



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Uma Arquitetura Interoperável de Automação Residencial para a Internet das Coisas

Matheus Martins Monteiro

Campina Grande, PB
Agosto de 2018

Matheus Martins Monteiro

Uma Arquitetura Interoperável de Automação Residencial para a Internet das Coisas

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Área de Concentração: Controle e Automação

Orientador: Danilo Freire de Souza Santos

Campina Grande, PB
Agosto de 2018

Matheus Martins Monteiro

Uma Arquitetura Interoperável de Automação Residencial para a Internet das Coisas

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovado em 06/08/2018

Gutemberg Gonçalves dos Santos Júnior
Universidade Federal de Campina Grande
Avaliador

Danilo Freire de Souza Santos
Universidade Federal de Campina Grande
Orientador

*Sempre que te perguntarem se podes fazer um trabalho,
respondas que sim e te ponhas em seguida a aprender como se faz.*

Agradecimentos

Agradeço primeiramente a Deus, que é a base de tudo. À minha família, que esteve comigo em todos os momentos, sempre me incentivando a correr atrás dos meus objetivos e que nunca mediram esforços para manter-me firme em busca do meu propósito. Aos meus professores, que me ensinaram todo o essencial necessário para eu me tornar um excelente profissional. Em especial, agradeço ao meu orientador Danilo Freire, pela paciência na orientação e incentivo, o que tornou possível a conclusão deste trabalho. Agradeço também meus amigos, que me aconselharam a seguir trilhando no caminho certo.

Resumo

Este trabalho apresenta uma solução para a comunicação entre dispositivos sensores e atuadores, os quais são capazes de se conectarem à Internet, e tem como finalidade a automatização de ambiente residenciais. Devido a existência de diferentes tipos de protocolos de comunicação para a Internet das Coisas (IoT), existe uma problemática ao se estabelecer uma comunicação entre dispositivos e controladores distintos, os quais os protocolos utilizados sejam distintos, o que ocorre normalmente quando consideramos fabricantes diferentes. Dessa forma, esse trabalho apresenta uma solução interoperável para automação residencial para o IoT. Para tanto, foi aplicado como especificação de dados o modelo de dados da *Open Connectivity Foundation*, chamado de oneIoTa. O oneIoTa oferece uma biblioteca de modelos de dados aberta, a qual pode ser utilizada por diferentes fabricantes de dispositivos. Para o transporte dessas informações, foi utilizado um protocolo padrão para a Internet das Coisas, o MQTT, o qual facilita a interoperabilidade entre os objetos. Com esse trabalho, portanto, espera-se que diferentes dispositivos com projetos distintos sejam capazes de comunicar-se entre si, facilitando ainda a portabilidade, caso necessário. Também espera-se que uma possível expansão da quantidade de dispositivos possíveis de serem inseridos no âmbito residencial, com a facilitação da aplicação de automatização do local em que estejam inseridos, com expectativa para aplicações futuras.

Palavras chave: Automação Residencial; Internet das Coisas; Interoperabilidade; Protocolos de Rede.

Abstract

This work presents a solution for the communication between sensor devices and actuators, which are able to connect to the Internet, and has as purpose the automation of residential environments. Due to the existence of different types of communication protocols for the Internet of Things (IoT), there is a problem when establishing a communication between different devices and controllers, which the protocols used are different, which is usually the case when we consider different manufacturers. Thus, this work presents an interoperable solution for residential automation for IoT. To do so, the data model of the Open Connectivity Foundation, called oneIoTa, was applied as data specification. OneIoTa offers an open data model library, which can be used by different device manufacturers. For the transport of this information, a standard Internet Protocol of Things (MQTT) was used, which facilitates interoperability between objects. With this work, therefore, different devices with different designs are expected to be able to communicate with each other, facilitating portability, if necessary. It is also expected that a possible expansion of the number of possible devices to be inserted in the residential scope, with the facilitation of the application of automation of the place in which they are inserted, with expectation for future applications.

Keywords: Home Automation; Internet of Things; Interoperability; Web Protocols.

Lista de Figuras

1.1	Exemplos de cenários de uso de IoT em Automação Residencial	15
2.1	Internet das Coisas	20
2.2	Volume de Pesquisas no Google sobre Wireless Sensor Networks e IoT	21
2.3	Tecnologias em ascensão no mercado	22
2.4	Exemplo de arquitetura MQTT	24
2.5	Rotina de Aprovação de Novo Modelo de Dado no OneIoTA	25
2.6	Microcontroladores utilizados no projeto.	27
2.7	Exemplo de uso de wildcard de nível único	29
2.8	Exemplo de uso de wildcard multinível	29
3.1	Diagrama Físico da Arquitetura do Sistema	30
3.2	Fluxograma de IoT para processamento de dados de temperatura genérico	31
3.3	Interface Gráfica de gestão da plataforma Mongoose OS	33
3.4	Esquema JSON do recurso <i>oic.r.temperature.json</i>	34
3.5	Exemplo de JSON do recurso <i>oic.r.temperature.json</i>	35
3.6	Retorno de assinatura no recurso <i>/OCF/devices/</i> utilizando a wildcard #	36
3.7	Retorno de assinatura no tópico <i>/OCF/rt/+/(ID do dispositivo)</i> especificando o dispositivo <i>esp01</i>	37
3.8	Retorno de assinatura no recurso <i>oic.r.temperature</i> utilizando a wildcard #	38
3.9	Retorno de assinatura no recurso <i>oic.r.temperature</i> especificando o dispositivo <i>esp01</i>	38
3.10	Retorno de assinatura no tópico <i>/OCF/rt/#</i>	39
3.11	Retorno da publicação no tópico <i>\$aws/things/thingName/shadow/get</i> especificando <i>esp8266_134A20</i> como o nome da coisa	40
4.1	Diagrama lógico da Arquitetura de software montada	41
4.2	Interface Gráfica da plataforma JSON Schema Validator	42
4.3	Validação do JSON	43

Lista de Tabelas

1	Situação 01	14
2	Situação 02	14
3	Situação 03	15
4	Descoberta de Dispositivos Ativos.	44
5	Acesso a Todos os Recursos de um Dispositivo Específico	44
6	Descoberta de Todos os Dispositivos Publicando um Recurso Específico	45
7	Acesso a Recurso Específico de um Dispositivo Específico	45
8	Acesso a Todos os Recursos de Todos os Dispositivos.	46
9	Descoberta de Recursos de Dispositivos Desconectados	46

Lista de Abreviaturas e Siglas

AMQP Advanced Message Queuing Protocol

AWS Amazon Web Services

HTTP Hypertext Transfer Protocol

IoT Internet of Things

JSON JavaScript Object Notation

MQTT Message Queuing Telemetry Transport

OCF Open Connectivity Foundation

OS Operational System

OTA Over-The-Air

P2P Peer-to-Peer

RAML RESTful API Modeling Language

RFID Radio-Frequency IDentification

TCP/IP Transmission Control Protocol / Internet Protocol

XMPP Extensible Messaging and Presence Protocol

Sumário

1	Introdução	13
1.1	Automação Residencial	13
1.2	Justificativa e Problemática	16
1.2.1	Proposta de Solução	16
1.3	Objetivos	17
1.4	Organização do Documento	18
2	Revisão Bibliográfica	19
2.1	Internet das Coisas	19
2.1.1	Histórico	20
2.1.2	Protocolo HTTP e suas limitações para a Internet das Coisas	22
2.2	MQTT	23
2.3	IoTivity e Open Connectivity Foundation	24
2.4	OneIoTA	25
2.5	Tecnologias Relacionadas	26
2.5.1	ESP 32	26
2.5.2	ESP 8266	27
2.6	Conceitos Básicos para o Trabalho	28
3	Solução Proposta	30
3.1	Arquitetura do Sistema	30
3.1.1	AWS IOT	32
3.1.2	Mongoose OS	32
3.1.3	Estrutura do Recurso	33
3.2	Princípios de Funcionamento	35
3.2.1	Descoberta de Dispositivos Ativos	35
3.2.2	Acesso a Todos os Recursos de um Dispositivo Específico.	36
3.2.3	Descoberta de Todos os Dispositivos Publicando um Recurso Específico.	37
3.2.4	Acesso a Recurso Específico de Dispositivos Específicos.	38
3.2.5	Acesso a Todos os Recursos de Todos os Dispositivos.	38
3.2.6	Acesso de Recurso de Dispositivo Desconectado da Rede.	39
4	Desenvolvimento e Validação	41
4.1	Ferramentas e Desenvolvimento	41
4.1.1	JSON Schema Validator	41
4.2	Validação	42
4.2.1	Rotina do MQTT	42
4.2.2	Validando Interoperabilidade	42
4.2.3	Analisando Cenários de Uso	43
5	Conclusão	47
5.1	Trabalhos Futuros	47

1 Introdução

Historicamente, o homem sempre foi o principal vetor para o desenvolvimento de processos artesanais e industriais, sendo usado como força bruta de trabalho para tal. Desta forma, além de todas as alienações e problemas sociais consequentes desse sistema, a queda de produção e eficiência de energia, devido à rusticidade das operações realizadas, impactava negativamente na sociedade e no desenvolvimento da mesma.

Com a revolução industrial, as máquinas assumiram o papel do homem nas tarefas, em especial as de cunho repetitivo, no qual uma certa operação, que não necessitava necessariamente da intervenção humana, deveria ser realizada diversas vezes. Como os equipamentos utilizados eram capazes de realizar este tipo de operação em um curto espaço de tempo, houve por consequência imediata um aumento da produção do produto final que, aliado ao menor custo de operação e a ausência de direitos trabalhistas para o maquinário, ocasionou uma crescente considerável do número e qualidade do maquinário utilizado nas indústrias, que foram substituindo os homens em tarefas específicas, fazendo-os focar apenas nas tarefas intelectuais, inerentes ao ser humano.

De fato, esse foi um divisor de águas na história da humanidade, visto que não somente naquele momento influenciou bastante o estilo de vida como um todo, como até hoje podemos perceber cada vez mais uma ascensão das máquinas e equipamentos de pequeno e grande porte operando nas indústrias como um todo.

Com o desenvolvimento da tecnologia, as tarefas específicas que os dispositivos eram capazes de realizar se expandiram para todos os campos do trabalho humano, desde o desenvolvimento de matérias primas, até os avanços mais recentes da medicina, agregando cada vez mais estes aparelhos no dia-a-dia do homem moderno.

Dessa forma, naturalmente surgiu a necessidade de que esses dispositivos tivessem a capacidade de interagir entre si, sendo capazes de realizar novos serviços e desenvolver novas soluções para as tarefas que já eram realizadas, expandindo ainda mais o potencial que esses instrumentos possuíam. Partindo dessa ideia, diversos modelos de comunicação e protocolos foram idealizados, permitindo essa interação entre equipamentos, cada um deles com aspectos distintos para os diferentes tipos de aplicação.

Nesse contexto, temos a apresentação deste Trabalho de Conclusão de Curso, cujo objetivo é apresentar uma solução para a interação entre dispositivos em um ambiente residencial, de forma a padronizar a forma com que isso é realizado, possibilitando a comunicação de uma gama extensa de dispositivos, e utilizando a mesma referência de comunicação.

1.1 Automação Residencial

Há alguns anos, implantar um sistema de Automação Residencial em uma residência era custoso, e apenas usuários com alto poder aquisitivo tinham acesso a esse recurso. Com o avanço da tecnologia, os custos foram reduzidos, e dessa forma foi se popularizando cada vez mais a técnica de automação em residências. Hoje, temos acesso a uma leque de possibilidades, desde o controle simples de lâmpadas até o total comando da residência, de forma remota e simples.

Entende-se por automação residencial um sistema de comunicação entre dispositivos e/ou serviços de interação com o usuário onde há geração de informação, envio e tomada de decisão baseada na informação. Os dispositivos comunicam-se entre si utilizando um padrão de comunicação e um protocolo que gerencia a troca de mensagens, os quais comumente são protocolos proprietários os

fabricantes. Um exemplo típico de uso desse sistema, ocorre em alarmes residenciais. Quando o sensor de abertura de porta, por exemplo, é acionado, temos que imediatamente um alarme disparado, e fecham-se todas as saídas da residência, protegendo assim os bens do local [5]. Entretanto, nem sempre o sistema de alarme se comunica com o sistema de iluminação, limitando assim a interação e criação de novos serviços.

Este é o cenário em que esse trabalho se propõe a trabalhar, criando um ecossistema interoperável de Internet das Coisas no âmbito residencial. Tendo em vista o ambiente apresentado, é possível extrair vantagens de algumas características inerentes ao local da solução, como por exemplo uma segurança mais flexível, se comparado ao modelo IoT Industrial, onde temos um protocolo de segurança mais rígido e complexo, pois constantemente em indústrias temos a troca de dados sigilosos e de alta importância, necessitando de um cuidado maior com seu transporte, o que não é o caso no que diz respeito à área residencial. Nessa última, temos como foco um sistema voltado ao conforto, praticidade e simplicidade das operações.

A fim de ilustrar o conceito discutido acima, segue algumas situações típicas de domicílios, e como a automação residencial influencia como solução das mesmas, descritas nas Tabelas 1, 2 e 3.

Tabela 1: Situação 01

Consideração Inicial	O usuário está ausente de sua casa, e deseja saber a situação da sua residência
Solução IoT	O usuário através de um aplicativo do seu celular ou plataforma online pode visualizar sua casa em tempo real, através de câmeras espalhadas na sua residência, ao mesmo tempo em que pode acionar alarmes, portas ou luzes, caso perceba algo de estranho, tudo isso feito de forma remota.

Tabela 2: Situação 02

Consideração Inicial	O usuário encontra-se ocupado, enquanto algum indivíduo solicita entrada pela campainha
Solução IoT	Sem muito esforço, o usuário é capaz de descobrir a identidade do indivíduo, permitir ou negar acesso à sua casa, bem como ligar aparelhos que distraiam o mesmo enquanto o usuário não se encontra disponível para recebê-lo pessoalmente.

Tabela 3: Situação 03

Consideração Inicial	Com vários produtos no âmbito residencial que contém controle remoto específico, o usuário sente pouca praticidade na manipulação de todos os controles para o acesso a seus aparelhos
Solução IoT	Com o IoT, o usuário tem um único controle, capaz de controlar qualquer aparelho conectado à rede em sua residência, permitindo ao usuário ainda mais praticidade e conforto ao seu lar.

Como é possível notar, existem inúmeras atividades do dia-a-dia que são facilitadas, ou são possíveis de serem executadas, com o auxílio da Internet das Coisas, fato este que influenciou positivamente para sua popularização. Não há limites para as aplicações de automação nas residências, e a tendência é que cada vez mais usuários integrem esse sistema em suas residências e no seu dia-a-dia. Outros exemplos de aplicação da Internet das Coisas na aplicação residencial é ilustrado no diagrama da Figura 1.1.



Figura 1.1: Exemplos de cenários de uso de IoT em Automação Residencial

Fonte: Adaptado de <http://MainStreet.com>

1.2 Justificativa e Problemática

Visando solucionar o problema de comunicação entre diferentes dispositivos com protocolos distintos, é necessário inicialmente idealizar um conjunto de regras capazes de permitir a comunicação entre tais aparelhos, de forma que se tenha um padrão de comunicação interoperável. Na ciência da computação, tal convenção é denominado *protocolo de comunicação*, e o mesmo é responsável por regular o envio e recebimento de dados entre os equipamentos, de forma que as mensagens sejam entregues de forma correta e legível, em termos de software, para o receptor.

Com isso em mente, é possível idealizar um protocolo completamente novo, o que não seria uma tarefa fácil já que um parâmetro de configuração completamente novo deveria ser criado, o qual deve reagir desde a camada de transporte a qual a mensagem deve seguir, até os critérios adotados para assegurar a segurança do protocolo.

Desta forma, os esforços foram voltados para escolher uma opção de protocolo já presente no mercado, o qual pudesse realizar essa comunicação entre os equipamentos de forma simples, segura e eficiente. Nesse contexto, será explicitado uma breve lista dos protocolos de redes mais populares, e o porquê suas implementações são dificultadas no cenário da Internet das Coisas.

- HTTP (*Hypertext Transfer Protocol*)[10]: É o sistema atual usado pelos navegadores de internet, porém no âmbito do IoT, existem sérias limitações para o seu uso, os quais serão discutidos nesse documento.
- XMPP (*Extensible Messaging and Presence Protocol*)[11]: É um protocolo *peer-to-peer* de mensagens instantâneas. Sua dificuldade de implementação no IoT é de fácil explicação, já que este protocolo foi idealizado principalmente para transporte de mensagens multimídia através de mensagens instantâneas, como exemplo podemos citar o transporte de arquivos de mídia como fotos ou vídeos, o que demanda da rede e do dispositivos a ser utilizado, uma quantidade maior de recursos disponíveis.
- AMQP (*Advanced Message Queuing Protocol*)[12]: Um dos mais populares protocolos de sistemas de mensagens é o AMQP, geralmente utilizado no âmbito corporativo, devido a seu alto desempenho, confiabilidade e interoperabilidade. Por esse motivo, a lista de recursos desde protocolo é extensa, o que o deixa complexo demais para redes e equipamentos cujo recursos são limitados.

Como discutido acima, existem diversos protocolos já idealizados e utilizados para as mais diversas aplicações, porém que não se adaptam perfeitamente às exigências da Internet das Coisas, seja pela complexidade muito alta, ou por suas características nativamente incompatíveis com as aplicações de IoT requeridas. Nesse contexto, será apresentado o protocolo MQTT (*Message Queuing Telemetry Transport*), cuja criação foi concebida para ser o protocolo padrão utilizado nas aplicações de dispositivos na Internet das Coisas. Os motivos que fazem deste protocolo o ideal para a aplicação serão explicitados no tópico abaixo.

1.2.1 Proposta de Solução

Foi abordado acima alguns dos mais populares protocolos existentes e o porquê seu uso é dificultado para aplicações IoT. De fato, existem inúmeros protocolos com as mais variadas características, para diversos tipos de aplicações. Porém, o que mais se encaixa com a proposta da

Internet das Coisas, e que hoje é o protocolo padrão adotado, por exemplo, pela própria Amazon na sua plataforma IoT (AWS IoT), é o MQTT. De acordo com Skerrett [3], diretor de marketing da fundação Eclipse, “*Me parece que o MQTT se tornou o padrão a ser suportado por qualquer provedor sério de soluções para IoT*”¹.

Os principais motivos para a utilização desse protocolo são:

- Simplicidade e leveza suficientes para viabilizar seu uso em hardwares com pouco poder de processamento, bem como em redes instáveis ou restritas;
- Por ser baseado no conjunto de protocolos de domínio público TCP/IP, a comunicação entre dispositivos pode ser realizada mesmo que eles estejam conectados a redes diferentes, aumentando sua flexibilidade;
- Somente a mensagem necessária é transmitida via MQTT, evitando sobrecarga no sistema e diminuindo a ocorrência de anomalias na rede.
- Possui camadas de segurança convenientes, que ajudam a proteger a mensagem, já que não se tem o controle exato de onde a mesma circula ou se existe uma tentativa de interceptação durante o envio, garantindo assim a integridade da mensagem e a entrega a seu destinatário.

Uma explanação mais ampla sobre o protocolo e como o mesmo funciona, será discutida com mais detalhes em um tópico específico mais adiante.

Além do protocolo de transporte, foi necessário definir uma linguagem para transmissão semântica das informações dos dispositivos, ou seja, um modelo de dados para representar as informações de sensores e atuadores em um contexto residencial. Dessa forma, foi escolhida a biblioteca de modelos de dados oneIota [4], o qual é uma biblioteca aberta e padronizada, voltada para a definição de recursos para a Internet das Coisas.

1.3 Objetivos

Este Trabalho de Conclusão de Curso, portanto, tem como objetivo projetar e implementar um ambiente de comunicação para a Internet das Coisas, com foco na automação residencial, utilizando para isso um arcabouço de comunicação interoperável. Nesse sentido, esse trabalho pretende desenvolver uma arquitetura padronizada, baseado no modelo de dados aberto e padronizado de recursos para o IoT, o oneIota [4], permitindo assim sanar as dificuldades em se conectar na mesma rede diferentes tipos de dispositivos.

Dentre as tecnologias a serem exploradas, pretendem-se investigar protocolos de comunicação como o MQTT [7] para a intercomunicação entre objetos em uma rede local, e serviços de integração em nuvem, como o Amazon AWS IoT [8].

¹Tradução livre do Autor

1.4 Organização do Documento

Diante do exposto, a organização do documento é descrita a seguir.

O Capítulo 2 apresenta uma breve revisão sobre o embasamento teórico da Internet das Coisas, bem como uma abordagem sobre alguns conceitos básicos que a acompanham, entre eles podemos citar a plataforma utilizada para comunicação neste trabalho, o MQTT, o modelo de dados adotado do OneIoTA, e as tecnologias de hardware utilizadas, neste caso os microcontroladores da família ESP.

Já no Capítulo 3, é apresentada a proposta de solução adotada, com sua arquitetura física e lógica detalhada, com uma análise sobre a estrutura da mensagem enviada e os softwares utilizadas no trabalho.

No Capítulo 4, temos por objetivo mostrar todo o desenvolvimento do projeto, bem como a validação do trabalho como um todo, confirmando que a solução adotada é de fato possível e eficiente no que se propõe a fazer.

Por último, temos a conclusão deste trabalho, abordando algumas discussões sobre os resultados obtidos, as dificuldades durante o projeto, assim como perspectivas futuras acerca desta tecnologia, no contexto de automação residencial.

2 Revisão Bibliográfica

Esta seção se reserva a explicitar os moldes em que o trabalho se baseia. De início, é abordado o conceito de Internet das Coisas em que o projeto se desenvolve. Subsequentemente, temos a explanação do protocolo MQTT e do modelo de dados do OneIoTA, o qual motivou essa proposta. Por fim, um detalhamento breve sobre o hardware utilizado no projeto e um resumo dos conceitos básicos a serem utilizados neste trabalho.

2.1 Internet das Coisas

Internet das Coisas, ou IoT, é o resultado do desenvolvimento de áreas que englobam diversos processos relacionados ao âmbito de microeletrônica e sistemas embarcados, interagindo com os mais variados tipos de tecnologia com foco social, cujo objetivo é a conexão de diferentes dispositivos na rede mundial de computadores [2]. Com o desenvolvimento de novas tecnologias, houve um aumento na perspectiva por soluções de IoT, graças ao seu grande potencial de solucionar e facilitar as tarefas em diversas áreas industriais e sociais.

Para facilitar o entendimento, podemos analisar a Internet das Coisas como uma expansão da já conhecida Internet tradicional, voltada a proporcionar que objetos dos mais variados tipos se conectem à mesma, de forma que permita-os interagir com serviços e terem acesso a sistemas com maior capacidade computacional e de comunicação. A grande vantagem dessa conexão com a Internet é a de ter um controle à distância do objeto alvo através de dispositivos conectados com ele pela rede, o que permite também que os próprios objetos sejam provedores de serviços para algum outro dispositivo conectado a ele remotamente [9].

Dessa forma, há um novo leque de oportunidades disponíveis, abrindo espaço para o avanço tecnológico de áreas como segurança, saúde, educação, entre outras que se beneficiam direta ou indiretamente com a implementação do IoT em seu ambiente de trabalho. Entretanto, alguns riscos e desafios ainda incidem sobre a Internet das Coisas, como dificuldades no âmbito social, bem como na parte técnica da sua implementação, porém esses obstáculos são trabalhados dia após dia para a minimização de suas consequências.

A Figura 2.1 ilustra algumas das aplicações do IoT.



Figura 2.1: Internet das Coisas

2.1.1 Histórico

O termo *"Internet of Things"*, foi utilizado pela primeira vez ainda na década de 90, por Kevin Ashton em seu trabalho denominado *"I made at Procter & Gamble"*, em 1999. Inicialmente, foi utilizado apenas como uma ferramenta para o desenvolvimento dos estudos com RFID e, portanto, não havia procura pelo termo até meados de 2010, quando teve início o aumento da expectativa de procura pelo IoT, devido principalmente ao avanço e popularização dos estudos de redes de sensores sem fio [9].

A Figura 2.2 demonstra como o interesse do IoT pela comunidade aumentou nos últimos anos, chegando a ultrapassar as pesquisas sobre redes de sensores sem fio, em 2010.

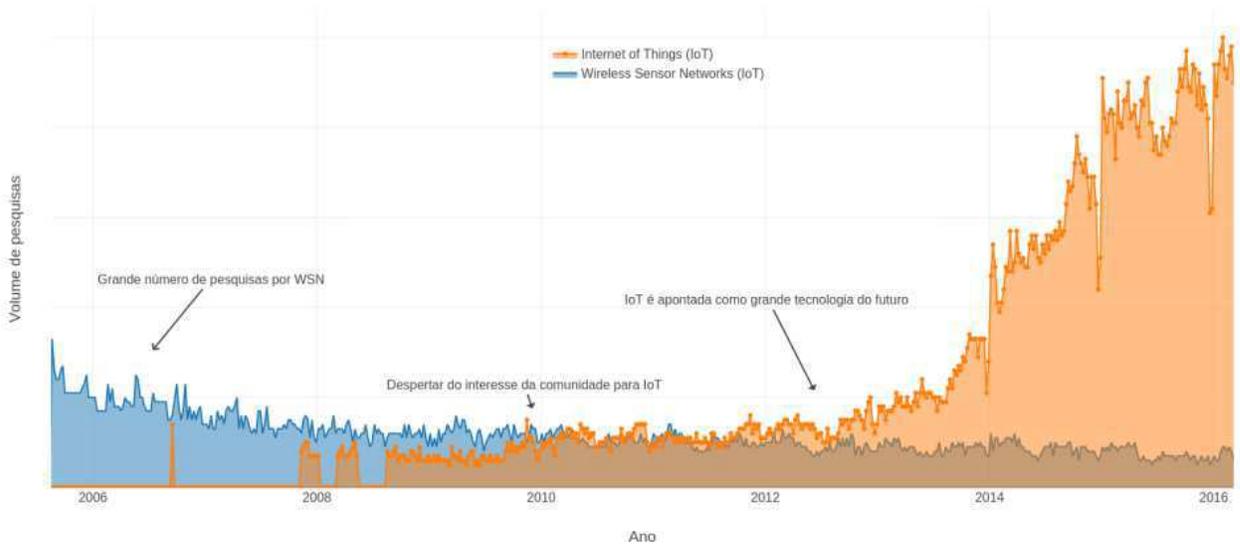


Figura 2.2: Volume de Pesquisas no Google sobre Wireless Sensor Networks e IoT

Fonte: Santos, Bruno P. et al. [9]

Com o passar do tempo, houve um aumento significativo da influência do IoT no mercado de tecnologia, o que gerou um crescente interesse das indústrias e universidades pelo seu uso, gerando o que podemos chamar de ciclo de interesse. A Figura 2.3 expressa a crescente ascensão das principais tecnologias na área de sistemas embarcados hoje no mercado, e a previsão de sua produtividade no futuro. Ainda sobre o gráfico, podemos fazer uma análise no que diz respeito da descoberta da inovação, até a produtividade industrial de fato. Em 2015, a Internet das Coisas estava no pico das expectativas, com um prazo estimado de três a sete anos para ser adotada pelo mercado, fato este que já acontece em 2018, ano em que este trabalho foi redigido. Outro detalhe importante exposto na Figura 2.3 diz respeito às tecnologias que antecederam o IoT, e que favoreceram ao aumento da perspectiva de descoberta desta nova ferramenta, incentivando o interesse da comunidade científica e industrial em explorar mais a fundo essa nova proposta de integração de dispositivos.



Figura 2.3: Tecnologias em ascensão no mercado

Fonte: Adaptado de <https://www.gartner.com>

2.1.2 Protocolo HTTP e suas limitações para a Internet das Coisas

Considerando a Internet, um dos principais protocolos adotados é o HTTP. Nesse sentido, é interessante discutir o porque de tal protocolo não ser amplamente utilizado na IoT. Sendo bastante utilizado nas aplicações de web, o HTTP se baseia na comunicação segundo a lógica solicitação/resposta. O cliente envia uma mensagem para o servidor solicitando um serviço, e o servidor posteriormente envia outra mensagem para o cliente, com a resposta da sua solicitação.

É um dos protocolos mais populares de rede hoje em dia, com inúmeras aplicações, dentre elas uma das principais é permitir a interação de usuários conectados no mundo todo à rede mundial de computadores, permitindo a comunicação cliente/servidor. Porém, algumas limitações simplesmente impossibilitam seu uso para fins específicos de IoT, dentre elas podemos citar [10]:

- O HTTP é um protocolo em que apenas um cliente e um servidor participam. Tal característica contrasta com o IoT, visto que constantemente se tem a necessidade de transmitir uma mensagem a todos os dispositivos da rede, o que seria custoso caso o protocolo adotado fosse esse.
- Existe uma alta complexidade de cabeçalhos e regras, cuja mensagens trocadas entre cliente e servidor geralmente necessitam de uma rede com poucas restrições de velocidade, o que dificulta sua utilização em redes mais simples.
- O protocolo necessita da solicitação do cliente ao servidor, formando assim um caminho unidirecional. No âmbito da Internet das Coisas, constantemente temos a necessidade de

solicitar valores de recursos de sensores e outros equipamentos, que geralmente são cliente na rede, caracterizando assim uma comunicação bidirecional, impossibilitando desta forma o uso do HTTP.

- Há uma sincronia envolvida entre a troca de mensagens, pois o cliente ao solicitar um serviço, espera até o servidor responder. Sendo assim, usar esse protocolo em redes não confiáveis no sentido de ter altas taxas de perdas de pacotes e/ou alta latência pode ser uma tarefa difícil. Para aplicações IoT, é recomendável utilizar uma comunicação assíncrona, dessa forma a própria rede se encarrega de achar a melhor forma de entregar a mensagem ao destinatário, mesmo em ambientes de rede de baixa confiabilidade.

2.2 MQTT

Dado o exposto anteriormente, o MQTT, ou *Message Queuing Telemetry Transport*, é um protocolo de comunicação, focado em transporte de mensagens simples, capaz de ser executado em redes TCP/IP, mesmo que com alta instabilidade, seja por uma alta latência, ou por uma instabilidade de conexão. É hoje o protocolo mais usado nas aplicações IoT, devido ao seu pouco requerimento de uso e sua capacidade de ser usado mesmo em hardwares mais fracos, motivos esses que popularizaram esse protocolo ao longo dos anos.

Em uma rede MQTT, temos sempre dois itens fundamentais para seu funcionamento:

- O *broker*, que é o servidor no MQTT, responsável por receber as mensagens de todos os clientes conectados a ele pela rede, além de gerenciar o envio de mensagens aos clientes que a requisitarem;
- E o cliente, que é a "coisa" de fato. De coisa, podemos entender como sendo um objeto ou algo mais subjetivo, como um aplicativo por exemplo, que tenha a capacidade de trocar mensagens com o broker.

O protocolo se baseia no método "*publish/subscribe*". Uma coisa publica um tópico específico contendo certa mensagem em algum instante para o *broker*, enquanto outra coisa assina esse tópico e recebe do *broker* a mensagem publicada nesse tópico. Dessa forma, todas as mensagens passam pelo *broker*, que age como um intermediário e gerenciador das mesmas, diferente de outros protocolos que usam comunicação P2P, ou *peer-to-peer*, em que há a comunicação direta entre emissor e receptor da mensagem. Uma das vantagens de se ter mensagens rotuladas por tópicos se dá pelo melhor controle sobre qual mensagem um cliente específico terá acesso.

Por exemplo, imagine o cenário de uma automação do controle de temperatura de uma sala qualquer, em que o IoT está sendo implementado. Nesse caso, temos um cliente/sensor coletando os dados de temperatura, e publicando seus dados no tópico *temp_Data*. O administrador desta sala planeja ver os dados em seu laptop, dessa forma ele assina o tópico *temp_Data* para colher os dados que o *broker* repassa, vindas do sensor. De posse dos dados, é possível observar a temperatura atual do cômodo. Porém, esse mesmo administrador quer ter acesso às configurações do sensor, para alterar a temperatura desejada, por exemplo. Assim, ele escolhe uma nova temperatura, e envia a mensagem para o broker por meio do tópico *config_Temperatura*. O *broker* irá repassar a mensagem ao sensor e apenas a ele, devido à necessidade prévia de assinatura neste tópico. O sensor de posse da mensagem, age conforme solicitado, fechando o ciclo do sistema IoT. Uma representação gráfica desse esquema pode ser vista na Figura 2.4 [1]:

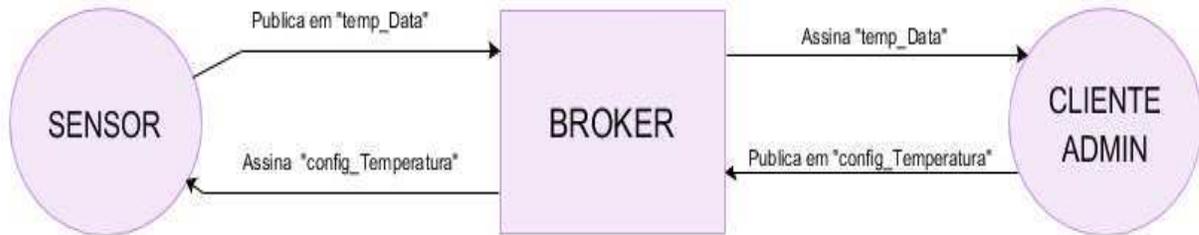


Figura 2.4: Exemplo de arquitetura MQTT

Sabendo como o protocolo de comunicação funciona, o próximo passo é determinar um padrão para a criação da mensagem, que deverá ser seguido por todos os dispositivos na rede, a fim de gerar interoperabilidade entre eles. É nesse contexto, que é inserida a solução pela padronização de dados, utilizando os modelos do OneIoTA.

2.3 IoTivity e Open Connectivity Foundation

Uma das grandes dificuldades técnicas que se tem atualmente é a da comunicação entre dispositivos com projetos diferentes, visto que se tem uma grande variedade de equipamentos e protocolos de comunicação os quais diferem entre si, muitas vezes impossibilitando a interação entre eles, dificultando o desenvolvimento de novos serviços e aplicações, e limitando assim o potencial total do IoT.

Com isso, o uso de padrões de comunicação ajuda na integração de um sistema composto por dispositivos dos mais variados tipos, permitindo que os mesmos tenham a capacidade de se comunicar entre si da forma mais transparente possível. A partir dessa ideia, várias iniciativas para definir padrões de interoperabilidade para o IoT surgiram, entre elas a *Open Connectivity Foundation* [5], que tem por missão projetar, desenvolver e certificar dispositivos para permitir aos fabricantes de tecnologia vender produtos certificados pela OCF, capazes de interagir com os dispositivos IoT atuais. Patrocinada pela OCF, temos o IoTivity [6], que é um projeto independente de código aberto cujo objetivo é implementar as especificações do Open Connectivity Foundation, de modo a criar um ecossistema de comunicação no qual bilhões de dispositivos com ou sem fio se conectem entre si através da Internet.

2.4 OneIoTA

OneIoTA é uma plataforma online e aberta para a comunidade, criada pela OCF, a qual contém os modelos de dados padronizadas para o uso na Internet das Coisas. As ferramentas web contidas no mesmo permitem a usuários criarem novos modelos para qualquer dispositivo IoT de forma simples e rápida, utilizando RAML [16] e JSON [15] para tal.

Os modelos de dados utilizados no OneIoTA são construtivos, de forma que permitem que novos modelos sejam criados se baseando nos modelos já existentes. Os novos modelos podem fornecer definições alternativas aos modelos já existentes, mas devem estar ligados a uma única definição central, que deve existir em todos os modelos e para qualquer dispositivo. Dessa forma, é garantida a interoperabilidade de dispositivos utilizando novos modelos, com os dispositivos que usam modelos já existentes, por exemplo [4].

Os modelos são definidos usando sempre RAML e esquemas JSON. Novos modelos seguindo o padrão OneIoTa criados por usuários são sempre analisados por um revisor da OCF antes de ser lançado na plataforma. O lançamento de novos modelos aprovados são armazenados em um repositório GitHub. Uma descrição gráfica do processo de inserção de novos modelos é detalhada na Figura 2.5.

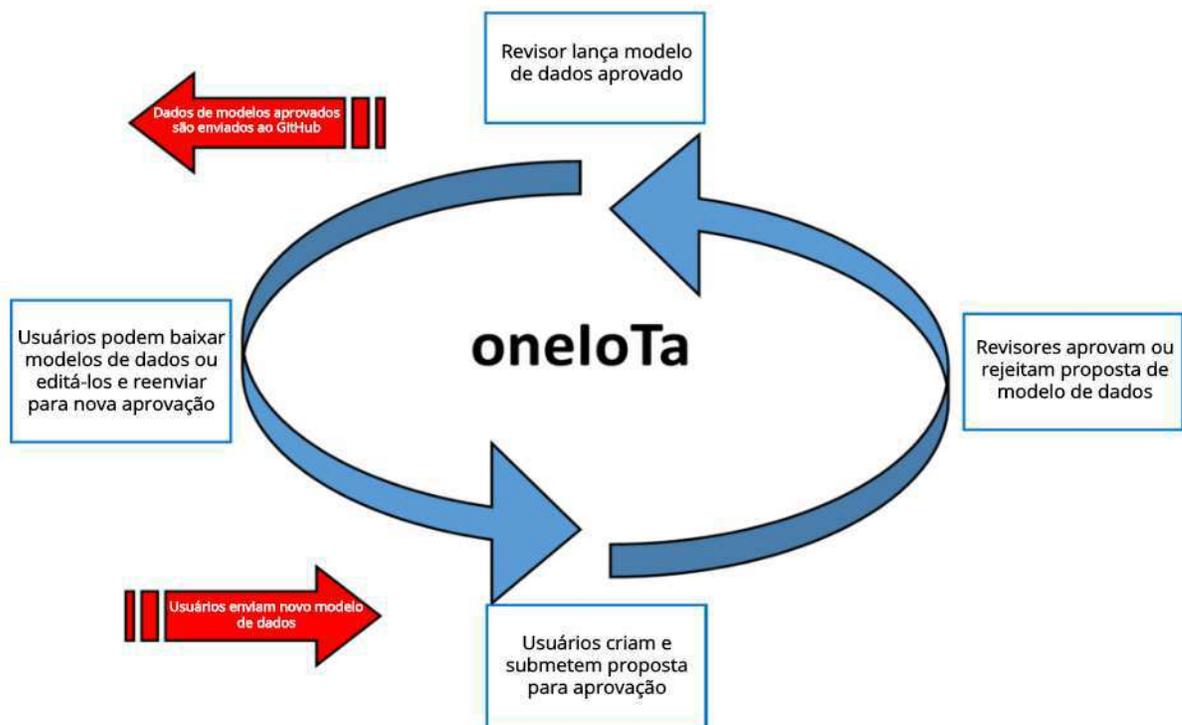


Figura 2.5: Rotina de Aprovação de Novo Modelo de Dado no OneIoTA

Fonte: Adaptado de <https://openconnectivity.org/developer/oneiota-data-model-tool>

2.5 Tecnologias Relacionadas

Abaixo estão relacionados os microcontroladores utilizados neste trabalho, para a Internet das Coisas com foco no ambiente residencial.

2.5.1 ESP 32

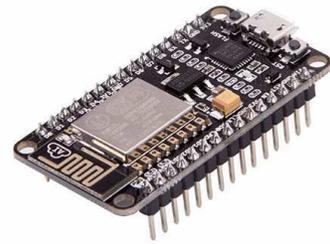
Um dos Hardwares utilizados na realização deste projeto consiste no ESP 32. Segue abaixo uma visão geral de suas especificações. Uma ilustração deste microcontrolador se encontra na Figura 2.6a ².

- CPU: Xtensa® Dual-Core 32-bit LX6
- ROM: 448 KBytes
- RAM: 520 Kbytes
- Flash: 4 MB
- Clock máximo: 240MHz
- Wireless padrão 802.11 b/g/n
- Conexão Wifi 2.4Ghz (máximo de 150 Mbps)
- Antena embutida
- Conector micro-usb
- Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode e P2P Power Management
- Modos de operação: STA/AP/STA+AP
- Bluetooth BLE 4.2
- Portas GPIO: 11
- GPIO com funções de PWM, I2C, SPI, etc
- Tensão de operação: 4,5-9V
- Taxa de transferência: 110-460800bps
- Suporta Upgrade remoto de firmware
- Conversor analógico digital (ADC)
- Distância entre pinos: 2,54mm
- Dimensões: 49 x 25,5 x 7 mm

²Datasheet disponível em: https://cdn.sparkfun.com/datasheets/IoT/esp32_datasheet_en.pdf



(a) Microcontrolador ESP 32



(b) Node MCU ESP8266

Figura 2.6: Microcontroladores utilizados no projeto.

2.5.2 ESP 8266

Completando a lista de hardwares utilizados, temos o Esp 8266 cuja características seguem listadas abaixo. Na Figura 2.6b é ilustrado um exemplar deste microcontrolador ³.

- Wireless padrão 802.11 b/g/n
- Antena embutida
- Conector micro-usb
- Modos de operação: STA/AP/STA+AP
- Suporta 5 conexões TCP/IP
- Portas GPIO: 11
- GPIO com funções de PWM, I2C, SPI, etc
- Tensão de operação: 4,5 9V
- Taxa de transferência: 110-460800bps
- Suporta Upgrade remoto de firmware
- Conversor analógico digital (ADC)
- Distância entre pinos: 2,54mm
- Dimensões: 49 x 25,5 x 7 mm

³Especificações disponíveis em: https://www.electrodragon.com/w/ESP-12F_ESP8266_Wifi_Board

2.6 Conceitos Básicos para o Trabalho

Para o correto estudo sobre IoT e como sua tecnologia é utilizada, e considerando o escopo desse trabalho, primeiro é necessário detalhar alguns conceitos base que serão amplamente discutidos posteriormente, e que necessitam de um conhecimento implícito adquirido.

- **Coisa:** É um dispositivo qualquer de uso cotidiano, o qual literalmente pode ser qualquer "coisa", como um fogão, um automóvel, ou até mesmo algo mais abstrato como um bairro. O objeto a ser definido como coisa varia conforme a aplicação.
- **Dispositivo:** Se trata de um sensor ou algo similar, associado à coisa. É responsável por informar o estado de um elemento, o qual é repassado para as outras coisas conectadas na rede.
- **Atuadores/Sensores:** Atuadores recebem informações e atuam de acordo, modificando seu estado para influenciar o contexto o qual está inserido. Sensores são elementos que avaliam o contexto ao qual estão inseridos, e enviam suas informações à coisa, objeto ao qual está associado. A coisa envia essas informações a qualquer dispositivo conectado com ela pela rede, seja outra coisa, unidades de armazenamento externas, ou a nuvem.
- **Nuvem:** Servidores remotos conectados uns aos outros, formando um único ecossistema global. São responsáveis por armazenar dados, gerenciar e executar aplicações, além de fornecer certos tipos de conteúdos e serviços online, tais como conteúdos de mídia e dados de softwares.
- **WildCards:** Ao fazer uma assinatura em um tópico MQTT, temos a possibilidade de usar um recurso chamado *wildcard*, que consiste em caracteres especiais reservados pelo protocolo, que podem ser utilizados para fins específicos. Uma possível tradução para este termo seria a de caractere coringa, sendo que consegue representar todos os elementos no nível em que é aplicada. Este recurso se aplica apenas quando se assina um tópico, sendo proibido sua utilização ao fazer publicações. Dito isto, especificaremos a seguir dois tipos de *wildcard*.
 - **Nível único:** Representado pelo símbolo `+`. É um *wildcard* capaz de substituir um único nível no tópico assinado.

Dessa forma, qualquer tópico combina com uma assinatura no nível específico em que essa *wildcard* foi utilizada.

Como exemplo, podemos tomar o tópico *myhome/groundfloor/+/temperature*. Baseado no conceito acima, essa assinatura poderia referenciar, ou não, os seguintes tópicos explicitados na Figura 2.7.



A list of five MQTT topics, each preceded by a status icon (green checkmark for success, red X for failure):

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / fridge / temperature

Figura 2.7: Exemplo de uso de wildcard de nível único

Fonte: <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>

- **MultiNível:** Representado pelo símbolo #. Diferente do nível único, esse caractere especial permite representar todos os tópicos do nível que ele se encontra, e dos níveis seguintes também. Por essa condição especial, esse caractere deve ser sempre o último caractere de um tópico assinado.

Um cliente assinando este tópico é capaz de receber todas as mensagens que combinem com os níveis que antecedem a *wildcard*. Se a assinatura consistir apenas nesta *wildcard*, significa que o cliente receberá toda e qualquer mensagem que os outros clientes publicarem ao broker.

Como exemplo, podemos tomar o tópico *myhome/groundfloor/#*. Baseado no conceito explicitado acima, essa assinatura referencia, ou não, os seguintes tópicos definidos na Figura 2.8.



A list of five MQTT topics, each preceded by a status icon (green checkmark for success, red X for failure):

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✓ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature

Figura 2.8: Exemplo de uso de wildcard multinível

Fonte: <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>

- **Shadows:** É um serviço disponibilizado pela Amazon na sua plataforma AWS IOT, que utiliza tópicos MQTT reservados para permitir que clientes gravem, atualizem ou obtenham o estado de um dispositivo conectado à rede. É composto basicamente de um arquivo de texto, o qual o dispositivo se utiliza para publicar informações, e que se mantém independente do estado atual do publicador na rede. Dessa forma, mesmo que o aparelho que publicou a mensagem fique offline, sua última mensagem está gravada na rede através do arquivo *shadow*, podendo ser acessada ou modificada por outros dispositivos, utilizando os tópicos reservados disponibilizados pela plataforma web.

3 Solução Proposta

Com base em todos os dados explicitados acima, bem como utilizando os hardwares citados anteriormente, foi possível definir uma solução para o problema proposto pelo trabalho.

3.1 Arquitetura do Sistema

Nesse contexto, é importante analisar os aspectos físicos da arquitetura, para compreender bem o funcionamento do sistema. A Figura 3.1 ilustra como o ecossistema foi construído.

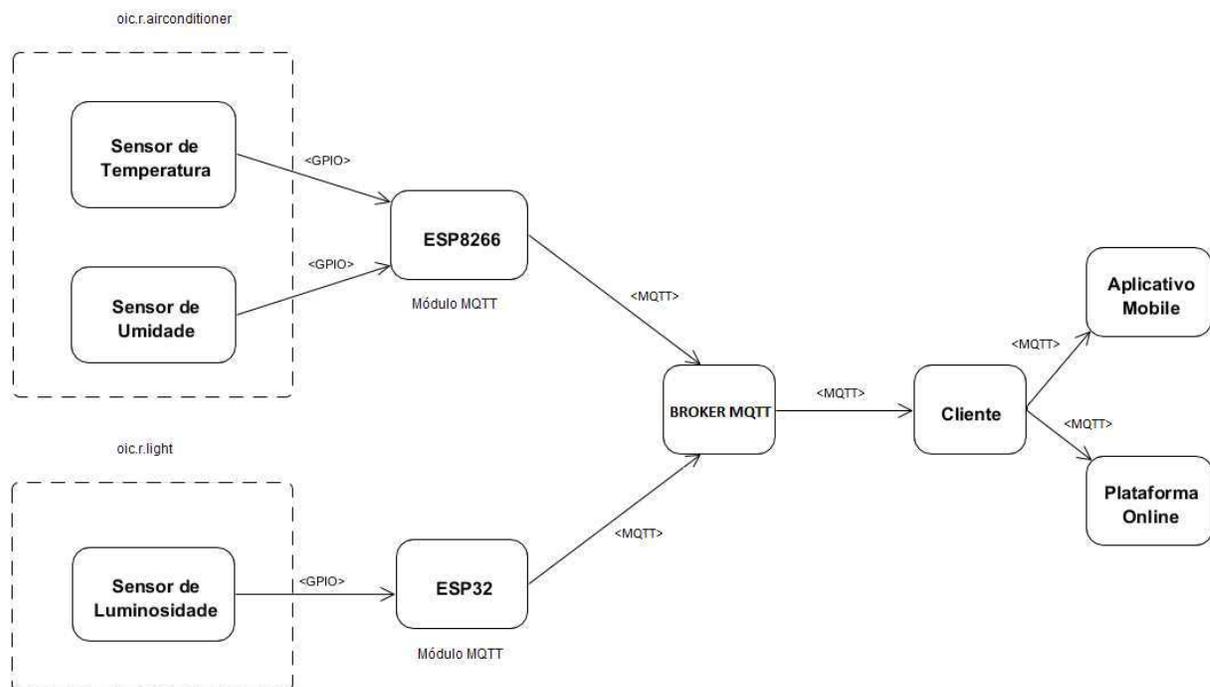


Figura 3.1: Diagrama Físico da Arquitetura do Sistema

Analisando o sistema do ponto de vista físico podemos notar a maior contribuição que este trabalho se propõe a oferecer, que consiste em usar os modelos do OneIoTa com o transporte via MQTT. Isto é possível, aplicando o processo de "Publish/Subscribe" em mensagens formatadas seguindo o padrão OneIoTa.

Para o fluxograma apresentado na Figura 3.2, foi adotado um exemplo genérico de uma aplicação de controle de temperatura em um ambiente de Internet das Coisas. Os dados do sensor chegam até o dispositivo, que publica o mesmo ao *broker*, o qual será encarregado de receber a mensagem do cliente e enviar para o dispositivo alvo a mensagem que o mesmo deverá requisitar. Ao mesmo tempo, recebe do dispositivo alvo uma atualização do valor do termostato, dada pelo usuário que visualizou previamente o dado, e envia essa informação de volta para o dispositivo controlador, fechando assim um caminho bidirecional entre as coisas, com o *broker* como intermediário.

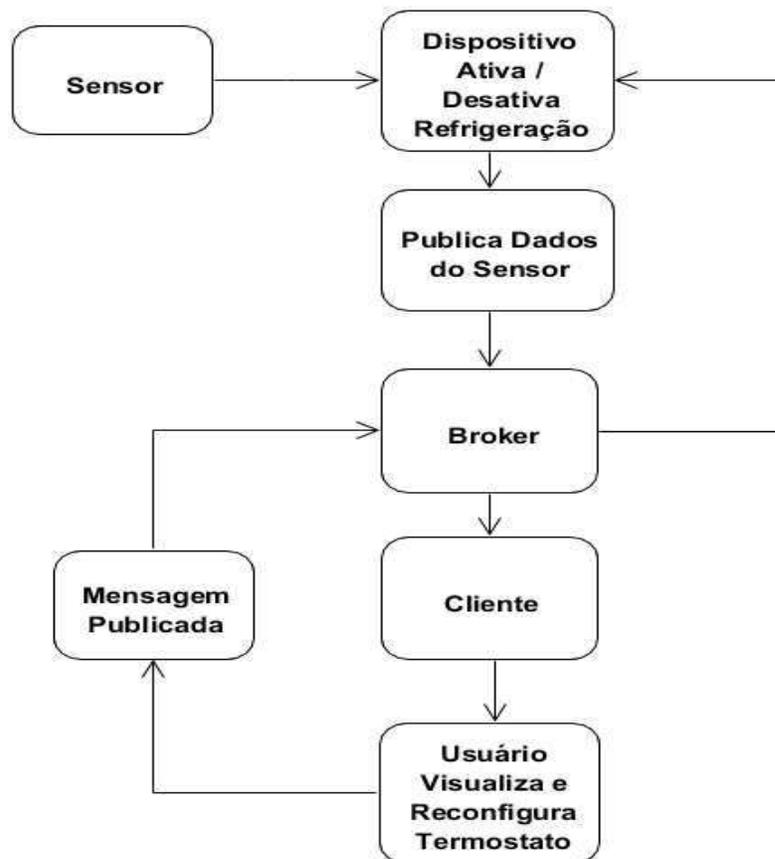


Figura 3.2: Fluxograma de IoT para processamento de dados de temperatura genérico

O modelo foi adotado seguindo a semelhança do protocolo utilizado para implementação do projeto, o MQTT. Os clientes utilizados no projeto são os microcontroladores ESP, os quais realizam a troca de informações com o *broker* via MQTT, usando a rede local como ponte de dados em primeiro nível, rede essa que deve ser conectada a Internet através de um *gateway*.

Usando uma visão lógica, temos um cliente que receberá o comando do usuário e publicará um tópico específico no *broker*, que por sua vez encaminhará a mensagem até o dispositivo alvo e o fará executar o comando do usuário. Após esse processo, o *broker* irá ler o estado do dispositivo e enviar a informação de volta para o cliente, assim o usuário terá a confirmação em uma interface visual da plataforma em nuvem, qual o estado atual do dispositivo de interesse.

Para manter a Interoperabilidade, como introduzido anteriormente, foi implementado o modelo de dados do OneIoTa, permitindo a padronização do sistema. O *broker* que estará sempre em execução, esperando uma publicação de algum tópico, irá receber um JSON com os dados do dispositivo, e executar os comandos que forem necessários, enviando também um arquivo texto de dados para o dispositivo ESP alvo (que se mantém em um estado de baixo consumo de energia até ser necessário seu uso), controlando o mesmo a executar a tarefa proposta.

Após a decisão do protocolo de comunicação a ser utilizado, e que padrão a mensagem a ser trabalhada deverá obedecer, restou decidir que *broker* seria utilizado para intermediar as trocas de

mensagens em nuvem (ou seja, na Internet). Para isso, diversas opções estavam disponíveis para escolha, e a mais adequada para esta aplicação foi utilizando a plataforma online da Amazon, a AWS IoT.

Por último, era preciso de um sistema operacional, capaz de configurar os dispositivos IoT, e que fosse possível programá-los com as rotinas desejadas. Dentre algumas opções, foi optado uma plataforma online gratuita e de código aberto, o Mongoose OS.

3.1.1 AWS IOT

O *Amazon Web Services*, ou AWS, oferece serviços de infra-estrutura para negócios na forma de serviços web, comumente conhecidos como computação na nuvem. Dessa forma, não houve nenhum custo com infra-estrutura de *broker*, já que foi utilizado o serviço da Amazon para tal, que tem acesso liberado de forma gratuita no período de um ano, com apenas algumas ferramentas com uso limitado. Assim, todo o gerenciamento, criação e manutenção do *broker* utilizado nesse trabalho foi submetido aos cuidados da Amazon.

3.1.2 Mongoose OS

O mongoose é um sistema operacional de código aberto, construído com foco em microcontroladores de baixo consumo, em especial para aplicações IoT. O sistema conta com recursos que facilitam o usuário que usa sua plataforma para desenvolver aplicações da Internet das Coisas, e permite programação nas linguagens C e JavaScript.

Seu grande diferencial em relação aos demais sistemas operacionais do mercado é seu tamanho reduzido e sua facilidade de configuração, já que o próprio sistema se encarrega de setar os parâmetros necessários para a conexão com protocolos como MQTT, SNMP, HTTP, entre outros. Com isso, o sistema permite de forma automática a comunicação com o *broker* e seus clientes via MQTT, caso de estudo nesse trabalho. O mongoose tem integração nativa com plataformas como Microsoft Azure⁴, Adafruit IO⁵, Google IoT Core⁶, bem como a própria AWS IoT, citada acima. Também permite atualizações *Over-The-Air (OTA)*, facilitando aplicações de longa duração e comerciais. Uma visualização da interface gráfica desta plataforma pode ser visualizado na Figura 3.3.

Dentre os dispositivos suportados, temos o esp32, esp8266, STM32F4 entre outros. Neste trabalho, dois diferentes dispositivos foram utilizados, o esp32 e esp8266, cujas características e vantagens serão discutidas logo abaixo.

⁴<http://www.azure.com>

⁵<https://io.adafruit.com/>

⁶<https://cloud.google.com/iot-core/>


```
1 {
2   "id": "http://openinterconnect.org/iotdatamodels/schemas/oic.r.temperature.json#",
3   "$schema": "http://json-schema.org/draft-04/schema#",
4   "description": "Copyright (c) 2016, 2017 Open Connectivity Foundation, Inc. All rights reserved.",
5   "title": "Temperature",
6   "definitions": {
7     "oic.r.temperature": {
8       "type": "object",
9       "properties": {
10        "temperature": {
11          "type": "number",
12          "description": "current temperature setting or measurement"
13        },
14        "units": {
15          "type": "string",
16          "enum": ["C","F","K"],
17          "description": "Units for the temperature value",
18          "readonly": true
19        }
20      }
21    }
22  },
23  "type": "object",
24  "allof": [
25    {"$ref": "oic.baseResource.json#/definitions/oic.r.baseresource"},
26    {"$ref": "#/definitions/oic.r.temperature"}
27  ],
28  "required": ["temperature"]
29 }
```

Figura 3.4: Esquema JSON do recurso *oic.r.temperature.json*

Fonte: <https://oneiota.org>

A Figura 3.4 apresenta o esquema JSON utilizado pelo OneIoTA, o qual define as regras de como criar uma mensagem JSON para lidar com um dado de temperatura. Na linha 6 da Figura 3.4, temos uma lista de definições a qual compõem o corpo da mensagem, e que devem estar presentes na mesma. A seguir, segue uma breve explicação sobre cada uma delas:

- Na linha 7, podemos observar a ID do recurso, que é o nome que identifica o recurso que estamos lidando, neste caso *oic.r.temperature*.
- A seguir o modelo nos diz que o recurso temperatura é do tipo objeto, cujo valor deve ser um número (linha 11), representando a temperatura atual medida ou ajustada (linha 12).
- Este mesmo recurso deve ser referenciado com uma unidade de medida do tipo string (linha 15), e deve conter apenas os caracteres referenciados (linha 16), que representam a temperatura em graus Celsius, Fahrenheit e Kelvin, respectivamente.

Dessa forma, temos uma regra em que todos os recursos deste tipo devem ser pautados, permitindo assim a interoperabilidade do sistema, já que temos um padrão de criação de mensagem deste recurso.

Um exemplo de mensagem JSON criada do recurso exemplo utilizando o modelo de dados do OneIoTA pode ser observado na Figura 3.5.

```
example: |
  {
    "id":          "unique_example_id",
    "temperature": 20.0,
    "units":       "C",
    "range":       [0.0,100.0]
  }
```

Figura 3.5: Exemplo de JSON do recurso *oic.r.temperature.json*

Fonte: <https://oneiota.org>

Vale ressaltar que os shadows da plataforma AWS IOT foram escritos utilizando a mesma regra, e por isso contem a mesma mensagem base que é publicada pelos clientes.

3.2 Princípios de Funcionamento

Para utilizar o modelo do OneIoTa no protocolo MQTT, houve a necessidade de adaptação da solução, afim de permitir a padronização na formatação de tópicos e mensagens com o MQTT. Para solucionar este problema, uma série de tópicos e assinaturas específicas foram idealizadas, para que a descoberta de dispositivos e acesso aos recursos dos mesmo fosse possível, compondo assim parte fundamental da solução proposta por este trabalho.

Nos tópicos que seguem são apresentados os problemas encontrados, seguidos de sua respectiva solução, proposta pelo autor do projeto.

3.2.1 Descoberta de Dispositivos Ativos

Uma das funcionalidades que este trabalho se propõe a oferecer é a de informar ao usuário quais os clientes estão conectados ao broker, e que estão ativos no momento da solicitação. Com isso, caso essa informação seja requerida, o sistema deveria retornar uma lista com todos os clientes online, contendo as seguintes informações para cada dispositivo encontrado:

- ID do dispositivo
- ID do Recurso que o dispositivo publica

Dessa forma, foi criada uma rotina em que os dispositivos publicam periodicamente para o tópico */OCF/devices/* mensagens de registro. Uma assinatura neste tópico utilizando a wildcard *#* retorna um JSON contendo ID do dispositivo e do recurso de cada controlador conectado na rede naquele momento. O retorno do JSON, conseqüente dessa assinatura, é ilustrado conforme a Figura 3.6.

OCF/devices/esp01

5 de ago de 2018 19:09:55

```
{
  "Device_ID": "esp8266_134A20",
  "Resource_ID": "oic.r.temperature"
}
```

OCF/devices/esp02

5 de ago de 2018 19:09:52

```
{
  "Device_ID": "esp8266_172515",
  "Resource_ID": "oic.r.light.brithness"
}
```

Figura 3.6: Retorno de assinatura no recurso */OCF/devices/* utilizando a wildcard #

3.2.2 Acesso a Todos os Recursos de um Dispositivo Específico.

O acesso é concedido devido aos dispositivos conectados publicarem constantemente no tópico */OCF/rt/nome-do-recurso/(ID do dispositivo)*. Uma assinatura no tópico */OCF/rt+/+(ID do dispositivo)* utilizando a ID do dispositivo retorna um JSON contendo a ID do dispositivo que fornece o recurso, bem como todos os recursos em si, apenas do dispositivo especificado. A Figura 3.7 demonstra o retorno de uma assinatura neste tópico.



Figura 3.7: Retorno de assinatura no tópico `/OCF/rt/+/(ID do dispositivo)` especificando o dispositivo `esp01`

3.2.3 Descoberta de Todos os Dispositivos Publicando um Recurso Específico.

A descoberta de dispositivos que publicam um certo tipo de recurso é retornada após assinar o tópico `/OCF/rt/nome-do-recurso/#`. Todos os dispositivos conectados publicam para tópicos como esse, e uma assinatura neste tópico utilizando a wildcard `#` deve retornar um JSON contendo a ID do dispositivo que fornece o recurso, bem como o recurso em si, formatado com o modelo de dados do OneIoTa, para cada dispositivo conectado na rede naquele momento. Na Figura 3.8 é possível visualizar o retorno de assinatura no recurso `oic.r.temperature` utilizando a wildcard `#`.

The image shows two screenshots of a REST client interface. Each screenshot displays a URL, a timestamp, and two buttons: 'Exportar' and 'Ocultar'. Below each header is a code block containing a JSON object representing a temperature reading.

Top Screenshot: URL: `OCF/rt/oic.r.temperature/esp01`, Timestamp: 19 de jul de 2018 01:26:41. JSON response:

```
{
  "rt": "[\"oic.r.temperature\"]",
  "id": "esp8266_134A20",
  "temperature": 43.172678,
  "units": "C",
  "range": "[0.0,100.0]"
}
```

Bottom Screenshot: URL: `OCF/rt/oic.r.temperature/esp02`, Timestamp: 19 de jul de 2018 01:26:41. JSON response:

```
{
  "rt": "[\"oic.r.temperature\"]",
  "id": "esp8266_134A20",
  "temperature": 93.620691,
  "units": "C",
  "range": "[0.0,100.0]"
}
```

Figura 3.8: Retorno de assinatura no recurso *oic.r.temperature* utilizando a wildcard #

3.2.4 Acesso a Recurso Específico de Dispositivos Específicos.

O acesso é concedido fazendo com que os dispositivos conectados publiquem periodicamente para o tópico `/OCF/rt/nome-do-recurso/(ID do dispositivo)` mensagens com seus recursos. Uma assinatura no tópico utilizando a ID do dispositivo retorna um JSON contando a ID do dispositivo que fornece o recurso, bem como o recurso em si. A Figura 3.9 apresenta um exemplo de retorno de uma assinatura neste tópico.

The image shows a screenshot of a REST client interface. It displays a URL, a timestamp, and two buttons: 'Exportar' and 'Ocultar'. Below the header is a code block containing a JSON object representing a temperature reading from a specific device.

URL: `OCF/rt/oic.r.temperature/esp01`, Timestamp: 19 de jul de 2018 01:38:11. JSON response:

```
{
  "rt": "[\"oic.r.temperature\"]",
  "id": "esp8266_134A20",
  "temperature": 75.349913,
  "units": "C",
  "range": "[0.0,100.0]"
}
```

Figura 3.9: Retorno de assinatura no recurso *oic.r.temperature* especificando o dispositivo *esp01*

3.2.5 Acesso a Todos os Recursos de Todos os Dispositivos.

Para descobrir todas as mensagens que estão sendo publicadas ao broker, o usuário deve assinar o tópico `/OCF/rt/#`. Uma assinatura no tópico utilizando a ID do dispositivo retorna todos os

recurso que estão sendo publicados no momento, bem como o dispositivo IoT utilizado que está publicando. A figura 3.10 ilustra o retorno dessa assinatura.

OCF/rt/oic.r.temperature/esp01	19 de jul de 2018 18:47:31
<pre>{ "rt": "[\"oic.r.temperature\"]", "id": "esp8266_134A20", "temperature": 31.140059, "units": "C", "range": "[0.0,100.0]" }</pre>	
OCF/rt/oic.r.humidity/esp01	19 de jul de 2018 18:47:31
<pre>{ "rt": "[\"oic.r.humidity\"]", "id": "esp8266_134A20", "humidity": 40, "desiredHumidity": 50 }</pre>	
OCF/rt/oic.r.light.brightness/esp03	19 de jul de 2018 18:47:31
<pre>{ "rt": "[\"oic.r.temperature\"]", "id": "esp8266_172515", "brightness": 50 }</pre>	

Figura 3.10: Retorno de assinatura no tópico /OCF/rt/#

3.2.6 Acesso de Recurso de Dispositivo Desconectado da Rede.

O acesso de recurso para dispositivos desconectados se dá através de publicação de uma mensagem vazia no tópico reservado do AWS IoT: *\$aws/things/thingName/shadow/get*. Esse método gera uma mensagem JSON completa, incluindo todos os metadados.

Vale salientar que o valor do recurso será o último valor publicado pelo dispositivo e pode não corresponder ao valor atual do recurso, pois o dispositivo pode estar em um estado inconsistente,

por exemplo momentaneamente desconectado.

Um exemplo de retorno JSON após publicação neste tópico reservado é apresentada na Figura 3.11:

```
{
  "reported": {
    "rt": "[\"oic.r.temperature\"]",
    "id": "unique_example_id",
    "temperature": 16.59277,
    "units": "C",
    "range": "[0.0,100.0]",
    "ota": {
      "fw_id": "20180615-171337/???",
      "mac": "5ECF7F134A20",
      "device_id": "esp8266_134A20",
      "app": "demo-js"
    },
    "uptime": 14.10913
  }
}
```

Figura 3.11: Retorno da publicação no tópico *\$aws/things/thingName/shadow/get* especificando *esp8266_134A20* como o nome da coisa

4 Desenvolvimento e Validação

Para validar todo o processo desenvolvido, foi criada uma rotina de testes de validação para o sistema, o que permitiu investigar e apurar o completo funcionamento do ecossistema montado.

4.1 Ferramentas e Desenvolvimento

O Mongoose OS foi utilizado como sistema operacional em todos os dispositivos, os quais além de gerenciar os dados internos, também integrava os dados que o hardware coletava com a nuvem. Nesse contexto, o AWS IoT foi utilizado como broker MQTT, recebendo as mensagens publicadas e enviando aos clientes cujo tópico estivessem assinados. Para esse tipo de aplicação, podemos utilizar diversos tipos de cliente, entre os mais comuns se destacam aplicativos para smartphones e plataformas online para computadores de mesa. Para o caso deste projeto, foi utilizada a plataforma de testes disponibilizada pela própria AWS IoT, cujo recursos incluem assinar e/ou publicar tópicos na rede.

Uma representação da arquitetura do projeto, do ponto de vista de software, pode ser visualizada na Figura 4.1.

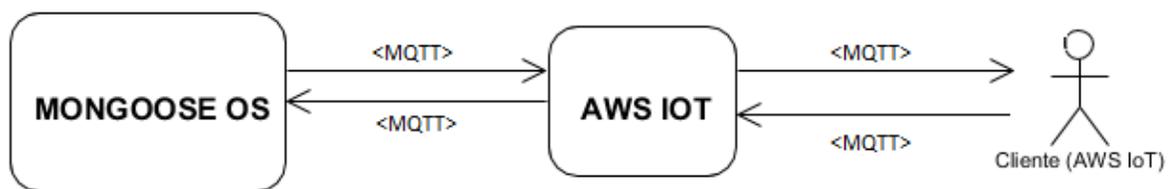


Figura 4.1: Diagrama lógico da Arquitetura de software montada

4.1.1 JSON Schema Validator

JSON Schema Validator é uma ferramenta web criada pela NewtonSoft, o qual permite validar se um JSON criado a partir de um modelo existente, de fato atende aos requisitos do modelo. O mesmo dispõe de duas colunas, onde se coloca o esquema que queremos adotar em uma, e na outra o JSON com a mensagem criada a partir desse modelo. Dessa forma, a plataforma compara e verifica se está tudo de acordo com o esquema, apontando erros e sugerindo modificações caso algo não esteja seguindo os padrões requeridos ⁷.

A interface da plataforma é ilustrada na Figura 4.2.

⁷<https://www.jsonschemavalidator.net>

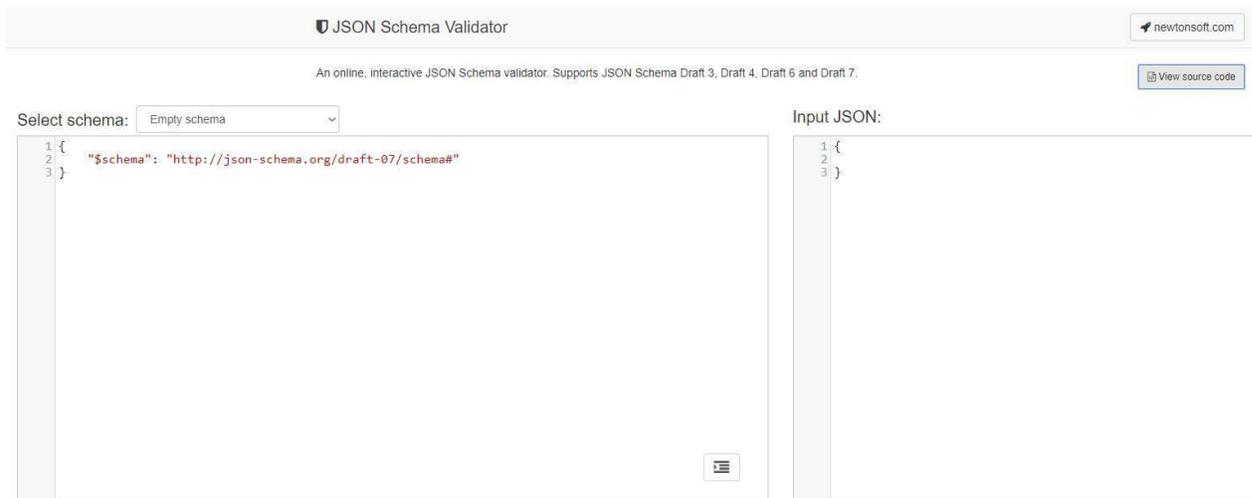


Figura 4.2: Interface Gráfica da plataforma JSON Schema Validator

Fonte: <https://www.jsonschemavalidator.net>

4.2 Validação

As subseções a seguir detalham como foi realizado o processo de validação da solução proposta à problemática.

4.2.1 Rotina do MQTT

Como todo protocolo, temos uma rotina de trabalho que sempre deve acontecer para o envio e recebimento de mensagens via MQTT. Em uma transação normal de dados, temos os seguintes procedimentos:

- 1. Um cliente se conecta ao broker e inicia o processo de publicação, enviando a mensagem que é endereçada ao broker através do uso de um tópico específico, o qual referencia a mensagem.
- 2. Outros clientes conectados ao broker solicitam acesso à mensagem publicada, na forma de assinaturas prévias no tópico da mensagem em questão.
- 3. O broker por fim, encaminha a mensagem a todos os clientes validados que assinaram o tópico da mensagem requisitada.

Em termos de validação, foi observado se a rotina do protocolo foi atendida de maneira correta, sem atrasos ou erros na mensagem, independente da rede utilizada. Após alguns testes consecutivos e repetitivos, foi constatada a execução correta da rotina MQTT proposta.

4.2.2 Validando Interoperabilidade

Para a correta validação do modelo de dados segundo o padrão do OneIoT, foi utilizado o validador Online *JSON Schema Validator*. Para isso, na ferramenta foi introduzido o modelo de dados que o recurso temperatura deveria seguir, o qual foi extraído do próprio *OCF OneIoT*.

Analisando o modelo apresentado na Figura 4.3, é possível verificar que o recurso temperatura tem um ID, uma temperatura que é um número que quantifica o recurso, e a unidade, que representa qual a escala termométrica que a temperatura referencia.

Assim, aplicando as informações utilizadas na plataforma, houve êxito na validação, com a mesma retornando uma mensagem de sucesso abaixo da coluna de dados à esquerda, que pode ser traduzida como: *Nenhum erro encontrado. Esquema JSON validado*⁸.



Figura 4.3: Validação do JSON

Fonte: <https://www.jsonschemavalidator.net>

Portanto, as mensagens transportadas e processadas via MQTT na solução proposta estão de acordo com os esquemas de formatação do OneIoTa.

4.2.3 Analisando Cenários de Uso

A seguir, foram realizados testes em vários possíveis cenários de uso do sistema, cujo critério de aceitação é de que o status de cada cenário ao fim dos testes fosse alcançado. Em todos os testes a seguir, o critério de aceitação foi atendido, validando o funcionamento esperado do sistema proposto.

Os seguintes cenários foram avaliados:

1. Cenário de validação para a descoberta de dispositivos ativos, descrito na Tabela 4;
2. Cenário de validação para o acesso a todos os recursos de um dispositivo específico, descrito na Tabela 5;
3. Cenário de validação para a descoberta de todos os dispositivos publicando um recurso específico, descrito na Tabela 6;
4. Cenário de validação para o acesso a recurso específico de dispositivos específicos, descrito na Tabela 7;

⁸Tradução livre do autor

5. Cenário de validação para o acesso a todos os recursos de todos os dispositivos, descrito na Tabela 8;
6. Cenário de validação para o acesso de recurso de dispositivo desconectado da rede. descrito na Tabela 9.

Tabela 4: Descoberta de Dispositivos Ativos.

Consideração Inicial	Usuário deseja saber quais dispositivos estão conectados em um instante qualquer.
Fluxo Normal do Cenário	Usuário assina o tópico <i>/OCF/devices/#</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada.
Status do Cenário ao Final	Usuário recebe mensagem de cada coisa conectada, contendo ID do dispositivo e do recurso que o mesmo contém de todos os clientes conectados à rede naquele instante.

Tabela 5: Acesso a Todos os Recursos de um Dispositivo Específico

Consideração Inicial	Usuário deseja saber qual recurso um dispositivo conhecido e conectado está publicando
Fluxo Normal do Cenário	Usuário assina no tópico <i>/OCF/rt/+/(ID do dispositivo online)</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada.
Status do Cenário ao Final	Usuário recebe mensagem retornando um JSON com a ID do dispositivo que fornece os recursos, bem como os recursos em si, apenas do dispositivo especificado.

Tabela 6: Descoberta de Todos os Dispositivos Publicando um Recurso Específico

Consideração Inicial	Usuário deseja saber quais dispositivos publicam um recurso já conhecido.
Fluxo Normal do Cenário	Usuário assina o tópico <i>“/OCF/rt/nome-do-recurso/#”</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada.
Status do Cenário ao Final	Usuário recebe mensagem retornando um JSON contendo a ID do dispositivo que fornece o recurso, bem como o recurso em si, formatado com o modelo de dados do ONEIOTA, para cada dispositivo conectado na rede naquele momento.

Tabela 7: Acesso a Recurso Específico de um Dispositivo Específico

Consideração Inicial	Usuário deseja saber qual o valor de um recurso conhecido em um dispositivo específico
Fluxo Normal do Cenário	Usuário assina o tópico <i>“/OCF/rt/nome-do-recurso/ID-do-dispositivo”</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada.
Status do Cenário ao Final	Usuário recebe mensagem retornando um JSON contendo a ID do dispositivo que fornece o recurso, bem como o recurso em si, formatado com o modelo de dados do ONEIOTA, para o dispositivo escolhido.

Tabela 8: Acesso a Todos os Recursos de Todos os Dispositivos.

Consideração Inicial	Usuário deseja saber qual o valor de todos os recursos em todos os dispositivos.
Fluxo Normal do Cenário	Usuário assina o tópico <i>"/OCF/rt/#</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada.
Status do Cenário ao Final	Usuário recebe mensagem retornando um JSON contendo a ID do dispositivo que fornece o recurso, bem como o recurso em si, formatado com o modelo de dados do ONEIOTA, para cada recurso de todos os dispositivos publicando na rede.

Tabela 9: Descoberta de Recursos de Dispositivos Desconectados

Consideração Inicial	Usuário deseja saber qual recurso um dispositivo conhecido e desconectado estava publicando
Fluxo Normal do Cenário	Usuário publica uma mensagem qualquer no tópico reservado <i>\$aws/things/thingName/shadow/get</i>
O Que Pode Falhar	Instabilidades da rede podem atrasar retorno da solicitação requisitada ou acusar falsos dispositivos desconectados.
Status do Cenário ao Final	Usuário recebe mensagem de cada coisa desconectada, contendo ID do dispositivo e do recurso que o mesmo contém do cliente offline especificado.

5 Conclusão

A Internet das Coisas é um conceito de comunicação que abrange os mais variados tipos de dispositivos e tecnologias, não é a toa que é um assunto bastante abordado atualmente, e a expectativa é que se popularize cada vez mais ao longo dos anos, quando a disponibilidade de coisas com capacidade de se conectar à Internet, e por consequência à outras coisas, utilizando os conceitos do IoT irá cada vez mais aumentar. Um dos motivos para esse aumento de expectativa nesse conceito de tecnologia é a capacidade de se expandir e se integrar em projetos já existentes, muitas vezes solucionando alguma dificuldade de maneira mais simples, além de integrar todas as suas coisas à nuvem, aumentando ainda mais as possibilidades de aplicação dessa ideia. Aliado a isso, existe o protocolo de comunicação MQTT atrelado aos propósitos de Internet das Coisas, no âmbito residencial ou comercial em geral.

Além disso, foi possível validar com precisão a principal problemática que este trabalho se propôs a solucionar, a qual consistiu no uso do modelo de dados OneIoTa aplicado ao protocolo MQTT e voltado à automação residencial. Com essa conquista, não apenas foi possível alcançar a interoperabilidade total entre os aparelhos com comunicação por MQTT, como também garantimos que, caso seja necessário utilizar o protocolo OCF como novo protocolo de comunicação por quaisquer motivos, a mesma interoperabilidade continuará existindo, visto que seu padrão de modelos de dados continua sendo obedecido.

Durante a execução deste projeto, algumas dificuldades foram observadas. A escolha de um sistema operacional integrado com protocolos para o IoT se tornou um desafio. Inicialmente o sistema operacional escolhido para ser utilizado junto aos microcontroladores ESP foi o sistema para a Internet das Coisas Zerynth [17], porém, devido algumas dificuldades técnicas sem solução até o momento, como falhas em sua comunicação via MQTT com sistemas em nuvem como o AWS IoT, forçaram a migração para o Mongoose OS, o qual foi descrito no trabalho. Dessa forma, dentre os impactos observados, a linguagem de programação a ser utilizada mudou de Python para JavaScript, adicionando novos desafios a implementação do sistema. Além disso, problemas de instabilidade de rede, como alta latência, dificultaram testes repetitivos ou cronometrados, gerando alguns desafios na fase de validação do trabalho.

Com as dificuldades enfrentadas neste projeto, foi possível identificar quais caminhos seriam os corretos a se seguir, sendo também os erros dessa forma um degrau para o conhecimento.

5.1 Trabalhos Futuros

Com o desenvolvimento das atuais tecnologias, a tendência é que cada vez mais as coisas sejam integradas à Internet, com capacidade de se comunicar entre si e com a nuvem, permitindo assim a aceleração da evolução dos dispositivos utilizados atualmente em IoT.

A perspectiva é que em um futuro próximo, sejam criados verdadeiros *ecossistemas tecnológicos*, capazes de realizarem tarefas de forma completamente autônoma, em sua totalidade. Com tanta necessidade por comunicação entre dispositivos, a demanda por protocolos de comunicação irá certamente aumentar, de tal forma que iremos contemplar o surgimento de novos protocolos de rede focados no IoT, seja residencial ou mesmo industrial, que precisarão de um padrão em comum a seguir para conseguirem se comunicar com dispositivos cujo protocolo de comunicação difere entre eles.

Com a uma padronização acertada, há um leque de novas possibilidades para o uso da Internet

das Coisas, seja no âmbito residencial, industrial ou acadêmico, já que desta forma, sem a limitação de dispositivos em se comunicarem entre si, há uma potencialização das aplicações de IoT. Além disso, com os resultados obtidos neste trabalho, é possível idealizar sistemas mais robustos e eficientes, mantendo a característica de interoperabilidade, podendo ser aplicado nos mais diversos campos da ciência, evidenciando a importância desta pesquisa e da realização deste Trabalho de Conclusão de Curso.

Referências

- [1] *Conhecendo o MQTT - IBM*. Disponível em: <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>
Acesso: 08 de Julho de 2018
- [2] MINERVA, Roberto; BIRU, Abyi; ROTONDI, Domenico. *Towards a definition of the Internet of Things (IoT)*. *IEEE Internet INitiative*, v. 1, 2015..
- [3] SKERRETT, I. *Case Study MQTT: Why Open Source and Open Standards Drive Adoption*. Disponível em: <https://ianskerrett.wordpress.com/2015/03/04/case-study-mqtt-why-open-source-and-open-standards-drives-adoption/>
Acesso: 10 de Julho de 2018
- [4] OneIoT *Website com descrição do OneIoT*. . Disponível em: <https://www.itsoniota.com/company/about-us/>
Acesso: 10 de Julho de 2018
- [5] OCF - Foundation. *Website com descrição Open Connectivity Foundation*. Disponível em: <https://openconnectivity.org/foundation>
Acesso: 10 de Julho de 2018
- [6] IOTIVITY. *Website com descrição IoTivity*. Disponível em: <https://iotivity.org/about>
Acesso: 10 de Julho de 2018
- [7] BANKS, Andrew; GUPTA, Rahul. *MQTT Version 3.1. 1. OASIS standard*, v. 29, 2014.
- [8] BARR, J. *AWS IoT, Cloud services for connected devices*. *AWS Blog*, 2015.
- [9] SANTOS, Bruno P. et al. *Internet das coisas: da teoriaa prática*. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2016.
- [10] HTTP - Berners-Lee, Tim, Roy Fielding, and Henrik Frystyk. *Hypertext transfer protocol*. HTTP/1.0. No. RFC 1945. 1996.
- [11] XMPP - Saint-Andre, P. (2011). *Extensible messaging and presence protocol (XMPP)*. Core (No. RFC 6120).
- [12] AMQP - Vinoski, S. (2006). *Advanced message queuing protocol*. *IEEE Internet Computing*, 10(6).
- [13] AWS IoT *Website com descrição do AWS IoT*. Disponível em: <https://aws.amazon.com/pt/iot/>
Acesso: 20 de Julho de 2018
- [14] Mongoose OS *Website com descrição do Mongoose OS*. Disponível em: <https://mongoose-os.com>
Acesso: 20 de Julho de 2018
- [15] JSON *Crockford, D. (2006). The application/json media type for javascript object notation (json)*. (No. RFC 4627).

- [16] Workgroup, R. A. M. L. (2015). *RAML-RESTful API modeling language*. . Disponível em: <http://raml.org/>
Acesso: 20 de Julho de 2018
- [17] Zerynth *Website com descrição Zerynth*. Disponível em: <https://www.zerynth.com/get-started/#what-is-zerynth>
Acesso: 20 de Julho de 2018