DISSERTAÇÃO DE MESTRADO

Arquitetura Base para Soluções de Internet das Coisas: Aplicações de Telemetria e Computação na Ponta Com Uso de Microsoft Azure nos modelos de IaaS, PaaS e SaaS

Jorge Andrade Seixas Maia

Brasília, dezembro de 2020

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

Arquitetura Base para Soluções de Internet das Coisas: Aplicações de Telemetria e Computação na Ponta Com Uso de Microsoft Azure nos modelos de IaaS, PaaS e SaaS

Jorge Andrade Seixas Maia

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica como requisito parcial para obtenção do grau de Mestre em Sistemas Mecatrônicos

Banca Examinadora

Prof. Eugênio Libório Feitosa Fortaleza, Ph.D, FT/UnB
Orientador _____

Prof. Daniel Mauricio Muñoz Arboleda, Dr, FT/UnB
Examinador Interno _____

Prof. Ivaldir Honorio de Farias Junior, Dr, UPE
Examinador Externo _____

FICHA CATALOGRÁFICA

MAIA, JORGE ANDRADE SEIXAS
Arquitetura Base para Soluções de Internet das Coisas: Aplicações de Telemetria e Computação na Ponta Com Uso de Microsoft Azure nos modelos de IaaS, PaaS e SaaS [Distrito Federal] 2020.
xvi, 136 p., 210 x 297 mm (ENM/FT/UnB, Mestre, Engenharia Mecânica, 2020).
Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Mecânica

1. Internet das Coisas	2. Protocolos
3. Edge Computing	4. Cloud Computing
I. ENM/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MAIA, J.A.S. (2020). *Arquitetura Base para Soluções de Internet das Coisas: Aplicações de Telemetria e Computação na Ponta Com Uso de Microsoft Azure nos modelos de IaaS, PaaS e SaaS*. Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 136 p.

CESSÃO DE DIREITOS

AUTOR: Jorge Andrade Seixas Maia
TÍTULO: Arquitetura Base para Soluções de Internet das Coisas: Aplicações de Telemetria e Computação na Ponta Com Uso de Microsoft Azure nos modelos de IaaS, PaaS e SaaS.
GRAU: Mestre em Sistemas Mecatrônicos ANO: 2020

Jorge Andrade Seixas Maia
Depto. de Engenharia Mecânica (ENM) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

RESUMO

A Internet das Coisas, embora seja um termo relativamente novo e com um entendimento muitas vezes difuso, surge no mercado de tecnologia como um agregador de novas soluções, frente as antigas comunicações máquina à máquina e tem se tornando peça fundamental na adição de inteligência aos cenários com necessidade de produtividade, confiabilidade da informação e melhoria de processos. Insumo básico para a Indústria 4.0, IoT é sinônimo de diminuição de custos em medições de serviços aplicados a diversos seguimentos, interligação de departamentos internos e principalmente um elo de conexão com o mundo físico, sem interação humana e consequentemente gerando uma maior confiabilidade dos dados.

Este trabalho se posiciona no apoio ao desenvolvimento de soluções de Internet das Coisas e seus desafios em dois cenários comuns e amplamente complexos que são respectivamente a telemetria, onde o foco é envio e recebimento de mensagens de dados e comandos, e o uso de computação na ponta, para uso de inteligência artificial ou processamento em tempo próximo ao real. Estes dois cenários têm diversas variações nas camadas da solução, tecnologias de telecomunicação, protocolos de aplicação, concentradores de mensagens, inteligência artificial, armazenamento de dados, análises dos dados e interoperabilidade. Esta dissertação tem como objetivo definir uma arquitetura de referência para os projetos que façam uso de telemetria e/ou de computação na ponta com uso de computação em nuvem, também definindo as principais camadas, protocolos de comunicação e tecnologias a serem utilizadas nos projetos, com aplicação na nuvem da Microsoft, o Azure.

Baseado em uma revisão da literatura exploratória a fim de garantir o estado da arte em arquiteturas e protocolos, seguida por simulações em ambiente de nuvem da Microsoft, este trabalho fará uso de aplicações nas verticais de gestão de infraestrutura e engenharia, controle de frotas de caminhões de lixo e integração de IoT com plantas industriais para suas experimentações.

Serão apresentadas as camadas que compõem a uma solução de IoT passando pelo dispositivo, seu enlace de comunicação, protocolos, segurança de comunicação e ingestão de mensagens permitindo para a posterior análise de dados. Arquiteturas possíveis e validadas por uso de dispositivos físicos e simulados compõe os resultados deste trabalho que se irá se utilizar de recursos de simulação em nuvem a fim também de se aplicar uma arquitetura de referência para projetos de telemetria e computação na ponta.

Palavras Chave: Internet das Coisas, Protocolos, Edge Computing, Cloud Computing.

ABSTRACT

The Internet of Things, although a relatively new term and with an often diffuse understanding, appears in the technology market as an aggregator of new solutions, facing the old machine to machine communications and has become a key part in adding intelligence to scenarios with a need for productivity, information

reliability and process improvement. Basic input for Industry 4.0, IoT is synonymous with cost reduction in service measurement applied to several segments, among which we can highlight agriculture, measurement and control of utilities, factories, logistics and facilities management, among others.

Scenarios that already make use of connectivity technologies, regularly have silos of information connected to a concentrator, but that do not communicate with each other, even being in the same physical environment. For these cases the use of IoT can do more than interconnect these islands of information, enabling more precise decision making, with less time and a higher level of confidence, with or without human interaction. The union of sales systems with production systems, is one of the examples that guarantees less time in the configuration of customized production lines and a consequent increase in competitiveness. Infrastructure management is another good case of use, in this we have application of Internet of Things generating more intelligent environments, ensuring savings in the use of resources and based on decision-making algorithms linked to the building, as well as the possibility of modeling a digital twin for management and remote monitoring in near real time.

This work is positioned to support the development of Internet of Things solutions and their challenges in two common and widely complex scenarios that are respectively telemetry, where the focus is on sending and receiving data messages and commands and the other is necessary the use of computing at the tip for use of artificial intelligence or processing in near real time. These two scenarios have several variations in the solution layers, telecommunication technologies, application protocols, message concentrators, artificial intelligence, data storage, data analysis and interoperability. This dissertation aims to define a reference architecture for projects that make use of telemetry and/or edge computing, also defining the main layers, communication protocols and technologies to be used in projects.

Based on a review of the exploratory literature in order to ensure the state of the art in architectures and protocols, followed by simulations in Microsoft cloud environment, this work will make use of applications in the verticals of infrastructure management and engineering, control of fleets of garbage trucks and integration of IoT with industrial plants.

The layers that make up a IoT solution will be presented, passing through the device, its communication link, protocols, communication security and message intake allowing for later data analysis. Possible and validated architectures through the use of physical and simulated devices compose the result of this work, which will be used from cloud simulation resources in order to also apply a reference architecture for telemetry and computational projects at the end.

Keywords: Internet of Things, Protocols, Edge Computing, Cloud Computing.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Internet das Coisas, traduzido do termo original Internet of Things, ou simplesmente IoT, aparece pela primeira vez no cenário de tecnologia em meados de 1999,em uma apresentação dentro da Procter & Gamble que almejava trazer à tona como gerir a cadeia de suprimentos por meio do uso de etiquetas de RFID, Identificação via Radiofrequência, e para chamar atenção dos executivos para a causa, Kevin Ashton, autor da apresentação, adicionou Internet no monitoramento dos itens rastreáveis por ser um termo que estava em alta na época (42). Baseada em uma etiqueta de Rádio frequência cuja patente remota de 1973, a Internet das Coisas criada por Ashton iria tomar forma no MIT dentro do Auto-ID Center Research Consortium. A ideia de Ashton à época era de que os computadores, instrumentos manipuláveis somente por humanos e que detinham o poder de conectividade e processamento deveriam prover cada vez mais dados e ser mais autônomos permitindo interação entre eles, bem como ter a inteligência necessária para ouvir, falar e entender o ambiente; na época, 1999, era a evolução mais provável da comunicação máquina para máquina, do inglês Machine to Machine (M2M) (43).

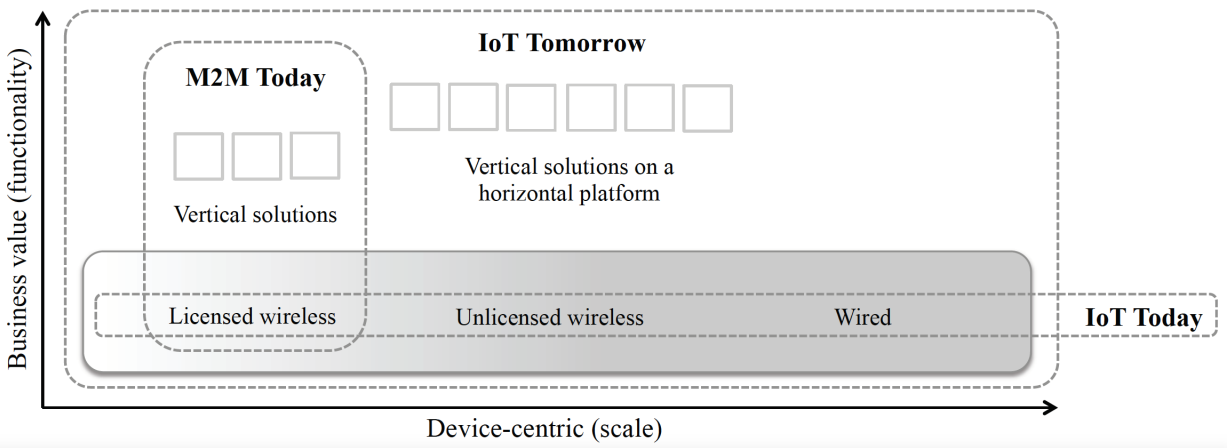

Figura 1.1: Evolução da comunicação M2M para IoT (1).

Mais de 20 anos depois, IoT é a tecnologia mais proeminente para conectar cidades, modernizar fábricas, monitorar pessoas e ativos de empresas. São diversas as verticais e aplicações que atualmente contam com Internet das Coisas como sua tecnologia, dentre elas podemos citar agricultura, medicina, construção civil, empresas de tecnologia, inteligência, logística, dentre outras (44)(42). A inovação no mercado de silício tem barateado a fabricação de chips, em conjunto com o crescimento de serviços de nuvem e conectividade tem facilitado a adoção de IoT em negócios. Diversas pesquisas e análises mostram predições de números sempre crescentes no que tange a adoção de dispositivos IoT, fabricantes e consultorias falam em bilhões de dispositivos IoT para 2020 como pode ser visto na Figura 1.2. Uma estimativa recente publicada pela SoftBank e ARM, aponta a estimativa de produção de dispositivos conectados na casa de 1 trilhão para 2035, conforme visto na (44).

Outra questão bastante relevante é o impacto econômico da adoção de IoT, no qual segmentos como energia, fabril, saúde, automação, software, e outros veem na Internet das Coisas uma nova fonte de receita,

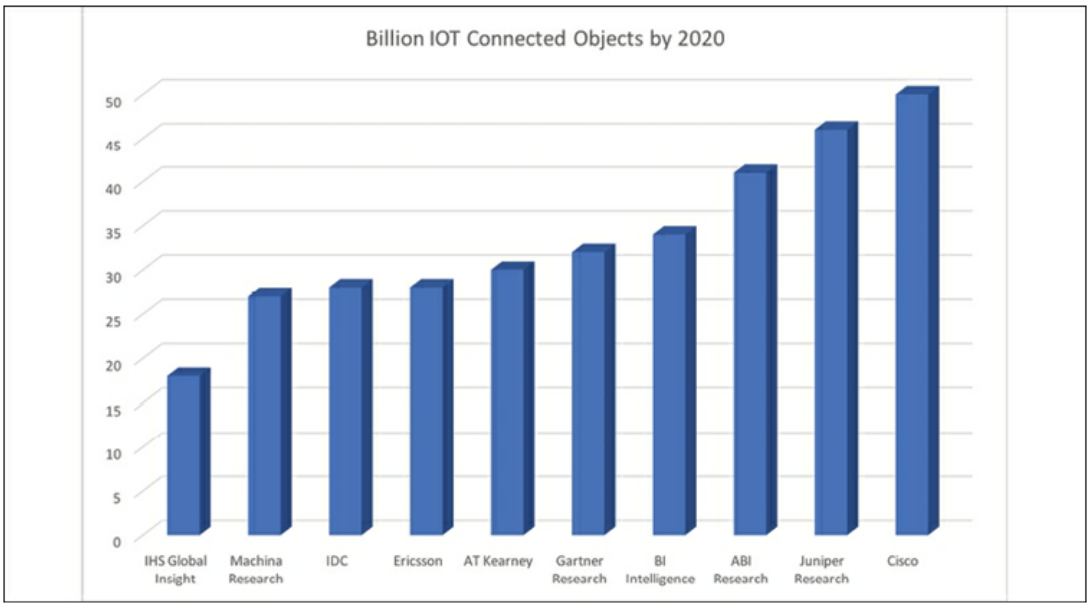

Figura 1.2: Números estimados de dispositivos (2).

seja pelo fornecimento de produtos e serviços, ou pela melhoria em seus processos advindos do uso de dados obtidos por meio da conectividade ou inteligência empregada por meio da Internet das Coisas (45) (46). O crescimento estimado no mercado de Internet das Coisas crescerá de US$1.3 Bilhões de dólares em 2013 para US$7.1 trilhões em 2020, este crescimento favorece não somente empresas, mas consumidores e a sociedade pois os serviços de uma forma geral serão melhorados, as linhas de produção se tornarão mais eficientes e os custos de produtos se tornarão menores (45).

A mesma evolução e as inovações que barateaream as tecnologias e facilitaram a evolução também potencializaram as opções e possibilidades de implantação da Internet das Coisas, trazendo à tona diversos desafios na sua adoção. Dezenas de arquiteturas, combinadas as centenas de possibilidades de combinações de protocolos, microcontroladores e meios de comunicação, acrescidas de formatos de envios de dados para posterior análise, fazem com que projetos sejam de difícil desenho e andamento (46) (45). Um projeto que não tem um desenho prévio visando todas as camadas e sua comunicação pode se tornar problemático na execução graças a escala de dispositivos, tornando o custo alto e o retorno financeiro baixo. Segurança e privacidade podem também se tornar um dificultador, uma vez que cada vez mais os dispositivos de IoT estão se aproximando das casas, leitos de hospital, supermercados, restaurantes e em um futuro próximo, poderão ser implantados em seres humanos (46).

1.1 OBJETIVOS DESTA DISSERTAÇÃO

Este trabalho tem como objetivo apresentar um desenho de arquitetura para Internet das Coisas de início a fim, extensível, com uso de plataforma de nuvem pública, a Azure, a fim de demonstrar as vantagens de tal tipo de plataforma no contexto de IoT.

1.1.1 Objetivos Específicos

- Identificar as arquiteturas de Internet das Coisas que estão sendo utilizadas.
- Demonstrar uma arquitetura de referência para um projeto de telemetria de dados com uso de IoT e nuvem
- Demonstrar uma arquitetura de referência para um projeto de IoT genérico que precise de processamento na ponta (Edge Computing)
- Aplicar a arquitetura em ambiente de nuvem a fim de guiar sua utilização na plataforma de IoT em uma nuvem pública

1.2 METODOLOGIA

Para o trabalho foi utilizada a metodologia apresentada no diagrama da figura 1.3.

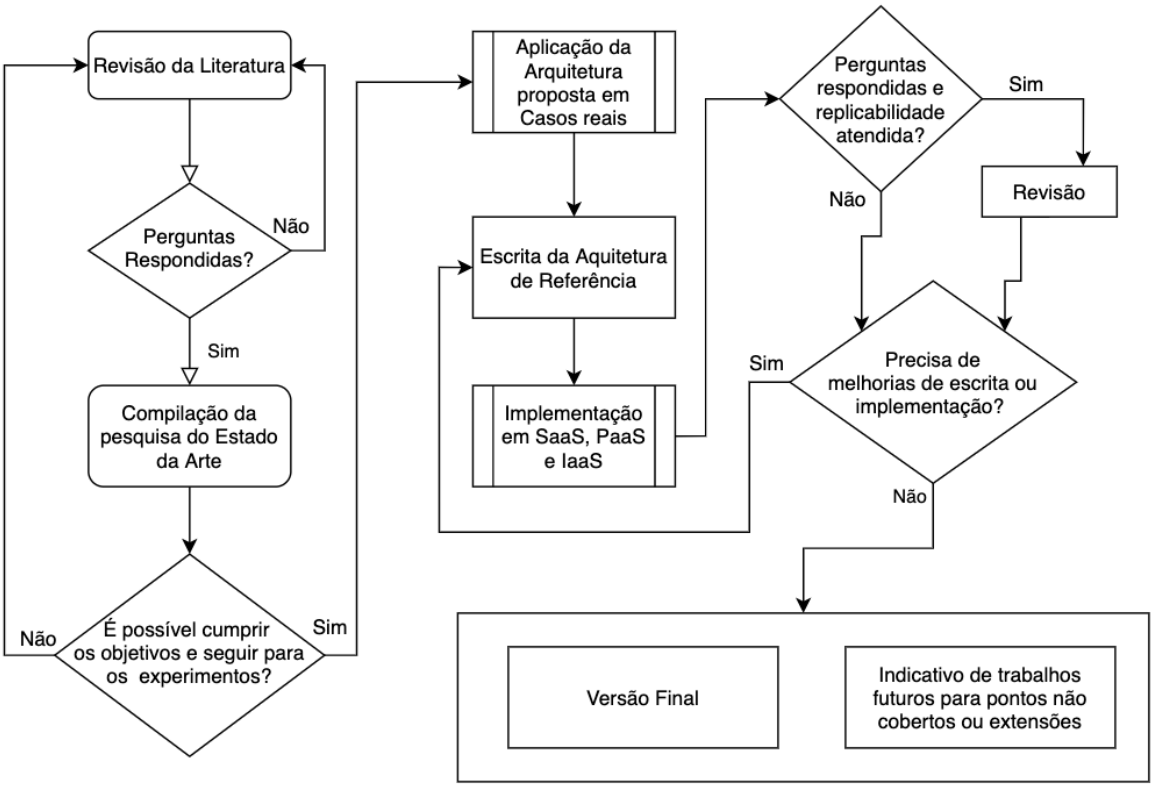

Figura 1.3: Diagrama de passos e etapas utilizadas no trabalho. Compilação do próprio autor.

1.2.1 Pesquisa

A pesquisa foi baseada em uma revisão exploratória da literatura a fim de garantir o estado da arte nos principais protocolos utilizados para projetos de Internet das Coisas, componentes de uma solução,

principais arquiteturas de interligação das diversas componentes e práticas recomendadas na união destas componentes.

Baseado em palavras chaves, e busca nos principais mecanismos e repositórios acadêmicos, os itens com maior cobertura do tema abordado foram selecionados, indiferente do tipo, sejam eles artigos, livros, compêndios e material de divulgação tecnológica de fabricantes. A relevância se baseou na resposta das perguntas e sua colocação em periódicos científicos.

Utilizou-se também de pesquisas sobre IoT feitas com comparativos de artigos para guiar algumas seções do trabalho, por ser um campo muito vasto de protocolos, hardwares, arquiteturas, camadas, estudos e etc.

Preferencialmente utilizou-se os periódicos da IEEE, ACM e Elsevier por terem maior aderência ao tema, não sendo descartados outros. Outro ponto de apoio foi o uso de literatura cinza e de mercado para trazer comparativos dentro das seções de pesquisa e manter um alinhamento do trabalho com o mercado. Em alguns tópicos foram utilizados mecanismos de busca convencionais como Google, Google Scholar, e nos casos que se referiam especificamente ao cenário nacional foram usados termos em português, para todos os outros o inglês foi o idioma de pesquisa, no caso dos fabricantes pesquisados, a linguagem era indiferente, como foi o caso da Microsoft, Google e Amazon, uma vez que os sites são localizados.

1.2.1.1 Palavras Chaves Usadas

Internet of things, iot, iot security survey, iot architectures survey, internet of things survey, physical and mac protocols for iot, Iot Challanges, IoT layers, Azure IoT, ioT Protocols, Internet of Everything, Edge Computing, fog computing, iot Gateway, IoT communication, Plano nacional de IoT, IoT no Brasil, Homologação de Dispositivos,

1.2.1.2 Perguntas de Pesquisa

- Quais as etapas de conexão de um dispositivo de Internet das Coisas
- Quais os protocolos e como é feita a operação com uma visão de arquitetura
- Quais as arquiteturas de Internet das Coisas estão sendo usadas, contemplando aplicações
- Quais os problemas de segurança para Internet das Coisas nas camadas e componentes
- Como desenvolver uma solução de internet das Coisas de início a fim
- O que é preciso para criar uma solução de IoT do zero

1.2.2 Aplicação da Arquitetura com Azure e Cenários Reais

Essa etapa consiste em aplicar serviços aos componentes, visando a exemplificação de uso diante da arquitetura proposta. As redes de comunicação foram adquiridas dos fornecedores para um projeto co-

mercial, sem limitações ou extensões, como é o caso de distribuidores ou acordos de desenvolvimento, garantindo assim uma experiência semelhante a de mercado e garantindo a validade dos testes.

Para todas as simulações, testes e provisionamento de serviços os códigos tanto de criação do ambiente, bem como versões e cargas foram salvos para que se permita a reprodutibilidade futura das simulações feitas, sendo todos incluídos como apêndices neste trabalho.

Nos casos de uso que serão apresentados no Capítulo 4, todos foram desenvolvidos durante a pesquisa dentro da empresa CrazyTechLabs, uma autorização formal de citação dos casos está anexada nesta dissertação no Anexo I. Dados sigilosos de projetos foram omitidos não retirando a validade da demonstração do uso para os fins de validação e enriquecimento das arquiteturas apresentadas neste trabalho.

1.3 A ESCOLHA DA MICROSOFT COMO PLATAFORMA DE IOT PARA AS IMPLEMENTAÇÕES

A escolha se deu por causa da colocação da Microsoft no mercado de Internet das Coisas com os serviços de nuvem, os investimentos anunciados por ela também levam, o mercado tanto acadêmico quando corporativo, a acreditar em suas soluções como sendo uma das principais na hora da escolha de provedor de serviços para IoT. Alia-se a isto a vasta documentação disponível e um número grande de parceiros comerciais e profissionais com experiência nas tecnologias ofertadas pela gigante. Presença global com mais de 50 regiões de nuvem, divididos em todos os continentes, compõem o Azure que tem serviços alinhados as necessidades de implementação global da Internet das Coisas em meios industriais, computação na ponta, telemetria, processamento de dados, Inteligência Artificial aplicada e armazenamento de dados.

Nos comparativos de mercado a Microsoft foi recentemente colocada em excelente posição para IIoT, pelo Gartner (47), que posicionou os serviços e soluções de IoT para indústria no seu quadrante mágico, o qual coloca empresas em destaque no setor analisado. Entre os pontos fortes está sua presença no mercado corporativo dentro de soluções de tecnologia, principalmente em IT e OT.

2 REVISÃO BIBLIOGRÁFICA - COMPONENTES DE UMA SOLUÇÃO DE INTERNET DAS COISAS

2.1 A INTERNET DAS COISAS

Kevin Ashton criou o termo Internet das Coisas em 1999 em uma pequena apresentação para a Procter & Gamble onde ele unia a cadeia de suprimentos à Internet, usando RFID para localizar produtos e mostrar na grande rede (42). Em um conceito simples de unir coisas a Internet ele cunhava o termo que hoje tem diversas definições no meio acadêmico e comercial. O Gartner (48), traz a definição de que a Internet das Coisas é uma rede de objetos físicos que contem tecnologia embarcada para se comunica, sentir ou interagir seus estados internos com o ambiente externo.

Segundo Shahid et al (49) em 2012, dois bilhões de pessoas usavam internet no mundo para navegar, com a Internet das Coisas vem para transformar essa Internet em algo muito mais vital, colocando dispositivos com tecnologia embarcada na vida das pessoas. A mudança de uma Internet antes usada para conectar-se pessoas por meio de computadores ou dispositivos pessoais para algo que agora conectar à objetos físicos que se comunicam em com o outro com ou sem interação humana irá nos fazer repensar o funcionamento de redes, serviços de provisionamento, computação e gestão.

Segundo Gazis et al (50), em 2013, nos anos 2000 o interesse na Internet das Coisas havia crescido, inclusive trazendo a mesma como sendo uma tecnologia disruptiva pelo US National Intelligence Council. Haddud et al (45), em seu trabalho trazem o conceito de que IoT pode ser entendida por uma combinação de três elementos principais: um meio união (*middle-ware*) na Internet, as coisas (sensores, dispositivos e etc) e uma camada de entendimento semântica. Citam também uma segunda definição que trata IoT como nós inteligentes e autoconfiguráveis interconectados em uma rede dinâmica e global, o que corrobora com a colocação de IoT como algo disruptivo por Gazis et al.

Lee (46), em 2015, cita em seu trabalho os termos *Internet of Everything* e *Industrial Internet* como sinônimos para a Internet das Coisas e reafirma o reconhecimento desta ser uma área de muito interesse para as futuras tecnologias e que vem crescendo a interesse no tema. O trabalho de Valencia et al, traz outros nomes como *Internet of Services*, *Internet of People*, *Internet of Agents*, *Internet of Content* como parte dos nomes dados a Internet das Coisas (51) em determinadas aplicações e a certamente existirão outros.

Voltando um pouco no tempo, mais precisamente nas décadas de 70 e 80, falando de dispositivos conectados a Internet antes de ser pensado o termo Internet das Coisas, temos o caso da máquina de refrigerantes do departamento de Ciência da Computação da *Carnegie Mellon University (CMU CS)* que avisava sobre o estoque na internet por meio de um acesso ao endereço IP do serviço (52), o serviço que saiu do ar no início dos anos 80 e foi restaurado posteriormente nos anos 90. Atualmente a universidade tem um estudo sobre o consumo de comida, no mesmo departamento, com uso de uma câmera web e algoritmos (53).

Conforme citado anteriormente no capítulo de introdução e com o caso da *Carnegie Mellon* (52), a evolução da comunicação máquina a máquina e web se deu durante várias décadas e agora o que estamos enxergando é uma evolução dinâmica na adoção da Internet das Coisas como sendo algo novo, relativamente sem maturidade em alguns campos e que requer uma junção de diversas variáveis para o sucesso (1). Um dos pontos importantes nesta evolução está no foco que a Internet das Coisas vem trazendo aos projetos e soluções (3), evoluindo de uma visão de hardware e conectividade para um modelo de trabalho em análise, inteligência e segurança, evoluindo e contemplando mais do que somente conectar os dispositivos e enviar dados, um bom exemplo disso é a figura 2.1.

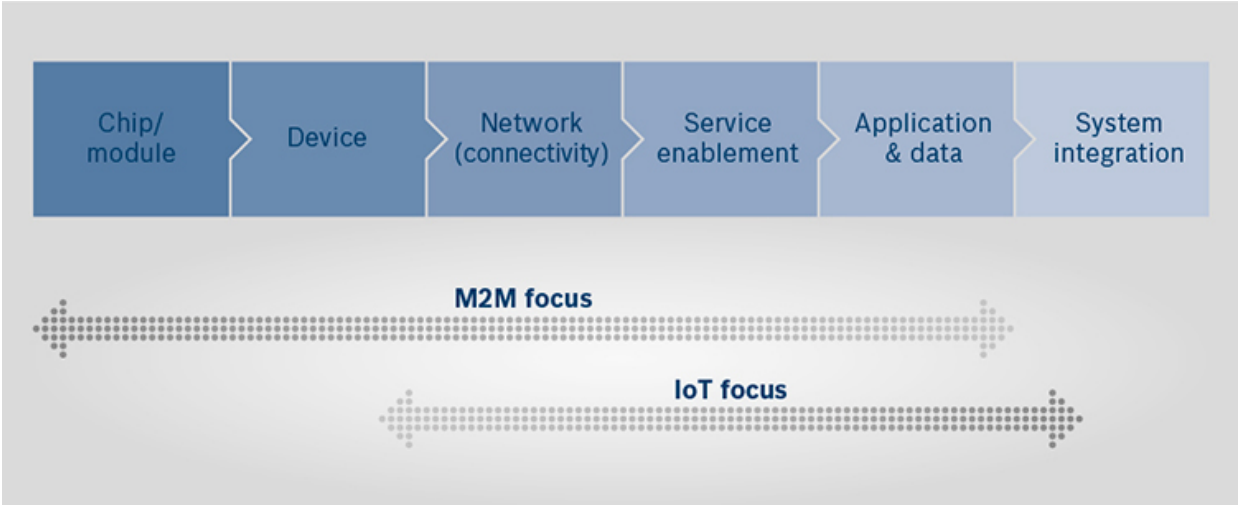

Figura 2.1: Mudança no foco entre M2M e IoT (3).

2.1.1 Números da Internet das Coisas

Segundo levantamento da Statista (4) os números de receita com IoT no Brasil no campo industrial trazem um crescimento de quase 250% entre 2016 e a projeção para 2021 pode ser vista na Figura 2.2.

Shanhong Liu (54) em seu recente estudo (2020), projeta uma visão de 21.5 bilhões de dispositivos conectados para 2025 nas diversas verticais, dentre elas casas inteligentes, veículos, cidades e dispositivos industriais. Ela traz também a projeção de 1.1 trilhão de dólares de gastos em 2023 para projetos de Internet das Coisas e um consumo da casa de 115 bilhões em 2020 para as *smart houses*. Um dado importante para os desenvolvedores e empresas também citado no estudo é que temos 620 plataformas de IoT no mercado, número este atualizado em 2019.

Espinosa et al (55) fizeram um levantamento sobre IoT na Europa e quando falam sobre o crescimento da rede de dispositivos conectados trazem números que variam de 100 bilhões de dispositivos em 2025, e também trazem um contraponto de que já existia uma previsão de números entre 20 e 100 bilhões de dispositivos para 2020. No que tange a adoção de IoT, alguns dos estudos citados mostram um investimento das companhias que estão adotando IoT da casa de 24% de seu orçamento de TI. Na Europa, a Alemanha trazia números de investimento que ultrapassariam os $35 bilhões de dólares seguida pela França e Reino Unido com gastos superiores a $25 bilhões de dólares.

O crescimento da Internet das Coisas pode ser visto também pelo surgimento de alianças de empresas

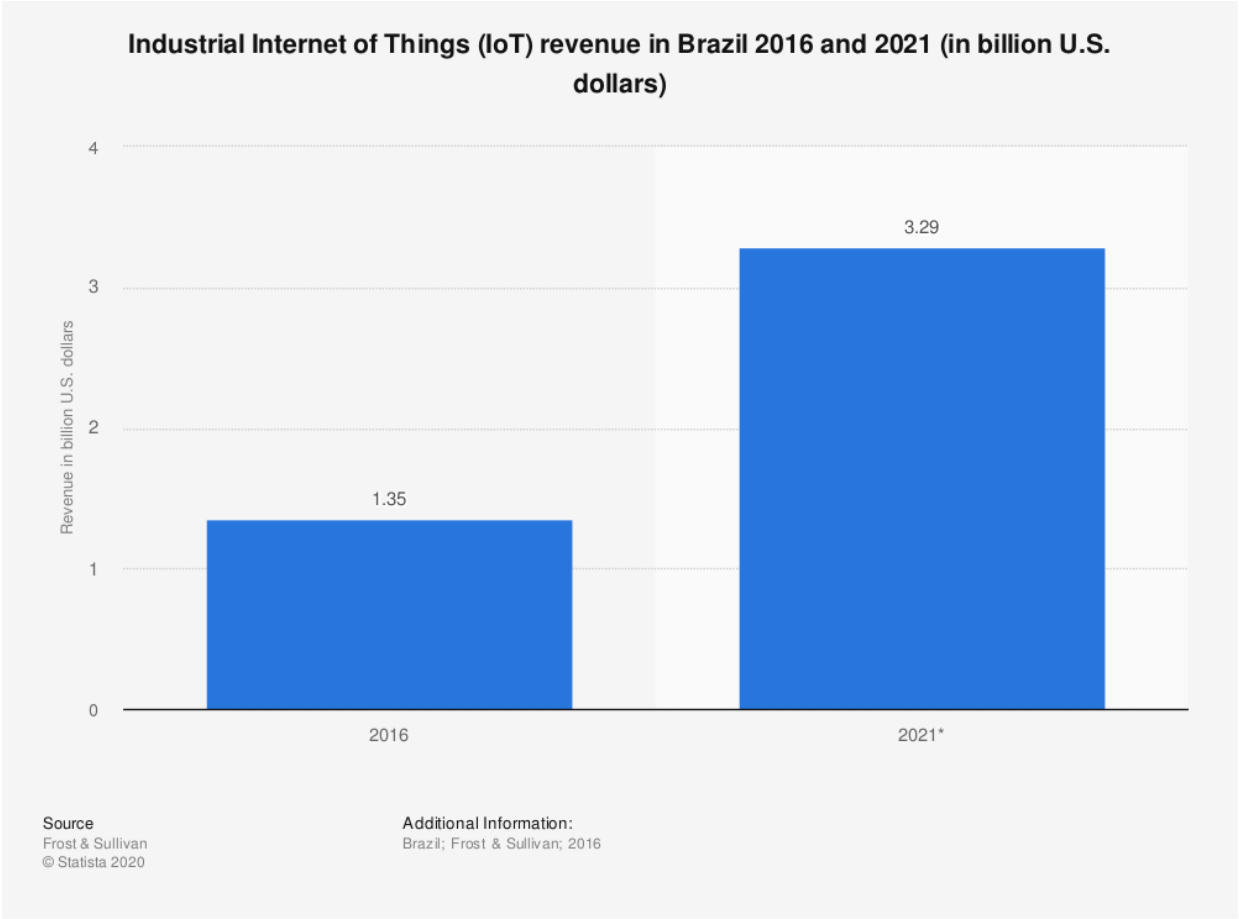

Figura 2.2: Receita de IIoT no Brasil (4).

em prol do desenvolvimento de padrões, algoritmos, tecnologias e outros em torno do tema, conforme pode ser visto na linha do tempo de IoT Alliances (Figura 2.3) compilado por David Excoffier, Ambient Ambient Computing Research Program Director na Orange Labs Meylan na França (5).

O levantamento sobre patentes referentes ao tema IoT feito por Sergio Oliveira et al (7) consolida informações de patentes nas bases do Patentscope do WIPO (World Intellectual Property Organization) e do Espacenet do (European Patent Office), o qual traz a China com um número de patentes bem maior que a soma dos outros países juntos no Patentscope, como pode ser visto na Figura 2.4. O Brasil não aparece com um número substancial de pedidos, porem os autores acreditam que isso deva mudar em breve com o início do plano nacional de IoT. O cenário no Espacenet contempla um número pequeno de patentes entre 2016 e 2018, na casa de 300 patentes com a maioria solicitada por países da Europa. Segundo as pesquisas feitas por eles, o maior número de patentes em ambas as bases estão classificadas sob o código H04L que é referente a transmissão digital de informações.

2.1.2 Importância e Aplicações da Internet das Coisas

Pesquisadores ainda estão analisando o impacto da IoT na economia e consolidando números e projeções no cenário produtivo, mas pode-se dizer que sua importância é grande na injeção de capital em novos serviços e produtos (55), com um aumento na receita de empresas e um gasto para adoção da casa

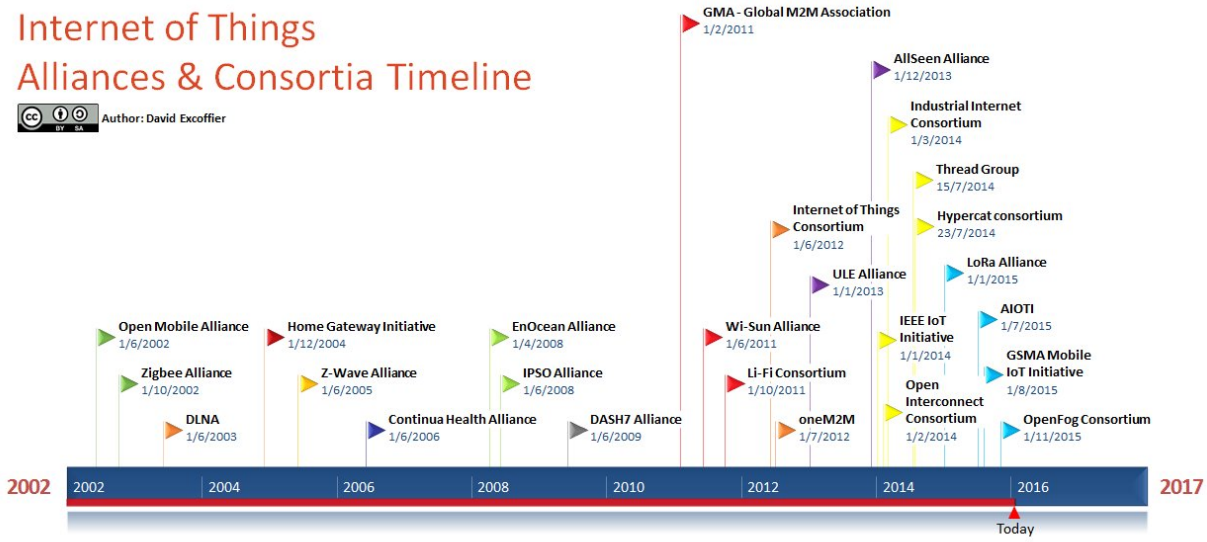

Figura 2.3: IoT Alliances por David Excoffier até 2016 (5).

de trilhões de dólares, conforme já citado anteriormente nos números.

Segundo Fleisch (6), que é especialista na análise econômica da computação pervasiva, várias tecnologias já tentaram reduzir os custos de transações na ponta, a aquisição de dados é uma necessidade e com a Internet das Coisas unindo o mundo físico por meio da computação pervasiva o custo será muitas vezes menor, como visto na figura 2.5 e, a partir daí, as novas aquisições tornam-se mais baratas e assim por diante.

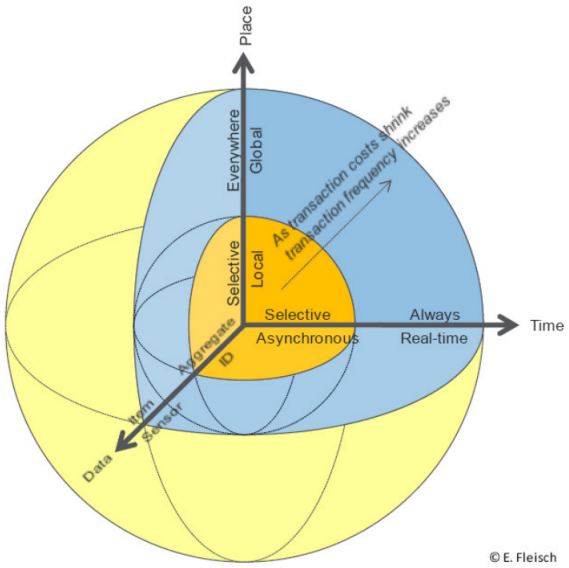

Figura 2.5: Aquisição de dados no mundo real com IoT (6).

No campo da tecnologia, novos processos têm surgido, baterias de longa duração, capacidade de armazenamento de baixo consumo e baixo custo (49), rádios de comunicação de longa distância e com custos cada vez menores (18), tornando a Internet das Coisas mais aderente a cada vez mais negócios, com isso

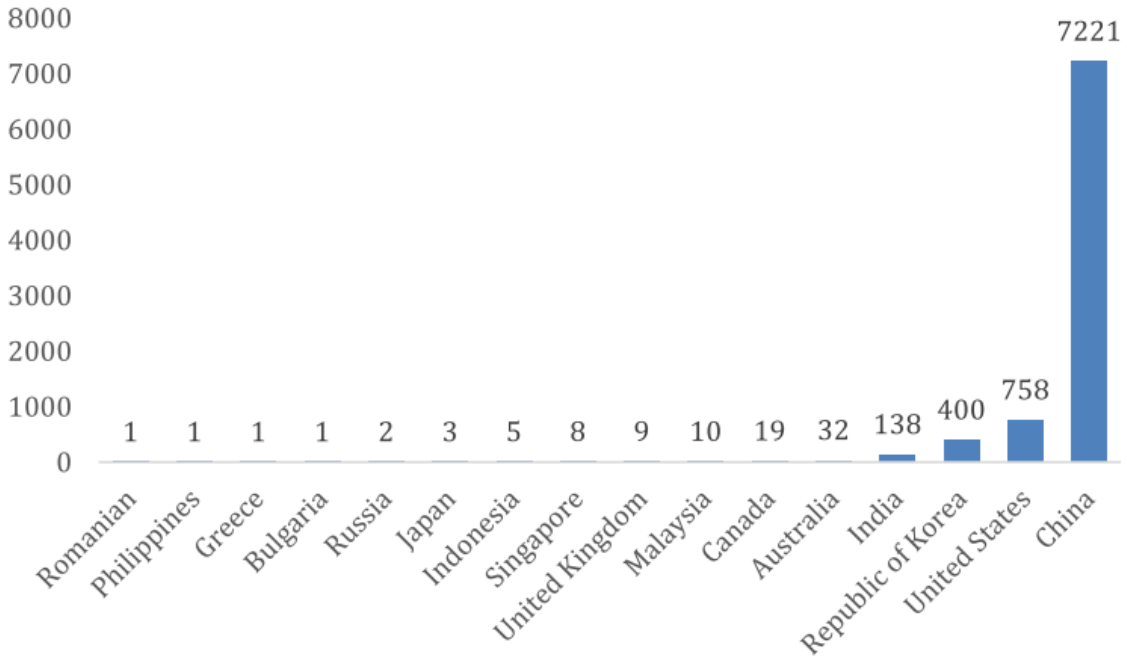

Figura 2.4: Patentes no Patentscope entre 2008 e 2018 divididas por país de origem do pedido (7).

gerando um impacto de diversas formas. A produtividade de pessoas e no trabalho ainda é pequeno haja visto que IoT ainda está em um estágio inicial de adoção e desenvolvimento (55), no campo temos diversos problemas que podem ser resolvidos com sensores e predições de qualidade, neste cenário a Internet das Coisas pode atuar na predição climática aplicada a microclimas (56) reduzindo o trabalho de fazendas e fornecendo ferramentas para o fazendeiro que permitirão uma tomada de decisão e um controle muito maior do campo.

Dispositivos que estão próximos à pessoas são de grande importância na medicina, se contemplarem segurança e confiabilidade adequada ao setor, podem auxiliar o tratamento de pacientes (57) em momentos onde o atendimento humano pode demorar, sendo este, além de um fator de bem-estar ao paciente, também um habilitador de um cuidado realmente mais avançado, tanto dentro quanto fora dos ambientes de saúde.

Aplicável a quase todas as verticais, o campo de medições chama bastante atenção e pode além de reduzir custos melhorar serviços, como é o caso de medições de água, gás e energia (58). Uma medida errada além de gerar descontentamento por parte do usuário do serviço, gera um custo de operação. Protocolos e dispositivos adequados podem fazer a tomada da medida com facilidade e baixo custo (59).

Um momento que estamos vivendo de redução de uso de recursos naturais, reutilização de recursos, prédios com iniciativa verde, cidades mais eficientes, menos desperdício, são fatores tanto favoráveis no crescimento da Internet das Coisas quando são iniciativas habilitadas pelo uso de IoT como facilitador e as vezes como agente gerador de dados para melhor decisão acerca dos fatos imediatos (1).

IoT depende de uma gama de novas tecnologias e desenvolvimentos e este é um fator de muita importância tanto para a comunidade científica quanto para os consumidores de tecnologias e empresas de serviços (60) com este aquecimento gerando um ciclo virtuoso no cenário de Internet das Coisas, temos o

crescimento de novas tecnologias, a geração de novos provedores de serviços, novos meios de comunicação, dentre outros, e aí inicia-se um momento de competição aplicando uma força intensa no mercado e com isso derrubando preços de tecnologias, ampliando o acesso a informação e novamente gerando uma demanda, por mais tecnologias e mais serviços, extremamente benéfica ao mercado como um todo.

Na perspectiva de coisas inteligentes estamos vivendo um cenário de aplicação de cognição e inteligência a dispositivos, uma forma de ter algo mais do que simplesmente conectar dispositivos (60), novos termos estão sendo cunhados para esta nova geração tais como *Brain-Empowered IoT*, *Cognitive Internet of Things*, *Intelligent Edge* (61), *Intelligent Cloud* dentre outros que trazem na sua essência a aplicação direta de inteligência artificial aos dispositivos, seja embarcando neles algoritmos, seja tratando os dados obtidos posteriormente na nuvem. Com isso temos novamente algo que alimenta um ciclo de novas necessidades de tecnologias, que por sua vez barateiam as antigas e popularizam seu uso para dar espaço a inovação. A figura 2.6 retrata uma ideia do ciclo virtuoso e contempla o momento atual do nascimento de novas tecnologias e a necessidade de novas tecnologias a fim de prover insumo a novas tecnologias (62) e assim por diante, uma realidade no cenário de Internet das Coisas.

Figura 2.6: Ciclo virtuoso do surgimento e necessidade de novas tecnologias. Adaptada de (8).

2.1.3 Brasil e o Plano Nacional de IoT

Em um estudo sobre a Internet das Coisas e o plano nacional de IoT para o Brasil, Isabela Sabo et al (63), chegaram à conclusão de que a Internet das Coisas é um campo que começou a ser explorado na segunda década dos anos 2000 e que o lançamento do plano de IoT vem para somar ao mercado nacional com inúmeras vantagens a população, desde ambientes mais inteligentes, serviços mais ágeis, até um sistema de saúde mais integrado com as pessoas. Eles enxergam também que a normatização de alguns pontos propostos pela legislação irá aperfeiçoar seu uso pela sociedade e abrir mais pontos de interação.

Maurício Angelo em seu artigo (64) trouxe uma visão da onda de IoT no mercado brasileiro e deixou uma questão interessante sobre o real funcionamento do plano nacional de IoT impulsionar a economia brasileira, a questão pertinente é que temos ações isoladas em funcionamento no Brasil a tempo, como ele mesmo cita o CESAR de Recife. Outro ponto trazido por ele é que a junção do Banco Nacional de Desenvolvimento Econômico e Social (BNDES) com o Ministério da Ciência, Tecnologia, Inovações e

Comunicações (MCTIC) lançando o plano nacional de IoT foi uma excelente iniciativa, que contou com consultas públicas a 200 especialistas, 190 organizações.

A versão final do relatório intitulado ”Relatório do plano de ação – Iniciativas e Projetos Mobilizadores” que é um dos produtos do estudo "Internet das Coisas: um plano de ação para o Brasil", contempla uma visão executiva para ação (9) (65). O plano destacou quatro frentes prioritárias para o país, são elas: Cidades, Fábricas, Saúde e Rural, conforme pode ser visto na figura 2.7.

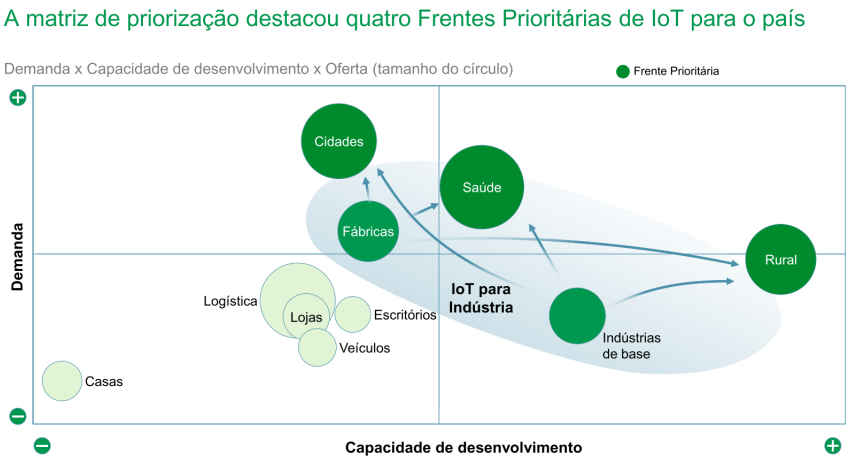

Figura 2.7: Frentes prioritárias do plano de IoT Brasil (9).

Para cada uma das frentes / ambientes foram detalhados os objetivos estratégicos conforme podemos ver abaixo (9):

Cidades

- Mobilidade: Reduzir tempos de deslocamento, considerando diferentes modalidades de veículos, e aumentar a atratividade do transporte coletivo.
- Segurança Pública: Aumentar capacidade de vigilância e monitoramento de áreas da cidade para mitigar situações de risco à segurança.
- Energia e Saneamento: Reduzir desperdício de *utilities* e criar rede de iluminação pública que habilite soluções de IoT de forma ampla na cidade.

Saúde

- Doenças Crônicas Melhorar: Melhorar a efetividade dos tratamentos de pessoas com doenças crônicas por meio do monitoramento contínuo de pacientes.
- Promoção e Prevenção: Prevenir situações de risco e controlar o surgimento de epidemias e de doenças infectocontagiosas por meio de soluções de IoT.
- Eficiência de Gestão: Aumentar a eficiência dos hospitais do SUS e unidades de atenção primária de saúde através da adoção de soluções IoT.

Rural

- Uso Eficiente de Recursos Naturais e Insumos: Aumentar a produtividade e qualidade da produção rural brasileira pelo uso de dados.
- Uso Eficiente de Maquinário: Otimizar o uso de equipamentos no ambiente rural pelo uso de IoT.
- Segurança Sanitária: Aumentar o volume de informações e sua precisão no monitoramento de ativos biológicos.

Indústrias

- Recursos e Processos: Aumentar a eficiência e a flexibilidade dos processos industriais usando soluções de IoT para a gestão de operações.
- Bens de Capital: Promover o desenvolvimento de novos equipamentos, produtos e modelos de negócios que incorporem soluções de IoT.
- Estoque e Cadeia de Suprimento: Promover a integração e cooperação nas cadeias de fornecedores de bens, componentes, serviços e insumos.

Inovação

- Promover a adoção de soluções desenvolvidas localmente para desafios do ambiente.

Segundo Maurício Angelo em seu artigo (64) tendo uma visão pessimista teremos até 2025 uma diminuição de 15% o tempo gasto com trânsito e em 20% os índices de criminalidade, na saúde poderemos diminuir o custo com manutenção de equipamentos na casa de 40%, auxiliar na previsão de pandemias, na área rural ele cita uma possível redução da casa de 20% no uso de insumos agrícolas e um aumento de 25% na produção das fazendas.

2.1.4 Desafios, Preocupações e Consequentemente Oportunidades

Internet das Coisas pode ser vista como uma solução para problemas existentes que envolve a união de hardware, software, processos, pessoas e aplicações existentes o que torna as coisas ou dispositivos em alguns casos os mecanismos de integração. O Processo de desenvolvimento é, portanto, muito complexo e como trata-se de um campo novo, os casos têm sido guiados a resolver um único problema para um ramo de atuação, o que traz a ideia de que as complexidades e desafios irão começar a surgir quando houver a necessidade de integração de tipos de negócios ou evolução dos projetos de primeira fase (28).

Provisionamento de dispositivos pensando em computação pervasiva é cada vez mais transparente ao usuário, vão envolver conceitos de nuvem combinados a Internet das Coisas, com isso, os desafios também são combinados. Com a Internet das Coisas temos a visão de conectar tudo e todos no mundo todo, indiferente da sua localização, em nuvem temos como premissa o fornecimento como serviço, escalando na necessidade do negócio e provendo custos adequados por uso. Sem falarmos dos desafios de segurança nesta interconexão do mundo real a nuvem, pode-se citar como um problema a ser resolvido a conectividade, latência, regulamentações regionais, gestão e orquestração destas conexões, controle de uso de

serviços, personalização sob fronteiras de línguas e culturas e principalmente a garantia de estabilidade e confiabilidade de uma solução de IoT (66). Em uma visão de campo sem necessariamente uma conexão direta a nuvem, mais distribuída e com dispositivos conversando entre si temos desafios que se unem aos anteriores e agora acrescentam segurança nos diversos nós de entrada e saída de dados desta rede, controles maiores na identidade dos dispositivos, e o que antes era pensado na nuvem sobre disponibilidade, monitoramento e gestão de serviços, agora se estende para a ponta com mais dispositivos, mais meios de comunicação e mais pontos de falhas (28).

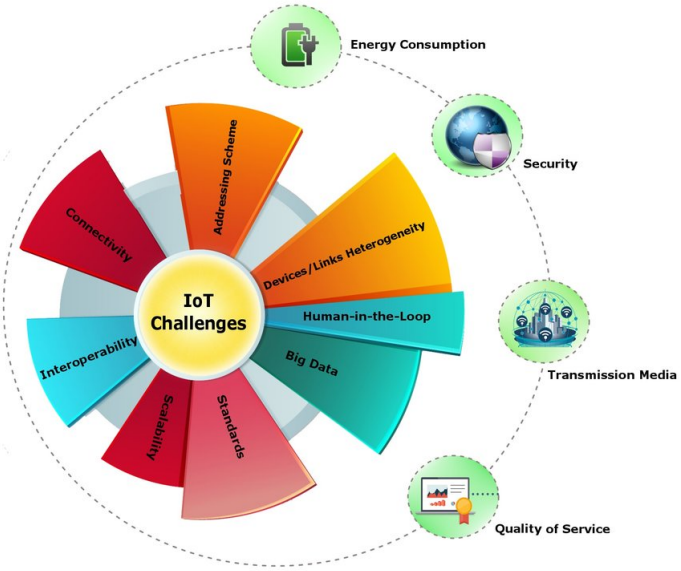

Figura 2.8: Alguns dos desafios e oportunidades da Internet das Coisas (10).

Um sistema completo de IoT pode unir diversos desafios combinados e com multidisciplinaridade dentro dos campos da tecnologia, abrangendo desde gestão, segurança (28), comunicações, desenvolvimento de software para aplicações e nuvem, desenvolvimento de sistemas embarcados, infraestrutura de redes de comunicação desde a locais a globais, armazenamento e análise de dados, dentre outros (67), mas indiscutivelmente um dos maiores desafios estudados presentes nas pesquisas é a segurança. Somente em 2017 houve um aumento de mais de 600%, onde na maioria das vezes o invasor não tinha como objetivo o dispositivo em si, mas sim o ambiente onde ele se inseria, isto aumentou as estatísticas deste tipo de crime e o crime cibernético ascendeu a segunda posição como o crime mais reportado globalmente (68). Sistemas onde temos sensores e nuvem, *Sensor-Cloud System* (SCS), ainda tem problemas de segurança e privacidade a serem resolvidos, embora usem os mecanismos conhecidos de criptografar os dados, muito ainda tem que ser feito já que agora falamos de roubo de identidades e outros tipos de ataque, o que com o hardware mais distribuído e cada vez mais difundido segundo Wang et al (69). Ainda segundo Wang, o uso de uma camada no meio pode ajudar neste desafio de segurança estabelecendo uma relação mais confiável entre as pontas, embora aumente a complexidade de gestão e governança.

Internet das Coisas só existe se e somente se um dispositivo existir, seja ele um software ou um hardware, quando temos o físico, mais um desafio é quanto ao desenvolvimento do circuito de forma que ele não tenha problemas de funcionamento, mas principalmente que ele cumpra duas das inúmeras características de um hardware de IoT: baixo consumo e ser seguro. Ser seguro não somente no software e na

comunicação, mas fisicamente também, garantindo o comportamento adequado e a proteção dos dados em condições adversas no campo (70).

O aumento da produção de dispositivos e insumos tecnológicos para suportar o crescimento da Internet das Coisas, tem gerado uma demanda por metais preciosos como insumos dos eletrônicos, também tem se usado tecnologias fósseis para geração de energia nas linhas de produção, além de que os dispositivos gerados anteriormente, atualmente e no futuro próximo vão gerar um aumento no lixo tecnológico que atualmente só tem aproximadamente vinte por cento do total reciclado (71). Os impactos das tecnologias de IoT no meio ambiente ainda são desconhecidos.

Um ponto também discutido é o impacto da Internet das Coisas na produtividade como forma de diminuir postos de trabalho, e neste tópico economistas e especialistas em tecnologia tem visto mudanças significativas no momento que em as novas tecnologias entram em campo. O temor de que milhões de pessoas percam seus postos de trabalho são contrapostos pela possibilidade de mudança na posição de trabalho com o próprio uso da tecnologia (72). Ainda na Pesquisa de Winburn et al. (72) ele traz a título de comparação, o estudo feito pela Information Technology & Innovation Foundation apresentando que mais de cinquenta por cento dos empregos que os trabalhadores faziam há cinquenta anos não existem mais hoje, estamos vivendo uma era de transformação, acelerada pela Internet das Coisas entrando em todas as verticais e muitos pontos de discussão ainda cabem sobre a teórica substituição do trabalho humano por máquinas, sejam elas robóticas ou simplesmente algoritmos. Ainda sobre o ponto da produtividade e da sua necessidade na sociedade moderna, Sandro Nizetic et al. (71) trazem a necessidade de urbanização e melhoria dos serviços nas cidades por causa do aumento populacional, isto irá gerar uma demanda cada vez maior por tecnologias e este é um ponto que pode ser enxergado como uma oportunidade, bem como um desafio de adoção e crescimento ordenado da implantação de tecnologias.

Internet das Coisas está entrando em todos os ramos de negócios e com isso cabe a integração com sistemas antigos, fechados ou não, com ou sem conectividade direta para nuvem. Estes dispositivos e sistemas são comumente chamados *Brownfield*, um exemplo pode ser visto na figura 2.9 onde uma máquina antiga sofreu a adição de sensores para coleta de dados e demonstração na web (11), o mesmo pode acontecer em fazendas, campos de extração de petróleo, e outras verticais. A interligação destes dispositivos / sistemas é uma necessidade em aplicações industriais, haja visto que o pátio de equipamentos é antigo e está em pleno funcionamento, cabendo às vezes integrar ao invés de simplesmente substituir (73) (74).

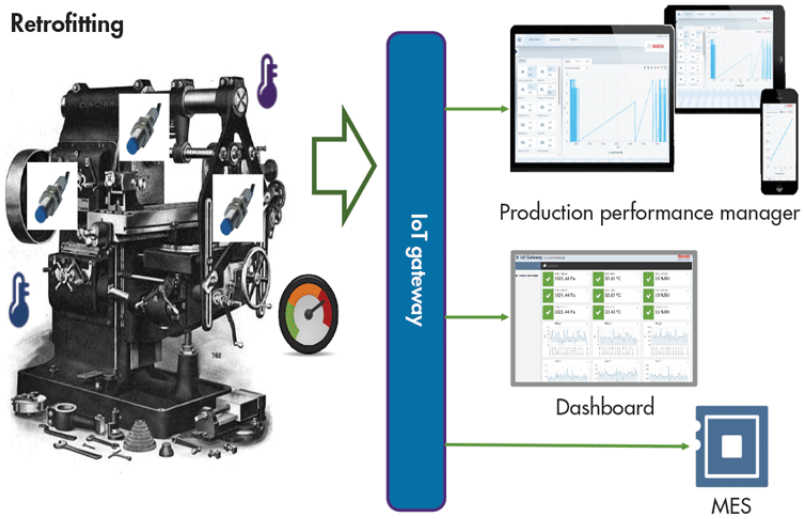

Figura 2.9: Renovação de um ambiente industrial sem troca de maquinário, somente adição de sensores e conectividade (11).

Em resumo temos como desafios a segurança, integração e a complexidade das tecnologias, na figura 2.8 podemos ver alguns dos pontos citados nesta seção, mas cabe ressaltar que todas as dificuldades citadas precisarão de mão de obra qualificada e por isso incluímos a multidisciplinaridade dos profissionais como um dos desafios para o desenvolvimento da IoT (10).

2.2 O QUE COMPÕE UMA SOLUÇÃO DE INTERNET DAS COISAS

Segundo Chan et al. (28), uma solução de Internet das Coisas deve ter uma percepção compreensível do mundo físico, uma transmissão confiável e um processamento inteligente, os quais podem se dividir em três camadas bem distintas que são respectivamente análogas as percepções anteriores uma camada de sensores, uma camada de rede e serviços e uma camada de aplicação.

Ray (10) traz uma visão parecida a de Chan, embora mais detalhada, e coloca como os blocos funcionais da Internet das Coisas os seguintes itens e suas funções:

- Dispositivo: Responsável pela base de uma solução de Internet das Coisas, ele é responsável pela intermediação do meio físico por meio de sensores e atuadores com as camadas superiores de comunicação e aplicação. Nele ficam alguns componentes que atuam no funcionamento, armazenamento e comunicação tanto interna entre sensores como externa.
- Comunicação: Este bloco faz a parte de união entre o dispositivo e os serviços em servidores remotos, neste bloco também estão os protocolos de comunicação.
- Serviços: Neste estão os serviços de IoT que são responsáveis por modelagem dos dispositivos, controle dos dispositivos, publicação/recepção de dados, análises e descobertas da rede.
- Gestão: Este grande bloco atua na governança da solução.

- Segurança: Gerenciamento de segurança, autenticação, autorização, integridade da rede de dispositivos e dos dados.
- Aplicação: Esta é a parte de integração com os usuários e sistemas, aqui entram as interfaces de dados, visualizações, telas de controle e APIs de integração.

Outros autores (73) (71) (75) citam blocos semelhantes aos apresentados por Ray (10) e Chan et al. (28) para uma solução de Internet das Coisas o que nos leva a enxergar uma visão geral de componentes conforme a visão da figura 2.10.

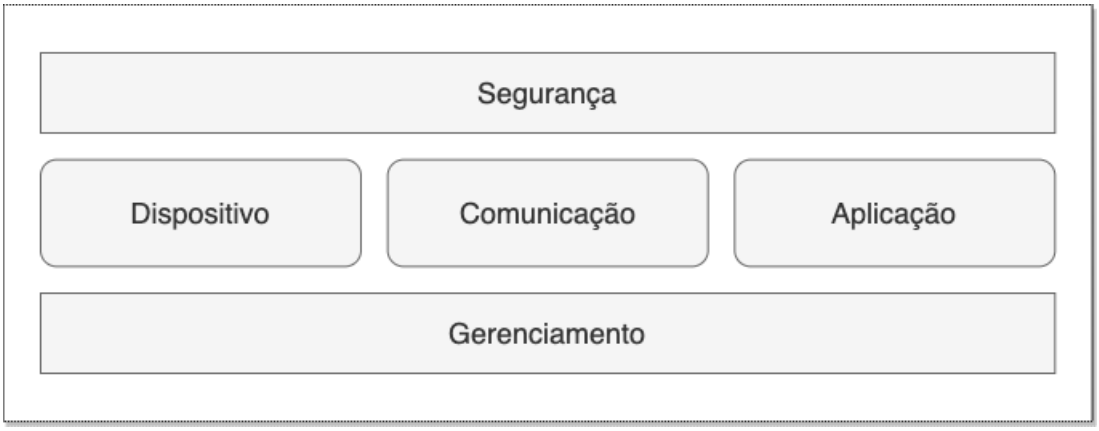

Figura 2.10: Visão simplificada de componentes de uma solução de Internet das Coisas, criada a partir de (10).

2.3 O DISPOSITIVO

Uma visão não completa, mas bem abrangente de um dispositivo, quando olhamos o interior da Coisa na IoT pode ser vista na figura 2.11 (10). O dispositivo é o coração de uma solução de Internet das Coisas e é composto por um *Hardware*, e necessariamente algum tipo de meio de conectividade com o meio exterior. A importância de um bom desenho de implementação do dispositivo pode ser fator de sucesso no projeto (76). Atualmente os dispositivos podem ser desde simples módulos de hardware com microcontroladores até um *Single Board Computing* (SBC) que tem um grande poder de processamento podendo embarcar um sistema operacional e processar algoritmos de Inteligência Artificial com auxílio de GPUs (77) (78).

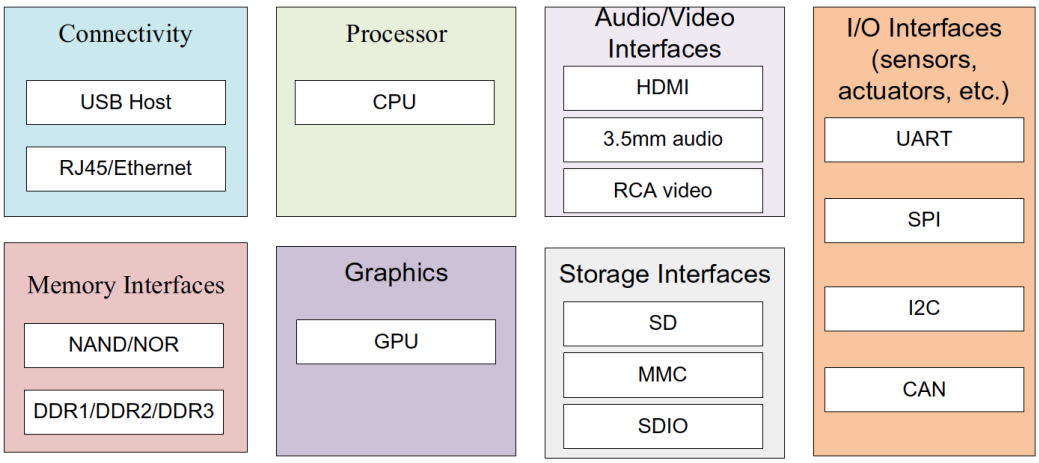

Figura 2.11: Alguns dos componentes de um dispositivo (10).

Um software deve ser embarcado no dispositivo podendo ser simplesmente um código de execução cíclica onde faz a varredura de portas internas a fim de ler seus sensores, até mesmo pode-se embarcar um pequeno sistema operacional com blocos de código mais complexos e um controle melhor do hardware sem que se tenha a necessidade de escrever cada instrução de acesso, dentre os sistemas operacionais podemos citar desde um pequeno *real-time OS* (RTOS) até uma distribuição Linux ou até mesmo uma distribuição própria da Microsoft para IoT (79) (80) (77). Com isso pode-se ter em execução desde um pequeno código que faz a leitura de sensores e envia para uma nuvem, até uma base de dados rodando localmente como é o caso do RavenDB, uma base extremamente poderosa com modos que vão de NOSQL a Séries Temporais, a qual em atualização recente passou também a permitir execução em qualquer Sistemas Operacionais Linux ARM64 com pouca memória e baixo uso de armazenamento, o que permite a execução em dispositivos de Internet das Coisas (81). Internet das Coisas é escala e para escalar o valor por dispositivo deve ser pequeno, quanto maior forem os recursos maior será o custo unitário e por consequência maior a complexidade de operação, maior o consumo de energia e assim por diante, o que pode inviabilizar o projeto (75).

2.3.1 Dispositivos de Edge Computing

Nos últimos anos vem crescendo a pesquisa e os investimentos em *Edge Computing*, uma tecnologia que aumenta os poderes do dispositivo de Internet das Coisas para um patamar de extensão dos serviços de rede. *Edge Computing* é um novo paradigma que pode ser tratado como extensões dos serviços de nuvens ou nós de *Fog computing networks*, por estarem na ponta, estão próximos dos sensores e atuadores e permitem uma ação próxima de tempo real, com menos comunicação essencialmente e com isso tornam os dispositivos mais robustos. Em alguns casos são facilmente colocados em local de um *gateway* de rede ou como um concentrador de outros dispositivos. Estes serviços podem ser orquestrados diretamente da nuvem por meio de contêineres de aplicações, com gestão totalmente remota, dando o poder de computação pervasiva a estes dispositivos (82) (83) (84) (85). A origem do termo *Edge Computing* vem da década de 90 quando a Akamai introduziu as *Content Delivery Networks* (CDNs) para melhorar a performance de download de conteúdo web colocando servidores nas bordas perto dos clientes e movimentando dados entre elas de acordo com a necessidade do público local.

Aplicações que dependem de funcionamento contínuo e de baixo tempo de resposta ou seja, alta confiabilidade de execução e grande dependência do serviço de IoT, são fortes candidatas ao uso de *Edge Computing* (82) (83). O sentido de embarcar aplicações mais robustas na ponta se faz mais presente quando um provedor de soluções como Microsoft, Amazon ou Google fornecem serviços de nuvem que rodam neste dispositivo de campo e podem ser controlados a distância com o um alto nível de confiabilidade e um custo aderente a este novo modelo de computação extremamente distribuída que é a Internet das Coisas. As vantagens em usar serviços prontos são grandes, uma vez que o custo de desenvolvimento e o tempo de início de operação com uso deles é pequeno frente a escrever tudo do zero, cada fornecedor tem seus conjuntos de soluções e precificação (76).

2.3.2 Fog Computing

Um ponto importante é se desambiguar os conceitos de *Fog Computing* e *Edge Computing*, a Cisco foi a pioneira em propor o termo *Fog Computing* como uma solução para ponta, que seria uma extensão da nuvem em uma rede de computação pervasiva com processamento de atividades sem dependência de terceiros, importante a definição que Fog está entre a nuvem e os dispositivos e a rede da ponta (69). *Edge Computing* é a forma de elevar os poderes de um dispositivo de ponta para promover maior inteligência e poder de processamento na ponta, na maioria das vezes o termo é trazido como uma extensão de serviços de nuvem com execução independente e desconectada, por vezes, muito falado por gigantes como Amazon, Google e Microsoft (82) (86) (76) (85) (84). Em resumo podemos de acordo com os autores levar a Fog Computing para a camada meio, borda ou até mesmo na ponta e Edge Computing para dentro da camada dos dispositivos, portanto podendo ambas serem um dispositivo ou um *gateway*, embora ainda tenhamos no mercado os conceitos se coincidindo, o que nos leva a importância de entender que existe cada vez mais a necessidade de levar a inteligência das nuvens para ponta e em alguns casos processar localmente indiferente do nome que venha a ser dado, pois ele depende às vezes de acordos comerciais ou de marketing de companhias.

2.4 COMUNICAÇÃO, PROTOCOLOS E CAMADA DE TRANSPORTE DE DADOS

Comunicação dentro de Internet das Coisas vem de encontro a dois desafios, *low-ower* para que tenha-se um baixo consumo, *low-cost* baixo custo, além disso integração com padrões de mercado, necessidade de confiabilidade e segurança (49). Outro ponto muito importante é que a comunicação tem papel chave na Internet das Coisas, sem dispositivos não temos IoT, mas sem comunicação ela também não existe, e neste cenário estamos vendo algo novo no mercado de telecomunicações que é o surgimento de serviços com pagamento por uso (87). Neste cenário onde precisamos enviar e receber mensagens com dados e comandos, cada solução tem sua própria necessidade, e por isso em alguns casos podemos ver a combinação de diversos protocolos de comunicação e várias tecnologias também, desde protocolos de comunicação até protocolos de aplicação, separados, ou juntos e em alguns casos com aplicações diferentes de acordo com o local de instalação do dispositivo. Para um bom arquiteto de soluções IoT é importante o conhecimento do que existe e como eles podem ser aplicados e combinados garantindo uma solução que consiga alcançar

Tabela 2.1: Camadas do Modelo OSI. Adaptada de (34).

OSI Layer	Major Function
Application (Layer 7)	Provides access to the network for applications.
Presentation (Layer 6)	Translates data from the format used by applications into one that can be transmitted across the network. Handles encryption and decryption of data. Provides compression and decompression functionality. Formats data from the applications layer into a format that can be sent over the network.
Session (Layer 5)	Synchronizes the data exchange between applications on separate devices.
Transport (Layer 4)	Provides connection services between the sending and receiving devices and ensures reliable data delivery. Manages flow control through buffering or windowing. Provides segmentation, error checking, and service identification.
Network (Layer 3)	Handles the discovery of destination systems and addressing. Provides the mechanism by which data can be passed and routed from one network system to another.
Data link (Layer 2)	Provides error detection and correction. Uses two distinct sub-layers: the Media Access Control (MAC) and Logical Link Control (LLC) layers. Identifies the method by which media are accessed. Defmes hardware addressing through the MAC sub-layer.
Physical (Layer 1)	Defines the physical structure of the network and the topology.

os resultados dentro do custo necessário e com os requisitos no negócio atendidos (2) (34).

Segundo Alhamedi et al. (34) os modelos atuais como OSI e TCP/IP estão ultrapassados e precisamos urgentemente de um novo modelo que contemple esta nova necessidade criada pela Internet das Coisas, fornecendo a comunicação entre as coisas, entre um mundo físico e um mundo virtual indiferente da natureza do objeto da comunicação.

2.4.1 Modelo OSI

O modelo de comunicações é o *Open Systems Interconect (OSI)*, foi um padrão criado no final da década de 70 e revisado no início dos anos 80, ele surgiu em uma época onde as máquinas somente falavam com máquinas do mesmo fabricante e a necessidade de comunicação com outros equipamentos emergia como uma necessidade. O padrão criado pela *International Organization for Standartization (ISO)* em 1978 determina 7 camadas que podem ser vistas na tabela 2.1, ainda hoje muitos protocolos e arquiteturas de comunicação de soluções se baseiam no modelo OSI para explicitar seu funcionamento conforme pode ser visto na figura 2.12 (34).

2.4.2 TCP/IP

O Modelo TCP/IP surgiu no final da década de 60 e se consolidou no início da década de 70 e até hoje é o motor da nossa conhecida internet que no seu início era um projeto do Departamento de Defesa dos Estados Unidos (DoD) chamado de *Advanced Research Projects Agency NETwork* (ARPANET), uma rede que tinha como foco a comunicação sobreviver enquanto o originador e o recebedor estivessem no ar. O TCP/IP faz o mesmo trabalho do modelo OSI, porém em 4 camadas, a figura 2.13 mostra uma relação entre os modelos (34).

Segundo Belo et al. (87), um dos pontos importantes na comunicação de dispositivos de IoT é a possibilidade de comunicação entre dispositivos na ponta e eles apontam alguns problemas no uso de

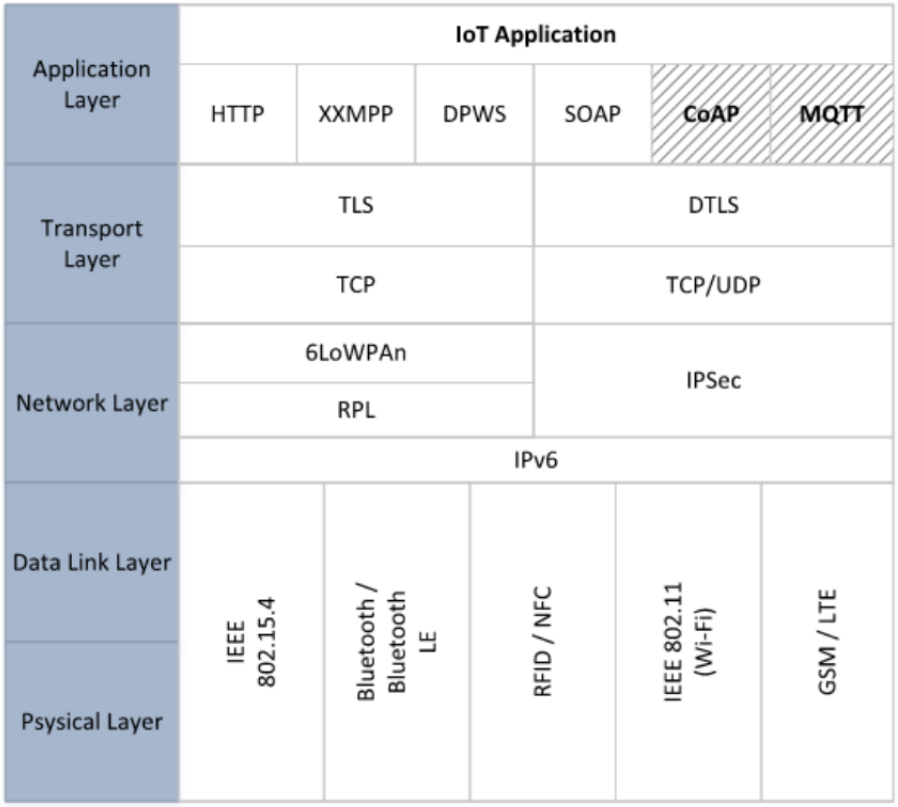

Figura 2.12: Protocolos de comunicação e aplicação aplicados as camadas do modelo OSI (12)

TCP/IP para este cenário.

- Pilha de Execução do Protocolo e Peso das Comunicações: O TCP/IP requer uma grande largura de banda, poder de processamento, memória e bateria para tudo isso. Ele requer recursos como *sockets* e *buffers* para conseguir resolver toda sua execução, isto torna muito pesado para dispositivos com maior restrição de hardware, memória e processamento.
- Fragmentação e Remontagem do Pacote: A arquitetura do TCP/IP permite o envio de uma diretiva de fragmentação de grandes blocos de dados em pequenos trechos para uma remontagem no recebimento. A fragmentação se dá quando tenta enviar uma mensagem maior que o *Maximum Transmission Unit* (MTU) da rede. Esta informação cria uma sobrecarga e a remontagem requer muito poder de processamento para sua execução.
- Esquema de Endereçamento: A pilha de execução TCP/IP adiciona metadados como cabeçalhos, campos extras e outros em cada camada, com isso temos processamento extra e maior consumo de memória e consequentemente sobrecarga do dispositivo.
- Segurança Incorporada: O TCP/IP não considerou segurança do desenho inicial, e por isso segurança é tratada dentro de cada camada, gerando uma sobrecarga de processamento.
- Retransmissão e Reconhecimento de Pacotes: Para uso em TCP/IP precisa-se readaptar as camadas superiores, com isso precisamos processar nas pontas dos dispositivos e consequentemente sobrecarregar o hardware a cada novo pacote.

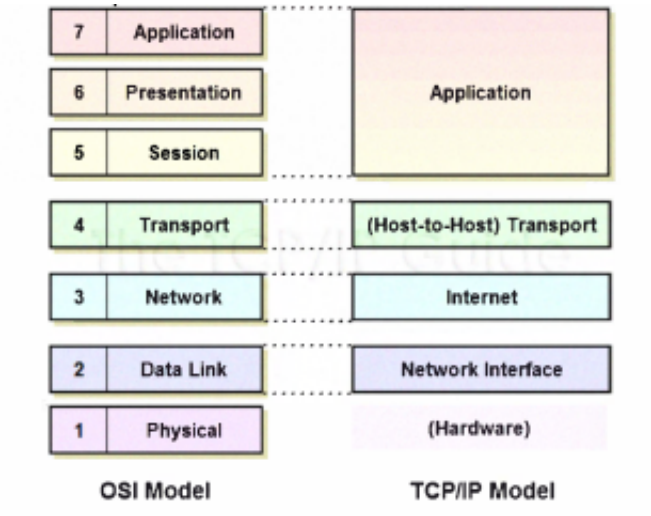

Figura 2.13: Relação entre os modelos OSI e TCP/IP (13).

- Controle e Detecção de Erros: a detecção de erros torna o tráfego de dados mais lento e gera sobrecarga nas pontas.
- Controle de Fluxo: TCP usa um mecanismo de controle de fluxo baseado em janelas. A rede pode enviar dados com muita velocidade, mas um dispositivo pode não conseguir processar, se um dispositivo começar a não processar, neste caso sua janela de recebimentos eventualmente pode ir para zero e com isso o receptor é visto pela rede como um dispositivo que não pode receber e o transmissor pode parar de enviar pacotes. Novamente, para dispositivos com baixo poder de processamento isto pode ser um problema.

Alguns destes problemas podem extrapolar a comunicação dispositivo para dispositivo e ser inviáveis também na comunicação de dispositivo para nuvem de acordo com as restrições de hardware dos dispositivos.

Em um *paper* de discussão da *Internet Society* (ISOC) de 2019, ainda sem ser oficial (14), uma conversa sobre uma proposta de um ”NEW IP” traz a temática de modernizações necessárias, dentre elas o suporte a redes heterogêneas (ManyNets), este tópico tem o objetivo de incluir diversas tecnologias como satélite por exemplo e conectar as ilhas de comunicação, trazendo maior suporte a mais tipos de dispositivos no futuro e melhorias de segurança e confiabilidade.

O pilar central deste novo protocolo seria o conceito de ManNets, conectando IoT, Redes industriais, satélites e outros. Um dos argumentos é que as novas tecnologias têm criado seus próprios protocolos, outro é que a diversidade de redes requer uma nova forma de pensar. O ”New IP” traria um espaço flexível de endereços e englobaria todos os futuros endereços (IPv4, IPv6, Semantic ID, Service ID, Content ID, People ID, Device ID, dentre outros).

No *Internet Experiment Note 48* [IEN48], um artigo de 1978, que dentre uma série de outros na época documentaram o trabalho para chegar a Internet, Vint Cerf disse: "*The basic objective of this project is to establish a model and a set of rules which will allow data networks of widely varying internal operation to be interconnected, permitting users to access remote resources and to permit intercomputer communication*

across the connected networks"(14).

A figura 2.14 apresenta uma demonstração do TCP/IP em 1977 interconectando pelo menos 3 tipos de redes (rádio, satélite e ARPANET), o que demonstra a conexão com e sem fio, sob redes sem fio e aplica a teoria da interconectividade na época (14).

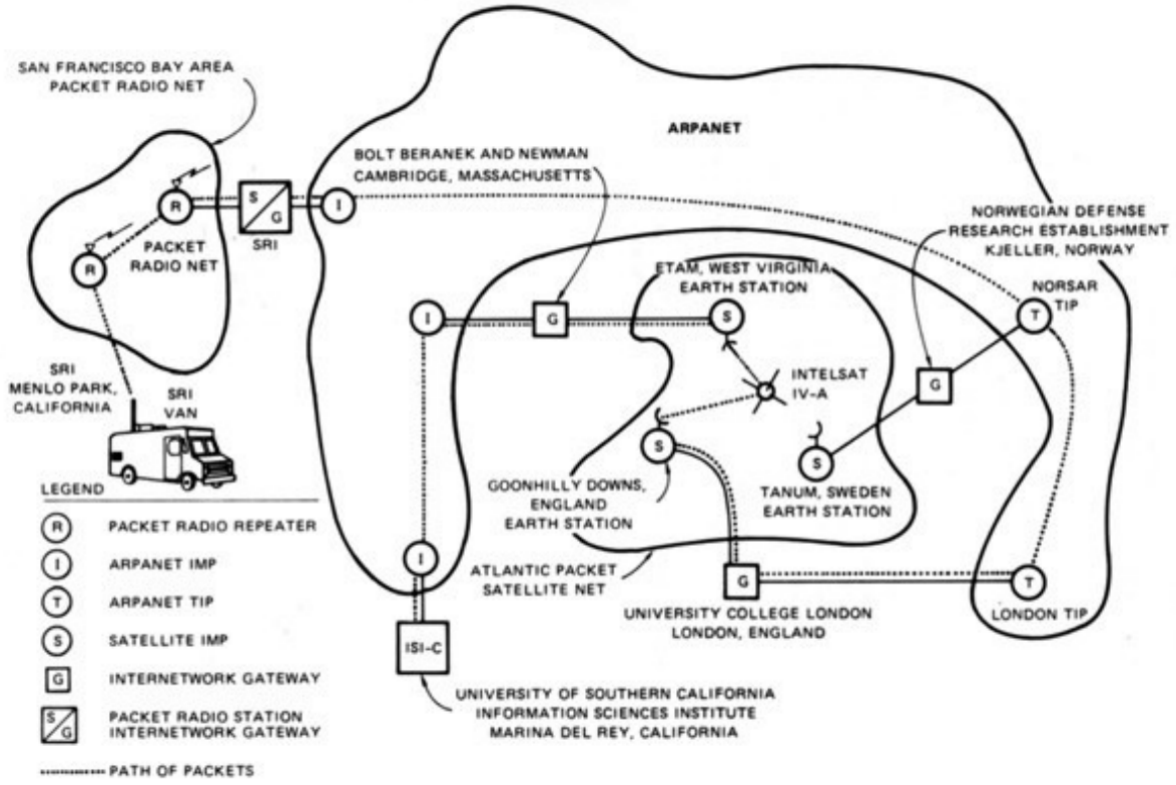

Figura 2.14: Demonstração do TCP/IP: link entre a ARPANET, PRNET, e SATNET em 1977 (14).

Os endereços IPs são usados para endereçamento a fim de conectar os pontos, atualmente temos dois formatos definidos por IPv4 e IPv6, sendo este último o mais recente surgido para resolver a falta de endereçamentos disponíveis da versão anterior. O padrão IPv4 usa 32 bits para criar um endereço único e exclusivo, formado por quatro blocos com variação de 0 a 255, separados por um ponto, o que traz para uma representação em base decimal um número binário de oito dígitos, chamado de octeto, um exemplo de um IPv4 é o endereço 16.1.140.234. O padrão IPv6 utiliza 128 bits para criar seu endereço, com oito grupos de números hexadecimais (base 16), variando entre 0-FFFF e separados por dois pontos, um exemplo de IPv6 é endereço 2001:0DB8:00AD:000F:0000:0000:0000:0001, por definição zeros a esquerda podem ser omitidos, fazendo que o endereço do exemplo seja simplificado na escrita por 2001:DB8:AD:F:0:0:0:1, uma outra regra é que podemos unir os blocos vazios por :: uma única vez no endereço, o que simplificaria novamente a escrita do endereço e o transformaria em 2001:DB8:AD:F::1 (36) (88).

2.4.3 Topologias

As principais topologias utilizadas no desenvolvimento de comunicação IoT são **estrela**, ***mesh*** e **ponto a ponto** (89).

2.4.3.1 Ponto a Ponto

É importante entender que na topologia ponto a ponto para que um dispositivo envie mensagem a outro ele precisa estar conectado a ele, o que faz com que a cobertura da rede se estenda apenas até o alcance de comunicação direta entre 2 dispositivos conforme pode ser vista na figura 2.15 (89).

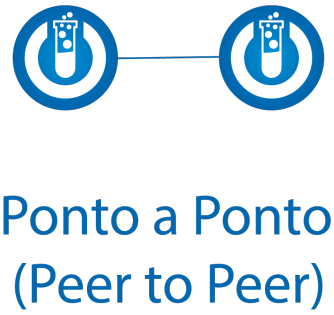

Figura 2.15: Topologia ponto a ponto

Fonte: Autor

2.4.3.2 Estrela

A topologia estrela faz uso da topologia ponto a ponto para introduzir o uso de concentradores e *gateways* de dados. Na topologia estrela todos os dispositivos se comunicam diretamente com o concentrador, e podem se comunicar de maneira indireta com os outros dispositivos da rede, usando o concentrador como intermediário (90).

Existem dois tipos de topologia estrela, elas são chamadas: estrela e estrela estendida. Uma rede que usa a topologia estrela estendida funciona como se houvesse várias redes estrela conectadas através de um dispositivo central. A figura 2.16 demonstra o funcionamento destas redes.

A grande vantagem de uma rede estrela é que sua performance é previsível, consistente e rápida, o pacote trafega somente por um ponto para chegar ao destino (89).

2.4.3.3 *Mesh*

Uma rede *mesh* consiste em três tipos de nós, sendo um *gateway* que faz a ponte com o mundo exterior, os nós de sensores simples e os nós de sensor / roteador, que são nós de sensor com capacidade de repetidor / roteamento(89). Esta topologia faz uso da comunicação ponto a ponto para conectar os dispositivos à rede, porém diferentemente da topologia estrela onde os dispositivos de ponta se conectavam apenas ao dispositivo central, neste caso todos os dispositivos podem estar interconectados. Na topologia *mesh* ainda

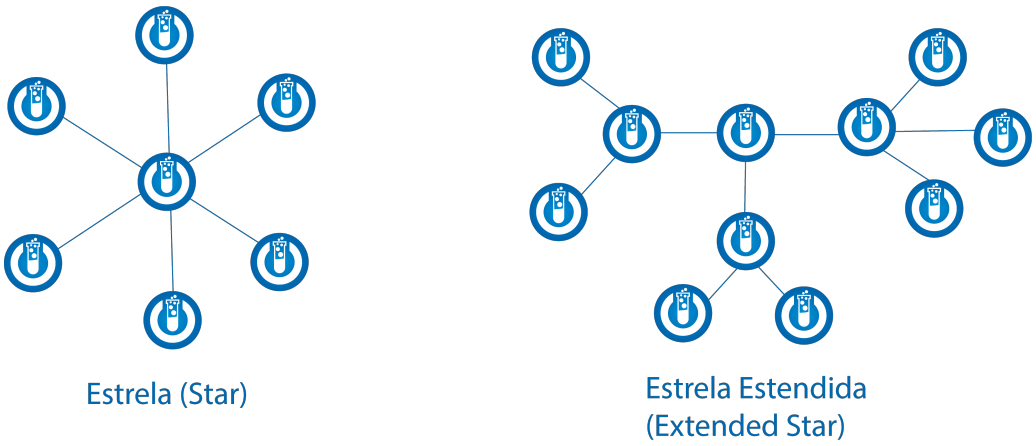

Figura 2.16: Topologia estrela

Fonte: Autor

é possível utilizar um *gateway* para levar os dados até os serviços de nuvem (90).

Uma rede *mesh* em que todos os dispositivos estão interconectados é chamada de *full mesh*, *mesh* cheia ou completa, entretanto não existe uma necessidade de ter todos os dispositivos interconectados para que a rede funcione corretamente, quando isso não ocorre a rede é chamada de *partial mesh* ou *mesh* parcial. A figura 2.17 demonstra as interconexões dos dois tipos de rede *mesh*.

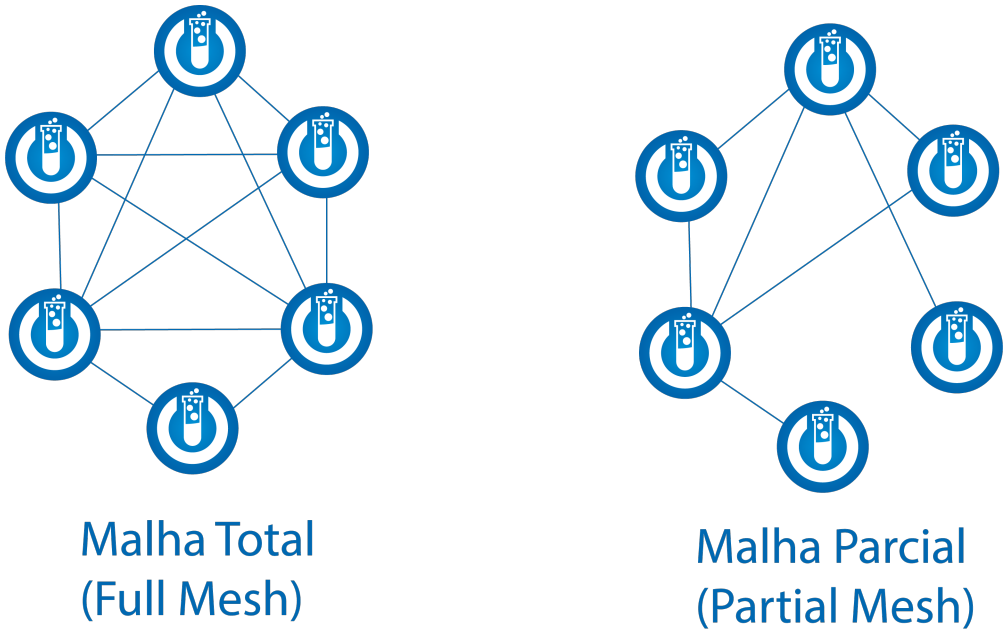

Figura 2.17: Topologia *mesh*

Fonte: Autor

A principal desvantagem das redes *mesh* é o seu nível de complexidade para implantação nos locais, além de maior latência, pois os pacotes passam por vários nós até atingir o *gateway*. E um dos pontos importantes quando se fala de *mesh*, é que os dispositivos estarão sempre recebendo e retransmitindo dados o que acarretará no consumo maior de energia para esta comunicação. Esta é uma rede que não

depende somente da distância entre dois nós e por isso pode se estender até grandes áreas com uso de todos os pontos (89).

2.4.4 *Wireless Sensor Networks*

Wireless Sensor Networks são redes de sensores com a função de agrupar sensores ou até mesmo serem redes sensíveis ao contexto para comunicação de sensores com a Internet por meio de um elo de saída ou concentração, as redes trabalham com algoritmos que norteiam a criação de novos nós de comunicação e também com nós que operam como roteadores de pacotes (91) (92).

Comumente usando mais de uma topologia, as redes de sensores em sua maioria convergem para uma arquitetura de pesos quase sempre igual (92), este modelo foi definido por Barrat, Barthélemy, Vespignani e é conhecido como (BBV), as extremidades ou bordas tem poder de roteamento ou *gateway* permitindo o roteamento de pacotes com ou sem poder de processamento/armazenamento e também a comunicação com a nuvem. A figura 2.18 mostra as diferentes topologias que podem ser usadas em uma rede de sensores sem fio. As aplicações são as mais diversas desde sensores industriais, biomédicos, casas/prédios/cidades inteligentes ou simplesmente em aplicações móveis de rastreamento ou telemetria aplicadas à robótica, veículos e outros (91).

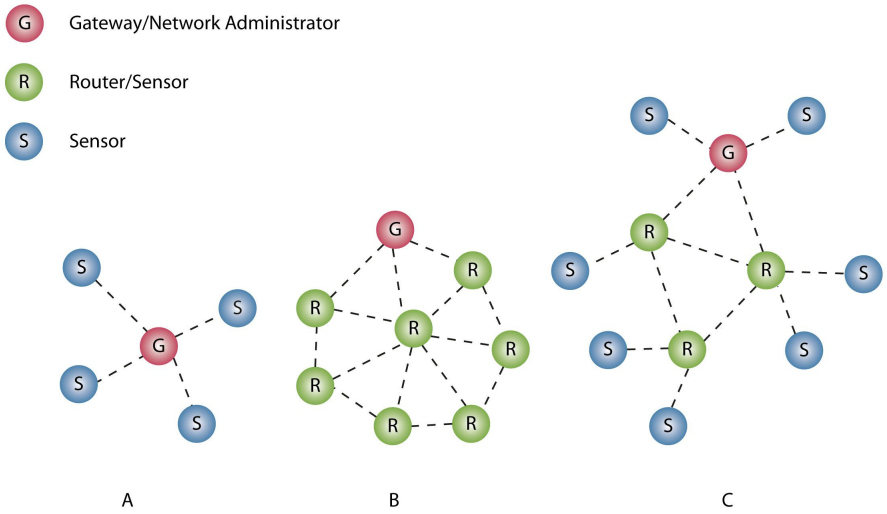

Figura 2.18: Diferentes topologias de WSN (15).

Uma das conclusões do trabalho de Barrat, Barthélemy, Vespignani (93) em 2004 foi que muitas das redes e modelos estudados poderia crescer de acordo com o modelo de pesos definido por eles. O modelo proposto traz uma visão da diversidade de variáveis de da forma dinâmica que se comporta uma rede de sensores, trazendo a correlação entre os agentes externos para dentro de cenários onde eles acreditavam ser o mais consistente possível para cenários de uso reais, tal conclusão fora corroborada posteriormente por Zhang(92) em 2012 quando ele disse que as redes de sensores convergiam para o modelo de pesos proposto pelos autores.

Segundo Macedonio (91) uma das preocupações com as redes de sensores está na segurança, outra problemática está em garantir a comunicação entre os pontos e a saída (92), redes inseguras são vulneráveis

a diversos tipos de ataques, dentre eles alteração de mensagens e consumo abusivo de recursos da rede degradando sua qualidade, neste cenário o diferencia são os protocolos, que são os responsáveis pela troca de chaves, criptografia e por garantir a integridade, desligamento e tráfego de dados.

Segundo Zhang (92), outra preocupação nesse tipo de rede é garantir que interferências não intencionais não atrapalhem a comunicação ou perturbem o ambiente aumentando a perda de pacotes, aumento de uso de bateria, um número maior de dispositivos para cobrir uma área, dentre outros. Algo comum neste cenário das interferências é o Wi-Fi de redes caseiras que operam para uma rede de sensores que trabalhe na frequência de 2.4GHz por exemplo (91). Estes problemas de interferência e outros são tratados a nível de protocolo e podem ter seus algoritmos otimizados para ambientes com determinadas características, ou até mesmo focados em maior segurança.

2.4.5 Protocolos de Comunicação

Dentre os protocolos e tecnologias de comunicação que temos para a Internet das Coisas, podemos dividir dois grandes grupos, os protocolos que estão nas camadas inferiores do modelo de comunicação OSI, com ou sem fio, mais ligadas ao hardware e os protocolos de camada de aplicação, que vão atuar em conjunto com algum meio de comunicação. No primeiro grupo, dentro do enlace físico, podemos fazer uma nova subdivisão por distância, ou alcance, da tecnologia, sendo basicamente 3 grupos: pessoais, regionais e de grandes áreas (WAN) conforme pode ser visto na figura 2.19.

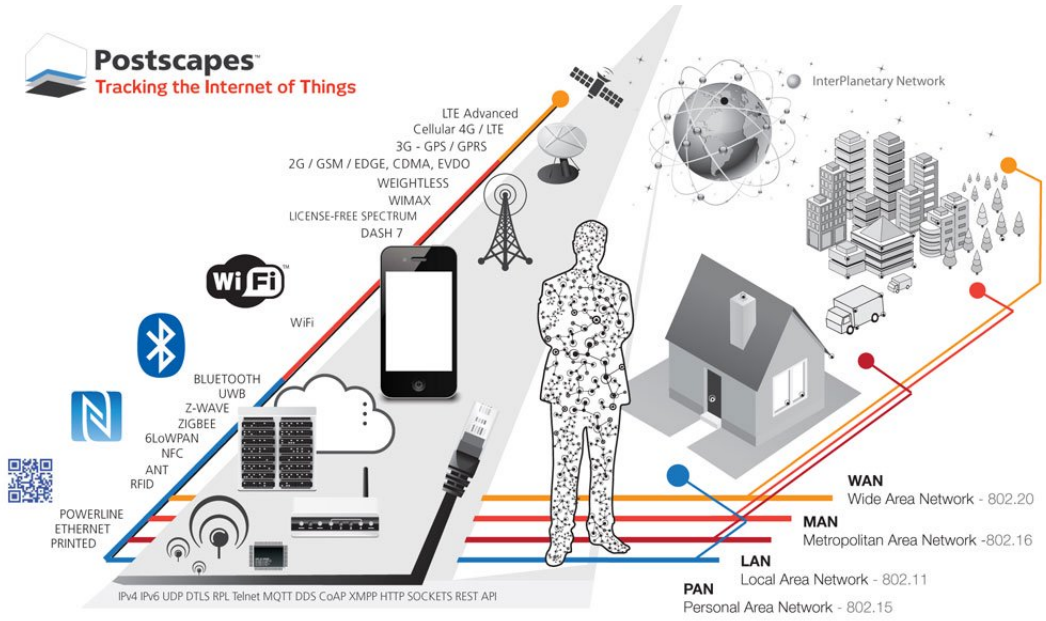

Figura 2.19: Uma visão geral de protocolos para IoT (16).

Neste capítulo serão apresentados alguns protocolos, a fim de complementar o entendimento das arquiteturas e como eles podem estar presentes atuando em conjunto ou separadamente na comunicação dos dispositivos, não sendo abordadas questões de profundidade técnica como enlaces de comunicação, definições de hardware ou avançadas de segurança.

2.4.5.1 LPWAN

Dentre os protocolos que se encaixam nas camadas iniciais do modelo OSI, podemos dizer que os que incorporam a tecnologia sem fio de *Low Power Wide Area Network* (LPWAN), longo alcance e pouco consumo de energia, vem ganhando popularidade na indústria graças a necessidade da Internet das Coisas de ter longevidade de bateria somada a possibilidade da implantação de dispositivos em locais afastados e garantindo com isso a diminuição do custo de operação da solução, bem como seu custo inicial de hardware que vem caindo para esta tecnologia (94) (17). Redes de *Low Power Wide Area* (LPWA) são um novo paradigma que vem para complementar as redes tradicionais, estima-se que aproximadamente um quarto dos dispositivos de IoT venham a se conectar usando uma das tecnologias de LPWA. Um dos pontos de crescimento vem de que muitas das soluções de M2M e IoT atuais, novas ou legadas, vem se baseando em redes celulares 2G e 3G, que em breve serão desligadas, mesmo com sua continuidade as redes atuais não tem a mesma performance de uso de bateria das LPWA, o que soma mais um ponto em favor deste novo paradigma, atualmente temos em operação algumas das tecnologias, dentre elas citamos LoRaWan, SigFox e NB-IoT (94).

Com possibilidades de uso sem troca de bateria por até 10 anos ou mais, distâncias que passam dos 10Km, as redes vêm ganhando números de soluções em produção e presença em todos os continentes, na figura 2.20, podemos ver o alcance versus a capacidade de tráfego de dados de algumas tecnologias de comunicação.

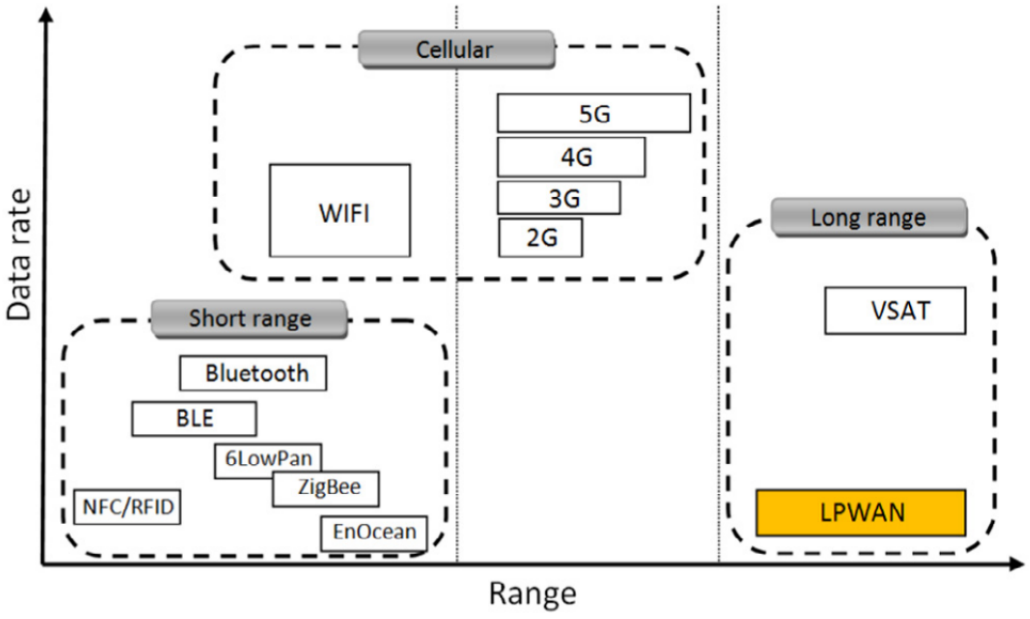

Figura 2.20: Posicionamento das tecnologias LPWAN analisando capacidade de tráfego de dados e distância (17).

2.4.5.2 Narrow Band Internet of Things

A *Narrow-Band Internet of Things* (NB-IoT) é uma tecnologia de comunicação de dados, *Low Power Wide Area* (LPWA), aplicada diretamente à dispositivos de Internet das Coisas com uso restrito de banda

e picos de transmissão de dezenas de Kbps, este é um protocolo disponível desde 2016 por meio de uma atualização da 3GPP para percepção e aquisição de dados de aplicações com baixa transferência de dados. As aplicações mais comuns são as de monitoramento e ambientes inteligentes, NB-IoT aceita conexões massivas com ultrabaixo consumo de energia, foco em mensagens de pequeno tamanho e suportado por uma rede de comunicação celular (95) (17).

Dentro do *Narrow Band*, temos além do NB-IoT, o *enhanced Machine Type Communication* (eMTC), que é uma derivação do protocolo LTE (LTE-M), onde foram feitos cortes para acomodar a comunicação entre as coisas com taxas de pico de *uplink/downlink* de até 1Mbps com conexões massivas. As principais aplicações são veículos conectados, *smart-health*, e outras que precisem de uma banda um pouco maior, maior controle da conexão e abrigando também o uso do protocolo IP. Aplica-se ao eMTC a longevidade da bateria e também a distância grande entre a torre e o dispositivo, permitindo uma maior utilização da tecnologia, uma visão mais técnica das diferenças entre o eMTC e o NB-IoT podem ser vistas na tabela 2.2 (35).

Em ambos, tanto no Categoria M1, eMTC, quanto no NB-IoT, podemos ter alcances que chegam ao subsolo para cobertura dos dispositivos e possibilidade de atualização remota do *firmware* dos dispositivos, uma das diferenças entre os dois que tem que ser citada é a responsividade, que no CAT-M chega em segundos enquanto no NB-IoT pode chegar a minutos, o que dependendo da aplicação pode inviabilizar (35) (95). Para exemplificar, poderíamos trazer um POS de vendas de cartão que hoje utiliza 3G ou 2G (96) para evoluir com um hardware de CAT-M1, enquanto um medidor de água ao invés de usar 2G ou 3G poderia usar um CAT-NB1, sendo ambos dentro da definição do termo NB-IoT.

Tabela 2.2: Diferenças entre NB-IoT e eMTC (35).

Technical index		**NB-IoT**	**LTE FDD eMTC**	**LTE TDD eMTC(3:1)**
Carrier bandwidth		200 kHz	1.4MHz	1.4Mhz
Peak rate	Uplink	66.7 kbps	375 kbps (half duplex) / 1 Mbps (full duplex)	200 kbps
	Downlink	32.4 kbps	FD: 800 kbps, HD: 300 kbps	750 kbps
Coverage (compared with GSM)		Increased 20 db	Increased 11 db	
Power Consumption		About 10 years	About 10 years	
Module cost		Less than $5 initially	Weaker than NB	
Connection		About 50 thousands / cells	Less than $10 initially	
Mobility		Cell reselection in idel stage	Cell switch in connection stage	
Phonetic ability		Nonsupport	Limited capacity	Weaker than FDD

A possibilidade de se combinar as tecnologias e complementar casos de uso é uma realidade quando fala-se de *Narrow Band*. Alguns fabricantes têm módulos que combinam os dois a fim de facilitar o desenvolvimento dos dispositivos com uma única pastilha de comunicação, como é o caso do SIM7080G que alia as duas tecnologias em um único módulo (97).

2.4.5.3 LoRa e LoRaWAN

LoRa é uma nova tecnologia que funciona em uma zona de banda não licenciada sub 1GHz para operar uma comunicação sem fio, o nome se deu pela funcionalidade de atingir grandes distâncias (*Long*

Range), uma camada física que modulariza ondas sem fio, baseada em uma tecnologia da década de 40 tradicionalmente usada por aplicações militares. LoRaWan é um protocolo de comunicação que se utiliza da camada física LoRa para o desenho da arquitetura, conforme pode ser visto na figura 2.21. Na rede LoraWan um dispositivo (nó) não está vinculado a um *gateway*, sendo seu dado recebido por mais de um *gateway*, cada um deles vai enviar os dados para os servidores baseados em nuvem por meio de um outro canal de comunicação, como redes celulares, wifi, Ethernet, satélite (18) (17).

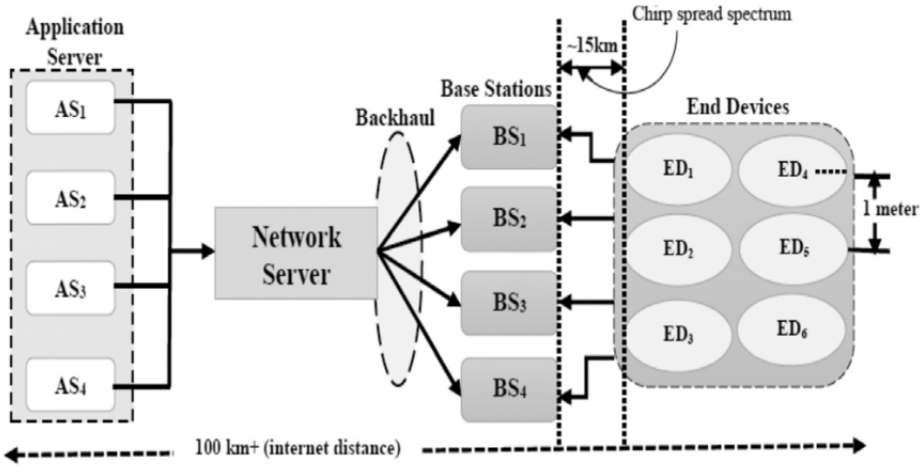

Figura 2.21: Arquitetura de uma rede LoRaWan (18).

Dentre as funcionalidades, temos taxas de transferência de 290 bps a 50 Kbps, e uma eficiência de bateria que chega a casa dos 10 anos, LoRa tem uma resistência muito grande a interferências e neste quesito chega a ser mais robusta que NB-IoT. Uma aplicação focada em casos de uso de telemetria e monitoramento, que pode tanto ser usada com servidores de aplicação públicos quanto privados, permitindo seu uso e adoção com modelos totalmente privados, com ou sem uso de redes de LoRaWan (18).

A Frequência do LoRa para o Brasil é o de 902 a 928 MhZ, porém nesta faixa de frequência temos limitações de outras tecnologias já utilizando a banda de 907,5-915,0 MHz, por isso a faixa adotada é a mesma da Austrália que vai de 915,0-928MHz (98).

2.4.5.4 Sigfox

A SigFox é uma tecnologia criada por uma empresa francesa que leva o mesmo nome, ela usa um método de transmissão de rádio padrão denominado BPSK, que utiliza a alteração da fase da onda de rádio para codificar os dados. Por ser uma tecnologia de banda muito estreita (*Ultra Narrow Band*) os dispositivos SigFox fazem um ótimo uso da potência disponível o que permite aplicações que utilizem comunicação em longas distâncias, mesmo em canais com interferências e ruídos, como monitoramento remoto (99). A cobertura SigFox é global, operando em bandas não licenciadas no mundo, as frequências de rádio variam de 862 a 928 MHz dependendo da região, no Brasil opera em 902 MHz (100).

Sua arquitetura é horizontal, possuindo duas camadas principais: a Network Equipment, que recebe as mensagens dos dispositivos, e a SigFox Support System, que processa os dados e envia para o usuário.

A rede é baseada na topologia estrela, um dispositivo não está conectado a uma estação base específica, a mensagem transmitida é recebida por qualquer estação base (99).

Outro aspecto do SigFox é o tamanho mensagens: 12 bytes (*payload* útil) de *upload* por mensagem no limite de 140 envios por dia. Para os 12 bytes de carga útil, a SigFox utilizara 26 butes no toal. O download é: 4 retornos por dia de 8 bytes (*payload* útil) (99).

2.4.5.5 Ethernet, Power over Ethernet e Ethernet Industrial

Ethernet é definida e padronizada pelo IEEE 802.3, podendo tanto operar em cima de meio físico de fibra óptica quanto em pares de cobre trançado, utilizada em 85% das redes locais no mundo. De forma geral, dispositivos (PCs ou outros) ligados a uma rede Ethernet possuem um endereço IP como seu identificador único frente aos outros dispositivos na mesma rede, podendo este endereço ser privado ou público. Uma visão de onde a Ethernet se encaixa dentro das camadas OSI pode ser vista na figura 2.22 (36). Uma visão das mídias, distâncias e velocidades podem ser vistas nas tabelas 2.3 e 2.4.

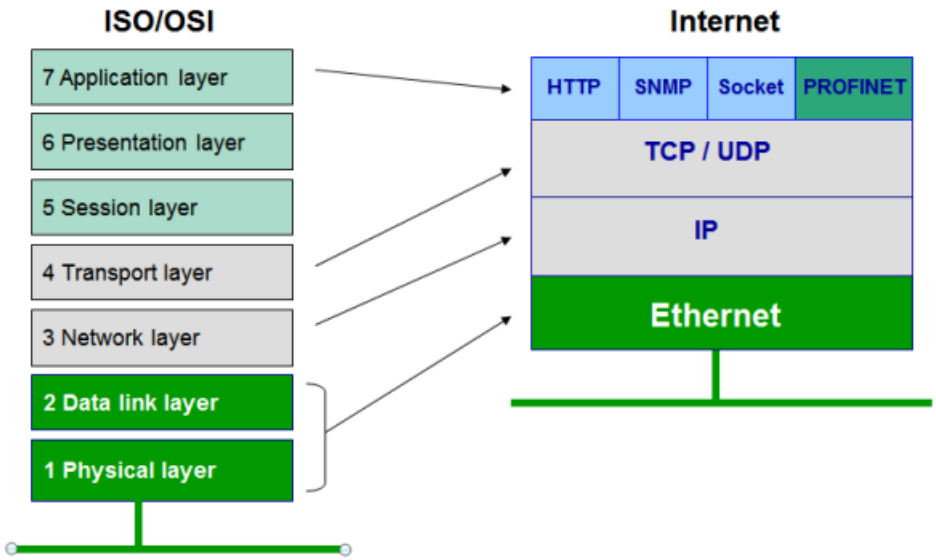

Figura 2.22: Uma visão de onde a Ethernet se encaixa no modelo OSI (19).

O *Power Over Ethernet* (POE) é uma tecnologia para que além da comunicação feita pelo protocolo Ethernet, em mídias de cabo de cobre trançado, no cabo físico passe também alimentação para o dispositivo conectado. A principal vantagem é a possibilidade de alimentar dispositivos distantes sem que haja a necessidade de outra infraestrutura para a rede elétrica, sendo um cabo responsável por tudo, bastante difundido no uso com câmeras IP, telefones Voip e uma opção excelente para dispositivos de IoT cabeados. Um sistema POE é composto por uma unidade de alimentação, (PSE) e um dispositivo energizado (PD), O PSE pode ser por exemplo um *switch* de rede POE que vai coordenar a injeção de energia e o tráfego *Ethernet*, já no dispositivo um circuito faz a separação dos dados e da energia por meio de um circuito próprio (37).

Um grupo de trabalho criado em 1999 criado dentro do IEEE (802) iniciou a formulação dos padrões e deste então eles vêm sendo atualizados e utilizados por fabricantes, a Cisco também criou uma definição

Tabela 2.3: Redes Ethernet adaptada de (36).

Speed	Descriptor	Media	Max. Length
1 Mb/s	1Base5	2-twisted wire pairs - copper	500 m
10 Mb/s	10Broad36	75 ohm cable TV (CATV) cable	3,600 m
	10Base2	RG 58 coax cable	185 m
	10Base5	Special 50 ohm coaxial cable	500 m
	10Base-FL	Multi-mode fiber (850 nm)	2,000 m
		Single mode fiber (1310 nm)	10,000 m
	10Base-T	Multi-mode fiber (850 nm)	100 m
100 Mb/s	100Base-FX	Multi-mode fiber (850 nm)	2,000 m
		Single mode fiber (1310 nm)	10,000 m
	100Base-T2	2 pairs UTP CAT3 or better, full-duplex	100 m
	100Base-T4	4 pairs UTP CAT3 or better, full-duplex	100 m
	100Base-TX	2 pairs UTP CAT5 or better, full-duplex	100 m
1 Gb/s	1000Base-CX	Copper jumper cable	15 m
	1000Base-LX	Long wavelength Multi/Single mode fiber	5,000 m
	1000Base-SX	Multi-mode fiber (850 nm) Core size 62,5 um 50 um	 275 m 500 m
	1000Base-T	4-CATe,CAT6 or better copper pairs	100 m
10 Gb/s	10GBase-SR	2 optical fibers, Full-Duplex serial single-mode	400 m
	10GBase-LR	2 optical fibers, Full-Duplex serial single-mode	10,000 m
	10GBase-ER	2 optical fibers, Full-Duplex serial single-mode	40,000 m
	10GBase-CX4	4- Twinaxial copper	15 m
	10GBase-T	4-CAT 6, 6a or 7	55 m CAT 6 100 m CAT 6a or 7
UTP = Unshielded Twisted Pair copper wire			

para seus dispositivos, a tabela 2.5 mostra as mudanças de versões do POE, com isso podemos ver a evolução no aumento da potência para os dispositivos conectados (37).

Existem muitos protocolos de Ethernet Industrial, dentre eles o EtherNet / IP, definido pelo padrão IEEE 802.3, projetado para uso em ambientes industriais em aplicações de automação e controle de processos, o EtherCat e protocolo de controle de transmissão Modbus (TCP). Todos estes usam a camada de link Ethernet padrão e protocolo de internet, juntamente com um protocolo de camada de aplicativo proprietário conforme vistos na figura 2.22 (36).

2.4.5.6 Rede Celular Convencional

A nossa telefonia móvel celular vem evoluindo desde a 2a geração, temos grandes avanços e falamos de menos de 15 anos para um salto de velocidade de transferência de dados de 64Kbps para 10Gbps, respectivamente falando do 2G para o 5G, este último ainda recente no cenário mundial (101) (35).

Tabela 2.4: Categorias de Cabos Ethernet adaptada de (36).

Cable Category	**Cable Type**	**Maximum Data Transmission Speed (varies by length)**	**Max Bandwidth**
3	UTP	10 Mbps	16 MHz
5	UTP	10/100 Mbps 1	100 MHz
5e	UTP	1 Gbps	100 MHz
6	UTP or STP	1 Gbps	250 MHz
6a	STP	10 Gbps	500 MHz
7	SSTP	10 Gbps	600 MHz
UTP= unshielded twisted copper pair STP = shielded twisted copper pair SSTP= screened, shielded twisted copper pair			

Tabela 2.5: Tabela de Diferenças entre revisões PoE. Adaptada de (37).

Standard Category	802.3af	802.3at (PoE Plus)	Cisco UPoE (Non-standard)	802.3bt (4PPOE)
Release year	2003	2009	2011	2018
PSE output voltage	44 $\sim$ 57V DC	44 $\sim$ 57V DC	44 $\sim$ 57V DC	44 $\sim$ 57V DC
Maximum current	350mA	600mA	1200mA	1920mA
PSE output Power	$\leq$ 15.4W	$\leq$30W	$\leq$ 60W	$\leq$ 100W
PD input voltage	36 $\sim$ 57V DC	42.5 $\sim$ 57V DC	42.5 $\sim$ 57V DC	42.5 $\sim$ 57V DC
PD Standard category	12.95W	25.5W	51W	74.55W
PD types	PD Type 1	PD Type 2		PD Type 3 and Type 4
Cable requirements	Cat. 3 and above	CAT-5 or better	CAT-5 or better	CAT-5 or better

A primeira geração de telefonia móvel celular surgiu no início dos anos 80, usando transmissão analógica para dispositivos de fala somente, já a segunda geração ou o 2G, como é comumente citado, fazia uso uso de compressão e descompressão, tinha dentro da mesma família o GSM que era o 2G, o GPRS que era a versão mais avançada também conhecida como 2.5G e seguido pelo EDGE que era o 2.75G (101).

A terceira geração já trazia transmissões de dados de pelo menos 200Kbit/s, esta era uma geração que já trazia a ideia de conectar computadores com o CDMA2000, TD-SCDMA e W-CDMA (UMTS) (101).

A quarta geração vem com taxas de transferência na casa de 20MB/s ou maiores, gestão de QoS, Conexão IP de ponta a ponta e muita mobilidade, o que nos habilita para conversas de vídeo, mensagens, aplicações de telefonia dentre outras que consomem muito tráfego.

A quinta geração ou o 5G, vem com três grandes objetivos, sendo a implementação de alta capacidade de escala e uma alta conectividade como a primeira delas, seguido por suporte a uma ampla gama de serviços, aplicações com possibilidades de requisitos diversas. Flexibilidade e eficiência é o último da tríplice de objetivos e vem para atuar em cenários de implantação diversos.

Hoje muitos dispositivos de comunicação M2M ou até mesmo IoT ainda estão baseados nestas tecnologias, o que traz uma problemática de modernização para novas tecnologias que esbarra em custos de transição, *lock-in* de empresas e muito mais. A rede 2G e 3G já estão na fila para desligamento, mas ainda

se sustentam por causa de dispositivos de telemetria ou os POS de transações comerciais, o que deve mudar nos próximos anos, todos estes são fortes candidatos a migração para tecnologias aderentes a LPWAN, seja de rede celular ou não (96).

2.4.5.7 Bluetooth

Bluetooth é um padrão para comunicação de curta alcance, concebido inicialmente pela Ericsson com uma ideia de substituição de cabos em 1994, o Bluetooth vem sendo incluído cada vez mais em equipamentos, dentre eles podemos citar telefones celulares, fones de ouvido, dispositivos médicos, veículos, impressoras dentre outros. A Ericsson juntou forças com a Intel Corporation, IBM, Nokia Corporation e Toshiba Corporation a fim de formarem o Bluetooth Special Interest Group (SIG) no início de 1998. A 3Com, Lucent / Agere Technologies Inc., Microsoft e Motorola juntaram se ao grupo no final de 1999, os grandes nomes da indústria unidos fizeram com que os trabalhos junto ao grupo e sua força no mercado elevassem o Bluetooth para padrões abertos e com isso garantiram a rápida aceitação e compatibilidade no mercado. A especificação Bluetooth resultante, desenvolvida pela Bluetooth SIG, está aberta e disponível gratuitamente no site oficial do Bluetooth. A especificação divide o protocolo em três grandes grupos, o primeiro responsável pelo transporte que contém as definições de rádio e a identificação de outros dispositivos na rede, bem como seu enlace inicial, outro responsável pelo *Middleware* embarca outros protocolos de mercado nas soluções, dentre eles o IP e o TCP, na terceira e última camada de aplicação, esta separação pode ser vista na figura 2.23 (20). Um ponto importante na evolução do protocolo foi a introdução do *Bluetooth Low Energy* ou BLE na versão 4, o que tornou o consumo de bateria menor uma vez que ele fica em modo de dormência enquanto aguarda outra transmissão, como a transferência é muito rápida graças a banda da tecnologia, a bateria pode durar muito tempo e habilitar este modo para dispositivos que precisam de longevidade sem manutenção de bateria (38).

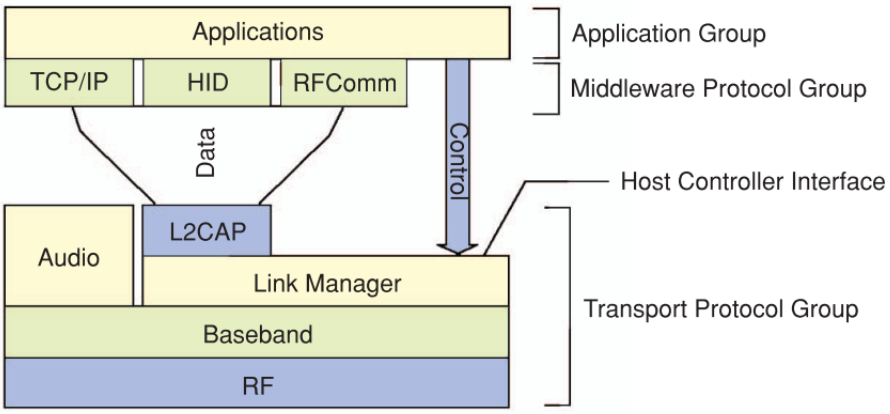

Figura 2.23: Pilha do protocolo Bluetooth (20).

O avanço persistente da tecnologia Bluetooth vem tornando cada vez maior a sua adoção nos dispositivos de Internet das Coisas, operando a 2.4 GHz, as taxas de transferência tem aumentado a cada nova versão e seu consumo de bateria diminuído. BLE ou Bluetooth smart permite que um usuário se conecte sem a necessidade de nenhuma infraestrutura (rede Ad Hoc), algumas previsões preconizam que até um terço dos dispositivos de IoT irão usar Bluetooth. Ainda com alguns problemas de segurança nas versões

Tabela 2.6: Comparação Técnica das versões do Bluetooth. Adaptada de (38).

Característica	**Bluetooth Clássico**	**Bluetooth 4.x**	**Bluetooth 5**
Rádio Frequência	2400 a 2483.5	2400 a 2483.5	2400 a 2483.5
Alcance (m)	Até 100	Até 100	Até 200
Técnica de acesso médio	Salto em Freqüência	Salto em Freqüência	Salto em Freqüência
Taxa de dados nominal (Mb/s)	1-3	1	2
Latência (ms)	<100	<6	<3
Topologia de rede	Piconet, Scatternet	Estrela em barramento, Mesh	Mesh
Solução Multi-hop	Scatternet	Sim	Sim
Conceito de Perfil	Sim	Sim	Sim
Nós	7	Ilimitado	Ilimitado
Tamanho da mensagem (bytes)	Até 358	31	255
Organismo de Certificação	Bluetooth SIG	Bluetooth SIG	Bluetooth SIG

4, atualmente na versão 5 temos um grande avanço não somente neste campo, mas também no consumo, dispositivos conectados, utilização da topologia *mesh* para conectividade, dentre outras. Uma visão de evolução simplificada pode ser vista na tabela 2.6 (102).

Segundo um estudo feito com M. Collotta et al. em 2018, o Bluetooth 5 vem com um desempenho significativo em comparação as outras versões, velocidade, alcance e capacidade de transmissão são os pontos fortes, além de segurança. Eles acreditam que nesta nova versão BLE pode se tornar um dos fortes candidatos a ser uma das melhores escolhas na comunicação sem fio da Internet das Coisas (38).

2.4.5.8 Z-Wave

Z-Wave é um padrão de comunicação sem fio proprietário (103) que promete a facilidade no desenvolvimento de dispositivos, na instalação por parte dos usuários dos dispositivos tido como habilitador de dispositivos *plug and play*, desenvolvido para ser uma solução para comunicação sem fio em *smart homes* e presente globalmente conforme visto na figura 2.24.

O Protocolo Z-Wave opera em diferentes frequências (103), que dependem de regulamentações locais, no Brasil opera nas frequências 919.8 MHz e 921.4 MHz (104).

Topologia e Alcance

Segundo a Silicon Labs (103), apesar de ter sido desenvolvido para comunicação *indoor* o alcance em ambiente aberto é de 100 metros enquanto em um ambiente fechado - casas, prédios - esse alcance diminui para uma distância média de 10 metros. O Protocolo faz uso da topologia *mesh*, com isso o problema de distância não se torna um limitante de acordo com o número de dispositivos conectados, o que torna a rede mais robusta a cada novo nó adicionado seguindo os limites da topologia e protocolo. A rede implementada para uso com Z-Wave funciona de tal forma que todos os dispositivos não alimentados por bateria operam como um repetidor, diferentemente dos dispositivos a bateria, o que lhes permitem operar com um consumo baixo. A rede permite a conexão de até 232 dispositivos.

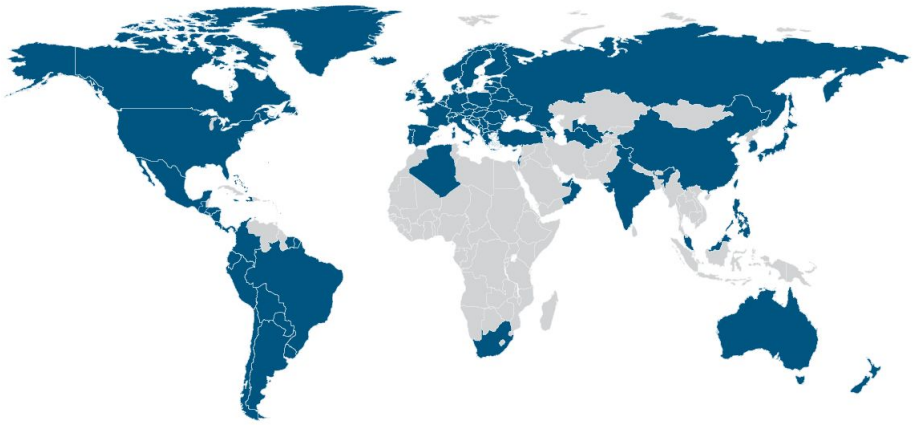

Figura 2.24: Presença do Z-Wave no mundo (21).

Segurança

Segundo a fabricante (103), o dado trafegado é garantido por uma camada de segurança, que além da confidencialidade se encarrega também da integridade e controle de pacotes trafegados, com essa garantia somente nós habilitados e seguros podem ler a mensagem e proceder com seu processamento ou transporte na malha.

Algumas das características da camada de segurança do Z-Wave são listadas a seguir (103):

- Segurança de ponta a ponta a nível de aplicação.
- Troca de chave temporária de 128 bits.
- Chave única para toda a rede.
- Algoritmo de codificação de bloco simétrico AES usando chave de 128 bits.
- Nós seguros e não seguros podem coexistir na mesma rede.
- Nenhuma solução de segurança na camada MAC e na camada de roteamento.
- Nós não seguros podem atuar como repetidores para nós seguros

2.4.5.9 ZigBee

ZigBee, foi desenvolvido e é mantido pela *ZigBee Alliance*, ele é um protocolo para *personal area networks*, é um padrão de comunicação sem fio de duas vias - envio e recebimento - de curto alcance (105), alcance esse que varia de 10 a 300 metros (106).

ZigBee é capaz de operar nas topologias estrela e *mesh* (106), o que o torna muito versátil. O fato de poder operar em uma rede *mesh* permite que sua área de cobertura seja expansível, enquanto a topologia estrela permite que um dispositivo se comunique com vários outros, usando chaves criptográficas de 128 bits e criptografia AES - *Advanced Encryption Standard* - para proteger a comunicação entre os dispositivos (105) (106).

2.4.5.10 Wireless Meter-Bus

Wireless Meter-Bus, ou como é mais comumente conhecido WM-Bus, é um padrão de comunicação europeu (EN 13757-4) que especifica comunicação entre medidores e registradores, concentradores ou *gateways* de dados (107). Amplamente usado em medidores e bastante difundido em dispositivos usados para telemetria de consumo de água, gás e energia.

O WM-Bus surgiu como uma continuação do Meter-Bus, ou M-Bus, para operar em cenário de WSN (106). O grupo *Open Metering System*, responsável pelo desenvolvimento tanto do M-Bus como do WM-Bus, propôs o uso do WM-Bus para este tipo de aplicação, particularmente nos casos de *smart meters* (108).

WM-Bus está definido em 3 diferentes frequências, que são: 168 MHz, 433 MHz e 868 MHz. Possui também 6 diferentes modos que podem ser operados nessas frequências, cada modo só pode ser operado na frequência na qual ele foi definido como pode ser observado na tabela 2.7 (107).

Tabela 2.7: Modos de operação do WM-Bus

Modo	Frequência (MHz)	Comentário
S (Stationary)	868	Envio de dados poucas vezes ao dia
T (Frequent Transmit)	868	Envio de dados várias vezes ao dia
C (Compact)	868	Mesmo uso que o modo T, porém alta taxa de envio
N (Narrowband)	169	Longo alcance
R (Frequent Receive)	868	Coletor pode receber dados de vários medidores, usando diferentes canais
F (Frequent Tx and Rx)	433	Comunicação bidirecional frequente

O WM-Bus tem suporte apenas para redes estrela, o que permite a uso de concentradores e *gateways* de dados, dispositivos que recebem dados de vários medidores e salvam esses dados, no caso dos concentradores, ou enviam esses dados para um outro serviço de dados, em sua maioria esses dados são enviados a serviços de nuvem, através da internet, onde podem ser armazenados por longos períodos e/ou processados.

O fato de o WM-Bus usar uma topologia de rede simples como a estrela traz também consigo o benefício de ter baixos requisitos (107), pode ser implementado em *MCU*'s de baixo consumo, o que faz com que custo final da solução fique menor.

2.4.6 Protocolos de Camada de Aplicação para Internet das Coisas

Diferente da Web que pode utilizar somente um protocolo de aplicação, a Internet das Coisas tem necessidades diferentes a cada novo cenário, por vezes é comum o uso de mais de um protocolo para uma mesma solução, semelhante aos protocolos de comunicação apresentados anteriormente (39). Nos tópicos anteriores foram apresentadas soluções de comunicação via rádio ou cabeada para interligar os dispositivos, agora, dentro da camada de aplicação, os protocolos mais conhecidos para uso em Internet das Coisas são o MQTT, AMQP, CoAP e HTTP (39) (109) (12), a tabela 2.8 traz um resumo destes protocolos. Uma visão mais abrangente sobre a localização destes protocolos em conjunto com outros e sua provável união em uma solução pode ser vista na figura 2.12.

Hyper Text Transport Protocol (HTTP), o protocolo da Web, possivelmente o mais usado em todos os sistemas de internet, é um protocolo dito como fundamental por alguns autores para o universo da internet, implanta diretamente o modelo de requisição e resposta e não trabalha com *push* sem necessariamente uma solicitação originando. Um ponto interessante do HTTP é a implantação do REST, o que permite que os sistemas consigam integrar o modelo de trabalho de estado de entidades com muita facilidade tendo as operações de *CRUD (Create, Read, Update and Delete)* facilitadas pelo uso dos métodos de POST, GET, PUT e DELETE do HTTP. Pode se imaginar este cenário para garantir a interoperabilidade do protocolo com os dispositivos de Internet das Coisas trazendo esta facilidade ao controle dos estados deles, o problema no HTTP é o seu peso de cabeçalho e ele operar em cima do TCP, o que aumenta em algumas vezes o tráfego na rede e requer um processamento maior no dispositivo para que ele faça a implementação da pilha toda e garanta uma conexão. Modificações que vieram no HTTP/2.0 permitem um uso da rede mais eficiente, reduzindo latência e permitindo uso de cabeçalhos compactados além de múltiplas trocas simultâneas na mesma conexão, esta evolução pode ser vista na figura 2.25 (23) (39) (22).

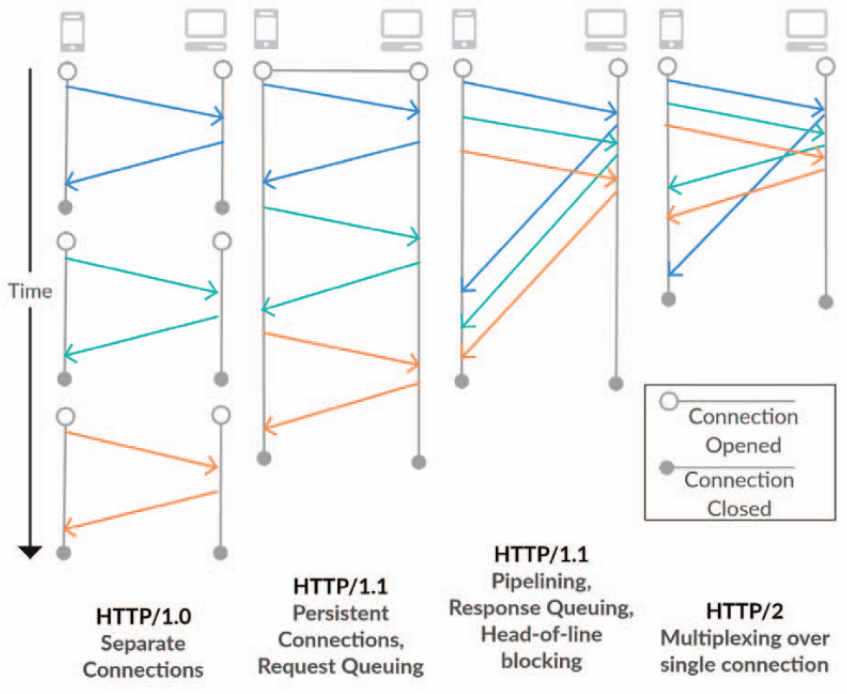

Figura 2.25: Evolução das conexões dentre as versões de HTTP (22).

O *Constrained Application Protocol* (CoAP) é um protocolo mais recente, surgido em 2010, onde o foco é a troca de dados para dispositivos com restrição de banda e processamento, suporta o modelo de *request/response* entre as aplicações e os *endpoints*, suportando autodescoberta de serviços e recursos, conceitos de URI e uma excelente integração com HTTP. O protocolo é baseado em conceitos REST - *Representational State Transfer* e tem um tamanho de cabeçalho (extra ao conteúdo) de 4 Bytes, com foco em transferir pequenas mensagens que não gerem fragmentação na rede. Operando sob UDP, ou seja sem a necessidade da sincronia de conexão e mais atento ao padrão de *request/response* , também tem redução de tempo e consumo em relação ao TCP, uma visão de uma conexão e troca de dados entre o terminal e o servidor podem ser vistas na figura 2.26, onde estão demonstradas tanto a possibilidade de receber

um aceite e resposta, quanto a possibilidade de somente enviar a mensagem, na figura 2.27 é possível perceber a simplicidade da integração anteriormente citada com o HTTP dentro de um ambiente maior de interconectividade (110) (23) (39).

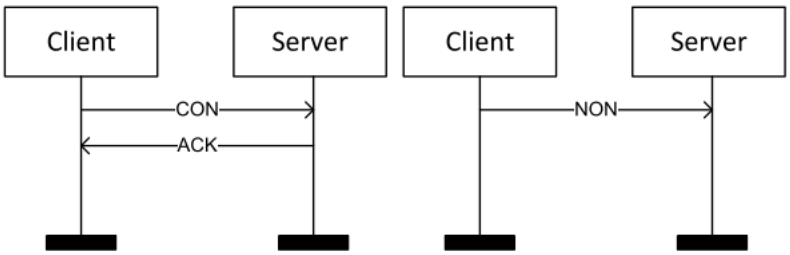

Figura 2.26: Visão de uma comunicação CoAP (12).

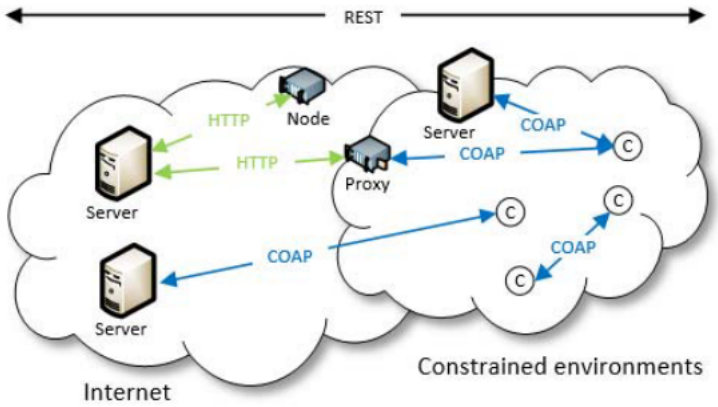

Figura 2.27: Visão de uma integração de comunicação entre CoAP e HTTP (12).

O *Message Queue Telemetry Transport Protocol* (MQTT) teve sua primeira versão em 1999, desenvolvido especificamente para troca de dados em comunicações Máquina à Máquina (M2M). O MQTT opera no modelo de *Publish - Subscriber*, ou seja, um dispositivo envia em um tópico e os outros que estão dispostos a receber devem estar inscritos no tópico específico, podendo os clientes publicarem em vários tópicos e se inscreverem em vários tópicos. O principal atrativo neste protocolo é o seu consumo extra além da mensagem que é de somente 2 bytes, com *payloads* (conteúdo) de até 256MB e três níveis de qualidade de serviço garantindo um baixo tráfego de mensagens. Muito utilizado em soluções de ponta e tido como o protocolo da Internet das Coisas, é padronizado e amplamente utilizado em serviços públicos de nuvem para IoT. A figura 2.28 demonstra uma comunicação entre dispositivos e um *broker* onde pode se ver claramente o funcionamento do protocolo MQTT (61) (23) (111) (39).

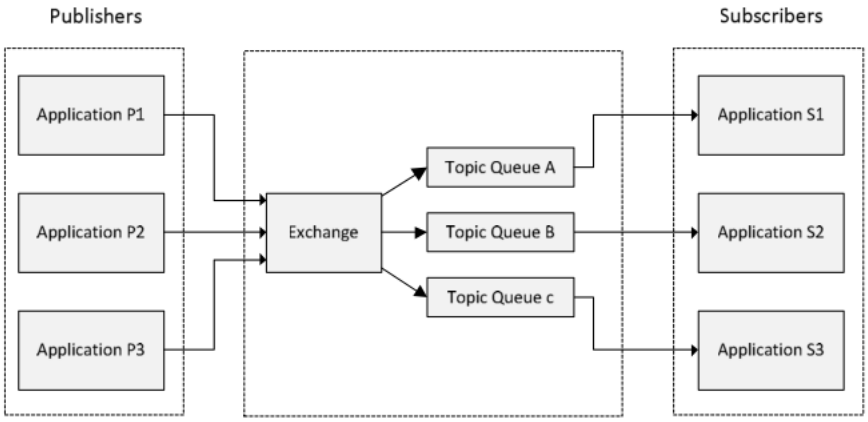

Figura 2.29: Visão de do funcionamento do AMQP (23).

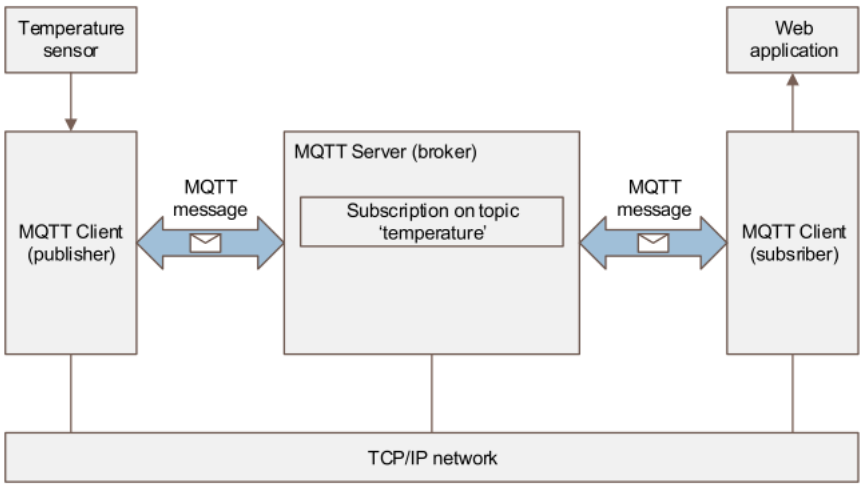

Figura 2.28: Visão de uma comunicação MQTT entre dispositivos e Broker (12).

O *Advanced Message Queuing Protocol* (AMQP) foi criado em 2003 com foco em troca de mensagens corporativas, desenvolvido dentro de um operador de mercado financeiro, a JPMorgan Chase em Londres, o protocolo foi projetado para que tivesse confiabilidade, segurança e interoperabilidade, suportando as operações tanto com requisição e resposta, quanto com publicação e assinatura (*publish/subscribe*), além dos modos de operação com enfileiramento confiável, roteamento de assinaturas baseadas em tópicos também fazem parte das funcionalidades. A operação é feita em cima de TCP com um cabeçalho fixo de 8 bytes, é utilizado em soluções de Internet das Coisas, principalmente em integrações entre *edge computing* e nuvem, podendo por vezes ser um *Middleware*, usa três níveis de qualidade de serviço e trabalha tanto com mensagens não confiáveis, ou de formato não estabelecido, quando com mensagens confiáveis de formato definido. Por ser um protocolo com muito poder de processamento, haja visto os modos de assinatura e enfileiramento, além dos controles de correção e segurança, o AMQP requer maior poder de processamento, energia e memória, por isso sua aplicação é mais vista em dispositivos menos restritos nestes quesitos. Uma visão do funcionamento pode ser vista na figura 2.29 (61) (23) (112) (39).

2.4.6.1 Ressalva quanto a cobertura da pesquisa de protocolos

Cabe uma ressalva nesta seção, onde foram citados e discutidos os principais protocolos encontrados na pesquisa exploratória sobre o tema, a atualização nesta lista é muito dinâmica, portanto esta pesquisa pode se tornar obsoleta. Uma visão bem abrangente sobre protocolos não citados aqui neste trabalho e novos pode ser vista na pesquisa de levantamento sobre IoT feita por W. Kassab e K. Darabkh (113) que faz um comparativo de pesquisas sobre tecnologias empregadas em Internet das Coisas, arquiteturas, protocolos e principalmente compara abordagens e abrangência destas pesquisas, um ponto alto deste trabalho é a citação de protocolos futuros.

Tabela 2.8: Análise Comparativa dos Protocolos de Mensagem para Sistemas IoT: MQTT, CoAP, AMQP and HTTP. Adaptado de (39).

Característica	**MQTT**	**CoAP**	**AMQP**	**HTTP**
Ano	1999	2010	2003	1997
Arquitetura	Client/Broker	Client/Serve ou Client/Broker	Client/Broker ou Client/Server	Client/Server
Abstração	Publish/Subscribe	Request/Response ou Publish/Subscribe	Request/Response ou Publish/Subscribe	Request/Response
Tamanho do cabeçalho	2 Byte	4Byte	8 Byte	Indefinido
Tamanho da mensagem	Pequeno e Indefinido (até 256 MB de tamanho máximo)	Pequeno e Indefinido (geralmente pequeno para ajustar a um único diagrama IP)	Negociável e Indefinido	Grande e Indefinido (depende do web server ou da tecnologia de programação)
Semântica/ Métodos	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Put, Delete	Consume, Deliver, Publish, Get, Select, Ack,Delete, Nack, Recover, Reject, Open, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete
Suporte para Cache e Proxy	Parcial	Sim	Sim	Sim
Qualidade de Serviço (QoS)/ Confiabilidade	QoS 0 - No máximo uma vez (Fire-and-Forget) QoS 1 - No máximo uma vez QoS 2 - Exatamente uma vez	Mensagem Confirmável (semelhante a No máximo uma vez)ou Não Verificável (semelhante a Pelo menos uma vez)	Settle Format (semelhante a No máximo uma vez) ou Unsettle Format (semelhante a Pelo menos uma vez)	Limitada (via Protocolo TCP)
Padrões	OASIS, Eclipse Foundations	IETF, Eclipse Foundation	OASIS, ISO/IEC	IETF and W3C
Protocolo de Transporte	TCP (MQTT-SN pode usar UDP)	UDP, SCTP	TCP, SCTP	TCP
Segurança	TLS/SSL	DTLS, IPSec	TLS/SSL, IPSec, SASL	TLS/SSL
Porta Padrão	1883/ 8883 (TLS/SSL)	5683 (UDP Port)/ 5684 (DLTS)	5671 (TLS/SSL), 5672	80/ 443 (TLS/SSL)
Formato de Codificação	Binário	Binário	Binário	Texto
Modelo de Licenciamento	*Open Source*	*Open Source*	*Open Source*	*Free*
Suporte Organizacional	IBM, Facebook, Eurotech, Cisco, Red, Halt, Software AG,Tibco, ITSO, M2Mi, Amazon Web Services (AWS), InduSoft, Fiorano	Large Web Community, Support,Cisco, Contiki, Erika, IoTvity	Microsoft , JP Morgan, Bank of America, Barclays, Goldman Sachs, Credit Suisse	Global Web Protocol Standard

2.5 ARMAZENAMENTO, PROCESSAMENTO E ANÁLISES

Medvedev et al. (114) em sua pesquisa sobre ingestão de dados nas plataformas de IoT levantou que em 2017 existiam mais de 300 plataformas em que poderia se ingerir e armazenar dados de Internet das Coisas. Uma plataforma de IoT é composta por um conjunto de softwares fornecidos de forma conjunta como serviço, sendo um ou mais serviços. Dentre estes podemos ter modelos de *Software as Service* (SaaS) ou *Platform as a Service* (PaaS), no modelo de SaaS o fornecimento é feito baseado em dispositivo

conectado e em alguns casos com custos de tráfego de dados cobrado à parte, no modelo de plataforma diversos serviços são ofertados e pagos por uso de acordo com a arquitetura definida pelo usuário/client (115) (116). Estas plataformas, indiferente do modelo de oferta, estão facilitando em muito a adoção e são habilitadoras para a Internet das Coisas, uma vez que a complexidade de criação da infraestrutura básica para a operação está pronta e validada por diversos cenários de uso. Um dos desafios apontados no estudo de Medvedev et al. foi a questão de se analisar a performance das plataformas, pois dentro de algumas existe uma miríade de serviços, alguns que podem ter a mesma funcionalidade apresentada de forma diferente, um exemplo de armazenamento e análise pode ser visto na figura 2.30.

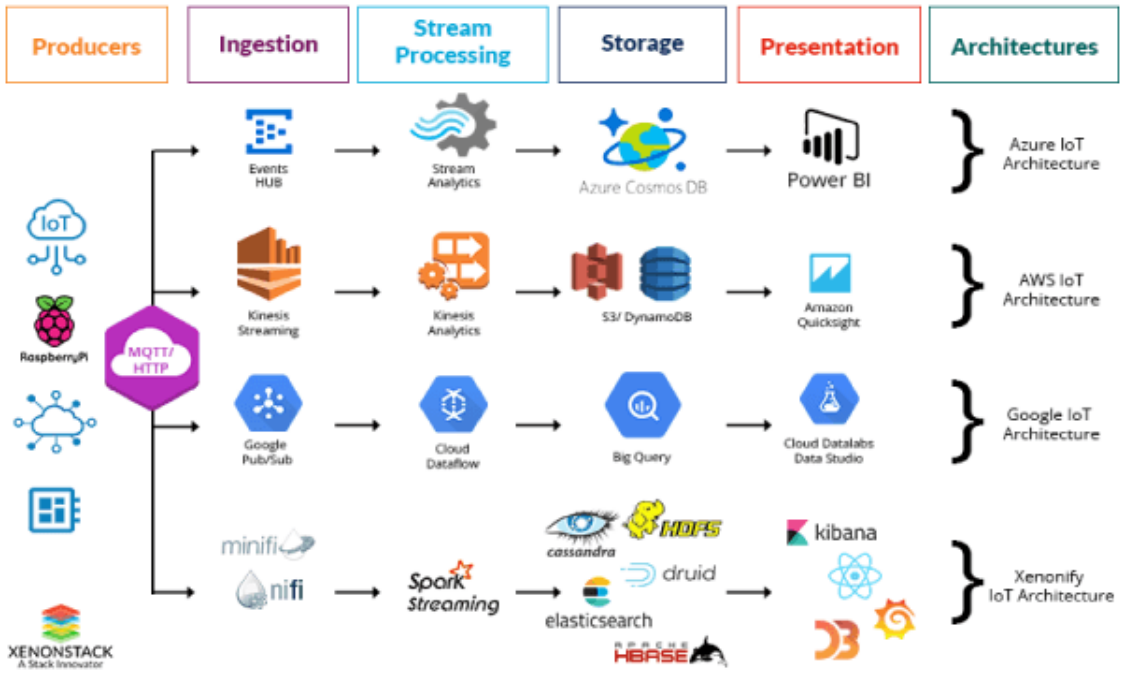

Figura 2.30: Um exemplo de ferramentas e do processo de ingestão de dados e análise em IoT (24).

Indiferente da plataforma ou da infraestrutura um dos pontos de atenção e necessidade depois da coleta dos dados pelo dispositivo e comunicação do dispositivo com a nuvem ou servidores privados é o ciclo de ingestão de dados e análise, bem como a inteligência aplicada ao processamento das informações para as tomadas de decisão. Indiscutivelmente a geração de dados vem crescendo a medida que mais dispositivos vem sendo conectados e com a melhoria das tecnologias e o barateamento das tecnologias, mais dados são coletados, as vezes em sistemas que já existiam, aumentando ainda mais a necessidade de armazenamento, processamento e análise destas informações. Uma loja americana de varejo, o Target, é capaz de inferir se uma mulher está grávida pela análise de seus dados históricos de compra, e isto vem se tornando uma crescente nos negócios cada vez mais conectados, agora contando também com sensores e análises em tempo quase real. Os dados gerados pela IoT podem ser redundantes, com formatos diferentes de acordo com os sensores ou concentradores, mesmo que referentes a mesma informação e podem conter anomalias oriundas de problemas de comunicação, fraude ou falha de coleta, estas características aliadas ao grande volume caracterizam o conceito de BIG DATA, que segundo alguns autores é composto por 4 Vs, que são respectivamente: Volume, Variedade, Velocidade e Veracidade. Outros autores removem a Veracidade e tratam o a definição com três Vs somente (117) (25).

Em sistemas tradicionais quando a necessidade de armazenar dados surge, bancos de dados relacionais são a escolha mais adotada, em Internet das Coisas a necessidade de uma escala horizontal onde a demanda de um acesso massivo tem que ser atendida não é bem tratada atendida por bases relacionais, além do tipo de dado ter características dinâmicas, com mudanças de *schema* constantes levam a escolha de bancos de dados NoSQL como uma escolha mais direta para armazenamento, dentro deste cenário temos modelos de dados do tipo chave valor, documentos, colunar e grafos, além de séries temporais. Uma outra forma de armazenar dados para IoT e BIG DATA é o uso de sistemas de arquivos para posterior processamento em lotes ou em processadores de dados de *Stream*. A escolha do armazenamento não se dá de forma única em um projeto de IoT, podemos usar mais de um modelo, mais de uma forma e muitos serviços diferentes ao mesmo tempo em etapas diferentes, não descartando nenhum, nem mesmo o relacional. O que importa é unir as tecnologias a fim de garantir a velocidade necessária, com integridade durante todo o processo (118) (117) (119).

No processo da análise e inferência para tomada de decisões, em alguns casos a necessidade de respostas imediatas leva uma parte ou todo o processo para ponta, com trabalho dentro do que anteriormente definimos como Edge Computing, atualmente as tecnologias de armazenagem e processamento de dados já estão funcionando neste modelo e com isso garantem o funcionamento não somente com a velocidade necessária, mas também podem habilitar soluções que dependam de uma segurança maior quanto ao tráfego de dados ou também solucionar a disponibilidade da solução no que tange quedas de conexão. Este modelo pode também ser usado de forma híbrida onde pode-se usar tanto a ponta quanto a nuvem com sincronia e balanceamento de carga de trabalho, por algumas vezes diminuindo o custo de processamento na nuvem o que pode ser mais um ponto positivo para implantação da solução, um exemplo pode ser visto na figura 2.31. Além da ponta, dados podem ser armazenados e processados também no meio do caminho, ou na *FOG computing* (25) (84) (85).

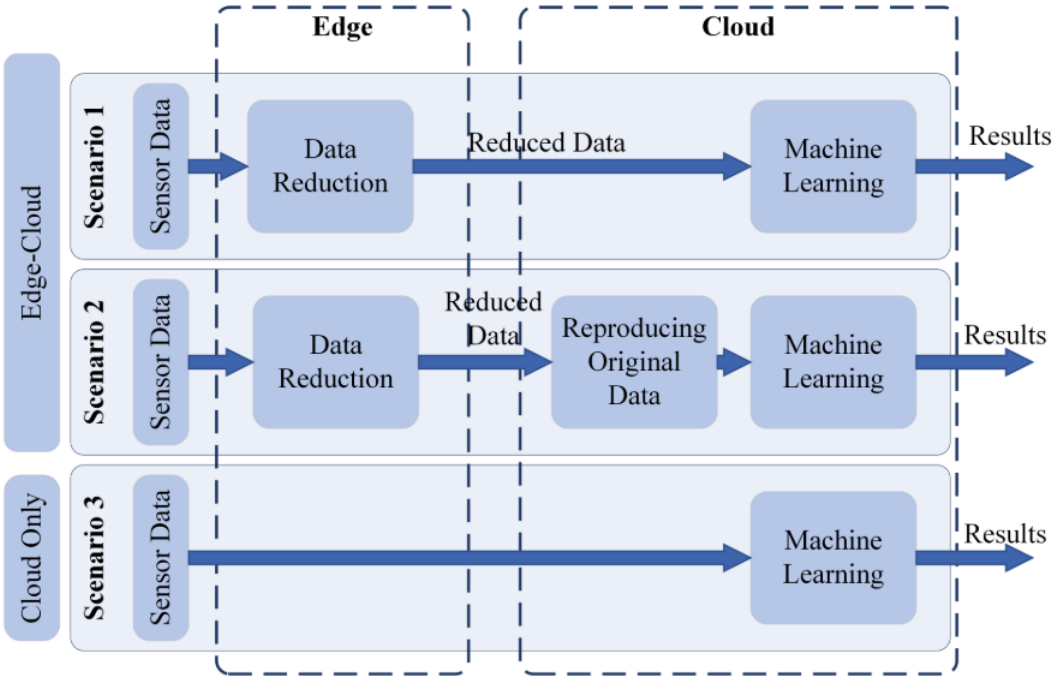

Figura 2.31: Um exemplo de redução de dados na ponta para processamento na nuvem (25).

2.6 ARQUITETURAS

Um dos requisitos de uma solução de IoT é que as coisas, ou os dispositivos estejam interconectados, uma arquitetura deve garantir as operações entre os mundos físico e virtual combinando dezenas de tecnologias, garantindo que os dados estejam íntegros, os dispositivos seguros, possibilitar o crescimento da rede, se adequar a mudanças no campo, adequar-se a novos modelos de negócios que venham a surgir dentre outros. A princípio uma arquitetura deve ser expansível, escalável, segura, integrar-se a outras e ser aderente a implantação dentro de custos e qualidade esperados (29) (120) (10) (10) (13).

2.6.1 Arquitetura em 3 camadas

Esta é uma arquitetura que não traz uma rigidez na sua definição e pela sua maleabilidade torna fácil a adoção, uma visão das camadas pode ser vista na figura 2.32. Ao mesmo tempo pode se tornar complexa em projetos de maior complexidade, exigindo um conhecimento melhor das tecnologias e suas integrações, bem como uma visão mais ampla das suas utilizações, tal maleabilidade faz com que ela também seja excelente para a extensibilidade, pois pode conter dentro de suas camadas objetos de acordo com a definição do projeto, W. Fang et al. (13) em seu trabalho descreveu da seguinte forma as camadas:

- Perception - Responsável por reunir informações de sensores e fazer a identificação destes objetos. Aqui entram etiquetas RFID, leitores de códigos de barras, câmeras, GPS, sensores, terminais e redes de sensores.

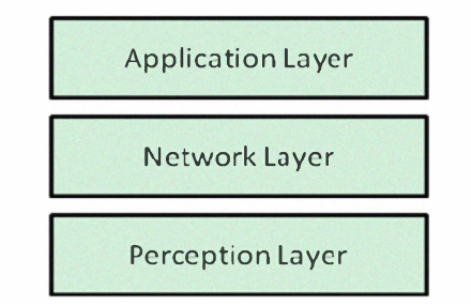

Figura 2.32: Arquitetura IoT de três camadas. Adaptada de (13).

- Network - O backbone entre o mundo físico e a aplicação, tem como principal função processar as informações e transmitir dados, incluindo a convergência de redes de comunicações privadas e públicas agindo como centro de processamento.
- Application - Esta camada traz a união dos anseios da indústria com a necessidade da Internet das Coisas para abrigar as aplicações e a interface com negócios e sistemas, bem como expor a gestão e informações para usuários finais, uma camada onde se explicita a personalização para a especificidade da aplicação de IoT.

A arquitetura de 3 camadas é amplamente citada nas pesquisas (76) (26) (13) (115) (31), está presente nas implementações de provedores de nuvem como Google, Microsoft, Amazon e IBM dentre outros. As 3 camadas também fazem parte da visão do Industrial Internet Consortium para sua arquitetura de referência (26), atualmente em sua versão 1.9 e que pode ser vista em uma visão expandida para contemplar as integrações com os diversos domínios funcionais na figura 2.33.

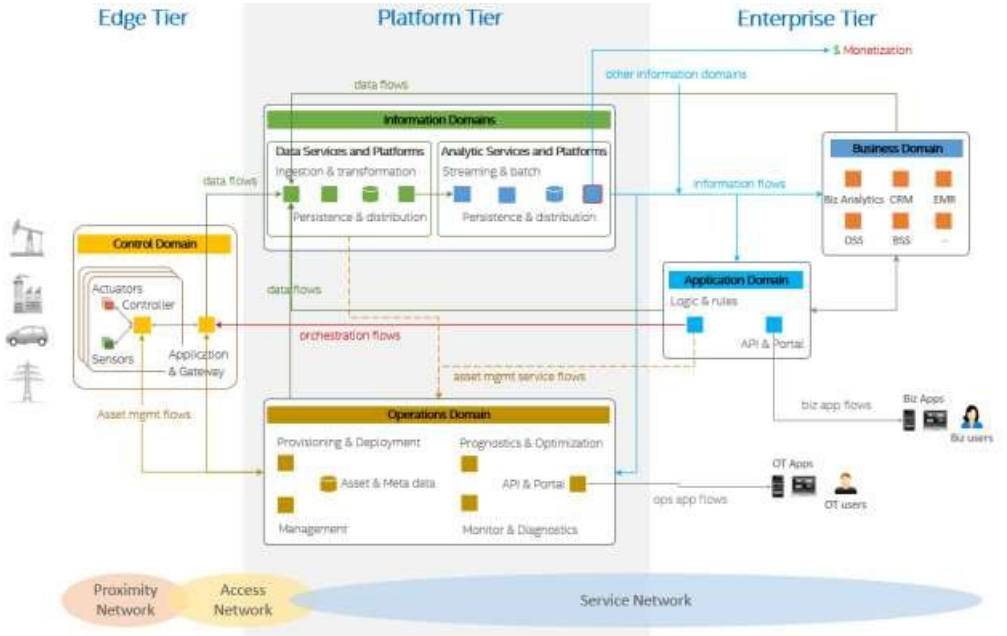

Figura 2.33: Industrial Internet Consortium - Arquitetura IoT com visão expandida para contemplar as integrações com os diversos domínios funcionais (26).

2.6.2 Arquitetura de 5 camadas

A Arquitetura de 5 camadas, figura 2.34, foi proposta por Chang-le Zhong, Zhen Zhu e Ren-gen Huang (27) em 2015 de acordo com a visão deles sobre as características e necessidades da Internet das Coisas, abaixo temos as características de cada camada:

- *Perception* - A principal função desta camada é a informação, coleta de dados e quando necessário ajudar no controle dos objetos desta camada, ela é a interface inicial entre mundo físico e a solução de IoT que utiliza esta arquitetura, composta por sensores, câmeras, RFID tags e leitores.
- *Network Access* - Com a função de ligar a camada dos sensores, tem a ação de um *gateway*, complementando a rede dos sensores, se utilizando de WIFI, Zigbee, redes *mesh* e outros para integrar e iniciar o processo de subida dos dados e coordenar a comunicação dos sensores.
- *Network Transmission* - Esta camada faz a comunicação entre aplicação e sensores usando métodos de longa distância integrando a comunicação privada com a internet.
- *Application Support* - Nesta camada entram serviços de nuvem, sistemas especialistas e processamento das informações, bem como gestão do sistema como um todo.
- *Application Presentation* - A funcionalidade desta camada é unir a aplicação a uma interface de usuário por meio de apresentação em sistemas existentes, relatórios, multimídia, realidade virtual dentre outras formas.

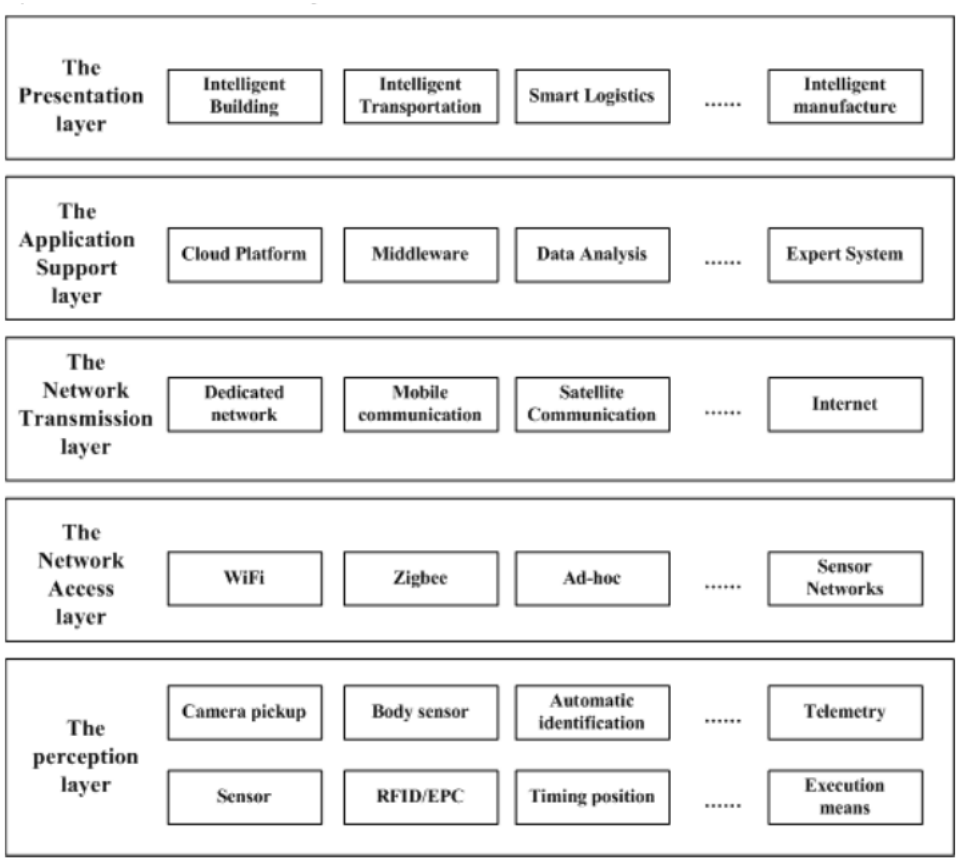

Figura 2.34: Arquitetura de 5 camadas (27).

2.6.3 Arquitetura de IoT da CCSA

Em meados de 2010 a China iniciou a padronização da Internet das Coisas, mesmo com o conhecimento de que a complexidade deste processo seja muito grande, com a intenção de criar regras no uso de protocolos, hardware de dispositivos, interconectividade, uso de redes de telecomunicação e tudo mais que seja insumo, produto ou processo no desenvolvimento de soluções de Internet das Coisas. Neste trabalho, muitas instituições de pesquisa foram envolvidas, bem como empresas também foram convocadas. Chen, Liu et al. fizeram um levantamento da visão da china sobre a Internet das Coisas (28), neste trabalho um dos pontos apresentados foi uma arquitetura aberta e genérica para a Internet das Coisas, dentro da criação de um padrão para IoT (IoT *Standard*), que considerasse diversos ramos de negócios e uma interoperabilidade grande, uma vez que muito era feito dentro de domínios específicos e com foco na solução. Uma arquitetura única é muito melhor segundo levantado por eles, pois diminuiria a integração destas aplicações com outras, uma vez que cada solução de IoT pode usar um protocolo diferente dentre outras particularidades. Dessa problemática surgiu a ideia de ser criado um padrão para uso em todas as áreas, tendo como um bom argumento a redução de tempo de desenvolvimento e custos. Uma arquitetura aberta e genérica deve trazer o conceito de interoperabilidade na sua essência e ter as seguintes características: As interfaces e protocolos devem padronizados baseando-se em sistemas que já tiveram êxito, facilitando assim a operação, que deve ser aberta e pública com isso permitindo todas as outras de nichos específicos a operarem com ela e por fim deve ser aberta, escalável e flexível com padrões claros e abertos, se adaptando

a diferentes necessidades dos negócios e sem impacto a performance por causa disso.

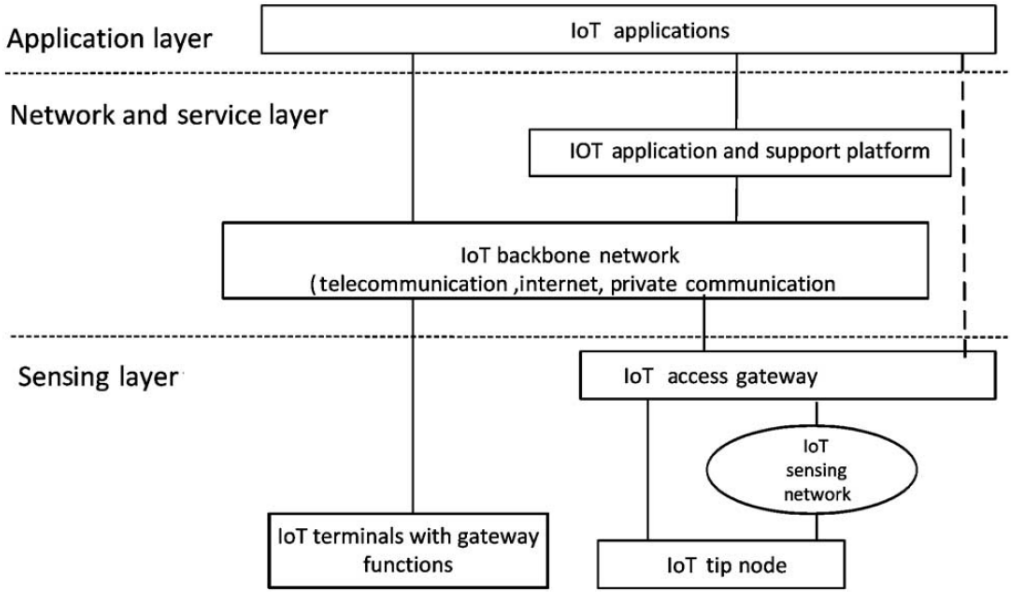

Figura 2.35: Arquitetura proposta pela CCSA (28).

A *China Communications Standards Association* (CCSA) propôs um modelo de referência para Internet das Coisas que consiste em 3 camadas, sendo uma para sensores, outra para comunicação e serviços e a de aplicação, conforme pode ser visto na figura 2.35. A arquitetura contempla três plataformas de funcionalidades:

- *Sensing and Gateway Platform* - Responsável pela conectividade com sensores, controles, leitores e posicionamento que devem ser enviados a camada de rede. Uma modularização do hardware, formato dos dados, interface de software foi desenhada para o *gateway* ou dispositivo, sendo que os dispositivos ou *gateway* podem conter seus próprios módulos. Redes de sensores são conectadas por um *gateway* e passam por uma interface comum que é responsável pela gestão dos sensores, uma outra característica desta plataforma é a autoconfiguração dos sensores e sua adaptação ao meio.

- *Resource and Administration Platform* - A camada de serviço e rede é a responsável pelo *backbone*, recursos de administração e outros relativos à comunicação. Esta é a camada onde entram as camadas físicas de interconexão de rede como 3G, fibra, Ethernet, satélite e outras privadas ou públicas. A administração e gestão nesta camada são capazes também de gerir segurança, *storage*, análises, além de gerir a própria rede de comunicação, autenticação de dispositivos, gestão de usuários, desenvolvedores e etc.

- *Open Application Platform* - Com base no desenvolvimento em módulos esta camada de aplicação tem funções comuns e baseadas em *application programming interface* (API). O desenvolvedor poderá desenvolver sua aplicação de IoT baseado nestas APIs, com controle e gestão das outras camadas por meio delas implementando um ambiente flexível, expansível e de fácil desenvolvimento.

A figura 2.36 que demonstra a aplicação para uma solução de mina de carvão, onde sistemas existentes de sensoriamento são colocados na camada de base e coletam informações em tempo real, conectados por *Ethernet* para superfície, garantindo a tomada de decisão dentro das minas, na figura 2.36 uma solução de campo de extração de petróleo inteligente captura dados em tempo real e provê a tomada de decisão garantindo a melhoria da extração.

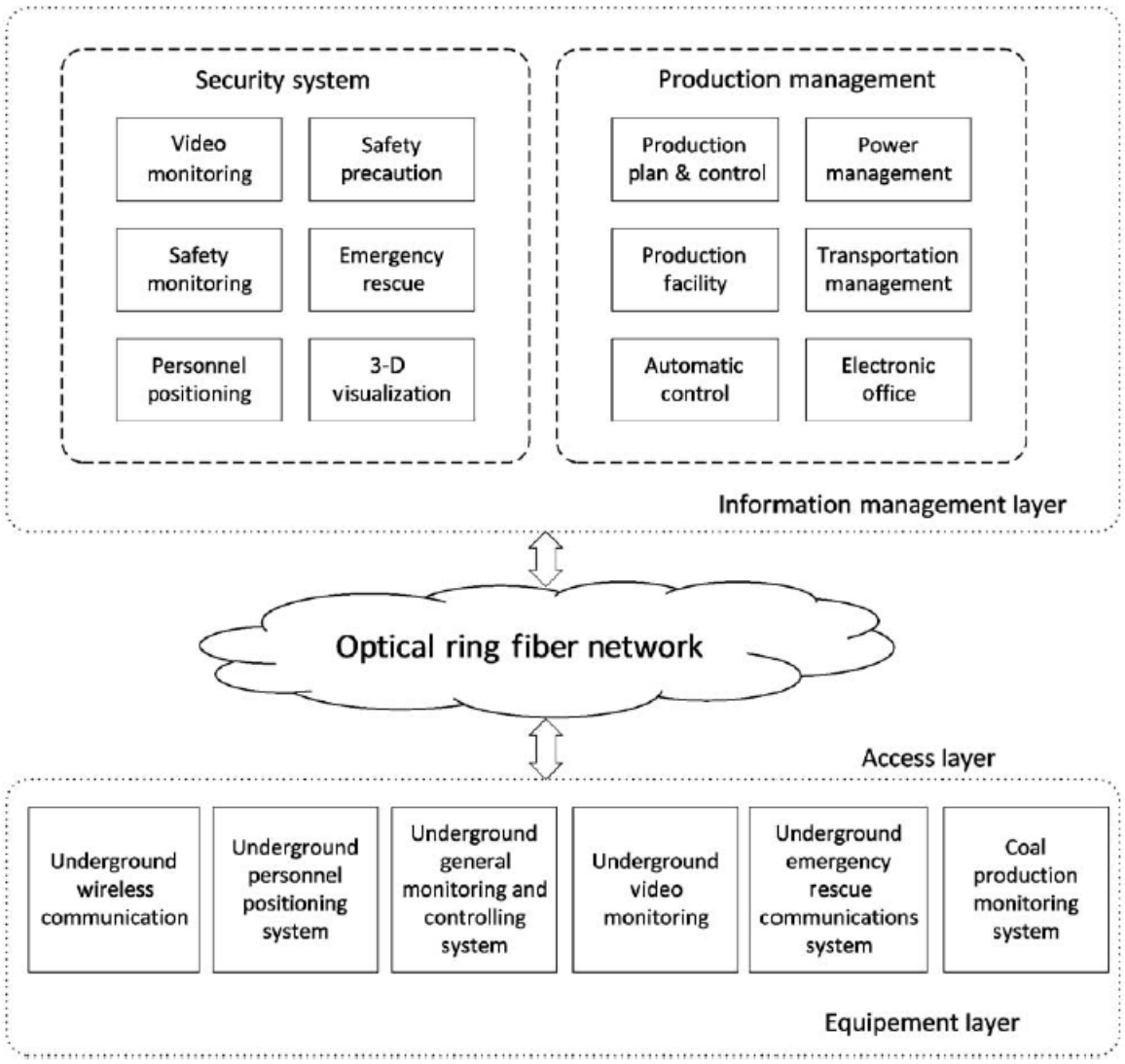

Figura 2.36: Aplicação de IoT para mina de carvão usando a Arquitetura Aberta da China (28).

2.6.4 Arquitetura Orientada a Serviços (SOA)

A Arquitetura orientada a Serviços tem quatro camadas e define sua base em objetos bem definidos que podem ser mantidos de forma separada provendo modularidade, uma visão geral pode ser vista na figura 2.38 (10) (29). As camadas são as seguintes (29):

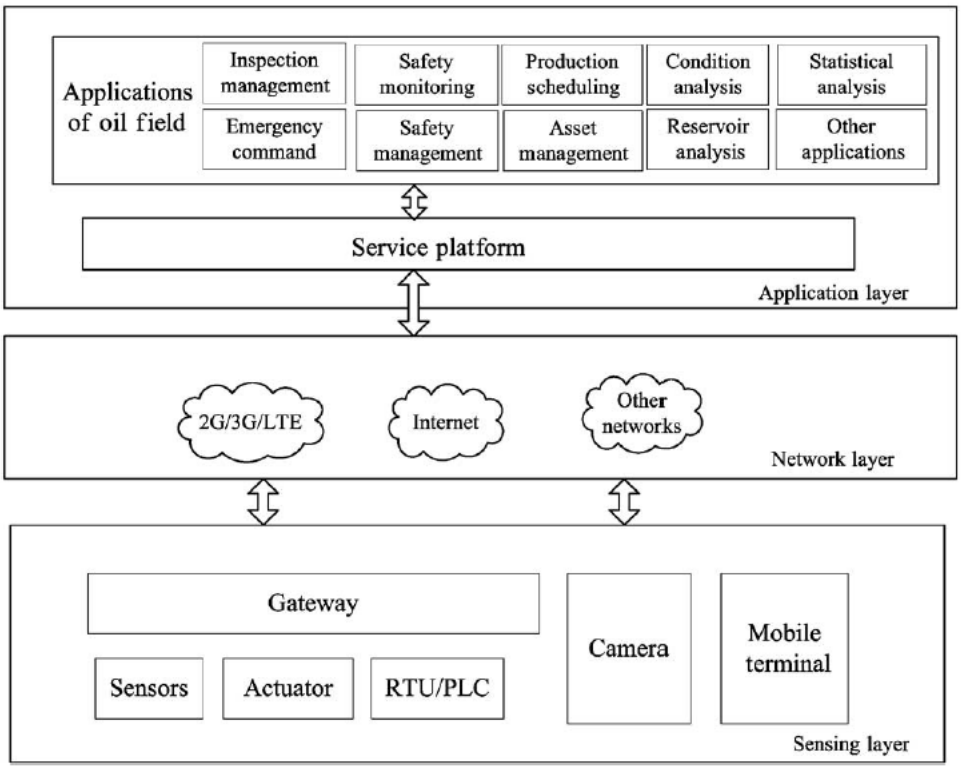

Figura 2.37: Aplicação de IoT para Campo de extração de petróleo usando a Arquitetura Aberta da China (28).

- Sensing Layer - Camada responsável pelos sensores e atuadores com a habilidade de troca de dados entre os dispositivos. Todo objeto nesta camada deve ter um identificador único, *universal unique identifier* (UUID) e os equipamentos nesta camada devem se preocupar com uso racional de recursos como bateria e rede por exemplo. A inserção de novos sensores deve ser feita de forma simples uma vez ou mais de uma vez, permitindo que características diferentes possam ser aplicadas, se comunicando e permitindo também topologias diferentes para as redes de sensores.

- Network Layer - Camada responsável pela infraestrutura de conectividade cabeada ou sem fio. Requisitos de qualidade de serviço, gestão da rede, tecnologias de pesquisa de dispositivos, segurança e privacidade, processamento de dados estão dentro desta camada.

- Service Layer - Camada responsável pelos serviços requeridos por usuários e aplicações. Esta é a camada de meio ou elo de ligação que habilita os serviços, busca informações para a camada de aplicação, ontologias e armazenamento de dados também.

- Interfaces Layer - Camada de métodos de interação entre usuários ou aplicações. Esta camada garante que um mesmo sensor, fornecido por diferentes fabricantes com dados em formatos diferentes sejam consultados da mesma forma, garantindo compatibilidade e intercâmbio de informações indiferente de detalhes de implantação na ponta.

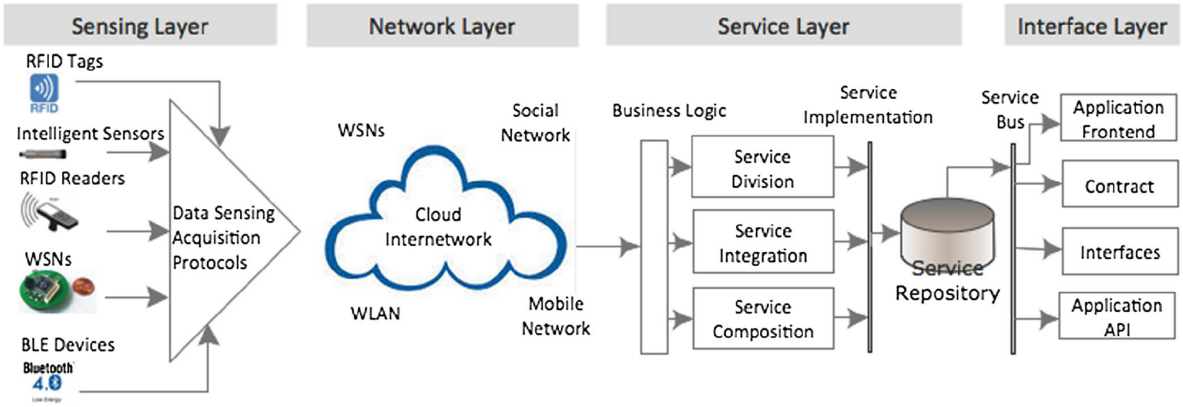

Figura 2.38: Arquitetura Orientada a Serviços (SOA) (29).

2.6.5 Arquitetura de um Gateway IoT

A arquitetura interna de um Gateway IoT é um pouco diferente de como é conhecido para outras utilizações de conectividade, pois nele podemos incluir uma camada de percepção, a qual interliga sensores ou redes de sensores a rede, além de também ter as funcionalidades normais de transporte de dados e interligação. Outra parte da arquitetura que é levemente modificada é a da camada de aplicação, que pode por vezes fazer a parte de concentrar, armazenar, rotear, unir, processar e aplicar inteligência aos dados coletados dos sensores (27) (121).

Nesta arquitetura temos uma camada de hardware composto por processadores ou microcontroladores e módulos de conectividade tanto interna quanto externa, o que definirá o hardware é a aplicação, quanto ao sistema operacional também é definido pela aplicação, onde aplicações simples podem ser regidas por um RTOS (sistema operacional de tempo real), mas em casos onde precisa-se de maior poder na aplicação um sistema operacional de base Linux pode ser mais indicado, faz parte também uma camada de abstração de hardware (HAL) para permitir a portabilidade do software sem o redesenho do hardware. Existe também uma necessidade de suportar os sensores e suas pilhas de execução e dos protocolos, bem como a gestão dos dispositivos e a segurança tanto de controle de perímetro do equipamento, quanto dos sensores, quanto da comunicação e apoio a gestão da informação. Um dos pontos importantes em um dispositivo de IoT, neste caso o *gateway* é no que regula sua atualização e esta deve ser feita de forma remota e sem necessidade de intervenção humana, para isso utilizam se do modelo de *Firmware Over the Air* FOTA, que traz melhorias, correções e atualizações de parâmetros e do software que roda no dispositivo. Uma visão das camadas e objetos deste desenho pode ser visto na figura 2.39 (30) (27).

2.6.6 Padronização e Comparativo dos trabalhos

Ainda muito se fala sobre as arquiteturas de Internet das Coisas, a força de uma padronização segundo alguns autores iria facilitar a adoção (122) (123), mas ainda não existe um padrão formado encontrado na pesquisa feita, uma visão comparativa pode ser vista na figura 2.40 para as arquiteturas SOA, 3 camadas, 5 Camadas e Middle-ware based, enquanto isso existem alguns projetos no Brasil (124) e no mundo que tentam criar um padrão para que se desenvolvam aplicações / soluções de IoT em cima de um desenho

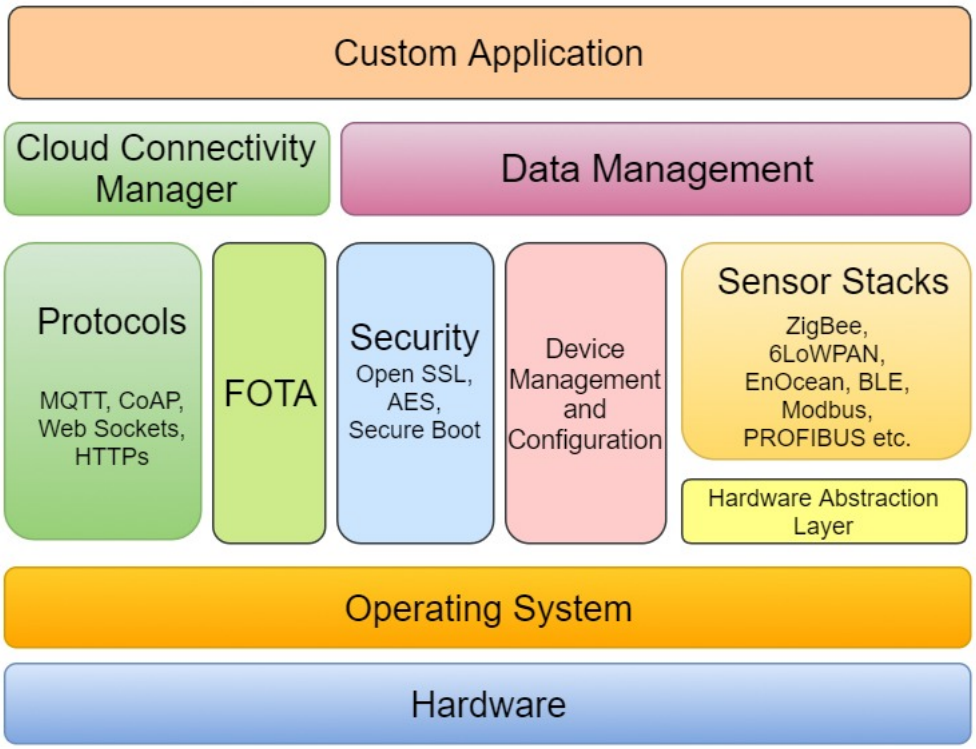

Figura 2.39: Arquitetura de gateway IoT (30).

único indiferente da aplicação, embora ainda sem muito sucesso no que tange a adoção massiva.

2.6.7 IoT como Serviço

Tem sido visto como uma prática comum o fornecimento de serviços já prontos para consumo para que somente sejam acoplados os dispositivos, tendo encapsulados os serviços e complicadores de uma camada meio de controle e gestão de dispositivo, entregando tudo como serviço, tanto no modelo SaaS quanto no modelo PaaS (116) (115) (10). As camadas de meio de conexão que interligam os sensores com a nuvem de serviços tem ganho bastante robustez (84) ultimamente, como é o caso da Microsoft com sua *framework* de *Edge Computing* e do Azure Sphere que vem facilitando relativamente a colocação de gestão na camada de ponta oferecendo a gestão dos dispositivos, segurança, sincronia e controle de sensores ligados por meio de sub conexões, tal comportamento do mercado também é visto pela Amazon (85).

Ainda não existe um serviço que englobe todas as camadas para desenvolvimento de soluções (116) (115), o que se pode achar no mercado são serviços que atendem uma determinada demanda, vendidos como uma aplicação, por exemplo, controle de irrigação de plantações, onde o serviço é entregue e pago somente por uso (125) e vinculado a um único fornecedor com uma implantação no modelo serviço, integrável aos sistemas do usuário consumidor.

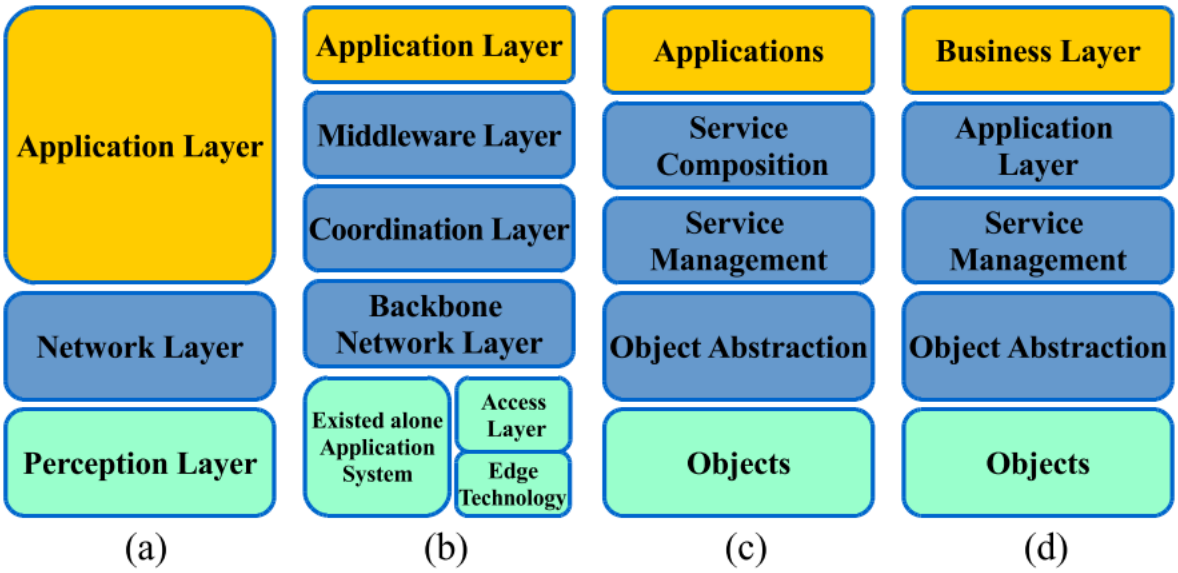

Figura 2.40: Relação entre camadas de algumas arquiteturas de IoT: (a) 3 camadas, (b) Middle-Ware based, (c) SOA, (d) 5 Camadas (31).

2.7 SEGURANÇA

A Internet das Coisas, conforme já visto anteriormente neste capítulo, é composta pela união de diversas tecnologias, protocolos e formatos de arquiteturas, onde podemos ter um uma solução dezenas de pontos de problema em relação a segurança. Cada protocolo citado opera em cima de uma camada ou uma tecnologia de rádio e tem suas limitações no que tange segurança, seja por falta de funcionalidades, seja por falta de recursos nos dispositivos para sua implantação (126) (105) (70). Uma visão de segurança deve ser aplicada a cada camada da arquitetura e no dispositivo, pois temos brechas de segurança e problemas que podem ser criados desde o hardware ao aplicativo de visualização (127) (40). Os ataques podem ocorrer diretamente em uma camada explorando uma brecha de segurança ou uma limitação, como também ocorrem de forma massiva a múltiplas camadas (126), uma visão da Microsoft (32) como atuar com os possíveis pontos de ataque pode ser visto na figura 2.41, onde são mapeadas zonas que dividem os componentes dentro da arquitetura para que um modelo de mitigação de problemas de segurança seja pensado e com isso uma otimização do uso das tecnologias com uma visão de segurança seja aplicada.

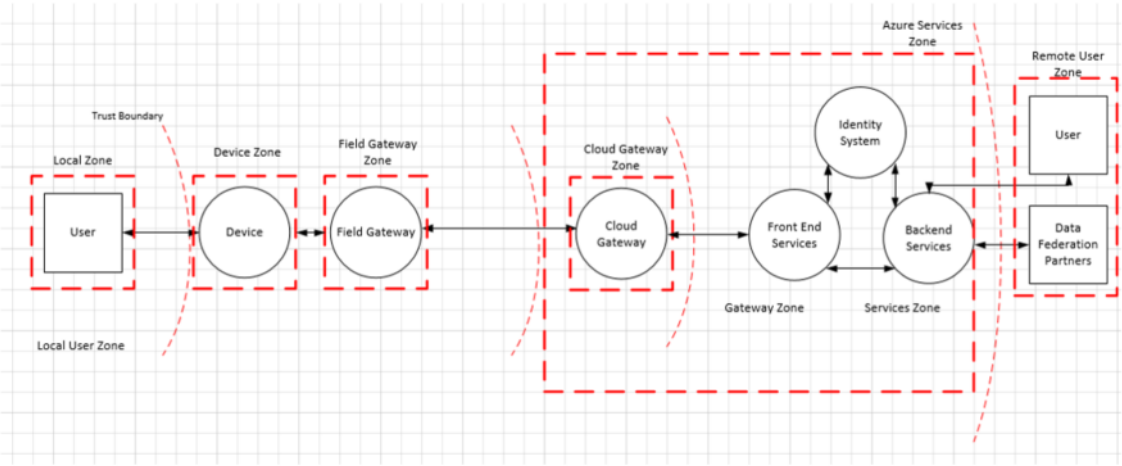

Figura 2.41: Proposta de segurança da Microsoft, separando a solução de IoT em zonas com os limites de confiança (32).

Dois pontos primordiais para um dispositivo de IoT são a energia e a segurança (70), Segundo Galen Hunt et al (40) um dispositivo altamente seguro deve ter sete propriedades implementadas e esta será uma premissa em um futuro próximo para todos os dispositivos conectados, uma vez que uma brecha de segurança em um brinquedo por exemplo, hoje temos dezenas deles conectados, pode expor toda uma rede local ou até mesmo o meio físico de acordo com o que ele tenha de sensores, uma câmera e um microfone por exemplo poderiam dar ao intruso uma observação total do ambiente. As sete propriedades que eles citam em sua proposta de dispositivo podem ser vistas na tabela 2.9, com os respectivos exemplos de problemas.

Tabela 2.9: Propriedades exigidas de dispositivos altamente seguros com exemplos. Adaptado de (40).

Propriedade	**Exemplos e Questões para Provar a Propriedade**
Hardwate-based Root of Trust / Raiz de Confiança Baseada em Hardware	Chaves criptográficas não falsificáveis geradas e protegidas por hardware. As contramedidas físicas resistem a ataques de canal lateral.
	O dispositivo tem uma identidade única e não falsificável do hardware?
Small Trusted Computing Base / Pequena Base de Baseada em Hardware	Chaves privadas armazenadas em um cofre protegido por hardware, inacessível ao software. Divisão do software em camadas de autoproteção.
	A maior parte do software do dispositivo está fora da base de computação confiável do dispositivo?
Defense in Depth / Defesa em Profundidade	Múltiplas mitigações aplicadas a cada ameaça. As contramedidas atenuam as consequências de um ataque bem-sucedido em qualquer vetor
	O dispositivo ainda está protegido se a segurança de uma camada do software do dispositivo for violada?
	Continua na próxima página

Propriedade	**Exemplos e Questões para Provar a Propriedade**
Compartmentalization / Compartimentalização	Barreiras impostas por hardware entre componentes de software evitam que uma violação em um se propague para outros.
	Uma falha em um componente do dispositivo exige a reinicialização de todo o dispositivo para retornar à operação?
Certificate-based Authentication / Autenticação Baseada em Certificado	O certificado assinado, comprovado por chave criptográfica não comprovável, prova a identidade e autenticidade do dispositivo.
	O dispositivo usa certificados em vez de senhas para autenticação?
Renewable Security / Segurança Renovável	A renovação traz o dispositivo para um estado seguro e revoga os ativos comprometidos por vulnerabilidades conhecidas ou violações de segurança
	O software do dispositivo é atualizado automaticamente?
Failure Reporting / Relatório de Falhas	Uma falha de software, como uma saturação de buffer induzida por um invasor investigando a segurança, é relatada ao sistema de análise de falha baseado em nuvem.
	O dispositivo relata falhas ao fabricante?

A tabela 2.10 traz um compilado feito por Veronica Valeros (41) apresentado em abril de 2020 com as variações da *Malwares* com foco em Internet das Coisas. A pesquisa foi feita com uso de técnicas de *Open-source intelligence* (OSINT), ou seja, com o recolhimento de dados por meio de fontes públicas abertas, e traz informações desde 2008. Um projeto abrigado no laboratório de pesquisas Strotosphere na Czech Technical University em Praga, nomeado de Aposemat (128) vem de encontro a problemática de segurança em Internet das Coisas, com o objetivo de proteger, analisar ameaças, executar e detectar ataques de *Malwares* em dispositivos.

Tabela 2.10: Primeira timeline dos IoT Malware vista, dados obtidos por correlação de informações usando técnicas OSINT (41).

Ano	Nome / Apelido
2008	Hydra
2009	Psyb0t / NetworkBluePill
2010	Chuck Norris
2011	Umbreon / Umreon / Rebonum / Neobrum
2012	Carna Botnet
2012	LightAidra / Linux Aidra
2013	Tsunami / Kaiten
2013	Linux Darlloz / Zollard
2014	Gafgyt / BASHLITE / Lizkebab / Torlus / Qbot / LizardStresser

Continua na próxima página

Ano	Nome / Apelido
2014	Spike / Dafloo / MrBlack / Wrkatk / Sotdas / AES.DDoS
2014	TheMoon
2014	Zendran
2014	Linux.Wifatch / Ifwatch / REINCARNA
2015	Linux Moose / Elan
2016	VPNfilter
2016	Mirai
2016	KTN-RM / Remaiten
2016	Hajime
2016	LUABot
2016	IRCTelnet / LinuxIRCTelnet / NewAidra
2016	NyaDrop
2017	Amnesia
2017	Linux.MulDrop.14
2017	BrickerBot
2017	Persirai
2017	Satori
2017	LinuxProxyM
2017	IoTroop / Reaper / IoTrooper
2017	Masuta
2017	GoScanSSH
2017	Okiru
2018	UPnProxy / ETERNALSILENCE
2018	DoubleDoor
2018	Hide ‘N Seek
2018	JenX / Jennifer / Jen-X
2018	Muhstik
2018	PureMasuta
2018	Torii
2019	Ares
2019	Mozi
2019	Silex
2019	Echobot
2019	Moobot
2019	Dark Nexus
2019	Handymanny
2020	Mukashi
2020	Rhombus

2.8 VISÃO GERAL DA PESQUISA FEITA SOBRE A INTERNET DAS COISAS

O Cenário de Internet das coisas tem uma diversidade de tecnologias reunidas e traz uma realidade desafiadora onde tem-se diversas demandas que trazem necessidades de integrações de tecnologias como pode ser visto nas arquiteturas apresentadas e nos desafios e aplicações de IoT. No quesito de criação de tecnologia, as pesquisas mostram que a China ainda é líder neste campo com o maior número de patentes registradas, mas todos os países estão crescendo suas bases, o Brasil como citado criou o plano nacional de IoT e com isso investimentos vem sendo feitos para trazer o país para um patamar de presença dentro deste campo.

Um dos pontos principais para IoT é comunicação, neste campo temos as comuinicações locais, as regionais e as mais amplas, cada qual com uma abordagem e uma aplicação, podendo ser combinadas para atingir um resultado. As pessoais vem amadurecendo e expandindo-se como é o caso do Bluetooth que atualmente permite comunicações de mais longo alcance e permite a aplicação em malha formando uma rede de dispositivos, os protocolos de LPWAN também trazem uma abordagem que vem de encontro ao custos de desenvolvimento, comunicação e operação. As tecnologias celular também vem ganhando novas funcionalidades com o passar do tempo, evoluindo para velocidades altas com baixa latência, bem como um maior número de dispositivos servidos com menos torres. Somam-se a estas tecnologias citadas outras como Ethernet, WIFI, 6LoWPAN e outras, o que faz com que a necessidade de atualização e pesquisa para escolha de uma tecnologia seja a cada dia mais difícil. Dentre os protocolos de aplicação o MQTT, COAP, AMQP e HTTP são vem difundidos nas pesquisas e trabalhos, e novamente temos evoluções acontecendo como é o caso do HTTP na versão 2.0 permite um uso da rede mais eficiente, reduzindo latência e permitindo uso de cabeçalhos compactados além de múltiplas trocas simultâneas na mesma conexão, com isso pode ser mais utilizado em dispositivos com restrições de comunicação. Estes são somente alguns protocolos, físicos ou de aplicação, dentro de um universo crescente de centenas de outros, estes não citados, podem ser até então menos conhecidos por terem características determinadas por sua aplicação a um determinado problema e podem ascender ao *mainstream* de acordo com uma nova necessidade que transforme sua adoção em escala global.

Nas arquiteturas, as apresentadas dividem-se em camadas e funcionalidades de objetos e percebe-se que uma grande visão em torno de tratar os componentes de forma separada já traz uma visão da necessidade de se observar a aplicação como forma de moldar os componentes. O caso da arquitetura da CCSA por exemplo forma uma base para diversas aplicações mas se especializa nos componentes, este caso também é observado nas arquiteturas de três e cinco camadas, e apresentado de forma mais direta a aplicação na arquitetura SOA. Ainda na definição de uma arquitetura temos componentes que tem sua própria arquitetura como são os casos do Edge Computing com os *gateways* que podem se transformar em componentes de FOG de acordo com a sua atuação e posicionamento, estes por sua vez trazem um subconjunto de componentes para processamento de dados que de acordo com a aplicação vão ter seus níveis de segurança e hardware também personalizados. Fora as que foram apresentadas existem variações e como este ainda é um campo relativamente novo de pesquisa muito ainda temos pela frente.

Em segurança temos novos ataques se consolidando e novas necessidades surgindo, dispositivos mais seguros e uma visão mais extensa sob este quesito vem tomando forma conforme apresentado neste capítulo, mas ainda não existe uma padronização quanto a sua aplicação. Os protocolos de comunicação tem

sua segurança as vezes distribuída em camadas e as vezes encapsulada dentro de uma aplicação.

Como as demandas são diferentes para as mais variadas aplicações, a pesquisa deste tema é ainda recente para alguns temas e para outros ela vem se aprimorando e transformando de acordo com as novas necessidades e problemas que surgem da união dos componentes e tecnologias. Uma profusão de estratégias vem surgindo para atender a necessidade da Internet das Coisas e a cada ano o amadurecimento das tecnologias envolvidas cria novos cenários de aplicação, o que deve culminar em uma configuração de componentes deve começar a se consolidar como um padrão. A padronização de algo único ainda é intangível como visto neste capítulo, mas a modularização e aplicações específicas já são passíveis e serem enxergadas.

Uma excelente visão sobre o quanto genérica pode ser uma solução de Internet das Coisas pode ser vista na figura 2.42, onde a demonstração de que estamos falando sobre algo que funciona para qualquer negócio, com diversos tipos de dispositivo, em diversas redes, com diversos protocolos, enfrentando graças a isso uma problemática de segurança desconhecida até que todas as pontas sejam definidas e estas somente podem ser desenhadas após uma boa definição do negócio.

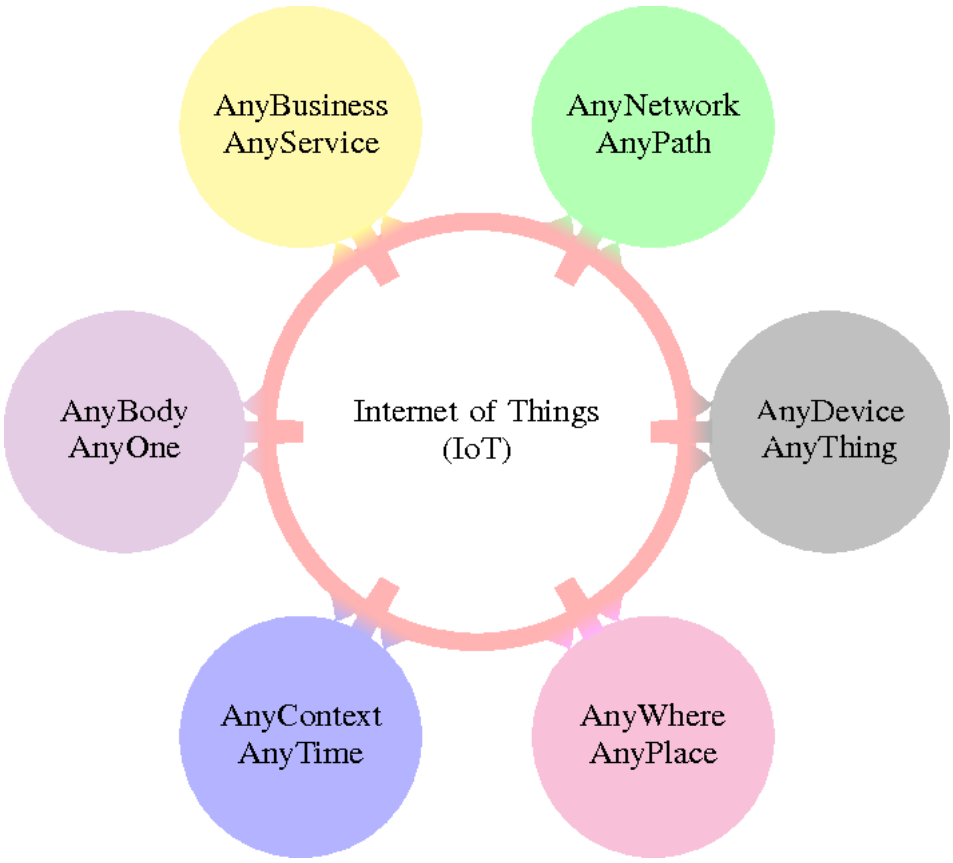

Figura 2.42: Uma visão de quanto genérica e abrangente pode ser uma solução de Internet das coisas. Retirada de (33)

3 SIMULAÇÕES E ARQUITETURAS DE REFERÊNCIA PROPOSTAS

Durante a pesquisa sobre o estado da arte em arquiteturas e protocolos, um dos pontos mais citados foi sobre a complexidade de projetos de Internet das Coisas, seja pelos desafios inerentes a união de tantas tecnologias em um único tópico, seja pela insegurança no quesito proteção de dados e garantias sobre o transporte efetivo e confiável dos dados. Em resumo, tem-se um tema que abrange diversas camadas, com necessidades de conhecimento que vão desde a eletrônica até o design de produtos e sua inserção no mercado, contemplando múltiplos perfis profissionais e abrangendo um projeto de IoT a diversas empresas desde sua concepção até sua implantação.

Baseado nestes anos de pesquisa e no trabalho de campo dentro da CrazyTechLabs, empresa fundada em 2014 que trabalha exclusivamente com Internet das Coisas e sistemas distribuídos com uso de computação em nuvem, traz-se a partir de agora uma proposta de uma arquitetura de base dividida em três camadas e com adoção de nuvem desde o início do projeto. Para todas as simulações e definições de componentes adotou-se a plataforma de nuvem da Microsoft, o Azure. É uma proposta também a observação dos tipos de projetos de Internet das Coisas dividida em dois grandes tipos: Projetos de telemetria onde a medição e a ação na ponta se dão com processamento centralizado na nuvem e Projetos que envolvem Inteligência na Ponta com uso de *Edge Computing / Fog Computing*.

3.1 TIPOS DE PROJETOS

Durante as pesquisas, simulações e o trabalho de acompanhamento de projetos no campo junto a CrazyTechLabs, notam-se diversos cenários de aplicação de Internet das Coisas, em diversos setores de negócios, e todos os projetos puderam ser divididos em dois grandes grupos, indiferente de suas aplicações, arquiteturas, tecnologias aplicadas, custo e força de trabalho necessária. Tal divisão é feita entre projetos de telemetria e projetos com uso de *Edge Computing*.

3.1.1 Projetos de Telemetria

Os projetos de telemetria são em sua maioria sistemas de medição e acompanhamento de campo, sem que haja necessidade de alto processamento dentro dos dispositivos, geralmente controlados por microcontroladores e podem estar divididos em diversos ramos de atividade. De acordo com sua necessidade de controle / medição estes podem ser atendidos por redes LPWAN, redes celular ou conexões locais cabeadas ou sem fio, auxiliadas por um *gateway* para sua saída do ambiente de ponta. Em alguns casos pode-se utilizar componentes dentro do *gateway* para ações de controle locais ou processamento de mensagens com finalidade de tradução de protocolo ou redução de dados, retenção de mensagens em caso de falha de conexão, não sendo o equipamento limitado somente a estas ações. Em um projeto deste tipo, o foco principal

está na coleta de medidas, em muitos casos não existe ação de retorno para o dispositivo, por exemplo pode-se analisar um medidor de água ou gás, este é um dispositivo que tem que integrar o uso dentro de faixas de tempo para posterior bilhetagem, ações na ponta tomadas pelo dispositivo são secundárias e por vezes inexistentes. Neste caso, a leitura tem duas características fundamentais: As leituras têm que chegar ao servidor e têm que ser totalmente confiáveis. Um ponto neste cenário é que a frequência nos envios pode sofrer alterações dentro de uma janela de tempo bem elástica, desde que o dado chegue e seja confiável a operação é válida e está dentro do esperado. Este exemplo dos medidores pode ser análogo a vários outros como medidores de temperatura de ambientes, rastreadores de veículos, rastreadores pessoais, estações meteorológicas e muitos outros, em alguns casos com uma adequação ao negócio nas regras de tempo de mensagem, janelas de envio, e as vezes comandos de retorno para solicitações de mensagens ou ações de campo, mas as características principais continuam em todos os casos, precisamos ter confiabilidade e conectividade.

Os cenários de aplicação deste tipo de projetos podem ir desde agricultura ou campo com ambientes remotos, prédios, indústrias, cidades e outros. Durante esta pesquisa foram acompanhados de perto projetos de medição de temperatura em caminhões, pesagem de caminhões na estrada sem balança, monitoramento de rede elétrica, monitoramento de ambientes, contagem de pessoas em ambientes e as características de cada um deles serão apresentadas mais a frente. Projetos desta natureza tem um volume de dados enviado pelo dispositivo relativamente baixo, o conteúdo de uma mensagem enviada pode ser visto na figura 3.1, estes dados são analisados para gerar alarmes ou gráficos em painéis de demonstração da operação que podem ficar no mesmo ambiente do sensor ou na nuvem, mapas são comuns em negócios onde a posição afeta a decisão que será tomada, como é o caso de um rastreamento de pessoas ou caminhões por exemplo. Nestes projetos pode-se ter uso de conceitos como o de *Digital Twins*, gêmeos digitais, que trazem os dados e metadados de caraterísticas do ambiente onde o sensor está inserido para um ambiente modelado digitalmente e com isso ações são tomadas no ambiente virtual e automaticamente transferidas ao ambiente real, bem como simulações e inferências também. Por exemplo podem-se ter sensores que usam dois tipos de comunicação, WIFI e cabeada, acompanhando em tempo real o tipo utilizado no campo pelo seu gêmeo no ambiente virtual, podendo alterar o tipo de comunicação por meio de comandos no gêmeo virtual e esta alteração repercutir para o dispositivo real.

Figura 3.1: Exemplo de conteúdo de mensagens em projetos de telemetria, extraídos de projetos realizados e simulações.

O cenário apresentado na figura 3.2 demonstra a utilização da coleta de dados por meio de sensores

ligados a dispositivos para geração de informação em um painel de monitoramento de forma simplificada.

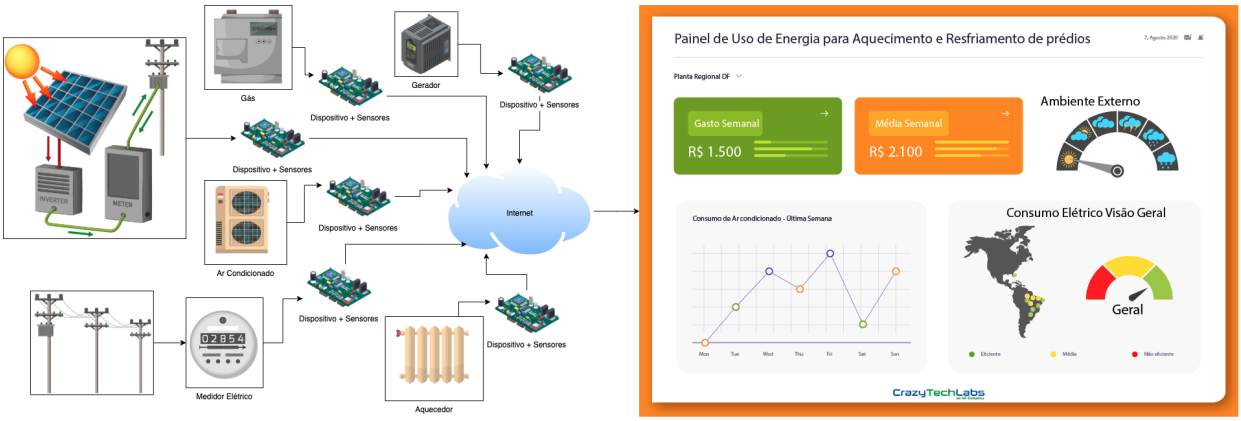

Figura 3.2: Exemplo de um projeto de telemetria com sensores enviando dados diretamente para nuvem.

3.1.2 Projetos com uso de Edge Computing

Os projetos com uso de *Edge Computing* possuem como característica principal a necessidade de processamento na ponta, seja por causa da natureza do negócio, seja por causa da ausência de uma conectividade confiável, embora esta última possa ser atendida por uma implementação de sincronia no *firmware*, código que roda em um dispositivo, ou pelo uso de um *gateway* com tal funcionalidade. Para negócios com a necessidade de tempo quase real, como indústrias, hospitais, controles de prédios inteligentes e outros, uma solução que tenha dentro de seu perímetro tudo que precisa para tomar decisões, apresentar dados, e controlar dispositivos conectados sem que haja conectividade com o meio externo e em períodos muito curtos de tempo, podem ser muito importantes se não cruciais para o funcionamento do local ou negócio. Os projetos desta natureza geralmente trazem componentes da nuvem para a ponta, economizando recursos, diminuindo o tráfego de dados, e por vezes podemos trazer também alguns componentes que ficam geralmente na nuvem, como uma camada de apresentação.

Um cenário de exemplo é uma indústria, onde existe uma necessidade de responder rapidamente a mudanças, garantir o segredo de processos e informações do chão de fábrica, estar apto a funcionar indiferente de ter conectividade com a internet ou não, além de integrar diversos tipos de comunicações e protocolos e algumas centenas de medições por linha de produção. Uma linha de produção além de um controle de motores e sensores tem, também, um processo de produção que precisa ser otimizado visando a eficiência da linha, para garantir o momento ótimo de trabalho tudo tem que estar em sintonia. Aqui cabe bem a aplicação de um aprendizado de máquinas, onde com dados antigos de funcionamento é possível se predizer problemas em futuro próximo, uma vibração maior com maior consumo de um motor podem ser indícios de uma falha em alguns minutos se uma ação não for tomada, com o uso da computação na ponta próxima a esta linha podemos ter essa inferência baseada em componentes que estão rodando localmente e em segundos retornar os ajustes necessários para que a falha seja revertida antes mesmo que aconteça, para isso, a união de serviços que rodam na nuvem e na ponta e um processamento local são fundamentais. A figura 3.3, traz um cenário de aplicação de *Edge Computing*, nesta perspectiva são citadas as possibilidades de termos a tradução de protocolos para obtenção de dados dos circuitos dos equipamentos de produção, aplicação de inteligência, armazenamento, conectividade com a nuvem e uma camada de apresentação

local, em um ambiente como esse é comum lidarmos com dezenas de protocolos proprietários, além dos de mercado, sem contar que alguns dados são integrados por meio de APIs REST e tudo isso tem que ser somado para tomadas de decisão. Câmeras com algoritmos de Inteligência Artificial, ou que somente enviam o *streaming* de vídeo para processamento no *Edge Device*, também são comuns atualmente, se todos esses dados fossem enviados a nuvem, para posterior processamento e retorno de dados e comandos para a ponta, não teríamos algo com a eficiência que é demandada no cenário industrial, além de custos extras.

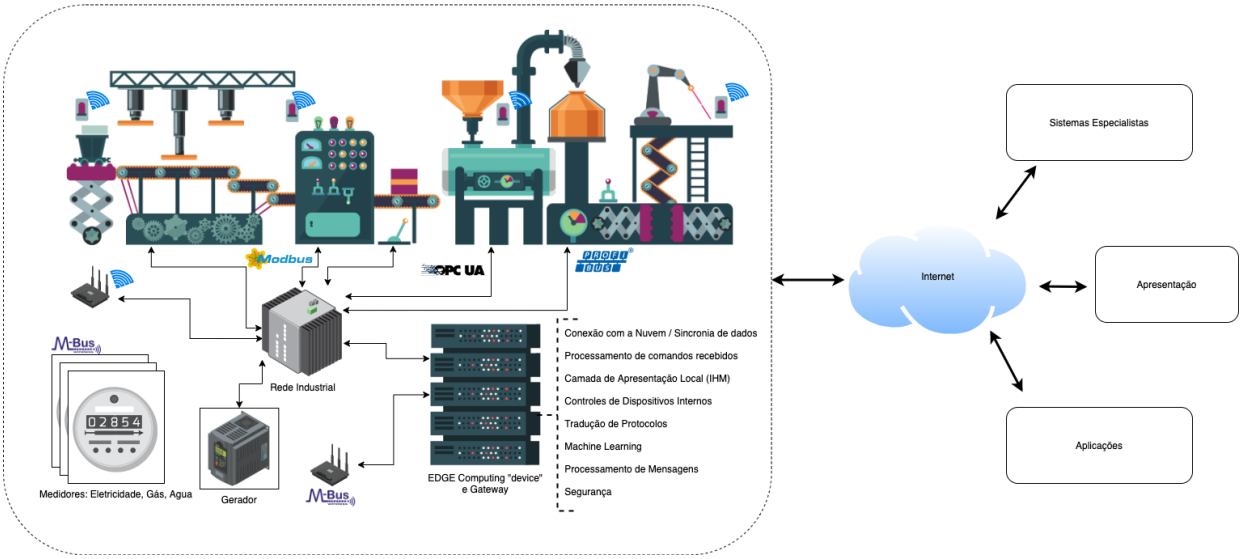

Figura 3.3: Exemplo da aplicação de Edge Computing em um projeto de Indústria com sensores e dados de PLCs sendo processados na ponta e posteriormente enviados ou não para nuvem e sistemas especialistas.

3.2 PROPOSTA PARA UMA ARQUITETURA DE IOT

A proposta é de uma abordagem em 3 camadas, disposta na figura 3.4, ligeiramente diferente da tradicional arquitetura de três camadas, focada no posicionamento lógico e físico dos componentes, dispondo componentes já conhecidos e utilizados em todas as arquiteturas de uma forma que comprovei ser mais simples de entendimento e gestão dentro dos projetos e simulações feitas. Outro ponto de decisão por utilizar a nova nomenclatura e a divisão, se deu pelo endereçamento direto das bordas de segurança que devemos ter em um projeto de IoT, com esta arquitetura tem-se três grandes blocos, interligados, com pontos de entrada e saída claros, a única questão que levantei durante as pesquisas e aplicações foi a questão da aplicação, que tanto na arquitetura de três camadas tradicional, quando na de cinco camadas era separada dos componentes de nuvem, neste quesito, entende-se que as vezes se tem uma separação, mas pode-se tratar no mesmo grande bloco, uma vez que as aplicações estarão também dentro de nuvens, públicas ou privadas, e no quesito segurança o impacto seria pequeno frente a simplificação obtida na implantação e documentação.

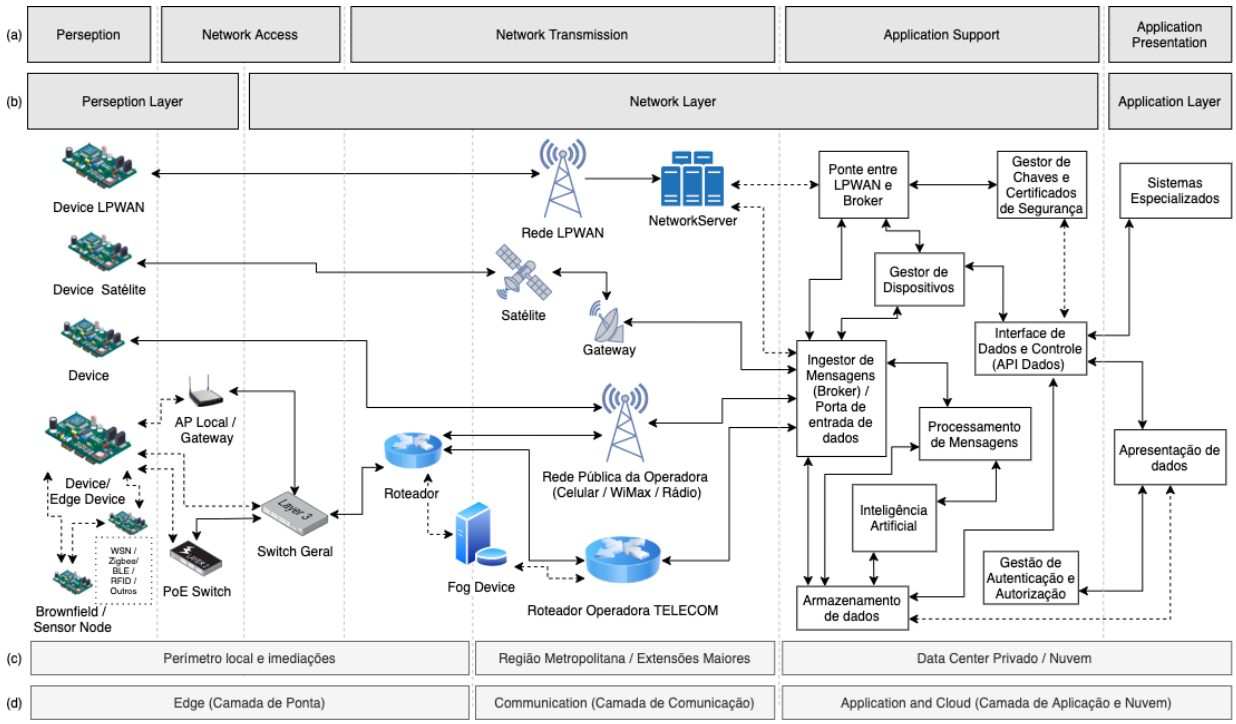

Figura 3.4: Arquitetura de IoT proposta (d), comparada com: arquitetura de 5 camadas (a), arquitetura de 3 camadas (b) e a localização dos componentes (c).

3.2.1 Camada de Ponta (*Edge*)

A primeira camada, nomeada de Camada de Ponta (*Edge*) é aonde tem-se tanto os sensores quanto a comunicação interna, a fronteira desta camada é definida pelo perímetro do local da aplicação, podendo ser estendido desde que continue no contexto local, por exemplo se existem dois locais geograficamente diferentes porém conectados para tráfego interno de dados via rádio ou uma linha privada indiferente do meio estes estariam nesta camada. Redes de sensores, *gateways*, servidores, dispositivos, roteadores internos e qualquer outro componente que faça parte da solução se incluem nesta camada, indiferente da funcionalidade. Nesta camada, se o uso de *Fog Computing* for definido como um componente da solução, ele estará presente dentro de sua fronteira, uma vez que faz parte da definição de ação para sensores e comunicação local.

No diagrama apresentado na figura 3.4 foram demonstradas algumas possibilidades de conectividade dos dispositivos, passando desde tecnologias LPWAN até as redes de sensores, de acordo com a necessidade podemos ter um dispositivo que trafega em diversas redes, sem fio ou cabeadas, e que também pode atuar como um *gateway* para sensores ou dispositivos mais antigos, o caso dos *brownfield devices* por exemplo. Os principais componentes desta camada e suas respectivas funcionalidades e comportamentos são descritos abaixo:

- **Dispositivos** – Os dispositivos são os componentes principais da Internet das Coisas, este componente que nomeio simplesmente de dispositivos trazem em sua composição na grande maioria um microcontrolador, acoplado a um ou mais sensores/atuadores e uma forma de comunicação que pode ser um modem celular (2G,3G,4G,5G), um rádio para uso de tecnologias LPWAN, um conector para

redes sem fio ou *Ethernet* ou até mesmo *Power over Ethernet*, dentre outros possíveis. Estes dispositivos são carregados com *firmwares* que atuam como clientes de serviços de nuvem, *brokers* de mensagens, implementam clientes de protocolos de aplicação como MQTT, AMQP ou HTPP, por exemplo.

- **Dispositivos de *Edge Computing*** – Estes dispositivos podem ser executados em hardwares microprocessados ao invés do dispositivo comum que tem um ambiente de execução mais restrito em relação a processamento, memória ou armazenamento por exemplo. Um dispositivo tido como *edge device* deve ser capaz de processar na ponta alguns serviços e operam com um sistema operacional para manter seu funcionamento, nos casos onde existem *frameworks* prontas para sua execução, como é o caso do IoT Edge da Microsoft, a execução destes serviços acontece dentro de contêineres e estes por sua vez são controlados por meio de propriedades e geridos pelo controlador de dispositivos IoT Hub, do Azure, a nuvem da Microsoft. Nada impede de uma execução independente de um fornecedor como a Microsoft, neste caso o dispositivo iria ter os serviços criados dentro dele pelo desenvolvedor e sua gestão também seria feita de forma autônoma, dentro de contêineres ou com aplicações instaladas no próprio sistema operacional, o que importa neste cenário é que o dispositivo seja capaz de processar e resolver seus problemas localmente, indiferente da conectividade nos casos onde se faça necessário. Exemplos deste caso de uso são algoritmos de Inteligência Artificial para inferência rodando perto da ponta, tratamento de imagens de câmeras com algoritmos de visão computacional, acionamento de alarmes baseado em leitura de dados, dentre outros tantos que envolvam o processamento na ponta. Pequenos servidores podem ser tratados como dispositivos de *Edge Computing*, vide o caso apresentado na figura 3.3 onde um servidor faz o papel do dispositivo.

- **Dispositivos do tipo não IP com comunicação / *Wireless Sensor Network*** – Os dispositivos que operam em rede *mesh*, ou por meio de protocolos de hardware, ou que formam uma rede de sensores podem se interligar a um dispositivo de Edge ou ter um gateway de saída para sua conectividade com a nuvem. Neste caso, alguns deles podem ser tratados na nuvem como dispositivos filhos de outros ou também pode-se incorporar a mensagem dos que estejam ligados ao gateway e autenticar somente o gateway na nuvem. Aqui trazemos protocolos como Bluetoooth, WMBus, Zigbee e outros para serem o componente de comunicação deste dispositivo.

- **Dispositivos antigos (*brownfield devices*)** – Estes são dispositivos antigos, sensores sem conectividade externa e que as vezes já estão no local em pleno funcionamento e então farão uso de outros dispositivos para sua saída para nuvem, geralmente somente são publicadores de dados.

- **Roteadores WIFI, Switches de rede, Switches POE** – Estes são componentes estruturais para interligar a rede interna com e o *backbone*, elo de ligação, com a rede externo.

- ***Gateway* de rede** – Este componente é responsável pela saída dos dados da rede interna para a rede pública.

- ***Fog computing* device** – Este dispositivo tem como meta atender com uma inteligência e processamento próximos ao ambiente de implantação da solução, podendo atender mais de um ambiente e garantindo uma menor latência de comunicação. Pode operar modularizado com aplicações específicas para o caso de uso ou genéricas da rede. Este componente tanto pode estar dentro da borda

da ponta, quando dentro da camada de comunicação, casos onde uma aplicação específica para um grande conglomerado empresarial, uma industria com 3 plantas próximas geograficamente, onde um dispositivo de *Fog* pode atender todas as plantas, mas atende somente elas, seria um caso onde ele estaria dentro da borda da ponta ao invés da camada de comunicação, se este fosse um serviço oferecido pela operadora de telecomunicações ou de nuvem, para todas as empresas da região, ele estaria dentro da camada de comunicação ao invés da ponta.

3.2.2 Camada de Comunicação (*Communication*)

A camada de Comunicação (Communication) abriga a comunicação fora do ambiente de ponta, trazendo os servidores de rede das LPWANs, os *gateways* de *last-mile* para comunicações via satélite, redes de operadoras de telefonia celular, torres de rádio, dispositivos de *Fog computing*, roteadores de redes públicas e quaisquer outros componentes de comunicação.

3.2.3 Camada de Aplicação (*Application and Cloud*)

Na camada de Aplicação (Application and Cloud) estão todos os componentes de processamento, armazenamento, apoio e suporte a aplicações, gestão de identidade e todos mais que façam parte da solução que estejam dentro de Data Centers tanto privados quanto de nuvens privadas e públicas. Em cenários onde a infraestrutura do projeto está dentro de uma nuvem, mas aplicações estão em outros provedores de serviço, ambos e a comunicação entre eles pertencem a esta camada.

Nesta última camada está o maior número de componentes e o maior nível de complexidade na escolha, pois é neste ponto que a necessidade de ingestão de mensagens e gestão de dispositivos, bem como a segurança e a integração com Inteligência Artificial devem estar interconectados de forma robusta para garantir o fluxo massivo de informação. Os conceitos de escalabilidade horizontal, onde aumentamos o número de instâncias dos serviços para garantir o provimento do serviço, e de escalabilidade vertical, onde aumentamos a capacidade de hardware da instância, se fazem presentes, de forma conjunta ou separada de acordo com o tipo de serviço de nuvem ou do Data Center.

Alguns podem ser unidos em um serviço de nuvem, ou dispensados do uso por uma simplificação, abaixo estão listados os principais componentes desta camada e sua função:

- **Ponte entre LPWAN e o Ingestor de Mensagens** – Este componente trata da integração das mensagens vindas do *Network Server* da rede LPWAN, uma vez que uma mensagem chega é necessário que se transforme aquele identificador de dispositivo da rede LPWAN em um dispositivo da rede local, neste quesito a segurança nesta integração vem de chaves geradas para o dispositivo por meio de um serviço de gestão de dispositivos, com acesso ao *broker* por meio de credenciais que devem ser resguardadas, também, com segurança e isto se dá por meio do componente de gestão de chaves.
- **Gestor de Dispositivos** – O gestor de dispositivos tem como função o gerenciamento do ciclo de vida dos dispositivos dentro da solução, garantindo a entrada segura, a habilitação ou não do dispositivo para que ele possa estar na rede de dispositivos e por fim a aposentadoria do dispositivo.

- **Ingestor de Mensagens (*broker*) / Porta de entrada de dados** – O Ingestor de Mensagens / eventos é a porta de entrada de dados para a solução, deve ser um manipulador de mensagens que pode operar via publicação e subscrição, ou como um roteador de mensagens, este componente também pode ser uma API REST, embora, neste caso a performance e o tráfego de dados não sejam adequados para uma solução que deve estar preparada para um tráfego massivo de dados.

- **Inteligência Artificial** – Componentes de Inteligência Artificial são comuns em projetos de Internet das Coisas, seja em um modelo de nuvem como treinador e ponta como inferência, seja no modelo de ambos os serviços rodarem na nuvem. Este componente pode ser acoplado ao processamento garantido um entendimento do dado em tempo de recebimento e gerando a consequente ação a ser tomada.

- **Armazenamento de Dados** – Indiferente de como se faça a ingestão, ou como se processe os dados, o Armazenamento de Dados é inerente a uma solução de IoT, muitas vezes dividido em diversos componentes, o armazenamento pode estar presente para um salvamento mais a quente, o que significa alta performance tanto de salvamento quanto de recuperação, quanto em um momento onde o armazenamento pode ser mais frio, onde se tem um tempo de resposta maior para as ações com os dados. Se liga diretamente as APIs e Sistemas Integrados, ao ingestor de mensagens, aos serviços de Inteligência Artificial, serviços de apresentação de dados e por vezes em modelos de sincronia com dispositivos de *edge computing*.

- **Gestão de Autenticação e Autorização** – Dispositivos precisam ser seguros e tornando-se necessária a Gestão de Autenticação deles, em outros componentes desta arquitetura como apresentação, sistemas especializados, APIs de integração, interfaces de gestão e outros também se faz necessária uma autenticação, bem como a garantia de que somente usuários autenticados possam ser autorizados de acordo com os níveis de permissão definidos pelo negócio, este componente age como um grande chaveiro e porteiro ao mesmo tempo para todos os componentes, garantindo tanto o acesso deles, quando o acesso para eles.

- **Apresentação de Dados** – Aqui entram sistemas de apresentação, aplicativos, painéis de controle, acesso de consultas específicas para criação de apresentação de informações, dentre outros que atuem na camada de Apresentação de Dados como mapas, APIs de gráficos e etc. Neste componente pode se materializar o conceito de *digital twins*, tanto para apresentação quanto também para elo de ligação entre algoritmos de simulação e inferência de dados.

- **Sistemas Especializados** – Sistemas Especialistas, ERPs, Sistemas Setoriais dentre outros fazem parte deste componente, geralmente integrado por meio de APIs de dados expostas, mas também pode acessar diretamente o armazenamento de acordo com a necessidade do negócio, interliga-se também com a gestão de autenticação e autorização e em alguns casos com o gestor de dispositivos.

- **Gestor de Chaves e Certificados de Segurança** – Senhas, certificados e dados que devem ser seguros devem estar armazenados neste componente que tem a função de garantir o acesso somente de requisições autenticadas e autorizadas, e também pode fazer uso de módulos de segurança de hardware para manter os dados seguros.

- **Interface de Dados e Controle (API de Dados)** – Uma API de exposição de dados e controle é quem deve fazer a integração entre ambientes externos e serviços/componentes de fora desta arquitetura e componentes internos, sendo a camada de fachada entre todos que precisam de dados e todos os dados previamente armazenados, bem como propriedades e dados referentes aos metadados dos sensores/dispositivos.

- **Processamento de Mensagens** – Este componente faz a ingestão das mensagens, recebendo de dispositivos autenticados e autorizados dados, bem como enviando dados para eles. Comumente este componente é responsável por rotas de mensagens e também pela autenticação de entrada e saída de dados.

3.2.4 Escalabilidade

Quando nuvem é a sua camada de infraestrutura e serviços, existem três modelos principais de contratação de serviços que são: IaaS onde tem-se a infraestrutura como serviço, SaaS que trata o software ou aplicação como um serviço e o PaaS onde as plataformas são entregues como serviço. Em IaaS, tem-se máquinas virtuais como principais bases para execução e implantação dos componentes desta camada, e neste caso tanto a escalabilidade vertical onde a capacidade da máquina virtual em relação a memória, processamento, disco, rede e etc. pode ser aumentada, quanto a vertical onde aumenta-se a quantidade de instâncias em execução desta máquina. Ainda neste cenário escalar a infraestrutura de forma horizontal pode se tornar uma tarefa complicada, não pelo fato de se criar uma nova instância, mas pela complexidade de fazer com que as várias instâncias funcionem em conjunto com seus componentes que rodam nelas. Uma base de dados por exemplo deve ser capaz de funcionar nesse modelo, um *broker* de mensagens deve estar apto a funcionar analogamente ao seu funcionamento inicial somente sendo mais performático com relação ao volume de dispositivos e mensagens entrando e saindo, o mesmo vale para todos os serviços. Uma outra solução no cenário de IaaS são os contêineres, com eles é possível garantir uma execução isolada de componentes da arquitetura com uma possibilidade maior de escala horizontal, não diminuindo, porém, a complexidade de execução dos componentes em escala. Quando a infraestrutura está em modelo de PaaS, ou seja, em plataformas, tem-se a possibilidade de escalar mais facilmente de forma horizontal, pois os serviços já vêm preparados para isto e sua administração sob a operação e seu crescimento é menos dependente de desenvolvimento e do conhecimento avançado dos detalhes internos para resolver esta questão, o aumento de carga de forma vertical às vezes acontece, de acordo com cada serviço, e por vezes pode ser representado pela contratação de outro serviço e sua disponibilidade em uma camada inferior a um serviço de balanceamento de carga de entrada. Em SaaS, a escala acontece de forma transparente para o usuário, pois o serviço deve estar preparado para atender a alta demanda, cabe aqui uma ressalva que se aplica a todos os modelos: antes de utilizar ou acoplar um serviço a sua arquitetura, os limites devem ser estudados para entender como adicionar este serviço a fim de que resolva sua demanda.

Estando a camada de aplicação em um Data Center ou em nuvem privada o modelo mais comum de adoção é de virtualização, também é pertinente o uso de contêineres e de sua orquestração por meio de gestores como Kubernetes por exemplo, neste cenário o modelo de aplicação de escala será idêntico ao de nuvem, com a diferença de que em um Data Center a responsabilidade sobre o hardware pode ser do contratante o que aumenta o nível de complexidade da operação e da gestão, haja visto que será também

preciso vencer as dificuldades de escalar horizontalmente os componentes.

3.2.5 Preocupações sobre Segurança

Esta seção trata de preocupações com segurança, e este tema é um dos pontos bastante importantes em soluções de Internet das Coisas, segurança tem que ser tratada desde o começo do desenho do projeto e deve ser pensada em cada componente e na união deles e posteriormente na solução como um todo, não deixando nenhum destes passos sem esforço destinado. Quanto menos bordas de conexão menos problemas de segurança na troca de informações entre as camadas, mas isto não torna a arquitetura proposta nem mais nem menos segura. Na figura 3.5 estão destacados em vermelho os pontos de atenção e abaixo uma observação ao longo da pesquisa e do acompanhamento / trabalho nos casos reais na CrazyTechLabs, como sendo o mínimo necessário para implantação da arquitetura proposta:

- **Bordas de Interligação** - Todo elo de comunicação deve ser pensado como um possível ponto de entrada ou vazamento de dados, no nosso caso em análise tem-se os dispositivos como o primeiro ponto de interligação, neste caso com o mundo real e suscetível a ataques físicos e virtuais, depois as saídas e entradas de cada camada, onde deve-se pensar em mitigar possíveis ataques na comunicação, embaralhamento de dados, roubo de identidade dentre outros e para essa mitigação podemos implantar o tráfego seguro nestas configurações com uso de certificados, túneis de dados seguros dentre outros. Um ponto importante nas bordas da camada de aplicação está não somente em proteger a comunicação, mas proteger os acessos, senhas seguras, dupla autenticação em aplicações de controle e gestão da solução dentre outros.

- **Dispositivos** - Aqui tem-se uma divisão em três tipos de dispositivos, aqueles que somente se comunicam com outros dispositivos e, portanto, podem se proteger por meio da segurança do protocolo do enlace físico (caso do LoRa, Zigbee, Zwave, Bluetooth e outros), desde que corretamente configurado. Outros dispositivos falam diretamente com a rede pública, são eles os de rede celular, LPWAN, Ethernet e outros, nestes também é preciso preocupação quanto a segurança, principalmente nos que, além do enlace físico, tem uma camada de aplicação que permeia as camadas, dispositivos IP por exemplo que usam protocolo MQTT precisam se preocupar com a camada de conexão e com a gestão de credenciais de acesso além de canais seguros de dados. Por último tem-se os dispositivos que somam mais de uma funcionalidade e rodam sistemas operacionais, os *edge devices*, estes além de todas as questões anteriores, tem a problemática de acessos indevidos ao sistema operacional, roubo de dados internos, ataques físicos e outros.

- **Comunicação interna, cabeada e sem fio** - Todos os dispositivos também estão suscetíveis a roubo de credenciais na comunicação que tem que ser segurada também, ataques que ficam escutando pacotes ou se introduzem no meio são comuns e se utilizam de *switches* desprotegidos, *Access Points* ou roteador Wifi interno para tornar os ataques reais.

- **Fog Device** - Os devices que controlam informações sejam de uma ou de mais de uma corporação que estão nas bordas ou na rede pública merecem especial atenção, pois eles podem sofrer ataques silenciosos e ter seu controle tomado sem que a ponta sequer tenha paradas, com isso se a comunicação

passante ou dados armazenados não estiverem devidamente protegidos, além de roubo de informações, pode-se ter roubo de identidade sem que o dispositivo sequer tenha sido atacado, gerando uma rede com informações sem confiabilidade e posteriormente um grande prejuízo no projeto.

- **Ingestor de Mensagens** - Este será o coração e a porta de entrada ao mesmo tempo, implantação de análises de *logs* para um monitoramento preditivo, credenciais de acesso forte, chaveiros de credenciais nas pontas e internamente na camada de aplicação além de habilitação de canais seguros para comunicações neste componente são fundamentais. Apesar de altamente robusto, ele em muitos cenários é único e um ataque pode parar todo a solução.

- **Interligação entre LPWAN e Solução** - Este elo de ligação ainda se faz necessário em muitas redes LPWAN, nele é preciso garantir a identidade do dispositivo, um controle de acesso baseado em origem e *tokens* de segurança, além de verificar integridade das mensagens e origem antes de efetivar a conversão de uma mensagem em uma mensagem assinada por um dispositivo acreditado na rede interna da solução.

- **API de Integração de controle e dados** - Esta é basicamente a API que integra a solução com o resto das aplicações e expõe os dados, além de controlar e gerir dispositivos e a solução em si, expondo métodos de controle, portanto precisa estar debaixo de um sistema de gestão de autorização forte, trafegando dentro de canais seguros e com métodos que estejam protegidos contra injeção de códigos maliciosos. Uma gestão em cima do acesso, clientes concorrentes com mesmas credenciais pode ser uma boa abordagem, além dos pensamentos anteriores.

- **Gestor de dispositivos** - Este componente precisa ser segurado contra-ataques, pois ele tem a chave de acesso que permite desde a criação de novos dispositivos, até a aposentadoria de dispositivos funcionais. O quanto mais ele estiver resguardado dentro de um núcleo com poucas exposições melhor.

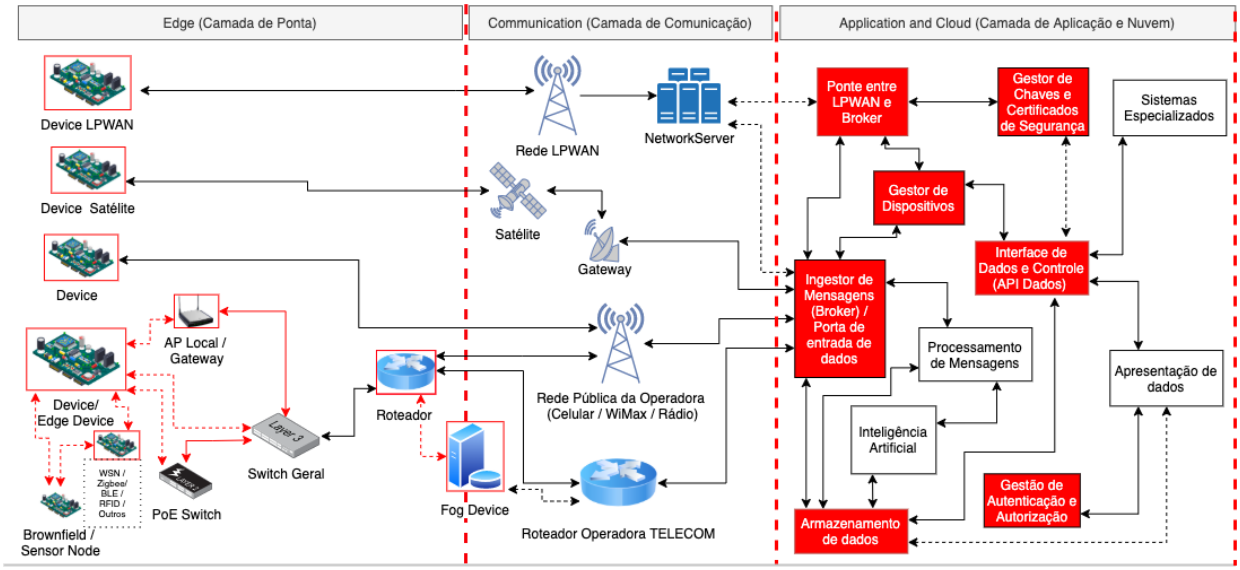

Figura 3.5: Pontos de atenção quanto a segurança na arquitetura proposta.

3.2.6 Aplicabilidade e Testes

Para teste de adequabilidade da arquitetura, será desenvolvida uma aplicação de telemetria com medições de temperatura, umidade e luminosidade implantada nos modelos de: Infraestutura como serviço (IaaS) com uso de uma máquina virtual servindo de infraestrutura para toda a solução, Plataforma como serviço (PaaS) com uso de serviços combinados a fim de seguir a arquitetura proposta e Software como Serviço (SaaS). Todos os testes serão feitos no Azure, a nuvem da Microsoft, em cada seção será apresentada uma variação da figura 3.4 a fim de contemplar os serviços utilizados ao invés dos componentes, alguns componentes podem ser omitidos da solução de acordo com sua necessidade. Adicionalmente, será desenvolvida uma aplicação com uso de PaaS com uso de *Edge Computing* para exemplificar a aplicação deste tipo de projeto.

3.2.6.1 Projeto de Telemetria no modelo de Infraestrutura como serviço (IaaS) / Data Center / Nuvem Privada

Para esta projeto implantado no modelo de PaaS, demonstrado na figura 3.6, os componentes de processamento, armazenamento, Ingestão de Mensagens / gestão dos dispositivos e apresentação dos dados foram encapsulados em uma máquina virtual, eles são respectivamente cobertos pelas seguintes aplicações: *Scripts* em Python, Influx DB, Mosquitto e Grafana. O Mosquitto é um *broker* de mensagens baseado no protocolo MQTT, o Influx é uma base dados que implanta o modelo de séries temporais, o Grafana é um motor de geração de gráficos e interfaces visuais que nesta aplicação funcionará como camada de apresentação. O componente de segurança da aplicação, nesta implantação está dividido entre o sistema de autenticação do Mosquitto para controle dos acessos de dispositivos e clientes de subscrição de mensagens, o *Network Security Group* (NSG) do Azure que está ligado a um IP público e ao dispositivo de rede conectado na máquina virtual com a função de controlar o tráfego da rede, estão liberadas as portas TCP 1883 para o Mosquitto, 23 para o acesso ao sistema operacional via SSH e 3000 para o Grafana, este último será o gestor dos usuários da aplicação por meio de login e senha. O componente de armazenamento que está presente dentro da máquina virtual por meio do Influx, em conjunto com todo o sistema operacional e outras aplicações instaladas é persistido no discos ligados a máquina e provido como todos os outros componentes como um serviço de infraestrutura no Azure, sendo os discos, também, uma implantação do componente de armazenamento.

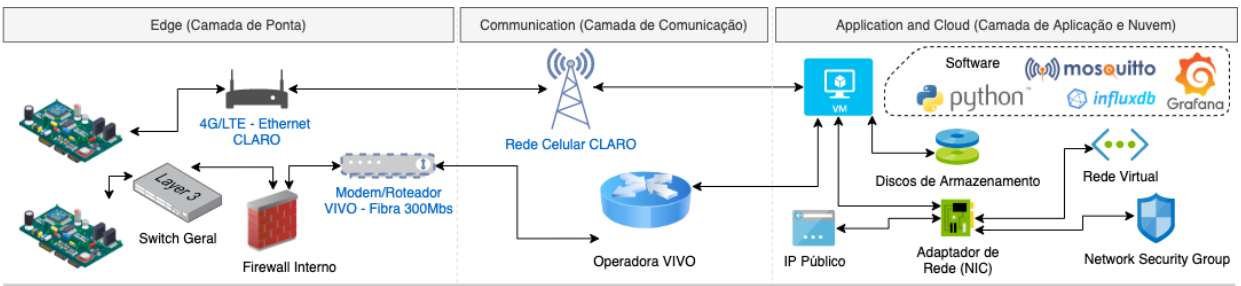

Figura 3.6: Diagrama de Arquitetura Aplicada em modelo de Infraestrutura como serviço.

Uma visão do funcionamento interno dos *scripts* e do ciclo de vida da mensagem de cada dispositivo

pode ser vista no diagrama exposto na figura 3.7 e um resumo de suas funcionalidades pode ser visto na lista abaixo:

- **Simulador de Dispositivos** - Simula um dispositivo se identificando na rede e posteriormente enviando mensagens para um tópico que se subdivide em andar e dispositivo, sendo o andar uma variação em tempo de envio entre 1 e 5, garantindo uma carga desigual. Foi introduzido um período de descanso aleatório entre os envios e todo o trabalho dura três minutos depois do início. Este *script* recebe dois parâmetros, um deles para definir quantos dispositivos deve executar por vez de forma paralela e o segundo para dizer o QoS do envio.
- **Processamento** - Neste *script* ocorre o processamento das mensagens, um cliente é conectado ao *broker* e assina o tópico de envio de todos os dispositivos por meio de um endereço curinga, garantindo que todos os andares serão recebidos e dentro deles todos os dispositivos que estão enviando. Uma vez que a mensagem chegue, seja processada e contabilizada para os resultados, ela é transformada em uma medida contendo o identificador do dispositivo, o andar, temperatura, umidade e luminosidade e esta medida é salva na tabela de medidas do Influx.
- **Cliente** - Para efeito de teste de chegada da mensagem no *broker* em um ponto diferente do processamento este *script* foi criado para fazer a subscrição ao tópico de forma análoga ao anterior e simplesmente contabilizar a chegada da mensagem para fins de contagem.

Foram criados três cenários para a simulação dos dispositivos e clientes de MQTT, além do processamento, variando a rede de envio e a posição do cliente externo conforme a tabela 3.1. Os testes utilizaram como infraestrutura de telecom uma conexão de banda larga de 300Mbps de fibra da Vivo, uma banda larga de 240Mbps da Claro e uma conexão 4G LTE da Claro, todas na cidade de Brasília, no bairro Lago Sul, entre os meses de agosto e outubro de 2020. Os testes foram feitos para os QoS 1, 2 e 3, simulando 1000 e 5000 dispositivos para cada QoS para cada cenário.

Tabela 3.1: Tabela de Rede usada nos testes para clientes e dispositivos

Cenário	**Rede dos Dispositivos**	**Local / Rede do Cliente**
1	VIVO	Dentro da Máquina Virtual
2	VIVO	Brasília - Usando Rede Claro Cabeada
3	Rede 4G LTE Claro	Brasília - Usando Rede Claro Cabeada

Considerações na implantação da arquitetura no modelo de IaaS feita segundo a figura 3.6:

- **Flexibilidade** - Pode-se trabalhar com diversos softwares, modelos, padrões de funcionamento, pois toda a estrutura está baseada em blocos, seja em máquinas virtuais ou contêineres sendo executados em Kubernetes. Neste quesito, podemos colocar a IaaS como o único modelo viável para alguns tipos de aplicação, principalmente as que têm regras muito diferenciadas ou volumes que não se encaixem nos padrões de PaaS e SaaS.
- **Controle** - Esta certamente é a forma mais aderente ao controle total tanto dos dados quanto da implantação da arquitetura, pois é possível se moldar desde o início o que é desejado e persona-

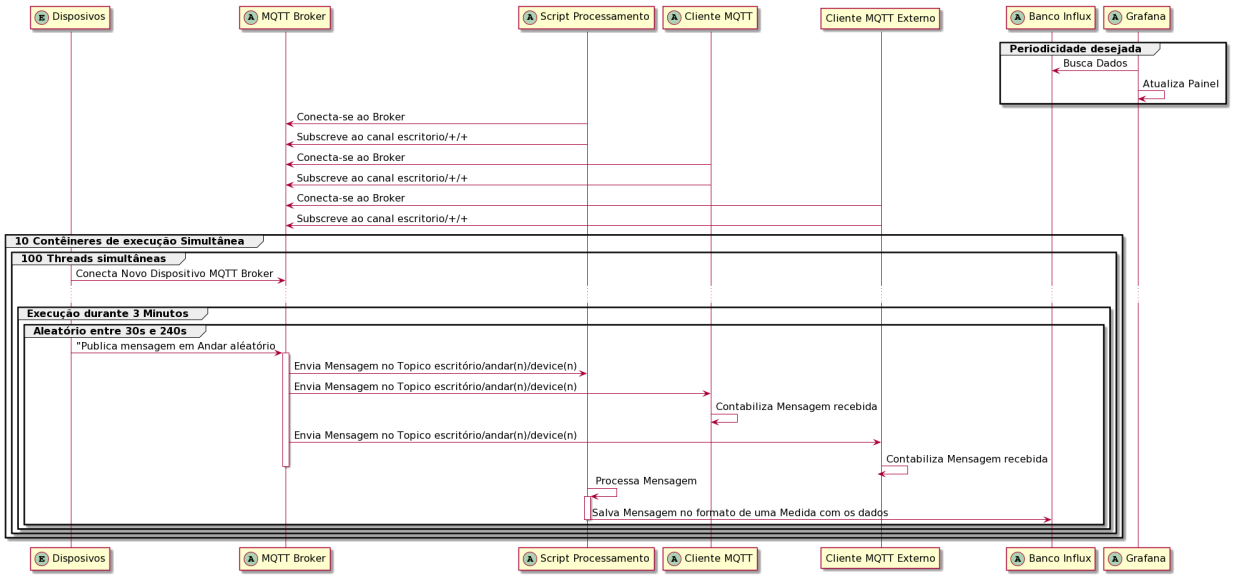

Figura 3.7: Diagrama do ciclo de vida da mensagem nos cenários. (A) Camada de Apresentação e Nuvem e (E) Camada de Ponta, o cliente Externo não está dentro das camadas.

lizar a qualquer tempo, sem *lock-in* o que pode liberar a troca de fornecedor a qualquer tempo e praticamente sem esforço extra de desenvolvimento.

- **Complexidade de Implantação** - Esta é a maior ressalva neste modelo, pois todos os serviços devem ser iniciados do zero, criados dentro da infraestrutura e posteriormente interligados o que pode levar tempo devido a curva de aprendizado das tecnologias envolvidas.

- **Escalabilidade** - Neste quesito, trazendo como exemplo os componentes deste teste e com as aplicações usadas, tem-se no *broker* a possibilidade escalar verticalmente aumentando os requisitos da máquina. Quando existe a necessidade de escala horizontal é pode-se usar a funcionalidade de ponte entre duas instâncias fazendo com que as mensagens de uma sejam enviadas a outra e vice-versa. Para o Grafana é possível a separação em um contêiner sendo este executado fora da máquina permitindo a criação de outras instâncias e seu apontamento para uma ou mais bases, já para a base de dados podem-se criar réplicas. A complexidade da escala em modelos como este estão na união de diversos serviços que lidam com cargas diferentes, todas devem ser geridas pela operação e os modelos de escala criados do zero sem facilitadores como existem em PaaS e SaaS onde os serviços são preparados para escalar horizontalmente e verticalmente de forma mais simples.

- **Segurança** - A segurança em IaaS padece do mesmo problema da implantação, cada componente interno tem seu modelo e às vezes a implementação conjunta pode ser trabalhosa. Olhando para a implementação feita tem-se um bloqueio de tráfego de portas, dispositivos e serviços de coleta e processamento de dados (*Scripts*) que foram geridos pelo Mosquitto e os usuários geridos pelo Grafana que sem uma centralização pode gerar falhas no processo de criação e gerenciamento de longo prazo, deixando brechas abertas. Outro ponto é que quando utiliza-se o modelo de IaaS se tem total gestão pelo sistema operacional o que requer um monitoramento constante nas atualizações, utilização e acessos. Na implantação feita a porta 23 ficou aberta para qualquer IP o que poderia garantir um acesso por meio de roubo de senha do usuário administrador por exemplo.

Artefatos gerados e Resultados

A *interface* visual de demonstração dos gráficos não tinha nenhum requisito direto, portanto o mínimo é demonstrado, na figura 3.8, em gráficos que trazem a média de valores da série criada a partir das mensagens enviadas por todos os dispositivos e separada por andar. Analisando os resultados de envio demonstrados na figura 3.9 pode-se notar que em todos os testes com uso de uma conectividade mais estável e com possibilidade de envio massivo não houve perdas de envio, problemas estes que se devem por causa do grande número de conexões e uso de tráfego o que é indiferente da banda disponível e sim está diretamente ligado ao tipo de conexão, no caso estamos falando de um modem 4G preparado para receber um número pequeno de conexões simultâneas, aqui falávamos de 1000 e 5000 enviando em intervalos tempo que acabavam por se sobrepor devido a natureza aleatória dos tempos de descanso entre uma mensagem e outra e também da tentativa de conexão. Nas contagens de mensagens recebidas, dividida por QoS e por cenário, demonstradas nas figuras 3.10 e 3.11, vemos um comportamento parecido do que fora observado nos envios.

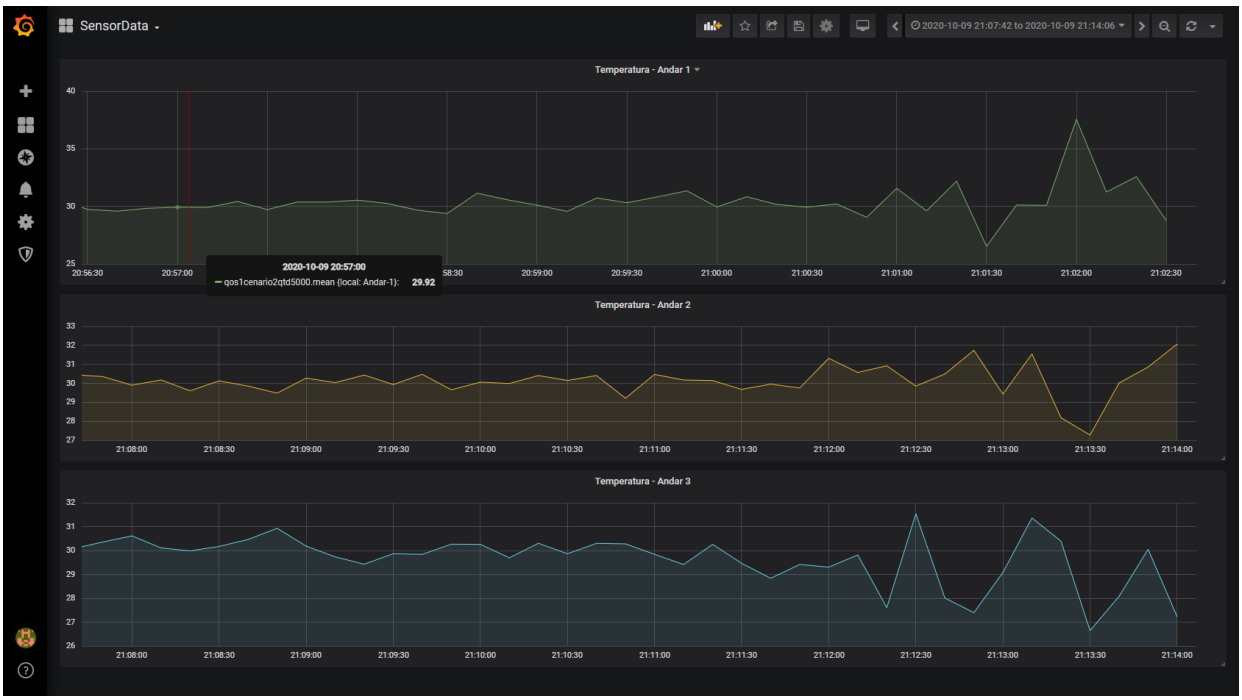

Figura 3.8: Tela do Grafana que apresenta um painel com dados sumarizados dos andares.

No MQTT, os QoS 1 e 2 geram uma sobrecarga na conexão, pois para o caso do 1 precisam enviar um aceite das mensagens e para o caso do 2 as mensagens devem ser únicas gerando uma sobrecarga ainda maior porque os aceites têm que vir do *broker* e do dispositivo para essa garantia conforme citados na revisão da literatura anteriormente. Dentro deste panorama onde existia a sobrecarga da conectividade conforme visto na contagem das mensagens enviadas aliado a sobrecarga de mensagens vemos uma duplicação de mensagens para o cenário 3 nos QoS 1 e 2, o que pode ser mitigado com um código que trate melhor os envios de acordo com as limitações de conexão oferecidas e seja mais adaptável a esta dificuldade.

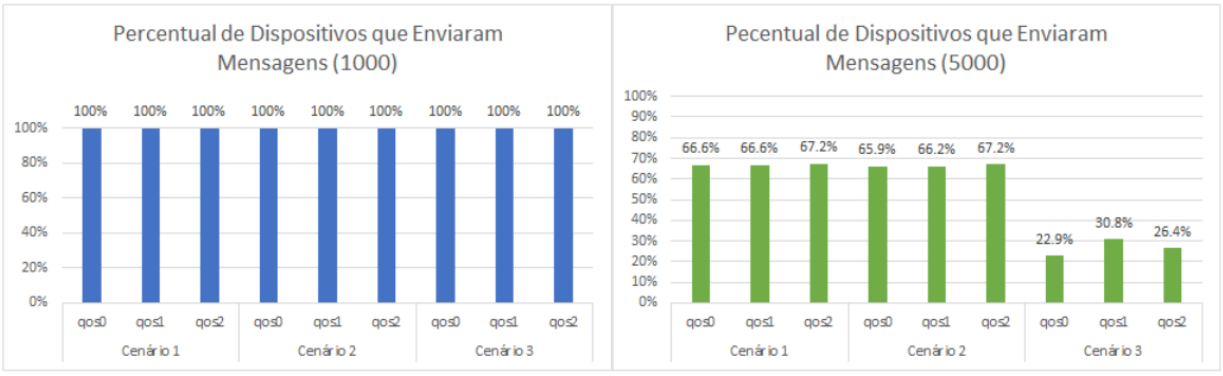

Figura 3.9: Percentuais de mensagens enviadas nos testes dividido por QoS e Quantidade de dispositivos e cenário

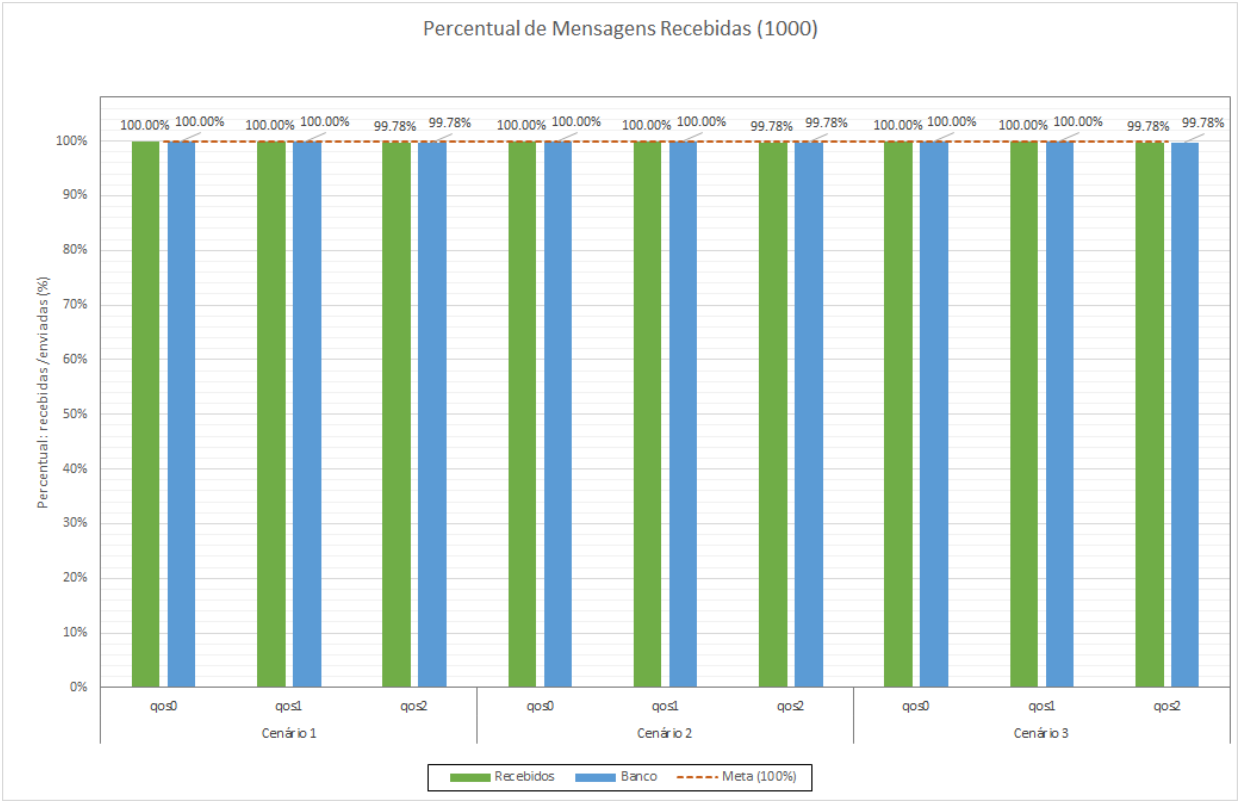

Figura 3.10: Percentuais de mensagens recebidas nos testes com 1000 dispositivos, dividido por QoS e Cenário

Os resultados foram satisfatórios, como o objetivo era validar a arquitetura, não foram feitos comparativos entre outros servidores ou aplicativos que tivessem a mesma funcionalidade dos que foram utilizados nos testes. Todos os dados de envio, contagens de mensagens recebidas, arquivos de código dos *scripts* e instruções para configuração do ambiente estão no Apêndice na seção de Resultados IaaS.

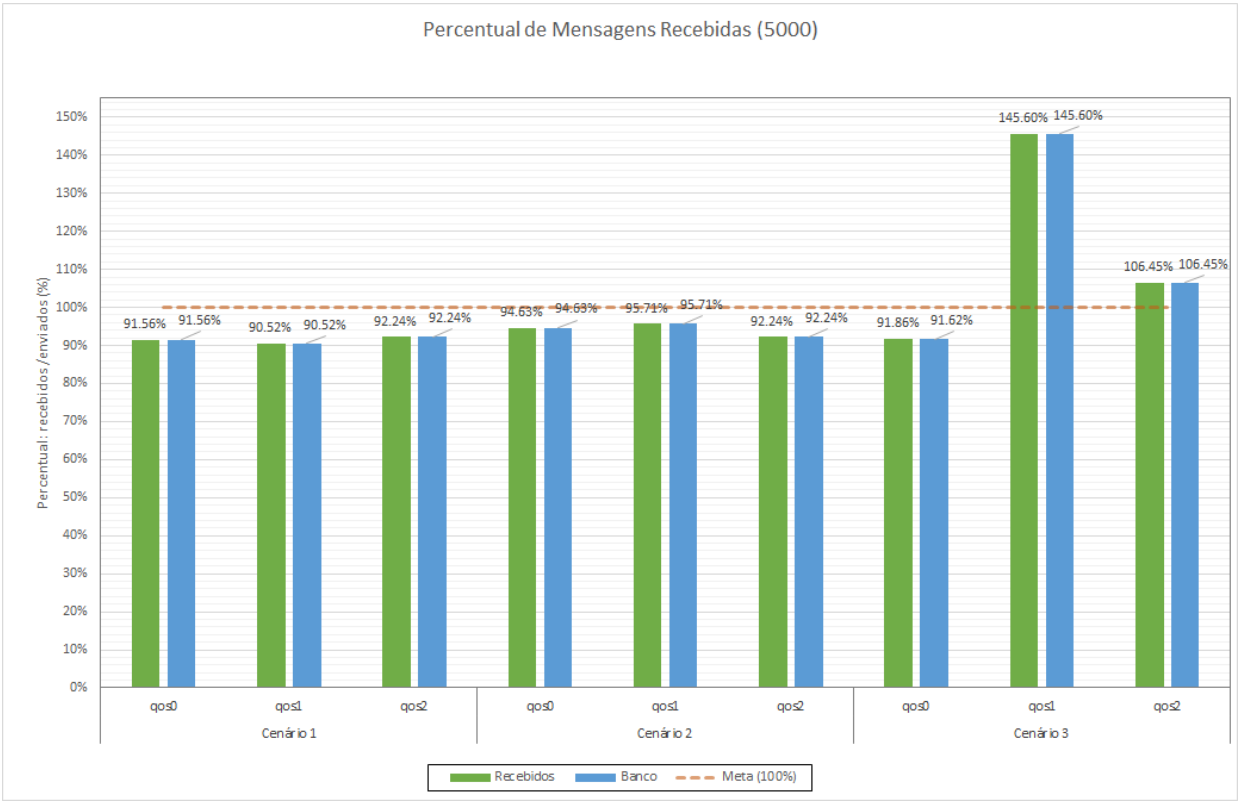

Figura 3.11: Percentuais de mensagens recebidas nos testes com 1000 dispositivos, dividido por QoS e Cenário

3.2.6.2 Projeto de Telemetria no modelo de Plataforma como Serviço (PaaS) do Azure

O uso de uma plataforma como serviço une diversos serviços e as problemáticas de gestão da infraestrutura falados no modelo de IaaS são totalmente ignorados agora. Em PaaS a preocupação está em dados e na união de serviços em prol da solução, embora seja bom ressaltar que apesar do cenário pareça mais simples, a complexidade pode ser grande principalmente se o modelo adotado não se adequar ao crescimento esperado da solução no tempo, tendo em vista que existem limites nos serviços.

Para esta implementação, utilizaremos os componentes e as disposições contidas na figura 3.12, na camada de nuvem existe uma vasta lista de serviços que podem atuar de forma parecida e às vezes para a mesma função, o que os diferencia principalmente são as funcionalidades, não de forma exclusiva, neste quesito contam também características de forma de uso, custo e de vez em quando a capacidade. Devido a multi especificidade de alguns serviços do Azure, tem-se uma gama de interligações entre os componentes apresentados, não necessariamente eles são dispostos de forma única, podemos por exemplo ter mais de uma conta de armazenamento, mais de uma implantação de Azure Functions, e assim por diante. Abaixo estão descritas as ligações e funcionalidades exploradas no diagrama da figura 3.12:

- **Ponte entre LPWAN e Ingestão de Mensagens** – Esta funcionalidade na arquitetura sugerida é controlada por uma Azure Function, com o gatilho de HTTP que será ativada a cada novo envio do servidor de rede LPWAN, este por sua vez, envia diversas mensagens de controle, solicitação de mensagens para o dispositivo, posicionamento e outras de acordo com o provedor de serviços escolhido. A função por sua vez ao receber a mensagem com o *payload* irá transformar a mensagem

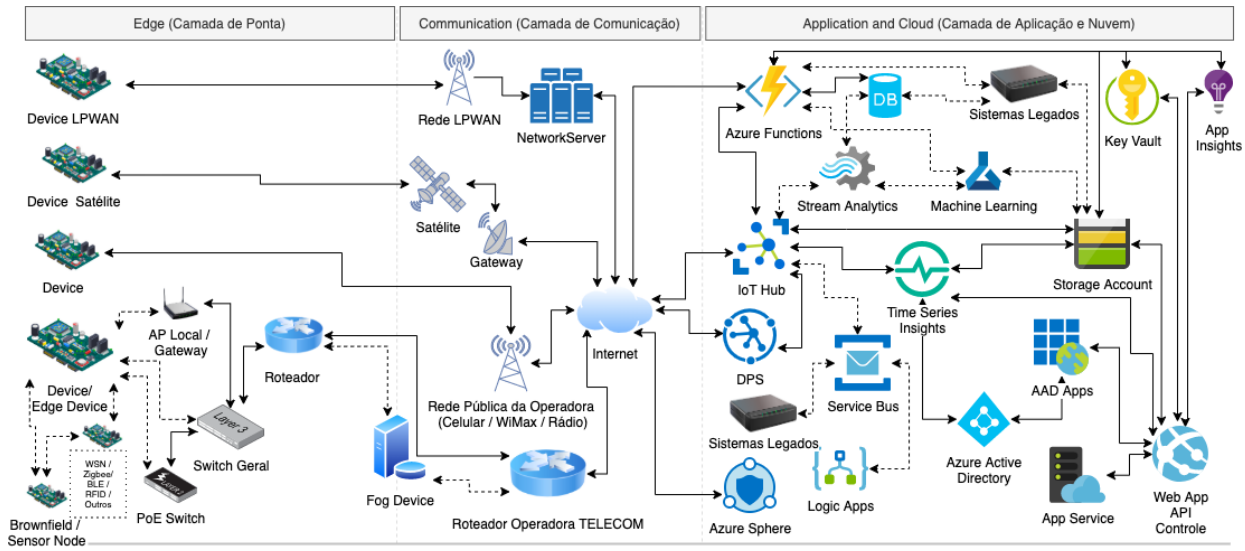

Figura 3.12: Diagrama de Arquitetura Aplicada em modelo de Plataforma como serviço no Azure.

recebida no formato que irá enviar ao IoT Hub para entrada na solução. Outro ponto muito importante que acontece aqui é a criação de uma instância do dispositivo LoRaWAN com as credenciais de entrada no IoT Hub, para tal, uma *string* de conexão com o Hub é necessária e para evitar a colocação em código, uma instância do Key Vault é levantada e configurada para permitir o acesso da Function. Com este processo garantimos a segurança tanto do acesso ao IoT Hub, quanto a identificação correta do dispositivo e com isso se assegura um nível alto de confiabilidade dos dados, pois na ponta a segurança entre os dispositivos e o servidor de rede é tratada pelo protocolo de rádio LoRa e pela integração interna entre o textitgateway da rede e o servidor de rede LoRaWAN.

- **Gestor de Chaves e Certificados** – Esta é a função do Key Vault dentro da arquitetura, ele pode não somente armazenar de forma segura senhas e certificados, mas também gerir tempos de expiração, controle de acesso a eles, bem como guardar logs de atividades.

- **Ingestão de dados, Gestor de dispositivos** – O IoT Hub faz a ingestão dado e a gestão de dispositivos, assim como seu ciclo de vida. Para uma garantia maior de segurança na entrada dos dispositivos na rede gerida por ele, utiliza-se o DPS (Device Provisioning Service), no momento da ingestão da mensagem é possível reter os dados para que clientes de processamento possam receber mesmo se conectando depois, nele é possível também rotear mensagens para clientes baseado em dados da mensagem ou dos dispositivos. No quesito comunicação MQTT, AMQP e HTTP são nativos, visando um melhor aproveitamento de recursos como o de dispositivos gêmeos já abordado previamente, um SDK está disponível para múltiplas linguagens de programação. O IoT Hub faz a gestão tanto de dispositivos de telemetria, quanto de dispositivos Edge, outro ponto é a gestão não somente de arquivos, ele também destina arquivos diretamente para uma conta de armazenamento, facilitando assim *dump* de dados ou arquivos de integração que os dispositivos tenham que enviar para cumprir algum processo que não seja tratado por mensagens ou comandos.

- **Processamento de Mensagens** – Este talvez seja o componente da arquitetura que tenha mais possibilidades dentro do Azure, quando se fala em criar uma arquitetura de referência com serviços. A

figura 3.13 mostra os cinco caminhos propostos e que não necessariamente tem que ser seguidos em todas as aplicações, nem tampouco com todos serviços listados ou limitando-se a eles. No primeiro caminho depois da ingestão da mensagem tem-se a passagem por um Azure Stream Analytics, um serviço que consegue analisar dados passantes, com múltiplas entradas e saídas, implementando janelas de tempo para agrupamento, possibilitando chamadas à Inteligência Artificial, salvamento e consultas direto em bases de dados, repasses para outros processadores de eventos, entre outros. O Azure Functions, uma implantação de funções de códigos no modelo *serverless* é o segundo caminho no diagrama e pode tratar mensagens que acabaram em tempo de chegada com uma alta capacidade de escala, baseado em gatilhos que disparam a ação, sendo um deles um gatilho ligado diretamente ao IoT Hub, ou seja, quando se tem uma mensagem enviada ele dispara seu conteúdo como parâmetro de entrada no código, a partir dele podemos usar as ligações já prontas para bases de dados, relacionais ou não relacionais (temporais se incluem), chamar uma API externa, uma chamada de inteligência artificial, um sistema especialista ou legado, de forma conjunta, separada, encadeada ou consolidando várias ações em uma única. Uma outra forma, demonstrada no caminho 3 é enviar os dados diretamente para uma conta de armazenamento, com isso pode-se usar um serviço para ler os dados e efetuar o processamento, este pode ser uma Azure Function, ou um leitor em *batch* que é executado de tempos em tempos ou acionado por um novo arquivo monitorando-se o repositório da conta de armazenamento. O quarto caminho é fazer o uso do Azure Time Series Insights, este serviço faz uso de uma interligação com o uma conta de armazenamento e com o as aplicações registradas no Active Directory para trazer uma das opções mais práticas dentre todas as outras apresentadas. Uma vez conectado, faz a leitura das mensagens, transforma dados em modelos com possibilidade de criação de hierarquia, permite que sejam feitas consultas, separa dados entre quente e frio, que são respectivamente o uso de uma resposta mais rápida as consultas (tempo atual), e uma resposta mais demorada nas consultas (para consultas com data mais antiga). Além da gestão do acesso a dados, o TSI também entrega uma SDK em Javascript para uso conjunto com as APIs de consulta aos dados para apresentar as consultas em formato de gráficos, tudo isso integrado ao Active Directory garantindo a gestão dos usuários. No último passo descrito, o de número 5, leva a mensagem por meio de um Service Bus, um barramento de mensagens com alto poder de escala e confiabilidade na entrega de mensagens, para uma Logic App que irá tomar uma ação, este cenário está baseado nas rotas do IoT Hub, que vai destinar uma mensagem para o Service Bus. Por exemplo podemos ter uma propriedade na mensagem que diz de onde ela veio e com isso roteamos para uma Logic App ou para um sistema especialista que receberá esta mensagem.

- **Inteligência Artificial** – Os serviços que implementam a IA dentro do Azure estão agrupados em uma suíte de desenvolvimento na qual algoritmos são treinados e publicados em APIs de inferência, tudo baseado em dados, neste caso, ingeridos pelo IoT Hub e salvos em bases de dados ou em contas de armazenamento. E em serviços cognitivos, prontos e entregues por meio de APIs, estes por sua vez trazem uma vasta gama de funcionalidades, tais como: visão computacional, entendimento de voz, texto e outros, todas as opções podem ser utilizadas em projetos de IoT, por meio das chamadas, seja por Azure Functions, Stream Analytics ou *scripts*.

- **Armazenamento** – O armazenamento pode ser feito por bases de dados como serviços, por contas de armazenamento e também pelo uso de bases de dados instaladas e geridas dentro de máquinas

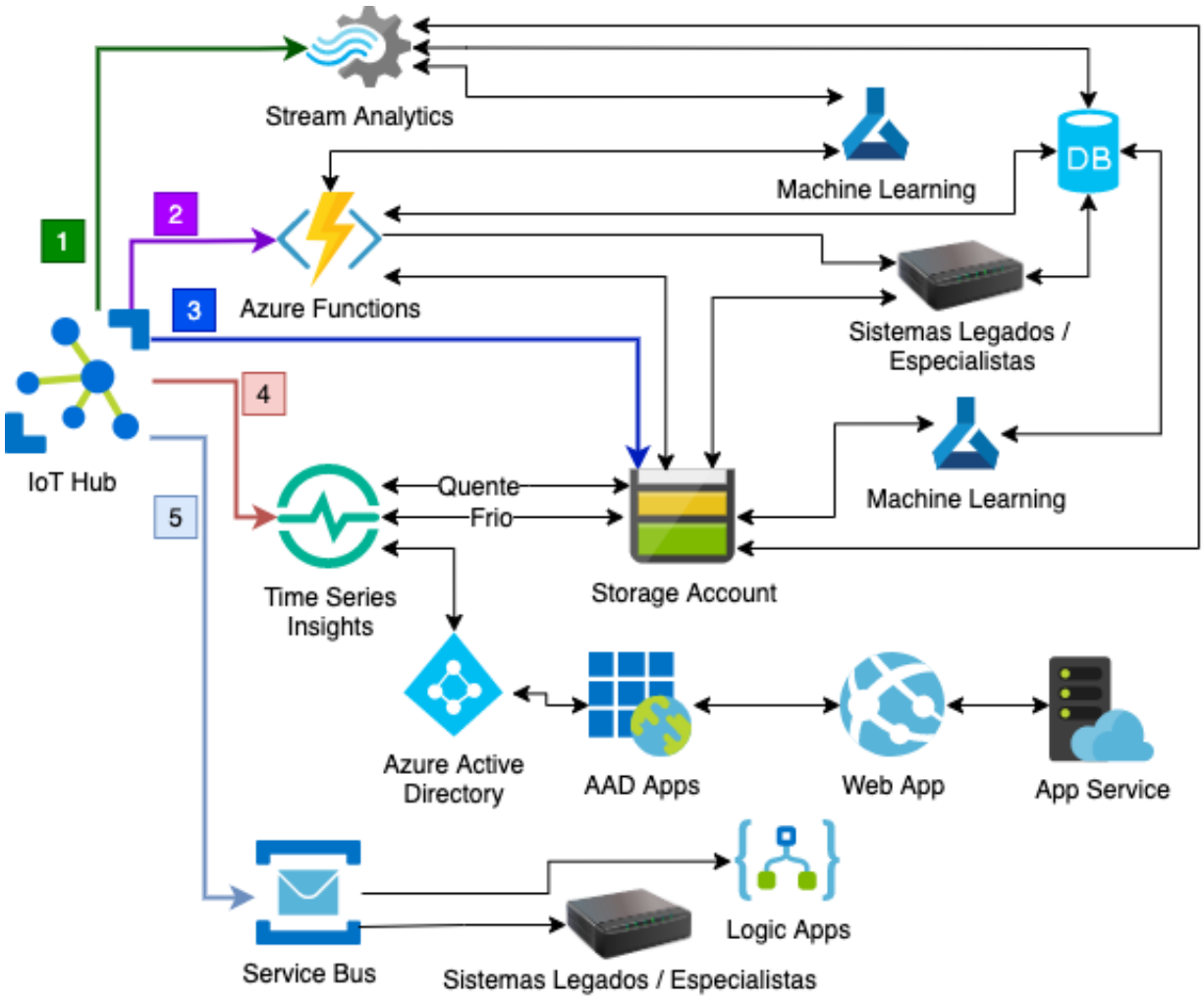

Figura 3.13: Caminhos propostos para processamento das mensagens ingeridas.

virtuais ou contêineres.

- **Segurança - Gestão de Autenticação e Autorização** – Toda a segurança de autenticação e autorização deveria passar pelo Active Directory, desde integrações de serviços até usuários de aplicações, com isso garantimos a unicidade para o controle de autenticação e autorização e a integração deles com todos os serviços. Para casos diferentes, deve-se também pensar em formas de acesso, controle de acesso de usuários as aplicações de ponta que não estão integradas antes de iniciar o desenvolvimento, não somente a preocupação com os usuários deve ser pensada, todos os elos de integração devem estar em canais seguros de tráfego de dados e as bases ou contas de armazenamento e pontos de entrada/saída devem receber atenção especial.

- **Interface de Dados e Controle (APIs)** – As APIs e Interfaces estão dentro de aplicações, geralmente em Web Apps, para sua execução utilizam o App Services, que são as plataformas onde a preocupação está somente nos dados e configurações, ao invés de gerir todo o servidor de aplicação como se estivéssemos no modelo IaaS.

- **Integração com sistemas legados e especialistas** – Esta integração é muito importante e aqui não cabe um serviço, mas sim uma prática, no desenho demonstrado aqui colocamos todos dentro da

mesma infraestrutura, em alguns casos o cenário é diferente, então faz-se necessária segurança na integração e nos canais de comunicação.

- **Acompanhamento de execuções no Código** – Um serviço que pode elevar a qualidade da solução é o Application Insights, ele se liga a plataformas e aplicações e monitora a execução em alguns casos na granularidade de métodos sendo executados, tal informação pode gerar tempos de acesso, visualização de motivos de degradação de serviço, falhas que estão acontecendo em tempo real e integrando este serviço com um monitoramento pode-se ter alarmes e ações tomadas de acordo com critérios preestabelecidos.

Quanto aos dispositivos, na aplicação da arquitetura, foram utilizadas placas de desenvolvimento baseadas nos microcontroladores ESP32 da Espressif e MT3620 da Mediatek implementando o Azure Sphere com comunicação usando LoRaWAN, GRPS, WIFI e Power Over Ethernet. Na finalidade de demonstrar o lado de comunicação, interligação e simplicidade do código para integração usando o SDK da Microsoft para o IoT Hub em linguagem C para todos os casos. A exposição de dispositivos físicos comprova as possibilidades, mas no quesito volume utilizei um gerador de dispositivos com carga definida de dez mil dispositivos enviando mensagens para o IoT Hub, em todos os casos físicos ou simulados o envio de dados respeitou as mesmas premissas definidas como teste, criando um dispositivo que enviaria temperatura, umidade e luminosidade em locais diferentes. Para o teste de volume, fora também utilizado um simulador de dispositivos, descrito no apêndice, o qual registrou dez mil dispositivos que enviaram mensagens por 1 hora em intervalos de 2 minutos, os dados de todos os dispositivos, incluindo os físicos foi ingerido pelo IoT Hub, processado pelo Time Series Insights, armazenado em uma conta de armazenamento e a possibilidade de visualização e consultas pôde ser feita por uma aplicação web utilizando HTML e Javascript, com apoio das bibliotecas do Time Series Insights para geração dos gráficos e Microsoft Graph para prover a autenticação no Azure Active Directory e permitir usuários autorizados a ver os dados e efetuar consultas. Uma visão dos gráficos, consultas e arquivos de armazenamento gerados pode ser vista nas figuras 3.14 e 3.15. Dentro dos testes feitos, os provedores de comunicação foram os mesmos dos ensaios feitos para aplicação da arquitetura no modelo PaaS para os modelos de dispositivos cabeados e WIFI, além destes utilizei uma conta de LoRaWAN de acesso a rede da American Tower e um chip de telefonia da operadora CLARO para os testes com o 2G GPRS.

Uma visão dos dispositivos físicos usados na exemplificação da arquitetura pode ser vista na figura 3.16, da direita para esquerda um dispositivo usando o Azure Sphere e conectividade WIFI, seguido por um ESP32 com conectividade LoRaWan, outro com alimentação e conectividade por meio de Power Over Ethernet e por fim o hardware com conectividade 2G GPRS para troca de mensagens, os dois últimos também fazem uso do ESP32 da Espressif como seu microcontrolador. Cada tecnologia de comunicação (Protocolos) tem sua forma e suas etapas de comunicação, e estão demonstrados no diagrama apresentado na figura 3.17, os componentes que não têm ação direta no processamento das mensagens, ou servem somente como roteadores de mensagens, bem como camadas e enlaces de comunicação física foram omitidas do desenho por não interferirem no entendimento da arquitetura.

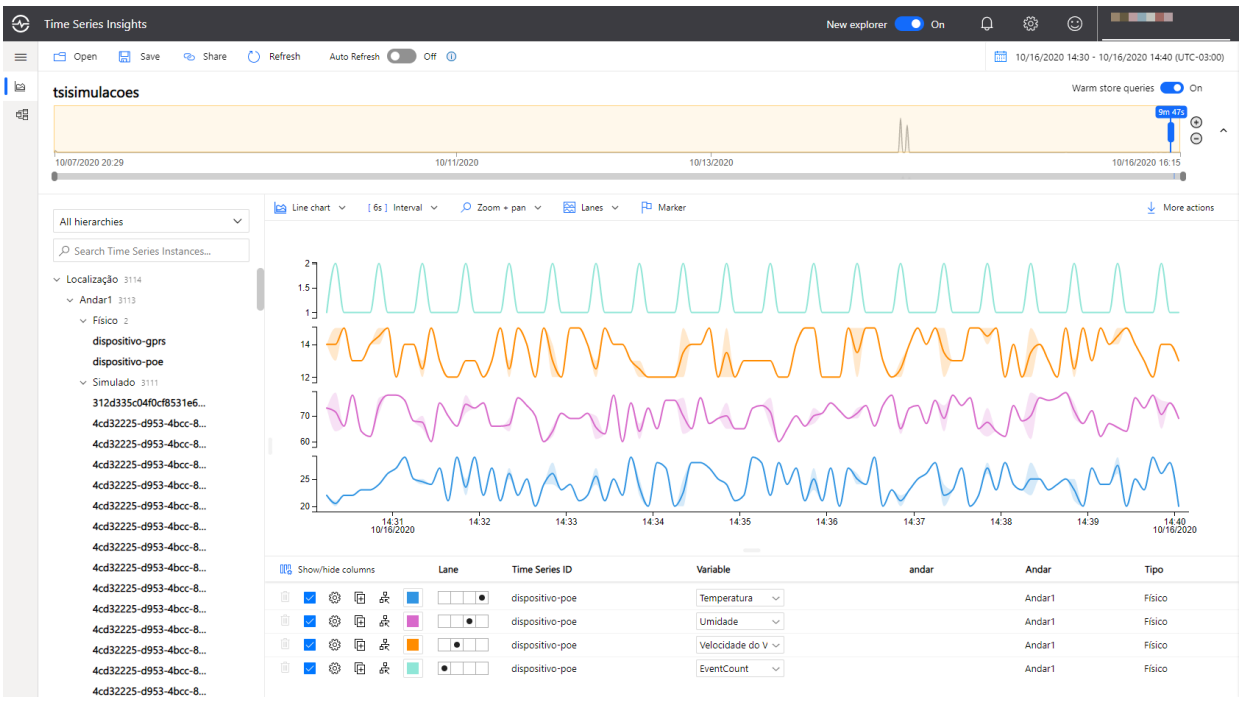

Figura 3.14: Apresentação e análise de Dados relativos ao dispositivo PoE apresentados no Time Series Insights Explorer, nesta tela é possível também se ver o agrupamento de dispositivos baseados em hierarquias, separando dispositivos simulados de dispositivos reais.

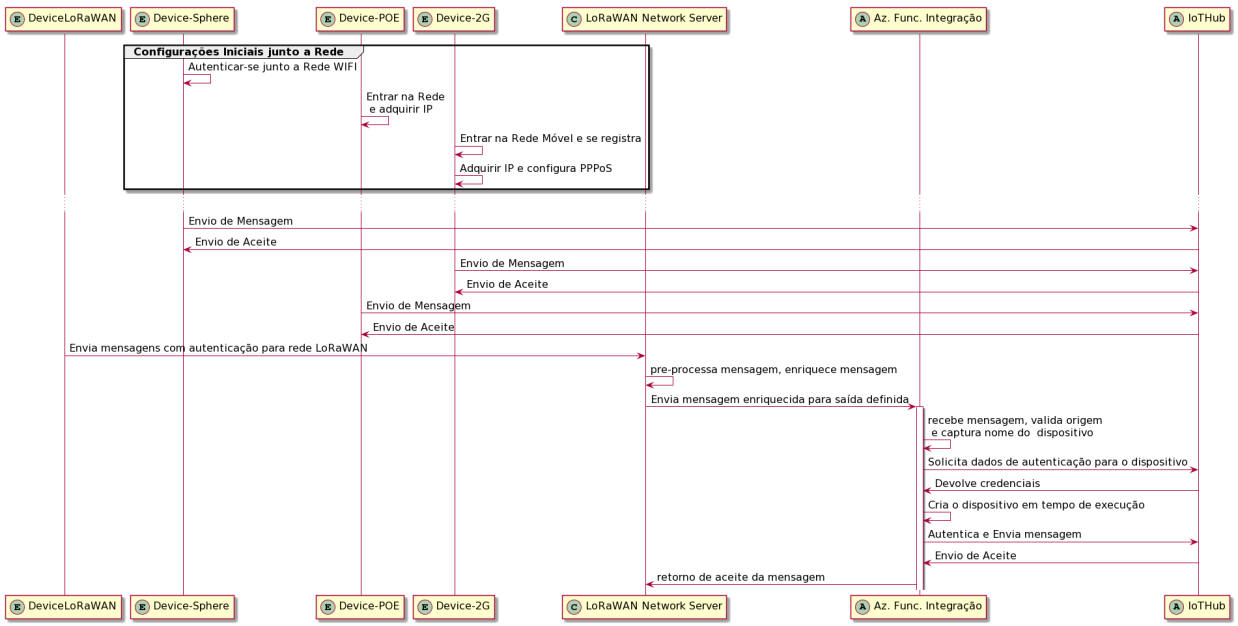

Figura 3.17: Diagrama de ações necessárias, simplificado, para comunicação dos dispositivos, as disposições entre as camadas dos componentes segue a seguinte distribuição no diagrama: (E) estão na camada de ponta, (C) estão na camada de comunicação e (A) em Aplicação e Nuvem

O Azure Sphere tem uma forma diferente de tratar os dispositivos, trata-se de uma solução com três pilares onde o dispositivo é seguro por uma forma própria de comunicação interna nos componentes de hardware, com um sistema operacional conectado diretamente à nuvem aliado a um serviço de gestão de

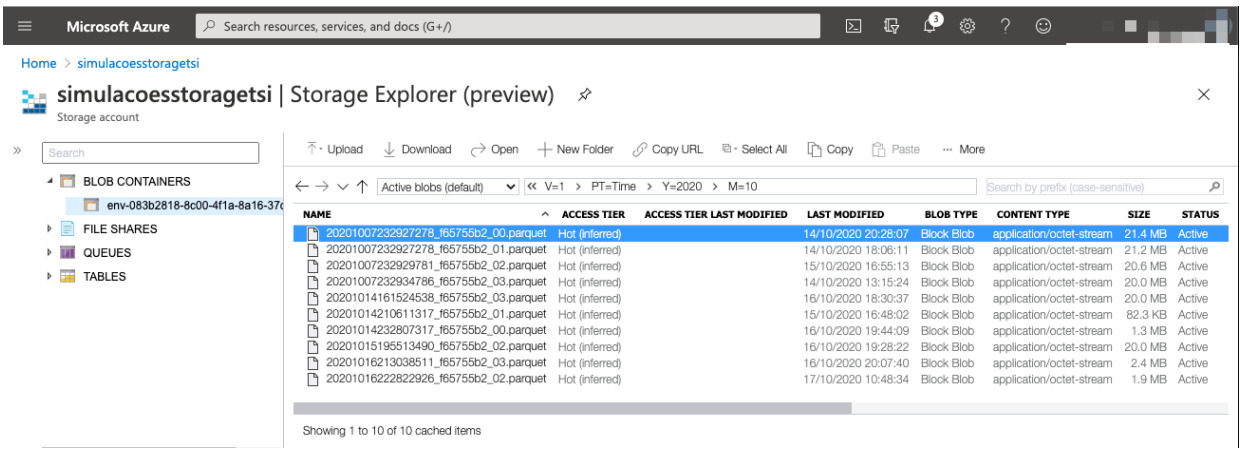

Figura 3.15: Arquivos gerados pela ingestão de dados controlada pelo Time Series Insights em uma conta de armazenamento permitindo integração imediata de qualquer outra ferramenta de processamento ou IA.

ciclo de vida de dispositivos também na nuvem, com isso tem-se dois momentos neste desenvolvimento, um que trata do sistema operacional e aplicações que rodam nele e um que lida diretamente dentro da aplicação. No caso da implementação feita estamos com a aplicação também se conectando na nuvem da Microsoft por meio do IoT Hub, mas a aplicação pode ser agnóstica e executar um fluxo que se conecta a outra nuvem ou a um *broker* privado por exemplo. O Azure Sphere implementa as sete propriedades de um dispositivo seguro descritas na seção de segurança do capítulo anterior. Outro ponto importante está nos aceleradores que a solução traz para o desenvolvimento, como o sistema operacional está em uma camada separada e gerida pela própria Microsoft, com isso o desenvolvedor não se preocupa com questões como segurança, atualização de aplicações (*firmwares*) e garantias de identificação do dispositivo, tudo isso é tratado pelo serviço de nuvem e garante uma diminuição no tempo de desenvolvimento e implanta uma solução altamente segura pronta para uso e implantação, uma visão destas divisões pode ser vista na figura 3.18.

Considerações na implantação da arquitetura no modelo de PaaS feita segundo a figura 3.12:

- **Flexibilidade** - Com menos flexibilidade nas bases da camada de aplicação, este modelo compensa essa premissa com uma gama de serviços que podem agir como peças de sua implementação. Atualmente dentro da plataforma Azure essa diversidade de componentes complementares tem tornado muito simples contornar, essa que às vezes é tida como uma problemática neste modelo de computação em nuvem. Em minha visão, para o uso em IoT, PaaS tem se tornado uma forma muito eficaz para este tipo de aplicação, uma vez que algo seja necessário e não esteja disponível pode-se adicionar componentes em IaaS.

- **Controle** - Existe controle dos dados e da forma no uso de PaaS, o que não se controla são as estruturas de infraestrutura.

- **Complexidade de Implantação** - A complexidade está na ligação dos componentes e na escolha, o que pode ser mitigado com um bom conhecimento do projeto a ser implantado, incluindo as necessidades de futuro. Esta modalidade certamente é muito menos trabalhosa que a IaaS para implantação, pois as camadas de infraestrutura não são um item de preocupação.

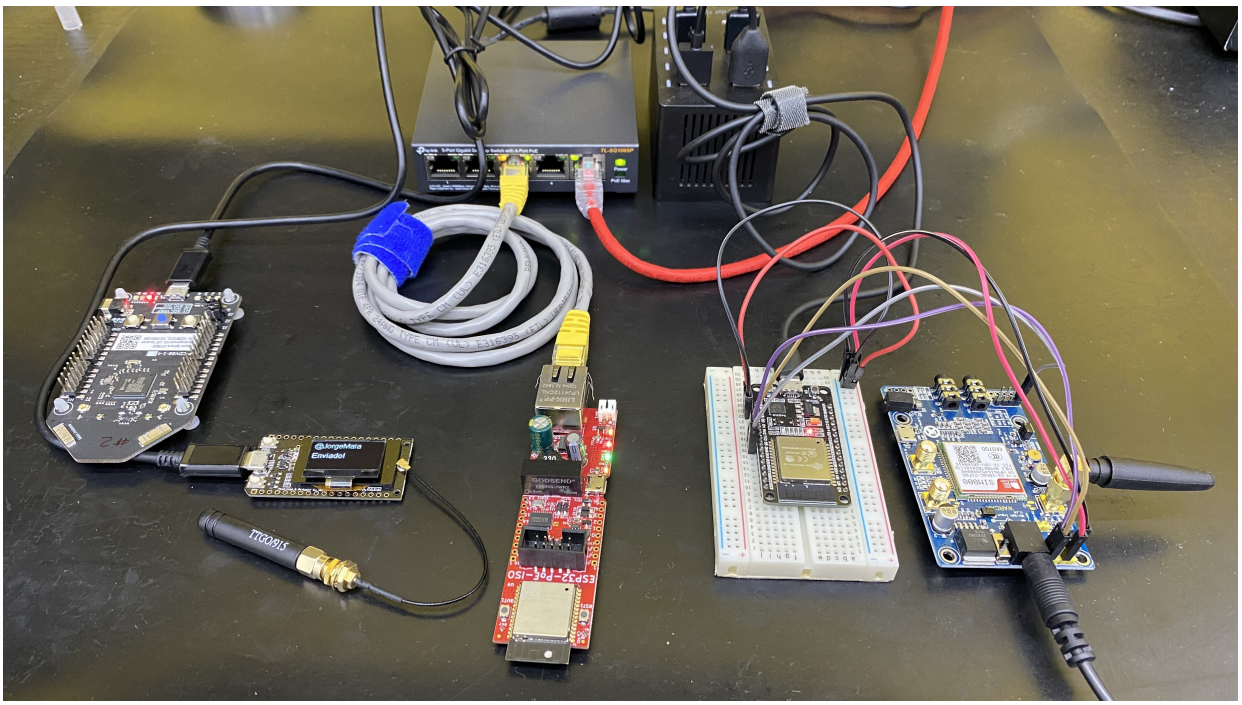

Figura 3.16: Fotos dos dispositivos físicos usados na exemplificação de uso da arquitetura proposta usando o modelo de Plataforma como Serviço (PaaS) no Azure.

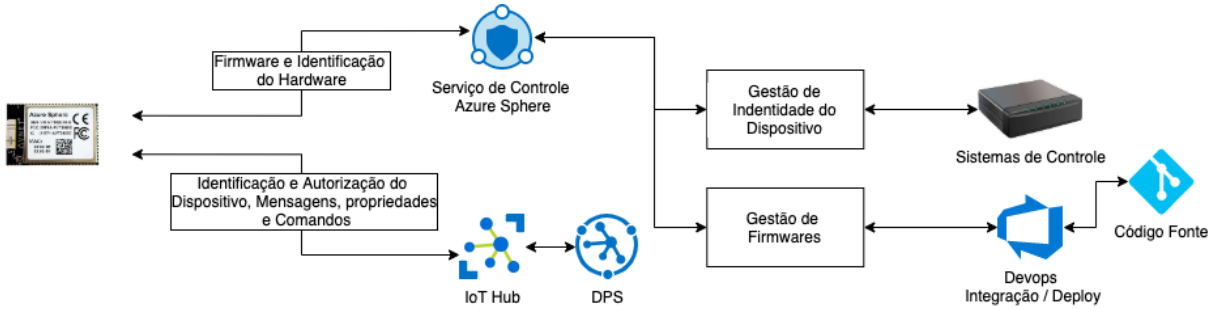

Figura 3.18: Comportamento da Solução Azure Sphere

- **Escalabilidade** - A escalabilidade pode ser tanto vertical quanto horizontal na maioria dos serviços dentro deste modelo, estes por sua vez são preparados para serem elásticos, fica aqui uma ressalva sobre os limites dos serviços e o conhecimento de como orquestrar esse crescimento. Uma das abordagens que mais gosto neste cenário é a do monitoramento e scripts de automatização para promover este crescimento, bem como ser proativo no retorno a patamares de tamanho menor quando a necessidade terminar. Um ponto perigoso neste caso de automatizar é que existe a necessidade de impor limites a esta inteligência, pois problemas de segurança como ataques podem forçar a um crescimento e estes devem ser mitigados de outra forma em algumas vezes. Durante os testes de implantação onde simulei dez mil dispositivos, mais quatro deles físicos, tive que aumentar uma instância do IoT Hub durante a implantação para garantir a ingestão de mais de 400.000 mensagens por dia e pude voltar no outro dia fazendo os custos se reduzirem a carga normal, neste caso de 4 dispositivos, operar normalmente. Esta situação foi alarmada pelo monitoramento provido pelo próprio serviço por meio de indicadores, neste caso usei o de número de mensagens no próprio painel de controle, fiz de forma manual, mas como citei acima pode-se automatizar facilmente com

leitura dos logs e ações tomadas a partir disso.

- **Segurança** - A segurança em PaaS está encapsulada nos componentes de serviço e cada qual tem sua abordagem, podendo ser integrados a um centro de segurança no próprio Azure. Outro ponto importante que foi incluído na sugestão de arquitetura aplicada a esta modalidade é a gestão de código, esta por sua vez pode identificar falhas que estejam sendo geradas por ataques e o controle da situação pode ser atingido mais cedo.

3.2.6.3 Projeto de Telemetria no modelo de Software como Serviço (SaaS) do Azure

No modelo de Software como serviço, a camada de aplicação se simplifica conforme pode ser visto na figura 3.19, agora a única preocupação do desenvolvimento é com o cliente do serviço implantado dentro do dispositivo e se houver necessidade de integração com sistemas especialistas com esse acoplamento. Os serviços expostos na implementação de PaaS são então encapsulados e as funcionalidades incluídas dentro do IoT Central que a partir de agora é a única porta de entrada e gestão de todos os dispositivos, naturalmente saídas e integrações são possíveis, feitas praticamente da mesma forma, porém sem a necessidade de uma instância de diversos serviços. Para o provisionamento de dispositivos tem-se as mesmas possibilidades de uso do serviço de provisionamento seguro e controle do ciclo de vida, como também o uso de *strings* de conexão. O custo do serviço se dá por dispositivo mês, com um limite de mensagens, que pode ser adicionado por pacotes extras, o trabalho então fica por conta de provisionar o dispositivo e usar o serviço para definir as necessidades de processamentos e alertas. Uma visão de análise de dados e gestão dos dispositivos pode ser vista nas figuras 3.20 e 3.21. Nesta implementação utilizamos dispositivos de telemetria baseados no SDK do IoT Hub como pode ser visto na figura 3.22, mas o serviço permite tanto este tipo de dispositivos, incluindo Azure Sphere e Edge Devices com módulos e inteligência na ponta.

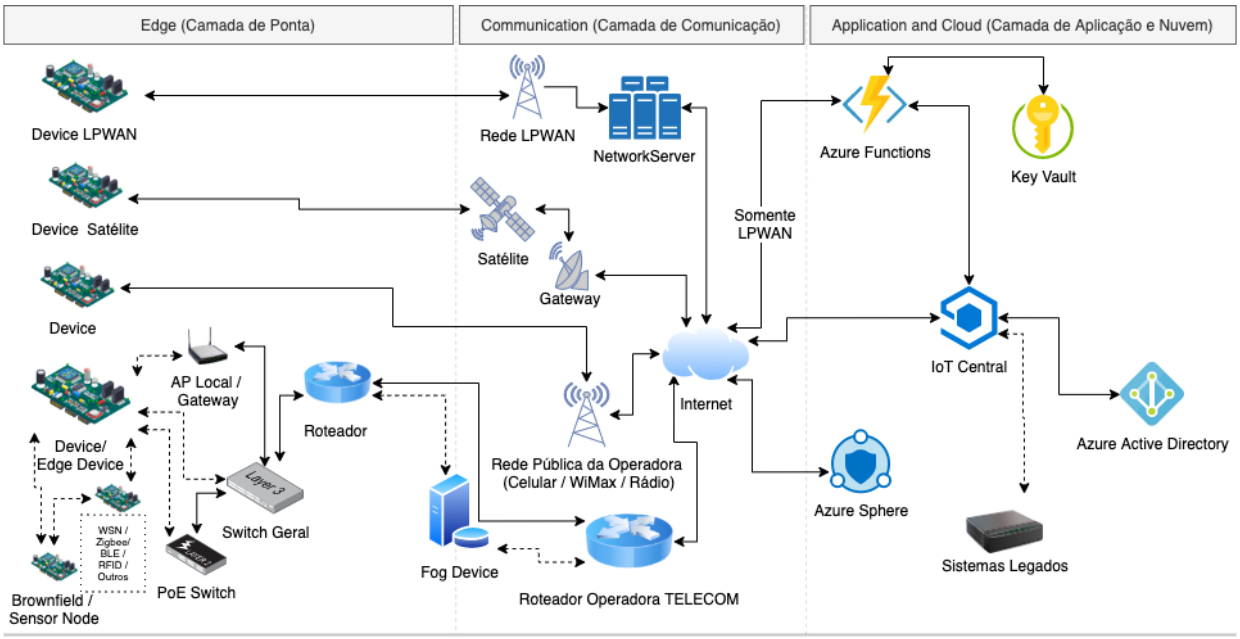

Figura 3.19: Diagrama de Arquitetura Aplicada em modelo de Software como serviço no Azure.

Considerações na implantação da arquitetura no modelo de SaaS feita segundo a figura 3.19:

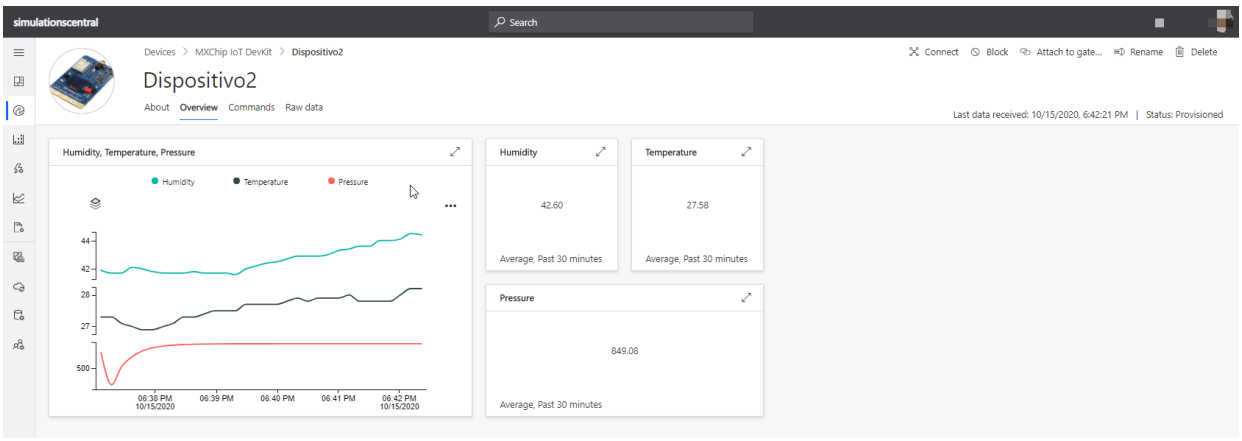

Figura 3.20: Visualização de dados enviados pelo dispositivo para o IoT Central por meio de componentes prontos e baseados no *payload* do dispositivo.

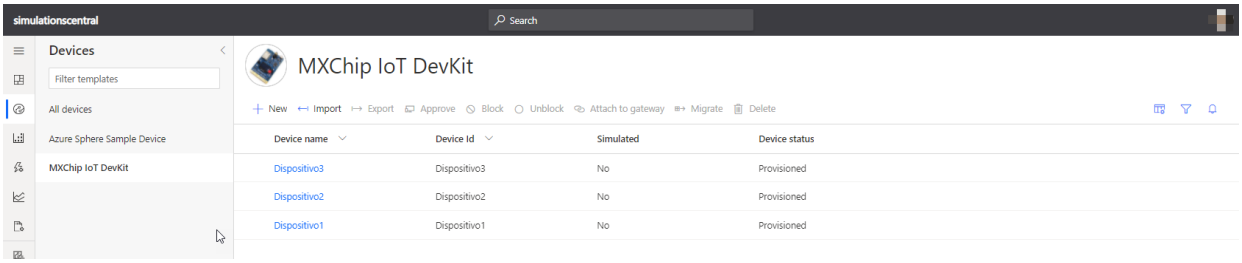

Figura 3.21: Dispositivos criados no IoT Central e com uma gestão simplificada pela plataforma.

- **Flexibilidade** - O modelo SaaS é conhecido por ser muito pouco tolerante a flexibilidade, no caso do IoT Central, serviço foco desta implantação, existe uma relativa flexibilidade, pois tem-se um serviço bem preparado que encapsula diversos componentes PaaS como funcionalidade, mesmo assim depende-se de implementação da Microsoft, o que pode engessar o uso diferente do PaaS e IaaS.
- **Controle** - Não se tem controle da execução nem ligação, depende-se do que é fornecido.
- **Complexidade de Implantação** - Zero de complexidade, pois o serviço guia o uso por meio de interfaces simples, oferece inclusive a possibilidade de criar toda uma rede de dispositivos simulados sem hardware para provas de conceito, o que facilita muito etapas iniciais de projetos de Internet das Coisas.
- **Escalabilidade** - A escalabilidade é definida pelos limites do serviço e o usuário não tem nenhum poder de alterar.
- **Segurança** - A segurança é simplificada para o usuário que agora se preocupa somente com integrações e a segurança da camada de ponta, todo o resto é tratado internamente no serviço, mas cabe atenção nos elos de interligação com aplicações externas.

Figura 3.22: Fotos dos dispositivos físicos usados na Aplicação da arquitetura proposta usando o serviço IoT Central no modelo de Software como Serviço (SaaS) no Azure.

3.2.6.4 Projeto de IoT com uso de Edge Computing usando o modelo de Plataforma como Serviço (PaaS) do Azure

O uso de Edge Computing é muito comum em cenários onde precisa-se de computação na ponta, próximo a sensores, dentro de redes internas e por vezes dividido em vários dispositivos na ponta com esta implementação. Quanto a arquitetura a mudança é feita na camada de ponta, principalmente dentro do dispositivo, que agora se torna mais inteligente e com o mesmo controle da nuvem sobre ele. A Microsoft implementa o conceito de Edge Computing com o IoT Edge, um framework baseada em contêineres que permite a gestão dos módulos, contêineres com aplicações, diretamente pelo IoT Hub, garantindo a flexibilidade, robustez e facilidade desta implementação. Os módulos podem ser desenvolvidos e implantados por meio de *scripts* de automação na nuvem. Neste ponto tem-se a única modificação feita da arquitetura de PaaS para esta implantação usando Edge que foi a inclusão de um serviço chamado Azure Container Registry, responsável por armazenar as imagens que serão executadas na ponta como módulos. Uma visão da arquitetura com os serviços pode ser vista na figura 3.23. O dispositivo pode ser implantado dentro de um hardware específico para uma aplicação de IA que tenha GPUs, uma máquina virtual dentro de um servidor, um hardware industrial, um Gateway dentre outros, desde que o sistema operacional e o hardware sejam capazes de executar o serviço que se baseia em contêineres, seja um ambiente com Docker ou para escala maiores com Kubernetes.

Acaba sendo papel deste dispositivo, às vezes, ser um tradutor de protocolos físicos e por isso existem hardwares prontos no mercado que se comunicam com várias tecnologias, um exemplo aqui é por exemplo um Gateway de LoRaWAN que faça as vias de um Network Server como utilizado na implementação de PaaS, neste caso teríamos dispositivos tanto LoRa quanto LoRaWAN se comunicando com este hardware

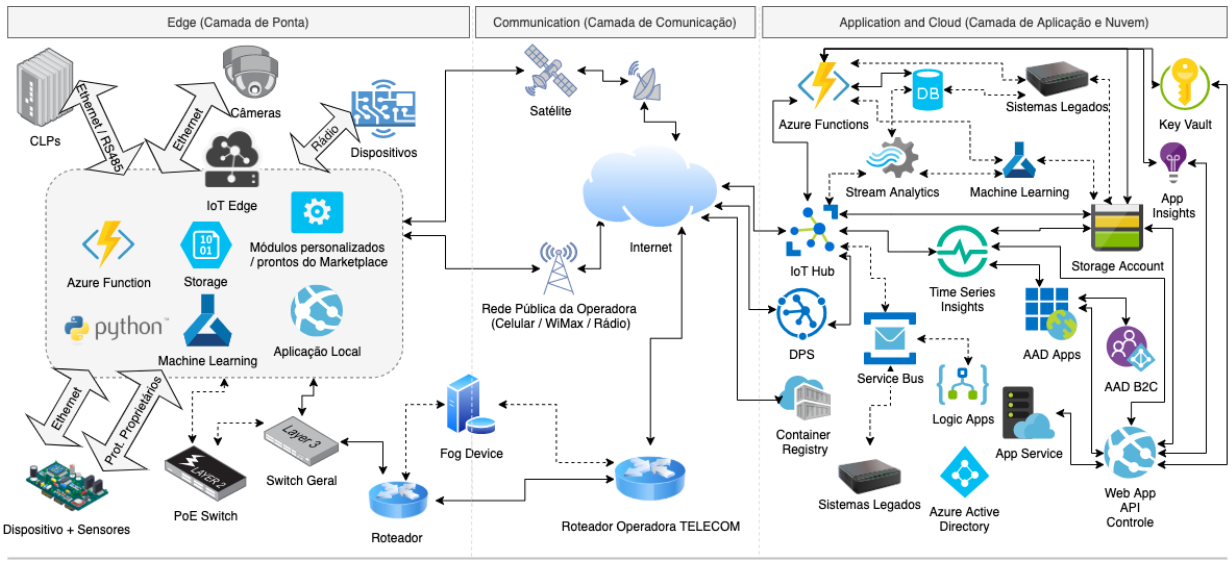

Figura 3.23: Diagrama de Arquitetura Aplicada, usando um *Edge Device* em modelo de Plataforma como serviço no Azure.

e o que antes era feito pelo Azure Functions na ponta, agora poderia ser feito aqui e o envio direto ao IoT Hub depois dessa tradução.

Na implementação de teste foi utilizado um medidor elétrico ligado a um transformador de corrente (figura 3.24 em um dos quadros de energia da CrazyTechLabs a fim de capturar a corrente passante imediata e também saber se existe uma sobre tensão ou uma subtensão na rede de uma das fases, o medidor entrega algumas dezenas de medições e análises, foram capturadas somente as duas. O IoT Edge implantado para esta coleta é composto por quatro módulos, sendo um de gestão do dispositivo, um de controle de mensagens, um para a comunicação Modbus que se integra diretamente ao conversor de RS485 ligado na USB do dispositivo e o último responsável por processar os dados recebidos e enviar para o IoT Hub. Toda parte de sincronia, segurança, mensageria, resiliência do dispositivo a falhas de energia e conectividade é tratada pelos dois módulos iniciais, que fazem parte do IoT Edge sem que tanha-se que executar nenhuma ação ou escrever nenhuma funcionalidade para tal.

O módulo de Modbus está disponível para uso sem custos e sem necessidade de desenvolvimento, seu funcionamento é o seguinte: ele faz leitura e escrita usando o protocolo Modbus tanto por meio de comunicação serial quanto encapsulado em TCP sobre Ethernet. Nesta implementação ele coleta dois registradores de 32bits cada, divididos em palavras de 16bits por endereço de consulta, isto significa que ao efetuar a leitura da corrente obtêm-se duas palavras de 16 bits em dois registradores e precisa-se transformar em uma de palavra somente, além disso o medidor aplica um multiplicador para que com isso ele tenha somente inteiros na comunicação, podemos ver na figura 3.25 como chegam os dados e como são enviados via mensagem para o módulo de tratamento. O Módulo de tratamento por sua vez ao receber as palavras procede com o tratamento e aplica o fator de divisão para que tenhamos o valor esperado e gera a saída para a próxima rota, que no caso é o IoT Hub, mas poderia ser uma IA de predição ou um armazenamento local.

As mensagens enviadas se utilizam de um padrão único para todas as medidas conforme podem ser visto na listagem 3.1, por causa disso se faz necessário um tratamento para demonstrar os dados das me-

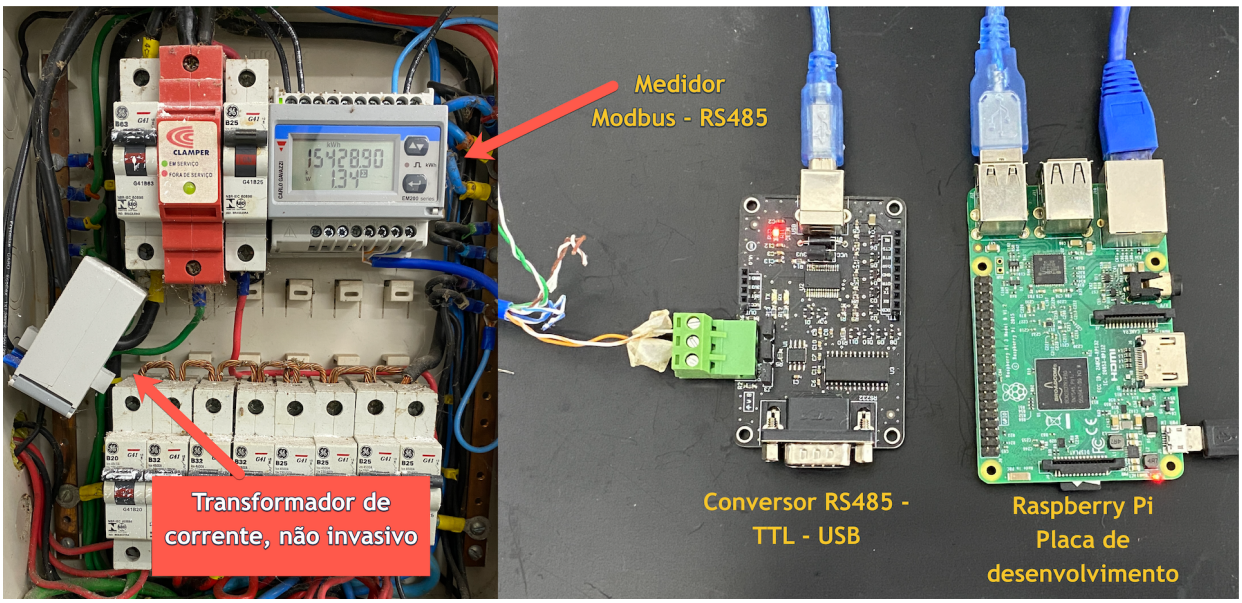

Figura 3.24: Dispositivo Edge e os componentes de medição e conversão usados.

didas de tensão e corrente da fase 1 enviados pelas medidas nomeadas de Voltagem fase 1 e Amperagem fase 1 nas mensagens, para isso pode-se utilizar uma funcionalidade do Time Series Insights que entende os dados e aplica um modelo ao dispositivo, conforme pode ser visto nas figuras 3.28, 3.27 e 3.26.

Listagem 3.1: Mensagens enviadas do módulo de processamento local do dispositivo

```
[IoTHubMonitor] [4:12:01 PM] Message received from [dispositivo-edge/
    ProcessamentoLocal]:
{
  "Medida": "Amperagem Fase 1 - (Atual)",
  "Hardware": "Medidor-Quadro-Principal",
  "DataHora": "2020-10-20 19:12:01",
  "ValorMedida": 8.5
}
[IoTHubMonitor] [4:12:08 PM] Message received from [dispositivo-edge/
    ProcessamentoLocal]:
{
  "Medida": "Voltagem Fase 1",
  "Hardware": "Medidor-Quadro-Principal",
  "DataHora": "2020-10-20 19:12:03",
  "ValorMedida": 218
}
```

Esta é uma extensão da abordagem de implantação da arquitetura no modelo de PaaS com a diferença de termos no dispositivo um IoT Edge, aumentando a capacidade de levar inteligência para ponta e garantindo uma resiliência maior em cenários onde precisa-se de um tempo de resposta menor ou que sofre com problemas de conectividade, dentre outros descritos anteriormente.

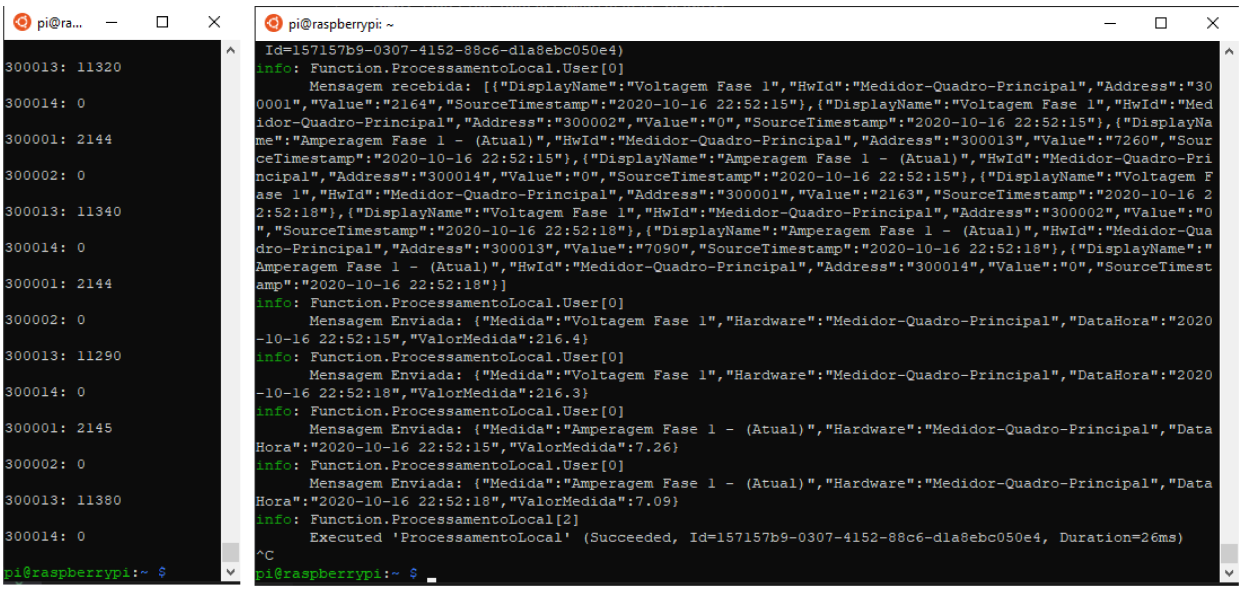

Figura 3.25: Mensagens recebidas do dispositivo em formato bruto e depois de tratamento enviadas ao IoT Hub dentro do módulo de processamento implementando uma Azure Function na ponta.

3.2.7 Limitações dos Recursos Apresentados

Nas implementações de SaaS e PaaS alguns serviços foram implementados e na tabela 3.2, os principais serviços estão listados e com seus respectivos limites. Para as soluções de PaaS o IoT Hub é o ingestor de mensagens, consequentemente devem ser observados seus limites, bem como o planejamento do número de instâncias ou a camada para uso, indiferente do uso de Edge Computing ou de simplesmente dispositivos de telemetria.(129) (130) (131) (132) Para os casos de uso de SaaS com o IoT Central, os limites estão a bem descritos a nível de número de mensagens, e como um Software como serviço, a bilhetagem traz o serviço cobrado de forma simples bem como seus limites de acordo com o objetivo do serviço que neste caso são as mensagens (133). Neste caso os limites do IoT Central são os de mensagens uma vez que tem-se todos os serviços praticamente encapsulados nele.

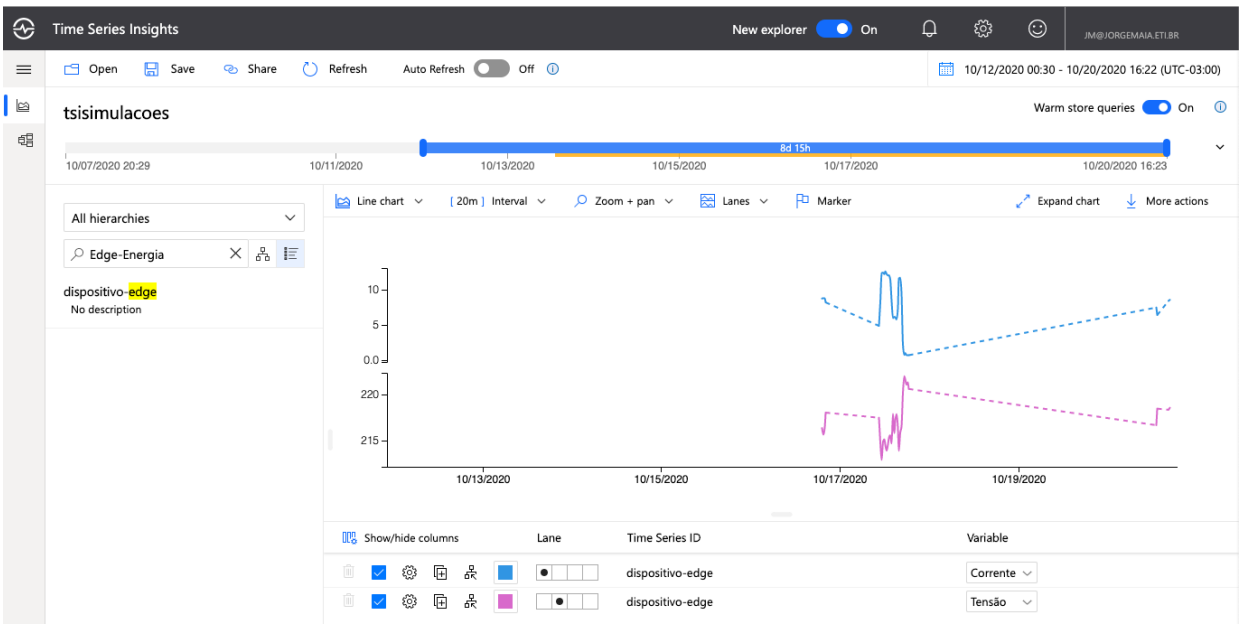

Figura 3.26: Gráfico de visualização de dados, interpolando dados quando eles não existem (linhas tracejadas).

Tabela 3.2: Limites de algumas Características de Recursos da Microsoft Azure (IoT Hub, IoT Central, Storage Account, Time Series Insights)

Recurso	**Característica**	**Limite**
IoT Hub	Tamanho máximo da mensagem (Dispositivo para Nuvem)	256 KB
	Tamanho máximo da mensagem (Nuvem para Dispositivo)	64 KB
	Número máximo de Dispositivos	1,000,000
	Throughput	Até 1111 KB/minuto (para Tier B1 e S1); Até 16 MB/minuto (para Tier B2 e S2); Até 814 MB/minuto (para Tier B3 e S3);
	Taxa de envio	278 mensagens/minuto (para Tier B1 e S1); 4.167 mensagens/minuto (para Tier B2 e S2); 208,333 mensagens/minuto (para Tier B3 e S3);
IoT Central	Tamanho da mensagem	4 KB
Storage Account	Capacidade máxima da conta de armazenamento	5 PiB
	Entrada Máxima	10 Gbps (regiões dos EUA e Europa), 5 Gbps (regiões fora dos EUA e Europa)
	Saída máxima para contas de armazenamento	50 Gbps (Todas as regiões)
Time Series Insights	Taxa de ingestão	1 Mbps
	Número máximo de propriedades	1000 (Gen2) para armazenamento quente e ilimitado para armazenamento frio

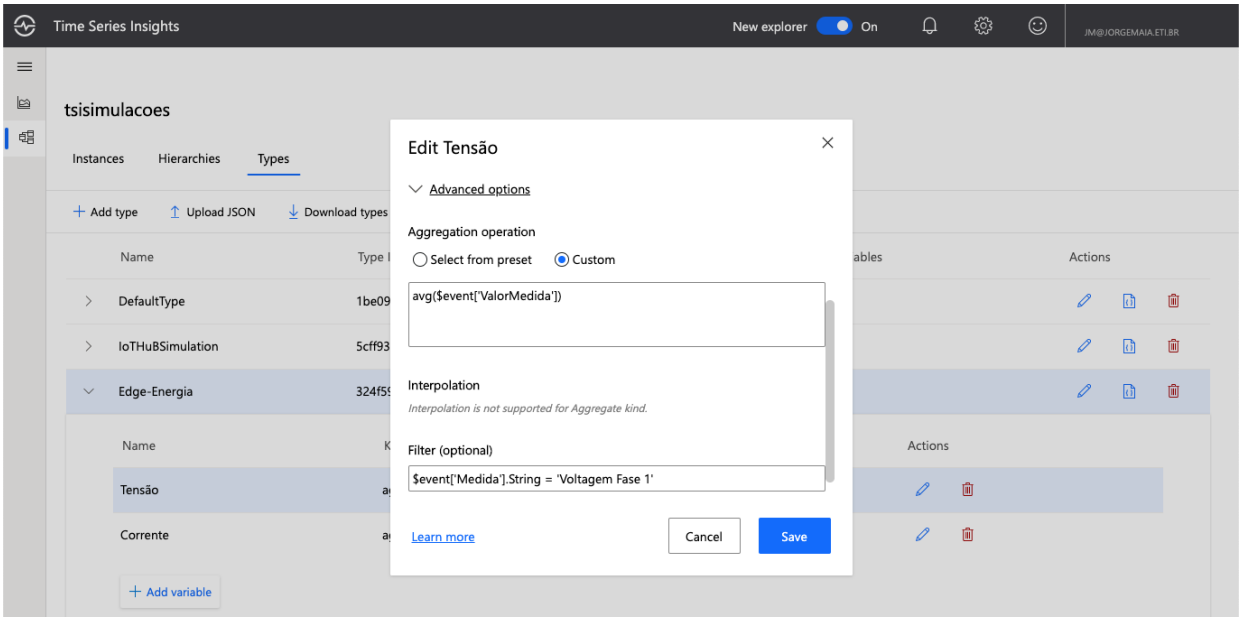

Figura 3.27: Definição das variáveis do tipo criado de acordo com o valores do *payload*, neste caso para tensão.

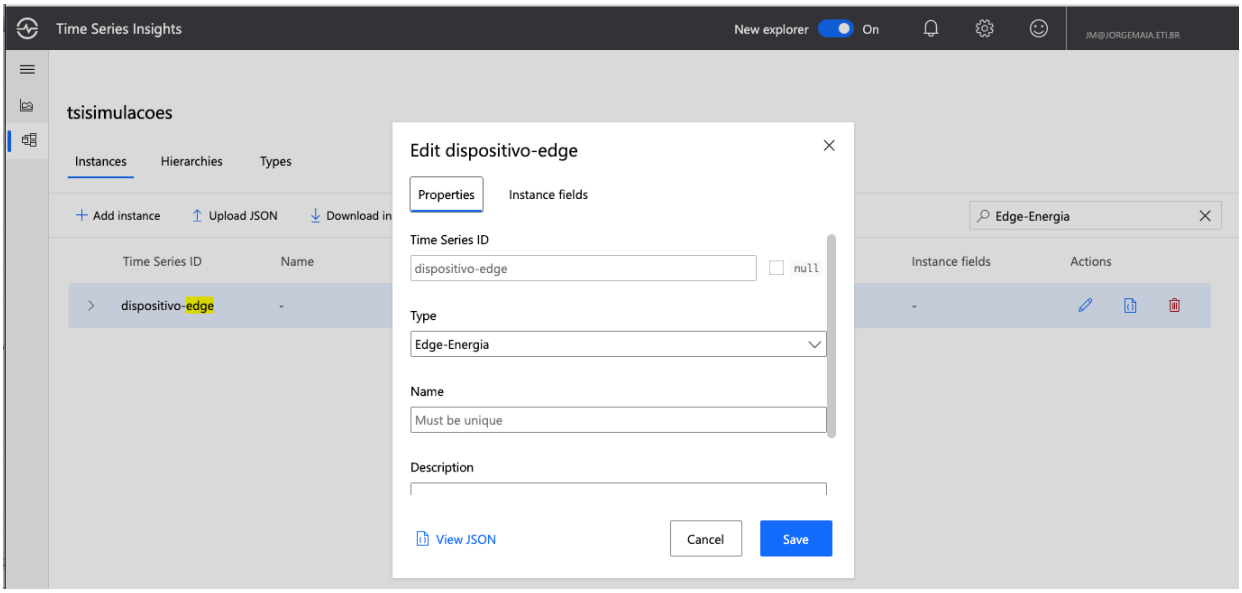

Figura 3.28: Atribuindo ao dispositivo o tipo personalizado que vai entender o *payload* e manipular os dados das séries temporais com essa nova definição.

4 CASOS DE USO REAIS DE SOLUÇÕES DE IOT

Este capítulo traz casos de uso reais de soluções de Internet das Coisas acompanhados e que implementam as arquiteturas e o uso combinado dos componentes apresentados no capítulo anterior. Todos os projetos apresentados a seguir foram desenvolvidos em clientes da CrazyTechLabs com a participação do pesquisador, alguns detalhes de implementação ou nomes foram omitidos por questões de confidencialidade entre as partes envolvidas, todos os direitos a seguir são da CrazyTechLabs e de seus clientes, o documento que autoriza a exposição dos casos está anexado a esta dissertação.

4.1 CASO DE USO DE PESAGEM E TELEMETRIA DE CAMINHÕES DE LIXO

Solução com desenvolvimento conjunto e consorciado onde juntaram-se a CrazyTechLabs, empresa brasileira, fundada em 2014, e a Engemon IT, integradora de soluções de tecnologia, fundada em 2017, para o projeto de pesquisa e desenvolvimento da solução de pesagem para caminhões de lixo, atendendo a demanda gerada pela Vital Engenharia, empresa do grupo Queiroz Galvão. Todo o desenvolvimento de pesquisa fora feito em Brasília-DF, em uma das sedes da CrazyTechLabs e os experimentos de campo na cidade de Recife-PE, dentro das dependências da Vital Engenharia, rotas dentro do estado e no aterro sanitário municipal. O pesquisador atuou como Arquiteto da solução, criando a arquitetura de hardware, software e comunicação, incluindo a descrição dos componentes, desenvolvimento de alguns módulos de software embarcado e em nuvem, e também a gestão dos profissionais envolvidos.

O projeto teve início em 2017 com a premissa de ter conhecimento de peso do caminhão em tempo próximo ao real, na estrada, sem uso de balanças ou sensores externos ao caminhão. A jornada de desenvolvimento teve como início uma busca por soluções que existissem no mercado tanto nacional quanto internacional, a qual resultou na inexistência de produtos prontos ou em desenvolvimento que atendessem a necessidade da pesagem fora da balança. Em 2020 após 3 anos de projeto, os resultados de campo e das provas de conceito executadas foram apresentados na 3RD IEEE International Conference on Industrial Cyber-Physical Systems, em junho de 2020, na universidade de Tampere, em Tampere na Finlândia.

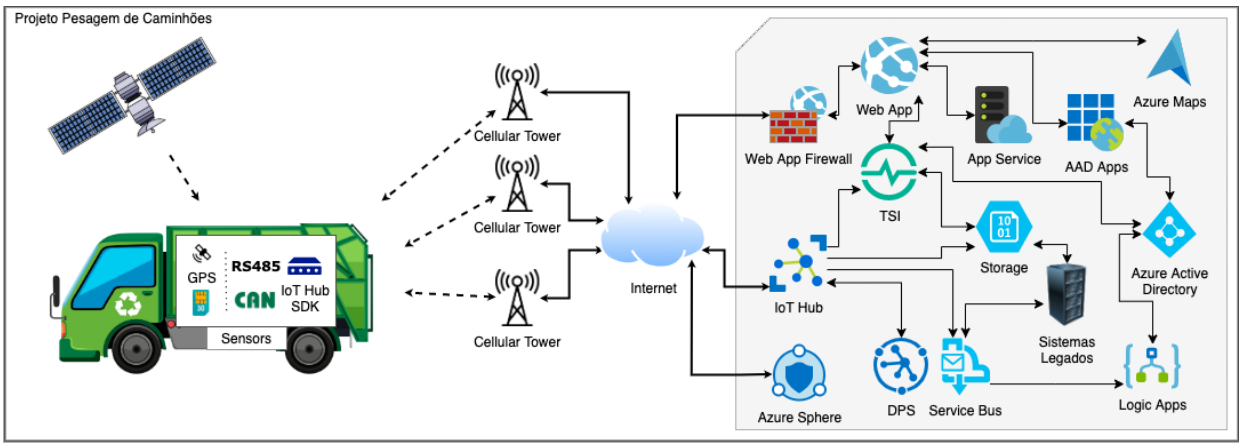

Figura 4.1: Arquitetura aplicada ao projeto de pesagem de caminhões.

O cenário do projeto acontece dentro da coleta de lixo, tarefa realizada no Brasil em sua grande maioria pela coleta com caminhões compactadores e descarregamento em aterro sanitário é feita por meio de setores e rotas pré-determinadas, com horários fixos e de cumprimento controlado e punitivo com multas em caso de falha ou mal prestação de serviços. As variáveis de complexidade podem ser climáticas, sazonais baseada em feriados, eventos, pandemias e outras, e estas modificam a carga a ser coletada e podem sofrer alterações não previsíveis, embora em sua maioria com dados tratados de forma correta podem evitadas ou previamente programadas.

Neste cenário a pesagem em rota é habilitadora para melhoria da operação com os seguintes tópicos:

- Trafegar dentro dos limites legais de carga para o veículo de acordo com sua composição de eixos e carroceria.
- Otimizar rotas com caminhões de acordo com carga e dados de passado para rota;
- Diminuir custos de manutenção por sobrepeso;
- Reduzir custos de operação (combustível, pneus, óleos e mecânica) por meio da otimização da frota
- Redução do custo de pessoal (horas extras, adicionais e outros) com otimização das rotas baseando-se na pesagem dos veículos ao longo do dia.

A solução atual é baseada em um microcontrolador com sensores espalhados pela carroceria e cabine, aquisição de dados via protocolo CAN e utiliza um algoritmo de cálculo para condensar os dados, histórico de medidas e regras de ajuste produzindo uma medida de peso com um *timestamp* que será enviada para a nuvem da Microsoft. Na arquitetura da solução, o principal orquestrador é o Azure IoT Hub, unido a alguns serviços para garantir desde a integridade do dispositivo do campo, tratado como um dispositivo de Internet das Coisas desde a concepção.

Atualmente, toda segurança atualmente é pautada na identificação deste dispositivo na rede de dispositivos, baseando-se em um serviço de provisionamento da própria Microsoft. Dados são validados para garantir que o dispositivo é ele mesmo durante sua existência e ciclo de conexão e reconexão.

Foram desenvolvidas mais de 15 versões de hardware de cabine e 4 de coleta de dados de sensores, feitas 3 visitas e testes em campo assistidos pelo time de desenvolvimento junto a Vital Engenharia e utilizadas mais de 200 horas de testes em laboratórios para ensaio de sensores e dos dispositivos. Uma equipe de 2 profissionais permanece desde o início do projeto e 18 profissionais de áreas distintas foram alocados, contratados, consultados ou prestaram assessoria até o momento.

A última mudança feita no projeto foi o redesenho do dispositivo de cabine para uso do Azure Sphere como orquestrador de todo o controle e comunicação visando um aumento de segurança na ponta e garantindo maior confiabilidade dos dados gerados para operação, uma visão da arquitetura atual pode ser vista na figura 4.1.

Como resultado deste projeto espera-se a melhoria na eficiência baseado em uso de insumos como combustível e pneus, manutenção e pagamento de horas extras que serão a partir do início do uso em toda frota, o quantitativo ainda está em estudo. Outro resultado direto será a queda de pagamento de multas por trafegar em rodovias acima do peso estabelecido para o veículo, este será de 100%, uma vez que o peso do caminhão será medido em rota e não mais somente no ponto final, garantindo além deste resultado uma otimização das rotas. O primeiro lote de implantação será de 265 caminhões previsto para 2021 e posteriormente a cobertura de todos os caminhões de outras operações de coleta o que pode totalizar um numero perto de 2 mil veículos.

4.2 CASO DE USO DE GESTÃO DE PRÉDIOS INTELIGENTES

O caso de uso de prédios inteligentes traz um cenário onde diversas variáveis de ambiente devem ser contabilizadas, analisadas e principalmente entendidas no contexto que se aplicam, durante o período da pesquisa alguns casos de uso foram acompanhados e eles seguem uma necessidade bem semelhante baseada principalmente em redução e otimização de custos seguida pela experiência de uso dos ambientes. Medições de ocupação, entendimento de uso, ativação e desativação de itens na planta como difusores de ar condicionado, controle de temperatura de ambientes, controle de iluminação, autorização de pessoas em locais restritos, alarmes de presença não autorizada, alertas de necessidade de limpeza dentre outros fazem parte do produto final de uma solução como esta, para tal, a CrazyTechLabs vem usando uma arquitetura de referência que se enquadra na proposta feita neste trabalho, composta por três camadas e com uso de *Edge Computing* neste cenário para diminuir o tráfego de dados e também garantir o pleno funcionamento dentro do ambiente indiferente de conectividade, uma vez que ao evoluir do monitoramento para o controle a solução torna-se indispensável para gestão e funcionamento do ambiente. O pesquisador atuou como Arquiteto, desenvolvedor e treinador dos times de clientes para as arquiteturas e desenvolvimento/gestão dos componentes utilizados.

Por questões de segurança e sigilo as integrações com os sistemas de controle e automação bem como os códigos fontes serão omitidos, toda a saída e integração com sistemas legados e ERPs também foi retirada do diagrama de arquitetura que contempla toda a arquitetura com exceção destas partes e pode ser visto na figura 4.2.

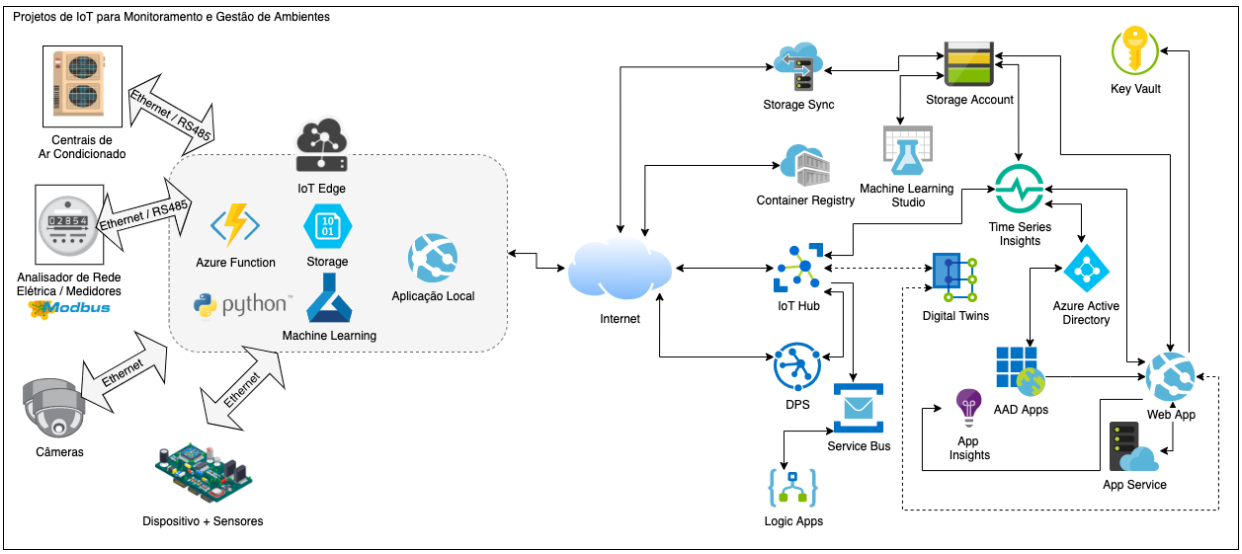

Figura 4.2: Arquitetura de referência para projetos de monitoramento de ambientes e gestão de espaços.

A parte de ponta neste caso é responsável principalmente pela interligação dos componentes de sensoriamento e atuação, neste cenário temos um alto volume de protocolos que, em sua maioria, falam diretamente com controladoras de automação e o IoT Edge da Microsoft implementa uma arquitetura interna baseada em contêineres e com gestão direta do IoT Hub o que permite que uma solução de base tenha versões diferentes em campo, gerenciadas de forma simples de acordo com versões de protocolos e equipamentos dispositivos nas plantas. Outro ponto importante neste cenário está na tomada de decisões baseadas em dados que acabaram de ser integrados ao sistema, aliada a esta necessidade tem-se, também, a carência de que essas decisões por vezes sejam tomadas baseadas em análises de dados anteriores e para isso treinamos uma Inteligência Artificial na nuvem com os dados anteriores e seus resultados para que um dos contêineres do IoT Edge seja responsável por essa inferência, este é um processo contínuo e requer tanto nuvem quanto ponta para esta orquestração, por isso temos uma sincronia de dados salvos na ponta com uma Storage Account no Azure acontecendo de tempos em tempos. O tráfego de mensagens entre ponta e nuvem se dá por meio de envios regulares com dados reduzidos para que uma camada de apresentação tome conta desta interface, para tal, o Time Series Insights tem sido o escolhido tanto para armazenamento e análise/enriquecimento do modelo inicialmente, quanto para consultas pela facilidade em criar hierarquias dentro da modelagem na entrada, agrupando dados de forma mais prática para a visualização e consultas, os outros componentes seguem a mesma implantação descrita na seção de PaaS da aplicação da arquitetura.

A união de Azure Functions e Módulos no IoT Edge também é feita para a leitura e entendimento de imagens de câmeras e sensores instalados nas edificações, como o IoT Edge opera como um ingestor de mensagens na ponta, em alguns cenários elencamos um deles para condensar dezenas de dispositivos e faz-se uma versão apropriada para esta finalidade diminuindo carga de uso de hardware, diferentemente do caso onde são processadas imagens e outros que demandam uso de GPUs por exemplo. Recentemente a Microsoft disponibilizou um serviço de Gêmeos Digitais e em um dos casos analisados de clientes da CrazyTechLabs o serviço foi incorporado a arquitetura conforme pode ser visto na figura 4.2, ele utiliza metadados dos dispositivos agrupando em hierarquias e promovendo um fator de facilitação em casos

de uso como o de prédios inteligentes onde temos hierarquias de andar, sala e local bem definidas, tal integração trouxe além de uma diminuição no tempo de desenvolvimento uma possibilidade de aplicar modelos tridimensionais com mais facilidade às soluções.

O projeto descrito nesta seção traz como resultado principalmente a economia de consumíveis, tanto se concessionárias de luz, água e gás, quanto de limpeza e outros da operação do prédio. A aplicação se fez em 35 localidades em duas empresas diferentes. O numero total de dispositivos é de 320, dentre eles alguns de telemetria ligados aos Edge Computing descritos. No quesito iluminação e consumo elétrico, tem-se economia da casa de 20%, os outros consumíveis ainda estão em teste em ambas implantações, uma delas com controle de insumos de limpeza para banheiros e na outra os dispositivos auxiliam na reposição de alimentação em refeitório e neste caso houve uma diminuição de desperdício com descarte baseado em contagem de pessoas por horário para reposição.

4.3 CASO DE USO INDUSTRIAL - LEITURA E ESCRITA DE DADOS EM CLPS COM COMUNICAÇÃO NÃO DISPONÍVEL DURANTE OPERAÇÃO

Os cenários fabris possuem uma característica muito semelhante ao apresentado nos casos de prédios inteligentes, contemplando integrações com dispositivos de automação, medidores de consumo, sensores de passagens, medições de temperatura, contadores e outros, mas com uma definição que engessa qualquer tipo de inovação que venha a ser proposta: o ambiente não pode parar. Esta definição trata diretamente do aspecto de redundância, controle de falhas, auto ajuste de linhas em tempo mínimo e muito mais, outra característica e que em processos de produção automatizados existem centenas de variáveis que impactam diretamente na produção, ela por sua vez gera alguns milhares de indicadores, alguns simples outros compostos que dependem de algoritmos que unem outros e tudo isso tem suma importância. O pesquisador atuou nos casos descritos a seguir como Arquiteto, desenvolvedor e treinador dos times de clientes.

Nos 2 casos analisados e acompanhados durante esta pesquisa em que fora implantada uma arquitetura semelhante a proposta anteriormente no capítulo de Simulações e Arquiteturas de Referência Propostas, dentro da seção de Edge Computing com PaaS, o uso se diferiu somente por causa das regras do processo em si, isto acarretou a escolha de componentes do Azure diferentes conforme descrito abaixo:

- **Planta de fábrica com alta segurança e com conectividade em janelas predefinidas** - Este caso trouxe uma necessidade de controlar uma linha de produção que operava em 2 turnos por dia com a missão de entender dados da linha, atuação em pontos de alarme ou configurações pontuais de velocidade de envase e controlar a quantidade de matéria prima. Foi utilizado um dispositivo de IoT Edge, rodando em sistema operacional Windows IoT Enterprise, com um módulo de armazenamento de informações, sincronizado quando tiver conectividade com a nuvem, um módulo de processamento de mensagens recebidas que recebe, entende, agrupa e gera alarmes, um módulo que resume dados em intervalos de tempo de 6 horas e envia para nuvem por meio do fluxo normal do IoT Hub, um cliente Web que apresenta as informações salvas em tempo real que permite envio de comandos para linha de produção, por fim um módulo cliente de OPC/UA que recebe dados do PLC ligado a linha

e também envia comandos para ele.

Na camada de nuvem os dados são ingeridos pelo IoT Hub e destinados ao salvamento direto em uma base de dados SQL Server para posterior apresentação em sistemas preexistentes por meio de consultas, é preciso citar que estes dados são somente uma pequena amostra enviada ao final de cada turno para compor um painel de macrovisão. Ainda na camada de aplicação também fora criada uma estrutura composta por uma base de dados não relacional que recebe uma carga quando a fábrica tem conectividade por meio de uma sincronia de arquivos vindos da ponta que depois do processamento fica disponível para visualização por meio de uma aplicação própria. Como os dados são enviados por sincronia e não se tem como precisar a janela, os registros são salvos na ponta com um *timestamp* local, com isso ao chegarem na nuvem temos o espelho no momento em que ocorreram para análise.

Neste cenário o número de indicadores e variáveis do ambiente chegam a alguns milhares, um volume de dados gerado na casa de dezenas de milhares de Bytes por semana, todo o ambiente é gerenciado remotamente por meio do IoT Hub que ao receber sinal de conectividade na ponta orquestra toda a atualização das imagens dos módulos, propriedades dos módulos e busca todo histórico de *logs* para avisos de necessidade de manutenção ao time de desenvolvimento. A sincronia do armazenamento local se dá pelo Storage Sync do próprio Azure, que coordena para que as contas de armazenamento de nuvem e do módulo da ponta estejam iguais em conteúdo.

Neste caso de uso o resultado se deu pela melhoria na IHM (Interface Homem Maquina) que antes era feita pela tela de um dispositivo acoplada a máquina e requeria um operador próximo ao equipamento, após a implantação, os dados do PLC eram lidos e com possibilidade de escrita também, agora de forma remota, permitindo que um operador ou o gerente do time pudesse operar mais de um equipamento com um tempo de resposta de milissegundos. Neste projeto foram implantados 2 dispositivos que coletavam de mais de uma linha de produção e toda as informações eram condensadas e enviadas para nuvem, com isso trazendo a possibilidade de acompanhamento remoto da planta, para 2021 estão previstas novas implantações utilizando a mesma arquitetura e componentes.

- **Planta fabril de com necessidade de IA** - Neste caso de uso que tinha como necessidade alguns entendimentos da linha de produção por meio de câmeras e sensores instalados na planta foi utilizado um dispositivo baseado em IoT Edge instalado dentro de um servidor com poder de processamento por meio de GPUs e sistema operacional Linux. O dispositivo era composto por 4 módulos além dos que compõem o IoT Edge. Um módulo destinado a receber e processar os dados das câmeras do circuito fechado, outro que se conectava com os dispositivos espalhados e recebia os dados dos sensores de passagem e outros, um terceiro que recebia dados de antenas RFID por meio de *socket* nos coletores ligados na rede local via Ethernet e no quarto uma instância do Stream Analytics que condensava todas as mensagens e criava mensagens específicas em janelas de tempo predefinidas.

  As mensagens eram enviadas diretamente sem armazenamento local e no IoT Hub eram roteadas de acordo com o tipo para Azure Functions que as salvava hora em bases de dados ligadas a sistemas de produção, hora em contas de armazenamento para uso em IA e consultas posteriores. Todos os alarmes e integrações com os sistemas de campo vinham dos sistemas especialistas e para o dispositivo IoT Edge o tráfego da nuvem era basicamente para mudanças de versões dos módulos e características de envio parametrizadas.

Este caso trouxe um ganho nos PCCs (Pontos Críticos de Controle) com o uso de câmeras do próprio circuito de monitoramento existentes depois da implantação foi possível verificar problemas com itens nas linhas, pessoas em áreas não permitidas e uma modificação foi feita para também efetuar contagem de itens com defeito gerando alertas ao gerente de produção da linha, com isso o tempo de resposta a incidentes foi reduzido e a aplicação tornou-se parte integrante do departamento de qualidade como um indicador de possíveis problemas o que antes era feito de forma mais espaçada e manualmente baseada em períodos de tempo somente. O projeto contou com a aquisição de 23 câmeras, sensores e dois dispositivos de Edge Computing implantados em uma unidade.

5 CONCLUSÃO

Neste trabalho foi apresentada uma arquitetura base de IoT eficiente e de simples implementação com três camadas, genérica o bastante e com componentes mínimos a fim de garantir a aplicabilidade em diferentes tipos de aplicação. A arquitetura se concentra em dividir os componentes por localização e com isso torna-se mais simples e menos complexa no quesito extensibilidade permitindo que alguns componentes estejam em uma ou outra camada de acordo com a aplicação. A divisão de ponta, comunicação e nuvem/aplicação se provou concisa e adaptou-se bem as implementações demonstradas bem como em aplicações reais com cenários de aplicação e escala diferentes umas das outras. Os componentes descritos nas camadas, necessários aos diferentes tipos de aplicação, podem ser utilizados nos cenários de telemetria, bem como onde existe a necessidade de computação na ponta com Edge Computing. A implantação foi demonstrada e validada com uso da nuvem pública da Microsoft, o Azure, e a eficiência observada e comprovada tanto nos testes de implantação descritos, quanto nos projetos reais em conjunto com a CrazyTechLabs, trazendo como resultado o ganho de tempo de desenvolvimento e também a possibilidade de aplicação em outras nuvens com serviços que contemples os componentes definidos.

No âmbito dos dispositivos, foi mostrado o funcionamento de alguns tipos de comunicação mais comuns na adoção de Internet das Coisas e demonstrada a integração de fim a fim, tanto de dispositivos IP quanto os não IP LPWAN, utilizando LoRaWan integrado ao ambiente do Azure com segurança e garantia de que o dispositivo está presente na ponta. Olhando de forma geral a Internet das Coisas, foi apresentada uma proposta de arquitetura em três camadas, com componentes mais comuns e necessários incluindo suas implantações nos modelos IaaS, PaaS e SaaS. Os detalhes de implementação foram validados pelos casos de uso reais apresentados e trazem a discussão dos limites, caminhos seguidos e principalmente possibilidades.

Foram feitos testes com uso de infraestrutura construída com IaaS e envio de dados simulados com uso de MQTT que trouxeram resultados satisfatórios e provaram o funcionamento, porém com a ressalva de escala no modelo que tem dificultadores para escalabilidade horizontal. No modelo de plataforma como serviço os testes feitos com uso do IoT Hub tratando da ingestão de mensagens e gestor dos dispositivos se provou eficiente e neste modelo um dos pontos altos demonstrados foi a facilidade de escala e de desenvolvimento, neste caso foi introduzida a comunicação de dispositivos que usam LPWAN e demonstrada como seria a ponte de comunicação entre os servidores de rede e a nuvem da Microsoft. Um dos pontos trazidos no trabalho foi a diferenciação de telemetria dos diferentes dispositivos, incluindo os que precisam de maior poder computacional, demonstrados e comprovados dentro da arquitetura proposta por meio de um dispositivo de IoT que monitorava o consumo e funcionamento da rede elétrica com transformação de dados na ponta, em ambos os modelos de dispositivos com PaaS, um dos serviços demonstrados para apresentação de dados e tratamento básico de informações trouxe a facilidade de desenvolvimento para o cenário de aplicação. Por último demonstrado um dispositivo com uso da camada de aplicação e nuvem totalmente encapsuladas em um único componente, o IoT Central, que trata todo o plano de execução de nuvem em um produto como serviço, embora com possíveis dificuldades de personalização, também se demonstra bem aderente a ideia de escala no que tange número de dispositivos e na facilidade de implantação

e gestão.

Toda pesquisa apresentada sustenta a proposição de que a nuvem é um elemento essencial para a IoT, sem ela não existe a possibilidade de escala com custos adequados e no momento necessário. Outro ponto apresentado na arquitetura proposta vem de encontro ao tipo de solução generalista que atende a todas as necessidades, este é um ponto delicado, que fora comprovado ser difícil de existir por este estudo, salvo se o destino seja um ramo de atividades específico, neste caso pode se ter uma solução única, às vezes como serviço, excluindo-se este cenário uma arquitetura generalista pronta não se sustenta. Nos casos de uso reais acompanhados foram apresentadas soluções que mesmo usando IoT Edge como motor da ponta, tinham composições totalmente diferentes nos módulos, rotas de mensagens, tempo de sincronia etc.

A prova de que a nuvem como fator de produtividade e tempo de entrada no mercado são um importante resultado também, mas dentro disso é preciso levantar uma discussão de que para o funcionamento e a escalabilidade estarem aderentes ao retorno de investimento é preciso avaliar o melhor serviço e seu uso em uma determinada necessidade, também é preciso trazer a necessidade do entendimento prévio do volume do projeto, quanto ele vai gerar em volume de dados, quanto vai precisar de processamento, qual a melhor forma de efetuar as predições, principalmente se discutir e implementar segurança, pois ela pode ser uma variável de geração de custos para o futuro do projeto.

Os modelos de serviço IaaS, PaaS e SaaS foram apresentados como opções, e todos acabam por ser complementares, sendo o PaaS é uma das melhores escolhas para projetos de porte maior e com muitas personalizações, enquanto o uso de SaaS com o IoT Central é a melhor opção para uma inserção imediata do produto no mercado, se tornando um forte aliado a provas de conceito e validações prévias em projetos maiores, já o IaaS é a opção mais custosa em relação de tempo e complexidade, mas pode ser aliada ao modelo de PaaS para serviços altamente especializados, bem como ser utilizada de forma isolada em projetos que necessitem de um controle total.

5.1 PRÓXIMOS PASSOS

Existem muitas lacunas quando falamos em unir componentes em uma arquitetura de Internet das Coisas, seja na ponta ou na camada de aplicação, muitas variáveis surgem da combinação ou não de serviços e componentes. A indústria que fabrica os insumos, sejam eles de hardware, software ou serviços, sempre vai tender a aumentar o seu *market share* com combinações de tecnologias favoráveis ao consumo. Muitos hoje falam em padronização, as pesquisas e experiência nos projetos de campo oferecem uma visão de que uma padronização pode não ser a saída. Hoje a Internet das Coisas é na verdade um grande guarda-chuva de tecnologias e esta combinação além de motivar o crescimento dos profissionais, permite que projetos com combinações certas saiam do papel de acordo com o cenário aplicado e atinjam os resultados esperados, caso existisse um engessamento geral isto poderia ser restringido. Ainda sobre padronização, é vista como benéfica quando restrita a áreas específicas como em um plano de segurança e privacidade, e também na definição de linhas gerais e de implantação de protocolos como já existe hoje em dia. Dentro de todas estas necessidades traz-se como possíveis próximos passos:

- Aplicar todos os componentes propostos nas implantações e fazer uma comparação entre os cenários

de uso de cada um dentro de sua aplicação proposta na arquitetura, cabe um estudo mais apurado sobre os limites em casos de uso distintos frente ao que foi apresentado.

- Um estudo aprofundado dos limites da camada de aplicação, principalmente quando se tem componentes em nuvens de fabricantes diferentes ou a combinação de pública e privada
- Testes de Stress em todas as implementações visando um referencial de trabalho para escalar
- Aplicações usando dispositivos com IoT Edge utilizando um hardware de coleta de diversas tecnologias de comunicação, trazendo como resultado uma medição de carga máxima de mensagens comparada a infraestrutura de hardware.
- Aplicação da arquitetura proposta para Edge Computing com PaaS em *gateways* de redes de sensores tratando todos os dispositivos com a funcionalidade de dispositivos gêmeos do IoT Hub, incluindo: Dispositivos filhos, não IP ou *browfield devices*.

REFERÊNCIAS BIBLIOGRÁFICAS

1 ALAM, M.; NIELSEN, R. H.; PRASAD, N. R. The evolution of M2M into IoT. *2013 1st International Black Sea Conference on Communications and Networking, BlackSeaCom 2013*, p. 112–115, 2013.

2 LEA, P. *IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security*. Birmingham: Packt Publishing Ltd., 2020. ISBN 978-1-83921-480-6.

3 BANAFA, A. *4 Challenges for the Internet of Things for it to Be a Success*. 2020. Disponível em: <https://datafloq.com/read/4-Challenges-Internet-Things-be-success/1399>.

4 STATISTA. • *Brazil IoT revenue 2021 | Statista*. 2020. Disponível em: <https://www.statista.com/statistics/968794/internet-of-things-revenue-brazil/>.

5 David Excoffier. *(8) D.E. on Twitter: "Internet of Things #Alliances and #Consortia #Timeline - #IoT is Accelerating - (tell me if some are missing) https://t.co/sxcrgM0soH" / Twitter*. 2020. Disponível em: <https://twitter.com/dexcoffier/status/689756518506446848/photo/1>.

6 FLEISCH, E. What is the Internet of Things? An Economic Perspective What is the Internet of Things - An Economic Perspective. *Economics, Management, and Financial Markets*, v. 5, n. 2, p. 125–157, 2010. Disponível em: <www.autoidlabs.org>.

7 SILVA, S.; EMILIA, M.; LEITÃO, S.; PRIESNITZ, C.; BEZERRA, M. View of Technological Prospecting: A Mapping of the Patent Applications Related of Internet of Things. n. 9, p. 155–170. Disponível em: <https://asrjetsjournal.org/index.php/American{_}Scientific{_}Journal/article/view/583>.

8 Jonny Leroy. *WE'RE IN A VIRTUOUS CYCLE*. 2020. Disponível em: <https://www.slideshare.net/ThoughtWorks/jonny-leroy-dataprivacyv2/2-WERE{_}IN{_}A{_}VIRTU>.

9 BNDES. Relatório do Plano de Ação. p. 1–65, 2017. Disponível em: <https://www.bndes.gov.br/wps/wcm/connect/site/269bc780-8cdb-4b9b-a297-53955103d4c5/relatorio-final-plano-de-acao-produto-8-alterado.pdf?MOD=AJPERES{&}CVID=m0jD>.

10 RAY, P. P. A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences*, King Saud University, v. 30, n. 3, p. 291–319, 2018. ISSN 22131248.

11 MATTEO, D. *What's the Difference Between "Brownfield" and "Greenfield" IIoT Scenarios? | Machine Design*. 2017. Disponível em: <https://www.machinedesign.com/automation-iiot/article/21835522/whats-the-difference-between-brownfield-and-greenfield-iiot-scenarios>.

12 HEDI, I.; ŠPEH, I.; ŠARABOK, A. IoT network protocols comparison for the purpose of IoT constrained networks. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, p. 501–505, 2017.

13 FANG, W. D.; HE, W.; CHEN, W.; SHAN, L. H.; MA, F. Y. Research on the application-driven architecture in internet of things. *Frontiers in Artificial Intelligence and Applications*, v. 293, p. 458–465, 2016. ISSN 09226389.

14 SHARP, H.; KOLKMAN, O.; SOCIETY, I. Discussion Paper : An analysis of the " New IP " proposal to the ITU-T. *Discussion Paper: An analysis of the New IP proposal to the ITU-T*, n. September, p. 1–13, 2019.

15 WIRELESSHART, A. *Industrial wireless networks — comparing the standards : Part 2*. 2017. 1–13 p. Disponível em: <https://www.processonline.com.au/content/wireless/article/industrial-wireless-networks-comparing-the-standards-part-1-252423732>.

16 HARWOOD, T. *IoT Standards & Protocols Guide | 2019 Comparisons on Network, Wireless Comms, Security, Industrial*. 2019. 1–14 p. Disponível em: <https://www.postscapes.com/internet-of-things-protocols/>.

17 MEKKI, K.; BAJIC, E.; CHAXEL, F.; MEYER, F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, Korean Institute of Communications Information Sciences, v. 5, n. 1, p. 1–7, mar 2019. ISSN 24059595.

18 SINHA, R. S.; WEI, Y.; HWANG, S. H. *A survey on LPWA technology: LoRa and NB-IoT*. [S.l.]: Korean Institute of Communications Information Sciences, 2017. 14–21 p.

19 Carl Henning. *Ethernet is not a protocol - PI North America Blog*. 2020. Disponível em: <https://us.profinet.com/ethernet-is-not-a-protocol/>.

20 P. McDermott-Wells. What is Bluetooth. *IEEE Potentials*, v. 23, n. 5, p. 33–35, 2005.

21 SILABS. *Z-Wave Global Regions - Silicon Labs*. 2020. Disponível em: <https://www.silabs.com/wireless/z-wave/technology/global-regions>.

22 MANZOOR, J.; DRAGO, I.; SADRE, R. The curious case of parallel connections in HTTP/2. *2016 12th International Conference on Network and Service Management, CNSM 2016 and Workshops, 3rd International Workshop on Management of SDN and NFV, ManSDN/NFV 2016, and International Workshop on Green ICT and Smart Networking, GISN 2016*, p. 174–180, 2017.

23 KOKSAL, O.; TEKINERDOGAN, B. Feature-driven domain analysis of session layer protocols of internet of things. *Proceedings - 2017 IEEE 2nd International Congress on Internet of Things, ICIOT 2017*, p. 105–112, 2017.

24 Navdeep Singh Gill. *IoT Analytics Platform for Real-Time and Stream Processing - XenonStack*. 2020. Disponível em: <https://www.xenonstack.com/blog/iot-analytics-platform/>.

25 GHOSH, A. M.; GROLINGER, K. Deep Learning: Edge-Cloud Data Analytics for IoT. *2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE 2019*, IEEE, p. 1–7, 2019.

26 IIOT. *The Industrial Internet of Things Volume G1: Reference Architecture*. [S.l.], 2019.

27 ZHONG, C. L.; ZHU, Z.; HUANG, R. G. Study on the IOT architecture and gateway technology. *Proceedings - 14th International Symposium on Distributed Computing and Applications for Business, Engineering and Science, DCABES 2015*, p. 196–199, 2016.

28 CHEN, S.; XU, H.; LIU, D.; HU, B.; WANG, H. A vision of IoT: Applications, challenges, and opportunities with China Perspective. *IEEE Internet of Things Journal*, v. 1, n. 4, p. 349–359, 2014. ISSN 23274662.

29 LI, S.; XU, L. D.; ZHAO, S. The internet of things: a survey. *Information Systems Frontiers*, v. 17, n. 2, p. 243–259, 2015. ISSN 1387-3326. Disponível em: <http://link.springer.com/10.1007/s10796-014-9492-7>.

30 EMBITEL. *IoT Gateway Architecture and How an IoT Gateway Works*. 2020. Disponível em: <https://www.embitel.com/blog/embedded-blog/understanding-how-an-iot-gateway-architecture-works>.

31 AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, v. 17, n. 4, p. 2347–2376, 2015. ISSN 1553-877X. Disponível em: <https://ieeexplore.ieee.org/document/7123563/>.

32 Microsoft Security. *IoT Security Architecture | Microsoft Docs*. 2020. Disponível em: <https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-architecture>.

33 ABDUR, M.; HABIB, S.; ALI, M.; ULLAH, S. Security Issues in the Internet of Things (IoT): A Comprehensive Study. *International Journal of Advanced Computer Science and Applications*, v. 8, n. 6, 2017. ISSN 2158107X.

34 ALHAMEDI, A. H.; SNASEL, V.; ALDOSARI, H. M.; ABRAHAM, A. Internet of things communication reference model. *2014 6th International Conference on Computational Aspects of Social Networks, CASoN 2014*, p. 61–66, 2014.

35 CHEN, M.; MIAO, Y.; HAO, Y.; HWANG, K. Narrow Band Internet of Things. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 5, p. 20557–20577, sep 2017. ISSN 21693536.

36 KAY, J. A.; ENTZMINGER, R. A.; MAZUR, D. C. Industrial ethernet- overview and best practices. *IEEE Conference Record of Annual Pulp and Paper Industry Technical Conference*, p. 18–27, 2014. ISSN 01902172.

37 YU, H.; WANG, N.; YU, W.; LIU, Z. LED lighting internet of things system based on PoE. *ACM International Conference Proceeding Series*, p. 47–52, 2019.

38 COLLOTTA, M.; PAU, G.; TALTY, T.; TONGUZ, O. K. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Communications Magazine*, v. 56, n. 7, p. 125–131, 2018. ISSN 15581896.

39 NAIK, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*, 2017.

40 HUNT, G.; LETEY, G.; NIGHTINGALE, E. The Seven Properties of Highly Secure Devices. *Microsoft Research NExT Operating Systems Technologies Group*, p. 1–10, 2017. Disponível em: <https://www.microsoft.com/en-us/research/publication/seven-properties-highly-secure-devices/>.

41 VERONICA, V. *Timeline of (Internet of Things) IoT Malware - Version 1 — Stratosphere IPS*. 2020. Disponível em: <https://www.stratosphereips.org/blog/2020/4/26/timeline-of-iot-malware-version-1>.

42 K. Ashton. That ' Internet of Things ' Thing. *RFiD Journal*, v. 22, p. 97–114, 2009.

43 GABBAI, A. *Kevin Ashton Describes " the Internet of Things "*. 2015. Disponível em: <http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/?no-ist>.

44 ARM. The Route to a Trillion Devices. *ARM Community*, n. June, p. 1–14, 2017.

45 HADDUD, A.; DESOUZA, A.; KHARE, A.; LEE, H. Examining potential benefits and challenges associated with the Internet of Things integration in supply chains. *Journal of Manufacturing Technology Management*, v. 28, n. 8, p. 1055–1085, 2017. ISSN 1741038X.

46 LEE, I.; LEE, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, "Kelley School of Business, Indiana University", v. 58, n. 4, p. 431–440, 2015. ISSN 00076813. Disponível em: <http://dx.doi.org/10.1016/j.bushor.2015.03.008>.

47 GARTNER. *Gartner Reprint*. 2020. Disponível em: <https://www.gartner.com/doc/reprints?id=1-2434LPHV{&}ct=200903{&}.>

48 GARTNER. *Definition of Big Data - Gartner Information Technology Glossary*. 2012. Disponível em: <https://www.gartner.com/en/information-technology/glossary/big-data>.

49 SHAHID, N.; ANEJA, S. Internet of Things: Vision, application areas and research challenges. *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, Elsevier B.V., v. 10, n. 7, p. 583–587, 2017. ISSN 15708705. Disponível em: <http://dx.doi.org/10.1016/j.adhoc.2012.02.016>.

50 GAZIS, V.; SASLOGLOU, K.; FRANGIADAKIS, N.; KIKIRAS, P.; MERENTITIS, A.; MATHIOUDAKIS, K.; MAZARAKIS, G. Architectural blueprints of a unified sensing platform for the internet of things. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, p. 0–4, 2013. ISSN 10952055.

51 MWAKWATA, C. B.; MALIK, H.; ALAM, M. M.; MOULLEC, Y. L.; PARAND, S.; MUMTAZ, S. Narrowband internet of things (NB-IoT): From physical (PHY) and media access control (MAC) layers perspectives. *Sensors (Switzerland)*, MDPI AG, v. 19, n. 11, jun 2019. ISSN 14248220. Disponível em: </pmc/articles/PMC6603562/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603562/>.

52 Carnegie Mellon. *The Carnegie Mellon University Computer Science Department Coke Machine*. 2020. Disponível em: <https://www.cs.cmu.edu/{~}coke/history{_}lon>.

53 DAIMLER, E. A. A study of Food Consumption by Carnegie Mellon ' s Computer Science department (or : observations of Bagel decay). p. 1–4, 2020.

54 LIU, S. *Internet of Things (IoT) - Statistics & Facts | Statista*. 2020. Disponível em: <https://www.statista.com/topics/2637/internet-of-things/>.

55 ESPINOZA, H.; KLING, G.; MCGROARTY, F.; O'MAHONY, M.; ZIOUVELOU, X. Estimating the impact of the Internet of Things on productivity in Europe. *Heliyon*, Elsevier Ltd, v. 6, n. 5, may 2020. ISSN 24058440. Disponível em: <https://doi.org/10.1016/j.heliyon.2020.e03935>.

56 JIN, X. B.; YU, X. H.; WANG, X. Y.; BAI, Y. T.; SU, T. L.; KONG, J. L. Deep learning predictor for sustainable precision agriculture based on internet of things system. *Sustainability (Switzerland)*, v. 12, n. 4, 2020. ISSN 20711050.

57 AL-HAMADI, H.; CHEN, I. R. Trust-Based Decision Making for Health IoT Systems. *IEEE Internet of Things Journal*, v. 4, n. 5, p. 1408–1419, 2017. ISSN 23274662.

58 TERÉS-ZUBIAGA, J.; PÉREZ-IRIBARREN, E.; GONZÁLEZ-PINO, I.; SALA, J. M. Effects of individual metering and charging of heating and domestic hot water on energy consumption of buildings in temperate climates. *Energy Conversion and Management*, Elsevier, v. 171, n. June, p. 491–506, 2018. ISSN 01968904. Disponível em: <https://doi.org/10.1016/j.enconman.2018.06.013>.

59 CARRATÙ, M.; FERRO, M.; PACIELLO, V.; PIETROSANTO, A.; SOMMELLA, P. Performance analysis of WM-bus networks for smart metering. *IEEE Sensors Journal*, v. 17, n. 23, p. 7849–7856, 2017. ISSN 1530437X.

60 WU, Q.; DING, G.; XU, Y.; FENG, S.; DU, Z.; WANG, J.; LONG, K. Cognitive internet of things: A new paradigm beyond connection. *IEEE Internet of Things Journal*, v. 1, n. 2, p. 129–143, 2014. ISSN 23274662.

61 MICROSOFT. *Intelligent Edge – Future of Cloud Computing | Microsoft Azure*. 2020. Disponível em: <https://azure.microsoft.com/en-us/overview/future-of-cloud/>.

62 MACARTHUR, E. *The Virtuous Circle*. [S.l.]: European Investment Bank, 2019. 36 p. ISBN 9789286140518.

63 REVIEW, D. B.; RECEBIDO, O. J. S. Revista de Direito, Governança e Novas Tecnologias. *Revista de Direito, Governança e Novas Tecnologias*, v. 4, n. 1, p. 92–108, 2017. ISSN 0740624X.

64 ANGELO, M. *A onda da IoT no mar brasileiro*. 2018. 58 – 61 p. Disponível em: <https://web.bndes.gov.br/bib/jspui/bitstream/1408/15541/1/AondadaIoTnomarbrasileiroHSMn.126fev2018.PDF>.

65 BNDES. *Estudo "Internet das Coisas: um plano de ação para o Brasil" - Brasil, país digital - #BrasilPaisDigital*. 2020. Disponível em: <https://brasilpaisdigital.com.br/estudo-internet-das-coisas-um-plano-de-acao-para-o-brasil/>.

66 BISWAS, A. R.; GIAFFREDA, R. IoT and cloud convergence: Opportunities and challenges. *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, p. 375–376, 2014.

67 ZHANG, Z. K.; CHO, M. C. Y.; WANG, C. W.; HSU, C. W.; CHEN, C. K.; SHIEH, S. IoT security: Ongoing challenges and research opportunities. *Proceedings - IEEE 7th International Conference on Service-Oriented Computing and Applications, SOCA 2014*, p. 230–234, 2014.

68 STOYANOVA, M.; NIKOLOUDAKIS, Y.; PANAGIOTAKIS, S.; PALLIS, E.; MARKAKIS, E. K. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys & Tutorials*, IEEE, v. 22, n. 2, p. 1191–1221, 2020. ISSN 1553-877X. Disponível em: <https://ieeexplore.ieee.org/document/8950109/>.

69 WANG, T.; ZHANG, G.; BHUIYAN, M. Z. A.; LIU, A.; JIA, W.; XIE, M. A novel trust mechanism based on Fog Computing in Sensor–Cloud System. *Future Generation Computer Systems*, Elsevier B.V., v. 109, p. 573–582, 2020. ISSN 0167739X. Disponível em: <https://doi.org/10.1016/j.future.2018.05.049>.

70 XU, T.; WENDT, J. B.; POTKONJAK, M. Security of IoT systems: Design challenges and opportunities. *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, v. 2015-Janua, n. January, p. 417–423, 2015. ISSN 10923152.

71 NIŽETIĆ, S.; ŠOLIĆ, P.; López-de-Ipiña González-de-Artaza, D.; PATRONO, L. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, Elsevier Ltd, v. 274, p. 122877, 2020. ISSN 09596526. Disponível em: <https://doi.org/10.1016/j.jclepro.2020.122877>.

72 WILBURN, K. M.; WILBURN, H. R. The Impact Of Technology On Business And Society. *Global Journal of Business Research*, v. 12, n. 1, p. 23–39, 2018.

73 CONDRY, M. W.; NELSON, C. B. Using Smart Edge IoT Devices for Safer, Rapid Response with Industry IoT Control Operations. *Proceedings of the IEEE*, v. 104, n. 5, p. 938–946, 2016. ISSN 15582256.

74 SISINNI, E.; SAIFULLAH, A.; HAN, S.; JENNEHAG, U.; GIDLUND, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, IEEE, v. 14, n. 11, p. 4724–4734, 2018. ISSN 15513203.

75 FARHAN, L.; KHAREL, R.; KAIWARTYA, O.; QUIROZ-CASTELLANOS, M.; ALISSA, A.; ABDULSALAM, M. A Concise Review on Internet of Things (IoT)-Problems, Challenges and Opportunities. *2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2018*, IEEE, p. 1–6, 2018.

76 PIERLEONI, P.; CONCETTI, R.; BELLI, A.; PALMA, L. Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison. *IEEE Access*, IEEE, v. 8, p. 5455–5470, 2020. ISSN 21693536.

77 LINUXCOM. *Linux and Open Source Hardware for IoT - Linux.com*. Disponível em: <https://www.linux.com/news/linux-and-open-source-hardware-iot/>.

78 NVIDIA. *NVIDIA Jetson Nano Developer Kit | NVIDIA Developer*. 2020. Disponível em: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>.

79 FREERTOS. *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. Disponível em: <https://www.freertos.org/>.

80 NXP IoT. *Windows 10 IoT*. 2020. Disponível em: <https://developer.microsoft.com/en-us/windows/iot/>.

81 OREN eini. *RavenDB now supports running on ARM64 machines | RavenDB NoSQL*. 2020. Disponível em: <https://ravendb.net/articles/ravendb-now-supports-running-on-arm64-machines>.

82 SHI, W.; CAO, J.; ZHANG, Q.; LI, Y.; XU, L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, v. 3, n. 5, p. 637–646, 2016. ISSN 23274662.

83 SHI, W.; COMPUTING, E. The Promise of Edge Computing. IEEE, n. 0018, p. 17–20, 2016.

84 MICROSOFT. *Azure IoT Edge*. 2018. Disponível em: <https://azure.microsoft.com/en-us/services/iot-edge/>.

85 AMAZON. *AWS IoT Greengrass – Amazon Web Services*. Disponível em: <https://aws.amazon.com/pt/greengrass/?nc1=h{_}.>

86 SHI, W.; COMPUTING, E. The Emergence of Edge Computing. IEEE, n. 0018, p. 17–20, 2016.

87 BELLO, O.; ZEADALLY, S.; BADRA, M. Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). *Ad Hoc Networks*, Elsevier B.V., v. 57, p. 52–62, 2017. ISSN 15708705.

88 IPV6BR. *Endereçamento IPv6*. 2020. Disponível em: <http://ipv6.br/post/enderecamento-ipv6/>.

89 PACELLE, M. *3 topologies driving IoT networking standards - O'Reilly Radar*. 2020. Disponível em: <http://radar.oreilly.com/2014/04/3-topologies-driving-iot-networking-standards.html>.

90 SOPARIA, J.; BHATT, N. A Survey on Comparative Study of Wireless Sensor Network Topologies. *International Journal of Computer Applications*, v. 87, n. 1, p. 40–43, 2014.

91 MACEDONIO, D.; MERRO, M. A semantic analysis of key management protocols for wireless sensor networks. *Science of Computer Programming*, Elsevier B.V., v. 81, p. 53–78, 2014. ISSN 01676423. Disponível em: <http://dx.doi.org/10.1016/j.scico.2013.01.005>.

92 ZHANG, D. G.; ZHU, Y. N.; ZHAO, C. P.; DAI, W. B. A new constructing approach for a weighted topology of wireless sensor networks based on local-world theory for the Internet of Things (IOT). *Computers and Mathematics with Applications*, Elsevier Ltd, v. 64, n. 5, p. 1044–1055, 2012. ISSN 08981221. Disponível em: <http://dx.doi.org/10.1016/j.camwa.2012.03.023>.

93 BARRAT, A.; BARTHÉLEMY, M.; VESPIGNANI, A. Modeling the evolution of weighted networks. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, v. 70, n. 6, p. 12, 2004. ISSN 1063651X.

94 RAZA, U.; KULKARNI, P.; SOORIYABANDARA, M. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys and Tutorials*, v. 19, n. 2, p. 855–873, 2017. ISSN 1553877X.

95 HOGLUND, A.; LIN, X.; LIBERG, O.; BEHRAVAN, A.; YAVUZ, E. A.; Van Der Zee, M.; SUI, Y.; TIRRONEN, T.; RATILAINEN, A.; ERIKSSON, D. Overview of 3GPP Release 14 Enhanced NB-IoT. *IEEE Network*, Institute of Electrical and Electronics Engineers Inc., v. 31, n. 6, p. 16–22, nov 2017. ISSN 08908044.

96 TELETIME-DESLIGAMENTO. *Tecnologia 3G pode ser desligada antes da 2G, prevê Anatel | TELETIME News*. Disponível em: <https://teletime.com.br/22/05/2018/tecnologia-3g-pode-ser-desligada-antes-da-2g-preve-conselheiro/>.

97 SIMCOM. *SIM7090G Module LPWA Wireless Solutions | SIMCom Wireless Solutions Co.,Ltd.* Disponível em: <https://www.simcom.com/product/SIM7090G.html>.

98 UGARTE, L. F.; GARCIA, M. C.; ROCHETI, E. O.; LACUSTA, E.; PEREIRA, L. S.; De Almeida, M. C. LoRa Communication as a Solution for Real-Time Monitoring of IoT Devices at UNICAMP. *SEST 2019 - 2nd International Conference on Smart Energy Systems and Technologies*, IEEE, p. 1–6, 2019.

99 SIGFOX. *Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)*. 2016. Disponível em: <https://www.sigfox.com/en{\%}0Ahttp://www.sigfox.com/>.

100 SIGFOX. *Radio Configurations | Sigfox build*. 2020. Disponível em: <https://build.sigfox.com/sigfox-radio-configurations-rc>.

101 SINGH, B. A Comparative Study of Mobile Wireless Communication Networks and Technologies. *International Journal of Computer Science and Information Technology Research*, v. 2, n. October, p. 634–637, 2012.

102 RITESH, K. V.; MANOLOVA, A.; NENOVA, M. Abridgment of bluetooth low energy (BLE) standard and its numerous susceptibilities for Internet of Things and its applications. *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems, COMCAS 2017*, v. 2017-Novem, p. 1–5, 2017.

103 SILICON-LABS. *Z-Wave Security - Silicon Labs*. 2020. Disponível em: <https://www.silabs.com/wireless/z-wave/specification/security>.

104 ZWAVEFREQUENCIES. *Z-Wave Frequencies*. Disponível em: <http://z-wave-assets.s3-us-west-2.amazonaws.com/docs/677/Z-Wave{_}Frequency{_}Chart.pdf?1522>.

105 DATTA, P.; SHARMA, B. A survey on IoT architectures, protocols, security and smart city based applications. *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*, 2017.

106 MENDES, T. D.; GODINA, R.; RODRIGUES, E. M.; MATIAS, J. C.; CATALÃO, J. P. *Smart home communication technologies and applications: Wireless protocol assessment for home area network resources*. [S.l.: s.n.], 2015. v. 8. 7279–7311 p. ISSN 19961073. ISBN 3512753299.

107 MOHAN, V. An introduction to wireless M-Bus. p. 1–19, 2015.

108 FACCHINI, F.; VITETTA, G. M.; LOSI, A.; RUSCELLI, F. On the performance of 169 MHz WM-Bus and 868 MHz LoRa technologies in smart metering applications. *RTSI 2017 - IEEE 3rd International Forum on Research and Technologies for Society and Industry, Conference Proceedings*, 2017.

109 THOTA, P.; KIM, Y. Implementation and Comparison of M2M Protocols for Internet of Things. *Proceedings - 4th International Conference on Applied Computing and Information Technology, 3rd International Conference on Computational Science/Intelligence and Applied Informatics, 1st International Conference on Big Data, Cloud Computing, Data Science*, p. 43–48, 2017.

110 COAP. *CoAP — Constrained Application Protocol | Overview*. 2020. Disponível em: <https://coap.technology/>.

111 MQTT. *MQTT*. 2020. Disponível em: <https://mqtt.org/>.

112 AMQP. *Home | AMQP*. 2020. Disponível em: <https://www.amqp.org/>.

113 KASSAB, W.; DARABKH, K. A. A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations. *Journal of Network and Computer Applications*, Elsevier Ltd, v. 163, n. April, p. 102663, 2020. ISSN 10958592. Disponível em: <https://doi.org/10.1016/j.jnca.2020.102663>.

114 MEDVEDEV, A.; HASSANI, A.; ZASLAVSKY, A.; JAYARAMAN, P. P.; INDRAWAN-SANTIAGO, M.; HAGHIGHI, P. D.; LING, S. Data ingestion and storage performance of IoT platforms: Study of OpenIoT. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 10218 LNCS, p. 141–157, 2017. ISSN 16113349.

115 Microsoft IoT. *Azure IoT – Plataforma de Internet das Coisas | Microsoft Azure*. 2020. Disponível em: <https://azure.microsoft.com/pt-br/overview/iot/>.

116 AMAZON. *Visão geral do AWS IoT Analytics – Amazon Web Services*. 2020. Disponível em: <https://aws.amazon.com/pt/iot-analytics/>.

117 KHARE, S.; TOTARO, M. Big Data in IoT. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, 2019.

118 MPEIS, P.; ROUSSEL, T.; KUMAR, M.; COSTA, C.; LAOUDIASDENIS, C.; CAPOT-RAY, D.; ZEINALIPOUR-YAZTI, D. The Anyplace 4.0 IoT Localization Architecture. *Proceedings - IEEE International Conference on Mobile Data Management*, v. 2020-June, p. 218–225, 2020. ISSN 15516245.

119 JIANG, L.; XU, L. D.; CAI, H.; JIANG, Z.; BU, F.; XU, B. An IoT-Oriented data storage framework in cloud computing platform. *IEEE Transactions on Industrial Informatics*, v. 10, n. 2, p. 1443–1451, 2014. ISSN 15513203.

120 RAMON. Experimentando Arquiteturas de Internet do Futuro com Internet das Coisas Ramon Pereira dos Santos Chaib Dezembro de 2017. 2017. ISSN 0090-7421.

121 VERMA, S.; KAWAMOTO, Y.; FADLULLAH, Z. M.; NISHIYAMA, H.; KATO, N. A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues. *IEEE Communications Surveys and Tutorials*, v. 19, n. 3, p. 1457–1477, 2017. ISSN 1553877X.

122 AL-QASEEMI, S. A.; ALMULHIM, H. A.; ALMULHIM, M. F.; CHAUDHRY, S. R. IoT architecture challenges and issues: Lack of standardization. *FTC 2016 - Proceedings of Future Technologies Conference*, n. December, p. 731–738, 2017.

123 AI, Y.; PENG, M.; ZHANG, K. Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks*, Elsevier Ltd, v. 4, n. 2, p. 77–86, 2018. ISSN 23528648. Disponível em: <https://doi.org/10.1016/j.dcan.2017.07.001>.

124 Instituto CESAR. *:: KNoT Network of Things ::*. 2020. Disponível em: <https://knot.cesar.org.br/>.

125 BRASILAGRO. *Empresas apresentam aplicações de Internet das Coisas no agronegócio | Brasilagro*. 2020. Disponível em: <https://www.brasilagro.com.br/conteudo/empresas-apresentam-aplicacoes-de-internet-das-coisas-no-agronegocio.html>.

126 HUSSAIN, F.; HUSSAIN, R.; HASSAN, S. A.; HOSSAIN, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Communications Surveys and Tutorials*, IEEE, n. c, p. 1–38, 2020. ISSN 1553877X.

127 VASHI, S.; RAM, J.; MODI, J.; VERMA, S.; PRAKASH, C. Internet of Things (IoT): A vision, architectural elements, and security issues. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, 2017. p. 492–496. ISBN 978-1-5090-3242-6. Disponível em: <http://ieeexplore.ieee.org/document/8117893/http://ieeexplore.ieee.org/document/8058399/>.

128 APOSEMAT. *Aposemat IoT Project — Stratosphere IPS*. 2020. Disponível em: <https://www.stratosphereips.org/aposemat>.

129 MICROSOFT. *Limitações de taxa de transferência de ingestão de streaming-Azure Time Series Insights Gen2 | Microsoft Docs*. 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/time-series-insights/concepts-streaming-ingress-throughput-limits>.

130 MICROSOFT. *Metas de desempenho e escalabilidade do Arquivos do Azure | Microsoft Docs*. 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/storage/files/storage-files-scale-targets{\#}azure-storage-account-scale-targ>.

131 MICROSOFT. *Understand Azure IoT Hub quotas and throttling | Microsoft Docs*. 2020. Disponível em: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-quotas-throttling{\#}quotas-and-throttl>.

132 MICROSOFT. *Azure IoT Hub scaling | Microsoft Docs*. 2020. Disponível em: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-scaling>.

133 MICROSOFT. *Pricing – Azure IoT Central | Microsoft Azure*. 2020. Disponível em: <https://azure.microsoft.com/en-us/pricing/details/iot-central/>.

APÊNDICES

I. CÓDIGOS E COMANDOS UTILIZADOS

I.1 CÓDIGOS REFERENTES DA SOLUÇÃO DE IOT IAAS

I.1.1 Criação da Máquina Virtual

Comando Azure CLI para criação de uma Máquina Virtual:

```
az vm create --name minhaVM --resource-group {ResourceName} --admin-username
    {username} --admin-password {pass} --size Standard_B1s --image UbuntuLTS
```

I.1.2 Abrir a porta do MQTT

```
az vm open-port --port 1883 --resource-group {ResourceName} --name {name}
```

I.1.3 Instalar o Influx DB

```
sudo apt install influxdb
```

I.1.3.1 Instalar os clientes do Influx

```
sudo apt install influxdb-client
```

I.1.3.2 Habilitar o serviço de http do Influx

```
sudo nano /etc/influxdb/influxdb.conf
```

Alterar o arquivo para habilitar http e em seguida reiniciar o serviço.

```
sudo service influxdb restart
```

I.1.3.3 Criar um *database* no Influx

```
> influx
> CREATE DATABASE coletamqtt
> Create User 'mqttuser' with password 'mqtt'
> GRANT ALL ON coletamqtt TO mqttuser
```

I.1.4 Usando Mosquitto como o Broker MQTTT

```
sudo apt-get install mosquitto
sudo apt-get install mosquitto-clients
```

I.1.4.1 Configurantdo para rodar no início da maquina

```
sudo systemctl enable mosquitto
sudo systemctl start mosquitto
```

I.1.5 Subscriber

I.1.5.1 Instalar Python 3 e pacotes necessários

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get install python3.8

sudo apt-get install python3-pip

sudo pip3 install paho-mqtt
sudo pip3 install influxdb
```

I.1.5.2 Ponte entre o Mqtt e o influx

```
1 import re
2 import json
3 from typing import NamedTuple
4 from datetime import datetime, timedelta
5 import threading
6 import time
7
8 import paho.mqtt.client as mqtt
```

```
9  from influxdb import InfluxDBClient
10
11 # escritorio/local/iddispositivo
12 # {"tipo": "d2c", "temperatura": 23.5, "umidade": 38.4, "luminosidade": 830}
13 MQTTIP = '52.225.223.230'
14 InfluxIP = '52.225.223.230'
15 topicoMQTT = 'escritorio/+/+'
16 regexMQTT = 'escritorio/([^/]+)/([^/]+)'
17 idclienteMQTT = 'PonteMQTTInflux'
18
19 influxdb_client = InfluxDBClient(InfluxIP, 8086, 'mqttuser', 'mqtt', None)
20
21
22 class Dados(NamedTuple):
23     temperatura: float
24     umidade: float
25     luminosidade: float
26     local: str
27     iddispositivo: str
28
29
30 def on_connect(client, userdata, flags, rc):
31     client.subscribe(topicoMQTT)
32
33
34 def on_message(client, userdata, msg):
35     print(msg.topic + ' ' + str(msg.payload))
36     sensor_data = _parse_mqtt_message(msg.topic, msg.payload.decode('utf-8'))
37     if sensor_data is not None:
38         _send_sensor_data_to_influxdb(sensor_data)
39
40
41 def _parse_mqtt_message(topic, payload):
42     message = json.loads(payload)
43     match = re.match(regexMQTT, topic)
44     if match and message["tipo"] == 'd2c':
45         local = match.group(1)
46         iddispositivo = match.group(2)
47         return Dados(float(message["temperatura"]), float(message["umidade"]),
               float(message["luminosidade"]), local, iddispositivo)
48     else:
49         return None
50
51
52 def _send_sensor_data_to_influxdb(sensor_data):
53     json_body = [
54         {
55             'measurement': 'Sensor Inteligente',
56             'tags': {
57                 'iddispositivo': sensor_data.iddispositivo,
58                 'local': sensor_data.local
59             },
```

```
60              'fields': {
61                  'temperatura': sensor_data.temperatura,
62                  'umidade': sensor_data.umidade,
63                  'luminosidade': sensor_data.luminosidade
64              }
65          }
66      ]
67      influxdb_client.write_points(json_body)
68      print('Medida Salva' + str(json_body))
69
70
71  def _init_influxdb_database():
72      databases = influxdb_client.get_list_database()
73      if len(list(filter(lambda x: x['name'] == 'SensorInteligente', databases))) ==
            0:
74          influxdb_client.create_database('SensorInteligente')
75      influxdb_client.switch_database('SensorInteligente')
76
77
78  def main():
79      _init_influxdb_database()
80
81      mqtt_client = mqtt.Client('MQTT2Influx')
82      # mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
83      mqtt_client.on_connect = on_connect
84      mqtt_client.on_message = on_message
85
86      mqtt_client.connect(MQTTIP, 1883)
87      mqtt_client.loop_forever()
88
89
90  if __name__ == '__main__':
91      print('MQTT to InfluxDB bridge')
92      main()
```

I.1.6 Simulações

```
1  import uuid
2  import json
3  from typing import NamedTuple
4  import paho.mqtt.client as mqtt
5  import threading
6  import random
7  import time
8  from datetime import datetime, timedelta
9
10
11 MQTTIP = ''
12
```

```
13 topicoMQTT = 'escritorio/+/+'
14 regexMQTT = 'escritorio/([^/]+)/([^/]+)'
15 idclienteMQTT = 'PonteMQTTInflux'
16 medidas = []
17
18 LOCK = threading.Lock()
19
20
21 class Dados(NamedTuple):
22     temperatura: float
23     umidade: float
24     luminosidade: float
25
26
27 class Contador(NamedTuple):
28     GUID: str
29     inicio: str
30     fim: str
31     contador: int
32
33
34 def medida2CSV(medida):
35     return "{},{},{},{}\n".format(medida.GUID, medida.inicio, medida.fim, medida.
           contador)
36
37
38 def on_connect(client, userdata, flags, rc):
39     # client.subscribe(topicoMQTT)
40
41
42 def on_message(client, userdata, msg):
43     print(msg.topic + ' ' + str(msg.payload))
44     msg = _parse_mqtt_message(msg.topic, msg.payload.decode('utf-8'))
45     print(msg.topic + ' ' + str(msg))
46
47
48 def _parse_mqtt_message(topic, payload):
49     message = json.loads(payload)
50     # match = re.match(regexMQTT, topic)
51     # if match and message["tipo"] == 'd2c':
52     #   local = match.group(1)
53     #   iddispositivo = match.group(2)
54     #   return Dados(float(message["temperatura"]),
55     #                float(message["umidade"]),
56     #                float(message["luminosidade"]))
57     if message["tipo"] == 'c2d':
58         return payload
59
60
61 def mqttf(GUID, TempoSleep):
62     mqtt_client = mqtt.Client(GUID)
63     andar = [1, 2, 3, 4]
```

```
64
65      topico = 'escritorio/Andar-' + str(random.choice(andar)) + '/' + str(GUID)
66      # mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
67      mqtt_client.on_connect = on_connect
68      mqtt_client.on_message = on_message
69
70      mqtt_client.connect(MQTTIP, 1883)
71      # mqtt_client.loop_forever()
72      contagem = 0
73      inicio = datetime.now()
74      fim = inicio + timedelta(minutes=3)
75      while fim > datetime.now():
76          temperatura = random.uniform(22.2, 37.8)
77          umidade = random.uniform(30.5, 62.3)
78          luminosidade = random.uniform(750, 950)
79          plj = {"tipo": "d2c",
80                 "temperatura": temperatura,
81                 "umidade": umidade,
82                 "luminosidade": luminosidade
83                 }
84          pls = json.dumps(plj)
85          print(pls)
86          mqtt_client.publish(topico, pls, qos=0, retain=False)
87          contagem = contagem + 1
88          time.sleep(TempoSleep)
89      LOCK.acquire()
90      medidas.append(Contador(GUID, inicio, fim, contagem))
91      LOCK.release()
92
93
94  def main(lote):
95      # parser = argparse.ArgumentParser(description='ID do Lote')
96      # python3 filename.py --id 5
97      # parser.add_argument('--id', dest='id', type=int, help='Passe o ID do Lote')
98      # args = parser.parse_args()
99      # print(args.id)
100     threads = []
101     for i in range(0, 100):
102         # GUID = str(args.id) + '-' + str(i)
103         GUID = "{}-{}".format(lote, i)
104         print("Main    : creating device {}".format(GUID))
105         x = threading.Thread(target=mqttf, args=(GUID, random.uniform(15, 75)))
106         x.start()
107         threads.append(x)
108     print("Devices Created!")
109     for thread in threads:
110         thread.join()
111
112
113 if __name__ == '__main__':
114     print('Creating 500 devices...')
115     lote = uuid.uuid4().hex[-4:]
```

```
116     main(lote)
117     now = datetime.now()
118     csv = open("{}.csv".format(now.strftime("media/{}-%Y-%m-%d-%H-%M".format(lote)
            )), "w")
119     csv.write("GUID,Inicio,Fim,Contador\n")
120     total = 0
121     for medida in medidas:
122         line = medida2CSV(medida)
123         print(line)
124         csv.write(line)
125         total = total + medida.contador
126     csv.write("TOTAL,,,{}\n".format(total))
127     csv.close()
128     print(str(total))
```

I.1.7 Processamento

```
1  import re
2  import json
3  from typing import NamedTuple
4  from datetime import datetime, timedelta
5  import threading
6  import time
7  import os
8
9  import paho.mqtt.client as mqtt
10
11 # escritorio/local/iddispositivo
12 # {"tipo": "d2c", "temperatura": 23.5, "umidade": 38.4, "luminosidade": 830}
13 MQTTIP = 'broker.hivemq.com'
14 topicoMQTT = 'escritorio/+/+'
15 regexMQTT = 'escritorio/([^/]+)/([^/]+)'
16 LOCK = threading.Lock()
17
18
19 # lastMessageTime = -1
20 # firstMessageTime = -1
21
22
23 class Dados(NamedTuple):
24     temperatura: float
25     umidade: float
26     luminosidade: float
27     local: str
28     iddispositivo: str
29
30
31 def createGlobalVar():
32     global contadorInst
```

```
33      contadorInst = {}
34      global contadorDisp
35      contadorDisp = {}
36
37
38  def getLastTime():
39      try:
40          return lastMessageTime
41      except:
42          return None
43
44
45  def setLastTime(time):
46      global lastMessageTime
47      lastMessageTime = time
48      if time is None:
49          print("Set last message time to none")
50      else:
51          print("Set last message time: {}".format(lastMessageTime.strftime("%Y-%m-%
                d-%H-%M")))
52
53
54  def getFirstTime():
55      try:
56          return firstMessageTime
57      except:
58          return None
59
60  def setFirstTime(time):
61      global firstMessageTime
62      firstMessageTime = time
63      if time is None:
64          print("Set last message time to none")
65      else:
66          print("Set last message time: {}".format(firstMessageTime.strftime("%Y-%m
                -%d-%H-%M")))
67
68
69  def on_connect(client, userdata, flags, rc):
70      client.subscribe(topicoMQTT)
71
72
73  def on_message(client, userdata, msg):
74      print(msg.topic + ' ' + str(msg.payload))
75      if getFirstTime() is None:
76          setFirstTime(datetime.now())
77      setLastTime(datetime.now())
78      sensor_data = _parse_mqtt_message(msg.topic, msg.payload.decode('utf-8'))
79
80
81  def _parse_mqtt_message(topic, payload):
82      message = json.loads(payload)
```

```
83      match = re.match(regexMQTT, topic)
84      if match and message["tipo"] == 'd2c':
85          local = match.group(1)
86          iddispositivo = match.group(2)
87          GUID = iddispositivo.split("-")[0]
88          LOCK.acquire()
89          if contadorInst.get(GUID) is None:
90              contadorInst[GUID] = 1
91          else:
92              contadorInst[GUID] = contadorInst[GUID] + 1
93          if contadorDisp.get(iddispositivo) is None:
94              contadorDisp[iddispositivo] = 1
95          else:
96              contadorDisp[iddispositivo] = contadorDisp[iddispositivo] + 1
97          LOCK.release()
98          return Dados(float(message["temperatura"]), float(message["umidade"]),
                float(message["luminosidade"]), local, iddispositivo)
99      else:
100         return None
101
102
103 def writeFromDict(file, endTime, values):
104     total = 0
105     file.write("Key,Value,{}\n".format(firstMessageTime.strftime("%Y-%m-%d-%H-%M")
            ))
106     for key, value in values.items():
107         file.write("{},{}\n".format(key, value))
108         total = total + value
109     file.write("TOTAL,{},{}".format(total, endTime.strftime("%Y-%m-%d-%H-%M")))
110
111
112 def writeCSVs():
113     while True:
114         print("Checking for message timeout...")
115         if (getLastTime() is not None) and (getFirstTime() is not None):
116             endTime = getLastTime() + timedelta(seconds=150)
117             now = datetime.now()
118             if endTime < now:
119                 print("Timeout reached! It has been {} seconds since the last
                        message.".format((endTime - getLastTime()).seconds))
120
121                 print("Creating output folder")
122                 try:
123                     os.mkdir("csvFiles/")
124                 except FileExistsError as exc:
125                     print(exc)
126                 try:
127                     os.mkdir("csvFiles/in/")
128                 except FileExistsError as exc:
129                     print(exc)
130
131                 print("Starting to write the files!")
```

```
132                 GUIDfileName = "csvFiles/in/instancias-{}.csv".format(now.strftime
                        ("%Y-%m-%d-%H-%M"))
133                 GUIDcsv = open(GUIDfileName, "w")
134                 LOCK.acquire()
135                 writeFromDict(GUIDcsv, endTime, contadorInst)
136                 GUIDcsv.close()
137 
138                 dispFileName = "csvFiles/in/dispositivos-{}.csv".format(now.
                        strftime("%Y-%m-%d-%H-%M"))
139                 dispcsv = open(dispFileName, "w")
140                 writeFromDict(dispcsv, endTime, contadorDisp)
141                 LOCK.release()
142                 dispcsv.close()
143                 print("Finished writing the files!")
144                 print("Files:\n\t{}\n\t{}".format(GUIDfileName, dispFileName))
145                 setFirstTime(None)
146                 setLastTime(None)
147                 createGlobalVar()
148             else:
149                 print("Timeout was not reached!")
150         time.sleep(15)
151 
152 
153 def main():
154     createGlobalVar()
155     x = threading.Thread(target=writeCSVs, args=())
156     x.start()
157 
158     mqtt_client = mqtt.Client('MQTT2CSV')
159     # mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
160     mqtt_client.on_connect = on_connect
161     mqtt_client.on_message = on_message
162 
163     mqtt_client.connect(MQTTIP, 1883)
164     mqtt_client.loop_forever()
165 
166 
167 if __name__ == '__main__':
168     print('MQTT to CSV bridge')
169     main()
```

I.1.8 Configuração Grafana

Instalando Grafana:

```
wget https://dl.grafana.com/oss/release/grafana_6.5.1_amd.deb
sudo dpkg -i grafana_6.5.1_amd.deb
sudo apt-get update
sudo apt-get install grafana
```

Iniciando Grafana:

```
sudo service grafana-server start
```

I.2 CÓDIGOS REFERENTES A CRIAÇÃO DA SOLUÇÃO DE IOT PAAS

I.2.1 Device Provisioning Service

Criar um Device Provisioning Service utlizando o Azure CLI:

```
az iot dps create --name SimulacoesDps --resource-group ResourceGroupName
```

I.2.2 IoT Hub

Criar um IoT Hub utilizando o Azure CLI:

```
az iot hub create --resource-group {ResourceName} --name Simulacoeshub --sku S1
    --partition-count 4
```

I.2.3 Criar os dispositivos no IoT Hub

I.2.3.1 Identidades de dispositivos nos cenários com POE e GPRS

Criar as identidades usando o Azure CLI:

```
az iot hub device-identity create -n Simulacoeshub -d dispositivo-gprs
az iot hub device-identity create -n Simulacoeshub -d dispositivo-poe
...
```

I.2.3.2 Identidade do dispositivo para caso com LoRaWAN

Na Azure Function é verificada a existência de um dispositivo com a identidade desejada, caso não exista , é criado um novo:

```
1         public static async Task<Device> GetDevice(string deviceId)
2         {
3             RegistryMgr = RegistryManager.CreateFromConnectionString(
                  ConnectionString);
```

```
4               List<Device> devices = (await RegistryMgr.GetDevicesAsync(999)).ToList
                    ();
5               Device device = null;
6               devices.ForEach(d => {if(d.Id == deviceId) device = d;});
7               if(device == null)
8               {
9                   device = await RegistryMgr.AddDeviceAsync(new Device(deviceId));
10              }
11              return device;
12          }
```

I.2.4 Azure Functions para envio de mensagens LoraWan para IoT Hub

I.2.4.1 Criar uma conta de Armazenamento

```
az storage account create --name <STORAGE_NAME> --location eastus2
    --resource-group <RGName> --sku Standard_LRS
```

I.2.4.2 Criar um aplicativo de funções utilizando o Azure CLI

```
az functionapp create --resource-group <RGName> --consumption-plan-location
    eastus --runtime dotnet --functions-version 2 --name <APP_NAME>
    --storage-account <STORAGE_NAME>
```

I.2.4.3 Publicar a Azure Function

Publicar a função com usando arquivo compactado:

```
az functionapp deployment source config-zip `
 -g <RGName> -n <APP_NAME> --src <PublishZip>
```

I.2.4.4 Executar a Azure Function

Enviar um requisição HTTP para a URL completa que é mostrada na saída do comando de publicação.

I.2.5 Time Series Insights

Criar um Azure Time Series Insights utlizando o Azure CLI:

```
az timeseriesinsights environment standard create -g {rg} -n {env} --location
    eastus --sku-name L1 --sku-capacity 1 --data-retention-time 31 --partition-key
    DeviceId1 --storage-limit-exceeded-behavior PauseIngress
```

I.2.5.1 Configurar integração com IoT Hub

Verificar a *shared_access_key* do IoT Hub criado:

```
az iot hub policy list -g {rg} --hub-name <IoTHubName> --query
    "[?keyName=='iothubowner'].primaryKey" --output tsv
```

Criar a integração:

```
az timeseriesinsights event-source iothub create -g {rg} --environment-name {env}
    -n eastus --consumer-group-name 'Default' --key-name iothubowner
    --shared-access-key <shared_access_key> --event-source-resource-id
    <es_resource_id> --timestamp-property-name DeviceId
```

I.2.5.2 Acesso aos dados

```
az timeseriesinsights access-policy create -g {rg} --environment-name {env} -n
    ap1 --principal-object-id 001 --description "some description" --roles
    Contributor Reader
```

I.2.6 Códigos de envio de mensagens para IoT Hub

I.2.6.1 Dispositivo com LoraWan

Código principal para integração do dispositivo com LoRaWan e IoTHub. O código completo da implementação pode ser encontrado no repositório citado na seção I.5.

```
1  using System.IO;
2  using System.Threading.Tasks;
3  using Newtonsoft.Json;
4
5  using Microsoft.AspNetCore.Mvc;
6  using Microsoft.AspNetCore.Http;
7  using Microsoft.Extensions.Logging;
8
9  using Microsoft.Azure.WebJobs;
10 using Microsoft.Azure.WebJobs.Extensions.Http;
```

```
11
12 namespace CrazyTechLabs.LoraIotHubBridge
13 {
14     public static class tctec_lorawan
15     {
16         [FunctionName("tctec-lorawan")]
17         public static async Task<IActionResult> Run(
18             [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)
                ] HttpRequest req,
19             ILogger log)
20         {
21             string requestBody = await new StreamReader(req.Body).ReadToEndAsync()
                ;
22             dynamic httpPayload = JsonConvert.DeserializeObject(requestBody);
23             string responseMessage = "Received message!";
24             await BridgeService.Bridge(httpPayload, log);
25             return new OkObjectResult(responseMessage);
26         }
27     }
28 }
```

I.2.6.2 Dispositivo ESP32 com GPRS

Código principal para integração de GPRS e IoTHub usando ESP32. O código completo da implementação pode ser encontrado no repositório citado na seção I.5.

```
1  void azure_task(void *pvParameter)
2  {
3
4      ESP_LOGI(TAG, "Connected to AP success!");
5
6      iothub_client_sample_mqtt_run();
7
8      vTaskDelete(NULL);
9  }
10
11 void app_main()
12 {
13     const TickType_t wait = 60000 / portTICK_PERIOD_MS;
14     createSemaphores();
15      initSim();
16
17     while(ppposInit() != 1);
18     writeLog("PPP Task Initialized.");
19
20     if ( xTaskCreate(&azure_task, "azure_task", 1024 * 5, NULL, 10, NULL) !=
           pdPASS ) {
21         printf("create azure task failed\r\n");
22     }
```

```
23 }
```

I.2.6.3 Dispositivo ESP32 com Ethernet

Código principal para integração usando Ethernet e o IoTHub no ESP32. O código completo da implementação pode ser encontrado no repositório citado na seção I.5.

```
1  void azure_task(void *pvParameter)
2  {
3      // xEventGroupWaitBits(wifi_event_group, CONNECTED_BIT,
4      //                     false, true, portMAX_DELAY);
5      const TickType_t wait = 10 / portTICK_PERIOD_MS;
6      while( xSemaphoreTake( xSemaphore, wait ) != pdTRUE );
7      // We were able to obtain the semaphore and can now access the
8      // shared resource.
9      ESP_LOGI(TAG, "Connected to AP success!");
10
11     iothub_client_sample_mqtt_run();
12
13     vTaskDelete(NULL);
14 }
15
16 #define PIN_PHY_POWER   12
17 void app_main()
18 {
19     xSemaphore = xSemaphoreCreateBinary();
20     if( xSemaphore != NULL )
21     {
22         ESP_LOGI(TAG, "Created semaphore!");
23         xSemaphoreGive( xSemaphore );
24     }
25     const TickType_t wait = 10 / portTICK_PERIOD_MS;
26     while( xSemaphoreTake( xSemaphore, wait ) != pdTRUE );
27     ESP_LOGI(TAG, "Took semaphore!");
28     // We were able to obtain the semaphore and can now access the
29     // shared resource.
30
31
32     tcpip_adapter_init();
33
34     ESP_ERROR_CHECK(esp_event_loop_create_default());
35     ESP_ERROR_CHECK(tcpip_adapter_set_default_eth_handlers());
36     ESP_ERROR_CHECK(esp_event_handler_register(ETH_EVENT, ESP_EVENT_ANY_ID, &
           eth_event_handler, NULL));
37     ESP_ERROR_CHECK(esp_event_handler_register(IP_EVENT, IP_EVENT_ETH_GOT_IP, &
           got_ip_event_handler, NULL));
38
39     eth_mac_config_t mac_config = ETH_MAC_DEFAULT_CONFIG();
40     eth_phy_config_t phy_config = ETH_PHY_DEFAULT_CONFIG();
```

```
41      phy_config.phy_addr = CONFIG_EXAMPLE_ETH_PHY_ADDR;
42      phy_config.reset_gpio_num = CONFIG_EXAMPLE_ETH_PHY_RST_GPIO;
43      gpio_pad_select_gpio(PIN_PHY_POWER);
44      gpio_set_direction(PIN_PHY_POWER,GPIO_MODE_OUTPUT);
45      gpio_set_level(PIN_PHY_POWER, 1);
46      vTaskDelay(pdMS_TO_TICKS(10));
47  #if CONFIG_EXAMPLE_USE_INTERNAL_ETHERNET
48      mac_config.smi_mdc_gpio_num = CONFIG_EXAMPLE_ETH_MDC_GPIO;
49      mac_config.smi_mdio_gpio_num = CONFIG_EXAMPLE_ETH_MDIO_GPIO;
50      esp_eth_mac_t *mac = esp_eth_mac_new_esp32(&mac_config);
51  #if CONFIG_EXAMPLE_ETH_PHY_IP101
52      esp_eth_phy_t *phy = esp_eth_phy_new_ip101(&phy_config);
53  #elif CONFIG_EXAMPLE_ETH_PHY_RTL8201
54      esp_eth_phy_t *phy = esp_eth_phy_new_rtl8201(&phy_config);
55  #elif CONFIG_EXAMPLE_ETH_PHY_LAN8720
56      esp_eth_phy_t *phy = esp_eth_phy_new_lan8720(&phy_config);
57  #elif CONFIG_EXAMPLE_ETH_PHY_DP83848
58      esp_eth_phy_t *phy = esp_eth_phy_new_dp83848(&phy_config);
59  #endif
60  #elif CONFIG_EXAMPLE_USE_DM9051
61      gpio_install_isr_service(0);
62      spi_device_handle_t spi_handle = NULL;
63      spi_bus_config_t buscfg = {
64          .miso_io_num = CONFIG_EXAMPLE_DM9051_MISO_GPIO,
65          .mosi_io_num = CONFIG_EXAMPLE_DM9051_MOSI_GPIO,
66          .sclk_io_num = CONFIG_EXAMPLE_DM9051_SCLK_GPIO,
67          .quadwp_io_num = -1,
68          .quadhd_io_num = -1,
69      };
70      ESP_ERROR_CHECK(spi_bus_initialize(CONFIG_EXAMPLE_DM9051_SPI_HOST, &buscfg, 1)
            );
71      spi_device_interface_config_t devcfg = {
72          .command_bits = 1,
73          .address_bits = 7,
74          .mode = 0,
75          .clock_speed_hz = CONFIG_EXAMPLE_DM9051_SPI_CLOCK_MHZ * 1000 * 1000,
76          .spics_io_num = CONFIG_EXAMPLE_DM9051_CS_GPIO,
77          .queue_size = 20
78      };
79      ESP_ERROR_CHECK(spi_bus_add_device(CONFIG_EXAMPLE_DM9051_SPI_HOST, &devcfg, &
            spi_handle));
80      /* dm9051 ethernet driver is based on spi driver */
81      eth_dm9051_config_t dm9051_config = ETH_DM9051_DEFAULT_CONFIG(spi_handle);
82      dm9051_config.int_gpio_num = CONFIG_EXAMPLE_DM9051_INT_GPIO;
83      esp_eth_mac_t *mac = esp_eth_mac_new_dm9051(&dm9051_config, &mac_config);
84      esp_eth_phy_t *phy = esp_eth_phy_new_dm9051(&phy_config);
85  #endif
86      esp_eth_config_t config = ETH_DEFAULT_CONFIG(mac, phy);
87      esp_eth_handle_t eth_handle = NULL;
88      ESP_ERROR_CHECK(esp_eth_driver_install(&config, &eth_handle));
89      ESP_ERROR_CHECK(esp_eth_start(eth_handle));
90
```

```
91      if ( xTaskCreate(&azure_task, "azure_task", 1024 * 6, NULL, 5, NULL) != pdPASS
            ) {
92          ESP_LOGI(TAG, "create azure task failed");
93      }
94  }
```

I.2.6.4 Azure Sphere com WiFi

Para Configuração do Azure Sphere é necessário registrar o dispositivo na nuvem e depois criar uma aplicação para ele, recomendo a utilização da documentação da Microsoft para os passos atualizados. O envio de versões de *firmwares* para o dispositivo também são feitas usando esta nuvem e nos testes usamos uma versão de teste que se inicia sem nenhum comando na linha de comando.

O Código principal para integração Azure Sphere com Wifi e o IoT Hub pode ser visto abaixo, todo o projeto e arquivos que estão no repositório do Github citado na seção I.5.

```
1   int main(int argc, char *argv[])
2   {
3       Log_Debug("Azure IoT Application starting.\n");
4
5       bool isNetworkingReady = false;
6       if ((Networking_IsNetworkingReady(&isNetworkingReady) == -1) || !
            isNetworkingReady) {
7           Log_Debug("WARNING: Network is not ready. Device cannot connect until
                network is ready.\n");
8       }
9
10      ParseCommandLineArguments(argc, argv);
11
12      exitCode = ValidateUserConfiguration();
13      if (exitCode != ExitCode_Success) {
14          return exitCode;
15      }
16
17      exitCode = InitPeripheralsAndHandlers();
18
19      // Main loop
20      while (exitCode == ExitCode_Success) {
21          EventLoop_Run_Result result = EventLoop_Run(eventLoop, -1, true);
22          // Continue if interrupted by signal, e.g. due to breakpoint being set.
23          if (result == EventLoop_Run_Failed && errno != EINTR) {
24              exitCode = ExitCode_Main_EventLoopFail;
25          }
26      }
27
28      ClosePeripheralsAndHandlers();
29
30      Log_Debug("Application exiting.\n");
```

```
31
32      return exitCode;
33  }
```

Envio de mensagens ao IoT Hub:

```
1   static void AzureTimerEventHandler(EventLoopTimer *timer)
2   {
3       if (ConsumeEventLoopTimerEvent(timer) != 0) {
4           exitCode = ExitCode_AzureTimer_Consume;
5           return;
6       }
7
8       // Check whether the device is connected to the internet.
9       Networking_InterfaceConnectionStatus status;
10      if (Networking_GetInterfaceConnectionStatus(NetworkInterface, &status) == 0) {
11          if ((status & Networking_InterfaceConnectionStatus_ConnectedToInternet) &&
12              (iotHubClientAuthenticationState ==
                    IoTHubClientAuthenticationState_NotAuthenticated)) {
13              SetUpAzureIoTHubClient();
14          }
15      } else {
16          if (errno != EAGAIN) {
17              Log_Debug("ERROR: Networking_GetInterfaceConnectionStatus: %d (%s)\n",
                        errno,
18                          strerror(errno));
19              exitCode = ExitCode_InterfaceConnectionStatus_Failed;
20              return;
21          }
22      }
23
24      if (iotHubClientAuthenticationState ==
            IoTHubClientAuthenticationState_Authenticated) {
25          telemetryCount++;
26          if (telemetryCount == AzureIoTPollPeriodsPerTelemetry) {
27              telemetryCount = 0;
28              SendSimulatedTelemetry();
29          }
30      }
31
32      if (iothubClientHandle != NULL) {
33          IoTHubDeviceClient_LL_DoWork(iothubClientHandle);
34      }
35  }
```

I.3 CÓDIGOS REFERENTES A CRIAÇÃO DA SOLUÇÃO DE IOT PAAS COM EDGE COMPUTING

I.3.1 Criar um ambiente para uso do dispositivo na ponta

I.3.1.1 Criando uma identidade para o dispositivo Edge

```
az iot hub device-identity create --device-id {IoTEdgeDeviceId} --edge-enabled
    --hub-name Simulacoeshub
```

I.3.1.2 Instalar o IoT Edge *runtime* no dispositivo físico

Instalar os pré-requisitos:

```
curl https://packages.microsoft.com/config/debian/stretch/multiarch/prod.list >
    ./microsoft-prod.list
sudo cp ./microsoft-prod.list /etc/apt/sources.list.d/
curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor >
    microsoft.gpg
sudo cp ./microsoft.gpg /etc/apt/trusted.gpg.d/
```

Instalar o container *engine*:

```
sudo apt-get install moby-engine
```

Instalar o daemon de segurança do IoT Edge:

```
apt list -a iotedge
sudo apt-get install iotedge
```

Configurar a conexão com IoT Hub no arquivo de configuração:

```
sudo nano /etc/iotedge/config.yaml
```

I.3.2 Modulo de processamento e envio de mensagens

Código Principal do módulo que processa e envia mensagens do dispositivo na ponta para o IoTHub. O código completo da implementação pode ser encontrado no repositório citado na seção GitHub.

```
1 public static class ProcessamentoLocal
```

```
{
    [FunctionName("ProcessamentoLocal")]
    public static async Task FilterMessageAndSendMessage(
                [EdgeHubTrigger("input1")] Message messageReceived,
                [EdgeHub(OutputName = "output1")] IAsyncCollector<Message> output,
                ILogger logger)
    {
        byte[] messageBytes = messageReceived.GetBytes();
        var messageString = System.Text.Encoding.UTF8.GetString(messageBytes);
        List<msgmodbus> MsgsInput = JsonConvert.DeserializeObject<List<msgmodbus
            >>(messageString);
        string[] l = MsgsInput.Select(b => b.DisplayName).Distinct().ToArray();
        foreach (var i in l)
        {
            var lista = MsgsInput.Where(f => f.DisplayName == i).OrderBy(e => e.
                SourceTimestamp).ThenBy(g => g.Address).ToArray();
            for (int a = 0; a < lista.Count(); a = a + 2)
            {
                byte[] primeiraparte = BitConverter.GetBytes(UInt16.Parse(lista[a
                    ].Value));
                byte[] segundaparte = BitConverter.GetBytes(UInt16.Parse(lista[a +
                     1].Value));
                var palavra32 = new byte[4] { primeiraparte[0], primeiraparte[1],
                    segundaparte[0], segundaparte[1] };
                var valorpreliminar = (double)BitConverter.ToInt32(palavra32, 0);
                dynamic payload = new ExpandoObject();
                payload.Medida = lista[a].DisplayName;
                payload.Hardware = lista[a].HwId;
                payload.DataHora = lista[a].SourceTimestamp;
                switch (lista[a].DisplayName)
                {
                    case "Voltagem Fase 1":
                        payload.ValorMedida = ((double)valorpreliminar / 10.0);
                        break;
                    case "Amperagem Fase 1 - (Atual)":
                        payload.ValorMedida = ((double)valorpreliminar / 1000.0);
                        break;
                    default:
                        payload.ValorMedida = valorpreliminar;
                        break;
                }
                var mensagemtexto = JsonConvert.SerializeObject(payload);
                Message msg = new Message(System.Text.Encoding.UTF8.GetBytes(
                    mensagemtexto));
                await output.AddAsync(msg);
            }
        }
    }
}
```

I.4 CÓDIGOS REFERENTES A CRIAÇÃO DA SOLUÇÃO DE IOT SAAS

I.4.1 Criando IoT Central

```
az iot central app create -n {appname} -g {resourceName} -s
    application-simulation -l eastus2
```

O arquivo utilizado na gravação do *firmware* dos dispositivos conectados ao IoT Central podem ser encontrados no repositório citado na seção I.5.

I.5 REPOSITÓRIO COM CÓDIGOS COMPLETOS

Os códigos completos de todas as simulações citadas nesse documento estão em um repositório público no GitHub, encontrado na url: https://github.com/jorgemaia/Arquitetura-IoT-Mestrado

II. TABELAS

II.1 TABELAS GERADAS NAS SIMULAÇÕES DE IAAS

Todas as tabelas citadas nesse documento que foram geradas nas simulações e utilizadas para fornecer os dados aos os gráficos estão disponíveis em um repositório público no GitHub, encontrado na url: https://github.com/jorgemaia/Arquitetura-IoT-Mestrado

	CENÁRIO 1								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	448			469			454		
2	462			456			451		
3	452			449			462		
4	445			447			458		
5	459			454			455		
6	447	4541	4541	469	4554	4554	450	4549	4549
7	465			449			454		
8	459			444			469		
9	453			456			447		
10	451			461			459		
TOTAL	4541			4554			4559		
	Percentual	1	1	Percentual	1	1	Percentual	0.99781	0.997806537

	CENÁRIO 2								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	463			460			458		
2	468			463			460		
3	463			460			461		
4	473			449			454		
5	459			453			455		
6	460	4608	4608	448	4558	4558	483	4574	4574
7	445			456			453		
8	465			443			447		
9	455			465			459		
10	457			461			455		
TOTAL	4608			4558			4585		
	Percentual	1	1	Percentual	1	1	Percentual	0.9976	0.997600872

	CENÁRIO 3								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	457			454			451		
2	464			457			448		
3	461			479			461		
4	448			450			477		
5	478			448			445		
6	461	4579	4579	467	4578	4578	442	4528	4528
7	454			443			455		
8	465			469			451		
9	444			448			452		
10	447			463			456		
TOTAL	4579			4578			4538		
	Percentual	1	1	Percentual	1	1	Percentual	0.9978	0.997796386

Figura II.1: Resultados dos Cenários com 1000 dispositivos

	CENÁRIO 1								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	1608			1727			1672		
2	1704			1695			1718		
3	1600			1685			1636		
4	1713			1646			1643		
5	1672			1714			1628		
6	1673	15174	15174	1666	15228	15228	1701	15330	15330
7	1627			1637			1696		
8	1753			1596			1617		
9	1636			1698			1681		
10	1587			1758			1627		
TOTAL	**16573**			**16822**			**16619**		
	Percentual	0.9156	0.916	**Percentual**	0.90524	0.905243134	**Percentual**	0.92244	0.922438173

	CENÁRIO 2								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	1626			1532			1726		
2	1628			1493			1533		
3	1550			1631			1666		
4	1522			1530			1739		
5	1630			1589			1672		
6	1522	14905	14905	1620	15019	15019	1705	15379	15379
7	1526			1654			1569		
8	1581			1598			1625		
9	1638			1497			1558		
10	1528			1548			1666		
TOTAL	**15751**			**15692**			**16459**		
	Percentual	0.9463	0.946	**Percentual**	0.95711	0.957111904	**Percentual**	0.93438	0.934382405

	CENÁRIO 3								
	qos0			qos1			qos2		
Instancia	Enviados	csv	Banco	Enviados	csv	Banco	Enviados	csv	Banco
1	398			145			1442		
2	292			1120			61		
3	998			141			1257		
4	287			1256			243		
5	214			112			152		
6	196	4591	4579	253	7114	7114	263	5973	5973
7	675			145			831		
8	1262			117			193		
9	347			87			903		
10	329			1510			266		
TOTAL	**4998**			**4886**			**5611**		
	Percentual	0.9186	0.916	**Percentual**	1.456	1.455996725	**Percentual**	1.06452	1.064516129

Figura II.2: Resultados dos Cenários com 5000 dispositivos

		Recebidos	Banco	Meta (100%)
Cenário 1	qos0	1	1	1
	qos1	1	1	1
	qos2	0.99780654	0.997807	1
Cenário 2	qos0	1	1	1
	qos1	1	1	1
	qos2	0.99780654	0.997807	1
Cenário 3	qos0	1	1	1
	qos1	1	1	1
	qos2	0.99779639	0.997796	1

Figura II.3: Tabela resumida para gerar gráficos - 1000 Dispositivos

		Recebidos	Banco	Meta (100%)
Cenário 1	qos0	0.91558559	0.915586	1
	qos1	0.90524313	0.905243	1
	qos2	0.92243817	0.922438	1
Cenário 2	qos0	0.94628912	0.946289	1
	qos1	0.9571119	0.957112	1
	qos2	0.92243817	0.922438	1
Cenário 3	qos0	0.91856743	0.916166	1
	qos1	1.45599673	1.455997	1
	qos2	1.06451613	1.064516	1

Figura II.4: Tabela resumida para gerar gráficos - 5000 Dispositivos

1000	Cenário 1			Cenário 2			Cenário 3		
	qos0	qos1	qos2	qos0	qos1	qos2	qos0	qos1	qos2
Número de Dispositivos que Enviaram	1000	1000	1000	1000	1000	1000	1000	1000	1000
Percentual	1	1	1	1	1	1	1	1	1

Figura II.5: Percentual de dispositivos que enviaram - 1000 Dispositivos

5000	Cenário 1			Cenário 2			Cenário 3		
	qos0	qos1	qos2	qos0	qos1	qos2	qos0	qos1	qos2
Número de Dispositivos que Enviaram	3331	3330	3362	3297	3308	3358	1143	1542	1321
Percentual	0.6662	0.666	0.6724	0.6594	0.6616	0.6716	0.2286	0.3084	0.2642

Figura II.6: Percentual de dispositivos que enviaram - 5000 Dispositivos

ANEXOS

I. AUTORIZAÇÃO DE USO DE NOME, IMAGEM E CITAÇÃO DE INFORMAÇÕES - CRAZYTECHLABS

Ao PPMEC, Faculdade de Tecnologia – Depto de Engenharia Mecânica - UnB

Ref: Autorização de uso de nome, imagem e citação de informações

Por meio desta, autorizamos o Sr. Jorge Andrade Seixas Maia a citar e apresentar em sua dissertação de Mestrado casos de uso de Internet das Coisas que tenham sido desenvolvidos pela CrazyTechLabs e nos quais ele tenha participado ativamente como consultor ou passivamente como observador externo. Aproveitamos para também declarar que o Sr. Jorge esteve ligado a esta empresa desde sua fundação.

Fica vedada toda a exposição de nomes de clientes, infraestrutura de projetos, demonstrações que exponham quantidades, numerários, códigos, diagramas, fotos e quaisquer outros materiais físicos ou virtuais relativos aos projetos acompanhados, sendo qualquer infringência uma declaração de não acordo com o acordo de confidencialidade assinado entre as partes. Dentre os casos supra citados o único com autorização de exposição do cliente é o projeto de pesagem de caminhões desenvolvido para o grupo Queiroz Galvão na cidade de Recife.

Brasília, 20 de outubro de 2020.

EMPRESA: CRAZYTECHLABS - TECNOLOGIA DA INFORMACAO EIRELI
CNPJ: 21.014.760/0001-28
INSCRIÇÃO ESTADUAL: 07.695.324/001-02
ENDEREÇO: SRTVN Lote P Edifício Brasília Rádio Center Sala 1126
CEP: 70719-900
CIDADE: Brasília
ESTADO: DF

Juliana Dias de Freitas
CPF: 834.712.941-04

CRAZYTECHLABS TECNOLOGIA DA INFORMACAO EIRELI:21014760000128

Assinado de forma digital por CRAZYTECHLABS TECNOLOGIA DA INFORMACAO EIRELI:21014760000128
Dados: 2020.10.20 11:19:23 -03'00'

CRAZYTECHLABS – TECNOLOGIA DA INFORMAÇÃO EIRELI
SRTVN Lote P Edifício Brasília Rádio Center Sala 1126 CEP: 70719-900 - Brasília/DF