



## **PROJETO DE SISTEMA PARA INTERCONEXÃO E MAPEAMENTO DINÂMICO ENTRE DISPOSITIVOS EM UM CENÁRIO DE INDÚSTRIA 4.0**

**Alberi Medeiros Santos <sup>1</sup>, Danilo Freire de Souza Santos <sup>2</sup>**

### **RESUMO**

Com o avanço das novas tecnologias de conectividade em seus mais diversos níveis, novos cenários de aplicação e serviços surgem em diferentes ambientes. Considerando o conceito de Internet das Coisas (IoT), onde diversos dispositivos e equipamentos estarão conectados a Internet, seja sendo monitorados ou controlados, um dos cenários de importante relevância é a Indústria. Com isso, vislumbramos a tendência da quarta revolução industrial, ou indústria 4.0, onde as fábricas estarão conectadas e novos serviços surgirão decorrentes dessa conectividade. Neste sentido, um dos desafios em viabilizar tal cenário é transformar e preparar as indústrias atuais para esse ambiente futuro. Espera-se que com as indústrias conectadas novos serviços surjam, principalmente considerando toda a tecnologia inerente do poder da inteligência artificial. Para tanto, um dos principais desafios é coletar e disponibilizar de maneira padronizada dados e informações relevantes para tais sistemas inteligentes. Esse desafio se torna relevante considerando que os equipamentos em um chão de fábrica são os mais diversos, de modo que seus sistemas de sensoriamento disponibilizam dados em diferentes modelos. Com este desafio em mente, portanto, podemos definir como primeiro problema a ser investigado a necessidade de viabilizar interoperabilidade entre diferentes equipamentos e serviços de modo padronizado. Soma-se a esse desafio a necessidade de desenvolver aplicações de modo simples, e ainda, manter a interoperabilidade do sistema como um todo. Neste contexto, de viabilizar a coleta padronizada e interoperável entre dispositivos para viabilizar o desenvolvimento de aplicações para a Indústria 4.0, é onde se enquadra este projeto. O principal objetivo deste projeto foi investigar padrões de interoperabilidade para a Internet das Coisas com a aplicabilidade para a Indústria 4.0. Como resultado obtido foi criado e testado um projeto protótipo que permite integração de aplicações e dispositivos IoT e opera com baixa latência.

**Palavras-chave:** Internet of Things, Interoperability, Industry 4.0.

<sup>1</sup> Aluno do curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, UFCG, Campina Grande, PB, e-mail: alberi.santos@ee.ufcg.edu.br

<sup>2</sup> Doutor, Professor, Departamento de Engenharia Elétrica, UFCG, Campina Grande, PB, e-mail: danilo.santos@dee.ufcg.edu.br

## ***SYSTEM DESIGN FOR DYNAMIC INTERCONNECTION AND MAPPING BETWEEN DEVICES IN AN INDUSTRY 4.0 SCENARIO***

### **ABSTRACT**

With the advancement of new connectivity technologies at their most diverse levels, new application and service scenarios emerge in different environments. Considering the concept of Internet of Things (IoT), where several devices and equipment will be connected to the Internet, whether being monitored or controlled, one of the scenarios of important relevance is the Industry. With this, we glimpse the trend of the fourth industrial revolution, or industry 4.0, where factories will be connected and new services will emerge as a result of this connectivity. In this sense, one of the challenges in enabling such a scenario is to transform and prepare current industries for this future environment. It is expected that with the connected industries new services will emerge, especially considering all the inherent technology of the power of artificial intelligence. Therefore, one of the main challenges is to collect and make available in a standardized way data and information relevant to such intelligent systems. This challenge becomes relevant considering that the equipment on a factory floor is the most diverse, so that their sensing systems provide data in different models. With this challenge in mind, therefore, we can define as the first problem to be investigated the need to enable interoperability between different equipment and services in a standardized way. Added to this challenge is the need to develop applications in a simple way, and also maintain the interoperability of the system as a whole. In this context, to enable standardized and interoperable collection between devices to enable the development of applications for Industry 4.0, is where this project fits. The main objective of this project was to investigate interoperability standards for the Internet of Things with applicability for Industry 4.0. As a result, a prototype project was created and tested that allows integration of IoT applications and devices and operates with low latency.

**Keywords:** Internet of Things, Interoperability, Industry 4.0.

## INTRODUÇÃO

Uma das principais revoluções nos últimos anos vem do conceito de Indústria 4.0. Este conceito, apesar do seu nome, engloba diversas áreas afins tendo em comum o aspecto de conectividade. Como exemplo, podemos explorar cenários nos quais indústrias são controladas remotamente através de sistemas remotos com possibilidade feedback tátil em comandos, o no que é chamado de Internet Tátil (AAZAM; HARRAS; ZEADALLY, 2019). Apesar de todas essas possibilidades, desafios ainda existem na primeira camada de comunicação para viabilizar tal cenário, ou seja, em como viabilizar a comunicação dos mais diversos equipamentos na conhecida Internet das Coisas.

A Internet das Coisas (IoT), por sua vez, fornece a capacidade para humanos e computadores aprenderem e interajam com bilhões de “coisas” como sensores, atuadores, serviços e outros objetos conectados à Internet. A realização de sistemas de IoT permite a integração contínua do mundo cibernético com o mundo físico e muda fundamentalmente a capacidade da interação humana com o mundo (SATYANA-RAYANAN, 2017). Cisco estima 50 bilhões de dispositivos conectados à Internet até 2020. Dessa forma, com aumento do número de dispositivos inteligentes e conectados, cresce a tendência de utilização de frameworks para IoT.

A adoção desses frameworks pode melhorar o desenvolvimento de aplicativos IoT através de rápida implementação, interoperabilidade, facilidade de manutenção, segurança e flexibilidade tecnológica (DERHAMY et al., 2015). Com isto, várias atividades de pesquisa e desenvolvimento estão sendo realizadas principalmente com foco em novo protocolos e arquiteturas. Essas atividades, em geral, têm como objetivo oferecer melhorias em aspectos relacionados a interconexão entre objetos (coisas), eficiência energética e de comunicação, novas APIs, e o desenvolvimento e integração de novos serviços.

Em paralelo, e considerando o cenário de Indústria 4.0 onde diversos setores de uma fábrica podem estar distribuídos em grandes áreas geográficas, serviços de Computação na Borda surgem para suportar aplicações com baixíssima latência e, portanto, com melhores aspectos de Qualidade de Serviço (QoS) para a produção de bens (SISINNI; AL, 2018). Uma visão deste cenário pode ser vista na Figura 1.

Com isso, esse modelo de distribuição computacional viabiliza diversos tipos de aplicações que demandam operações com baixa latência ou aspectos de tempo real. Apesar dessas aplicações poderem ser utilizadas em diferentes domínios, como na saúde (CARROLL; CNOSSEN; SIMONS, 2007) (ZHAO; WANG; NAKAHIRA, 2011), os sistemas de Computação na Borda e IoT podem ser amplamente explorados em ambientes industriais. Por exemplo, podem ser utilizados como sistemas de segurança em ambientes industriais, avaliando em tempo real os dados gerados por sensores, atuadores e controladores distribuídos na rede industrial, e realizando ações de segurança

Figura 1 – Visão comparativa entre diferentes níveis computacionais.



Fonte: Dados da pesquisa (2020)

e controle quando necessário (SISINNI; AL, 2018).

Com o cenário apresentado, alguns desafios surgem no que tange a integração dos dispositivos nas redes industriais de primeiro nível, ou seja, na integração das “coisas” do chão de fábrica de uma indústria. Um dos principais desafios para o processamento de dados nos serviços está em como interpretar informações heterogêneas de diversos dispositivos na rede de borda (SHI; AL, 2016). Por se tratar de dispositivos com diferentes propósitos e projetos cada um tende a seguir diferentes normas e padrões de comunicação.

Portanto, de modo a ser passível de oferecer serviços inteligentes e interoperáveis aos nós da rede, um serviço ou gateway na borda em uma indústria deve ser capaz de interpretar os dados de modo homogêneo. Nesse sentido, um dos serviços base seria a realização da transcodificação de dados de dispositivos da rede para um formato único e interoperável, de modo a facilitar o desenvolvimento de serviços de processamento na borda da rede.

Destaca-se aqui também a demanda cada vez maior do uso de inteligência artificial para o desenvolvimento de serviços para a indústria. Esses sistemas inteligentes precisam interpretar os dados para a tomada de decisão, e o uso de dados padronizados pode facilitar bastante tal processo.

Além do mapeamento e padronização, estes dispositivos interoperáveis em gateways devem ser capazes de executar aplicações diretamente em seus ambientes. Estas aplicações devem ser facilmente programáveis, de modo a viabilizar uma maior agilidade na configuração dele. Essa configuração dinâmica pode ser realizada através do uso de modelos, em um ambiente de programação dinâmico oferecido por

arcabouços de IoT.

Para viabilizar essa configuração dinâmica, arcabouços como o NODE-Red e o Calvin IoT oferecem ambientes dinâmicos para a programação de aplicações IoT utilizando atores e/ou modelos. Em relação ao uso de padrões de interoperabilidade, padrões como o OCF e o OPC-UA oferecem uma padronização semântica dos dados para diferentes domínios, de modo a facilitar sua manipulação por diferentes equipamentos.

Nesse cenário de interoperabilidade e configuração dinâmica para gateways industriais é onde se enquadra esse trabalho. Sendo assim, foram realizadas investigações e comparações em cenários diferentes utilizando ferramentas e tecnologias como o Node-RED, protocolo MQTT, plataforma de nuvem Azure, plataforma de nuvem AWS e DTDL (Digital Twins Definition Language), visando o desenvolvimento de um cenário mais completo simulando o chão de fábrica de uma indústria 4.0. Com isso, obteve-se um projeto protótipo para a comunicação com dispositivos na Internet das Coisas utilizando os padrões explorados, com foco em uma aplicação industrial.

## **MATERIAIS E MÉTODOS**

A pesquisa foi conduzida de acordo com uma metodologia Project-based Learning e Problem-solving Learning, sendo assim, realizada a resolução de problemas inerentes ao domínio de aplicação industrial, nos quais se fez necessário o uso das seguintes tecnologias, alguns conceitos importantes do protocolo MQTT, ferramenta NODE-Red, arcabouços de nuvem para IoT e digital twins.

### *Protocolo MQTT*

Message Queuing Telemetry Transport (MQTT) é um padrão de comunicação muito utilizado no campo da indústria por ser leve e eficiente, permitir comunicações bidirecionais, escalabilidade para conexão de milhões de dispositivos IoT, confiabilidade e suporte e segurança (MQTT, 2021). Foi desenvolvido originalmente em 1999 por Andy Stanford-Clark (IBM) e Arlean Nipper (Arcom), hoje em dia as principais plataformas IoT, provedores de serviços em nuvem IoT e muitos gateways e dispositivos IoT Edge suportam conectividade com MQTT (HIVEMQ, 2021).

MQTT é um protocolo de publish/subscribe leve e requer largura de banda mínima para conectar um dispositivo IoT, além disso, diferente do padrão HTTP ele é orientado por eventos e permite que as mensagens sejam enviadas aos clientes (HIVEMQ, 2021).

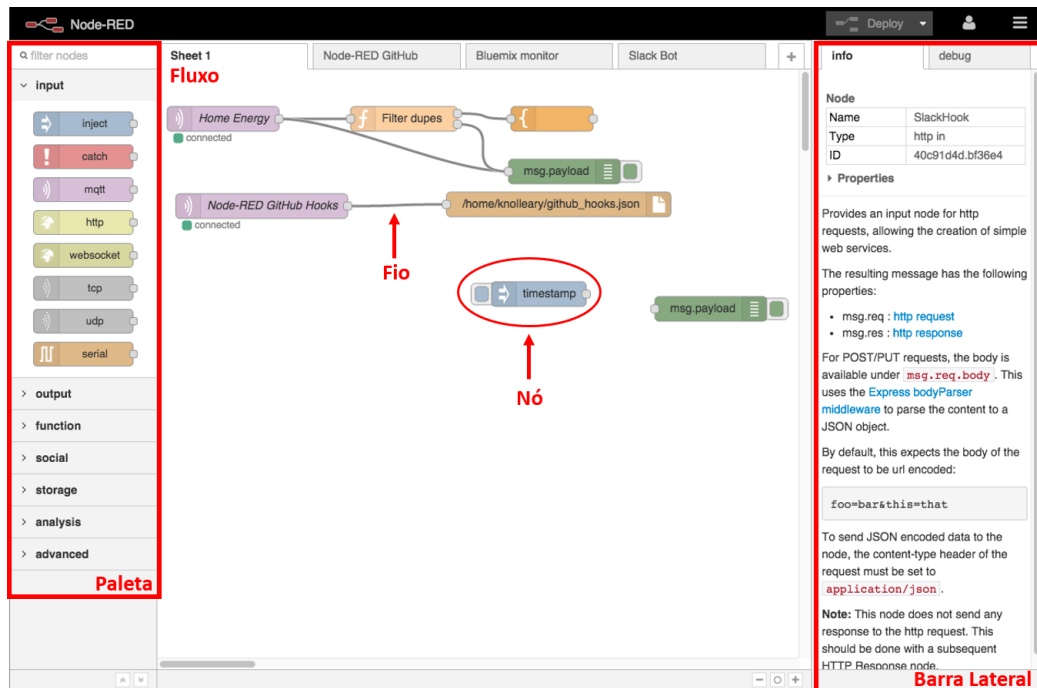
### *Node-RED*

O framework, para desenvolvimento de aplicações IoT (Internet of Things) baseado em modelos, Node-Red de código aberto utiliza o conceito de programação gráfica para conectar hardware de dispositivos, Application Programming Interface (APIs) e

serviços online como parte da Internet das Coisas ([AZURE.MICROSOFT.COM](https://azure.microsoft.com), 2021).

A sua programação é baseada em fluxo, o que facilita bastante a prototipagem de aplicações IoT utilizando poucas ou nenhuma linha de código. O fluxo é composto por a conexão de blocos de código predefinidos e reutilizáveis, chamados de nós, que trocam mensagens entre si. A sua interface intuitiva é mostrada na figura 2.

Figura 2 – Interface do Node-RED.



Fonte: Dados da pesquisa (2020)

### Azure IoT

Responsável por fornecer o portfólio de serviços e recursos, o Azure IoT da Microsoft é capaz de atender desde a borda até a nuvem, de acordo com as necessidades do desenvolvedor para melhorar o projeto e requisitos de setor ([AZURE.MICROSOFT.COM](https://azure.microsoft.com), 2021).

A plataforma possui vários recursos tais como serviços centrais IoT baseados em SaaS (software como serviço), serviço IoT Edge para dispositivos de borda, serviço IoT Hub para conectar, gerenciar e dimensionar dispositivos IoT, e o serviço IoT Digital Twin capaz de replicar e simular dispositivos físicos reais ([PANDUMAN; SUKARIDHOTO; TJAHJONO, 2019](#)).

### AWS IoT

O AWS IoT reúne os serviços de IoT para soluções de uso industrial, de consumo e comerciais, sendo altamente especializado e abrangente, da borda à nuvem, segurança multicamadas, integração superior com IA e com capacidade para bilhões de dispositivos e trilhões de mensagens ([AMAZON. . . , 2021](#)).

A AWS IoT possui diversos serviços como o Amazon FreeRTOS, o AWS IoT Greengrass, AWS IoT core que facilita a conexão de dispositivos com serviços em nuvem que pode ser combinado com o AWS IoT Things para exibir dados enviados pelos dispositivos, entre outros que fornecem alta escalabilidade e flexibilidade para a plataforma (PANDUMAN; SUKARIDHOTO; TJAHJONO, 2019).

### DTDL

Digital Twins Definition Language (DTDL) é uma linguagem para descrever modelos de dispositivos IoT Plug and Play, dispositivos digital twins e digital twins lógicos, sendo assim, a modelagem permite que soluções IoT configurem digital twins de todos os tipos em uma única solução (DIGITAL..., 2021). Usando o DTDL os dispositivos são descritos usando o JSON-LD (JavaScript Object Notation for Linked Data), variação do JSON.

Existem seis classes de metamodelos que compõem o DTDL: Interface, Telemetria, Propriedade, Comando, Relacionamento e Componente. Além disso, o DTDL fornece uma linguagem de descrição de dados compatível com muitos formatos de serialização populares (DIGITAL..., 2021).

Na figura 3 é mostrado um modelo típico, escrito como uma interface DTDL. Ele descreve planetas, com nome, massa e temperatura. Além disso, o Planet modelo expressa conexões com outras entidades (Crater e Moon) (DTDL, 2021).

Figura 3 – Modelo de gêmeo digital.

<pre>[   {     "@id": "dtmi:com:contoso:Planet;1",     "@type": "Interface",     "@context": "dtmi:dtdl:context;2",     "displayName": "Planet",     "contents": [       {         "@type": "Property",         "name": "name",         "schema": "string"       },       {         "@type": "Property",         "name": "mass",         "schema": "double"       },       {         "@type": "Telemetry",         "name": "Temperature",         "schema": "double"       }     ]   },   {     "@type": "Relationship",     "name": "satellites",     "target": "dtmi:com:contoso:Moon;1"   },   {     "@type": "Component",     "name": "deepestCrater",     "schema": "dtmi:com:contoso:Crater;1"   } ] {   "@id": "dtmi:com:contoso:Crater;1",   "@type": "Interface",   "@context": "dtmi:dtdl:context;2" }, {   "@id": "dtmi:com:contoso:Moon;1",   "@type": "Interface",   "@context": "dtmi:dtdl:context;2" } ]</pre>	<pre>     "@type": "Relationship",     "name": "satellites",     "target": "dtmi:com:contoso:Moon;1"   },   {     "@type": "Component",     "name": "deepestCrater",     "schema": "dtmi:com:contoso:Crater;1"   } ] {   "@id": "dtmi:com:contoso:Crater;1",   "@type": "Interface",   "@context": "dtmi:dtdl:context;2" }, {   "@id": "dtmi:com:contoso:Moon;1",   "@type": "Interface",   "@context": "dtmi:dtdl:context;2" } ]</pre>
---	---

Fonte: Adaptada [14]

Para o acompanhamento do projeto, aplicou-se a metodologia de desenvolvimento ágil Scrum aplicado a pesquisa tecnológica e inovação, com objetivo de reduzir ao mínimo possível a sobrecarga de processos e facilitar as mudanças e a interação entre o orientando e orientador durante todo o desenvolvimento do projeto. Sendo

assim, o projeto foi dividido em pequenas tarefas envolvendo análise e apresentação de resultados das pesquisas e referencial teórico, além de tarefas específicas para o desenvolvimento do projeto. As atividades eram realizadas e a cada 15 dias feita a apresentação e discussão do andamento da tarefa, possíveis bloqueios que impedia seu desenvolvimento e os passos a seguir.

As atividades foram realizadas priorizando cenários simulados para a validação dos resultados, considerando as restrições de distanciamento social inerentes ao cenário atual de pandemia, desta maneira, está sendo possível realizar a maioria das atividades sem a necessidade de acesso ao laboratório.

## **DESENVOLVIMENTO**

Inicialmente, foram desenvolvidos dois microprojetos para Internet das Coisas e a Indústria utilizando as tecnologias alvos do projeto, sendo assim:

O primeiro microprojeto desenvolvido buscou solucionar um problema encontrado em um exemplo de cenário industrial, no qual, foi necessário realizar a comunicação entre um dispositivo simulado e a nuvem. Para a resolução do problema foi utilizado o framework NODE-Red que possibilitou a configuração eficiente de um dashboard simples capaz de simular um dispositivo com botões de interação e mensagens de status, características comumente encontradas em cenário de indústria, também foi utilizada a plataforma de computação em nuvem Amazon Web Services (AWS) e seus serviços que possibilitaram a comunicação com o framework por meio do protocolo de comunicação MQTT.

O segundo microprojeto buscou implementar dispositivos digitais que podem simular condições de desempenho do dispositivo real, conceitos de Digital-Twin (DT), além da comunicação entre dispositivo simulado e nuvem. O framework utilizado para simular o dispositivo continuou sendo o NODE-Red, no entanto, o modelo de fluxo foi alterado. A plataforma de computação em nuvem adotada para esse projeto foi a Azure da Microsoft, tendo em vista seu serviço Azure Digital-Twin. Também foram utilizados o serviço Azure Functions e o Hub IoT que usa dispositivos IoT Plug and Play compatível com o protocolo MQTT.

Na segunda fase do projeto, foi alterado o cenário do segundo microprojeto de forma a aumentar a quantidade de dispositivos e variação coordenada da taxa de envio de mensagens. Para isso, foi confeccionado um diagrama inicial com os dispositivos a serem simulados no Node-RED, alterado o fluxo do Node-RED adicionando novos dispositivos e sensores simulados, adicionados novos dispositivos ao Hub IoT e adaptada a função do Azure para atualizar diferentes sensores com o mesmo código. Por fim, foram realizados testes de latência e taxa de transmissão da solução implantada com intuito de mensurar sua eficiência. Para realizar os experimentos foi utilizada a



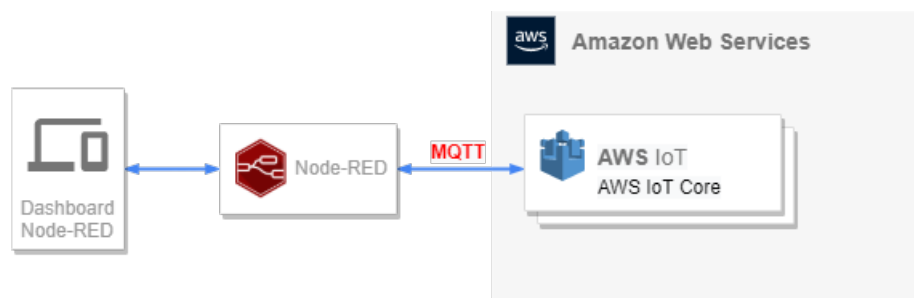
ferramenta wireshark observando a comunicação entre o Node-RED e a nuvem.

## RESULTADOS E DISCUSSÕES

### Primeiro microprojeto

Focando a interoperabilidade, o cenário utilizado para o primeiro microprojeto realiza a comunicação entre o dispositivo simulado por meio de um Dashboard simples implementado no framework NODE-Red e a nuvem usando a plataforma AWS e seus serviços para comunicação IoT, nesse caso foi escolhido o protocolo MQTT. Este projeto é ilustrado na Figura 4.

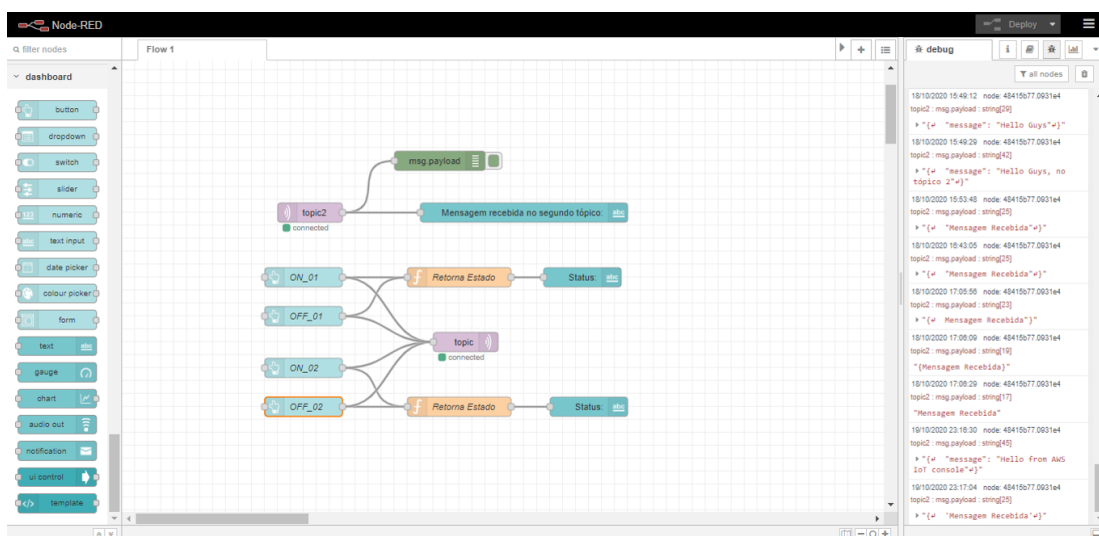
Figura 4 – Cenário proposto para o primeiro microprojeto.



Fonte: Elaborado pelo autor.

Para o desenvolvimento do dispositivo simulado foi utilizada a ferramenta para programação baseada em fluxo, Node-RED, a qual facilita a prototipagem de aplicações IoT, o fluxo desenvolvido pode ser observado na figura 5.

Figura 5 – Modelo configurando o dashboard e comunicação entre NODE-Red e a nuvem.



Fonte: Dados da pesquisa (2021).

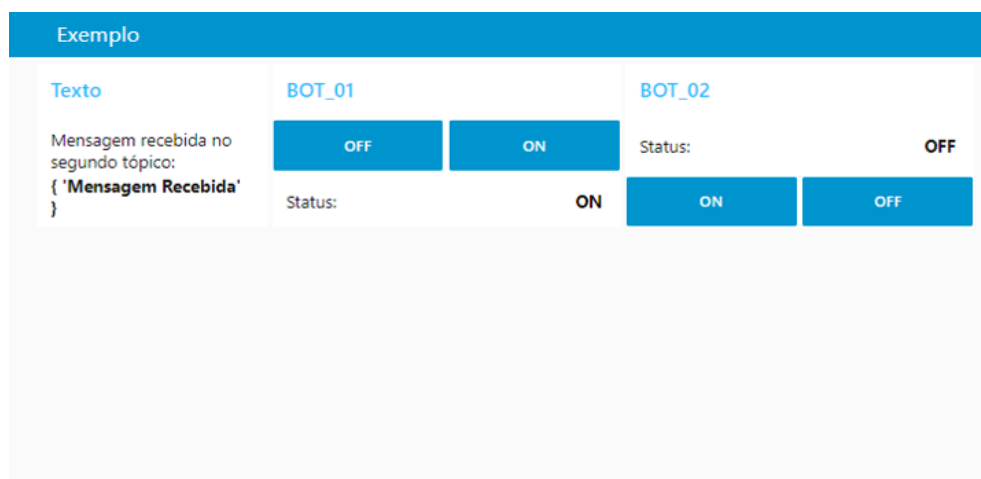
A plataforma de computação em nuvem escolhida para esse microprojeto foi a Amazon Web Services (AWS), na qual, foi utilizado o serviço AWS IoT Core que gerencia e possibilita a conexão com grande quantidade de dispositivos e aplicativos de

nuvem, além de ser confiável e possuir suporte a diversos protocolos de comunicação, como o MQTT. Principais nós utilizados para gerar o fluxo:

- mqtt-in: utilizado para assinar e escutar o tópico do MQTT conectado com o AWS IoT Core;
- ui-button: adiciona botões ao dashboard para que o usuário possa interagir;
- function: nó programável, utilizado para criar uma estrutura condicional e avaliar o estado do botão;
- ui-text: recebe as mensagens do tópico assinado e atualiza como texto no dashboard;
- mqtt-out: assina o broker MQTT e publica o status dos botões.

O fluxo mostrado é responsável por gerar de forma eficiente e rápida um dashboard capaz de simular um dispositivo com botões de liga e desliga, além de exibir mensagens de status da comunicação, por meio deste, é possível obter interações com o usuário e enviar diretamente para a nuvem. A figura 6 mostra o dashboard.

Figura 6 – Dashboard capaz de realizar interação com o cliente.

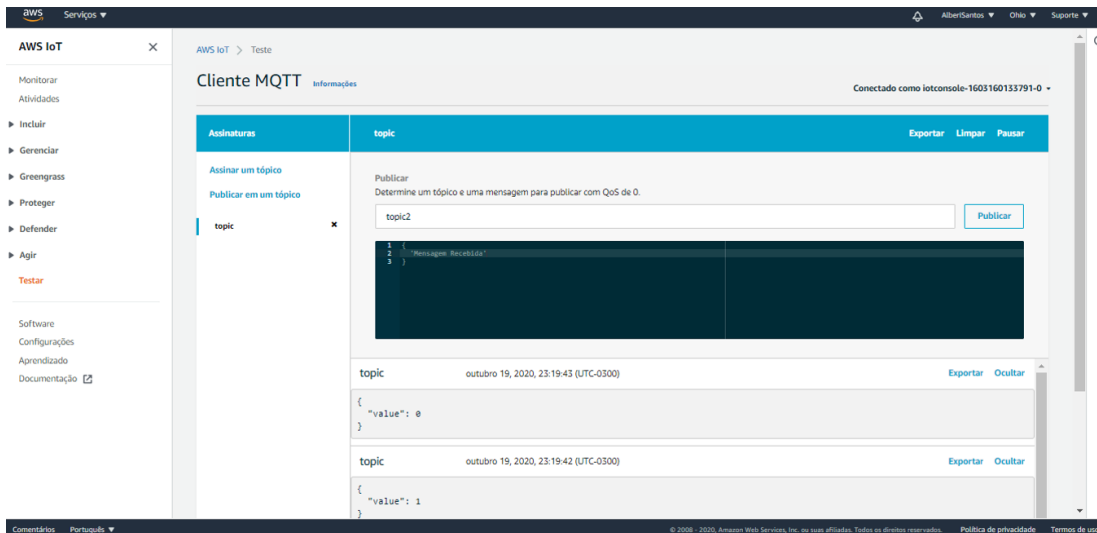


Fonte: Dados da pesquisa (2021).

No AWS IoT Core, foi criado um dispositivo individual (“coisa”) para receber a comunicação enviada a partir do nosso gateway implementado no Node-RED, para isso foi utilizado o protocolo de comunicação MQTT. A figura 7 mostra os testes realizados para confirmar eficiência da comunicação.

O protocolo MQTT se mostrou eficiente para a realização da comunicação entre dispositivos e seu uso foi facilitado com o serviço AWS IoT, tornando mais eficiente e rápida, além de transparente, no entanto, a plataforma de computação em nuvem exige um extenso protocolo de segurança, sendo necessária a utilização de diversas

Figura 7 – Mensagens enviadas e recebidas nos tópicos do MQTT na nuvem.



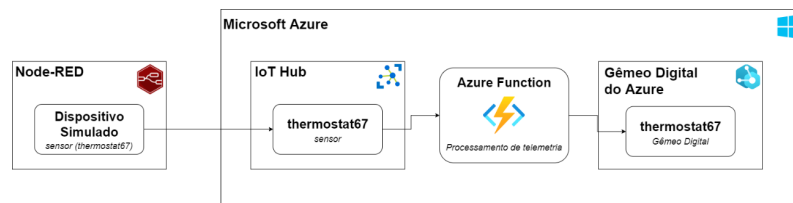
Fonte: Dados da pesquisa (2021).

chaves para a comunicação eficiente. O framework NODE-Red com sua lista de nós de código aberto se mostrou uma boa opção para Gateway, possibilitou a comunicação de maneira rápida e simples entre a plataforma de computação em nuvem e o dispositivo simulado.

### Segundo microprojeto

Aplicando conceitos de Digital Twin, no segundo microprojeto foi implementada uma solução visando a simular a comunicação de um sensor utilizado na indústria e a nuvem. Utilizando o Node-RED foi configurado um fluxo, o qual envia e recebe dados simulados de temperaturas diferentes para a plataforma Azure de computação em nuvem. O cenário proposto é representado na Figura 8 seguir.

Figura 8 – Cenário proposto para o segundo microprojeto.



Fonte: Dados da pesquisa (2021).

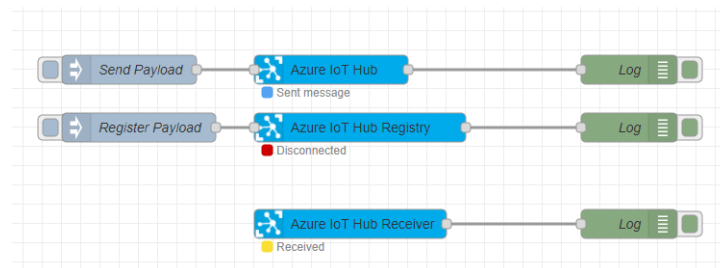
Foram utilizados os seguintes serviços da plataforma Azure:

- IoT Hub: serviço que possibilitou a comunicação bidirecional dos dispositivos com a nuvem;
- Azure Functions: utilizado para criar as funções que auxiliam o processamento de fluxos IoT, executar códigos solicitados a partir de eventos, entre outros;

- Azure Digital Twins: cria grafos de conhecimento baseado em modelos digitais de ambiente inteiros;
- Event Grid: permite compilar facilmente aplicativos com arquiteturas baseadas em eventos.

Inicialmente foi montado um fluxo, utilizando o Node-RED, para simular um dispositivo IoT. O fluxo utilizado está conectado diretamente com o Hub IoT do Azure, dessa forma ele pode gerar novos dispositivos IoT diretamente sem precisar acessar a plataforma do Azure, além disso, ele é responsável por estabelecer a comunicação com a nuvem utilizando o protocolo MQTT (Figura 9).

Figura 9 – Modelo de fluxo para configurar comunicação com a nuvem e dispositivo simulado.



Fonte: Dados da pesquisa (2021).

Em seguida foram realizados testes para confirmar a comunicação entre o dispositivo configurado no NODE-Red e o Hub IoT, os resultados são mostrados na figura 10 a seguir, na qual, podemos observar os dados do dispositivo simulado no NODE-Red sendo recebidos no Hub IoT.

Figura 10 – Monitoramento de mensagens no Hub IoT.

```
C:\Program Files\Microsoft SDKs\Azure\.NET SDK\v2.9>az iot hub monitor-events --output table --hub-name recurso-hubiot
Starting event monitor, use ctrl-c to stop...
event:
  component: ''
  interface: ''
  module: ''
  origin: thermostat67
  payload: '59'
event:
  component: ''
  interface: ''
  module: ''
  origin: thermostat67
  payload: '82'
```

Fonte: Dados da pesquisa (2021).

Assim, com a configuração do fluxo e da instância do Hub IoT pronta, foi criada uma instância Digital Twin (DT) e utilizado um modelo pré-criado de gêmeo digital que descreve suas propriedades usando a linguagem de programação independente Digital Twin Definition Language (DTDL) baseado em JSON-LD. O modelo em questão possui a única propriedade para descrever um termostato, mostrado na figura 11.

Figura 11 – Modelo de dados para o Digital-Twin.

```
{
  "@id": "dtmi:contosocom:DigitalTwins:Thermostat;1",
  "@type": "Interface",
  "@context": "dtmi:dtdl:context;2",
  "contents": [
    {
      "@type": "Property",
      "name": "Temperature",
      "schema": "double"
    }
  ]
}
```

Fonte: Dados da pesquisa (2021).

Por fim, foi configurado aplicativo de funções para realizar a telemetria, sendo responsável por processar os dados do Hub IoT e atualizar a propriedade de temperatura do DT de acordo com os dados recebidos do dispositivo simulado no Node-RED. A figura 12 ilustra o resultado sendo confirmado.

Figura 12 – a. Mensagem sendo enviada do dispositivo Node-RED; b. Variável do gêmeo digital sendo atualizada de acordo com os dados enviados.



Fonte: Dados da pesquisa (2021).

### Cenário simulando chão de fábrica de uma indústria

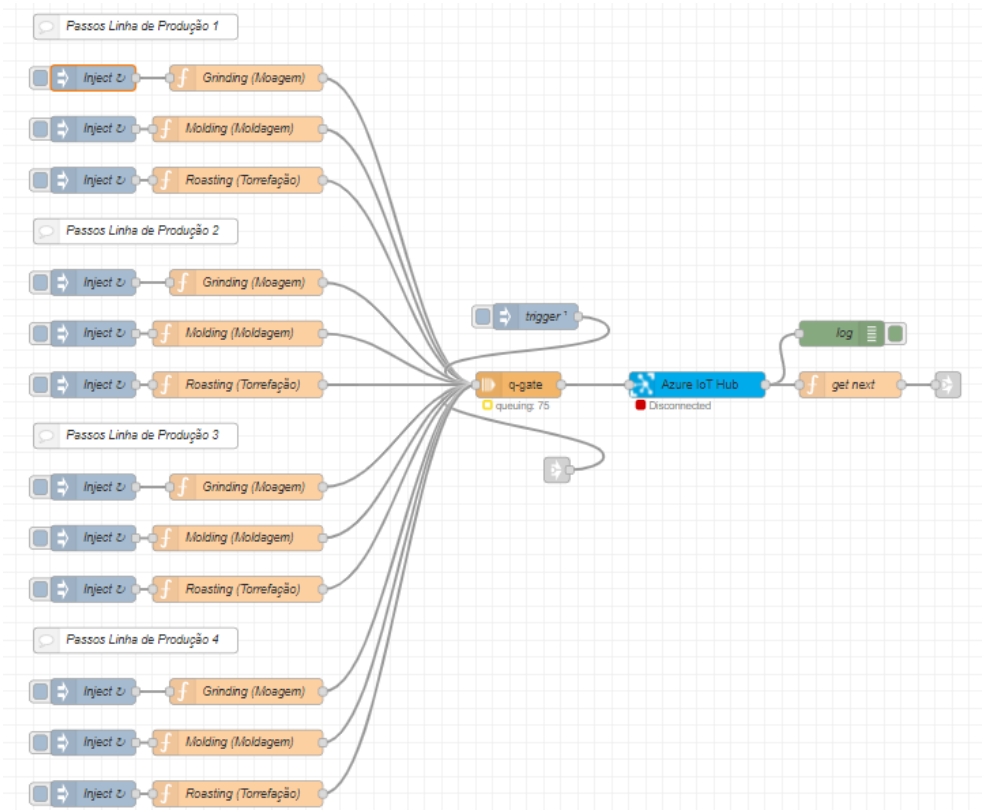
O fluxo do Node-RED usado no segundo projeto foi alterado para enviar dados aleatórios de cada sensor em um período de tempo fixo. Para isso foi alterado o nó inject e adicionado um nó função para cada dispositivo.

O novo fluxo do Node-RED completo simula 12 dispositivos do chão de fábrica. Foi criado um dispositivo gêmeo no Hub IoT para cada dispositivo simulado no Node-RED, esses dispositivos recebem as mensagens enviadas do Node-RED utilizando o protocolo MQTT e geram eventos para acionar as funções do Azure que vão tratar as mensagens e atualizar o Gêmeo Digital.

Nesse novo cenário foi encontrado um erro que gerava atrasos de aproximadamente um segundo, esse erro acontecia, pois, o nó utilizado necessita reestabelecer a conexão sempre que altera o dispositivo de destino, com isso, eram perdidas as mensagens que chegavam durante esse processo. Para minimizar a perda de mensagens,

foi adicionada uma fila ao fluxo que evitou a perda, no entanto a taxa de comunicação continuou sendo afetada por esse erro. A figura 13 mostra o fluxo finalizado.

Figura 13 – Nova composição do fluxo do Node-RED



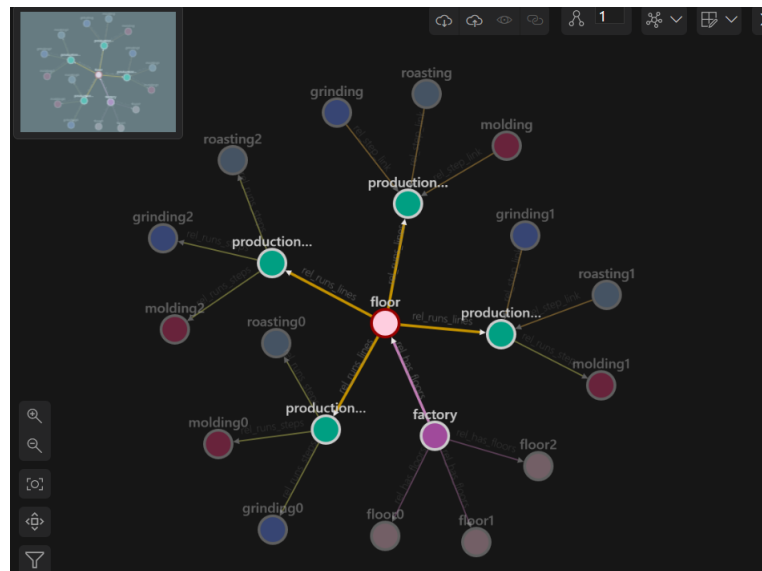
Fonte: Dados da pesquisa (2021).

Foram adicionados mais gêmeos digitais para uma análise com vários dispositivos se comunicando ao mesmo tempo, a Figura 14 mostra o grafo com todos os gêmeos digitais e suas conexões. Para esse cenário, utilizou-se sete modelos de linguagem de definição:

- Factory: Possui como propriedades o nome e localização da fábrica, além disso, possui relacionamento com os andares da fábrica;
- Floor: Suas propriedades são ID do andar e temperatura, relaciona-se com as linhas de produção;
- Production-line: As propriedades são ID da linha e status de operação, relaciona-se com as etapas da linha de produção;
- Production-step: Modelo base do qual as etapas com propriedades semelhantes podem herdar ID da etapa, hora de início, passo final, status da etapa e relacionamento com a linha de produção.

- Production-step-roasting: Além dos campos herdados, também possui as propriedades temperatura do chassi, tempo de funcionamento, consumo de energia e velocidade do ventilador;
- Production-step-grinding: Além dos campos herdados, possui temperatura do chassi, tempo para terminar o processo, consumo de energia e força das chapas atuadoras;
- Production-step-molding: Possui os campos herdados e tem as propriedade temperatura do chassi e consumo de energia.

Figura 14 – Novo grafo de conexões dos gêmeos digitais.



Fonte: Dados da pesquisa (2021).

Com o intuito de mensurar a eficiência da solução implantada, foram realizados experimentos para medir a latência. O gráfico ilustrado na figura 15 mostra o resultado obtido em experimento com 12 dispositivos conectados se comunicando simultaneamente e com 20 minutos de duração.

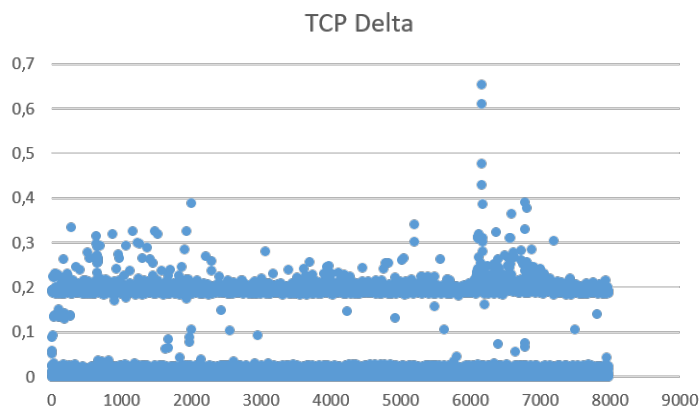
Foi possível observar no gráfico uma latência média de 88,85ms durante a realização do experimento.

Além da latência, também foram coletados os dados de taxa de transmissão. O gráfico obtido é mostrado na figura 16, no qual podemos observar a taxa de transmissão média de 1343,001 bytes por segundo durante o experimento.

## CONCLUSÃO

A metodologia Project-based Learning e Problem-solving Learning se mostrou eficiente no aprendizado, permitindo uma visão mais ampla do conteúdo e possibilitando o aprendizado aprofundando em desenvolvimento IoT.

Figura 15 – Gráfico da latência.



Fonte: Dados da pesquisa (2021).

O framework Node-RED é uma boa opção como gateway, principalmente por apresentar facilidade em sua implementação, com isso, economiza tempo do desenvolvimento. Além disso, é compatível com comunicação em diversas plataformas de computação em nuvem, assim como conexão com dispositivos IoT. O protocolo MQTT também corresponde a expectativa com boa comunicação entre os dispositivos.

Foram encontradas dificuldades ao implementar o serviço Digital Twin do Azure, mas foi possível contornar os problemas utilizando sugestões encontradas na documentação do Azure. Também houve problemas ao utilizar o nó de conexão Azure IoT Hub que limitou a taxa de transmissão das mensagens enviadas para a nuvem, com isso, os experimentos foram prejudicados, impedindo a realização de testes mais exigentes.

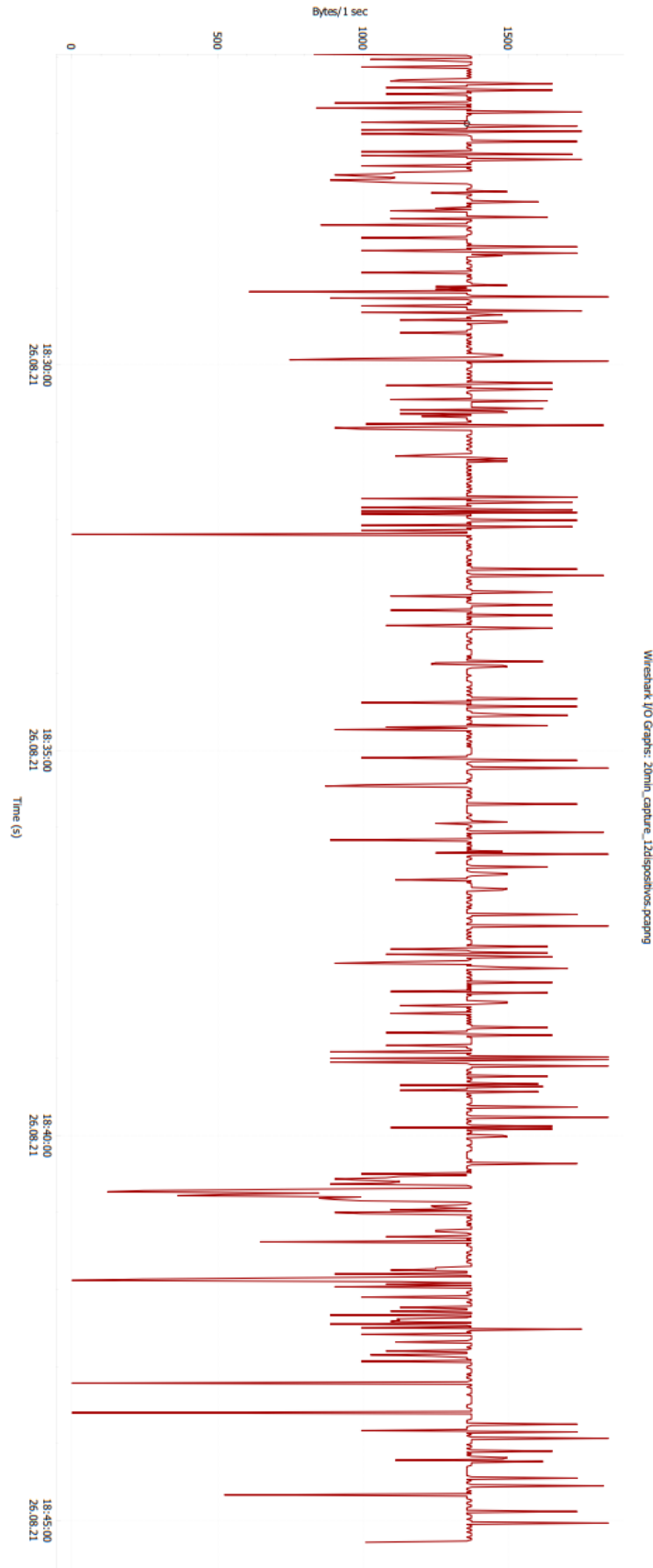
Apesar dos obstáculos enfrentados, podemos concluir que os objetivos do trabalho foram alcançados, tendo sido feito o estudo e avaliação de protocolos de comunicação de dispositivos utilizados por dispositivos industriais, padrões de interoperabilidade para Internet das Coisas, focados em interoperabilidade, arcabouços para integração de aplicações e dispositivos IoT, voltados para um desenvolvimento baseado em modelos e o desenvolvimento de projetos protótipos para a comunicação com foco em uma aplicação industrial.

## AGRADECIMENTOS

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil. Agradecemos ao PIBITI/CNPq-UFCG pela oportunidade para realização da pesquisa.



Figura 16 – Taxa de transmissão em bytes por segundo.



Fonte: Dados da pesquisa (2021).

## REFERÊNCIAS

AAZAM, M.; HARRAS, K. A.; ZEADALLY, S. Fog computing for 5g tactile industrial internet of things: Qoe-aware resource allocation model. **IEEE Transactions on Industrial Informatics**, v. 15, n. 5, p. 3085–3082, 2019. páginas 3

AMAZON Web Services, Inc. AWS IoT Applications Solutions. mar. 2021. Disponível em: <<https://aws.amazon.com/iot/>>. páginas 6

AZURE.MICROSOFT.COM. Azure iot | microsoft azure. mar. 2021. Disponível em: <<https://azure.microsoft.com/en-us/overview/iot/>>. páginas 6

CARROLL, D.; CNOSSEN, R.; SIMONS, D. Continua: an interoperable personal healthcare ecosystem. **IEEE Pervasive Computing**, v. 6, n. 4, p. 90–94, 2007. páginas 3

DERHAMY, H. et al. A survey of commercial frameworks for the internet of things. **2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)**, p. 1–8, 2015. páginas 3

DIGITAL Twins Definition Language. mar. 2021. Disponível em: <<https://github.com/Azure/opensdtw-dtdl/blob/master/DTDL/v2/dtdlv2.md>>. páginas 7

DTDL. mar. 2021. Disponível em: <<https://docs.microsoft.com/pt-br/azure/digital-twins/concepts-models>>. páginas 7

HIVEMQ. mar. 2021. Disponível em: <<https://www.hivemq.com/blog/how-to-get-started-with-mqtt/>>. páginas 5

MQTT. mar. 2021. Disponível em: <<https://mqtt.org/>>. páginas 5

PANDUMAN, Y. Y. F.; SUKARIDHOTO, S.; TJAHJONO, A. A survey of iot platform comparison for building cyber-physical system architecture. **2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)**, p. 238–243, 2019. páginas 6, 7

SATYANARAYANAN, M. The emergence of edge computing. **Computer**, v. 50, n. 1, p. 30–39, 2017. páginas 3

SHI, W. i.; AL et. Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, v. 3, n. 5, p. 637–646, 2016. páginas 4

SISINNI, E.; AL et. Industrial internet of things: Challenges, opportunities, and directions. **IEEE Transactions on Industrial Informatics**, v. 14, n. 11, p. 4724–4734, 2018. páginas 3, 4

ZHAO, W.; WANG, C.; NAKAHIRA, Y. Medical application on internet of things. **Proceedings of ICCTA**, p. 3–8, 2011. páginas 3