

Uma abordagem teórica e prática em um protocolo para IoT

A theoretical and practical approach to an IoT protocol

DOI:10.34117/bjdv7n4-684

Recebimento dos originais: 26/03/2021

Aceitação para publicação: 30/04/2021

Walter Andrés Vermehren Valenzuela

Formação Acadêmica: Doutor em Engenharia elétrica

Endereço: Av. São Jorge, 1547, São Jorge, Manaus-Amazonas

E-mail: wvalenuela@uea.edu.br

Victor Enrique Vermehren Valenzuela

Formação Acadêmica: Doutor em Engenharia elétrica

Endereço: Av. Mário Ypiranga, 987 Apto 404T1 - bairro Adrianópolis, Manaus-Amazonas

E-mail: vvalenuela@uea.edu.br

Vinicius Tavares Coelho

Formação Acadêmica: Ensino Médio Técnico - Fundação Nokia de Ensino

Endereço: Av. Duque de Caxias, 1696, Praça 14 de Janeiro, Manaus-Amazonas

E-mail: vtc.eng17@uea.edu.br

Italo da Silva Santos

Formação Acadêmica: Ensino Médio Técnico - Fundação Nokia de Ensino

Endereço: Rua Amazonas, nº122, Educandos, Manaus-Amazonas

E-mail: idss.eng16@uea.edu.br

Luís Eduardo Mena Barreto Pereira

Formação Acadêmica: Ensino Médio Técnico - Instituto Federal do Amazonas -CMDI

Instituição atual: Universidade Estadual do Amazonas -Est

Endereço: Rua Uatumã, nº68, Conjunto colina do aleixo, Manaus-Amazonas

E-mail: lembpr.eng16@uea.edu.br

Thiago Godinho Almeida

Formação Acadêmica: Ensino Médio - Escola Professor Jonathas Pontes Athias;

Endereço: Avenida Constantino Nery, nº 2229, Manaus-Amazonas

E-mail: tga.eng@uea.edu.br

Phellipe Tocchetto Dinardi

Formação acadêmica: superior incompleto;

Endereço: R. Raimundo Nonato de Castro, 550 - santo agostinho, Manaus-Amazonas

E-mail: ptd.eng16@uea.edu.br

RESUMO

O protocolo de mensagem MQTT possibilita a interconexão de dados e interações em tempo real entre sistemas, softwares e máquinas que desempenham um papel importante em aplicações em Indústria 4.0 e Internet das Coisas (IoT). Esse trabalho realizou uma análise qualitativa de caráter descritivo dos conceitos e arquitetura do MQTT. Como resultado da análise das características dos brokers, foi possível determinar o Mosquitto como melhor opção para uma aplicação no cenário proposto para esse estudo. Em experimentos, o protocolo apresentou um bom desempenho, com um rápido tempo de resposta, praticidade de aplicação e opções de qualidade de serviço (QoS). Logo, concluímos que o protocolo é adequado para aplicativos que usam dispositivos IoT para análise em tempo real, monitoramento, manutenção preventiva, entre outros usos.

Palavras-chave: MQTT 1. Internet das coisas 2. Indústria 4.0 3.

ABSTRACT

The MQTT message protocol turns possible the interconnection data and real-time interactions between systems, software, and machines that play an important role in applications in Industry 4.0 and Internet of Things (IoT). This work conducted a qualitative analysis of the descriptive character of the concepts and architecture of the MQTT. As a result of the analysis of the characteristics of the brokers, it was possible to determine the mosquito as best option for an application in the proposed scenario for this study. In experiments, the protocol showed good performance, with a fast response time, application practicality and quality of service (QoS) options. In essence, it can be affirmed that the protocol is suitable for applications that use IoT devices for real-time analysis, monitoring, preventive maintenance, among others uses.

Keywords: MQTT 1. Internet of Things 2. Industry 4.0 3.

1 INTRODUÇÃO

Há dois séculos a Inglaterra iniciou a mecanização de seus processos de fabricação ao utilizar o carvão como fonte de energia que gradativamente substituiu os métodos artesanais de fabricação por maquinários movidos a vapor. Tal troca gerou consequências sociais e econômicas que revolucionaram os métodos de produção e as relações sociais (SILVA *et al*, 2018).

A segunda fase da revolução industrial intensificou o desenvolvimento dos processos industriais, pois foi impulsionado por inovações técnicas ao utilizar o petróleo, aço e eletricidade. Nesse período ocorreram avanços nos meios de transportes e o surgimento de linhas de montagens com esteiras rolantes a partir de um novo modelo de produção em massa. Além disso, foi possível avançar em diversos campos da comunicação e da eletricidade que mais tarde, já na metade do século XX desencadeou a terceira revolução industrial (SILVA *et al*, 2018).

Esse período foi marcado pelo uso de semicondutores provocando avanços significativos de automação e robótico nas linhas de produção. Também houve criação de novos computadores e softwares que associados desenvolveram a internet, otimizaram os métodos de comunicação e de telefonia móvel. As novas tecnologias desenvolvidas, nessa fase, possibilitaram que as informações fossem transmitidas cada vez mais rápido e estimularam a interação entre as pessoas do mundo todo de modo que tudo estivesse conectado (SAKURAI & ZUCHI).

Hoje, Tessarini (2018) define a quarta fase da industrialização como Indústria 4.0 em que novas tecnologias permitem a comunicação, integração e controle de informações por meio de sistemas cibernético-físico atrelado a internet das coisas, IoT. Isso permite que quaisquer objetos tanto fabris como do dia a dia possam estar conectados à rede mundial de computadores.

Em virtude disso estima-se que nos próximos anos haverá mais coisas do que pessoas interagindo na web (Cisco IBSG, 2011). É necessário que tais coisas sejam de baixo custo, o que implica baixa capacidade de processamento, armazenamento e comunicação (TORRES et al., 2016). Por isso também é necessário que os protocolos de comunicação possuam simples codificação, baixa latência e baixo consumo de energia e para isso alguns protocolos foram criados com essas características para essa finalidade.

Nesse estudo o protocolo MQTT (Message Queuing Telemetry Transport [MQTT Version 3.1.1 2014]) será abordado de forma teórica e prática partir de uma análise qualitativa de sua arquitetura e funcionamento para aplicações com sistemas IoT.

2 REFERÊNCIA TEÓRICA

2.1 PROTOCOLOS.

Protocolo, de acordo com a definição de dicionário (Dicionário Aurélio, 2010), é uma "reunião de regras que regulam os atos públicos", mas já em um contexto de comunicação, protocolo é um conjunto de regras, parâmetros, sintaxe e informações que serão usados como

padrão para a troca de dados. O protocolo de comunicação serve para que haja uma comunicação entre duas ou mais entidades.

Na computação moderna, o uso destes protocolos é essencial para que haja comunicação entre os dispositivos eletrônicos. Por exemplo, existe o protocolo MQTT, que é largamente usado para a comunicação entre dispositivos, ele tem como vantagens

suas regras e sintaxe simples, permitindo usá-lo em equipamentos menos poderosos, como sensores e microcontroladores.

2.2 INDÚSTRIA 4.0.

Idealizada originalmente na Alemanha, a Indústria 4.0 traz o conceito de interconectividade da linha de produção ao usar ferramentas como o sensoramento e a conexão sem fio, em essência, ela apoia e estimula a automação, troca de informações e o uso de conceitos computacionais.

Assim como as outras revoluções industriais, esta nova indústria emergente tem como objetivo atingir um nível mais elevado de eficiência produtiva, para isso, ele usa os conceitos de tecnologia moderna para digitalizar, otimizar, customizar e adaptar as máquinas, diminuindo a necessidade de interação humana e enfatizando a autonomia desses equipamentos. (Lu, Y., 2017)

Para estimular essa nova indústria, nove aspectos serão usados: Big data e sua análise, robótica autônoma, simulações digitais de produtos, integração digital do sistema de produção, integração de internet, segurança cibernética, armazenamento "em nuvem", monitoramento preciso e realidade aumentada. (Vaidya, S., Ambad, 2018)

2.3 - IOT (INTERNET OF THINGS).

IoT é uma interconexão entre máquinas e dispositivos que compartilham dados através da internet, permitindo uma vigorosa comunicação de dispositivo-dispositivo e humano-dispositivo. Esse último possibilita inúmeras ferramentas disponíveis às pessoas, de certa forma, esta é umas das ideias das cidades inteligentes.

3 METODOLOGIA

3.1 CONCEITO

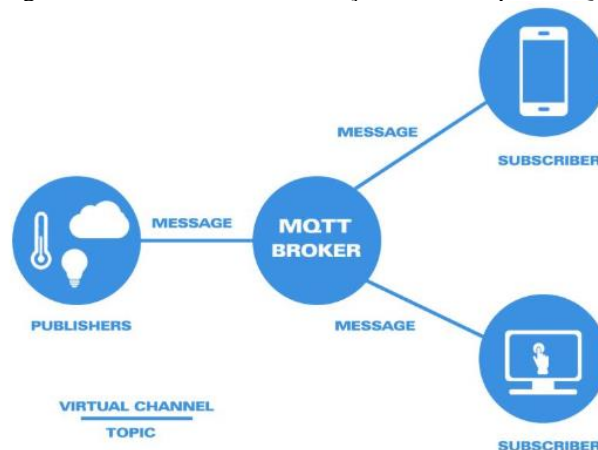
O protocolo MQTT (Messaging Queue Telemetry Transport) foi criado por Dr. Andy Stanford, da International Business Machine Corporation (IBM) e Arlen Nipper, da Arcom (agora Eurotech) em 1999. Segundo MQTT official web page (2019), é um protocolo de mensagens publicador / assinante, extremamente simples e leve, projetado para dispositivos com restrições, baixa largura de banda, alta latência ou redes não confiáveis. Seus princípios básicos são: minimizar a largura de banda de rede e os requisitos de dispositivo, tais princípios fazem com que seja propício o seu uso em "Machine-to-Machine" (M2M) e "Internet of Things" (IoT).

3.2 MODELO DE COMUNICAÇÃO

O protocolo MQTT permite comunicação assíncrona entre cliente e servidor (chamado de broker), segundo PERALTA et al. (2016) este protocolo permite transmitir mensagens de um cliente (publicador) para um ou vários clientes (assinantes), que chegam até o broker e este se encarrega de encaminhar as mensagens para os assinantes. A figura 1 representa o modelo de comunicação utilizado pelo MQTT.

De acordo com AL-FUQAHA et al. (2015), o MQTT possui três componentes básicos: o assinante, o publicador e o broker. Cada dispositivo que se interessa em um determinado tópico, deve assiná-lo para que o *broker* possa informar quando publicadores publicarem mensagens neste tópico. O *broker* também verifica a autorização dos publicadores e assinantes.

Figura 1 – Modelo de Comunicação Utilizado pelo MQTT

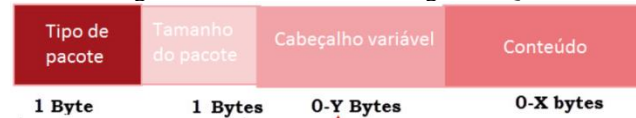


Fonte: Novus, 2019

3.3 MENSAGENS

As mensagens do protocolo possuem um cabeçalho fixo composto por dois *bytes*, o primeiro indica o tipo de mensagem e alguns *flags* que são: entrega duplicada (DUP), qualidade de serviço (QoS) e RETAIN. Segundo MARTINS, ZEM (2016), DUP é ativado quando há tentativa de reenvio de mensagens (PUBLISH, PUBREL, SUBSCRIBE ou UNSUBSCRIBE) que tenham QoS maior que zero, o *flag* QoS indica o nível de garantia de entrega de uma mensagem PUBLISH e o RETAIN indica se uma mensagem PUBLISH deve permanecer no broker após ter sido entregue a todos os assinantes. O cabeçalho fixo também possui bytes para representar a quantidade total de bytes que a mensagem contém. A figura 2 apresenta o modelo de mensagem do MQTT.

Figura 2 – Formato da mensagem MQTT



Fonte: Steve's Internet Guide, 2019

Algumas mensagens possuem um cabeçalho variável e o próprio conteúdo da mensagem, chamado de *payload*. O quadro 1 apresenta com mais detalhes as mensagens do protocolo.

Quadro 1 - Descrição das mensagens MQTT

Nome	Descrição	Fluxo
CONNECT	Solicitação do cliente para se conectar ao servidor	Cliente → Servidor
CONNACK	Reconhecimento do CONNECT	Servidor → Cliente
PUBLISH	Publica uma mensagem em um tópico	Cliente → Servidor ou Servidor → Cliente
PUBACK	Reconhecimento de um PUBLISH	Cliente → Servidor ou Servidor → Cliente
PUBREL	Reconhecimento de um PUBACK	Cliente → Servidor ou Servidor → Cliente
PUBCOMP	Reconhecimento de um PUBREL	Cliente → Servidor ou Servidor → Cliente
SUBSCRIBE	Solicitação de assinatura	Cliente → Servidor
SUBACK	Reconhecimento de um SUBSCRIBE	Servidor → Cliente
UNSUBSCRIBE	Cancelamento de uma assinatura	Cliente → Servidor
UNSUBACK	Reconhecimento de um UNSUBSCRIBE	Servidor → Cliente
PINGREQ	Pedido PING	Cliente → Servidor
PINGRESP	Resposta PING	Servidor → Cliente
DISCONNECT	Solicitação de desconexão	Cliente → Servidor

Fonte: Os autores

No âmbito dos projetos em IoT, os protocolos de comunicação desempenham papéis de alta importância. Dentre os disponíveis, o protocolo MQTT, que tem alcançado altos números de adesão, proporciona uma estrutura bastante completa para as aplicações. Para elucidar a compatibilidade do MQTT com as aplicações IoT basta citar as principais características do protocolo:

- **Simple configuração:** É possível fazer configurações com linhas de comando de baixa complexidade, sem a perda de padronização.
- **Comunicação bidirecional:** Qualquer integrante de uma rede com MQTT pode desempenhar o papel de receptor e/ou de fornecedor de informação.
- **Modelo Publisher/Subscriber:** As informações são enviadas na estrutura de tópicos, assim o receptor não recebe informações que não sejam úteis para seu funcionamento.

- Assíncrono: O sistema utilizado permite um funcionamento em que o fornecedor de informações não necessita de confirmações para funcionar.
- Segurança: O protocolo MQTT permite a utilização de ferramentas para a segurança da rede, como por exemplo designar dados de login e senha para cada cliente conectado à rede. Em adicional, os brokers que são parte do protocolo podem utilizar ferramentas de criptografia.
- Baixo consumo de energia, conforme apresentado no quadro 2.

Quadro 2 - Consumo de energia dos protocolos https e MQTT com SSL

	3G		Wifi	
	HTTPS	MQTT with SSL	HTTPS	MQTT with SSL
% Battery/Message	0.01709	0.00010	0.00095	0.00002
Messages/Hour	1708	160278	3628	263314
Messages Received per 1024 Sent Messages	240	1024	524	1024

Fonte: Fong, Ming 2017

- Bom desempenho em áreas com problemas de transmissão
- Em destaque também, além das informações apresentadas anteriormente, é preciso fazer um estudo sobre *brokers* disponíveis para utilização em uma rede com MQTT, visto que a escolha correta viabiliza o bom funcionamento do projeto. Existem atualmente vinte e dois brokers disponíveis, mas para uma comparação simples serão apresentados aqui quatro exemplos: ActiveMQ (Classic e Artemis), Mosca, Mosquitto, Ponte, conforme o quadro 3.

Quadro 3 - Comparação dos brokers

				
Multiprotocolo	Sim	Não	Não	Sim
Linguagem	Java	JavaScript	C	JavaScript
Versão MQTT	3.1.1	3.1 e 3.1.1	5.0, 3.1 e 3.1.1	3.1.1
Estado de Implementação	Em andamento (Artemis)	Estável	Estável	Em andamento
Extras	Suporte à STOMP, AMQP e OpenWire	Pode ser usado como autônomo ou embarcado	Biblioteca em C para a implementação dos clientes	Realiza ponte (interligação) entre HTTP, MQTT e CoAP
Criador	Apache	Mateo Collina	Eclipse	Eclipse

Fonte: Os autores

Com a análise dos parâmetros apresentados, o broker Mosquitto foi escolhido para a aplicação a ser desenvolvida, pois sua linguagem de implementação é amplamente usada com microcontroladores. É relevante destacar alguns pontos em relação aos brokers apresentados:

- Na versão atual, apesar de estável, o broker Ponte ainda apresenta *features* em estado de implementação, como por exemplo seus módulos de segurança.
- O ActiveMQ Artemis encontra-se em processo de adaptação aos moldes de implementação da Apache, em especial à versão ActiveMQ 5.
- O broker Mosca é estável, porém limitado. Sua implementação é mantida pela comunidade, sendo o broker utilizado como base para outros projetos.
- Mosquitto foi o único dentre os quatro a apresentar suporte para três versões do MQTT. A biblioteca em C oferecida pelo Mosquitto também é um ponto forte na escolha, facilitando a implementação de clientes.

4 RESULTADOS

Nessa etapa foi criado um possível cenário prático associado à indústria: o *publisher* Sensor01 seria o sensor instalado em uma máquina medindo a sua temperatura externa constantemente, ou seja, a medição da temperatura ambiente; o *subscriber* Leitura01 seria o dispositivo que recebe as leituras para controle da temperatura. Como a etapa foi uma simulação, ambos os clientes juntamente com o *broker* estão rodando na mesma máquina no terminal Prompt de Comando com Windows 10.

A segunda etapa também utiliza o Mosquitto *broker* e Mosquitto clientes no terminal, porém com a adição de um novo cliente. Desta vez, dentro do cenário de testes na indústria, este seria o cliente controlado por um técnico que poderia publicar informações e receber também, todas essas funcionalidades sendo utilizadas pelo seu smartphone. Este cliente novo está sendo executado em um *smartphone* com sistema operacional Android.

4.1 REDE COM BROKER E CLIENTE NA MESMA MÁQUINA

Primeiramente, dentro do Prompt de Comando, o Mosquitto *Broker* foi iniciado utilizando o comando **mosquitto -v -p 1884** (-v inicia o *broker* com as configurações padrões; -p escolhe a porta do *broker*). As linhas imprimidas posteriormente mostram algumas informações do *broker*.

Figura 3 - *Broker* no terminal iniciado

```
C:\Program Files\mosquitto>mosquitto -p 1884 -v
1568381561: mosquitto version 1.6.4 starting
1568381561: Using default config.
1568381561: Opening ipv6 listen socket on port 1884.
1568381561: Opening ipv4 listen socket on port 1884.
```

Fonte: Os autores

Em seguida, já em outra instância do terminal do Windows 10, foi iniciado um cliente como *subscriber* do *broker* em um tópico novo chamado de temperatura com QoS 2 e o ID como Leitura01 representando o dispositivo que monitora a temperatura na máquina industrial. O comando utilizado foi **mosquitto_sub -h localhost -p 1884 -t temperatura -i Sensor01 -q 2** (-h configura o host do *broker*; -t é o tópico; -i configura o ID do Cliente e o -q configura o QoS).

Figura 4 - *Subscriber* inscrito no tópico “temperatura”

```
C:\Program Files\mosquitto>mosquitto_sub -h localhost -p 1884 -t temperatura -i
Leitura01 -q 2
```

Fonte: Os autores

A figura 5 mostra o *broker* recebendo a requisição do *subscriber* com o seu ID, tópico e a configuração de QoS.

Figura 5 - *Broker* recebendo uma inscrição

```
C:\Program Files\mosquitto>mosquitto -p 1884 -v
1568382728: mosquitto version 1.6.4 starting
1568382728: Using default config.
1568382728: Opening ipv6 listen socket on port 1884.
1568382728: Opening ipv4 listen socket on port 1884.
1568382744: New connection from ::1 on port 1884.
1568382744: New client connected from ::1 as Leitura01 (p2, c1, k60).
1568382744: No will message specified.
1568382744: Sending CONNACK to Leitura01 (0, 0)
1568382744: Received SUBSCRIBE from Leitura01
1568382744:     temperatura (QoS 2)
1568382744: Leitura01 2 temperatura
1568382744: Sending SUBACK to Leitura01
```

Fonte: Os autores

A próxima etapa é iniciar o *publisher* no *broker* e no mesmo tópico do Cliente Leitura01 para posteriormente enviar mensagens. Este cliente representa, dentro do cenário prático, o sensor instalado na parte externa da máquina industrial. Uma terceira instância do Prompt de Comando foi iniciado e o seguinte comando foi utilizado: **mosquitto_pub -h localhost -p 1884 -t temperatura -i Sensor01 -m 14 graus -q 2** (-m é usado para as mensagens).

Figura 6 - *Publisher* enviando mensagem

```
C:\Program Files\mosquitto>mosquitto_pub -h localhost -p 1884 -t temperatura -i
Sensor01 -m "14 graus" -q 2
C:\Program Files\mosquitto>
```

Fonte: Os autores

A Figura 7 mostra o *broker* recebendo a requisição do Cliente *publisher* com Id Sensor01, QoS 2 e que uma mensagem deve ser enviada para o *subscriber* Leitura01.

Figura 7 - *Broker* recebendo uma inscrição

```
1568383045: Received PINGREQ from Leitura01
1568383045: Sending PINGRESP to Leitura01
1568383086: New connection from ::1 on port 1884.
1568383086: New client connected from ::1 as Sensor01 (p2, c1, k60).
1568383086: No will message specified.
1568383086: Sending CONNACK to Sensor01 (0, 0)
1568383086: Received PUBLISH from Sensor01 (d0, q2, r0, m1, 'temperatura', ... (8
bytes))
1568383086: Sending PUBREC to Sensor01 (m1, rc0)
1568383086: Received PUBREL from Sensor01 (Mid: 1)
1568383086: Sending PUBCOMP to Sensor01 (m1)
1568383086: Sending PUBLISH to Leitura01 (d0, q2, r0, m1, 'temperatura', ... (8
bytes))
1568383086: Received DISCONNECT from Sensor01
1568383086: Client Sensor01 disconnected.
1568383086: Received PUBREC from Leitura01 (Mid: 1)
1568383086: Sending PUBREL to Leitura01 (m1)
```

Fonte: Os autores

Por fim, o *subscriber* recebe a mensagem “14 graus”. Essa leitura está representando a temperatura ambiente onde a máquina industrial está operando. A Figura 8 mostra esse evento.

Figura 8 - *Subscriber* recebendo uma mensagem

```
C:\Program Files\mosquitto>mosquitto_sub -h localhost -p 1884 -t temperatura -i
Leitura01 -q 2
14 graus
```

Fonte: Os autores

4.2 REDE MOSQUITTO COM CLIENTE EM MOBILE

Nesse experimento foi utilizado um *smartphone* com Android conectado na mesma rede LAN do *broker* e dos clientes simulando o cliente no qual o técnico está sob controle tanto de *subscriber* quanto de *publisher*. O aplicativo mobile utilizado foi o MQTT Dashboard.

A Figura 9 mostra a requisição do cliente pelo celular no *broker*.

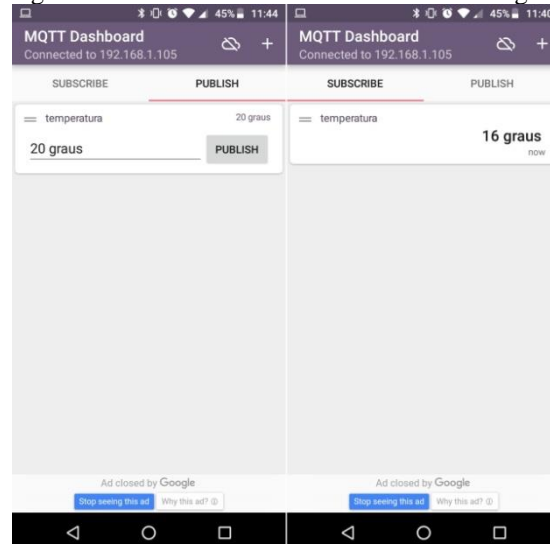
Figura 9 - Broker recebendo uma inscrição de cliente

```
1568384455: New client connected from ::1 as Leitura01 (p2, c1, k60).
1568384455: No will message specified.
1568384455: Sending CONNACK to Leitura01 (0, 0)
1568384455: Client Leitura01 already connected, closing old connection.
1568384455: New client connected from ::1 as Leitura01 (p2, c1, k60).
1568384455: No will message specified.
1568384455: Sending CONNACK to Leitura01 (0, 0)
1568384455: Socket error on client <unknown>, disconnecting.
1568384455: New client connected from 192.168.1.119 as LeituraCelular1568388107576 (p2, c1, k60).
1568384455: No will message specified.
1568384455: Sending CONNACK to LeituraCelular1568388107576 (0, 0)
1568384455: Received SUBSCRIBE from Leitura01
1568384455:     temperatura (QoS 2)
1568384455: Leitura01 2 temperatura
1568384455: Sending SUBACK to Leitura01
```

Fonte: Os autores

As telas da Figura 10 fazem a demonstração, respectivamente, de um cliente em um smartphone recebendo uma mensagem no tópico temperatura e em seguida publicando uma informação no mesmo tópico.

Figura 10 - Cliente no celular recebendo mensagem



Fonte: Os autores

A Figura 11 mostra o *subscriber* de ID Leitura01 recebendo a mensagem. Dentro do cenário hipotético no qual a máquina industrial está operando, foi possível perceber uma alteração de temperatura no ambiente externo agora captada pelo técnico e enviada para o dispositivo de controle.

Figura 11 - Subscriber no computador recebendo mensagem

```
C:\Program Files\mosquitto>mosquitto_sub -h localhost -p 1884 -t temperatura -i
Leitura01 -q 2
20 graus
```

Fonte: Os autores

5 CONCLUSÃO

Pela Metodologia acerca do MQTT apresentada, notou-se que o mesmo atende a algumas características tanto de Indústria 4.0 quanto de IoT. Visto que o protocolo mencionado possui simples sintaxe e regras de envio e recebimento de mensagens, pode então ser empregado em dispositivos de baixo consumo. Nas comparações entre os *brokers*, o Mosquitto foi o escolhido por apresentar melhores resultados dentre as características analisadas..

Para exemplificar a facilidade do MQTT, foi realizado um experimento com a implementação do Mosquitto. Um cenário hipotético que ilustre o conceito de Indústria 4.0 foi idealizado: uma máquina industrial está em pleno curso enquanto um sensor representando o cliente mensura a temperatura ambiente no qual ela está sujeita; outro dispositivo cliente é responsável pelo e controle da temperatura, então ele recebe os dados do sensor toma decisões a partir disso; um técnico conectado na rede do protocolo MQTT como cliente pelo seu smartphone pode enviar temperatura mensuradas e receber as leituras do sensor, podendo também tomar decisões.

Pelos resultados obtidos, observamos a praticidade de implementação de protocolo MQTT através de um estudo de caso com o *broker* Mosquitto em um cenário industrial moderno no qual se necessita de rapidez nas trocas de mensagens e a utilização de dispositivos de baixo consumo, ideal para o conceito de Indústria 4.0.

AGRADECIMENTOS

Este trabalho conta com o apoio da Samsung Eletrônica da Amazônia Ltda., AGIN - Agência de Inovação da UEA e FUEA - Fundação Universitas de Estudos Amazônicos.

REFERÊNCIAS BIBLIOGRÁFICAS

MQTT official webpage. [Online]. Available: <http://mqtt.org/faq>. Acessado em 11/09/2019

PERALTA, Goiuri, et al. **Fog Computing Based Efficient IoT Scheme for the Industry 4.0**. Espanha: IK4-Ikerlan Technology Research Centre, Information and Communication Technologies Area, 2016.

Ala Al-Fuqaha, Senior Member, IEEE, et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 4, FOURTH QUARTER 2015

MARTINS, Ismael, ZEM, José. ESTUDO DOS PROTOCOLOS DE COMUNICAÇÃO MQTT E COAP PARA APLICAÇÕES MACHINE-TO-MACHINE E INTERNET DAS COISAS 2015

Dicionário Aurélio da Língua Portuguesa - 8ª Ed. 2010

Lu, Y. (2017). **Industry 4.0: A survey on technologies, applications and open research issues**. Journal of Industrial Information Integration, 6, 1–10. <https://doi.org/10.1016/j.jii.2017.04.005>

Vaidya, S., Ambad, P., & Bhosle, S. (2018). **Industry 4.0 - A Glimpse**. Procedia Manufacturing, 20, 233–238. <https://doi.org/10.1016/j.promfg.2018.02.034>

Team, I. (n.d.). **5 Areas Where The IoT Is Having The Most Business Impact**. Retrieved September 12, 2019, from Forbes Insights website: <https://www.forbes.com/sites/insights-hitachi/2017/12/18/dont-get-caught-unprepared-when-it-comes-to-iot-security/#74e9c7f4d800>

Nord, J. H., Koohang, A., & Paliszkievicz, J. (2019). **The Internet of Things: Review and theoretical framework**. Expert Systems with Applications, 133, 97–108. <https://doi.org/10.1016/j.eswa.2019.05.014>

Camero, A., & Alba, E. (2019). **Smart City and information technology: A review**. Cities, 93(April), 84–94. <https://doi.org/10.1016/j.cities.2019.04.014>

SAKURAI, R.; ZUCHI, J. D. **AS REVOLUÇÕES INDUSTRIAIS ATÉ A INDÚSTRIA 4.0**. Revista Interface Tecnológica, v. 15, n. 2, p. 480-491, 30 dez. 2018.

EVANS, Dave. **THE INTERNET OF THINGS: How the Next Evolution of the Internet Is Changing Everything**. Cisco Internet Business Solutions Group (IBSG), 2011.

SILVA, Elcio B; SCOTON, Maria; DIAS, Eduardo M; PEREIRA, Sérgio. **AUTOMAÇÃO E SOCIEDADE: quarta revolução industrial, um olhar para o Brasil**. Brasport, 2018.

Torres, A. B. B., Rocha, A. R., and de Souza, J. N. **ANÁLISE DE DESEMPENHO DE BROKERS MQTT EM SISTEMA DE BAIXO CUSTO.** In Anais do XXXVI congresso da sociedade brasileira de computação, 2016.

Steve's Internet Guide. Understanding the MQTT Protocol Packet Structure. Disponível em: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>. Acesso em 30 set 2019

Novus, Artigo: Modelo Cliente-Servidor do Modbus TCP e Publish-Subscribe do MQTT: quando utilizar cada um deles e quais suas vantagens e desvantagens?. Disponível em: https://www.novus.com.br/site/default.asp?TroncoID=053663&SecaoID=0&SubsecaoID=0&Template=../artigosnoticias/user_exibir.asp&ID=618088&idioma=55. Acesso em 30 set 2019.

Fong, Ming: Powering the Internet of Things with MQTT. Disponível em: <https://www.automation.com> . Acesso em 10 set 2019.

A. B. B. Torres, A. R. Rocha, and J. Neuman De Souza, “Análise de Desempenho de Brokers MQTT em Sistema de Baixo Custo,” An. do XXXVI Congr. da Soc. Bras. Comput., no. Julho, pp. 2804–2815, 2016.

CARUSO, José Mário. **Controle de Velocidade de um MIT Tracionando um Veículo Elétrico.** Taubaté, 2007.

ERBER, Pietro. “Gasolina e Tomada”: o carro elétrico está perto. Rio de Janeiro: **Revista do Empresário da ACRJ**, maio/junho, 2010.

FRÉ, Paulo; MARCELINO, Márcio Abud; ADAMI, José Feliciano. Sensor Kelvin para detecção de Tensão. **Revista Sodebras [on line]**. v. 10, n.117, Set./2015, p. 147-152. ISSN 1809-3957. Disponível em: <<http://www.sodebras.com.br/edicoes/N117.pdf>>. Acesso em 04 out. 2015.

MUNDO EDUCAÇÃO. Figura 2. Disponível em: <http://mundoeducacao.bol.uol.com.br/geografia/tipos-graficos.htm>. Acesso em 26 jan 2016.

PACHECO, Marco Aurélio C. **Algoritmos Genéticos: Princípios e Aplicações.** Disponível em: <http://www.ica.ele.puc-rio.br/Downloads/38/CE-Apostila-Comp-Evol.pdf>. Acesso em 04 dez 2010.