHA et al., 2016);
3. **Qos 2 (*exactly once*)**: garante que a mensagem seja entregue exatamente uma vez (CUNHA et al., 2016) com envio de confirmação de recebimento e confirmações de recebimento de confirmações de recebimento.

Após a escolha do Qos e conexão com o *Broker*, o protocolo MQTT especifica que é responsabilidade do cliente garantir que o intervalo de tempo em que as mensagens demoram a ser enviadas não exceda um valor de *Keep Alive*. Caso o valor seja excedido, o cliente envia uma mensagem de controle PINGREQ. PINGREQ avisa ao *Broker* que ele ainda está vivo (*online*). Se o *Broker* não receber um PINGREQ ou qualquer mensagem de um cliente específico, o *Broker* fecha a sessão. O máximo intervalo de *Keep Alive* é 18h 12 min e 15 seg, se o valor for setado para 0, o recurso é desativado. Se o problema de conexão ainda persistir, a biblioteca iniciará o recurso chamado “*Client Take Over*”, fazendo com que o *Broker* feche a sessão e envie uma requisição de uma nova conexão com o cliente (STANFORD-CLARK; NIPPER, 2014).

Devido as características de controle de conectividade, desacoplamento, o MQTT é altamente desacoplado do espaço, uma vez que os sensores que produzem os dados não necessitam conhecer a identidade dos clientes que estão interessados naquela informação. Enquanto que no CoAP é o contrário, o consumidor entre em contato direto com o produtor, o que exige a identificação de ambas as partes, acoplamento de tempo, no MQTT, os produtores podem publicar mesmo que os consumidores estejam desconectados e, inversamente, os consumidores podem ser notificados mesmo que não haja produtores. Enquanto que no CoAP a informação só fica disponível enquanto o produtor estiver disponível, e a interação entre os nós precisa ser ao mesmo tempo. Acoplamento de sincronização, que indica o nível de sincronismo da realização de atividades em relação a outras. No MQTT, os produtores não são bloqueados enquanto produzem eventos e consumidores podem ser assincronamente notificados da ocorrência de um evento enquanto efetuam outra atividade, o que é exatamente oposto ao que ocorre no CoAP. E também ao fato de o MQTT ser mais voltado para uma comunicação “muitos para muitos”, para transportar mensagens entre vários clientes através de um servidor central broker e utilizando os protocolos da pilha TCP/IP, o CoAP é mais voltado para uma comunicação “um para um”, para transferência de informação entre cliente e servidor, utilizando o protocolo UDP/IP (BANKS; GUPTA,), o MQTT foi preterido para utilização como protocolo de troca de mensagem do trabalho.

2.5 Manipulador Robótico

Segundo o *Robot Institute of America*, braço robótico é um manipulador multifuncional reprogramável concebido para realização de tarefas, como soldagem, montagem e içamento de materiais. Já para o dicionário *Merriam-Webster*, manipulador robótico é um dispositivo automático capaz de realizar tarefas complicadas e repetitivas, ou seja, são equipamentos concebidos para a realização de tarefas demasiadamente onerosas e perigosas para serem realizadas por seres humanos.

A estrutura típica de um manipulador robótico é composta por elos e juntas de movimento relativo, nas juntas são acoplados acionadores responsáveis por realizar os movimentos do braço robótico (OTTONI, 2010). Fisicamente, o braço robótico é fixado a base (piso, ou plataforma) de um lado e ao instrumento utilizado (garra, maçarico, ponta de solda e etc) pelo outro.

A Figura ?? ilustra a estrutura típica de um manipulador robótico descrita no parágrafo anterior.

No manipulador robótico, os acionadores acoplados as juntas podem ser atuadores elétricos, pneumáticos ou hidráulicos, esses atuadores conferem aos robôs uma capacidade de carga muito maior que a humana (OTTONI, 2010), na parte do punho se encontra várias

juntas próximas que permitem a orientação do órgão terminal nas posições necessárias para a execução da tarefa, essas juntas do punho são agrupadas num pequeno volume de modo a não mexer de forma demasiada a garra quando acionadas.

Na maioria dos robôs industriais as juntas se restringem as juntas rotativas e prismáticas (OTTONI, 2010). A junta rotativa gira em torno de uma linha imaginária estacionária chamada de eixo de rotação. A junta prismática move em linha reta e é composta de duas hastes que deslizam entre si.

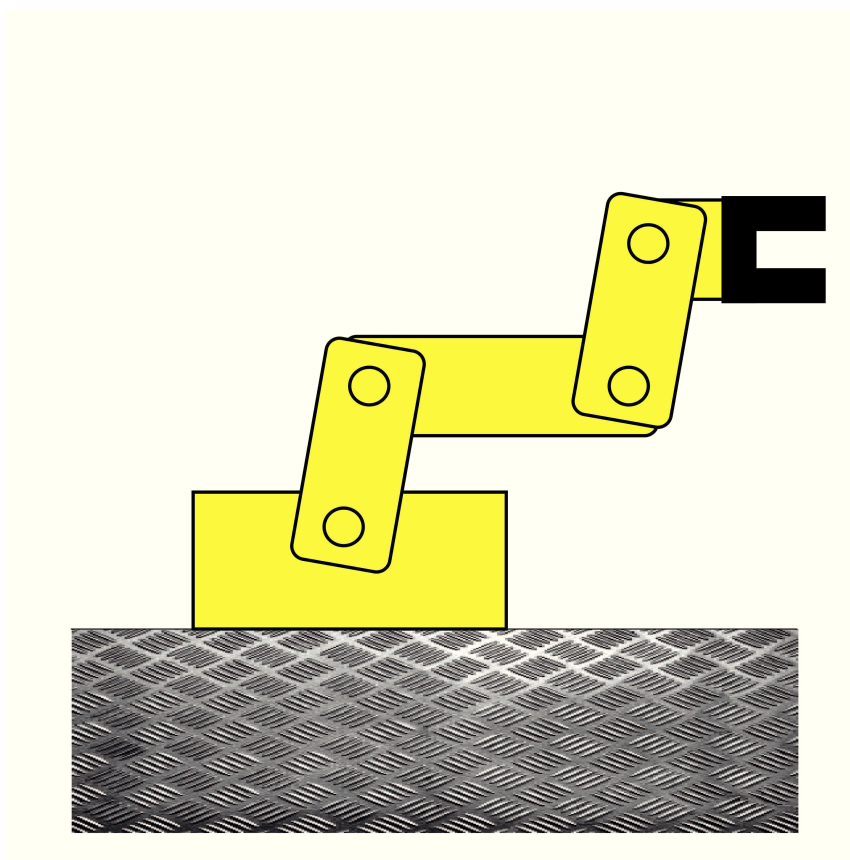
Outro conceito importante em manipuladores robóticos é o conceito de grau de liberdade. Este conceito está relacionado a capacidade de movimentação que o braço robótico possui no espaço bidimensional ou tridimensional. Cada junta de movimento do braço robótico define um ou dois graus de liberdade, dessa forma o número de graus de liberdade de um manipulador robótico é igual a soma dos graus de liberdade de suas juntas. Quanto maior o número de graus de liberdade mais difícil se torna a cinemática, dinâmica e controle do braço (OTTONI, 2010). Para manipuladores robóticos o arranjo adotado é de 3 graus de liberdade e 3 acionadores para movimentação do órgão terminal (OTTONI, 2010), com esse arranjo é possível alcançar qualquer ponto dentro de um espaço limitado ao redor do braço (PAZOS et al., 2015).

A movimentação do braço robótico pode ser dividida em vertical transversal, rotacional transversal e radial transversal. A primeira é aquela onde a garra movimenta verticalmente de cima para baixo, a segunda movimentação é aquele movimento em que a garra se move horizontalmente para a esquerda ou para a direita, a movimentação radial transversal é o movimento de aproximação ou afastamento da garra (OTTONI, 2010).

Já os movimentos realizados pelo punho do braço robótico são classificados em rolagem (*Roll*), arfagem (*Pitch*) e guinada (*Yaw*). O primeiro está relacionado a rotação do punho em torno do braço, o segundo a rotação do punho para cima e para baixo e o terceiro a rotação do punho para a esquerda ou para a direita.

Uma das principais características de um braço industrial é a sua capacidade de carga, ou seja, o peso máximo de trabalho que ele consegue sustentar sem ocasionar perda demasiada de sua precisão. Esse valor é obtido através da medição na posição mais crítica do braço robótico, ou seja, a posição em que o braço se encontra na sua extensão máxima. As cadeias cinemáticas fechadas ou abertas podem ser utilizadas como forma de aliviar o peso. A cadeia de cinemática aberta é aquela em que o robô partindo da sua base chega ao seu punho por meio de um único caminho através de uma sequência elo-junta-elo (OTTONI, 2010). A Figura 6 elucida o exemplo da cadeia cinemática aberta.

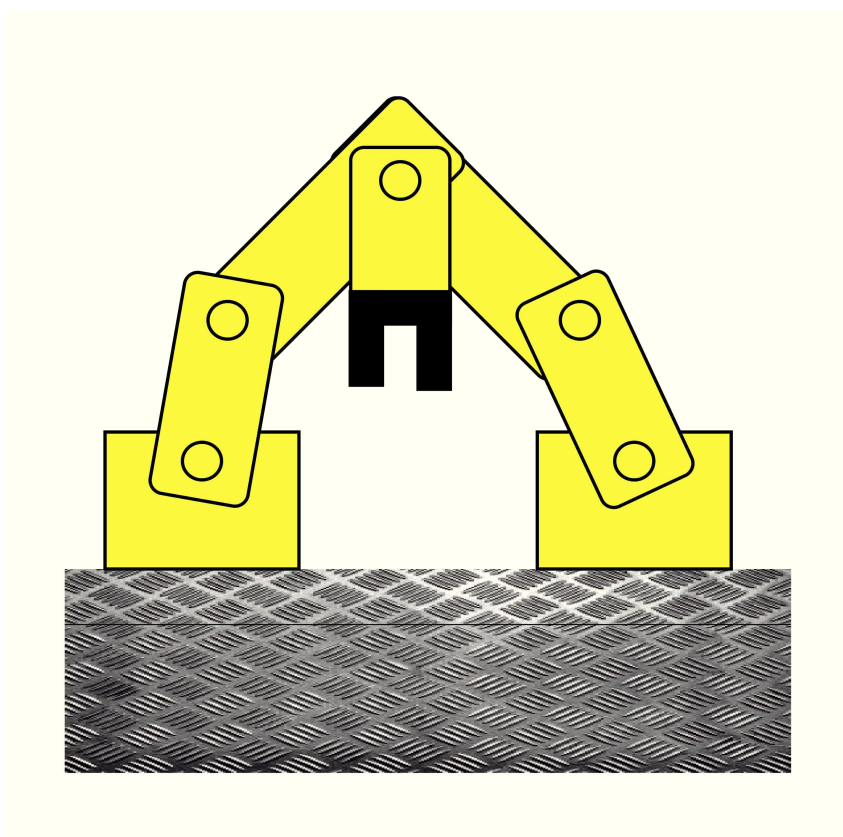
Figura 6 – Cadeia Cinemática Aberta



Fonte: Próprio Autor

Já na cadeia de cinemática fechada o braço não possui um único caminho para chegar até o punho. Os elos podem estar conectados entre si de tal maneira que se torna possível percorrer um caminho que parta da base e retorne a ela por outro caminho após atingir o punho. A Figura 7 abaixo elucida o que foi dito no parágrafo acima.

Figura 7 – Cadeia Cinemática Fechada



Fonte: Próprio Autor

2.6 Considerações Finais

Nesse capítulo foram apresentados alguns conceitos e definições das três principais áreas nas quais este trabalho se baseia. A solução desenvolvida conta com a aplicação dos conceitos aqui apresentados. O próximo capítulo descreve um conjunto de trabalhos associados ao tema dessa dissertação.

Estado da Arte

Os serviços e produtos gerados através do novo paradigma de conectividade e comunicação da Internet das Coisas podem gerar uma receita 300 bilhões de dólares além de adicionar 1.3 bilhões de dólares a economia mundial em 2020 (LUND et al., 2014). Este capítulo apresenta brevemente trabalhos sobre MQTT, Internet das Coisas e Robótica Industrial.

3.1 Internet das Coisas e a Indústria

Visando esse futuro promissor de ganhos econômicos, diversos trabalhos já são realizados com o intuito de aplicar a Internet das Coisas nos mais diversos setores da economia. Já se realizam pesquisas no setor automotivo (MENDIRATTA; DEY; SONA, 2017), agricultura de precisão (DOLCI, 2017), hospitalar (STRADOLINI et al., 2018), rastreamento veicular (HUSNI et al., 2016), agropecuária (MAINA, 2017), segurança veicular (RUENGITTINUN; PAISALWONGCHAROEN; WATCHARAJINDASAKUL, 2017) entre outros. O setor Industrial é outro setor da economia que busca usufruir dos ganhos econômicos advindos da Internet das Coisas (DOLCI, 2017).

Diversas pesquisas são realizadas com o intuito de integração de tecnologias IoT ao ambiente industrial. Se tem estudo que trata da integração da malha de atuadores e sensores *wireless* com o intuito de criar ambientes IoT (YACCHIREMA; ESTEVE; PALAU, 2017). Esses ambientes são povoados por dispositivos como sensores, atuadores e aplicações de software de forma transparente para o usuário. Seguindo o mesmo conceito de ambientes IoT, existe estudo em que é tratada a questão da interoperabilidade de controladores lógicos programáveis com tecnologias IoT (HEMMATPOUR et al., 2017). Já em outro trabalho, a preocupação está na integração dos dispositivos das redes *FieldBus* baseadas no *Real Time Ethernet* (RTE) (BELLAGENTE et al., 2016), no caso do trabalho a rede *PROFINET*, ao ambiente da Internet das Coisas. Existe trabalho que prevendo a grande quantidade de dispositivos conectados à rede, discute acerca do tráfego de informações que será gerado pelos dispositivos IoT (DIAZ-CACHO et al., 2015).

3.2 Indústria, Internet das Coisas e MQTT

Entre as diversas tecnologias IoT que a indústria busca integrar ao ambiente industrial, o protocolo MQTT vem sendo utilizado na troca de mensagens. Ele é considerado ideal para uso em ambientes restritos e em dispositivos com bateria, processador e memória limitada. O MQTT juntamente com o CoAP são dois excelentes protocolos para transporte de mensagem no ambiente da Internet das Coisas. O CoAP é um protocolo para ambientes limitados em termos de comunicação, como também o MQTT é, o protocolo de troca de mensagens CoAP é baseado na especificação *Representational State Transfer* (REST), porém ele é capaz de trabalhar com o HTTP (SHELBY; HARTKE; BORMANN, 2014).

A arquitetura do protocolo CoAP é dividida em duas camadas: requisição/resposta e mensagem. A camada de mensagem é responsável pelo o gerenciamento das trocas de mensagens sobre a *User Datagram Protocol* (UDP). A camada de requisição/resposta é responsável por fornecer um formato comum de comunicação com a aplicação do usuário. As mensagens são identificadas por um identificador (ID) usado para confiabilidade da informação (SHELBY; HARTKE; BORMANN, 2014).

No entanto, embora os dois protocolos tenham sido concebidos para aplicação em ambientes com comunicação limitada, o protocolo de troca MQTT apresenta as seguintes vantagens em relação ao CoAP:

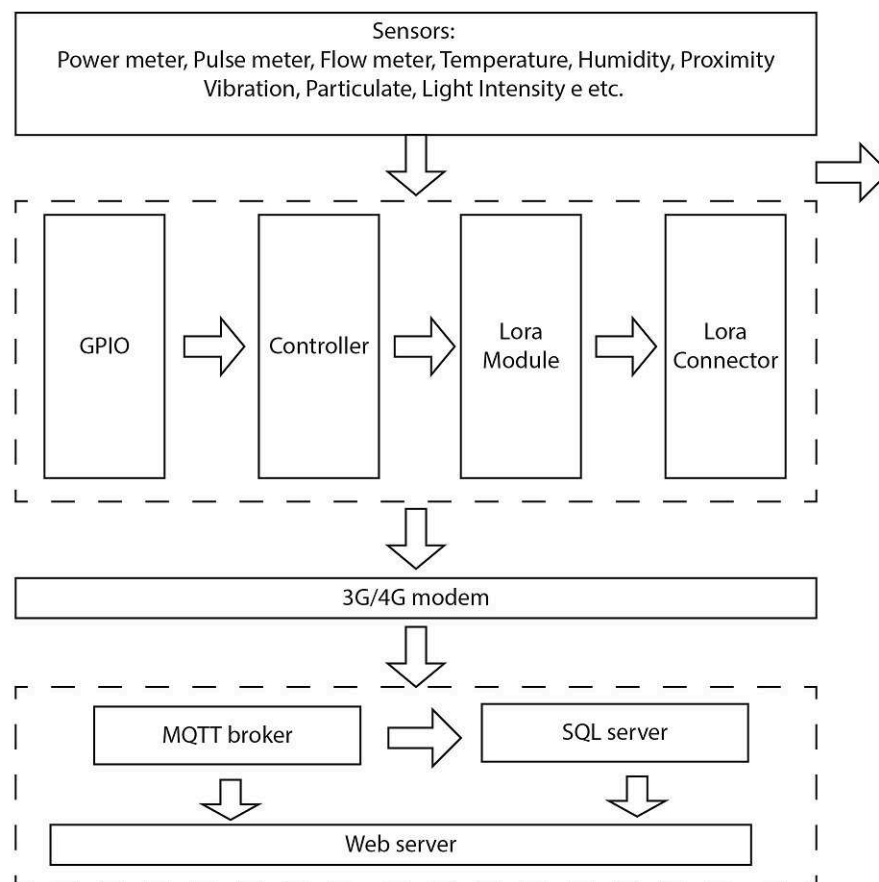
1. O transporte com pequeno *overhead* torna o MQTT uma interessante solução para redes com restrição de recursos, com baixa largura de banda e alta latência (BELLAVISTA; ZANNI, 2016);
2. O MQTT é mais voltado para uma comunicação "muitos para muitos", para transportar mensagens entre vários clientes através de um servidor central Broker e utilizando os protocolos de pilha TCP/IP, o CoAP é mais voltado para uma comunicação "um para um", para transferência de informação entre cliente e servidor, utilizando o protocolo UDP/IP (BELLAVISTA; ZANNI, 2016);
3. O MQTT é altamente desacoplado do espaço, uma vez que os sensores que produzem os dados não necessitam conhecer a identidade dos clientes que estão interessados naquela informação. Enquanto que no CoAP é o contrario, o protocolo exige a identificação de ambas as partes (BELLAVISTA; ZANNI, 2016);
4. No MQTT, os produtores podem publicar mesmo que os consumidores estejam desconectados e, inversamente, os consumidores podem ser notificados mesmo que não haja produtores. Enquanto que no CoAP a informação só fica disponível enquanto o produtor estiver disponível, e a interação entre os nós precisa ser ao mesmo tempo (BELLAVISTA; ZANNI, 2016);

5. No MQTT, os produtores não são bloqueados enquanto produzem eventos e consumidores podem ser assincronamente notificados da ocorrência de um evento enquanto efetuam outra atividade, o que é exatamente oposto ao que ocorre no CoAP (BELLAVISTA; ZANNI, 2016).

Devido à essas características, e principalmente a característica de ser um protocolo de comunicação "muitos para muitos", na maioria das pesquisas existentes, o protocolo de comunicação MQTT é utilizado em cenários onde a quantidade de dispositivos se comunicando é grande. O protocolo é utilizado em sistemas que buscam o monitoramento de ambientes industriais, análise comparativa de performance com outros protocolos industriais da Internet das Coisas e em situações de estimativa de latência na comunicação.

Na aplicação proposta em Gan et al. (2018), a questão da aquisição e gerenciamento de dados é tratada por meio da implementação de uma aplicação IoT que realiza a aquisição de dados dos equipamentos referentes ao consumo de energia de uma fábrica de pães. A Figura 8 ilustra o fluxo da informação na aplicação.

Figura 8 – Diagrama da Aplicação IoT



Fonte: Adaptado de (GAN et al., 2018)

A Figura 8, mostra que a arquitetura utilizada no sistema de gerenciamento IoT de energia proposto foi a de 3 camadas, essa arquitetura é formada pela camada de percepção (*perception layer*), camada de transmissão e a camada de aplicação.

A camada de percepção está diretamente ligada aos processos industriais, ela é a camada da arquitetura onde os sensores inteligentes, medidores de potência inteligente, transformadores de corrente, medidores de vazão, sensores de temperatura, sensores de umidade e outros estão localizados. Essa camada também é responsável por realizar a interface com os sistemas Sistemas de Supervisão e Aquisição de Dados (SCADA) utilizados na indústria. Já a camada de transmissão é responsável por transmitir os dados provenientes dos sensores e do sistema SCADA para o servidor na nuvem. A camada de aplicação é responsável por mostrar em tempo real a análise do consumo de energia da fábrica.

Já no sistema de gerenciamento IoT industrial e ambiental proposto em Klymash et al. (2018), os protocolos MQTT e CoAP foram utilizados em conjunto com o padrão *Narrowband-IoT* (NB-IoT) nas trocas de mensagens, o NB-IoT permite que a comunicação de telefonia móvel seja utilizada por dispositivos com capacidades limitadas. Na arquitetura, os dois protocolos foram utilizados na camada de aplicação, foram responsáveis por conectarem sensores e controlarem dispositivos em canais com baixa largura de banda e pouca robustez. A Figura 9 a capacidade de monitoramento a nível global e regional, do sistema proposto.

Figura 9 – Aplicação a nível global e regional



Fonte: Retirado de (KLYMASH et al., 2018)

A utilização do algoritmo de clusterização *K-means* para agrupamento dos sensores, permitiu a utilização do sistema de gerenciamento IoT a nível global e regional com uma

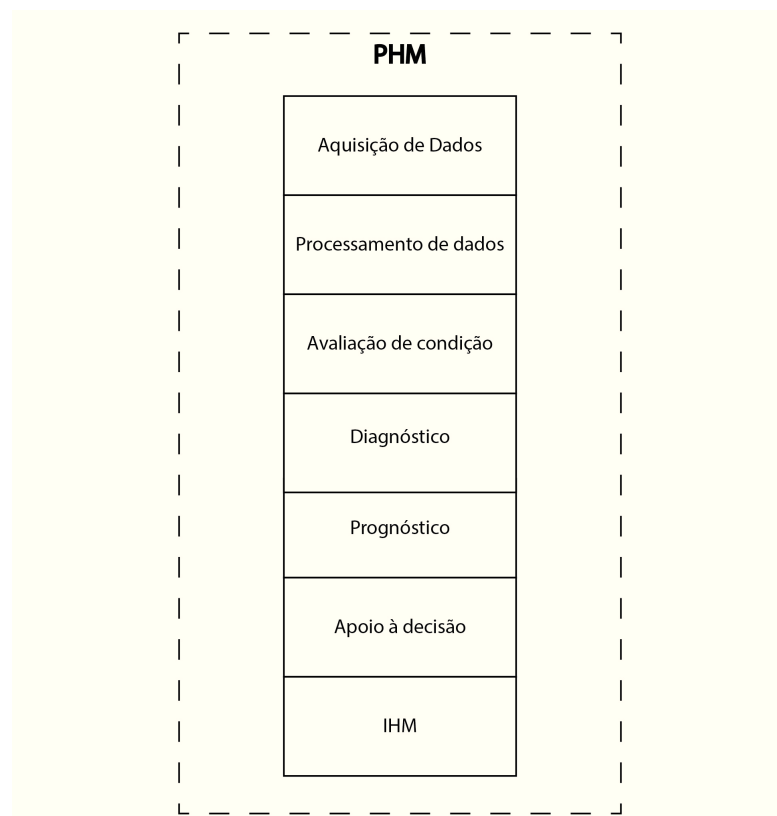
performance satisfatória.

Já no trabalho de Parmar et al. (2018), uma luva inteligente é proposta com o intuito de gerenciar o estado de operação de equipamentos do chão de fábrica. Sensores instalados nas máquinas, são responsáveis por enviar informações sobre o estado de operação do equipamento para o *Broker* MQTT localizado em uma Raspberry Pi, esse Broker é responsável por gerenciar as trocas de mensagens na aplicação. Através da luva o operário tem acesso a informação do estado de funcionamento da máquina, através da luva inteligente, o operário pode desligar a máquina, pois a luva inteligente envia uma mensagem para a Raspberry Pi, que controla os relés das máquinas do chão de fábrica.

Seguindo o pensamento de prevenção de acidentes, no trabalho de Ayad, Terrissa, Zerhouni (2018), é abordada a importância dos sistemas *Prognostic Health Management* (PHM) na detecção de anomalias nos sistemas industriais. Nesses sistemas, algoritmos são usados para detectar anomalias, diagnosticar falhas e prever a *Remaining Useful Life* (RUL) dos equipamentos. O PHM pode fornecer muitas vantagens para os usuários, pois reduz custos operacionais e de manutenção, prolonga a vida útil dos equipamentos e diminui os riscos de acidentes por mal funcionamento dos equipamentos.

Os sistemas PHM são constituídos por sete módulos. A figura 10 ilustra os módulos dos programas PHM.

Figura 10 – Módulos dos Sistemas de Prognóstico PHM



Fonte: Adaptado de (AYAD; TERRISSA; ZERHOUNI, 2018)

Esses sete módulos são:

1. **Aquisição de dados:** é o processo de coleta e armazenamento dos dados digitais de sensores ou transdutores;
2. **Processamento de dados:** os dados adquiridos são processados para extração, redução e seleção de recursos relevantes e indicadores que podem fornecer informações sobre o comportamento do sistema, a presença de anomalias e a evolução da degradação;
3. **Avaliação de condição:** este módulo trata da classificação e detecção dos estados do sistema. Ele pode ser assimilado à detecção de falhas;
4. **Diagnóstico:** Este módulo concentra-se na detecção, isolamento e identificação de causas de falhas;
5. **Prognóstico:** O objetivo do módulo é prever a falha antes que ocorra assim como seu RUL;
6. **Apoio à decisão:** Sua principal função é recomendar as melhores ações de manutenção;
7. **Interface Homem Máquina (IHM):** Este módulo poderia ser construído em uma interface de máquina humana regular.

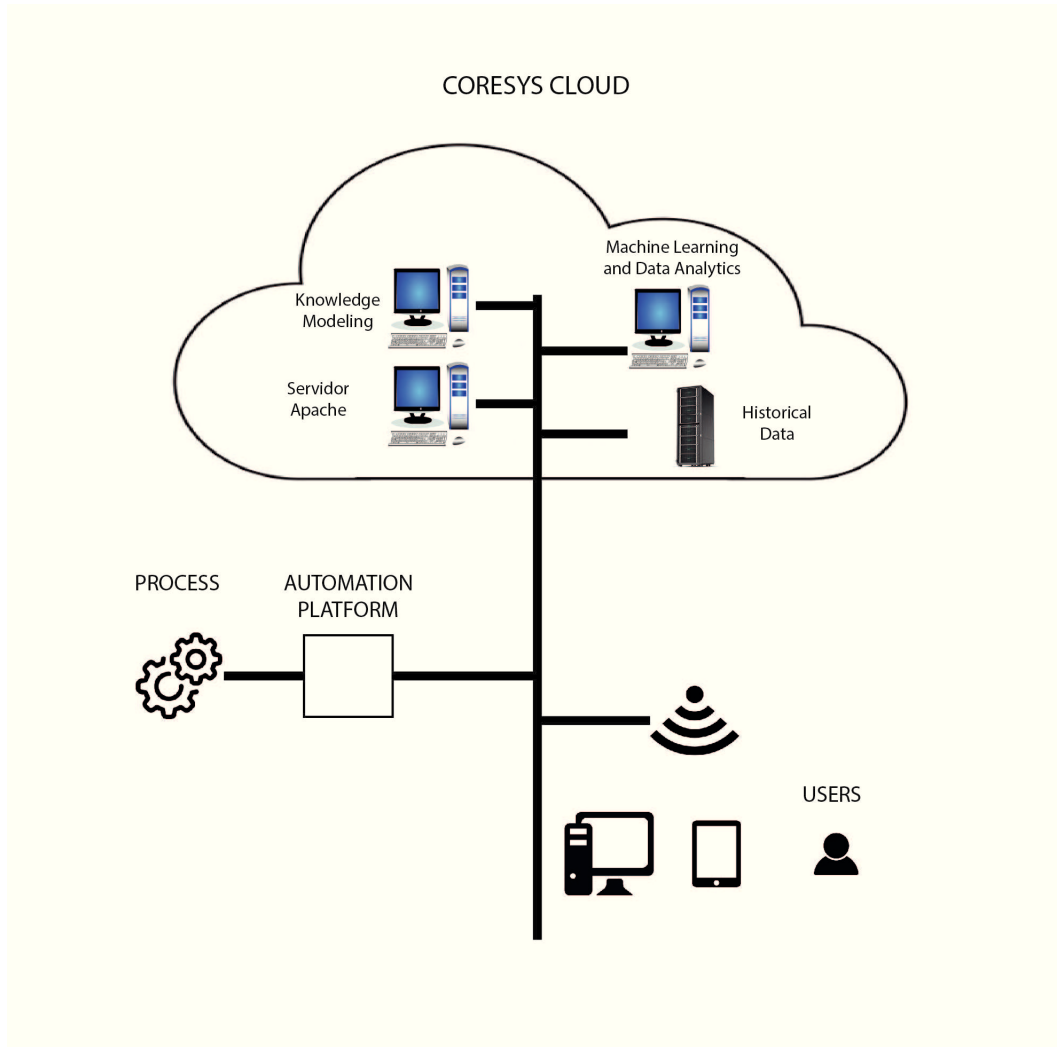
Diante da facilidade de comunicação oferecida pela Internet das Coisas Ayad, Terrissa, Zerhouni (2018) propõem a integração do sistema PHM ao ambiente industrial da Internet das Coisas, pois a integração permite a avaliação do estado dos equipamentos em tempo real, o que melhora o desempenho dos sistemas PHM. Sendo assim, os autores propuseram e implementaram uma solução inteligente baseada em MQTT e Cloud Computing para avaliação/manutenção de equipamentos industriais a qualquer hora, por meio da solução proposta os autores resolveram eficientemente o problema de gestão do estado de funcionamento de máquinas localizadas em uma planta industrial.

Com o mesmo intuito de monitoramento de condições, no trabalho de Arévalo, Diprasetya, Schwung (2018), os autores tratam da questão da digitalização da indústria e a tendência crescente no uso de técnicas de aprendizado de máquina para monitoramento de condições, detecção de falhas e otimização de processo. As abordagens tradicionais usam tecnologias de TI centralizadas em um servidor local para fornecer esses serviços.

Atualmente, a computação em nuvem pode substituir as estruturas de TI locais por um serviço remoto, que pode ser pago de acordo com as necessidades do cliente. Sendo assim no trabalho os autores propuseram uma arquitetura baseada em nuvem para monitoramento de condições com base no aprendizado de máquina.

O sistema é implementado usando a *Dempster-Shafer Evidence Theory* (DSET). A figura a seguir abaixo ilustra o *framework* proposto.

Figura 11 – *Framework* Proposto



Fonte: Adaptado de (ARÉVALO; DIPRASETYA; SCHWUNG, 2018)

Através da Figura 11, percebemos os principais componentes da solução proposta. O gateway OPC-UA/MQTT permite a comunicação entre o servidor OPC da plataforma de automação e o broker MQTT do CoreSys-Cloud. O gateway comunica com a plataforma de automação através do OPC-UA. Na aplicação, o MQTT estabelece a comunicação entre CoreSys-Cloud e o aplicativo da web.

No CoreSys-Cloud quatro serviços foram implementados, sendo eles:

1. **Machine Learning e Data Analytics:** Este é o módulo de aprendizagem e análise de dados usado para avaliação do monitoramento de estados. O MLDA é escrito em Python. A biblioteca scikit-learn foi utilizada para a implementação das funções de aprendizado de máquina do MLDA;

2. **Flask:** O Flask é um software usado para desenvolvimento de aplicações web. As razões para escolha do Flask foram sua estrutura leve e o fato de ser uma biblioteca Python. Estes fatos permitem uma integração mais fácil do Flask ao MLDA, uma vez que ambos são escritos na mesma linguagem de programação;
3. **Servidor de banco de dados MySQL:** responsável por armazenar os resultados da avaliação do MLDA;
4. **Servidor MQTT:** o servidor MQTT é responsável por estabelecer a comunicação entre o gateway OPC-UA / MQTT e o CoreSys-Cloud. O servidor MQTT se comunica com o Flask que fornece um método de autenticação usando o nome de usuário e senha, o que impede uma conexão desconhecida com o Servidor MQTT. O Flask permite o desenvolvimento de uma aplicação web que serve como interface entre o broker MQTT do CoreSysCloud e o banco de dados MySQL. A aplicação web obtém dados do gateway OPC-UA / MQTT através do protocolo de comunicação MQTT. Uma função é chamada sempre que o gateway publica dados para o broker MQTT. Esta função armazena os dados publicados no banco de dados MySQL.

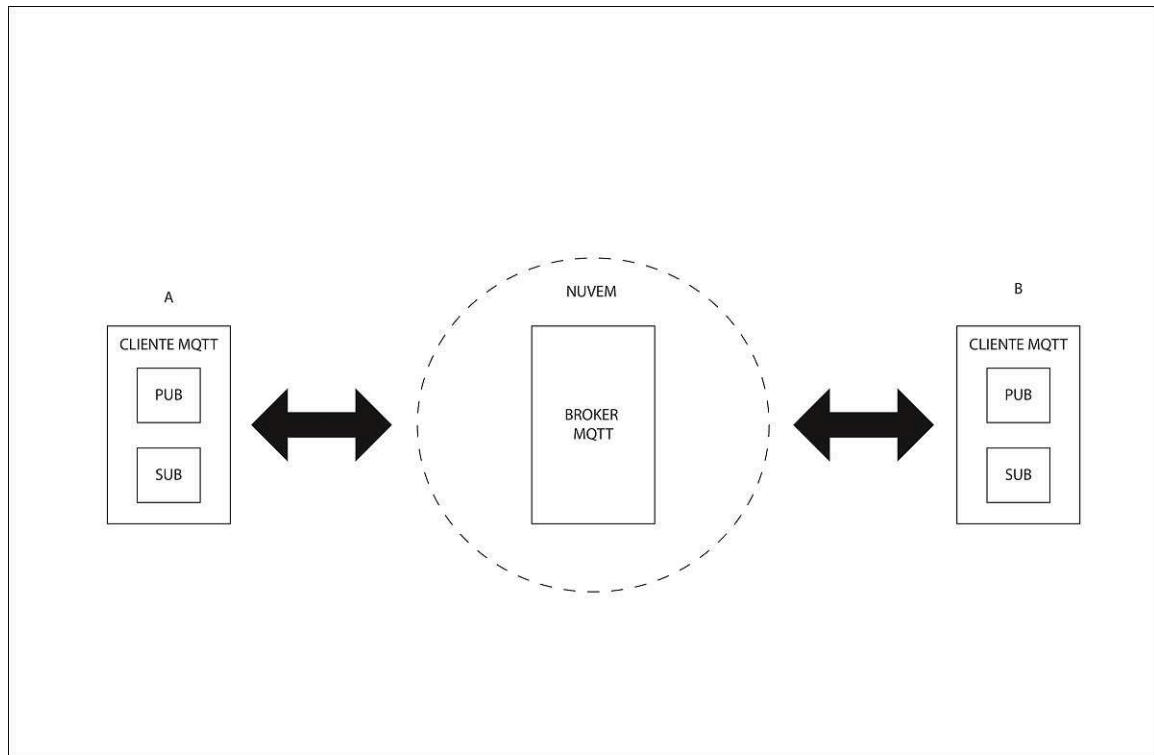
O CoreSys-Cloud oferece uma alternativa de estrutura de TI para acesso à serviços de bancos de dados, aprendizado de máquina, aplicativos móveis e conectividade com plataformas de automação para empresas de porte pequeno e médio.

Em vista dos trabalhos apresentados, ficou evidente que as situações em que o MQTT é aplicado, são situações onde se tem um grande volume de clientes, para transportar mensagens entre vários clientes através de um servidor central Broker e utilizando os protocolos de pilha TCP/IP.

Já nos trabalhos em que o MQTT é aplicado na estimativa de latência em comunicações os cenários são mais simples. No trabalho de Ferrari et al. (2017) o cenário de trabalho proposto foi resumido a dois dispositivos IoT que enviaram os seus dados para um servidor na nuvem e o servidor enviou novamente os dados para os dispositivos IoT. A transferência de dados foi realizada por meio do protocolo MQTT com Qos = 0.

A implementação do servidor na nuvem foi mantida o mais simples possível para não influenciar nos valores medidos de latência. A figura 12 abaixo ilustra o esquema utilizado no trabalho.

Figura 12 – Representação Gráfica da Aplicação



Fonte: Adaptado de (FERRARI et al., 2017)

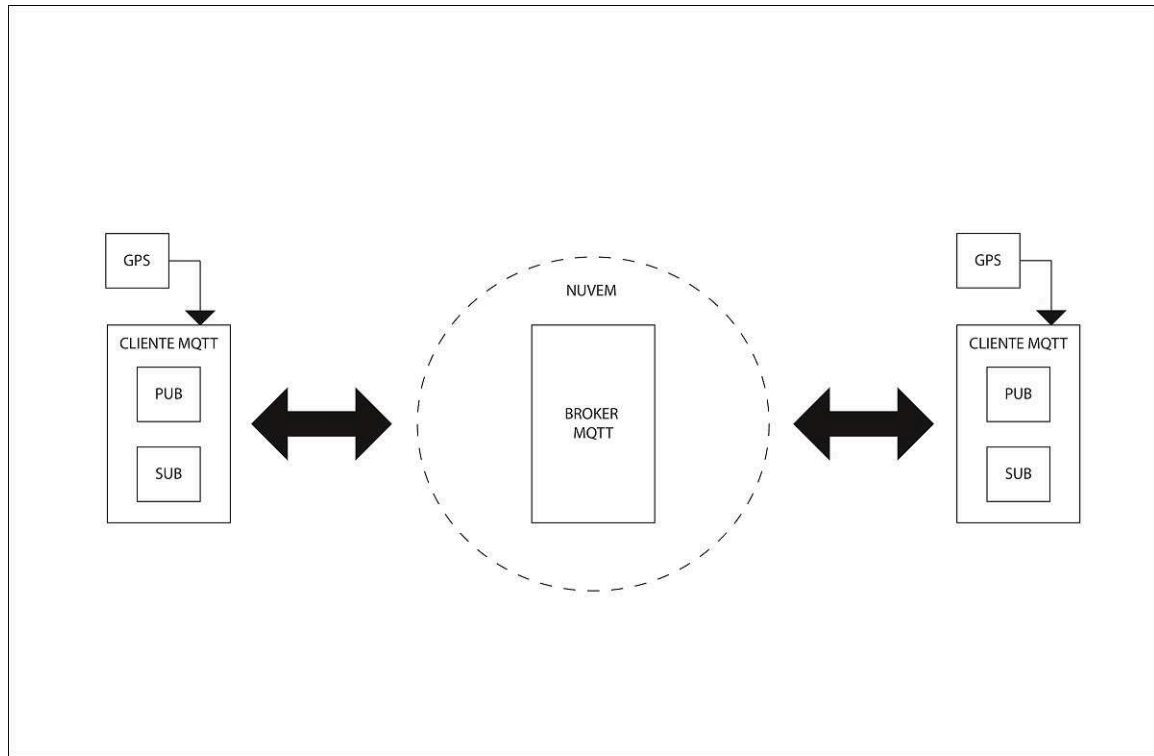
No experimento tanto os clientes MQTT A e B possuem as funções de *publisher* e *subscriber* na rede. O cliente A publica no tópico A o qual o cliente B é *subscriber*, imediatamente depois do cliente MQTT B receber a mensagem do tópico A ele publica no tópico B, o qual o cliente MQTT A é *subscriber*. Dessa forma foi obtida a latência *round trip* de cada pacote enviado. Os clientes MQTT foram implementados por meio de dispositivos IoT. O primeiro dispositivo foi um IoT2040 da *Siemens*, ele é um dispositivo IoT que utiliza o *Intel Quark x1020*, possui 1GB de RAM, 2 portas *Ethernet*, 2 interfaces RS232/485 e sistema operacional *Yocto Linux*. O outro dispositivo utilizado foi um computador industrial da *Intel* que utiliza o *Intel i3-5000*, possui 4 GB de RAM e sistema operacional *Windows 7*.

Para o experimento, o dispositivo IoT2040 foi colocado no campus da Universidade de Brescia (Itália) e o computador industrial na Escola de Engenharia de São Carlos (Universidade de São Paulo – Brasil). Os dois dispositivos foram conectados por meio de internet de alta velocidade. O *broker* foi situado no Reino Unido. Utilizando essa estrutura simples, os autores obtiveram valores para a latência menores que 300 ms entre continentes e menor que 50 ms para o mesmo continente.

Dando continuidade ao interesse de averiguar a latência em comunicações intercontinentais, no trabalho de Ferrari et al. 2018 um *software* é proposto para validação da latência na transferência de dados de um ponto local a uma rota intercontinental.

O *software* é baseado em *Node-Red*, MQTT e no *Global Positioning System* (GPS). O GPS foi utilizado para uma boa sincronização e acuracidade nos valores obtidos de latência. A Figura 13 ilustra a arquitetura do experimento.

Figura 13 – Representação Gráfica da Aplicação



Fonte: Adaptado de (FERRARI et al., 2018)

A arquitetura utilizada reproduziu a situação de serviços baseados na *Industrial Internet of Things* (IIoT), ou seja, a informação coletada no campo foi enviada para o servidor na nuvem onde os dados foram processados e a decisão final tomada.

Os dispositivos utilizados para implementar os dois clientes MQTT foram o IoT2040 da *Siemens*, dispositivo embarcado criado pela *Siemens* para aplicações a nível de campo. Os dois clientes MQTT são sincronizados por meio do protocolo *Network Time Protocol* (NTP) equipados com módulos GPS (*Symmetricom Datum 6000*).

Para execução do experimento, um dos dispositivos IoT2040 foi colocado no campus da Universidade de Brescia (Itália) e o outro na Escola de Engenharia de São Carlos (Universidade de São Paulo – Brasil). Os dois dispositivos foram conectados por meio da internet de alta velocidade de cada universidade. No trabalho o $Qos = 1$ foi o que apresentou menor latência de ponto a ponto.

Seguindo com o mesmo foco de averiguar a latência na troca de mensagens entre dispositivos localizados em continentes diferentes, Oliveira et al. (2018), os autores avaliaram o desempenho dos protocolos MQTT e WebSocket na troca de mensagens a nível continental. O protocolo WebSocket foi desenvolvido para atender constantes troca de dados entre

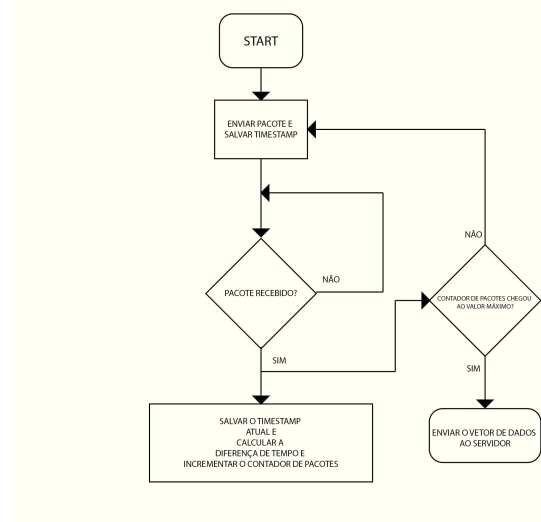
cliente e servidor não suportado anteriormente pelo HTTP. O protocolo consiste em um canal de comunicação bidirecional que funciona através de um único *socket* e possui comunicação assíncrona (em contraste com o protocolo HTTP), em outras palavras, ambos os lados podem enviar dados a qualquer momento enquanto a conexão for existente.

Tomando estes aspectos, os autores do trabalho tiveram como objetivo, usando ESP8266 (SoC com IEEE 802.11) e Servidores Node.js para troca de dados, medir a latência calculando a diferença de tempo entre pacotes trocados entre um servidor e um ESP8266 usando ambos os protocolos.

Para comparação, foram considerados aspectos qualitativos (RTT e documentação) e testes quantitativos. Os testes quantitativos foram realizados no Laboratório de Informática Industrial (LII), localizado no Núcleo de Pesquisa e Inovação em Tecnologia da Informação (nPITI), com um dispositivo ESP8266 e um computador para ser o servidor, os únicos conectados à rede local, criada pelo roteador TP-Link TL WR 741 ND (Roteador sem fio N 150Mbps). O computador, usado como servidor, foi um Samsung NP270E4E, com CPU Intel (R) Celeron (R) 1007U e 4Gb de RAM e sistema operacional Ubuntu 17.10. Os servidores foram escritos em linguagem de programação Javascript usando o Node.js.

Os dispositivos utilizados foram alocados em continentes diferentes, um foi posicionado na Itália e outro no Brasil. A Figura 23 ilustra o algoritmo utilizado para medição da latência.

Figura 14 – Algoritmo Utilizado para Medição da Latência



Fonte: Adaptado de (OLIVEIRA et al., 2018)

O algoritmo foi executado no servidor e no dispositivo ESP8266. Uma maior incerteza nas medições foi observada nas medições realizadas no servidor, em comparação com as medições realizadas no dispositivo, analisando o desvio padrão da média das amostras.

Portanto, o dispositivo ESP8266 foi escolhido para calcular o *Round Trip Time* (RTT) e enviar os dados para o servidor no final dos cálculos.

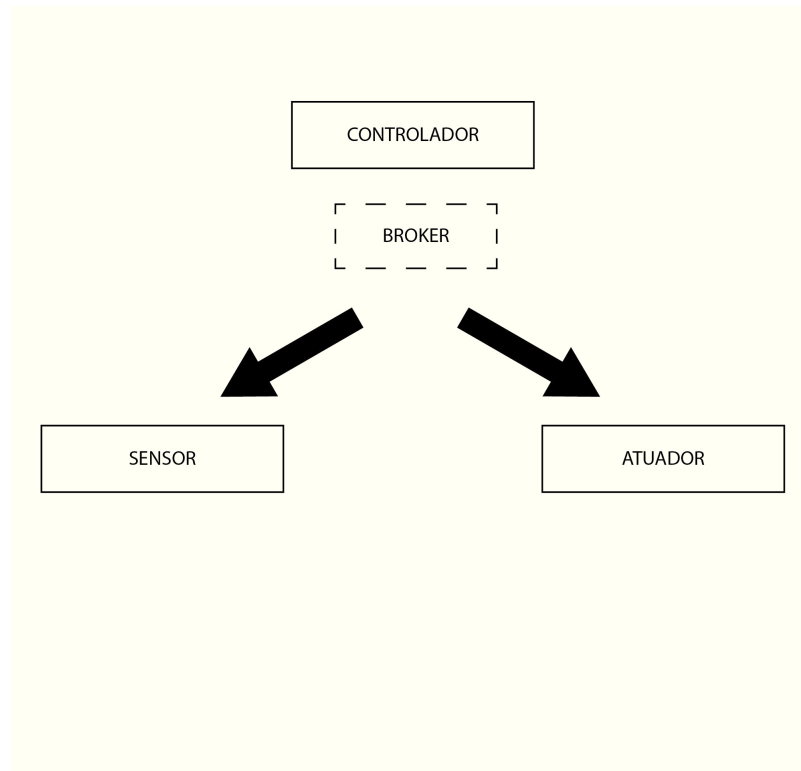
Os valores médios do tempo de entrega fim-a-fim usando os dois protocolos para os caminhos Brasil-Itália-Brasil e Itália-Brasil-Itália foram abaixo de 300 milissegundos. Considerando somente uma direção, os valores de tempo obtidos foram abaixo de 150 milissegundos. Para os dois protocolos de troca de mensagem, os valores médios de latência obtidos foram semelhantes para os tamanhos de pacotes analisados. Sendo assim, Através do trabalho os autores verificaram que ambos os protocolos de comunicações são promissores para serem usadas em projetos IoT. Ambos possuem documentação com exemplos, descrição de métodos e limitações das bibliotecas.

No entanto, a biblioteca WebSocket ocupa mais memória do dispositivo que a biblioteca MQTT, e seu uso não é viável dependendo da aplicação. Em particular, aquelas em que o recurso de memória é muito restrito. Sob as condições de testes quantitativos, o protocolo WebSocket se mostrou mais apropriado para aplicações que exigem RTT abaixo de um milissegundo (OLIVEIRA et al., 2018).

Já no trabalho de Iglesias-Urkia et al (2018), os autores apresentam uma comparação teórica e empírica entre MQTT e CoAP em termos de latência e payload, com o intuito de ajudar projetistas de sistemas a tomarem a melhor decisão sobre qual protocolo de comunicação é mais adequado para o seu projeto. O protocolo CoAP foca na transferência de documentos web, similar ao protocolo HTTP. Porém, é voltado para dispositivos restritos, ou seja, que possuem recursos limitados.

Para comparar os protocolos, uma arquitetura que atende a ambos e permiti comparação foi projetada. A figura a seguir mostra a arquitetura utilizada para realização da parte experimental do trabalho.

Figura 15 – Arquitetura Utilizada para Avaliação dos Protocolos



Fonte: Adaptado de (IGLESIAS-URKIA et al., 2017)

Na Figura 15 três agentes podem ser identificados no cenário apresentado: sensores, atuadores e controladores. Os sensores, atuadores são dispositivos que estão encarregados de interagir diretamente com o mundo físico. Os controladores, estando acima na pirâmide de automação, comunicam com sensores e atuadores com o intuito de gerenciar suas tarefas ou coleta de dados.

Respeitando essa hierarquia, os autores realizaram uma análise comparando *payload*, escalabilidade e a latência dos protocolos MQTT e CoAP para diferentes Qos. Como a rede utilizada não apresentou perdas por congestionamento, a diferença no parâmetro *payload* não foi grande, no entanto, a diferença foi maior em termos de latência. Enquanto as latências do MQTT foram na ordem dos milissegundos, no CoAP elas foram na casa das centenas de microssegundos. Essa diferença é fortemente afetada pelos protocolos subjacentes da camada de transporte. Sob o tráfego da rede, o CoAP provou ser mais leve e rápido que o MQTT. Mesmo assim, o MQTT é capaz de cumprir as especificações de tempo e *payload* sem nenhum problema, além do mais, a sua fácil configuração e desacoplamento de *publishers/ subscribers*, entre outras características, são facilidades que devem ser levados em conta ao se projetar uma aplicação no ambiente IoT (IGLESIAS-URKIA et al., 2017).

Já no trabalho de Rocha et al (2018), os autores realizaram a comparação entre dois protocolos bem conhecidos no campo da Internet das Coisas e Internet das Coisas Indus-

trial (IIoT): o MQTT e OPC UA. A comparação foi realizada com base na quantidade de recursos consumidos por cada protocolo para o envio de mensagens e no tempo gasto para envio e recebimento de mensagens.

Considerando os resultados apresentados pelos testes, foi inferido que o MQTT tem a vantagem de utilizar menos dados para transmitir o mesmo *payload*, além de ter tempos de transmissão ligeiramente menores. A maior diferença se encontrou no fato de que na transmissão de uma mensagem para vários clientes, o protocolo MQTT apresentou uma grande vantagem no envio de mensagens (ROCHA et al., 2018). No entanto, não se pode depreciar a magnitude do Protocolo OPC UA, que tem vários outros serviços além da troca de dados, no trabalho o modelo de troca de dados foi limitada apenas ao modelo de *publisher/ subscriber*. Serviços como modelagem de dados, espaço de endereçamento, gerenciamento de alarmes e eventos, histórico variável, controle de acesso, entre muitos outros serviços oferecidos.

3.3 Considerações Finais

Diante dos trabalhos apresentados, fica claro que existem trabalhos que exploram a utilização de diversas tecnologias de infraestrutura IoT, como MQTT e CoAP no ambiente industrial da Internet das Coisas, entretanto, os cenários onde o protocolo MQTT é aplicado são cenários simples, em uma rede IoT é esperado ter uma quantidade grande de nós, quanto maior a quantidade de nós nos cenários, mais preciso são os resultados obtidos.

Somado ao fato de os cenários de aplicação serem simples, são escassos os trabalhos que tratam da aplicação da tecnologia MQTT aos sistemas robóticos industriais dentro do cenário da Internet das Coisas. A aplicação do protocolo MQTT auxilia no desenvolvimento de aplicações para o ambiente industrial IoT, pois padroniza a comunicação de dispositivos e aplicações no ambiente da Internet das Coisas por meio de um protocolo padrão de troca de mensagens, a padronização na comunicação diminui o tempo de desenvolvimento do sistema em termos de custos e horas de programação gastas no desenvolvimento de qualquer sistema.

Sendo assim, o trabalho proposto busca integrar um sistema de controle e monitoramento IoT baseado em MQTT a área da robótica industrial e estimar, a nível local, a eficiência na comunicação entre os equipamentos.

O próximo capítulo apresenta os materiais e o procedimento proposto para execução do trabalho.

x

Solução Proposta e Metodologia

Neste capítulo, são apresentadas as etapas do trabalho para determinação dos parâmetros analisados e os materiais utilizados no processo. São apresentadas explicações detalhadas, rigorosas e exatas de toda ação desenvolvida no trabalho de pesquisa. As explicações englobam desde formas de tabulação até o tratamento dos dados.

Para avaliação da performance do sistema IoT proposto, uma rede baseada no modelo de 3 camadas cabeada, composta por 25 nós sensores virtuais MQTT e 4 agentes físicos MQTT foi constituída.

Através das fórmulas 1, 2, 3 os parâmetros métricos de performance da rede, latência e taxa de transmissão de pacotes foram calculados e para uma maior acuracidade nas medições os elementos da rede utilizados mais diretamente nas medições, no caso o *publisher* P1 e o braço robótico, foram sincronizados por meio do protocolo PTP.

O processo de sincronização inicia com uma mensagem de sincronização de tempo enviado para a rede. Todos os componentes (*slaves*) calculam a diferença de tempo (*delay*) entre seu relógio local e o enviado pelo *master* e então a diferença é compensada através da troca de uma série de mensagens entre o relógio *master* e os relógios dos *slaves*. Após 40 minutos de sincronização entre o braço robótico e o *publisher* P1, os valores para os cálculos dos parâmetros de performance do sistema proposto foram obtidos.

A grande vantagem da utilização desse protocolo é a sincronização de tempo que ele oferece entre equipamentos de diferentes fabricantes via protocolo padronizado. A sincronização dos elementos garante uma maior acuracidade nas medições realizadas, pois o relógio dos dispositivos se encontram sincronizados.

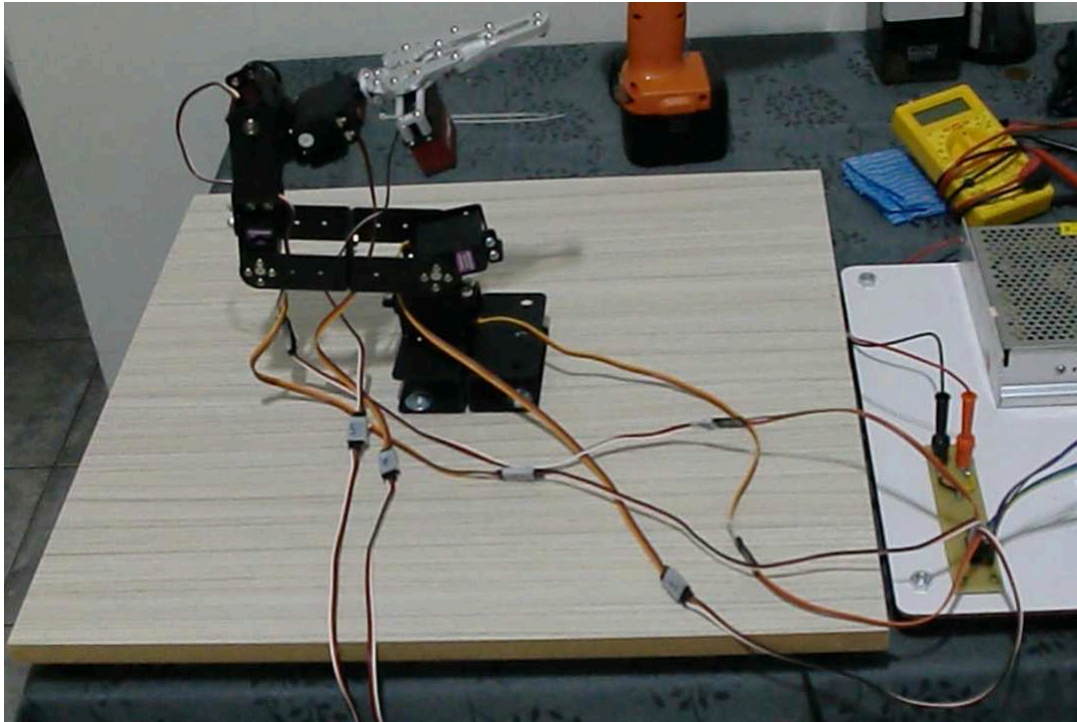
4.1 Materiais

No trabalho as placas *BeagleBone Black*, *Arduino UNO* e o *shield ethernet W5100* foram utilizados na montagem da parte experimental. A *BeagleBone Black* é uma plataforma de desenvolvimento de baixo custo desenvolvida pela *Texas Instruments*, classificada como *hardware* livre por meio da licença *Commons SharedAlike* (COLEY, 2013).

Já o microcontrolador *Arduino* utilizado na versão *UNO* é da família *Atmel AT-MEGA16U2*. O *Ethernet Shield W5100* é um dispositivo que permite ao *Arduino UNO* conectar de forma rápida e fácil permitindo o envio de informações remotamente (D'AUSILIO, 2012).

O braço robótico, Figura 16), utilizado no trabalho possui 5 graus de liberdade e uma garra.

Figura 16 – Braço Robótico



Fonte: Próprio Autor

Ele foi montado utilizando servos motores *Tower Pro MG946* e peças metálicas que podem ser combinadas de acordo com a necessidade do usuário.

4.2 *Node-Red*

O *Node-Red* é uma ferramenta para prototipação de *software* baseado em sistema embarcado e IoT (FERRARI et al., 2017). Esta ferramenta utiliza o conceito de programação gráfica para conectar hardware de dispositivos, *Application Programming Interface* (APIs) e serviços online como parte da Internet das Coisas. (BLACKSTOCK; LEA, 2014).

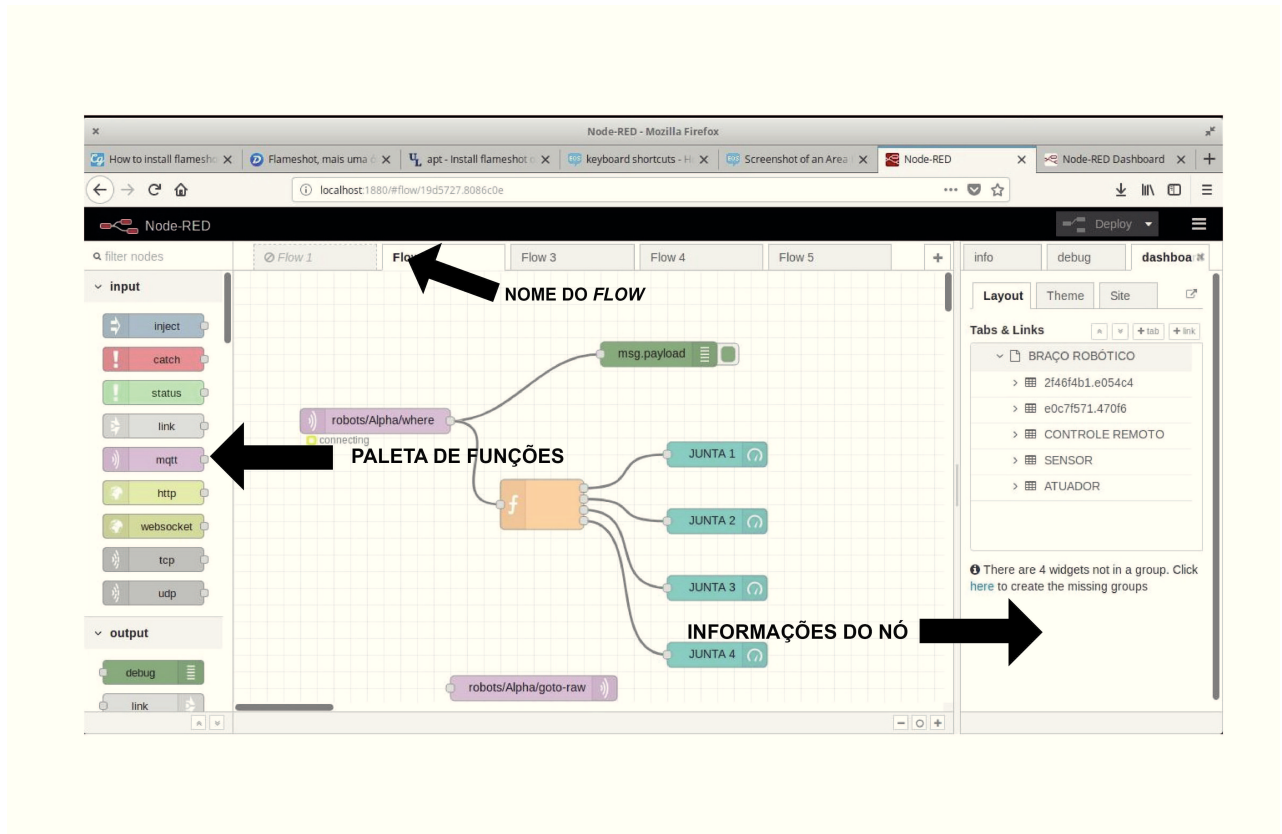
A programação gráfica acontece por meio do estabelecimento da conexão entre os blocos (nós) que possuem códigos pré-definidos. Este tipo de programação auxilia o usuário que não possui domínio sobre as ferramentas e conceitos sobre as estruturas utilizadas no desenvolvimento de sistemas embarcados (*Arduino UNO*, *BeagleBone* e etc)

a desenvolverem aplicações visando o ambiente da Internet das Coisas (FERRARI et al., 2017).

Apesar de o *Node-Red* utilizar a programação gráfica, é possível desenvolver uma vasta gama de aplicações avançadas de maneira fácil e rápida. A edição da programação é feita por meio do navegador de internet (*browser*). As aplicações (*flows*) são construídas por meio da escolha dos nós necessários por meio de *drag and drop*, estabelecimento da conexão entre eles e configuração dos nós *Javascript*. Além do mais, a plataforma possui mais de 225 mil módulos de pacotes para extensão das capacidades de recursos da paleta de funções (BLACKSTOCK; LEA, 2014).

A figura a seguir ilustra o ambiente de desenvolvimento do *Node-Red*.

Figura 17 – Ambiente ferramenta *Node-Red*



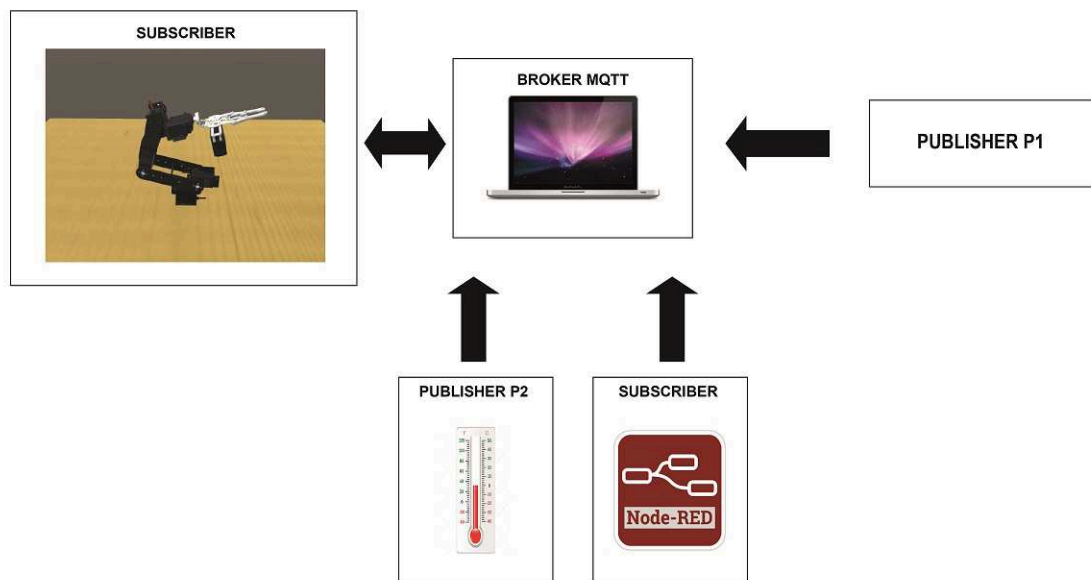
Fonte:Próprio Autor

Por ser baseado em *Node.js* o *Node-Red* pode ser utilizado no topo da camada de comunicação das redes com um baixo custo de *hardware*. Essa característica de baixo custo de *hardware* faz com que a ferramenta seja perfeita para ser utilizada com sistemas embarcados, como(*Arduino, BeagleBone e Raspberry* (FERRARI et al., 2017).

4.3 Metodologia

Para avaliação da performance do sistema proposto, uma rede baseada no modelo de arquitetura IoT de 3 camadas, cabeada composta por 4 agentes MQTT foi constituída e 25 nós sensores virtuais foram simulados. Os nós sensores foram utilizados para dar mais veracidade do cenário industrial laboratorial. Esses nós sensores publicam mensagens periodicamente e com níveis de Qos= 0 e 1. A figura a seguir ilustra a rede física constituída para testes.

Figura 18 – Rede IoT utilizada



Fonte: Próprio Autor

Como se pode observar através da Figura 18, a rede é composta por 2 agentes *publisher* e 2 agentes *subscriber* MQTT. Cada agente da rede tem a seguinte função:

1. **Publisher P1**: responsável por publicar as posições das juntas a serem atualizadas;
2. **Publisher P2**: responsável pela publicação do valor do sensor de temperatura;
3. **Subscriber**: responsável por atualizar os valores das juntas do braço robótico;
4. **Node-Red**: fornece uma interface de controle e visualização de informações a cerca do braço robótico e da temperatura do sensor.

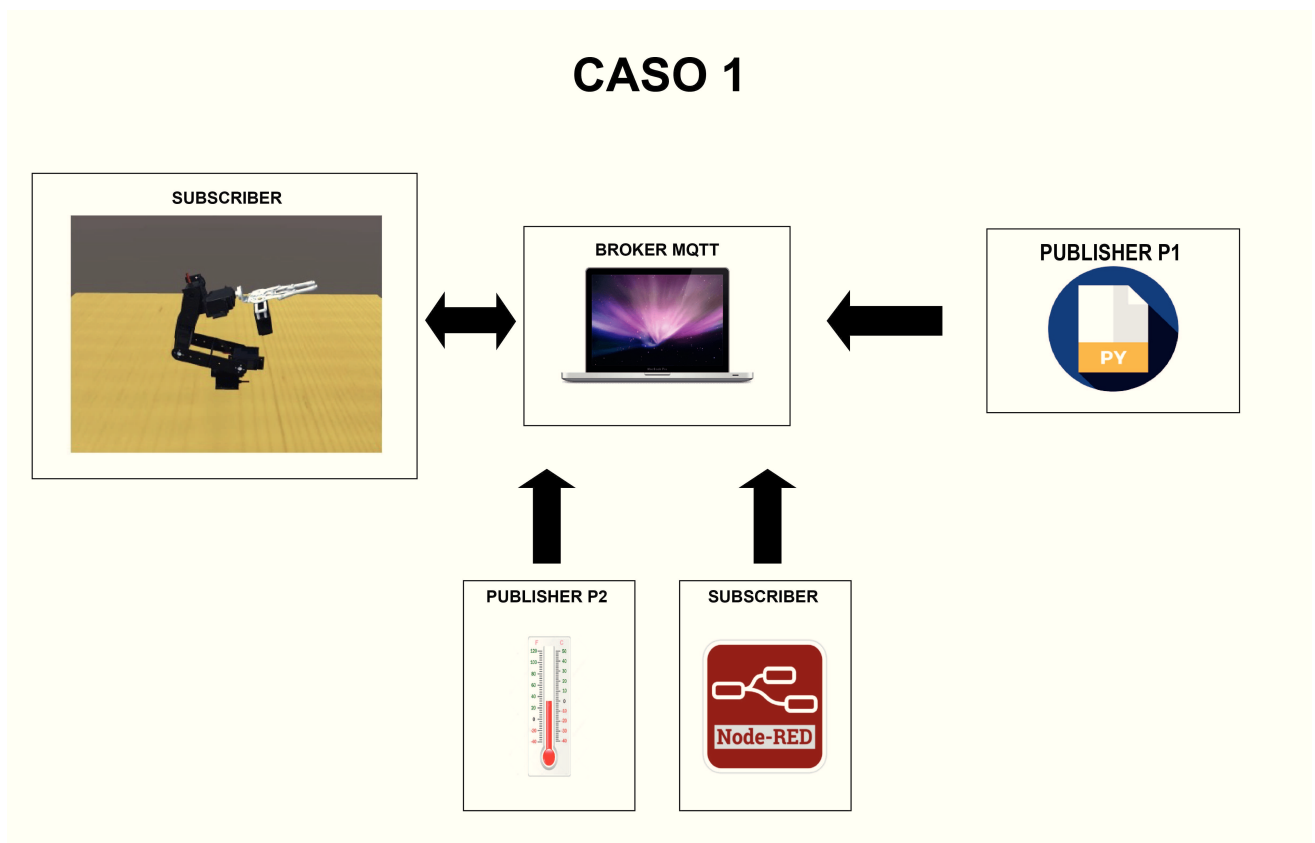
Para avaliação da performance da rede, dois casos foram estabelecidos. O que mudou de um caso para o outro foi o agente responsável por controlar o braço robótico e publicar

as mensagens no tópico referente ao cálculo da latência no braço robótico. A escolha por dois cenários de estudo aconteceu devido o interesse de verificar o comportamento do controle do braço robótico, nos casos em que o usuário do sistema opte por utilizar um *publisher* escrito em linguagem de programação tradicional, no caso a linguagem python, e no caso em que opta por utilizar a linguagem gráfica, que apresenta a vantagem de facilitar o desenvolvimento de aplicações para o programador. Somado a isso, também o interesse de verificar se essa escolha acarreta algum efeito sobre o controle do braço robótico.

No cenário 1, Figura 19, o usuário é responsável por configurar a trajetória a ser desempenhada pelo o braço robótico por meio de um *script* em python.

Este *script* é responsável por publicar as informações de atualização do braço robótico utilizadas para o controle do mesmo. Essas informações também foram utilizadas para cálculo da latência na comunicação do sistema IoT proposto baseado na arquitetura *publisher/subscriber*.

Figura 19 – Rede IoT do caso 1



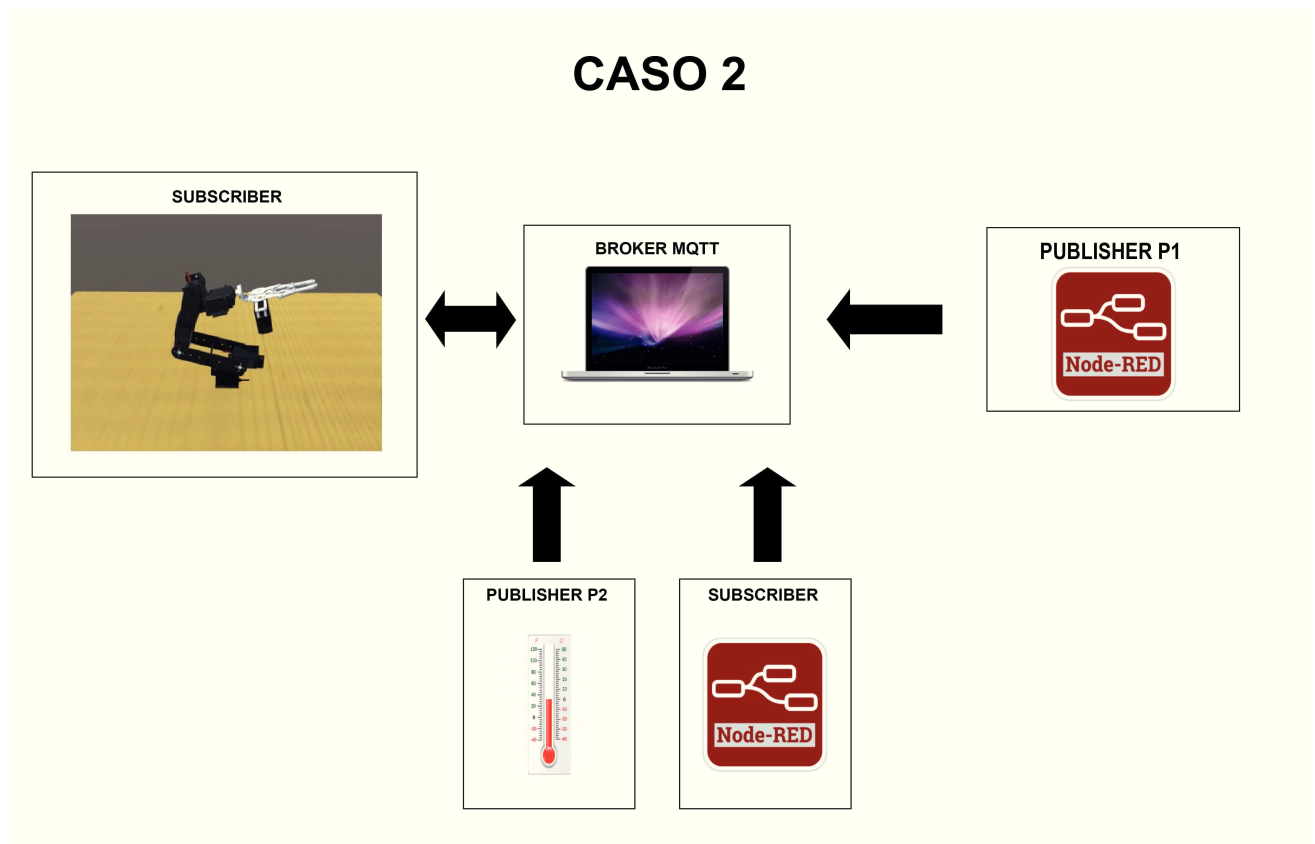
Fonte: Próprio Autor

Cada agente MQTT da rede física, tem as seguintes funções:

1. **Publisher P1**: responsável por realizar o controle do braço robótico por meio do tópico *robot/Alpha/lat*, sua função é semelhante ao controlador de um braço robótico, ele realiza a trajetória pré-programada pelo o usuário;
2. **Publisher P2**: tem como função publicar a informação acerca da temperatura do ambiente a cada 12 ms. As suas publicações tem como objetivo sobrecarregar a rede com tráfego de informações para que os valores obtidos por meio dos testes sejam os mais próximos possíveis de um caso real;
3. **Subscriber**: responsável por receber e atualizar os valores das juntas do braço robótico;
4. **Node-Red**: tem como função oferecer uma interface amigável na qual os usuários do sistema possam ter acesso as informações do braço robótico por meio de um desktop.

Já no caso 2, Figura 20, o que difere para o caso 1 é o responsável pelo controle do braço robótico.

Figura 20 – Rede IoT do caso 2

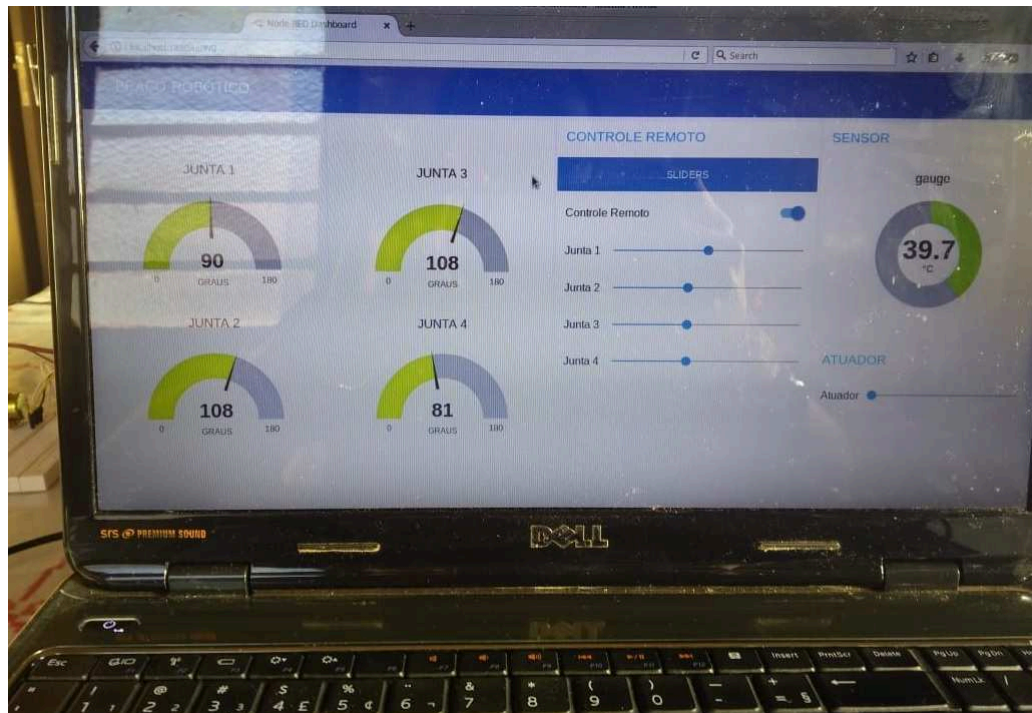


Fonte: Próprio Autor

Como se pode observar através da Figura 20, a aplicação web baseada Node-Red torna responsável por publicar as informações acerca do braço robótico. Sendo assim, o usuário é responsável por controlar diretamente o braço robótico, esse controle permite que o usuário realize manobras corretivas ou preventivas em situações que oferecem risco para os demais funcionários presentes no local. As informações de controle enviadas pela aplicação web, foram utilizadas para atualizar para o cálculo da latência da comunicação no canal aplicação web braço robótico.

No caso 2, como mostra a Figura 21, a aplicação web além de controlar o braço robótico, tem função de monitorar o braço robótico, sendo assim responsável por controlar o braço robótico e por também oferecer uma interface na qual os usuários do sistema possam ter acesso as informações do braço robótico por meio do *browser* de qualquer *desktop*.

Figura 21 – Aplicação Web



Fonte: Próprio Autor

Para ambos os casos, todas as trocas de mensagens entre *publisher* P1 e braço robótico foram configuradas com o nível de Qos = 2 para aumentar o nível de tráfego de mensagens na rede e verificar a latência do sistema na condição de troca de mensagens mais exigente. Em adição, com o intuito de tornar o cenário o mais próximo de um cenário real, 25 nós sensores virtuais foram criados com o intuito de congestionar o tráfego da rede, esses sensores publicam mensagens para o *Broker* periodicamente com o mesmo intuito de aumentar a quantidade de pacotes que trafegam pela rede a *BeagleBone Black* publica periodicamente no tópico *robot/Alpha/joints*, referente a posição atual do braço robótico.

Para o cálculo dos parâmetros de performance, em ambos os casos, imediatamente antes do *publisher* P1 enviar mensagem para o *Broker*, o *publisher* P1 adiciona em um arquivo do tipo *Comma-Separated Values* (CSV), uma linha contendo o valor do contador de pacotes enviados e um número do tipo de dado flutuante denotando a hora ou data que certo evento ocorreu (*timestamp*) em milissegundos. O formato de arquivo CSV foi usado devido a sua simplicidade de leitura e escrita. Quando a mensagem chega ao seu destino, no caso, a *BeagleBone Black* do braço robótico, acrescenta em um arquivo uma *string* de dados contemplando o número do pacote recebido com seu respectivo *timestamp*. E em adição, acrescenta o *timestamp* em milissegundos do pacote recebido.

Para que o *timestamp* e número de identificação do pacote pudessem trafegar pela rede, foram adicionados ao tópico *robots/Alpha/lat* dois campos. Sendo assim o corpo da mensagem do tópico *robots/Alpha/lat* possuiu o seguinte formato:

$$\text{message} = (\theta_1, \theta_2, \theta_3, \theta_4, id_{PACOTE}, t_{PUBLISHER})$$

O $t_{PUBLISHER}$ é um tipo de dado ponto flutuante referente ao *timestamp*, id_{PACOTE} é do tipo inteiro e armazena o valor de identificação do pacote, $\theta_1, \theta_2, \theta_3, \theta_4$ são referentes ao valor de posição das juntas do braço robótico e são do tipo ponto flutuante.

A Tabela 1, ilustra o formato do arquivo CSV gerado a cada chegada de pacote na *BeagleBone Black*.

Tabela 1 – Exemplo da tabela gerada pelo *subscriber*

id_{PACOTE}	$T_{PUBLISHER}$	$T_{SUBSCRIBER}$
...
2500	1532024879.755643	1532024879.759628
...

Fonte: Próprio Autor

Como mostra a tabela acima, o arquivo gerado possui três colunas referentes ao número de identificação do pacote, o seu *timestamp* de publicação e o seu *timestamp* de chegada a controladora do braço robótico, no caso, a *BeagleBone Black*.

A primeira coluna, id_{PACOTE} , representada por um inteiro, teve como função auxiliar na contagem do número de pacotes enviados pelo *publisher* ao braço robótico. O número de pacotes total enviados foi utilizado no cálculo da taxa de transmissão média. As colunas $T_{PUBLISHER}$ e $T_{SUBSCRIBER}$, responsáveis por armazenar os valores de *timestamp*, foram utilizados para o cálculo da latência existente no sistema IoT. A diferença entre os dois valores expressam a latência obtida para cada pacote enviado pelo o *PUBLISHER* P1.

4.3.1 Parâmetros de Performance

Utilizando os dados dos arquivos CSV gerados no caso 1 e caso 2, a Taxa de Transmissão de Pacotes (TTP) pôde ser calculada da seguinte forma:

$$TTP = \frac{id_{FINAL} - id_{INICIAL}}{t_{FINAL} - t_{INICIAL}} \quad (1)$$

Os valores t_{FINAL} e $t_{INICIAL}$ representam os valores de *timestamp* em milissegundos do último e primeiro pacotes enviados e id_{FINAL} e $id_{INICIAL}$ os identificadores dos pacotes final e inicial. A taxa de transmissão de pacotes oferece uma medida para a velocidade da conexão. A Taxa de transmissão de pacotes é comparável ao fluxo de tráfego em uma via, quanto maior a auto-estrada e quanto maior for o limite de velocidade, mais o tráfego se move através.

Assim como o fluxo de tráfego, que pode ser afetado por obstruções, acidentes e estradas mal projetadas, a taxa de transferência de dados está sujeita a inúmeras interferências externas.

A latência de cada pacote foi calculada através da seguinte fórmula:

$$\Delta t(K) = t_{SUB}(K) - t_{PUB}(K) \quad (2)$$

Onde t_{PUB} é o instante de tempo registrado antes do envio do k-ésimo pacote e t_{SUB} é o *timestamp* registrado após a chegada do pacote ao braço robótico. A fórmula é uma simples subtração dos valores referentes ao *timestamp* de envio e *timestamp* de chegada de cada pacote ao braço robótico.

A latência média para cada caso foi calculada pela seguinte fórmula:

$$Latência_{MÉDIA} = \frac{\sum_{i=0}^K \Delta t(K)}{n} \quad (3)$$

A fórmula se resume a uma somatória das latências obtidas para cada pacote, dividida pelo número total de pacotes enviados. A latência média obtida pode ser definida como o valor que mostra para onde se concentram os dados de uma distribuição. Ela também é interpretada como um valor significativo de uma lista de números. Os valores de uma lista de números podem ser representados por meio da escolha aleatória de um número. Se todos os números forem iguais, o número escolhido aleatoriamente será a média. Então, a média pode ser calculada por meio da combinação dos números de maneira específica e da geração de um valor significativo.

Através do cálculo do parâmetro de latência média, se tornou possível o conhecimento do tempo médio entre o início de um evento que no trabalho foi o envio da mensagem pelo *publisher* P1, e o momento em que os seus efeitos se tornam perceptíveis que no trabalho foi a atualização da posição do braço robótico. Em alguns casos, a latência é medida através do envio de um pacote que é devolvido para o remetente e o RTT é considerado a latência (*ping*).

4.4 Considerações Finais

Neste capítulo, foi apresentado o procedimento para viabilizar a construção e análise de performance da rede. O próximo capítulo descreve os resultados obtidos com a aplicação dos procedimentos descritos no sistema IoT proposto.

Resultados

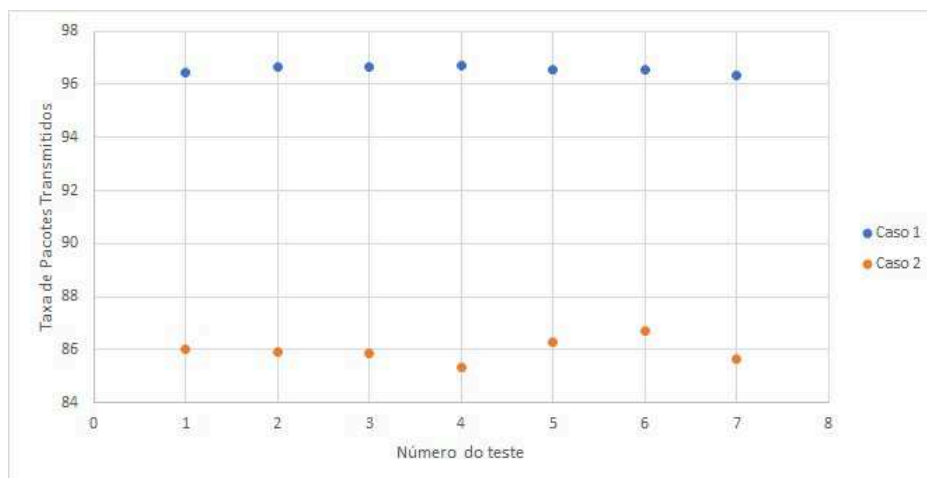
Para os dois cenários, o tempo de cada teste durou em torno de 49 segundos, durante este tempo o *publisher* P1 enviou mensagens ao tópico referente ao cálculo da latência, a taxa de 100 pacotes por segundo, gerando arquivos com 4900 pacotes em média. Para uma maior confiabilidade do valor médio de latência e de transmissão de pacotes, para cada caso, os testes foram repetidos 7 vezes para maior confiabilidade nos valores obtidos.

O tamanho da mensagem transmitida entre o *publisher* P1 e o *Broker* foi de 4 bytes.

5.1 Taxa de Transmissão de pacotes do publisher P1

Utilizado a Fórmula 1 para o cálculo da taxa de transmissão de pacotes descrita na metodologia, as taxas médias obtidas para os dois casos ficaram na faixa de 85.68 pacotes/segundo e 96.70 pacotes/segundo. A Figura 22, elucida o que foi dito no período acima.

Figura 22 – Taxa de Pacotes Transmitidos



Fonte: Próprio Autor

As taxas de transmissão de pacote obtidas em cada teste, refletem a quantidade de pacotes recebidos pelo braço robótico nos dois casos considerados. A taxa padrão de publicação de pacotes nos dois cenários foi de 100 pacotes/segundo e a taxa de atualização de cada servomotor é de 50 pacotes/segundo. A taxa de atualização foi descoberto através da divisão do ciclo de operação de 20 ms do servo motor, descoberto através do *datasheet*, por 1 segundo.

Através da Figura 22, foi possível mostrar que para o sistema IoT proposto baseado na arquitetura *publisher/subscriber* do protocolo MQTT, as taxas de transmissão de pacotes obtidas em ambos os casos foram satisfatórias, pois em ambas situações as taxas de transmissão média de pacotes são maiores que o necessário, garantido que o braço robótico opere sem perdas significativas na atualização de sua trajetória.

Se porventura as taxas de transmissão de pacotes obtidas fossem menores do que 50 pacotes/segundos, essas taxas menores abririam uma brecha para que falhas na trajetória do braço ocorressem caso fosse necessário a sua operação no máximo de sua capacidade. E uma falha na trajetória ocasionaria perda de resolução do braço robótico que resultaria em retrabalho e retrabalho gera perda de produtividade e desperdício de matéria-prima.

5.2 Latência Média da Rede

A latência pode ser definida como o tempo que leva para um sinal (bit) ou pacote de dados ser enviado do computador original até ao seu destino. Também pode ser definida como um intervalo de tempo entre estimulação e resposta, de forma mais específica, um intervalo de tempo entre a causa e o efeito de alguma mudança física no sistema que está sendo observado. A latência é fisicamente uma consequência da velocidade limitada com a qual qualquer interação física pode se propagar. O valor dessa velocidade de propagação é sempre menor ou igual à velocidade da luz. Consequentemente, todo sistema físico experimentará algum tipo de latência, independentemente da natureza da estimulação a que foi exposto (FERRARI et al., 2017).

A definição precisa de latência depende da natureza da estimulação e do sistema que está sendo observado. Em comunicações, o limite inferior de latência é determinado pelo meio que está sendo usado para comunicações. Em sistemas de comunicação bidirecional, a latência limita a taxa máxima que a informação pode ser transmitida, já que muitas vezes há um limite na quantidade de informação que pode transitar pela rede em qualquer momento. No campo da interação homem-máquina, a latência tem um forte efeito na satisfação e usabilidade do usuário. Os jogos online são sensíveis à latência (CALDIÉRI, 2016).

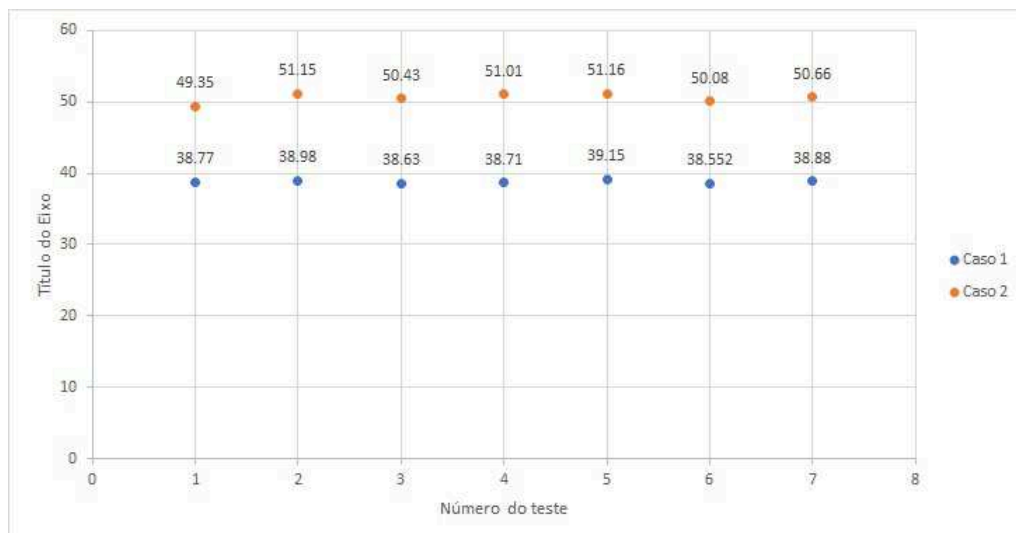
Um jogador com uma conexão de internet de alta latência pode mostrar respostas lentas apesar do tempo de reação apropriado. Isso dá aos jogadores com conexões de baixa latência uma vantagem técnica. A minimização da latência é de grande interesse

no mercado de capitais, mais precisamente na parte do mercado onde a negociação algorítmica é usada para processar atualizações de mercado e dar ordens em milissegundos. A negociação de baixa latência ocorre nas redes usadas pelas instituições financeiras para conectar-se a bolsas de valores e redes de comunicação eletrônica (ECNs) para executar transações financeiras. A latência pode ser causada por muitos fatores inerentes ao ambiente, como problemas Ethernet, congestão da rede, longas distâncias, perdas de pacotes ou mesmo aplicações mas escritas (FERRARI et al., 2017).

E como sabemos o protocolo de comunicação MQTT é um protocolo desenvolvido com base na pilha TCP/IP, e por ser baseado no TCP, a rede sempre se preocupa sobre o estado das conexões e para tal o lado que recebe dados precisa enviar regularmente pacotes confirmando a recepção de pacotes anteriores. Quem está enviando necessita de receber estas confirmações (conhecidas como *acknowledgements*). Se levarmos em conta que os link de comunicações nunca são perfeitos, os efeitos destas imperfeições têm um impacto maior quanto maior for a latência da ligação, sendo assim, qualquer valor de latência muito alta causará um efeito nocivo sobre as sessões estabelecidas, a latência alta fará com que as sessões da rede se iniciem de forma lenta e prossigam de forma mais lenta do que uma rede onde a latência é baixa (YACCHIREMA; ESTEVE; PALAU, 2017).

A Figura 23 traz os resultados obtidos de latência para os dois casos estudados no trabalho.

Figura 23 – Latência



Fonte: Próprio Autor

A Figura 23, traz a representação da latência obtida pelos pacotes enviados pelo *publisher* P1 ao braço robótico em cada teste realizado. No gráfico o eixo das ordenadas representa o valor da latência medida para o referente teste, o número do teste é representado pelo valor inteiro do eixo das abscissas.

Como se pode perceber através do gráfico, para o caso 1, a latência se manteve em torno do valor de 38 ms. O pico máximo e mínimo de latência foram de 39.15 ms e 38.552 ms. Levando em conta os 7 testes realizados, o valor médio de latência foi de 38.81 ms com desvio padrão de 0.192711 ms.

Para o caso 2, o gráfico mostra que os valores de latência obtidos, foram valores próximo, com picos máximo e mínimo de latências iguais a 51.16 ms e 49.35 ms. Levando em conta os 7 testes realizados, a latência média no segundo caso foi de 50.55 ms com desvio padrão de 0.613082 ms.

Para o sistema IoT proposto, quanto menor for a latência, menor será o tempo necessário para que um dado trafegue de um lugar para outro. Essa velocidade na resposta permite que o braço robótico seja gerenciado pelo sistema IoT proposto com eficiência e rapidez, caso as latências médias fossem altas, o desempenho do braço robótico seria atrapalhado, o que ocasionaria ao monitorar ou controlar o braço robótico uma certa lentidão no controle e monitoramento.

Essa lentidão causaria diminuição na confiança do sistema IoT proposto, pois se em um sistema IoT similar fosse aplicado ao setor hospitalar e ocorresse uma variação importante nos batimentos cardíacos num paciente de UTI, o monitor cardíaco desse paciente deve ativar um alarme em poucos segundos com a menor latência possível. Outro exemplo: se a conexão utilizada por um robô cirurgião apresentar alta latência, isso afetará os procedimentos cirúrgicos realizados pelo robô cirurgião o que afetará diretamente a qualidade do procedimento cirúrgico tornando inviável a sua utilização. Para esses sistemas é essencial a confiabilidade e a execução de tarefas em prazos compatíveis com a ocorrência de eventos externos.

Conclusão

Analisando do ponto de vista dos parâmetros de performance medidos, a rede IoT proposta baseada no protocolo de comunicação IoT MQTT se mostrou adequada e satisfatória para aplicação no cenário da robótica industrial da Internet das Coisas Industrial, pois o braço robótico utilizado no trabalho possui servomotores com capacidade de atualização de 50 pacotes/segundo e a rede proposta baseada em MQTT apresentou para os dois casos estudados taxas de transmissão de pacotes média para o braço robótico de 96.57 pacotes/segundo e 85.98 pacotes/segundo.

Estes valores são muito superiores ao exigido para controle do braço robótico de 50 pacotes/segundo, somado a isso, a rede proposta baseada em MQTT apresentou bons valores de latência. Um bom valor de latência, colabora para o bom desempenho do braço robótico, pois quanto maior a latência, mais devagar o braço robótico responde aos comandos dados e isso afeta a operação do braço robótico.

Olhando do ponto de vista de infraestrutura, a utilização de um sistema baseado em MQTT e na ferramenta Node-Red traz os benefícios das duas ferramentas para o setor robótico da Internet das Coisas. Esta ferramenta utiliza o conceito de programação gráfica para conectar o hardware de dispositivos, Interface de Programação de Aplicativos (APIs) e serviços online. Por utilizar a programação gráfica é uma ferramenta que facilita a qualquer pessoa o aprendizado e desenvolvimento de aplicações IoT, pois facilita a programação das aplicações para os usuários que possuem pouca experiência em programação.

Já o protocolo de comunicação MQTT permite que os equipamentos e objetos IoT se comuniquem e trabalhem cooperativamente e de forma simplificada. Por possuir uma codificação simples, suas linhas de código são pequenas e podem funcionar em sistemas precários, pois as suas mensagens não contêm dados além das informações necessárias, o que torna o processamento de informações mais ágil. O seu funcionamento é eficaz em redes de internet limitada e também em hardware de dispositivo com restrições.

Outros dois benefícios são que como o protocolo é de domínio público, ele pode ser modificado para atender às necessidades de suas operações e implementado onde você

desejar e toda a comunicação é feita por rede cabeada ou sem fio e vários equipamentos possuem suporte nativo para essa tecnologia.

Sendo assim o sistema proposto oferece facilidade para o compartilhamento de informações para aplicações ou desenvolvimento web, padronização na comunicação entre os dispositivos e oferece um sistema de gerenciamento de conteúdo ao usuário que funciona em qualquer browser utilizado, facilitando o controle e visualização de informações dos equipamentos e sensores.

A utilização de sistemas IoT em conjunto com a robótica tendem a trazer grandes benefícios para a área robótica, pois a aplicação desses sistemas IoT, fornecem uma infraestrutura de serviços compartilhada para o benefício da robótica industrial. Quando conectados, os robôs podem se beneficiar de poderosos recursos de computação, armazenamento e comunicação, pois o sistema pode coletar, processar e compartilhar informações de vários robôs ou outros objetos conectados a rede.

Além de dotar os sistemas robóticos de poderosas capacidades, a aplicação de tecnologias IoT reduzem o custo dos equipamentos. Assim, é possível construir robôs inteligentes leves, de baixo custo, e mais inteligentes por terem uma parte maior de processamento localizada nos sistemas IoT. Embora os robôs possam se beneficiar de várias vantagens dos sistemas IoT, existem muitas questões que ainda devem ser exploradas a fundo, como por exemplo, até que nível de dependência de tais sistemas não é prejudicial, pois se um robô depende muito de tais sistemas, uma falha na rede pode comprometer o seu funcionamento e tais sistemas também podem ficar lentos devido a problemas de rede.

Como trabalhos futuros, para uma avaliação mais acurada da performance da rede, se torna necessário a utilização de um relógio com uma melhor sincronização, pois o relógio utilizado no experimento foi o do próprio notebook. Um relógio mais preciso oferece um valor de medição mais próximo do real.

Outro estudo de relevância seria a inserção de mais elementos de controle a rede e também a associação da rede a uma rede industrial tradicional, por exemplo a *Modbus*, essa associação de uma rede IoT a uma rede tradicional traria o benefício de integrar o legado de uma rede tradicional muito utilizada nas indústrias devido a sua simplicidade e robustez ao contexto da Internet das Coisas e a inserção de mais elementos, provê um cenário de teste mais real.

Outro estudo interessante seria a aplicação da rede IoT baseada no protocolo de comunicação MQTT aqui apresentada em outros tipos de cenários, pois a comunicação baseada na arquitetura *publisher-subscriber* facilita a comunicação de dispositivos de fabricantes diferentes. A sua aplicação em outros cenários daria uma ideia da sua capacidade de atender requisitos além dos aqui exigidos.

Referências

- AL-QASEEMI, S. A. et al. IoT architecture challenges and issues: Lack of standardization. **FTC 2016 - Proceedings of Future Technologies Conference**, n. December, p. 731–738, 2017. Disponível em: <<https://doi.org/10.1109/FTC.2016.7821686>>.
- ARÉVALO, F.; DIPRASETYA, M. R.; SCHWUNG, A. A cloud-based architecture for condition monitoring based on machine learning. In: **2018 IEEE 16th International Conference on Industrial Informatics (INDIN)**. [s.n.], 2018. p. 163–168. ISSN 2378-363X. Disponível em: <<https://doi.org/10.1109/INDIN.2018.8471970>>.
- AYAD, S.; TERRISSA, L. S.; ZERHOUNI, N. An iot approach for a smart maintenance. In: **2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)**. [S.l. : s.n.], 2018.p.210 – –214.
- BANKS, A.; GUPTA, R. **OASIS Standard MQTT Version 3.1. 1**.
- BELLAGENTE, P. et al. Enabling PROFINET devices to work in IoT: Characterization and requirements. In: **2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings**. [S.l.: s.n.], 2016. p. 1–6.
- BELLAVISTA, P.; ZANNI, A. Towards better scalability for iot-cloud interactions via combined exploitation of mqtt and coap. In: **2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)**. [s.n.], 2016. p. 1–6. Disponível em: <<https://doi.org/10.1109/RTSI.2016.7740614>>.
- BLACKSTOCK, M.; LEA, R. Toward a distributed data flow platform for the web of things (distributed node-red). In: **ACM. Proceedings of the 5th International Workshop on Web of Things**. [S.l.], 2014. p. 34–39.
- CALDIÉRI, M. R. Implementação do modbus para aplicações de sistemas de controle via rede sem fio. Universidade Estadual Paulista (UNESP), 2016.
- COLEY, G. Beaglebone black system reference manual. **Texas Instruments, Dallas**, 2013.
- CUNHA, M. J. da et al. Proposal for an iot architecture in industrial processes. In: **IEEE. Industry Applications (INDUSCON), 2016 12th IEEE International Conference on**. [S.l.], 2016. p. 1–7.

- DIAZ-CACHO, M. et al. Iot integration on industrial environments. In: IEEE. **Factory Communication Systems (WFCS), 2015 IEEE World Conference on**. [S.l.], 2015. p. 1–7.
- DOLCI, R. Iot solutions for precision farming and food manufacturing: Artificial intelligence applications in digital food. In: IEEE. **Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual**. [S.l.], 2017. v. 2, p. 384–385.
- D'AUSILIO, A. Arduino: A low-cost multipurpose lab equipment. **Behavior research methods**, Springer, v. 44, n. 2, p. 305–313, 2012.
- EVANS, D. A Internet das Coisas Como a próxima evolução da Internet está mudando tudo. **Cisco**, p. 1–13, 2011. ISSN 1089-7801. Disponível em: <<https://doi.org/10.1145/2966986.2967034>>.
- FERRARI, P. et al. Delay Estimation of Industrial IoT Applications Based on Messaging Protocols. **IEEE Transactions on Instrumentation and Measurement**, v. 67, n. 9, p. 2188–2199, 2018. ISSN 00189456. Disponível em: <<https://doi.org/10.1109/TIM.2018.2813798>>.
- _____. Evaluation of communication latency in industrial IoT applications. **2017 IEEE International Workshop on Measurement and Networking, M and N 2017 - Proceedings**, 2017. Disponível em: <<https://doi.org/10.1109/IWMN.2017.8078359>>.
- GAN, S. et al. Iot based energy consumption monitoring platform for industrial processes. In: **2018 UKACC 12th International Conference on Control (CONTROL)**. [s.n.], 2018. p. 236–240. Disponível em: <<https://doi.org/10.1109/CONTROL.2018.8516828>>.
- GOMATHI, R. et al. A survey on iot technologies, evolution and architecture. In: IEEE. **2018 International Conference on Computer, Communication, and Signal Processing (ICCCSP)**. [S.l.], 2018. p. 1–5.
- HEMMATPOUR, M. et al. Diig: a distributed industrial iot gateway. In: IEEE. **Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual**. [S.l.], 2017. v. 1, p. 755–759.
- HUSNI, E. et al. Applied internet of things (iot): Car monitoring system using ibm bluemix. In: IEEE. **Intelligent Technology and Its Applications (ISITIA), 2016 International Seminar on**. [S.l.], 2016. p. 417–422.
- IGLESIAS-URKIA, M. et al. Towards a lightweight protocol for industry 4.0: An implementation based benchmark. In: **2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)**. [s.n.], 2017. p. 1–6. Disponível em: <<https://doi.org/10.1109/ECMSM.2017.7945894>>.
- KLYMASH, M. et al. Designing the industrial and environmental monitoring system based on the internet of things architecture. In: **2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)**. [s.n.], 2018. p. 187–190. Disponível em: <<https://doi.org/10.1109/IDAACS-SWS.2018.8525545>>.

LACERDA, F.; LIMA-MARQUES, M. Da necessidade de princípios de Arquitetura da Informação para a Internet das Coisas. **Perspectivas em Ciência da Informação**, v. 20, n. 2, p. 158–171, 2015. ISSN 1413-9936. Disponível em: <<https://doi.org/10.1590/1981-5344/2356>>.

LAMPKIN, V. et al. **Building smarter planet solutions with mqtt and ibm websphere mq telemetry**. [S.l.]: IBM Redbooks, 2012.

LUND, D. et al. Worldwide and regional internet of things (iot) 2014–2020 forecast: A virtuous circle of proven value and demand. **International Data Corporation (IDC), Tech. Rep**, v. 1, 2014.

MAINA, C. wa. Iot at the grassroots—exploring the use of sensors for livestock monitoring. In: IEEE. **IST-Africa Week Conference (IST-Africa), 2017**. [S.l.], 2017. p. 1–8.

MENDIRATTA, S.; DEY, D.; SONA, D. R. Automatic car parking system with visual indicator along with iot. In: IEEE. **Microelectronic Devices, Circuits and Systems (ICMDCS), 2017 International conference on**. [S.l.], 2017. p. 1–3.

OLIVEIRA, G. M. B. et al. Comparison between mqtt and websocket protocols for iot applications using esp8266. In: **2018 Workshop on Metrology for Industry 4.0 and IoT**. [s.n.], 2018. p. 236–241. Disponível em: <<https://doi.org/10.1109/METROI4.2018.8428348>>.

OTTONI, A. L. C. Introdução à Robótica. n. October, p. 6, 2010.

PATEL, K. K.; PATEL, S. M. Internet of Things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. **International Journal of Engineering Science and Computing**, v. 6, n. 5, p. 6122–6131, 2016. Disponível em: <<https://doi.org/10.4010/2016.1482>>.

PAZOS, N. et al. ConnectOpen - Automatic integration of IoT devices. **IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings**, p. 640–644, 2015. Disponível em: <<https://doi.org/10.1109/WF-IoT.2015.7389129>>.

ROCHA, M. S. et al. Performance comparison between opc ua and mqtt for data exchange. In: **2018 Workshop on Metrology for Industry 4.0 and IoT**. [s.n.], 2018. p. 175–179. Disponível em: <<https://doi.org/10.1109/METROI4.2018.8428342>>.

RUENGITTINUN, S.; PAISALWONGCHAROEN, J.; WATCHARAJINDASAKUL, C. Iot solution for bad habit of car security. In: IEEE. **Ubi-media Computing and Workshops (Ubi-Media), 2017 10th International Conference on**. [S.l.], 2017. p. 1–4.

SAHA, A. K. et al. IOT-based drone for improvement of crop quality in agricultural field. In: **2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)**. [s.n.], 2018. p. 612–615. Disponível em: <<https://doi.org/10.1109/CCWC.2018.8301662>>.

SANTOS, B. P. et al. Internet das coisas: da teoriaa prática. **Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribu{\i}dos**, 2016.

SHELBY, Z.; HARTKE, K.; BORMANN, C. **The constrained application protocol (CoAP)**. [S.l.], 2014.

SINGH, S.; SINGH, N. Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. **Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on**, p. 1577–1581, 2015. Disponível em: <<https://doi.org/10.1109/ICGCIoT.2015.7380718>>.

STANFORD-CLARK, A.; NIPPER, A. Mq telemetry transport. 2014.

STRADOLINI, F. et al. Live demonstration: An iot cloud-based architecture for anesthesia monitoring. In: IEEE. **Circuits and Systems (ISCAS), 2018 IEEE International Symposium on**. [S.l.], 2018. p. 1–1.

VERMESAN, O.; FRIESS, P. **Internet of things: converging technologies for smart environments and integrated ecosystems**. [S.l.]: River Publishers, 2013.

YACCHIREMA, D.; PALAU, C.; ESTEVE, M. Smart IoT Gateway For Heterogeneous Devices Interoperability. **IEEE Latin America Transactions**, v. 14, n. 8, p. 3900–3906, 2016. ISSN 1548-0992. Disponível em: <<https://doi.org/10.1109/TLA.2016.7786378>>.

YACCHIREMA, D. C.; ESTEVE, M.; PALAU, C. E. Design and implementation of a Gateway for Pervasive Smart Environments. **2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings**, p. 4454–4459, 2017. Disponível em: <<https://doi.org/10.1109/SMC.2016.7844933>>.