

UNIVERSIDADE DE PASSO FUNDO

Programa de Pós-Graduação em
Computação Aplicada

Dissertação de Mestrado

PLATAFORMA WATSON PARA INTERNET DAS COISAS

LUIS MAURICIO BRAGA



UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

PLATAFORMA WATSON PARA INTERNET DAS COISAS

Luis Mauricio Braga

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Marco Antônio Sandini Trentin

Passo Fundo
2020

CIP – Catalogação na Publicação

B813p Braga, Luis Mauricio
Plataforma Watson para internet das coisas [recurso eletrônico] / Luis Mauricio Braga. – 2020.
1.9 MB ; PDF.

Orientador: Prof. Dr. Marco Antônio Sandini Trentin.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2020.

1. Internet das coisas. 2. Raspberry Pi (Computador).
3. Inteligência artificial. 4. Plataforma Watson IoT.
I. Trentin, Marco Antônio Sandini, orientador. II. Título.

CDU: 004.738.5

Catalogação: Bibliotecária Jucelei Rodrigues Domingues - CRB 10/1569

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO

LUÍS MAURÍCIO BRAGA

Aos 10 dias do mês de julho do ano de dois mil e vinte, às quatorze horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo (UPF), a sessão pública de defesa do Trabalho de Conclusão de Curso “Plataforma Watson para Internet das Coisas”, de autoria de Luís Maurício Braga, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada - PPGCA. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Marco Antônio Sandini Trentin, Marcelo Trindade Rebonatto e Luiz Eduardo Schardong Spalding. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato **APROVADO**. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



Prof. Dr. Marco Antônio Sandini Trentin - UPF
Presidente da Banca Examinadora
(Orientador)



Prof. Dr. Marcelo Trindade Rebonatto - UPF
(Avaliador Interno)



Prof. Dr. Luiz Eduardo Schardong Spalding - Hospital São Vicente de Paulo
(Avaliador Externo)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA

AGRADECIMENTOS

Agradeço a Deus, pela sua iluminação, que me proporcionou força, garra e determinação durante essa longa caminhada na conclusão deste trabalho.

Agradeço primeiramente à minha família, minha mãe, meu pai e meu irmão, que me deram suporte durante a realização do mestrado que sempre me deram muito amor e incentivo em tudo.

A minha esposa Joelma pela companhia e apoio, que de forma presente me acompanhou em todos os momentos e compreendeu a minha ausência, em momentos de muito estudo, e à minha filha Nathália.

Ao orientador, Prof. Dr. Marco Antônio Sandini Trentin, por sua orientação, incentivo e paciência, que tornaram possível a conclusão desta dissertação.

Um muito obrigado também aos demais professores do PPGCA da Universidade de Passo Fundo, pela compreensão e ensinamentos.

Por fim, dedico este trabalho à toda minha família, amigos e todas as pessoas que juntas, me ajudaram e me fizeram crescer, me dando segurança, apoio e, acima de tudo, incentivo para que eu chegasse até esta etapa de minha vida.

Obstáculos são aquelas coisas assustadoras que você vê quando desvia seus olhos de sua meta.

Henry Ford

PLATAFORMA WATSON PARA INTERNET DAS COISAS

RESUMO

Este projeto foi desenvolvido para uso em uma sala de ordenha, na cidade Sertão/RS, município o qual abriga grande produção de leite da região do Alto Uruguai, buscando trazer contribuições a esta atividade econômica. A plataforma WATSON para Internet das Coisas (IoT) é um serviço gerenciado, hospedado na nuvem, projetado para simplificar a obtenção de valores dos dispositivos da IoT. A evolução do conceito de IoT ocorreu devido à convergência de várias tecnologias, como análises em tempo real, aprendizado de máquina, sensores de mercadorias e sistemas embarcados. Neste sentido, o objetivo deste trabalho foi o de implementar o protótipo de uma plataforma capaz de estabelecer a interoperabilidade entre componentes de hardware e software. Foi proposto um modelo de sistema baseado em uma arquitetura de IoT, ligado a tecnologia IBM WATSON. A partir da implementação do protótipo da plataforma de IoT, ligado a tecnologia IBM WATSON, que interpreta o texto enviado pelo usuário, foi possível observar que a mesma é capaz de otimizar e facilitar gestão do ambiente da produção de leite. Esta plataforma permite o uso de interface que assegura aos usuários uma melhor visualização das operações e condições como temperatura, umidade, CO e luminosidade, captados através de sensores.

Palavras-chave: Raspberry PI. WATSON. Internet das Coisas.

WATSON PLATFORM FOR INTERNET OF THINGS

ABSTRACT

This project was developed for use in a milking parlor, in the city of Sertão / RS, a municipality that houses a large milk production in the Alto Uruguay region, seeking to bring contributions to this economic activity. The WATSON platform for Internet of Things (IoT) is a managed service, hosted in the cloud, designed to simplify obtaining values from IoT devices. The evolution of the IoT concept occurred due to the convergence of several technologies, such as real-time analysis, machine learning, merchandise sensors and embedded systems. In this sense, the objective of this work was to implement the prototype of a platform capable of establishing interoperability between hardware and software components. A system model based on an IoT architecture was proposed, linked to IBM WATSON technology. From the implementation of the prototype of the IoT platform, linked to the IBM WATSON technology, which interprets the text sent by the user, it was possible to observe that it is able to optimize and facilitate the management of the milk production environment. This platform allows the use of an interface that ensures users a better view of operations and conditions such as temperature, humidity, CO and light, captured through sensors.

Keywords: Raspberry PI. WATSON. Internet of things.

LISTA DE FIGURAS

Figura 1. Evolução da IoT	16
Figura 2. Possível cenário de IoT.....	19
Figura 3. Raspberry PI zero	31
Figura 4. Chip mpc3008.....	32
Figura 5. Sensor LDR.....	32
Figura 6. Sensor MQ7.....	33
Figura 7. Sensor DTH 11.....	33
Figura 8. Protótipo inicial.....	34
Figura 9. Programação Raspberry PI.....	35
Figura 10. Fluxo raiz do da skill associada ao serviço do Assistant do WATSON. ...	36
Figura 11. Painel de diálogo da interação do Serviço.....	36
Figura 12. Painel de diálogo do Serviço.....	37
Figura 13. Text to Speech, responsável pela conversão de texto para áudio... ..	38
Figura 14. Speech to text, responsável pela conversão de áudio para texto.. ..	38
Figura 15. Painel integração IBM Cloud.....	39
Figura 16. Utilizados Adaptive cards.....	40
Figura 17. Painel de Databases.....	41
Figura 18. Exemplo de registro de sensor.....	42
Figura 19. Exemplo de registro de leitura.....	42
Figura 20. Estrutura do Projeto com a utilização do Node.js.....	42
Figura 21. Tela de Interface Portal Azure.....	43
Figura 22. Hardware utilizado.....	44
Figura 23. Esquema da tecnologias utilizadas.....	45
Figura 24. Exemplo de conversa com WATSON e sensor de umidade.....	46
Figura 25. Exemplo de conversa com WATSON e sensor de luz	46
Figura 26. Exemplo de conversa com WATSON e sensor de CO.....	47
Figura 27. Interface do aplicativo mostra o resultado da temperatura.....	47
Figura 28. Interface do aplicativo mostra o resultado da umidade.....	48
Figura 29. Interface do aplicativo mostra o resultado da CO.....	48
Figura 30. Interface do aplicativo mostra o resultado da luminosidade.....	48
Figura 31. Interface do aplicativo mostra o resultado.....	49

LISTA DE SIGLAS

APD - Adaptive Page Display
API - Application Programming Interface
BLE - Bluetooth Low Energy
CC - Computação Cognitiva
CPS - CyberPhysical System
EIoT - Energy Internet of Things
IA - Inteligência Artificial
IaaS - Infrastructure as a Service
IIoT - Industry Internet of Things
IoT – Internet of Things
LoRa - Long Range (LoRa)
LTE-M - Long Term Evolution for Machines
NB-IoT - Narrowband IoT
PaaS - Platform as a Service
REST - Representational State Transfer
RFID - Radio-Frequency Identification
SaaS - Software as a Service
TO - Tecnologia Operacional
VM – Virtual Machine
NPM - Gerenciador de Pacotes de Dependências
NOSQL – Banco de Dados não Relacional

SUMÁRIO

1. INTRODUÇÃO	14
2. REVISÃO DA LITERATURA.....	18
2.1 IOT PLATAFORM.....	18
2.1.1. IoT Industrial	23
2.1.2. A conectividade	24
2.2. A TECNOLOGIA IBM WATSON.....	26
2.2.1. IBM Cloud.....	28
2.2.2. História	29
2.3. MICROSOFT AZURE.....	31
2.4. REST.....	32
2.5. TELEGRAM.....	33
3. A PLATAFORMA MILKBOT	35
3.1. HARDWARE UTILIZADO NA PLATAFORMA.....	35
3.1.1. MPC3008.....	36
3.1.2. LDR	36
3.1.3. MQ-7.....	37
3.1.4. DTH11	37
3.2. PROGRAMAÇÃO DO HARDWARE.....	38
3.3. APLICAÇÃO NO SERVIÇO WATSON ASSISTANT	39
3.3.1. Aplicação de serviços adicionais utilizados	42
3.4. INTEGRAÇÃO	42
3.5. BANCO DE DADOS	44
3.6. APLICAÇÃO	46
3.7. SERVIDOR DE AUTENTICAÇÃO DO BOT	47
4. ESTUDO DE CASO.....	50
5. CONCLUSÃO.....	57
REFERÊNCIAS.....	60
APÊNDICE A – APLICAÇÃO SERVER UTILIZADO NODE.JS.....	64

1. INTRODUÇÃO

A internet das Coisas representa um paradigma, que permite que as pessoas interajam com um ambiente por meio de dispositivos eletrônicos, sem ter conhecimento técnico dos sistemas. A convergência de diversas tecnologias surgidas nas últimas décadas permitiu a criação de ambientes interativos repletos de recursos e possibilidades [1].

Internet das Coisas, em inglês, *Internet of Things* (IoT), concebido como uma rede de dispositivos conectados à Internet e capazes de interagir entre si. Assim, a IoT é um sistema de dispositivos de computação inter-relacionados, máquinas mecânicas e digitais, objetos, animais ou pessoas que recebem identificadores exclusivos (UIDs) e a capacidade de transferir dados através de uma rede sem a necessidade de humano para humano ou interação humano-computador [2] [3].

A evolução do conceito de IoT ocorreu devido à convergência de várias tecnologias, como análises em tempo real, aprendizado de máquina, sensores de mercadorias e sistemas embarcados. Isso fomentou melhoria nos campos tradicionais de sistemas embarcados, em redes de sensores sem fio, sistemas de controle e automação. No mercado consumidor, a tecnologia IoT passou a estar associada ao conceito de "casa inteligente", cobrindo dispositivos e eletrodomésticos (como luminárias, termostatos, sistemas e câmeras de segurança residencial e outros eletrodomésticos) compatíveis com um ou mais ecossistemas comuns, podendo ser controlado por dispositivos como smartphones e ferramentas inteligentes.

Um dos bens considerados mais valiosos, principalmente no atual cenário, é a informação, uma vez que muitos processos ou atividades geram algum tipo de informação para serem armazenadas [4].

A comunidade científica e empresarial prevê o conjunto dessas novas técnicas como uma nova revolução no campo. A plataforma de IoT é um dos pilares da Agricultura 4.0, a qual possibilita em tempo real estar conectada com a localização e desempenho de máquinas, equipamentos e sensores, possibilitando uma produtividade mais dinâmica e produtiva.

Já no ramo do mercado corporativo também busca-se incorporar aplicações móveis para que essas possam ser integradas aos seus sistemas como são os sistemas compatíveis com dispositivos móveis, almejando maior agilidade. Existe uma grande demanda da qualidade dos serviços prestados onde o desempenho correto de suas funções torna-se cada vez mais necessário [5].

O desenvolvimento de aplicativos para dispositivos móveis também tem se tornado um grande nicho de mercado, o que vem crescendo vertiginosamente. Para uma maior mobilidade e agilidade vislumbrou-se a necessidade de desenvolvimento de um aplicativo, visando adotar o processo de otimização e qualidade na persistência de dados.

A plataforma WATSON é uma tecnologia destinada aos serviços cognitivos da IBM, otimizando os negócios de diversas organizações, já que permite a integração dos fluxos de trabalho. Dessa maneira, aliada à IoT esta plataforma é responsável pela inovação no gerenciamento dos ativos, gerando valor aos negócios em razão da conectividade, controle, visualização e armazenamento de dados otimizados oferecidos.

Sendo assim, este estudo foi desenvolvido visando a implementação de um protótipo para uso em uma sala de ordenha, que permita acessar informações destas, através das novas tecnologias disponíveis e de sensores, em tempo real, através de um smartphone conectado à Internet. Sendo assim, foi na cidade Sertão/RS, município o qual possui Bacia Leiteira que está entre a maior produção de leite da região do Alto Uruguai com cerca de 600 produtores, com uma produção de 35.000 milhões de litros leite e 6.800 animais em lactação, que um recurso digital como esse pode trazer contribuições a esta atividade econômica.

Diante disso, surge o seguinte problema de pesquisa: é possível, com baixo custo, implementar uma solução de IoT para pequenas propriedades de leites?

Nesse sentido, o presente trabalho tem por objetivo implementar o protótipo de uma plataforma, capaz de estabelecer a interoperabilidade entre componentes e estruturas das tecnologias, propondo um modelo de sistema baseado em uma arquitetura de IoT, ligado a tecnologia IBM WATSON. Os objetivos específicos são: apresentar as plataformas IoT que são usadas no meio rural; e descrever os elementos de funcionamento e implementação da plataforma IoT.

Esta dissertação está estruturada da seguinte forma: O Capítulo 2 descreve a revisão da literatura e os trabalhos relacionados, apresentando duas plataformas usadas para IoT. O Capítulo 3 descreve os elementos que constituem a plataforma proposta, a arquitetura, funcionamento, o modelo de banco de dados e detalhes de implementação. No Capítulo 4 é apresentado o estudo de caso, a fim de avaliar o funcionamento da plataforma. Por fim, o Capítulo 5 contém as considerações finais.

2. REVISÃO DA LITERATURA

2.1 IOT PLATAFORM

O conceito de uma rede de dispositivos inteligentes foi discutido em 1982, com uma máquina de venda automática de Coca-Cola modificada na Universidade Carnegie Mellon se tornando o primeiro dispositivo conectado à Internet, capaz de relatar seu inventário e se as bebidas recém-carregadas estavam frias ou não [6].

O artigo de Mark Weiser, de 1991, sobre computação onipresente, "O Computador do Século XXI", bem como locais acadêmicos como UbiComp e PerCom, produziu a visão contemporânea da IoT. Em 1994, Reza Raji descreveu o conceito no IEEE Spectrum como "(mover) pequenos pacotes de dados para um grande conjunto de nós, de modo a integrar e automatizar tudo, de eletrodomésticos a fábricas inteiras" [7].

Entre 1993 e 1997, várias empresas propuseram soluções como a Microsoft at Work ou NEST da Novell. O campo ganhou força quando Bill Joy imaginou a comunicação dispositivo a dispositivo como parte de sua estrutura "Seis Webs", apresentada no Fórum Econômico Mundial em Davos em 1999 [7].

O termo "Internet das Coisas" provavelmente foi cunhado por Kevin Ashton, da Procter & Gamble, mais tarde no Auto-ID Center do MIT, em 1999. Assim, reitera-se que o termo IoT ou Internet das Coisas, foi adotado por Kevin Ashton, tendo em vista nomear uma rede de objetos, coisas, software. A partir disso, o conceito evoluiu, com aplicações diversas, como sensores e/ou mesmo atuadores capazes de interagir, extraindo ou resgatando dados, estando conectados à Internet e até mesmo entre si, podendo ser coletados por um *hardware* (equipamento) em comunicação direta com o *software* (aplicação). Nesse ponto, ele via a identificação por radiofrequência (RFID) como essencial para a IoT, o que permitiria aos computadores gerenciar todas as coisas individuais [8].

Definindo a Internet das Coisas como "simplesmente o momento em que mais 'coisas ou objetos' estavam conectados à Internet do que pessoas", a Cisco Systems estimou que a IoT "nasceu" entre 2008 e 2009, com a proporção de coisas/pessoas crescendo de 0,08 em 2003 para 1,84 em 2010 [9]. Até o final de

2020, estima-se que este crescimento alcance 50 bilhões, conforme demonstrado na Figura 1.

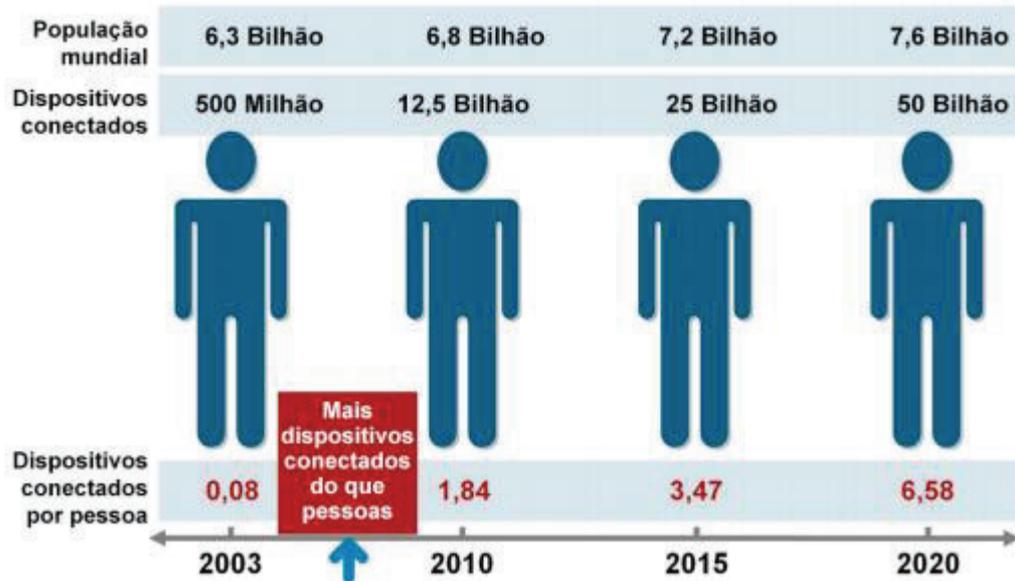


Figura 1. Evolução da IoT [10]

A exponencial expansão da quantidade e diversidade de serviços e produtos oferecidos pela internet, bem como suas formas de acesso, mudando do básico computador de mesa ou mesmo notebook aos diversos tipos de dispositivos móveis, com custos cada vez mais acessíveis, e capacidades de conexão sem fio, criou a maior oportunidade do mercado tecnológico do século 21 [11].

O mercado brasileiro utiliza-se intensamente de IoT em vários segmentos. A partir da obtenção dos dados feita por um dispositivo, busca-se algumas diretrizes, dentre as quais a tradução e representação de dados, transporte e abstração de dados e, por fim, processamento dos dados coletados [12]. Os dispositivos de IoT fazem parte do conceito mais amplo de automação residencial, que pode incluir iluminação, aquecimento e ar condicionado, sistemas de mídia e segurança. Os benefícios à longo prazo pode incluir economia de energia, garantindo automaticamente que as luzes e os eletrônicos sejam desligados [13].

Como exemplo desses benefícios, cita-se que em agosto de 2018, a empresa Toyota Tsusho iniciou uma parceria com a Microsoft para criar ferramentas para auxiliar no acompanhamento de criação de peixes usando o pacote de aplicativos Microsoft Azure para tecnologias IoT relacionadas ao gerenciamento de

água. Este pacote de aplicativos possibilita que a ferramenta seja aplicada em nível empresarial, proporcionando o monitoramento remoto, bem como manutenções de modo otimizado [14].

Desenvolvidos em parte por pesquisadores da Universidade de Kindai, os mecanismos das bombas de água usam inteligência artificial para contar o número de peixes em uma correia transportadora, analisar o número de peixes e deduzir a eficácia do fluxo de água a partir dos dados fornecidos pelos peixes. Os programas de computador específicos usados no processo se enquadram nas plataformas Azure Machine Learning e Azure IoT Hub [14].

Dessa forma, os dispositivos IoT beneficiam diversos tipos de aplicações. Uma casa inteligente ou automatizada pode basear-se em uma plataforma ou hubs que controlam dispositivos e aparelhos inteligentes. Por exemplo, usando o HomeKit da Apple, os fabricantes podem ter seus produtos e acessórios domésticos controlados por um aplicativo em dispositivos iOS como o iPhone e o Apple Watch [11].

Pode ser um aplicativo dedicado ou aplicativos nativos do iOS, como o Siri. Isso pode ser demonstrado no caso do Smart Home Essentials da Lenovo, que é uma linha de dispositivos domésticos inteligentes que são controlados pelo aplicativo Home da Apple ou pelo Siri sem a necessidade de uma ponte Wi-Fi [11]. Uma aplicação importante de uma casa inteligente é fornecer assistência a pessoas com deficiência e idosos. Esses sistemas domésticos usam tecnologia assistida para superar as deficiências específicas de um proprietário [15].

Também existem hubs de casa inteligente dedicados, oferecidos como plataformas independentes para conectar diferentes produtos de casa inteligente e incluem o Amazon Echo, Google Home, HomePod da Apple e SmartThings Hub da Samsung. Além dos sistemas comerciais, existem muitos ecossistemas de código aberto não proprietários, incluindo Home Assistant, OpenHAB e Domoticz.

O controle por voz pode ajudar os usuários com limitações de visão e mobilidade, enquanto os sistemas de alerta podem ser conectados diretamente aos implantes cocleares usados por usuários com deficiência auditiva. Eles também podem ser equipados com recursos de segurança adicionais. Esses recursos podem incluir sensores que monitoram emergências médicas, como quedas ou convulsões. A tecnologia de casa inteligente aplicada dessa maneira pode oferecer aos usuários mais liberdade e maior qualidade de vida.

Destacam-se ainda os seguintes tipos de serviços e transformações decorrentes da IoT:

-*Smart Cities*: Uma diversidade de serviços “inteligentes” que visam melhorar e otimizar as atividades urbanas como o controle e monitoramento de tráfego, gerenciamento de estacionamento, segurança pública e comunicação são apenas alguns dos exemplos já em utilização;

-*Smart Environment*: Monitoramento e gestão da utilização de energia, condições da água e do ar, despejo de lixo entre outros;

-*Smart Health*: Serviços de saúde utilizados através de sensores pela área médica;

-*Smart Production and Logistics*: O contínuo monitoramento da demanda de mercado, capacidade da produção, variação de preços e custos, mantém as indústrias alinhadas a realidade, evitando desperdícios, prejuízos, melhor prevendo necessidades futuras, assim fornecendo dados para a melhor tomada de decisão.

-*Smart Agriculture*: A aplicação da tecnologia faz-se presente no cultivo preciso, que baseado em diversos dados como temperatura, umidade, solo, ventos e pragas maximizam o sucesso do cultivo. O controle da demanda comparado ao estoque também pode ser realizado no sentido de reduzir perdas, sendo que todos esses dados podem ser previstos com antecedência e mesmo disponibilizados em tempo real para dispositivos móveis, atualmente de fácil acesso nas áreas rurais [6].

Existem inúmeras aplicações de IoT na agricultura, como coleta de dados de temperatura, precipitação, umidade, velocidade do vento, infestação de pragas e conteúdo do solo. Esses dados podem ser usados para automatizar técnicas agrícolas, tomar decisões informadas para melhorar a qualidade e a quantidade, minimizar riscos e desperdícios e reduzir o esforço necessário para gerenciar as culturas. Por exemplo, os agricultores agora podem monitorar a temperatura e a umidade do solo de forma remota, e até aplicar dados adquiridos da IoT a programas de fertilização de precisão [9].

O termo "IoT corporativa" (EIoT) refere-se a dispositivos usados em configurações comerciais e corporativas. Em 2019, foi estimado que a EIoT representou 9,1 bilhões de dispositivos. Os dispositivos industriais de IoT adquirem e analisam dados de equipamentos conectados, tecnologia operacional (TO), locais e pessoas. Combinado aos dispositivos de monitoramento de TO o EIoT ajuda a regular e monitorar os sistemas industriais [16].

Assim, percebe-se que a IoT já impacta um grande número de setores, seja na indústria, agricultura, serviços públicos, educação, infraestrutura, saúde e outros. A Figura 2 ilustra uma gama de possibilidades de onde a IoT pode estar presente, denotando assim sua evolução:

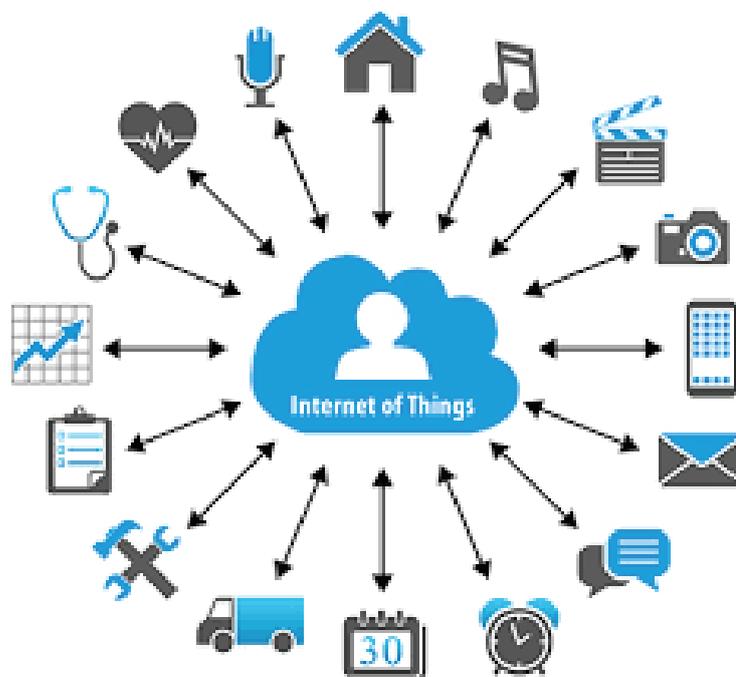


Figura 2. Possível cenário de IoT [18]

Assim, reitera-se que a IoT está se tornando um tópico cada vez mais crescente e importante. É um conceito que não só tem o potencial de impactar como vivemos, mas também como trabalhamos.

A IoT pode realizar a integração de vários dispositivos com recursos de detecção, identificação, processamento, comunicação, atuação e rede. Com base em um espaço ciber-físico inteligente altamente integrado, ele abre as portas para criar novas oportunidades de negócios e mercado para a fabricação [17].

O controle de rede e o gerenciamento de equipamentos de manufatura, o gerenciamento de ativos e situações ou o controle de processos de manufatura colocam a IoT dentro do campo das aplicações industriais e da fabricação inteligente também. Os sistemas inteligentes da IoT permitem a fabricação rápida de novos produtos, a resposta dinâmica às demandas dos produtos e a otimização em tempo real das redes de produção e cadeia de suprimentos, através da conexão em rede de máquinas, sensores e sistemas de controle.

Sistemas de controle digital para automatizar controles de processo, ferramentas do operador e sistemas de informações de serviço para otimizar a segurança da planta estão dentro do alcance da IoT, mas também se estende ao gerenciamento de ativos por meio de manutenção preditiva, avaliação estatística e medições para maximizar a confiabilidade [16].

2.1.1. IoT Industrial

Os sistemas de gerenciamento industrial, também podem ser integrados às redes inteligentes, otimizando a produção, distribuição e o consumo de energia ,permitindo melhorias com segurança, bem como medições, controles automatizados, otimização da fábrica, serviços médicos e segurança e outras funções são fornecidas por um grande número de sensores em rede.

A IoT industrial (IIoT) na fabricação gera tanto valor comercial que levou à Quarta Revolução Industrial, também conhecida como Indústria 4.0. O potencial de crescimento da implementação da IIoT gera em torno US \$ 12 trilhões em PIB global até 2030. A análise industrial de *big data*¹ desempenhará um papel vital na manutenção preditiva de ativos de fabricação, embora essa não seja a única capacidade do big data industrial [18].

Os sistemas ciber-físicos (CPS) são a tecnologia principal do *big data* industrial e serão uma interface entre o mundo humano e o cibernético. Os sistemas ciber-físicos podem ser projetados seguindo a arquitetura 5C (conexão, conversão, cyber, cognição, configuração), e transformará os dados coletados em informações acionáveis e, eventualmente, interferirá nos ativos físicos para otimizar processos [18].

Por exemplo, um sistema de prognóstico desenvolvido será capaz de reconhecer e monitorar a degradação das correias de serra de fita, mesmo que a condição esteja mudando, aconselhando os usuários quando é o melhor momento

¹O acesso ao "Big Data" auxilia na automação, de forma inteligente, definida por volume, variedade, veracidade, valor, validade e verificação dos dados, bem como informações e diagnóstico automatizado. . A aplicabilidade do Big Data está no tratamento desse volume de dados, que vem de variadas fontes e que demandam alta velocidade de processamento, na busca por um valor [18, p. 35].

para substituir a correia. Isso melhorará significativamente a experiência do usuário e a segurança do operador, além de economizar em custos [6].

Igualmente, voltado para a agricultura, Rosa [19] afirma que os dispositivos IoT, embora necessitem de investimentos significativos, possibilitam melhorias notórias para a agricultura de precisão como a assertividade na escolha e aplicação de recursos, aumentando a qualidade do plantio e conseqüentemente, rentabilidade, uma vez que otimiza o processo de tomada de decisão do produtor.

Cabe citar ainda que podem ser instalados sensores nos equipamentos que permanecem conectados em rede, melhorando a produtividade, pois ajuda os produtores a controlar as atividades desde o plantio até o armazenamento e logística da cadeia produtiva [20].

Para Costa et al. [21] a IoT destinada ao atendimento do setor de agronegócio favorece o controle de todos os níveis da cadeia produtiva, detectando e monitorando a produção, além de acompanhar o processamento de alimentos. Este gerenciamento contribui com a identificação de falhas e problemas que precisam ser sanados rapidamente para não impactar a produção. Além disso, atua no controle de pragas e infecções, bem como de variáveis agrometeorológicas.

2.1.2. A conectividade

Selecionar a tecnologia de conectividade mais adequada é uma das decisões críticas que as empresas precisam tomar em sua estratégia de lançamento da IoT, pois a conectividade é um componente importante em uma solução de IoT de ponta a ponta [19]. Pois, há uma variedade de tecnologias disponíveis para IoT, como celular tradicional (3G/4G), área ampla de baixa potência, Wi-Fi, Bluetooth e muito mais [26]. Todo caso de uso tem necessidades específicas, que se traduzem em certos requisitos de tecnologia que determinam a escolha da tecnologia de conectividade mais adequada. Segundo Santos [22] esses requisitos de tecnologia podem ser agrupados em três categorias: técnica, comercial e relacionada ao ecossistema, fornecendo uma abordagem estruturada que as empresas podem usar para analisar seus requisitos.

Os requisitos técnicos envolvem cobertura, eficiência energética, taxa de dados, outros recursos relevantes para aplicações específicas (comunicação bidirecional, mobilidade, localização). Os requisitos comerciais tratam da qualidade de serviço, custo, segurança, escalabilidade. Por sua vez, os requisitos do ecossistema abrangem prova futura, alcance global e interoperabilidade [21].

Atualmente, as principais tecnologias de comunicação utilizados na IoT são:

- WI-FI: Baseado no padrão IEEE 802.11, foi desenvolvido como uma opção sem fio para o popular padrão cabeado IEEE 802.3. Utilizando a arquitetura TCP/IP e possuindo grande compatibilidade garantida pela Wi-Fi Alliance, conecta-se de forma nativa com a Internet, estando atualmente já integrado aos notebooks, TVs e smartphones. O Wi-Fi possui razoável alcance de comunicação e excelente capacidade de tráfego de dados, mas possui implementação por código complexa, de requisitos de processamento altos e elevado consumo de energia, o que dificulta sua implementação em equipamentos pequenos e baratos [6].

- Bluetooth: Criado principalmente para conexões de curta distância, possui baixo consumo de energia, aceita várias conexões simultâneas e opera com baixas taxas de transferência de dados. O Bluetooth Low Energy é uma versão da tecnologia em que, reduzindo-se a capacidade de transmissão de dados, muitas vezes não necessária para alguns dispositivos IoT, reduz drasticamente o consumo de energia, tornando o equipamento ao qual aplicado capaz de manter-se funcional por períodos muito maiores [6].

- ZigBee: Gerenciada e certificada pela ZigBee Alliance, o protocolo possui baixa transmissão de dados, baixo consumo de energia, baixo custo e devido a sua topologia de funcionamento consegue conectar milhares de pontos ao mesmo tempo, incluindo a internet através de um node específico adicionado a rede, tornando-a eficaz em muitos casos de IoT como Automação residencial [7].

Apesar do contínuo avanço da tecnologia e o aumento do poder de processamento e memória, nenhuma tecnologia ou solução única é ideal para atender a todos os possíveis casos de uso da IoT, e certas tecnologias coexistem, como complemento, e não como padrões concorrentes.

De acordo com a análise realizada por Brown [7] para implantações de IoT em áreas remotas e amplas podem ser adotadas as tecnologias: LoRa (Lo ng Ra nge) é o protocolo de comunicação sem fio de longo alcance; Narrowband IoT

(NB-IoT) destinada para conectividades de baixa complexidade onde a 3G e 4G não é eficaz como ambientes muito fechados; e Long Term Evolution for Machines (LTE-M) que coexistem com as redes móveis proporcionando a integridade de dados, e são bons complementos e, juntas, abordarão uma grande parte desse mercado.

O ecossistema dinâmico aberto da LoRa é ideal para redes privadas com implantação personalizada, enquanto o NB-IoT e o LTE-M são apoiados pelas principais operadoras móveis, oferecendo conectividade padronizada com alcance global. Outras tecnologias proprietárias, como a Sigfox, podem abordar certos segmentos de nicho, mas sua prova futura deve ser testada pelo tempo. Para aplicativos que requerem maior taxa de transmissão, as opções de tecnologia mais adequadas são LTE ou Wi-Fi, dependendo do escopo das implantações de IoT [7].

2.2. A TECNOLOGIA IBM WATSON

O IBM WATSON é uma plataforma de computação cognitiva desenvolvida originalmente pela IBM para responder a perguntas sobre o Jeopardy². O WATSON foi construído no projeto DeepQA da empresa, e seus recursos avançados de resposta a perguntas (QA) estão sendo utilizados agora pela IBM e pelos clientes corporativos em uma variedade de aplicativos [7].

WATSON recebeu o nome do primeiro CEO da IBM, Thomas J. WATSON, e a plataforma WATSON foi desenvolvida para receber perguntas que são expressas em linguagem natural e, em seguida, utilizar algoritmos, inteligência artificial e uma grande quantidade de dados (Big Data) para entender completamente as perguntas e retornar a resposta o mais precisa possível à pergunta.

A IBM forneceu ao WATSON milhões de documentos, incluindo enciclopédias, dicionários, obras literárias, bancos de dados, taxonomias, artigos de notícias e outros materiais de referência para formar sua base de conhecimento [24]. Hoje, a tecnologia IBM Watson expandiu-se para uma ampla variedade de aplicativos comerciais, incluindo a condução de novos recursos da IoT com

² É um site em que os usuários podem criar e personalizar seus próprios modelos de jogos de risco gratuitamente.

Blockchain, segurança, nuvem e outras tecnologias para obter informações dos bilhões de sensores incorporados em máquinas, eletrodomésticos, carros e até hospitais.

A IBM também está aplicando o WATSON à cognição incorporada, onde está implementando a inteligência artificial (IA) do WATSON, incluindo a utilização do WATSON em robôs, avatares e espaços. A plataforma WATSON apresenta diversas possibilidades, a qual permite modelos customizados, que facilita a construção de modelos e classificadores baseados em *machine learning* - processados na nuvem [12].

Assim, faz uso da computação cognitiva (CC) que adota modelos computadorizados para simular o processo de pensamento humano em situações complexas em que as respostas podem ser ambíguas e incertas. Em geral, o termo 'computação cognitivo' tem sido usado para se referir a novos hardwares e/ou softwares que imitam o funcionamento do cérebro humano e melhoram os recursos humanos [7].

A CC entende a escrita e fala natural, avaliando todo seu contexto, entonação e subjetividades. O real entendimento do sentido geral da informação, e não somente de palavras individuais, faz WATSON, em semelhança ao processo humano, buscar conhecimento, criar hipóteses, avaliá-las e então atuar conforme necessário.

Além disso, a CC descreve plataformas tecnológicas que, em termos gerais, são baseadas nas disciplinas científicas da inteligência artificial e processamento de sinais. Essas plataformas abrangem aprendizado de máquina, raciocínio, processamento de linguagem natural, reconhecimento de objetos, interação homem-computador, geração de diálogo e narrativa, entre outras tecnologias

Nesse sentido, a CC é um novo tipo de computação que tem o objetivo de dispor de modelos mais precisos de como o cérebro/mente humana sente, raciocina e responde ao estímulo. Os aplicativos da CC vinculam a análise de dados e a Adaptive Page Display (APD) para ajustar o conteúdo para um tipo específico de público. Dessa forma, o hardware e os aplicativos da CC se esforçam para ser mais efetivos e mais influentes no design [7].

De acordo com Pinto [24], alguns recursos que os sistemas cognitivos podem expressar são:

- Adaptativo: Eles podem aprender à medida que as informações mudam e à medida que as metas e os requisitos evoluem. Resolvem ainda ambiguidades e toleram imprevisibilidade, além de serem projetados para alimentar dados dinâmicos em tempo real ou quase em tempo real.

- Interativo: Interagem facilmente com os usuários para que possam definir suas necessidades confortavelmente. Eles também podem interagir com outros processadores, dispositivos e serviços em nuvem, bem como com pessoas.

- Iterativo e com estado: Ajudam na definição de um problema, fazendo perguntas ou encontrando informações adicionais da fonte, se uma declaração do problema for ambígua ou incompleta. Conseguem ainda lembrar as interações anteriores em um processo e retornar informações adequadas para o aplicativo específico naquele momento.

- Contextual: Eles podem entender, identificar e extrair elementos contextuais, como significado, sintaxe, hora, local, domínio apropriado, regulamentos, perfil do usuário, processo, tarefa e objetivo. Recorrem a várias fontes de informação, incluindo informações digitais estruturadas e não estruturadas, bem como entradas sensoriais (visual, gestual, auditiva ou fornecida pelo sensor).

Cabe ressaltar que a plataforma WATSON analisa perguntas em diferentes palavras-chave e fragmentos de frases, a fim de encontrar frases estatisticamente relacionadas. A principal finalidade de WATSON não está na criação de um novo algoritmo para esta operação, mas na capacidade de executar rapidamente centenas de algoritmos de análise de linguagem comprovados simultaneamente.

Quanto mais algoritmos encontrarem a mesma resposta independentemente, maior a probabilidade de o WATSON estar correto. Uma vez que o WATSON possui um pequeno número de possíveis soluções, ele pode verificar seu banco de dados para verificar se a solução faz sentido ou não.

2.2.1. IBM Cloud

O IBM Cloud é uma plataforma em nuvem constituída pelos projetos de código aberto mais populares do mundo que permite construir e executar aplicativos.

Com modelos de implantação integrados abrangendo nuvens públicas, dedicadas, locais e híbridas, representando uma nova maneira de experimentar a nuvem.

A computação em nuvem IBM é um conjunto de serviços de computação em nuvem para negócios oferecidos pela empresa de tecnologia da informação IBM. A nuvem IBM inclui infraestrutura como serviço (IaaS), software como serviço (SaaS) e plataforma como serviço (PaaS) oferecida por modelos de entrega em nuvem pública, privada e híbrida, além dos componentes que compõem essas nuvens [24].

A IBM oferece três plataformas de hardware para computação em nuvem. Essas plataformas oferecem suporte interno para virtualização. Neste contexto, a IBM oferece soluções de infraestrutura de aplicativos IBM Websphere que suportam modelos de programação e padrões abertos para virtualização.

A camada de gerenciamento da estrutura em nuvem IBM inclui o middleware IBM Tivoli. As ferramentas de gerenciamento fornecem recursos para controlar as imagens com provisionamento e desprovisionamento automatizados, monitorar operações e uso do medidor, acompanhando os custos e alocando o faturamento [7].

A última camada da estrutura fornece ferramentas integradas de carga de trabalho. Cargas de trabalho para computação em nuvem são serviços ou instâncias de código que podem ser executadas para atender a necessidades comerciais específicas. A IBM oferece ferramentas para colaboração, desenvolvimento e teste baseados em nuvem, desenvolvimento de aplicativos, análises, integração business-to-business e segurança.

2.2.2. História

A computação em nuvem da IBM surgiu da união de suas tecnologias de computação e virtualização de mainframe. Conhecida como a empresa de virtualização original, os primeiros experimentos da IBM em virtualização ocorreram na década de 1960 com o desenvolvimento da máquina virtual (VM) nos sistemas operacionais CP-40 e CP-67 [7]. O CP-67, um hipervisor usado para teste e desenvolvimento de software, habilitou o compartilhamento de memória entre VMs, dando a cada usuário seu próprio espaço de memória virtual. Com a máquina

particionada em VMs separadas, os mainframes podem executar vários aplicativos e processos ao mesmo tempo, tornando o hardware mais eficiente e econômico. A IBM começou a vender a tecnologia VM para o mainframe em 1972 [24].

Em fevereiro de 1990, a IBM lançou os servidores baseados no RS / 6000 (que mais tarde ficou conhecido como processador POWER). Os servidores, em combinação com o mainframe IBM, foram criados para virtualização complexa e de missão crítica. Os servidores de sistemas de energia incluem hipervisores PowerVM com mobilidade de partição ativa e compartilhamento de memória ativa. A migração ao vivo foi introduzida com o POWER6 em maio de 2007 [24].

Em seguida, a IBM procurou implementar padronização e automação em sua tecnologia, a fim de acompanhar a proliferação de dados produzidos por hardware e data centers cada vez mais eficientes. Essa combinação de virtualização, padronização e automação levou ao desenvolvimento da computação em nuvem IBM.

A IBM começou a desenvolver uma estratégia para a computação em nuvem em 2007, anunciando que planejava criar nuvens para clientes corporativos e fornecer serviços para preencher o que considerava lacunas nos ambientes em nuvem existentes. Em outubro de 2007, a IBM anunciou uma parceria com o Google para promover a computação em nuvem nas universidades. Além de doar hardware e máquinas, as duas empresas também forneceram um currículo para ensinar aos alunos sobre computação em nuvem

A IBM alegou em abril de 2011 que 80% das empresas da Fortune 500 estavam usando a nuvem IBM e que seus softwares e serviços foram utilizados por mais de 20 milhões de clientes finais, com clientes como American Airlines, Aviva, Carfax, Frito-Lay, India Companhia de seguros de vida e 7-Eleven. Em 4 de junho de 2013, a IBM anunciou sua aquisição do SoftLayer, para formar uma Divisão de Serviços em Nuvem IBM. Em 4 de março de 2014, a IBM adquiriu o Cloudant. Além disso, ele oferece aos desenvolvedores acesso instantâneo a mais de 150 serviços incluindo IoT e APIs de WATSON.

2.3. MICROSOFT AZURE

Para Sosinsky (2011, p. 3) [25] “a computação na nuvem refere-se a aplicativos que são executados pela internet, utilizando virtualização, abstraindo os recursos físicos para o usuário”. O Microsoft Azure é uma plataforma de serviços da Microsoft na nuvem, isto é, tecnologias, nas quais fornecem um servidor e infraestrutura necessária para armazenar, visando recursos que ocorrem em tempo real, bem como possibilita serviços para o desenvolvimento de aplicativos [23] [24].

O Azure foi anunciado em outubro de 2008, iniciado com o codinome "Project Red Dog", e lançado em 1 de fevereiro de 2010 como "Windows Azure" antes de ser renomeado "Microsoft Azure" em 25 de março de 2014 [25]. Os componentes da Microsoft Azure podem ser usados em aplicativos locais, em ambientes virtualizados, não dependendo do hardware e não afetando, por conseguinte, a execução de outros aplicativos. Sendo assim, cada componente da Plataforma desempenha seu respectivo papel [26] [27].

O Microsoft Azure é um serviço de computação em nuvem criado pela Microsoft para criar, testar, implantar e gerenciar aplicativos e serviços por meio de data centers gerenciados pela Microsoft. Ele fornece software como serviço (SaaS), plataforma como serviço (PaaS) e infraestrutura como serviço (IaaS) e suporta muitas linguagens de programação, ferramentas e estruturas diferentes, incluindo software e sistemas específicos da Microsoft e de terceiros [26].

Para IaaS, o Azure permite que os usuários iniciem máquinas virtuais Microsoft Windows e Linux de uso geral, bem como imagens de máquinas pré-configuradas para pacotes de software populares. A maioria dos usuários executa o Linux no Azure, algumas das muitas distribuições Linux oferecidas, incluindo o Azure Sphere da Microsoft, baseado na Linux. Para PaaS, os desenvolvedores podem publicar e gerenciar sites com facilidade [25].

Os clientes podem criar sites em PHP, ASP.NET, Node.js ou Python, ou selecionar entre vários aplicativos de código aberto de uma galeria para implantar. Isso inclui um aspecto da oferta de plataforma como serviço (PaaS) para a plataforma Microsoft Azure. Foi renomeado para Web Apps em abril de 2015 [26]. Os WebJobs são aplicativos que podem ser implantados em um ambiente do Serviço de Aplicativo para implementar o processamento em segundo plano que

pode ser chamado em uma programação, sob demanda ou executado continuamente. Os serviços Blob, Tabela e Fila podem ser usados para se comunicar entre WebApps e WebJobs e para fornecer estado [25].

2.4. REST

REST ou RESTful (Transferência do Estado Representativo) é uma técnica de desenvolvimento de software utilizada em sistemas hipermídios distribuídos como a rede de internet. Sistemas que seguem os princípios REST são referenciados também como “RESTful”. O padrão REST efetua sua comunicação cliente-servidor de forma simples com a utilização de URIs e requisições HTTP sendo elas GET/POST/DELETE/PUT/HEAD [28].

É uma plataforma que pretende ser vista como uma imagem do design de como a aplicação se comportará: uma rede de websites (um estado virtual), onde o usuário progride com uma aplicação selecionando as ligações (transições do estado), tendo como resultado a página seguinte (que representa o estado seguinte da aplicação) que está sendo transferida ao usuário e apresentada para seu uso [29].

O termo REST é geralmente usado para descrever qualquer interface que envie dados HTTP sem uma camada adicional de mensagem como SOAP³ ou *session tracking*⁴ via *cookies*⁵. É possível desenvolver um sistema de software de acordo com as restrições impostas pelo estilo arquitetural REST usando dados HTTP e interagindo com a Web. Também se torna possível projetar interfaces HTTP + XML que não condizem com os princípios REST [30].

³ É uma especificação de protocolo de mensagens para troca de informações estruturadas na implementação de serviços da web em redes de computadores. Seu objetivo é proporcionar extensibilidade, neutralidade e independência.

⁴ *Session tracking* é um mecanismo usado para manter o estado sobre uma série de pedidos do mesmo usuário, ou seja, os pedidos provenientes do mesmo navegador em algum período de tempo (THE JAVATM TUTORIAL, 1999).

⁵ *Cookies* é um pequeno texto de identificação enviado do servidor (página acessada), utilizado para posterior acesso, como por exemplo, dados de senhas e logins salvos pelo navegador (MOZDEV, 2014).

2.5. TELEGRAM

O aplicativo TELEGRAM foi lançado no ano de 2013, atuando no processo de comunicação de forma rápida e segura a partir de chats privados que permite que os usuários enviem mensagens textuais, fotos, vídeos e outros arquivos. Também possibilita a criação de grupos com maior quantidade de usuários. Possui uma infraestrutura robusta e um sistema de criptografia eficiente em razão dos múltiplos data centers, o que assegura o compartilhamento dos arquivos em nuvem [31].

Cabe citar que proporciona aos seus usuários uma comunicação instantânea, criando verdadeiros ambientes de interação social, pois os grupos possuem uma capacidade para até 200 mil indivíduos. Por estar alocado em nuvem, não há a preocupação em perder informações ou efetuar backups. Com isso, o aplicativo atua como um gerenciador de arquivos, minimizando os problemas como memória e processamento [32].

3. A PLATAFORMA MILKBOT

Neste capítulo serão descritas as características e tecnologias utilizadas para o desenvolvimento da Plataforma Milkbot. Cabe ressaltar que é um bot, o qual possui a sua funcionalidade ligada a sensores, nos quais são captados dados referentes à temperatura, à umidade e à qualidade do ar. Deste modo, permite a visualização das informações do dispositivo, com suporte para ser conectado a novos sensores. Todos os códigos desenvolvidos no presente trabalho encontram-se disponíveis no GitLab⁶.

3.1. HARDWARE UTILIZADO NA PLATAFORMA

A seguir apresentam-se os materiais utilizados na montagem do hardware, dentre os quais destacam-se: Raspberry PI Zero, chip MPC3008, sensor de luminosidade LDR, sensor de CO modelo MQ-7 e o sensor de temperatura e umidade do ar DTH11. A Figura 2 traz uma visão do Raspberry PI Zero rodando Raspbian, com uma linguagem Python para a leitura dos valores dos sensores a cada 5 minutos. A escolha do Raspberry PI se deu em razão de proporcionar um processamento mais potente e um hardware mais completo e com maiores possibilidades de expansão do projeto a um custo reduzido.

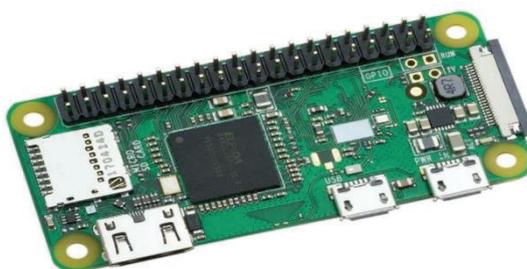


Figura 3. Raspberry PI Zero. [33]

O Raspberry PI Zero apresenta como especificações um processador

⁶ Disponível em:< <https://gitlab.com/luisbraga19/bot>>

Broadcom BCM2835 ARM11 de 1GHz Single-core, a memória: 512MB LPDDR2 RAM, o wifi: 802.11 b/g/n e, por fim, o GPIO de 40 pinos.

3.1.1. MPC3008

Na Figura 4 pode ser visualizado o componente chip MPC 3008. É um chip que disponibiliza portas analógicas (leitura de variações de voltagem) para o Raspberry PI, convertendo estes valores em valores digitais este chip é responsável pelas leituras dos sensores de CO, MQ-7 e o sensor de luminosidade LDR.

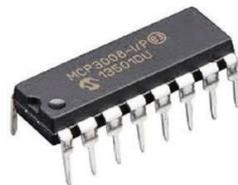


Figura 4. Chip mpc3008. [34]

3.1.2. LDR

O sensor de luminosidade utilizado pode ser observado na Figura 4. Este sensor é conhecido como LDR, o qual retorna um valor analógico. Basicamente é um resistor, cujo valor varia com a luminosidade do ambiente.



Figura 5. Sensor LDR. [35]

3.1.3. MQ-7

O MQ-7 é um sensor que tem por finalidade de medir a concentração de monóxido de carbono (CO). Sua capacidade de detecção tem a precisão de 10-10.000ppm. O mesmo pode ser visualizado conforme a Figura 6.



Figura 6. Sensor MQ7. [36]

3.1.4. DTH11

A Figura 7 demonstra o sensor DTH 11, o qual inclui um componente medidor de umidade e um componente NTC para temperatura, ambos conectados a um controlador de 8-bits.

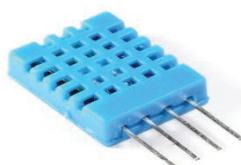


Figura 7. Sensor DTH 11. [37]

O interessante neste componente é o protocolo usado para transferir dados de temperatura e umidade do ar. É utilizado de modo diferente dos outros, pois disponibiliza informações digitais dos valores lidos, não precisando do MPC3008. Os resistores são resistores de 10k em serie para formar um divisor de tensão para que a voltagem do sensor MQ-7 seja reduzida de 5 volts para 3.3 volts que é a tensão de operação do Raspberry PI.


```

1 import Adafruit_DHT
2 import RPi.GPIO as GPIO
3 import time
4 import json
5 import requests
6
7 import spidev
8 from numpy import interp
9 import RPi.GPIO as GPIO
10 spi = spidev.SpiDev()
11 spi.open(0,0)
12 GPIO.setmode(GPIO.BOARD)
13 def analogInput(channel):
14     spi.max_speed_hz = 1350000
15     adc = spi.xfer2([1,(8+channel)<<4,0])
16     data = ((adc[1]&3) << 8) + adc[2]
17     return data
18
19 sensor = Adafruit_DHT.DHT11
20 pino_sensor = 2
21 def enviaLeitura(sensor, valor):
22     headers = {'Content-Type': 'application/json'}
23     params = {'sensor': sensor, 'read': valor.__str__()}
24     url = "https://66.42.33.81/read"
25     response = requests.post(url, data=json.dumps(params), headers=headers)
26
27 umid, temp = Adafruit_DHT.read_retry(sensor, pino_sensor);
28 luz = analogInput(1)
29 luz = interp(luz, [0, 1023], [100, 0])
30 coDois = analogInput(0)
31 coDois = interp(coDois, [0, 1023], [10, 10000])
32 enviaLeitura("co2", coDois);
33 enviaLeitura("temperatura", temp);
34 enviaLeitura("umidade", umid);
35 enviaLeitura("luz", luz);

```

Figura 9. Programação Raspberry PI.

Nesse código visualiza-se o sistema responsável por capturar dados dos sensores. Observa-se que no intervalo das linhas 27 a 31 são efetuadas as leituras das portas analógicas. Já no intervalo das linhas 21 a 26 é criada a função que comunica com o servidor. Esta função é utilizada nas linhas 32 a 35, após a leitura dos sensores e envia ao servidor.

3.3. APLICAÇÃO NO SERVIÇO WATSON ASSISTANT

Os serviços desenvolvidos pelo WATSON Assistant possuem inteligência cognitiva avançada. Deste modo, utiliza-se o NLU (*Natural Language*

Understanding), o qual considera-se com um raciocínio automatizado e tecnologia de aprendizado de máquinas.

As habilidades contêm o treinamento para responder às perguntas desenvolvidas, onde são criadas as habilidades que o WATSON usa para identificar a intenção para o desenvolvimento de análise do fluxo de conversação. Este fluxo é o conjunto da base de conhecimento e a lógica treinada, onde o *assistants* é o conjunto de *Skills*. O conjunto agrupa as bases de conhecimento como um único assistente, visando dar ao mesmo uma ou mais habilidades para poder lidar com diversos cenários, o que pode ser visualizado na Figura 10.

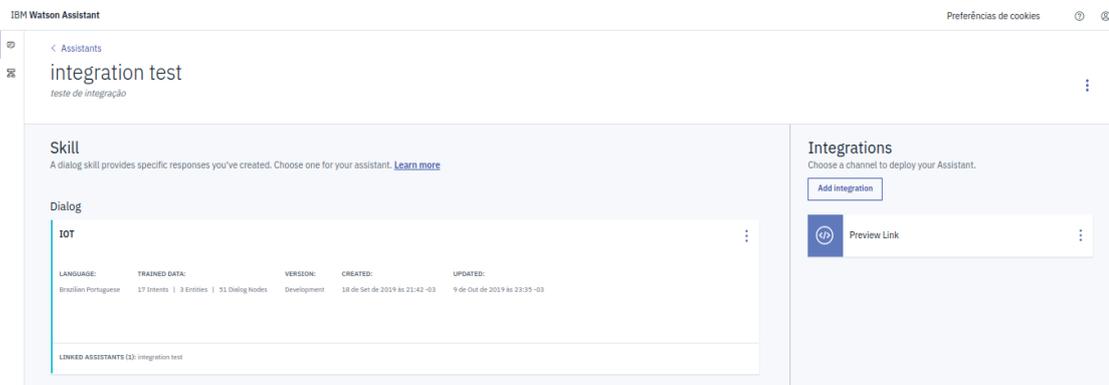


Figura 10. Fluxo raiz do da skill associada ao serviço do Assistant do WATSON.

Uma das características mais importantes do WATSON Assistant é o seu gerenciador de diálogos, chamado de Dialog, como demonstrado na Figura 11.

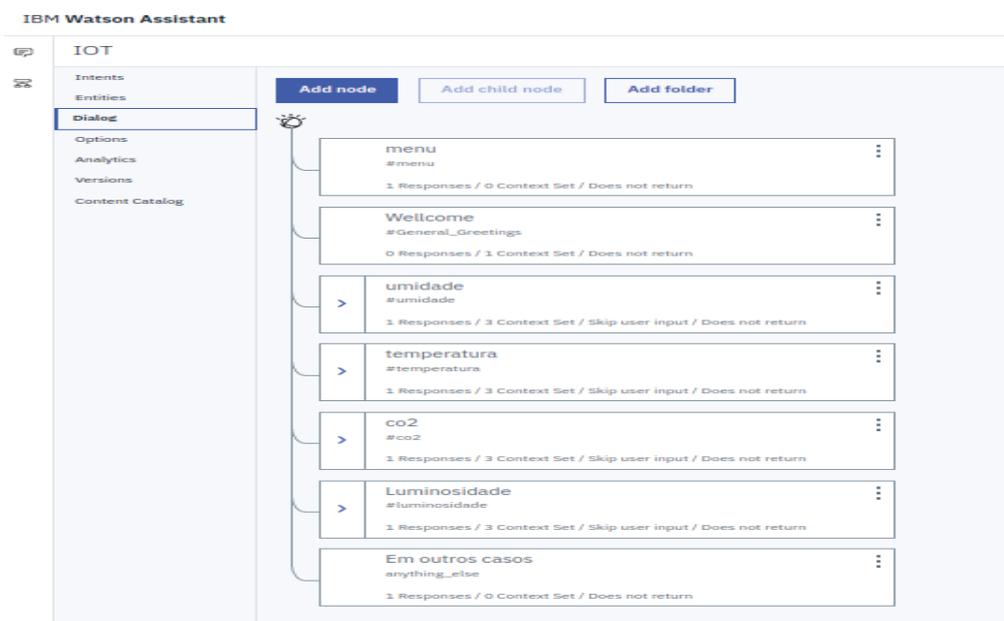


Figura 11. Painel de diálogo da interação do Serviço.

O Dialog é uma interface web para edição do fluxo de diálogo, equiparando-se a uma árvore de decisões. A ordem da captura de dados de diferentes sensores é realizada de cima para baixo e da esquerda para a direita.

No momento em que o usuário digita algo e envia para o chatbot, o WATSON Assistant aciona a IA para reconhecer a intenção e as entidades citadas. Após esta análise, os nós do fluxo de diálogo são verificados, onde todos os fluxos de conversação foram desenvolvidos com as mesmas perguntas.

A seguir na Figura 12 visualiza-se o painel de diálogo.

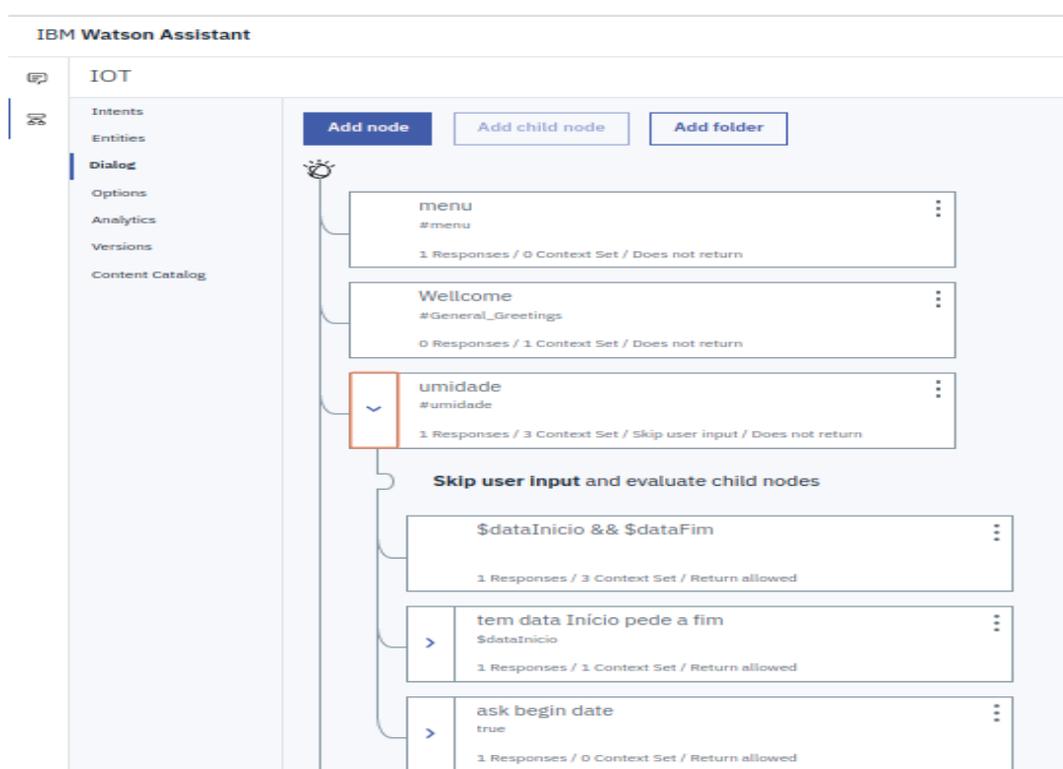


Figura 12. Painel de diálogo do Serviço.

Ao estabelecer a correspondência de intenções e/ou entidades da frase com a condição de algum dos nós, o WATSON Assistant executa a ação programada onde o filtro atua como sensor, data inicial e data final, dando continuidade ao fluxo de diálogo. Os nós do Dialog são normalmente compostos por situações e respostas, cujas perguntas serão utilizadas na filtragem dos registros que serão utilizados nos cálculos.

3.3.1. Aplicação de serviços adicionais utilizados

Sobre a aplicação de serviços adicionais utilizou-se o *text to speech* inicialmente, conforme a Figura 13. Neste caso, o resultado da conversão é enviado para o *text to speech*, tendo em vista possibilitar gerar um áudio semelhante ao humano, relacionando o texto escrito com as interfaces de programação.

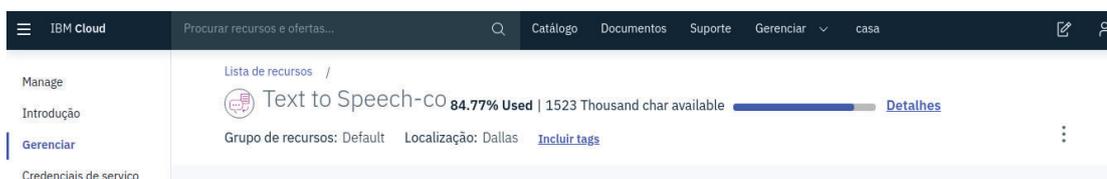


Figura 13. Text to Speech, responsável pela conversão de texto para áudio.

Quando o servidor recebe um áudio, este executa o serviço *speech to text*, conforme visualiza-se na Figura 14.

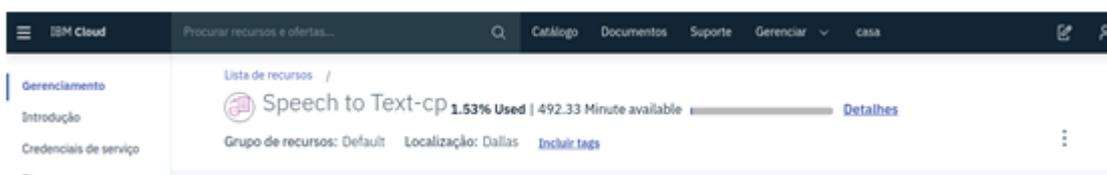


Figura 14. Speech to text, responsável pela conversão de áudio para texto.

O *speech to text* transcreve automaticamente o áudio, o qual nem sempre tem uma boa qualidade, independente do formato, possibilitando identificar rapidamente o que está sendo falado. Sendo assim, reconhece o que foi enviado e devolve em forma de texto, iniciando-se o fluxo da conversação.

3.4. INTEGRAÇÃO

A IBM disponibiliza para diversas linguagens de programação tais como Curl, NET, GO, Java, Python, Ruby, Node. Essas linguagens possibilitam a

integração dos serviços, atuando no atendimento, monitoramento, gerenciamento e outras aplicações. A escolha do tipo de tecnologia depende de vários fatores, sendo que neste estudo considerou-se a simplificação do processo e a funcionalidade.

Nesta implementação optou-se em utilizar NodeJs, tendo em vista a redução de custos com infraestrutura e o aumento do desempenho de projetos e a capacidade que tem em processar conexões simultâneas utilizando apenas uma única *thread*. Apresenta uma programação assíncrona que compartilha recursos e um I/O não bloqueante.

Utilizado a API disponibilizada pela IBM para comunicação com o sistema baseada em node⁷, como demonstrado na Figura 15 o IBM Cloud baseia-se no Cloud Foundry.

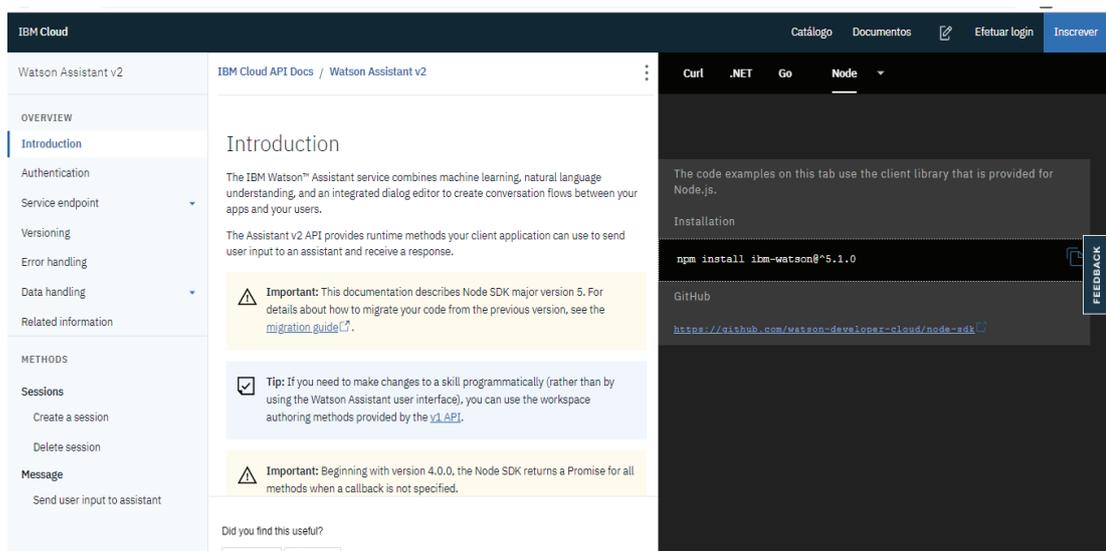


Figura 15. Painel integração IBM Cloud.

É possível aproveitar o ecossistema de frameworks e serviços de tempo de execução em crescimento, a partir do momento em que o IBM Cloud fornece um painel possibilitando criar, visualizar, gerenciar seus aplicativos e serviços e monitorar o uso dos recursos do seu aplicativo.

Para exibição dos resultados foram utilizados Adaptive cards, conforme visualiza-se na Figura 16. É neste ponto que o bot faz as chamadas dos serviços de terceiros para fazer o que ele se propõe.

⁷ Disponível em: <<https://cloud.ibm.com/apidocs/assistant/assistant-v2?code=node>>.

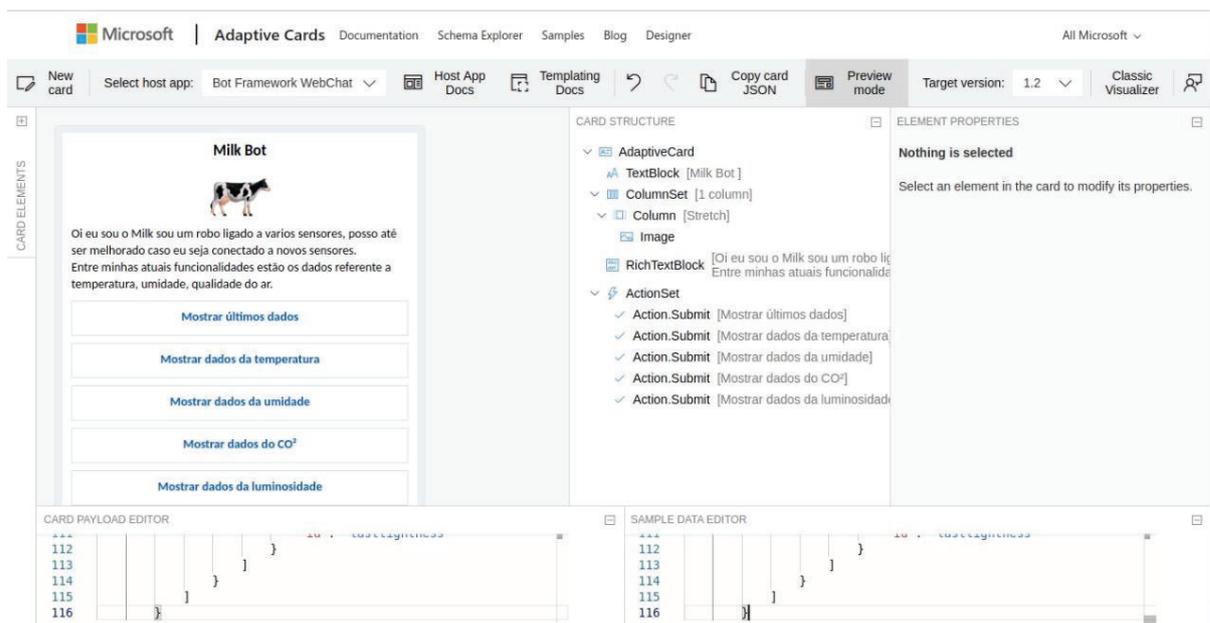


Figura 16. Utilizados Adaptive cards.

Forma gráfica de exibição, na qual os cartões adaptáveis são trechos de interface do usuário independentes de plataforma, criados em JSON, bem como refere que aplicativos e serviços podem trocar abertamente. Quando entregue a um aplicativo específico, o JSON é transformado na interface do usuário nativa que se adapta automaticamente ao ambiente, bem como auxilia a projetar e integrar a interface de usuário leve para todas as principais plataformas e estruturas.

3.5. BANCO DE DADOS

O banco de dados escolhido para ser utilizado no projeto MilkBot foi o Mongo que possibilita o uso de dois documentos (um para armazenar os sensores e outro para armazenar as leituras que esse sensor informou ao servidor). Cita-se que o mongoDB⁸, é um banco de dados NOSQL, conceito novo de armazenamento de dados em estrutura dinâmica. A hospedagem do banco de dados utilizado foi a Atlas, que é uma plataforma que disponibiliza de forma gratuita 600MB.

⁸ Disponível em: <<https://www.mongodb.com/cloud/atlas>>.

Na Figura 17, demonstra-se a preparação da base dados para o recebimento dos dados via REST, onde o *reads* é responsável por armazenar as leituras vindas do Raspberry PI e o e o documento *sensors* é onde ficam cadastrados os tipos de sensores. Os dados são recebidos pelo bot e enviados ao mongodb pelos pontos da api `https://<IP SERVIDOR BOT>/read` e `https://<IP SERVIDOR BOT>/sensor`.

The screenshot shows the MongoDB Atlas interface for a database named 'test'. On the left, there is a sidebar with a search bar and a list of namespaces: 'test', 'reads', and 'sensors'. The main panel displays the 'test' database overview, including a 'CREATE COLLECTION' button and a table of collections.

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
reads	2336	230.41KB	101B	1	76KB	76KB
sensors	4	424B	106B	2	72KB	36KB

Figura 17. Painel de Databases.

Na Figura 18, é demonstrado o registro da listagem, o cadastro de sensores e a inclusão de novos sensores.

The screenshot shows the MongoDB Atlas interface for the 'test.sensors' collection. The main panel displays the collection overview, including a 'FILTER' input field and a 'Find' button. Below the filter, there are two JSON documents representing sensor records.

```

QUERY RESULTS 1-4 OF 4

{
  "_id": ObjectId("5d5f1e4995f7e61b3fba7b52"),
  "sensorType": "co2",
  "scale": "ppm",
  "createdAt": 2019-08-22T22:59:21.357+00:00,
  "updatedAt": 2019-08-22T22:59:21.357+00:00,
  "_v": 0
}

{
  "_id": ObjectId("5d5f47ee95f7e61b3fba7b79"),
  "sensorType": "umidade",
  "scale": "%",
  "createdAt": 2019-08-23T01:57:02.297+00:00,
  "updatedAt": 2019-08-23T01:57:02.297+00:00,
  "_v": 0
}

```

Figura 18. Exemplo de registro de sensor.

Na Figura 19 pode ser visualizado o registro dos dados de leituras, registrada por um dos sensores.

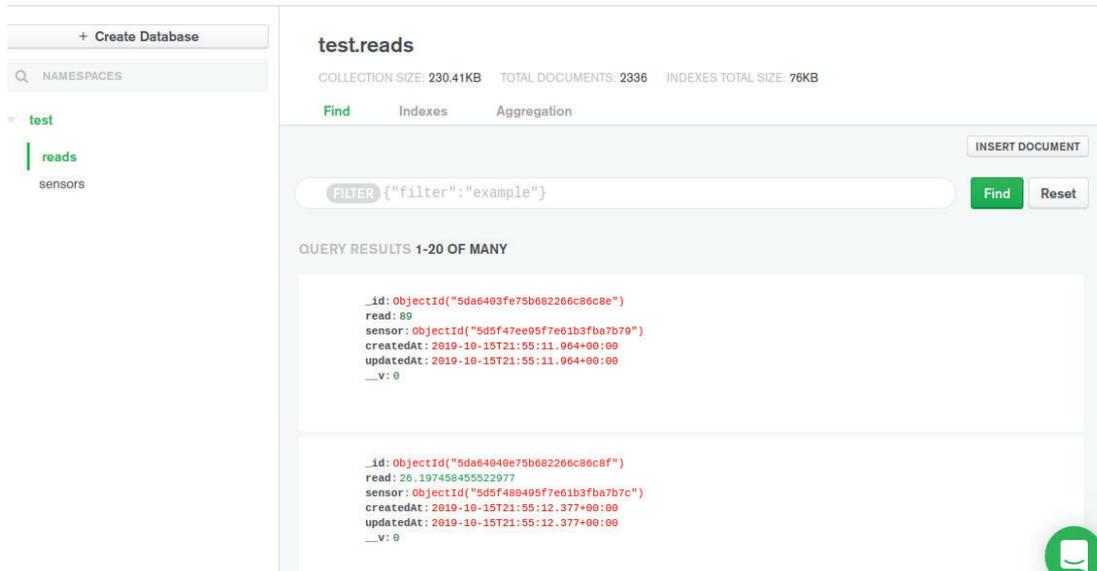


Figura 19. Exemplo de registro de leitura.

3.6. APLICAÇÃO

No tocante à aplicação do hardware foi criado e utilizado Node.js, com a seguinte estrutura, como observa-se na Figura 20.

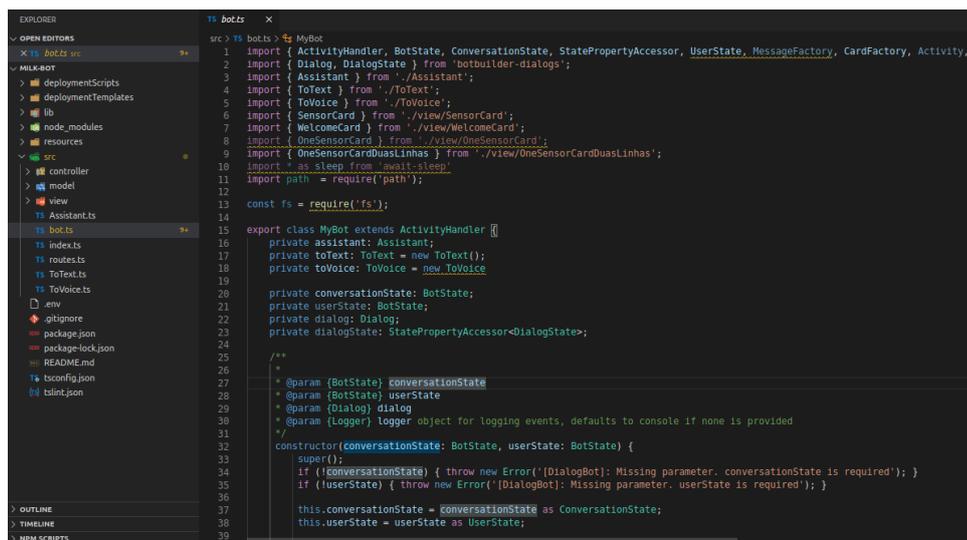


Figura 20. Estrutura do Projeto com a utilização do Node.js.

Para o desenvolvimento da API, a qual consome o serviço do WATSON Assistant, deve ser instalado em sua máquina o Node.js e o NPM. O NPM (Node

Package Manager) é apenas um pacote de gerenciamento de módulos de códigos JS para instalar junto ao Node.js. Após, suas aplicações precisam ser incluídas junto ao Node, visto ser muito usado no lado do servidor para atender as demandas de aplicações web.

As bases de dados NoSQL são baseadas em JSON (JavaScript Object Notation), portanto, sua comunicação com Node.js é bastante intuitiva e fácil de ser implementada. Com isso, não é necessário converter modelos de dados, pois os mesmos dados podem ser enviados para o *front-end* sem a necessidade de nenhum tipo de tratamento ou conversão.

3.7. SERVIDOR DE AUTENTICAÇÃO DO BOT

A seguir é apresentado o servidor de autenticação do bot, que se encontra integrado ao Azure (Registro de Canais de Bot). A partir desta integração, o Azure se comunica com o bot (escrito em node hospedado na Vultr)⁹ e faz interação com o TELEGRAM. O processamento realizado pelo bot ocorre a partir da mensagem recebida pelo bot, oriunda do TELEGRAM, utilizando o serviço da Azure, por sua vez ele trata se for uma mensagem de voz ou texto, caso seja de voz é utilizado o serviço *speech-to-text* do WATSON e transcrito para texto, com a mensagem em texto, ela é encaminhada para o WATSON para a interpretação da mensagem, utilizando IA para extrair as variáveis, que no caso são o tipo de sensor a ser consultado e o intervalo de tempo e o mesmo é retornado para o bot para que por sua vez faça a consulta no banco de dados (mongoDB) e retorne para o mensageiro instantâneo (TELEGRAM).

Porém, caso a mensagem inicialmente tenha vindo em forma de voz a mesma também será respondida em forma de voz utilizando o recurso *text-to-speech* do WATSON. Na figura 21 é possível visualizar a interface do usuário do Azure para reconhecer o bot, sendo este configurado com êxito para se comunicar com os usuários no TELEGRAM.

⁹ Serviço de implantação de infraestrutura de servidores por meio de sua plataforma de nuvem.

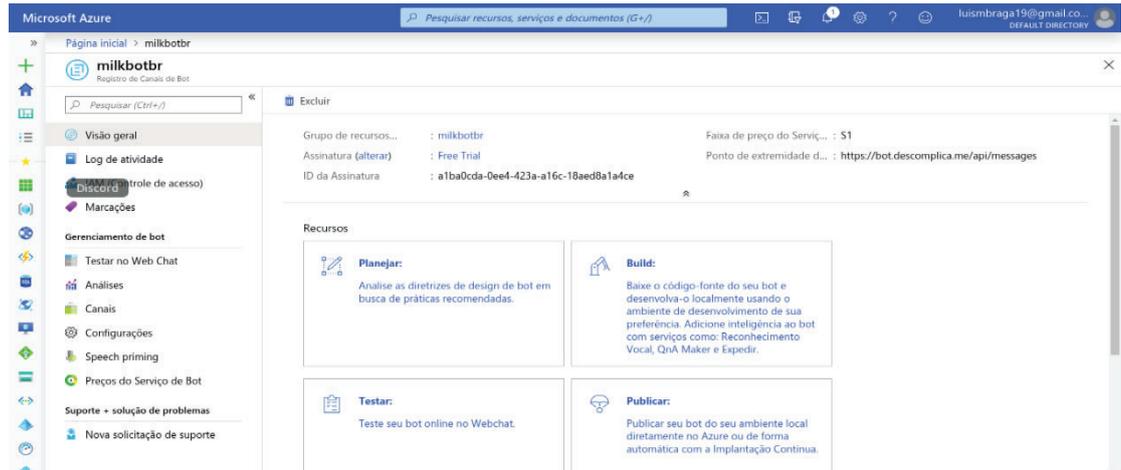


Figura 21. Tela de Interface Portal Azure.

O serviço de Bot do Azure fornece um ambiente onde os mesmos serão executados dentro do TELEGRAM, interagindo com os bots enviando mensagens, comandos e solicitações embutidas.

4. ESTUDO DE CASO

Este capítulo apresenta o estudo de caso desenvolvido com o objetivo de explorar as funcionalidades para a plataforma MilkBot à arquitetura de IoT. Com isso, são descritos os aspectos relacionados à atualização dos componentes e sua função em conformidade às características de um sistema robótico, bem como a gerência de dispositivos.

O estudo de caso foi desenvolvido utilizando a seguinte metodologia: foram conectados a plataforma um dispositivo IoT com web services, a fim de demonstrar que a entrega dos dados via o *bot* para TELEGRAM cumpre o requisito para realizar a comunicação entre esses dispositivos. Também são utilizadas as funcionalidades da plataforma *web services* para enviar comandos e mostrar valores enviados dos sensores.

O hardware utilizado no estudo de caso foi a placa Raspberry PI Zero rodando *cron raspbian*. Foi utilizado conversor A/D mpc3008, sensor de luminosidade, sensor de CO modelo MQ7 e sensor de temperatura e umidade do ar (DTH11), como observa-se na Figura 22.

A escolha da placa Raspberry PI decorre da sua popularidade na utilização em projetos IoT. Ressalta-se ainda que a proposta do projeto é a parte da entrega dos dados via o bot com conjunto de dispositivos em vários locais sem uso de aplicativo próprio, de uso exclusivo, conectado por meio de IoT de maneira simples. Dessa forma, faz-se uso do serviço do TELEGRAM, sendo este baseado mensagens instantâneas baseado na nuvem, para gerenciar os dispositivos e possibilitar a visualização dos dados, os quais são enviados por estes dispositivos.

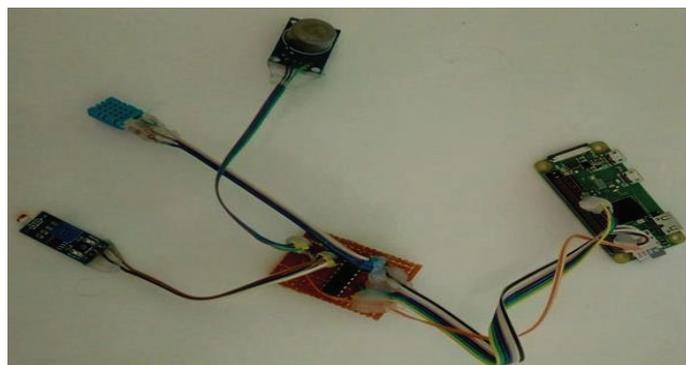


Figura 22. Hardware utilizado.

O estudo foi executado conforme o disposto (Raspberry PI Zero rodando Raspbian) onde realizam-se as leituras dos sensores e envia-se para o servidor do bot através de um serviço REST da aplicação. As leituras dos sensores são iniciadas via gerenciador de tarefas, os quais estão configurados para serem enviados a cada 5 minutos. Conforme esquema, demonstrado na Figura 23. As informações são disponibilizadas via TELEGRAM.

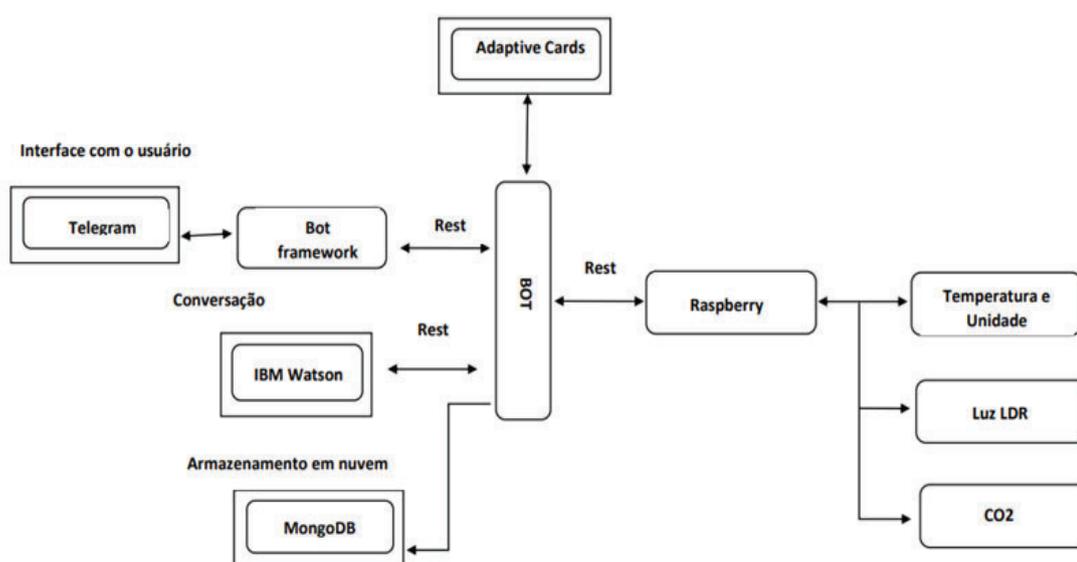


Figura 23. Esquema das tecnologias utilizadas.

O desenvolvimento das interfaces web services seguiu uma metodologia de prototipação com o objetivo de criar analogias que permitissem aos usuários uma melhor visualização. Sendo assim, foram criados comandos de atalhos, isto é, comandos que oferecem praticidade e agilidade que retornam os últimos dados do sensor, dentre os quais destacam-se: a temperatura, a umidade, CO e a luminosidade.

Um exemplo de interação do WATSON com a Plataforma desenvolvida pode ser visualizado na Figura 24, onde percebe-se que nas conversas o WATSON consegue interpretar singularidades de palavras, bem como as datas nominais, por exemplo, o “ontem” que foi traduzido para a data início de 16/10/2019 e data fim para 17/10/2019.

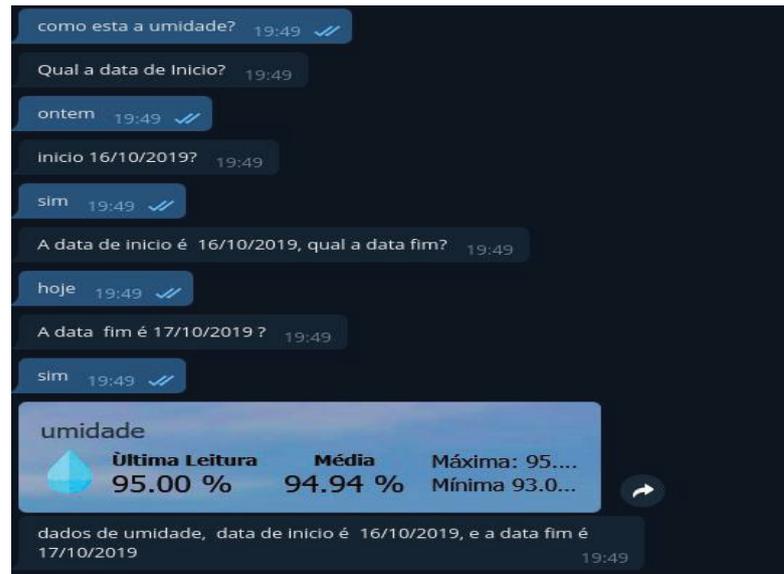


Figura 24. Exemplo de conversa com WATSON e sensor de umidade.

Já na Figura 25, observa-se um exemplo de conversa entre o WATSON e o sensor de luz, na qual o WATSON consegue interpretar singularidades de palavras, bem como a data nominal, por exemplo, o “hoje” que foi traduzido para a data de 17/10/2019.

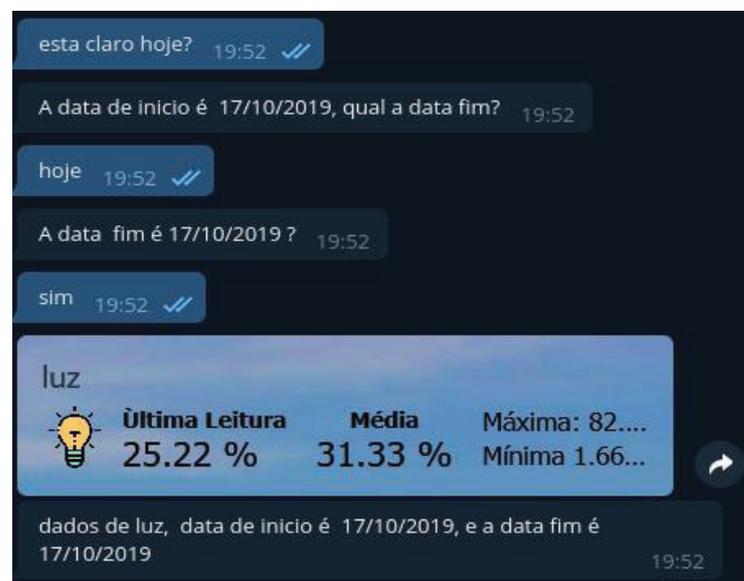


Figura 25. Exemplo de conversa com WATSON e sensor de luz.

Neste cenário, o WATSON também consegue traduzir intervalos, como por exemplo, “semana passada” para o intervalo de 06/10/2019 a 12/10/2019.

A demonstração do resultado do sensor do ícone de concentração de CO, como pode ser visualizado na Figura 26, teve como propósito mostrar a informação em condições normais e com o sinal de alerta da qualidade do ar.

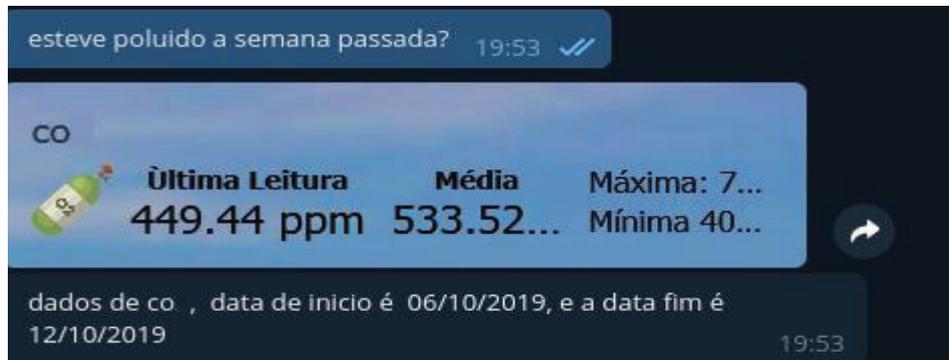


Figura 26. Exemplo de conversa com WATSON e sensor de CO.

A Figura 27 apresenta a interface do aplicativo em relação à temperatura, ressaltando que foi criado um comando com o exemplo “/temperatura” para simplificar o uso e assim acessar direto os últimos valores de leitura do referido sensor, trazendo também a média da temperatura com a mínima e máxima.



Figura 27. Interface do aplicativo mostra o resultado da temperatura.

Importante destacar que do WATSON foi utilizado os recursos de reconhecimento de frases escritas; reconhecimento de frases faladas; emissão de frases em voz do Azure, na qual basicamente é quem recebe os dados do comunicador instantâneo (no caso TELEGRAM) e joga para o bot que está rodando na Vultr. o reconhecimento de frases escritas que entra o IA, já o reconhecimento de voz basicamente é a transcrição da fala para texto e usa o reconhecimento de texto da mesma forma. Estrutura do Projeto com a utilização do Node.js e o código

TypeScript é utilizado somente em ambiente de desenvolvimento e é totalmente convertido para JavaScript.

A interface quanto à umidade pode ser visualizada na Figura 28, na qual criou-se um comando com o exemplo “/umidade” para simplificar o uso do usuário e assim acessar diretamente os últimos valores de leitura do referido sensor, trazendo também a média da umidade com a mínima e máxima.



Figura 28. Interface do aplicativo mostra o resultado da umidade.

A representação em relação ao CO, conforme visualiza-se na Figura 29, apresenta o acesso a um comando com o exemplo “/co” para simplificar o uso do usuário e assim acessar diretamente os últimos valores de leitura do referido sensor, trazendo também a média da CO com a mínima e máxima.



Figura 29. Interface do aplicativo mostra o resultado da CO.

Na Figura 30, destaca-se a interface no tocante a luminosidade, ilustrando que o comando “/luminosidade” permite simplificar o acesso do usuário e, assim acessar diretamente os últimos valores de leitura do referido sensor, apresentando também a média da luminosidade com a mínima e máxima.

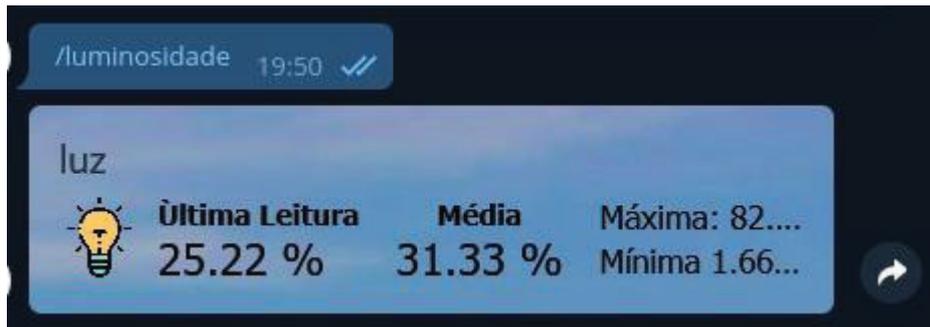


Figura 30. Interface do aplicativo mostra o resultado da luminosidade.

E por fim, assim como os demais atalhos visualizados anteriormente, desenvolveu-se uma interface para obter todos os registros de forma simplificada, conforme exposto na Figura 31.



Figura 31. Interface do aplicativo mostra o resultado.

O desenvolvimento das interfaces possibilitou a customização da interação incorporando a identidade visual do dispositivo, contribuindo para o gerenciamento dos recursos.

O server é uma aplicação desenvolvida em nodeJS que faz a comunicação todos os envolvidos, ele é responsável por receber os dados dos sensores e salvar no banco de dados, ele faz a comunicação do WATSON mantendo a sessão do usuário e devolvendo as respostas conforme necessidade sendo elas diálogos ou conclusões, o acesso do TELEGRAM ao server é feito através de um túnel utilizando o Microsoft Azure que faz a ligação entre o TELEGRAM e o server (APÊNDICE A).

5. CONCLUSÃO

Internet das Coisas (IoT) é um sistema de dispositivos de computação que alia dispositivos mecânicos com digitais, possibilitando a transferência de dados mediante uma rede que não necessita de interação humano-humano ou humano-para-computador.

A definição da IoT evoluiu devido à convergência de várias tecnologias, análises em tempo real, aprendizado de máquina, sensores de mercadorias e sistemas embarcados. Os campos tradicionais de sistemas embarcados, redes de sensores sem fio, sistemas de controle, automação (incluindo automação residencial e predial) e outros contribuem para permitir a Internet das coisas.

Há uma série de preocupações sérias sobre os perigos no crescimento da IoT, especialmente nas áreas de privacidade e segurança, e, conseqüentemente, as iniciativas setoriais e governamentais para resolver essas preocupações começaram.

A partir da implementação do protótipo da plataforma de IoT, ligado a tecnologia IBM WATSON, foi possível observar que a mesma é capaz de melhorar a gestão do ambiente de produção. Esta plataforma permite o uso de interface que assegura aos usuários uma melhor visualização das operações e condições como temperatura, umidade, CO e luminosidade, captados através de sensores.

Como proposta de trabalhos futuros são indicados alguns pontos que podem ser evoluídos, testados e/ou implementados, sendo que destacam-se:

- Criação de uma interface para a Web, a fim de prover uma série de facilidades aos usuários, mesmo os que não possuem muito conhecimento em informática.
- Criação de uma *Application Programming Interface (API)* para padronizar a troca de informações/mensagens entre os dispositivos e a plataforma.
- Melhorias da IA. Sistema Especialista para apoiar o produtor de leite na análise de dados sobre seu rebanho e produção, no diagnóstico de problemas, e na sugestão de medidas corretivas.
- Adicionar câmera de vídeo. A capacidade de conectividade entre os “aparelhos inteligentes”, como as câmeras, computadores e smartphones.

REFERÊNCIAS

- [1] VERMESAN, O.; FRIESS, P. Internet of things: converging technologies for smart environments and integrated ecosystems. River Publishers, 2013. Disponível em: <https://www.academia.edu/11819209/Internet_of_Things_Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems>. Acesso em: 17 de jun. 2020.
- [2] LEE, I.; LEE, K. The internet of things (IoT): applications, investments and challenges for enterprises. Business Horizons, Elsevier Science Publishers B.V., v. 58, n.4, p. 431-440, 2015.
- [3] RESEARCH, A. The internet of things will drive wireless connected devices to 40.9 billion in 2020. 2014. Disponível em: <<https://www.abiresearch.com/press/the-internet-of-things-will-drive-wireless-connect>>. Acesso em: 24 nov. 2019.
- [4] TEÓFILO, R.B.; FREITAS, L.S. O uso da tecnologia da informação como ferramenta de gestão. 2007. Disponível em:<http://www.aedb.br/seget/artigos07/652_SEGET%20roro.pdf>. Acesso em: 10 out. 2019.
- [5] NUNES, L. M.; RUARO, F.L.M. EASYT- Sistema Integrado de Mobilidade Urbana. PUC. Porto Alegre, 2012. Disponível em:<http://revistaeletronicas.pucrs.br/ojs/index.php/graduacao/articleview/12419/8337>. Acesso em:10 out. 2019.
- [6] ROUSE, M. Internet das coisas (IoT). 2019. Consultado em 14 de agosto de 2019.
- [7] BROWN, E. Who needs the internet of things?. 2016. Disponível em: <https://www.linux.com/news/who-needs-internet-things/>. Acesso em 12 mar. 2020.
- [8] LUZ, M; HUMENHUK, L. microsoft virtual academy: a internet das coisas – fundamentos de IoT. 2015. In: PACHECO, T. M; ARAÚJO, M. A. P. Sistema de gerência centralizada de fechaduras microcontroladas. Disponível em: <<https://pdfs.semanticscholar.org/ed3e/f277a68ff5516d076ddb66ebb92b416c4931.pdf>>. Acesso em: 21 nov. 2019.
- [9] HENDRICKS, D. O problema com a Internet das coisas. Datastore de Londres. Autoridade da Grande Londres. 2015. Disponível em: <<https://data.london.gov.uk/blog/the-trouble-with-the-internet-of-things/>>.
- [10] CISCO INTERNET BUSINESS SOLUTIONS GROUP (IBSG). A Internet das Coisas: como a próxima evolução da Internet está mudando tudo. 2011. Disponível em: <https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iodt_ibsg_0411final.pdf> Acesso em: 18 de jun. 2020
- [11] WEISER, M. (1991). O computador para o século XXI. Americano científico. 265 (3): 94-104. Código de Bib: 1991SciAm.265c.94W. doi: 10.1038/scientificamerican0991-94. Arquivado no original (PDF) em 11 de março de 2015. Recuperado em 5 de novembro de 2014.
- [12] KASTURI, K. et al. A Review of Architecture and Applications for Internet of Things. 2016. Disponível em: <https://www.researchgate.net/publication/309391188_A_review_of_architecture_and_applications_for_IOT> Acesso em: 17 de jun. 2020
- [13] PONTIN, J. (29 de setembro de 2005). ETC: Seis Webs de Bill Joy. Revisão de tecnologia do MIT. Consultado em 17 de novembro de 2013.

- [14] RAJI, R.S. (1994). Redes inteligentes para controle. IEEE Spectrum. 31 (6): 49–55. doi: 10.1109 / 6.284793.
- [15] ASHTON, K. That ‘internet of things’ thing. RFID Journal, vol. 22, p. 97-114, 2009. Disponível em: < <http://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>>. Acesso em: 10 nov. 2019.
- [16] MAGRASSI, P. (2 de maio de 2002). Por que uma infra-estrutura universal de RFID seria uma coisa boa. Relatório de pesquisa da Gartner G00106518.
- [17] EVANS, D. A Internet das coisas: como a próxima evolução da Internet está mudando tudo. 2011. Disponível em: < https://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf>. Acesso em: 13 mai. 2020.
- [18] WONG, C.-Y.; WANG, L. Trajectories of science and technology and their co-evolution in BRICS: insights from publication and patent analysis. Journal of Informetrics, v. 9, p. 90-101, 2015.
- [19] ROSA, R. P. Dispositivos IoT aplicáveis à agricultura intensiva e os resultados já alcançados. 2017. Disponível em: <https://riuni.unisul.br/bitstream/handle/12345/4059/Artigo_Final_Rafael_Rosa_v3.pdf?sequence=1&isAllowed=y> Acesso em: 27 de jul. 2020
- [20] EMBRAPA. Internet das coisas pode ajudar a melhorar produtividade agrícola (2018) Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/31786119/internet-das-coisas-pode-ajudar-a-melhorar-produtividade-agricola>>. Acesso em: 27 de jul. 2020
- [21] COSTA, C. L.; OLIVEIRA, L.; MOTA, L. M. S. Internet das coisas (IOT): um estudo exploratório em agronegócios Internet of Things (IOT): na exploratory study in agribusiness (2018) Disponível em: <<https://www.ufrgs.br/cienagro/wp-content/uploads/2018/10/Internet-das-coisas-IOT-um-estudo-explorat%C3%B3rio-em-agroneg%C3%B3cios.docx-Cain%C3%A3-Lima-Costa.pdf>>. Acesso em: 27 de jul. 2020.
- [22] BROWNING, D. IoT started with a vending machine. 2018. Disponível em: < <https://www.machinedesign.com/automation-iiot/article/21836968/iot-started-with-a-vending-machine>>. Acesso em: 28 de jul. 2020.
- [23] RIBEIRO, K. L. Internet das coisas erradas sufoca a inovação. InformationWeek. 7 de julho de 2014. Recuperado em 10 de novembro de 2014.
- [24] PINTO, R. V. Digital sovereignty or digital colonialism? Sur: International Journal of Human Rights, number 27, 2018. Disponível em: <<http://sur.conectas.org/es/soberania-digital-o-colonialismo-digital/>>. Acesso em: 27 de julho de 2020.
- [25] SOSINSKY, B. Cloud computing bible. New Jersey: John Wiley and Sons, 2011. Disponível em: < <https://arpatapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf>>. Acesso em: 28 jul. 2020.
- [26] LINHARES, R. Integração de aplicações e serviços utilizando computação na nuvem com a plataforma Microsoft Windows Azure. Universidade Regional de Blumenau. Centro de Ciências Exatas e Naturais. Curso de Ciência da Computação, 2011.
- [27] KOMMALAPATI, H. Windows Azure platform for enterprises. MSDN Magazine, [S.l.], v. 25, n. 2, 2010. Disponível em: < <http://msdn.microsoft.com/en-us/magazine/ee309870.aspx>>. Acesso em: 16 mai. 2019.
- [28] ROINE, J. Negócios modernos com tecnologia Microsoft Azure. 2019. (PDF) (e-book para download). ShareGate.

- [29] CHAPPEL, D. Apresentando a plataforma de serviços Azure: uma breve análise do Windows Azure, Net Services, SQL Services e Live Ser5tvices. Califórnia, 2008. Disponível em: <https://docplayer.com.br/5196684-Apresentando-a-plataforma-de-servicos-azure.html#show_full_text>. Acesso em: 24 out. 2019.
- [30] LI, H. Introducing Windows Azure: learn Windows Azure to create the nextgeneration, cloud-based applications. New York: Apress, 2009.
- [31] MARTINS, P. G. M; RAMALHO, R. A. S; GRACIANO, H. L. S. Análise do uso do aplicativo telegram para o controle de processos de manutenção de aeronaves. Inf. Inf., Londrina, v. 25, n. 1, p. 171 – 188, jan./mar. 2020.
- [32] SILVA, A. M; MATTOS, R. IBM Watson como Ambiente para Desenvolvimento e Execução de um Chatbot – um estudo de caso aplicado ao processo de atendimento ao usuário. 2018. Disponível em: <https://www.researchgate.net/publication/329873825_IBM_Watson_como_Ambient_e_para_Desenvolvimento_e_Execucao_de_um_Chatbot_-_Um_Estudo_de_Caso_Aplicado_ao_Processo_de_Atendimento_ao_Usuario> Acesso em: 27 de jul. 2020
- [33] <https://www.raspberrypi.org/products/raspberry-pi-zero/>. Acesso em: 20 dez. 2019.
- [34] <https://www.sparkfun.com/products/15099>. Acesso em: 20 dez. 2019.
- [35] <https://www.sparkfun.com/products/9088>. Acesso em: 20 dez. 2019.
- [36] <https://www.sparkfun.com/products/9403>. Acesso em: 20 dez. 2019.
- [37] <https://www.sparkfun.com/products/10916>. Acesso em: 20 dez. 2019.

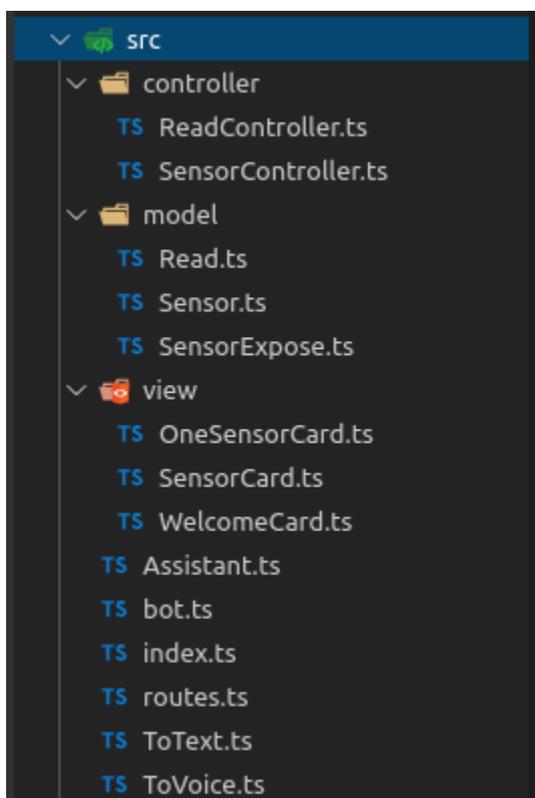
APÊNDICE A – APLICAÇÃO SERVER UTILIZADO NODE.JS

```
{
  "name": "milkbot",
  "version": "1.0.0",
  "description": "Milk Bot",
  "author": "Milk Bot",
  "license": "milkbot",
  "main": "./lib/index.js",
  "dependencies": {
    "adaptivecards": "^1.2.0",
    "adaptivecards-templating": "0.1.0-alpha.1",
    "await-sleep": "0.0.1",
    "botbuilder": "~4.5.3",
    "botbuilder-ai": "^4.5.3",
    "botbuilder-dialogs": "^4.5.3",
    "dotenv": "^7.0.0",
    "fs-extra": "^8.1.0",
    "ibm-watson": "^4.3.4",
    "mongoose": "^5.6.9",
    "mongoose-string-query": "^0.2.7",
    "mongoose-timestamp": "^0.6.0",
    "replace": "~1.1.0",
    "restify": "~8.2.0",
    "typegoose": "^5.9.0"
  },
  "devDependencies": {
    "@types/dotenv": "6.1.0",
    "@types/restify": "7.2.9",
    "nodemon": "^1.18.11",
    "tslint": "~5.14.0",
    "typescript": "~3.3.3"
  }
},
```

Package.json -> configuração da aplicação e suas dependências

→ Tscconfig -> configuração do typescript

→ Tslint -> configuração do verificador de código



Estrutura do código fonte

→ Controllers

ReadController	3
createRead	5
getAllLastRead	+9
getAllRead	3
getLastRead	+9
getPrecision	2
getReads	+9

ReadController -> responsável por operações com o documento leitura

SensorController	1
createSensor	1
getAllSensor	3

SensorController -> responsável por operações com documento do sensor

→ Model

Read	●
read	1
sensor	1

Read -> Estrutura do documento de leitura

Sensor	●
scale	1
sensorType	2
SensorType	●
LUZ	2
SOUND	
TEMPERATURE	
UMIDADE	1

Sensor -> estruturado documento de sensor

→ View

OneSensorCard	1
render	3
templatePayload	+9

OneSensorCard -> responsável pela plotagem de resposta do adaptive card de um sensor

∨	🔗 SensorCard	1
>	📦 render	9
	🔑 templatePayload	+9

SensorCard-> responsável pela plotagem de resposta do adaptive card resumido com todos os sensores

∨	🔗 Assistant	2
>	📦 closeChat	5
>	📦 sendMessage	+9
>	📦 sessionStart	+9
	🔑 assistantId	
	🔑 service	
	🔑 sessionId	1
	[📦] constructor	1

Assistant -> responsável pela comunicação com o serviço da IBM WATSON Assistant

∨	🔗 MyBot	●
>	📦 callFromAttachment	+9
>	📦 callFromBtn	3
>	📦 callFromText	1
	📦 getLastData	3
>	📦 getLastSensor	3
>	📦 sendReaders	3
>	📦 welcomeMessage	2
	🔑 assistant	
	🔑 conversationState	
	🔑 dialog	
	🔑 dialogState	
	🔑 toText	
	🔑 toVoice	2
	🔑 userState	
>	[📦] constructor	8
	[📦] fs	1

Bot -> responsável pela interação com o fluxo do assistant

>	server.listen() callback	2
>	server.post('/api/messages') ...	3
[@]	adapter	
[@]	conversationState	1
[@]	ENV_FILE	
[@]	memoryStorage	
[@]	mongoose	1
[@]	myBot	
[@]	server	
[@]	userState	1

Índex -> responsável pelo start da aplicação

∨	Route	1
∨	[@] constructor	4
>	server.get('/img/:name') ...	+9
>	server.get('/read') callback	8
>	server.post('/read') callback	8
>	server.post('/sensor') call...	5
[@]	errors	1
[@]	fs	1

Route -> responsável pelo mapeamento dos serviços REST para manutenção de sensor e leituras

∨	ToText	2
>	createDir	5
>	decodeVoice	1
>	downloadAttachmentAnd...	7
🔑	service	1
[@]	constructor	1
[@]	axios	1
[@]	fs	1

ToText -> responsável pela comunicação com a api Speech to text da IBM

∨	ToVoice	●
>	createDir	5
>	encodeVoice	+9
	synthesize	1
🔑	service	1
[@]	constructor	
[@]	fs	1
[@]	path	1



UPF

UNIVERSIDADE
DE PASSO FUNDO

UPF Campus I - BR 285, São José
Passo Fundo - RS - CEP: 99052-900
(54) 3316 7000 - www.upf.br