

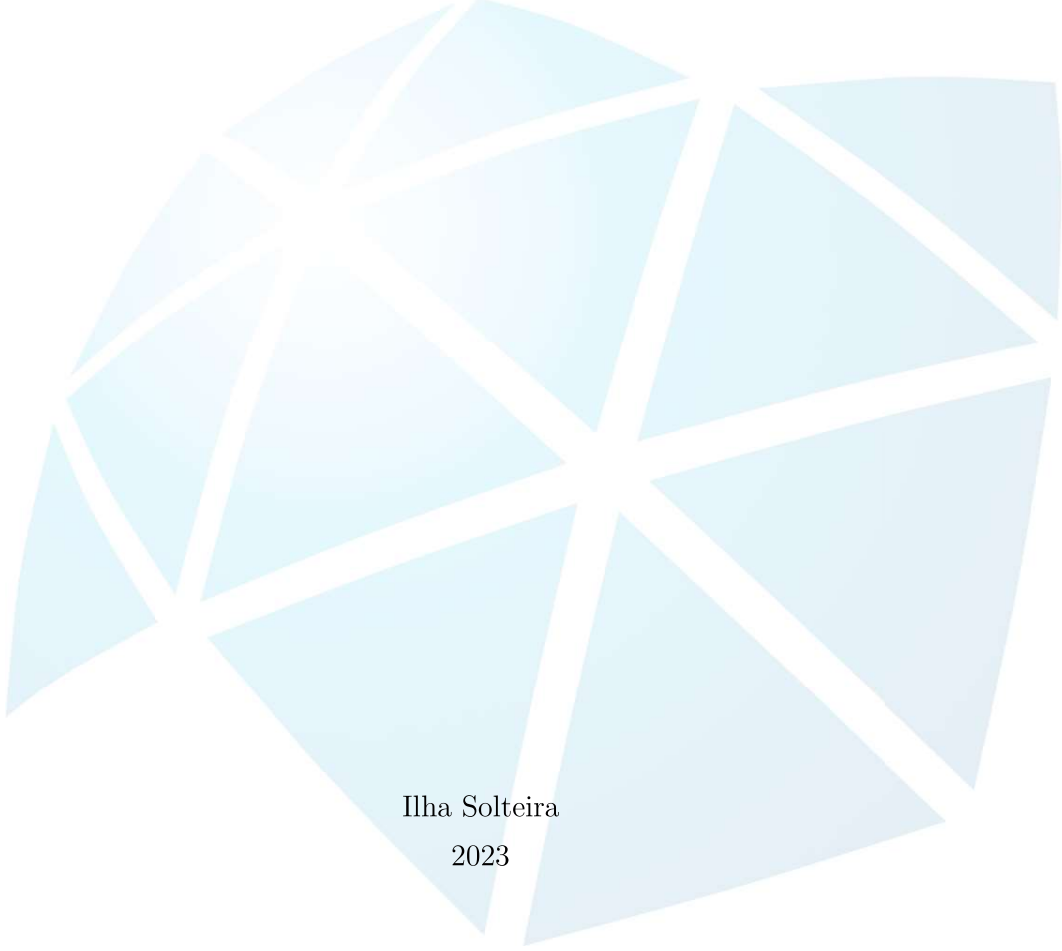
UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

FACULDADE DE ENGENHARIA

CÂMPUS DE ILHA SOLTEIRA

EDUARDO LOPES DA CRUZ

Sistema de monitoramento da qualidade do ar em ambientes internos implementado em
nuvem Azure



Ilha Solteira

2023

EDUARDO LOPES DA CRUZ

Sistema de monitoramento da qualidade do ar em ambientes internos implementado em
nuvem Azure

Dissertação apresentada à Faculdade de Engenharia – UNESP – Câmpus de Ilha Solteira como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica. Área de Conhecimento: Automação.

Prof. Dr. Alexandre César Rodrigues da Silva
Orientador

Ilha Solteira
2023

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

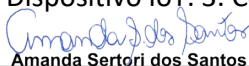
Cruz, Eduardo Lopes da.
C957s Sistema de monitoramento da qualidade do ar em ambientes internos
implementado em nuvem Azure / Eduardo Lopes da Cruz. -- Ilha Solteira: [s.n.],
2024
120 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de
Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2024

Orientador: Alexandre César Rodrigues da Silva

Inclui bibliografia

1. Qualidade do ar. 2. Dispositivo IoT. 3. Computação em nuvem.


Amanda Sertori dos Santos

Bibliotecária - CRB/8-9061
Seção Técnica de Referência, Atendimento ao
Usuário e Documentação
Diretoria Técnica de Biblioteca e Documentação

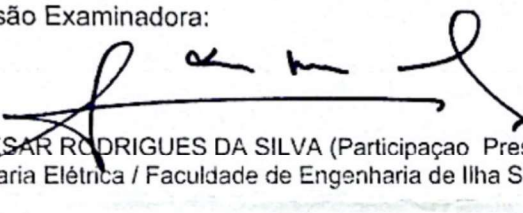
CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: SISTEMA DE MONITORAMENTO DA QUALIDADE DO AR EM AMBIENTES INTERNOS IMPLEMENTADO EM NUVEM AZURE

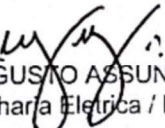
AUTOR: EDUARDO LOPES DA CRUZ

ORIENTADOR: ALEXANDRE CESAR RODRIGUES DA SILVA

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área:
Automação pela Comissão Examinadora:


Prof. Dr. ALEXANDRE CESAR RODRIGUES DA SILVA (Participação Presencial)
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Prof. Dr. MARCOS ANTONIO ESTREMOTE (Participação Presencial)
Departamento de Tecnologia em Análise e Desenvolvimento de Sistemas / Centro Universitário de Santa Fé do Sul - UNIFUNEC


Prof. Dr. MARCELO AUGUSTO ASSUNÇÃO SANCHES (Participação Presencial)
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Ilha Solteira, 16 de janeiro de 2024

RESUMO

Neste trabalho apresenta-se o desenvolvimento de uma solução IoT para medição das condições de qualidade do ar em ambientes internos utilizando serviços em nuvem para coleta, visualização e armazenamento dos dados do ambiente. O projeto utiliza-se do microcontrolador ESP8266 usado como dispositivo IoT para leitura dos dados coletados pelos sensores DHT22, MICS6814 e DSM591A, fazendo o envio dos dados para os serviços em nuvem fornecidos pela plataforma Microsoft Azure através da Internet via WiFi usando os protocolos TCP/IP e MQTT. Foi desenvolvido um ambiente em nuvem Azure para coleta, armazenamento e visualização dos dados utilizando os serviços em nuvem IoT Hub, Blob Storage e IoT Central, visando reduzir ao máximo o custo de implementação usando a maior quantidade de serviços em nuvem de forma gratuita. Para coleta de dados, o dispositivo IoT foi instalado em um laboratório de informática na Escola Técnica Estadual de Jales (São Paulo) entre 27 de julho e 18 de agosto, com a leitura de dados realizada as quintas e sextas-feiras, das 08 às 17h, em intervalos de 30 minutos a cada leitura. Os resultados obtidos foram satisfatórios visto que o dispositivo IoT e os serviços em nuvem atenderam as necessidades do projeto, os dados coletados podem ser vistos em forma de gráficos. Por fim, o uso de serviços em nuvem apresentaram custo baixo devido às configurações de utilização escolhidas para cada serviço e atenderam as necessidades de comunicação entre o dispositivo IoT, armazenamento e visualização dos dados através da Internet.

Palavras-chave: qualidade do ar. dispositivo IoT. computação em nuvem.

ABSTRACT

On this paper presents the development of an IoT solution for measuring air quality conditions in indoor environments using cloud services to collect, visualize and store environmental data. The project uses the ESP8266 microcontroller used as an IoT device to read the data collected by the DHT22, MICS6814 and DSM591A sensors, sending the data to Azure cloud services over the Internet via WiFi using the TCP/IP and MQTT protocols. An Azure cloud environment was developed to collect, store and visualize data using the IoT Hub, Blob Storage and IoT Central cloud services, aiming to reduce the implementation cost as much as possible by using the largest number of cloud services for free. For data collection, the IoT device was installed in a computer laboratory at the Jales State Technical School (São Paulo) between July 27th and August 18th, with data reading carried out on Thursdays and Fridays, from 8am to 5pm, at intervals of 30 minutes for each reading. The results obtained were satisfactory, as the IoT device and cloud services met the project's needs. The data collected can be seen like graphics. Finally, the use of cloud services presented a low-cost due to the chosen configurations and met the communication needs between the IoT device, storage and visualization of data through the Internet.

Keywords: air quality. iot device. cloud computing.

LISTA DE FIGURAS

Figura 1 – Arquitetura cliente-servidor convencional.	32
Figura 2 – Exemplo de uso da capacidade computacional e ociosidade de recursos.	32
Figura 3 – Virtualização Direto em <i>hardware</i> (a) e em SO hospedeiro (b).	33
Figura 4 – Processo de virtualização de <i>data centers</i> .	34
Figura 5 – Comparação entre virtualização (a) e containerização (b).	36
Figura 6 – Arquitetura do serviço cloud IaaS.	38
Figura 7 – Diferença entre os serviços disponíveis nas plataformas IaaS, PaaS e SaaS.	39
Figura 8 – Acesso à listagem detalhada de serviços gratuitos Azure.	42
Figura 9 – Acesso à listagem detalhada de serviços gratuitos Azure.	43
Figura 10 – Comunicação entre as extremidades do IoT Hub.	44
Figura 11 – Estrutura do Azure IoT Central.	46
Figura 12 – Estrutura de armazenamento Blob Storage.	47
Figura 13 – ESP8266 (a) e seu esquema de pinos (b).	49
Figura 14 – Multiplexador CD74HC4067 (a) e seu esquema de pinos (b).	51
Figura 15 – Sensor de gases MICS6814 (a) e seu esquema elétrico (b).	52
Figura 16 – Sensor de partículas DSM591A (a) e seu diagrama de blocos (b).	52
Figura 17 – Sensor de partículas DSM591A (a) e seu diagrama de blocos (b).	54
Figura 18 – Processo de <i>Handshake</i> do protocolo TCP.	56
Figura 19 – Colaboração entre os protocolos MQTT e TCP/IP.	58
Figura 20 – Funcionamento do MQTT.	59
Figura 21 – Camadas do projeto de medição da qualidade do ar.	63
Figura 22 – Estrutura do código-fonte gravado no ESP8266.	64
Figura 23 – Integração física dos componentes.	68
Figura 24 – Integração física dos componentes elaborado no <i>software</i> Fritzing.	68
Figura 25 – Integração entre o dispositivo IoT e os serviços em nuvem	69
Figura 26 – Configuração para criação de um novo IoT Hub.	70
Figura 27 – Validação das configurações e criação do IoT Hub.	71
Figura 28 – Criação de um novo dispositivo no IoT Hub.	72
Figura 29 – Validação das configurações e criação do IoT Hub.	72
Figura 30 – Configuração de conexão do dispositivo IoT e Iot Hub.	73

Figura 31 – Criação de um novo contêiner de armazenamento	74
Figura 32 – Acesso aos contêineres vinculados a uma conta de armazenamento.	75
Figura 33 – Configuração de rota do IoT Hub para contêiner do Blob Storage.	76
Figura 34 – continuação da configuração de rota do IoT Hub para contêiner do Blob Storage.	77
Figura 35 – Configuração final de rota do IoT Hub para contêiner do Blob Storage.	78
Figura 36 – Visualização das rotas do IoT Hub.	78
Figura 37 – Estrutura de armazenamento do serviço Blob Storage.	79
Figura 38 – Visualização dos arquivos de dados e suas propriedades.	80
Figura 39 – Dados presentes no arquivo de blobs.	80
Figura 40 – Página principal do IoT Central	82
Figura 41 – Configuração de novo aplicativo IoT Central	82
Figura 42 – Configuração de novo aplicativo IoT Central	83
Figura 43 – Comunicação entre as camadas e serviços do projeto.	83
Figura 44 – Criação de novo dispositivo no IoT Central.	84
Figura 45 – Área de criação de novos modelos de dispositivo IoT Central	85
Figura 46 – Configuração dos dados a serem exibidos.	85
Figura 47 – Configuração dos dados a serem exibidos.	86
Figura 48 – Dados para a conexão com IoT Central.	86
Figura 49 – Interface de exibição dos dados no IoT Central.	87
Figura 50 – Dados referentes a temperatura do ambiente.	88
Figura 51 – Dados referentes a umidade relativa do ar do ambiente.	89
Figura 52 – Dados referentes a monóxido de carbono(CO) do ambiente.	89
Figura 53 – Dados referentes a amônia (NH_3) do ambiente.	90
Figura 54 – Dados referentes ao dióxido de nitrogênio (NO_2) do ambiente.	90
Figura 55 – Dados referentes partículas em suspensão de até 10 m (PM10) do ambiente.	91
Figura 56 – Dados referentes a partículas de até 2,5 m (PM25) do ambiente.	92
Figura 57 – Código-fonte em C gravado no ESP8266 - Parte 01.	103
Figura 58 – Código-fonte em C gravado no ESP8266 - Parte 02.	104
Figura 59 – Código-fonte em C gravado no ESP8266 - Parte 03.	105
Figura 60 – Código-fonte em C gravado no ESP8266 - Parte 04.	106
Figura 61 – Código-fonte em C gravado no ESP8266 - Parte 05.	107
Figura 62 – Código-fonte em C gravado no ESP8266 - Parte 06.	108
Figura 63 – Código-fonte em C gravado no ESP8266 - Parte 07.	109
Figura 64 – Código-fonte em C gravado no ESP8266 - Parte 08.	110

LISTA DE QUADROS

Quadro 1 – Índices de qualidade do ar e seus principais efeitos	21
Quadro 2 – Principais poluentes atmosféricos e suas fontes de emissão	22
Quadro 3 – Principais poluentes do ar interno e suas principais fontes.	24
Quadro 4 – Poluentes e parâmetros legais vigentes no Brasil.	25
Quadro 5 – Características do microcontrolador ESP8266	49
Quadro 6 – Gases detectados pelo sensor MICS6814 e suas escalas	51
Quadro 7 – Parâmetros necessários para autenticação no serviço em nuvem.	64
Quadro 8 – Dados lidos em 27 de julho de 2023.	111
Quadro 9 – Dados lidos em 28 de julho de 2023.	112
Quadro 10 –Dados lidos em 03 de agosto de 2023.	112
Quadro 11 –Dados lidos em 04 de agosto de 2023.	113
Quadro 12 –Dados lidos em 10 de agosto de 2023.	113
Quadro 13 –Dados lidos em 11 de agosto de 2023.	114
Quadro 14 –Dados lidos em 17 de agosto de 2023.	114
Quadro 15 –Dados lidos em 18 de agosto de 2023.	115

LISTA DE ABREVIATURAS E SIGLAS

m	micrômetro
ACK	<i>Acknowledgement</i>
ADC	<i>Analog-to-Digital Converter</i>
API	<i>Application Program Interface</i>
BI	Business Intelligence
BSA	Basic Service Area
CETESB	Companhia Ambiental do Estado de São Paulo
CONAMA	Conselho Nacional do Meio Ambiente
COV	Compostos Orgânicos Voláteis
CPU	<i>Central Processor Unit</i>
CSV	<i>Comma-separated values</i>
DaaS	Dispositivo como Serviço
FPGA	<i>field-programmable gate array</i>
GB	gigabytes
HDL	<i>Hardware Description Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
I2C	<i>Inter-Integrated Circuit</i>
I2S	<i>Inter-IC Sound Bus</i>
IaaS	infraestrutura como serviço
IBM	<i>International Business Machines</i>

ID	Identificador
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
JSON	<i>JavaScript Object Notation</i>
KVM	<i>Kernel Virtual Machine</i>
Kwh	Kilowhatt hora
LAN	<i>Local Area Network</i>
LED	<i>Light Emitting Diode</i>
LoRaWAN	<i>Long Range Wide Area Network</i>
MAC	<i>Media Access Control</i>
MB	megabytes
MHz	megahertz
MP	Material Particulado
MQTT	<i>Message Queuing Telemetry Transport</i>
OMS	Organização Mundial de Saúde
PaaS	Plataforma como Serviço
PNMA	Política Nacional do Meio Ambiente
ppm	partículas por milhão
PWM	<i>Pulse With Modulation</i>
QoS	<i>Quality of Service</i>
QR	<i>Quick Response</i>
RISC	<i>Reduced Instruction Set Computer</i>

SaaS	Software como Serviço
SD	<i>Secure Digital</i>
SDK	<i>Software Development Kit</i>
SDK	emphSoftware Development Kit
SED	Síndrome do Edifício Doente
SO	Sistema Operacional
SPI	<i>Serial Peripheral Interface</i>
SYM	<i>Synchronize</i>
TCP/IP	<i>Transmission Control Protocol/ Internet Protocol</i>
TIC	Tecnologia da Informação e Comunicação
tWh	terawatt-hora
UTF-8	<i>8-bit Unicode Transformation Format</i>
VPN	<i>Virtual Private Network</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	JUSTIFICATIVA	15
1.2	OBJETIVO	15
1.3	TRABALHOS RELACIONADOS	16
1.4	ORGANIZAÇÃO DO TEXTO	18
2	ATMOSFERA E POLUENTES DO AR	20
2.1	PRINCIPAIS POLUENTES DO AR E SUAS FONTES EMISSORAS	22
2.2	POLUENTES DO AR INTERNO	23
2.2.1	Material Particulado	25
2.2.2	Compostos Orgânicos Voláteis	26
2.2.3	Compostos Inorgânicos Gasosos	26
2.2.4	Poluentes de Origem Biológica	28
2.2.5	Alergênicos Biológicos	29
2.2.6	Síndrome do Edifício Doente	29
2.3	PADRÕES DA ANVISA	29
3	SERVIDORES, VIRTUALIZAÇÃO E SERVIÇOS EM NUVEM	31
3.1	INFRAESTRUTURA DE SERVIDORES	31
3.2	VIRTUALIZAÇÃO	33
3.3	CONTÊINERES	35
3.4	PLATAFORMAS EM NUVEM PARA VIRTUALIZAÇÃO E <i>CONTAINERS</i>	37
4	FUNDAMENTOS TEÓRICOS E MATERIAIS	40
4.1	AMBIENTE DE INSTALAÇÃO DO DISPOSITIVO IOT	40
4.2	MICROSOFT AZURE	41
4.2.1	Azure IoT Hub	43
4.2.2	Azure IoT Central	45

4.2.3	Azure Blob Storage	47
4.3	DISPOSITIVOS FÍSICOS	48
4.3.1	Microcontrolador ESP8266	48
4.3.2	Multiplexador analógico-digital CD74HC4067	50
4.3.3	Sensor detector de gases MICS6814	50
4.3.4	Sensor de partículas DSM501A	52
4.3.5	Sensor de temperatura e umidade DHT22	53
4.4	CONEXÃO DE REDE WIFI E PROTOCOLOS DE COMUNICAÇÃO	54
4.4.1	Comunicação WiFi via Protocolos TCP/IP	54
4.4.2	Protocolo MQTT	56
4.4.3	SDK IoT Hub Client ESP8266 NodeMCU	59
4.4.4	Considerações finais sobre o capítulo	61
5	METODOLOGIA	62
5.1	IMPLEMENTAÇÃO FÍSICA	63
5.2	CONFIGURAÇÃO DOS SERVIÇOS EM NUVEM	69
5.2.1	Configuração IoT Hub	69
5.2.2	Serviço de armazenamento Blob Storage	73
5.2.3	Configuração IoT Central	81
5.3	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO	87
6	RESULTADOS	88
7	CONCLUSÃO	93
	REFERÊNCIAS	96
	APÊNDICE A – CÓDIGO FONTE GRAVADO NO MICROCONTROLADOR ESP8266	103
	APÊNDICE B – DADOS DE LEITURAS FEITAS PELO DISPOSITIVO IOT ARMAZENADOS NO SERVIÇO BLOB STORAGE	111

1 INTRODUÇÃO

A atmosfera terrestre levou bilhões de anos para se estabilizar, composta atualmente por nitrogênio, oxigênio e outros elementos essenciais a vida no planeta terra. As preocupações com a atmosfera terrestre apresentam dois panoramas principais: o aquecimento global e as questões relativas à poluição do ar, que impacta na saúde e no meio ambiente em geral (Ribeiro; Custodio; Pereira, 2020).

As políticas públicas relacionadas as questões ambientais surgiram com maior ênfase a partir da Convenção de Estocolmo de 1972, onde foram tratadas as questões relacionadas a poluição atmosférica, da água e do solo. No Brasil, estas políticas têm início a partir da lei 6.938 de 31 de agosto de 1981, denominada Lei da Política Nacional do Meio Ambiente (PNMA) (CETESB, 2022).

Partículas e gases poluentes aspirados causam danos ao aparelho respiratório e outros órgãos vitais, podendo até levar em alguns casos ao desenvolvimento de câncer. Segundo o Ministério da Saúde (2021), as alterações na qualidade do ar pelas emissões de fontes poluentes podem causar problemas de saúde que vão de dores torácicas, rinite alérgica, asma e em casos mais graves ao câncer de pulmão.

A qualidade do ar é avaliada em termos de concentrações de poluentes em índices não naturais ou que interfiram na atmosfera de um local ou região (Ribeiro; Custodio; Pereira, 2020). As medições de poluentes atmosféricos vêm cada vez mais sendo realizadas por equipamentos mais modernos, sendo utilizadas estações automáticas de monitoramento da qualidade do ar, determinando a concentração de poluentes em tempo real na atmosfera.

A Resolução CONAMA (Conselho Nacional do Meio Ambiente) 491/2018 propõe padrões de qualidade do ar baseado em padrões científicos definidos pela Organização Mundial de Saúde (OMS), referência para os níveis de concentração de cada poluente, que se ultrapassados podem causar riscos à saúde da população (CONAMA, 2020)

Ainda assim, mesmo com os programas de monitoramento de qualidade do ar que visam assegurar o controle dos níveis padrões que garantam a saúde da população, como os realizados pela CETESB (Companhia Ambiental do Estado de São Paulo) , existe a demanda por sistemas de medição de qualidade do ar em ambientes internos, públicos ou

privados, para que os usuários destes ambientes possam realizar um maior controle dos elementos ali presentes como temperatura, umidade, partículas em suspensão e outros agentes poluentes tóxicos ou biológicos.

A OMS (Organização Mundial de Saúde) alerta para a Síndrome do Edifício Doente (SED), que ocorre quando pessoas que ocupam um mesmo ambiente sofrem sintomas de doenças, ou se sentem mal, sem haver motivos aparentes para isso. Os sintomas se tornam mais fortes quando a pessoa está presente no edifício e tendem a diminuir quando se está fora dele (Perdrix *et al.*, 2005).

Tendo em vista que em grandes centros urbanos a população passa cerca de 80% do tempo em um ambiente interno (Hewitt; Ashworth; Mackenzie, 2019), se faz necessário atualmente analisar as condições de umidade, temperatura e qualidade do ar dos ambientes em que passamos a maioria do nosso tempo, visando a saúde e bem-estar dos indivíduos ali presentes e potencializando a produtividade.

Sistemas de medição da qualidade do ar em tempo real tem se tornado uma necessidade em ambientes de uso coletivo e até mesmo em residências, pois a pandemia provocada pela COVID-19 nos apresentou de maneira traumática os riscos à saúde que um ambiente em condições ruins relacionadas a qualidade do ar (mais precisamente relacionadas a partículas em suspensão) pode provocar a uma população.

A detecção dos valores dos elementos climáticos e de substâncias que estejam no ambiente pode fornecer informações valiosas para o monitoramento da qualidade do ar do ambiente, que afetem a saúde humana (Ribeiro; Custodio; Pereira, 2020).

Atualmente no mercado existem muitos sensores de partículas, gases, umidade e temperatura de baixo custo (Karaoghlanian *et al.*, 2022). O uso destes recursos no projeto de dispositivos de medição da qualidade do ar em ambientes internos é uma opção extremamente viável devido a fácil aquisição, baixo custo e facilidade de implementação.

Sensores de medição de temperatura, umidade, fumaça, poeira, partículas em suspensão e gases como metano e monóxido de carbono podem ser utilizados em conjunto com dispositivos microcontroladores para aferir as condições de clima e qualidade do ar de um ambiente interno e fornecer um diagnóstico para os usuários deste ambiente.

Aliado ao uso de sensores, tem-se a necessidade do acompanhamento em tempo real das condições de qualidade do ar de um ambiente, estando presente no ambiente ou até mesmo remotamente, através da comunicação dos dispositivos físicos de medição conectados com aplicativos *web* e serviços em nuvem de monitoramento e controle remoto.

A tecnologia em nuvem tem-se mostrado uma opção bastante viável para suprir a necessidade de execução de aplicações baseadas em plataformas *web* para monitoramento

e controle de projetos com sensores e atuadores devido a seu baixo custo de implementação, a não necessidade de aquisição de equipamentos adicionais como servidores e dispositivos de rede e principalmente a alta disponibilidade dos dados a partir de qualquer lugar, tornando o controle em tempo real de tais projetos ainda mais amplo (microsoft.com, 2022c).

Um sistema ou aplicativo executado em nuvem possui as seguintes características principais: não necessita de infraestrutura de servidores in loco ou de rede para ser executado e possui fácil disponibilização para seus usuários através da internet.

O provedor de serviço em nuvem fica encarregado de realizar toda a configuração para execução do sistema *web*, disponibilizando serviços de infraestrutura e rede sob demanda, ou seja, a quantidade de recursos que serão disponibilizados dependerá das necessidades da aplicação *web* ou no tipo de serviço contratado.

A maioria provedores de serviços em nuvem disponíveis no mercado atualmente possuem serviços que são totalmente gratuitos, outros serviços pagos no formato assinatura e a maioria dos recursos são cobrados sob demanda, pagando somente o que usar e pelo tempo que usar.

Existem inúmeros serviços em nuvem disponíveis atualmente no mercado, com diversas arquiteturas como IaaS (infraestrutura como serviço) , PaaS (Plataforma como Serviço) , SaaS (*Software* como Serviço) e mais atualmente a DaaS (Dispositivo como Serviço) (amazon.com, 2022).

Este projeto tem como proposta a criação de uma solução IoT utilizando o microcontrolador ESP8266, os sensores DHT22, MICS6814 e GP2Y1014AU0F para medição das condições da qualidade do ar em ambientes internos.

Em conjunto foi desenvolvido uma aplicação em nuvem na plataforma Microsoft Azure utilizando os serviços IoT Hub para comunicação entre o microcontrolador e a nuvem, para armazenamento dos dados coletados foi utilizado o serviço Blob Storage, para visualização dos dados através da Internet é utilizado o serviço Azure IoT Central e para geração de gráficos mais elaborados relacionados aos dados armazenados é utilizado o serviço Power BI.

Na seção 1.1 são apresentadas as justificativas relacionadas ao projeto, na seção 1.2 são descritos os objetivos e na seção 1.3 é apresentada uma revisão bibliográfica abordando diversos trabalhos relacionados ao tema do projeto.

1.1 JUSTIFICATIVA

Os avanços nos processos de fabricação de dispositivos microeletrônicos, microcontroladores, sensores e atuadores proporcionaram a redução nos custos de aquisição e a disponibilidade de inúmeros modelos com diferentes características e funcionamento, fazendo com que os projetos de sistemas com microcontroladores pudessem ser projetados com maior complexidade, variedade de elementos e capacidade de aquisição de dados.

Com isso, aumentou-se a capacidade de comunicação de tais projetos com outras tecnologias de *softwares* capazes de funcionar como interface de controle, visualização e manipulação dos dados coletados. Criar sistemas multiplataforma que podem ser acessados de qualquer dispositivo e em qualquer lugar utilizando plataformas em nuvem é uma necessidade atual, visto que a maioria das pessoas possui um telefone celular ou outro dispositivo eletrônico com acesso à Internet.

O desenvolvimento de um equipamento de medição das condições climáticas e de qualidade do ar em um ambiente interno em conjunto com um sistema em nuvem de monitoramento e controle destes dados facilita a visualização das condições do ambiente em tempo real, proporcionando a possibilidade de mudanças pontuais nestas condições para melhorar o bem-estar e preservar a saúde das pessoas que residem ou utilizam este ambiente.

1.2 OBJETIVO

O objetivo principal deste trabalho foi desenvolver um sistema IoT (*Internet of Things*) para medição dos parâmetros de qualidade do ar em ambientes internos relacionados a temperatura, umidade, partículas em suspensão, monóxido de carbono (CO_2), amônia (NH_3), e dióxido de nitrogênio (NO_2).

O gás (CO_2) foi adotado como parâmetro de medição por ser produzido pela queima de combustíveis, principalmente em veículos. Já o gás (NH_3) foi adotado devido à grande produção agrícola na região, pois é utilizado como fertilizante em diversas culturas. Por fim, o gás (NO_2) foi adotado na medição, pois é altamente tóxico e utilizado como parâmetro no controle da qualidade do ar.

Também foi desenvolvido uma solução em nuvem para leitura, armazenamento e monitoramento dos dados dos sensores, proporcionando visualização em tempo real das condições do ar do ambiente interno através da Internet e em qualquer dispositivo que suporte a execução de um navegador de internet ou instalação de aplicativo.

Outro objetivo que pode ser destacado é a utilização da maior quantidade de serviços

em nuvem possível para a implementação dos requisitos de *software*, visando minimizar o uso de infraestrutura física para armazenamento e processamento dos dados, ferramentas de software e linguagens de programação.

1.3 TRABALHOS RELACIONADOS

As preocupações relacionadas a medição da qualidade do ar em ambientes vêm aumentando ao longo dos anos. Embora no estado de São Paulo a CETESB possua uma rede de estações de monitoramento da qualidade do ar na região metropolitana, litoral e interior do estado, esta medição não é realizada em ambientes internos (CETESB, 2022).

A medição dos elementos presentes no ar de um ambiente interno é de extrema importância para preservar a saúde e bem-estar das pessoas que o frequentam. Um ambiente em boas condições climáticas e de qualidade do ar proporcionam melhor produtividade e preservam a saúde.

Existem muitos sensores de medição de gases disponíveis no mercado atualmente, cada um deles com parâmetros técnicos diferentes. Bauerová *et al.* (2020) realizaram a comparação entre diferentes tipos de sensores de medição de partículas e gases e destacam que nem o melhor tratamento de dados relacionados as medições podem ajudar no projeto de medição de qualidade do ar se os sensores utilizados forem inadequados ou possuírem má qualidade na medição.

A utilização de tecnologias baseadas em IoT (*Internet of Things*) vem sendo cada vez mais adotada na produção de dispositivos de medição de qualidade do ar e muitos trabalhos vem sendo desenvolvidos nesta área.

Wall *et al.* (2021) desenvolveram uma aplicação onde os dados do sensor BME680 são coletados utilizando um microprocessador ESP32 que enviam estes dados para um microcomputador Raspberry Pi, responsável por armazenar e exibir estes dados.

Ljubojevic *et al.* (2016) empregaram a qualidade do ar como um fator que contribuiu para a qualidade de vida humana geral. Sua solução de monitoramento consistia em sensores de temperatura, umidade e compostos orgânicos voláteis totais. Os dados do sensor foram processados por um Arduino Uno e armazenados localmente em um cartão SD (*Secure Digital*) para posterior análise.

Gabel, Koller and Hertig (2022) elaboraram um equipamento de monitoramento de baixo custo da exposição atmosférica para várias substâncias como ozônio, dióxido de carbono, material particulado entre outros. O dispositivo também faz medições e análises meteorológicas. Foi constatado que sensores eletroquímicos tem desempenho maior que sensores convencionais aplicados a locais urbanos.

Com a utilização de IoT composto por vários sensores em conjunto com algoritmos de aprendizado de máquina, Mumtaz *et al.* (2021) elaboraram um sistema de medição para qualidade do ar com capacidade de reportar as condições do ar interno em tempo real utilizando uma página *web*, além de gerar alertas de detecção de anomalias na qualidade do ar.

Liu *et al.* (2020) utilizou um conjunto de sensores e atuadores em conjunto com redes neurais para detectar anomalias no clima interno em prédios industriais. Foram utilizados dispositivos IoT e serviço em nuvem Azure para monitoramento remoto e controle dos dispositivos.

A aplicação de sensores incorporados a algum padrão de rede de comunicação de dados facilita o processo de aquisição dos dados remotamente, Postolache, Pereira and Girão (2009) apresentou uma rede baseada em WiFi para seu sistema de processamento de dados de qualidade do ar com base em redes neurais. No trabalho foi utilizado o protocolo TCP/IP (*Transmission Control Protocol/ Internet Protocol*) para realizar a comunicação entre o nó sensor e o computador responsável por realizar o processamento dos dados.

Outro trabalho a associar a medição de variações climáticas com projetos de automação foi o de Ribeiro, Custodio and Pereira (2020), onde criada uma rede do tipo *mesh* utilizando padrão de rede ZigBee e IEEE (*Institute of Electrical and Electronics Engineers*) 1451.5 para a coleta de dados ambientais por longos períodos. Aliado a rede *mesh* foram aplicadas técnicas de mineração de dados, com objetivo de avaliar as mudanças climáticas ocorridas durante um período.

Já Chen *et al.* (2014) recorreu a dispositivos móveis, conexão *bluetooth* e códigos QR (*Quick Response*) para disponibilizar dados sobre os níveis de CO_2 em ambientes internos. O foco principal do projeto foi facilitar a visualização dos dados para as pessoas através da leitura de código QR.

Outra tecnologia que vem sendo empregada na integração de sensores pode ser vista no trabalho de Cunha and Valin (2020), onde a proposta foi desenvolver uma rede de sensores capturando variações no clima como temperatura, luminosidade e pressão atmosférica utilizando tecnologia LoRaWAN (*Long Range Wide Area Network*) sem fio conectados a dois servidores, um responsável por pelo gerenciamento da rede e das informações enviadas pelos sensores e outro servidor de aplicação para executarem ações específicas conforme os dados recebidos do servidor de rede.

Serviços em nuvem também estão sendo utilizados para processar dados em tempo real através de aprendizagem de máquina. Wazwaz *et al.* (2023) em seu trabalho

desenvolveram uma aplicação utilizando serviços em nuvem Microsoft Azure como IoT Edge e Stream Analytics em conjunto com acelerômetros presentes em *smartphones* para reconhecimento de atividade humana. Três modelos de aprendizagem de máquina foram implementados e diferentes experimentos mostraram que o uso de serviço em nuvem melhoraram o desempenho da aplicação.

Peter *et al.* (2022) em seu trabalho empregaram tecnologias inteligentes de monitoramento e normalização de falhas em sistemas de distribuição usando uma das plataformas de serviços em nuvem mais populares, o Hub IoT do Microsoft Azure, juntamente com alguns dos serviços relacionados. Um protótipo de *hardware* foi construído com base em parte de uma rede real de sistema de distribuição subterrânea, e as técnicas de monitoramento e normalização de falhas foram integradas para formar um sistema. Tal sistema com integração IoT reduz efetivamente a queda de energia sofrida pelos clientes de aproximadamente 1 hora para menos de 5 minutos, sendo capaz de melhorar o índice de duração média de interrupção do sistema nas concessionárias de energia elétrica de forma significativa.

1.4 ORGANIZAÇÃO DO TEXTO

No capítulo 1 apresenta-se a introdução geral com a contextualização do trabalho, breve apresentação sobre atmosfera, clima e qualidade do ar, serviços em nuvem e uma revisão sobre trabalhos que abordam o mesmo tema com relevância na área.

No capítulo 2 apresentam-se as questões relacionadas a atmosfera terrestre, focando seus principais poluentes, as legislações vigentes relacionadas aos padrões de qualidade do ar e os problemas de saúde que podem ocorrer devido à poluição e má qualidade do ar.

No capítulo 3 apresentam-se os conceitos sobre infraestrutura de servidores, computação em nuvem com foco nos principais serviços disponíveis, sua estrutura e principais características.

No capítulo 4 são apresentados os principais serviços em nuvem da plataforma Microsoft Azure com foco nos serviços IoT Hub, Blob Storage e IoT Central. Também são apresentadas as características do protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*) usado para enviar os dados de telemetria dos sensores para a nuvem e TCP/IP utilizado para conexão a Internet via WiFi.

São abordadas as características dos dispositivos físicos do projeto como o microcontrolador ESP8266, multiplexador analógico-digital CD74HC4067 e sensores MICS6814, DHT22 e GP2Y1014AU0F, usados na camada física do projeto.

No capítulo 5 apresentam-se as características do desenvolvimento e implementação do

projeto, o método de comunicação com os serviços em nuvem e os métodos de aquisição, armazenamento, processamento e disponibilização dos dados coletados.

No capítulo 6 são apresentados os gráficos elaborados usando o *software* Power BI com os resultados obtidos referentes a coleta de dados realizada pelo dispositivo IoT e armazenado no serviço em nuvem.

Por fim, no capítulo 7 são apresentadas as conclusões sobre o projeto e os trabalhos futuros que podem ser desenvolvidos a partir dele.

2 ATMOSFERA E POLUENTES DO AR

A atmosfera é um dos elementos que compõem o meio ambiente e que possui um papel vital na vida do planeta: serve como escudo protetor contra radiação solar e é a principal fonte do ar que respiramos. No entanto, tem sofrido impactos em relação à ação humana e produz uma série de mudanças no planeta, sejam modificações climáticas ou qualidade do ar (Dean *et al.*, 2021).

Ribeiro, Custodio and Pereira (2020) destaca que a atmosfera terrestre, chamada comumente de ar, é composta por: nitrogênio (78%), oxigênio (20,8%), gás carbônico (0,9%) e o restante (0,3%) por outros gases como argônio, hélio, metano, dióxido de enxofre, vapor d'água e outros gases.

A respiração humana e de várias outras espécies de animais é feita através da troca de gases no sistema respiratório, absorvendo o oxigênio livre no ar e eliminando o gás carbônico do metabolismo do corpo. Ao aspirar o ar de um ambiente, além do oxigênio essencial a vida, também trazemos para dentro de nosso organismo as impurezas que ali presentes como partículas, gases e outros contaminantes de natureza biológica como vírus e bactérias (Ribeiro; Custodio; Pereira, 2020).

Ayode (1994) descreve a poluição atmosférica como quaisquer substâncias em um meio diferente do seu natural, a ponto de afetar este meio danosamente. Ou seja, o ar de um ambiente é poluído quando algumas substâncias não naturais a este ambiente são introduzidas, trazendo malefícios aos que nele se situam.

A resolução 03/90 do Conselho Nacional do Meio Ambiente (CONAMA (2020)) de 28 de junho de 1989 afirma que poluentes atmosféricos são qualquer forma de matéria ou energia com intensidade e em quantidade, concentração, tempo ou característica em desacordo com os níveis estabelecidos, e que tornem ou possam tornar o ar impróprio, nocivo ou ofensivo a saúde. Além disso, esta mesma resolução define dois padrões de qualidade do ar:

1. **Padrões Primários:** concentrações de poluentes que, se ultrapassarem os níveis padrões, poderão afetar a saúde da população;
2. **Padrões Secundários:** concentrações de poluentes abaixo das quais se prevê o

mínimo efeito adverso sobre o bem-estar da população, assim como o mínimo dano a fauna, flora e meio ambiente em geral.

O índice da qualidade do ar é uma ferramenta onde, para cada poluente, é calculado um índice, o qual é um valor adimensional. Para cada nível do índice o ar recebe uma qualificação e uma cor, sua classificação é determinada pelo maior índice (CETESB, 2022).

Esta qualificação do ar está vinculada a norma legal (Resolução CONAMA n° 491/2018), independente do padrão de qualidade/meta intermediária em vigor. No Quadro 1 apresenta-se os índices e seus efeitos a saúde.

Quadro 1 – Índices de qualidade do ar e seus principais efeitos

Qualidade	Índice	Efeito
N1-Boa	0-40	-
N2-Moderada	41-80	Pessoas de grupos sensíveis (crianças, idosos, pessoas com doenças respiratórias e cardíacas) podem apresentar sintomas como tosse seca e cansaço. A população em geral não é afetada.
N3-Ruim	81-120	Toda a população pode apresentar sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta. Pessoas de grupos sensíveis (crianças, idosos, pessoas com doenças respiratórias e cardíacas) podem apresentar efeitos mais sérios na saúde.
N4-Muito Ruim	121-200	Toda a população pode apresentar agravamento dos sintomas como tosse seca, cansaço, ardor nos olhos, nariz e garganta e ainda falta de ar e respiração ofegante. Efeitos ainda mais graves à saúde de grupos sensíveis (crianças, idosos e pessoas com doenças respiratórias e cardíacas).
N5-Péssima	> 200	Toda a população pode apresentar sérios riscos de manifestações de doenças respiratórias e cardiovasculares. Aumento de mortes prematuras em pessoas de grupos sensíveis.

Fonte: Adaptado de (CETESB, 2022).

O problema da qualidade do ar é determinado inicialmente pelas fontes emissoras e potencializada pela atmosfera, ainda que a atmosfera possua capacidade difusora de ar, proporcionando uma “autolimpeza”. Fatores determinantes a má qualidade do ar são a presença intensa de fontes emissoras, sendo elas naturais (vulcões, pólen de plantas), fixas (indústrias, mineração, queima de lixo), móveis (veículos a base de combustíveis fósseis ou etanol) e químicas (Castelhano, 2021).

2.1 PRINCIPAIS POLUENTES DO AR E SUAS FONTES EMISSORAS

Os riscos ligados a qualidade do ar são divididos em 3 grupos: estéticos, tóxicos e irritantes. Os efeitos estéticos são aqueles danos causados nas estruturas físicas das construções em virtude dos poluentes em suspensão na atmosfera como desgaste, sujeira ou perda de cor.

Os efeitos irritantes causam danos à saúde em um grau de risco menor como ardência e incômodo nos olhos e narinas. Por fim, os efeitos tóxicos trazem severas consequências a saúde, ocasionando problemas no sistema respiratório, cardiovascular e nervoso, dependendo do poluente e seu nível de concentração (Castelhano, 2021). No Quadro 2 apresentam-se os principais poluentes atmosféricos e suas respectivas fontes de emissão.

Quadro 2 – Principais poluentes atmosféricos e suas fontes de emissão

Classificação	Tipo	Poluentes
Fontes fixas	Combustão	Material particulado, dióxido de enxofre e trióxido de enxofre. Monóxido de carbono, hidrocarbonetos e óxidos de nitrogênio.
	Processo industrial	Material particulado (fumos, poeiras e névoas), gases: SO_2 , SO_3 , ácido clorídrico (HCl) e hidrocarbonetos, Mercaptanos, ácido fluorídrico (HF), H_2S , NOx, Pb
	Queima de resíduos sólidos	Material particulado. Gases: SO_2 , SO_3 , HCl, NOx
	Outros	Hidrocarbonetos, material particulado.
Fontes móveis	Veículos automotores	Material particulado, monóxido de carbono, óxidos de nitrogênio, hidrocarbonetos, aldeídos e óxidos de enxofre
	Aviões e barcos	Óxidos de enxofre e óxidos de nitrogênio
	Locomotivas, etc.	Ácidos orgânicos, hidrocarbonetos e aldeídos
Fontes naturais		Material particulado — poeiras, gases — SO_2 , SO_3 , HCl, NOx, hidrocarbonetos
Reações químicas		Poluentes secundários — O_3 , aldeídos, ácidos orgânicos, nitratos orgânicos, aerossol fotoquímico, etc.

Fonte: Adaptado de Biblioteca Virtual em Saúde do Ministério da Saúde (Saúde, 2021).

As consequências causadas pelos efeitos tóxicos de poluentes presentes no ar na maioria das vezes não são imediatas, de modo que o organismo demora certo tempo até apresentar estes efeitos. Partículas e gases poluentes aspirados podem causar danos ao aparelho respiratório e outros órgãos, apresentando potencial cancerígeno.

Hewitt, Ashworth and Mackenzie (2019) afirmaram que a poluição do ar pode ter

causado até 2019 aproximadamente 3 milhões de mortes prematuras por ano, o dobro do número de mortes causadas por acidentes de trânsito.

Como apresentado no Quadro 2, a maioria das fontes de poluentes são provenientes da queima de resíduos ou de combustíveis fósseis e dos processos industriais. Com os avanços tecnológicos dos processos de produção e aumento do consumo, cada vez mais elementos poluentes são inseridos ao ar que respiramos, diminuindo a qualidade do ar atmosférico e principalmente o ar que inalamos em ambientes internos.

2.2 POLUENTES DO AR INTERNO

Os indivíduos que residem em centros urbanos passam cerca de 80% do tempo em ambientes internos, e a qualidade do ar em ambientes internos tem relação direta com a saúde e conforto das pessoas. Aliado a isso tem-se o aumento do uso de sistemas de ar-condicionado que diminuem a circulação do ar externo, podendo gerar um acúmulo de poluentes presentes no ambiente interno.

As características do ar interno estão diretamente relacionadas a qualidade do ar externo, mas também podem ser modificadas pelas ações realizadas no próprio ambiente interno como cozinhar, fumar, atividades com uso de componentes químicos, mobília ou materiais utilizados na construção (Hewitt; Ashworth; Mackenzie, 2019).

Inúmeros fatores podem contribuir como contaminantes do ar, os principais elementos físico-químicos que afetam a qualidade do ar interno são apresentados no Quadro 3, destacando os materiais particulados (MP) , compostos orgânicos voláteis (COV), compostos inorgânicos gasosos e fumaça de cigarros. Além disso, estes elementos também podem ser de natureza biológica como fungos, vírus e bactérias (Jones, 1999).

Segundo dados da OMS, apenas 10% da população mundial está exposta a níveis médios anuais de MP10 (diâmetro aerodinâmico menor que 10 μ) ou MP 2,5 (diâmetro aerodinâmico menor que 2,5 μ) que atendem as recomendações mínimas estabelecidas.

Na legislação brasileira, que teve sua última atualização em 2018 a partir da Resolução n.º 491/2018 do CONAMA, são considerados seis compostos que, se introduzidos ao meio, podem ser considerados poluentes: monóxido de carbono (CO), fumaça, Dióxido de Nitrogênio (NO_2), ozônio troposférico (O_3), Material Particulado MP10 e MP 2,5, dióxido de enxofre (SO_2) e Chumbo (PB) (CONAMA, 2020).

Silva *et al.* (2013) ainda destaca que pelo menos 22% dos pacientes admitidos em unidades de pronto-atendimento na cidade de Porto Alegre com queixas respiratórias por problemas virais entre novembro de 2008 e dezembro de 2010 tem correlações entre variáveis meteorológicas e poluição do ar, e que este número poderia aumentar

Quadro 3 – Principais poluentes do ar interno e suas principais fontes.

	Poluentes	Principais fontes
Poluentes de origem não biológica	Compostos Orgânicos Voláteis(COV)	Adesivos, tintas, solventes, materiais de construção, combustão, fumaça de tabaco.
	Dióxido de carbono(CO_2)	Atividade metabólica, combustão, motores veiculares em garagens.
	Monóxido de carbono(CO)	Ar extremo, queima de combustíveis, motores veiculares (garagens).
	Dióxido de enxofre(SO_2)	Queima de combustíveis, aquecedores de água, fornos, fogões, aquecedores a gás ou a querosene, fumaça e tabaco.
	Óxido de nitrogênio (NO)	Ar extremo, queima de combustíveis, motores veiculares (garagens).
	Dióxido de nitrogênio (NO_2)	Ar extremo, queima de combustíveis, motores veiculares (garagens).
	Formaldeídos(H_2CO)	Materiais de isolamento, móveis, madeira compensada.
	Hidrocarbonetos policíclicos aromáticos (HPA)	Queima de combustíveis, fumaça de cigarro.
	Ozônio (O_3)	Reações fotoquímicas, campos eletrostáticos (equipamentos eletrônicos)
	Radônio	Solo, materiais de construção (pedras, concreto).
	Material particulado	Re-suspensão, fumaça de tabaco, combustão.
	Fibra de asbesto ou amianto	insulação, materiais anti-chama
	Calor	Metabolismo humano, sistema de ar condicionado, cozinha.
Origem biológica	Alergênicos	Poeira, animais domésticos, insetos.
	Pólen	Plantas de exterior e de interior.
	Microrganismos (fungos, bactérias, vírus)	Pessoas, animais, plantas e vasos, sistemas de ar condicionado.
	Esporos de fungos	Solos, plantas, alimentos, superfícies internal.

Fonte: Adaptado de (Jones, 1999).

significativamente nas décadas seguintes.

Os padrões nacionais de qualidade são divididos em duas categorias:

1. Padrões de qualidade do ar intermediários – PI: valores estabelecidos como valores temporários a serem cumpridos em etapas;
2. Padrão de qualidade do ar final – PF: valores guia definidos pela OMS em 2005.

No Quadro 4 apresentam-se os parâmetros legais para poluentes vigentes no Brasil segundo o Conselho Nacional do Meio Ambiente (CONAMA, 2020).

- 1- Média aritmética anual;
- 2- Média horária;
- 3- Máxima média móvel obtida no dia;

Quadro 4 – Poluentes e parâmetros legais vigentes no Brasil.

Poluente	Período	PI-1 ($\mu\text{g}/\text{m}^3$)	PI-2 ($\mu\text{g}/\text{m}^3$)	PI-3 ($\mu\text{g}/\text{m}^3$)	PF ($\mu\text{g}/\text{m}^3$)	ppm
MP10	24h	120	100	75	50	
	Anual ¹	40	35	30	20	
MP2.5	24h	60	50	37	25	
	Anual ¹	20	17	15	10	
SO2	24h	125	50	30	20	
	Anual ¹	40	30	20	-	
NO2	1 Hora ²	260	240	220	200	
	Anual ¹	60	50	45	40	
O3	8h ³	140	130	120	100	
Fumaça	24h	120	100	75	50	
	Anual ¹	40	35	30	20	
CO	8h ³	-	-	-	-	9
PTS	24h	-	-	-	240	
	Anual ⁴	-	-	-	080	
Chumbo	Anual ¹	-	-	-	0.5	

Fonte: Adaptado de (CONAMA, 2020).

4- Média geométrica anual.

No Brasil, entre 2015 e 2019, o número de internações hospitalares por doenças respiratórias na população foi de 5.926.687, segundo as informações do Sistema Único de Saúde (SUS, 2022), com uma média de 282,0/100 mil habitantes. Em 2019, as internações por doenças respiratórias aumentaram 2,4% entre os beneficiários de planos de saúde. Problemas respiratórios levaram a 506,9 mil internações, representando 5,9% das causas (ANS, 2022).

De acordo com Kelvin and Halperin (2020) bebês e crianças geralmente possuem alto risco de infecções por vírus no trato respiratório devido a sua imaturidade do sistema imunológico. Enquanto Dias *et al.* (2020) justifica os altos percentuais dos idosos que contraem doenças respiratórias por conta da diminuição das funções pulmonares e devido à idade avançada, consequentemente surgem as comorbidades, acentuando o risco de um avanço mais grave das doenças respiratórias.

Nas subseções 2.2.1 a 2.2.5 encontram-se o detalhamento dos principais contaminantes físico-químicos que são prejudiciais à saúde demonstrado no Quadro 4.

2.2.1 Material Particulado

O material particulado é composto de uma mistura física e química de fumaça, poeira e outros tipos de materiais sólidos e líquidos suspensos na atmosfera. As principais fontes de emissão de material particulado na atmosfera são veículos automotores, processos industriais, queima de biomassa e ressuspensão de poeira do solo.

A CETESB (2022) classifica os materiais particulados em 3 categorias:

1. **Partículas totais em suspensão (PTS):** aquelas partículas cujo diâmetro aerodinâmico é menor que 50 μ . Uma parte destas partículas inaláveis pode causar problemas a saúde da população ou afetar sua qualidade de vida;
2. **Fumaça (FMC):** material particulado suspenso na atmosfera proveniente do processo de combustão, também chamado comumente de fuligem;
3. **Partículas inaláveis (MP10 e MP2,5):** aquelas partículas cujo diâmetro aerodinâmico é menor que 10 μ m, dividida em duas subcategorias: partículas inaláveis finas com diâmetro aerodinâmico inferior a 2,5 μ m e partículas inaláveis grossas com diâmetro entre 2,5 e 10 μ m. As partículas finas podem atingir os pulmões enquanto as partículas grossas ficam retidas na parte superior do sistema respiratório (órgãos fora da caixa torácica).

2.2.2 Compostos Orgânicos Voláteis

É definido como qualquer composto que participa de reações fotoquímicas ou que possui reatividade fotoquímica. Estes compostos têm uma definição menos rigorosa, pois são considerados compostos orgânicos que se encontram em estado gasoso ou em vapor (Ribeiro; Custodio; Pereira, 2020).

Uma parte dos COV (Compostos Orgânicos Voláteis) presentes em ambientes internos vem do meio externo, pois a queima de combustíveis fósseis ocasiona em altas produções destes compostos. Fontes internas como materiais de construção, acabamento, mobília, produtos de limpeza e inseticidas também colaboram com o aumento em ambientes internos. Vale ressaltar que a cocção de alimentos a partir das cozinhas também colabora para a deterioração da qualidade do ar do ambiente e produz altas concentrações de COV, mesmo que temporariamente.

Apesar das concentrações de cada COV encontrado em ambientes internos sejam normalmente inferiores aos limites de tolerância, podem causar irritações nos olhos, nas vias respiratórias e na pele, diminuição da capacidade respiratória e reações alérgicas. Em concentrações muito elevadas, elementos como benzeno podem ser tóxicos aos rins e fígado, distúrbios gastrointestinais e agir como elemento carcinogênico (CETESB, 2022)).

2.2.3 Compostos Inorgânicos Gasosos

Compostos inorgânicos são substâncias formadas por átomos ou moléculas de no mínimo dois elementos diferentes, e que não contenham em sua estrutura átomos de

carbono, formando cadeias ligados ao hidrogênio. Dentre os compostos inorgânicos os mais comuns que devem ser levados em relação a qualidade do ar são:

1. **Dióxido de carbono (CO_2):** é expelido naturalmente através da respiração humana, na combustão em veículos automotores, sendo incolor e inodoro, tendo sua concentração em ambientes internos variando entre 700 e 2.000ppm (partes por milhão) (Moura, 2018). O (CO_2) é um asfixiante e irritante do sistema respiratório e pode causar danos à saúde humana se encontrado em altas concentrações no ambiente (acima de 30.000ppm). Em concentrações moderadas pode causar sensação de desconforto e sensação de ambiente “abafado”;
2. **Monóxido de carbono (CO):** é um gás incolor, inodoro e com toxicidade considerável, formado através da combustão de materiais que contenham carbono em locais com baixo nível de oxigênio (Moura, 2018). Ao ser absorvido pelo organismo humano pode causar a diminuição dos níveis de oxigênio no sangue, causando efeitos mais tóxicos no coração e cérebro;
3. **O monóxido de nitrogênio (NO) e dióxido de nitrogênio (NO_2):** são formados em ambientes de altas temperaturas e pressões, onde o NO tem sua formação mais frequente na queima de combustíveis em motores veiculares, enquanto a produção de NO_2 está ligado a equipamentos de queima de gás, querosene, madeira e na fumaça de tabaco (Moura, 2018). Pode interferir no transporte de oxigênio para os tecidos musculares, produzindo efeitos parecidos com os causados pelo CO, podendo provocar inflamações respiratórias, edema ou enfisema pulmonar;
4. **Dióxido de enxofre (SO_2):** é formado através da queima de combustíveis fósseis que contenham impurezas ou compostos a base de enxofre, altamente solúvel em água, formando o ácido sulfúrico e sulfuroso (Moura, 2018). Sua inalação em doses elevadas causa danos ao sistema respiratório inferior e exposições crônicas podem levar a diminuição da função pulmonar, além de deteriorar equipamentos e móveis, pois os compostos formados quando em contato com a água são altamente corrosivos;
5. **Ozônio (O_3):** é um composto altamente reativo, formado facilmente quando o ar passa por um campo eletrostático. Naturalmente a fonte mais importante de ozônio é o ar externo, porém, ele pode ser produzido em ambientes internos através de equipamentos eletrônicos como copiadoras e impressoras a laser (Moura, 2018). É um agente oxidante que pode causar forte irritação nos pulmões, garganta e olhos.

2.2.4 Poluentes de Origem Biológica

A pandemia da COVID-19 apresentou de forma inegável a necessidade de preocupação a riscos biológicos, pois as gotículas expiradas por pessoas contaminadas podem permanecer no ar por um longo período, dependendo das condições do ambiente como umidade, temperatura, se está aberto ou fechado (Ribeiro; Custodio; Pereira, 2020).

Os microorganismos presentes no ar (principalmente interno) como fungos, bactérias, vírus entre outros podem causar irritações, alergias, doenças e outros efeitos tóxicos. O ser humano é contaminado por via aérea quando o agente microbiano é inalado e retido no trato respiratório em local propício para seu desenvolvimento. Fatores como a taxa de imunidade do indivíduo, dimensão das partículas, profundidade de penetração e dosagem mínima do agente capaz de provocar a doença são determinantes em relação a infectividade.

Como os principais efeitos da qualidade inadequada do ar em ambientes internos e externos são causados no sistema respiratório, as doenças respiratórias são aquelas que apresentam maior debilitação progressiva da saúde. Os itens abaixo apresentam alguns contaminantes microbiológicos que podem ser prejudiciais à saúde:

1. **Vírus:** as infecções respiratórias virais são as doenças mais comuns que afetam os seres humanos, sendo causa elevada de mortalidade e diminuição da produtividade e qualidade de vida. Os vírus se propagam pelas correntes de ar, ressuspensão de material particulado ou em gotículas dispersadas pela saliva. Alguns dos vírus mais comuns encontrados em ambientes são o *Rhinovirus* conhecido como resfriado comum, *Influenza* que causa infecção aguda no sistema respiratório e o *Adenovirus* que causa infecções nas vias aéreas superiores como faringite e amigdalite (Silva *et al.*, 2013);
2. **Bactérias:** inúmeras bactérias podem ser veiculadas pelo ar de um ambiente, que podem ser causadoras de infecções e respiratórias, pneumonia, febre, dores de cabeça, fraqueza, dores musculares, irritações na pele, faringite, laringite, infecção pulmonar entre outras doenças;
3. **Fungos:** qualquer infecção de origem fúngica é geralmente chamada de micose, podendo ser superficial, subcutânea ou sistêmica. Ocorrem geralmente quando um patógeno deste tipo encontra um hospedeiro debilitado, facilitando sua proliferação (USP, 2022). Alguns tipos de fungos podem causar doenças como infecções orais ou pulmonares, tuberculose e pneumonia.

2.2.5 Alergênicos Biológicos

Uma alergia é considerada uma resposta exagerada do sistema imunológico a um elemento estranho ao organismo, ou seja, uma hipersensibilidade a um estímulo externo específico. Esta hipersensibilidade é uma resposta antigênica mais intensa do que a considerada normal (VIGGIANO, 2022).

Há quatro tipos de reações de hipersensibilidade, a reação anafilática é considerada a mais comum relacionada a qualidade do ar, incluindo condições alérgicas comuns a asma. Ácaros, pólen, pelos e poeira de origem biológica são os principais alergênicos encontrados em ambientes internos. É importante ressaltar que grande parte da população sofre de alguma doença alérgica, tendo a asma como a doença que está se tornando a mais comum.

2.2.6 Síndrome do Edifício Doente

Síndrome do edifício doente é um termo criado na década de 70, a partir da climatização de ambientes fechados, selados do meio externo, tendo seus usuários apontando condições negativas relacionadas a qualidade do ar interno. É uma condição em que um conjunto inespecífico de sintomas relacionados a qualidade do ar, afetando um percentual considerável de ocupantes do ambiente, e que tendem a desaparecer ou diminuir quando as pessoas que apresentaram os sintomas deixam o local (Perdrix *et al.*, 2005)).

A síndrome do edifício doente ocorre quando pessoas que ocupam um mesmo ambiente sofrem sintomas de doenças, ou se sentem mal, sem haver motivos aparentes para isso. Os sintomas se tornam mais fortes quando a pessoa está presente no edifício e tendem a diminuir quando se está fora dele (Perdrix *et al.*, 2005).

Existem um conjunto de fatores que podem ser a causa da síndrome do edifício doente, como má distribuição do ar, insuficiência da renovação do ar interior, controle deficiente de temperatura e manutenção inadequada no sistema de climatização.

Os principais fatores relacionados a esta síndrome podem ser a poeira, contaminantes biológicos, contaminantes químicos, fumaça e contaminantes gerados pelo metabolismo humano.

2.3 PADRÕES DA ANVISA

A resolução RE nº 9, de 16 de janeiro de 2003, da ANVISA (2022) estabelece padrões de referência para qualidade do ar interior, em ambientes climatizados artificialmente, de

uso público e coletivo.

Na resolução estão listados os valores máximos recomendados (VMR) para os seguintes parâmetros: contaminação microbiológica, dióxido de carbono, aerodispersóides (MP), além de parâmetros de temperatura, umidade, velocidade, taxa de renovação e grau de pureza.

Um ambiente é considerado aceitável quando livre de contaminantes que apresentem riscos à saúde dos ocupantes ou que apresente um mínimo de 80% dos ocupantes deste ambiente sem queixas ou sintomas de desconforto (ANVISA, 2022).

O VMR para contaminação microbiológica deve ser igual a $750 \text{ ufc}/m^3$ (unidade de formação de colônias por metro cúbico) de fungos, para a relação I/E (Interno / Externo) igual 1,5, onde I representa a quantidade de fungos interior e quantidade de fungos no ambiente exterior. Quando o VMR for ultrapassado ou a relação I/E for maior que 1,5, é necessário fazer um diagnóstico relacionado as fontes poluentes para uma desinfecção ou intervenção corretiva, sendo inaceitável a presença de patogênicos ou toxigênicos (ANVISA, 2022).

Para a contaminação química os VMR são de 1000 ppm para dióxido de carbono e $80 \mu\text{g}/m_3$ para aerodispersóides no ar, como indicador de pureza e limpeza do ambiente climatizado.

As faixas recomendáveis para temperatura de Bulbo Seco (medida com termômetro comum) nas condições internas de verão devem variar entre 23 e 26°C, enquanto para condições de inverno a faixa recomendável deve variar entre 20 e 22°C. A seleção da faixa depende da finalidade do local onde o equipamento climatizador está instalado (ANVISA, 2022).

Em relação à umidade relativa do ar, nas condições internas para verão deverá variar entre 40 e 65%, a exceção de áreas de acesso que poderão chegar até 70%. Para condições internas de inverno a faixa recomendável deve variar de 35 a 65% (ANVISA, 2022).

3 SERVIDORES, VIRTUALIZAÇÃO E SERVIÇOS EM NUVEM

Neste capítulo descrevem-se os conceitos básicos fundamentais sobre infraestrutura de servidores, virtualização e containerização, abordando os aspectos sobre estes conceitos aplicados a serviços em nuvem.

3.1 INFRAESTRUTURA DE SERVIDORES

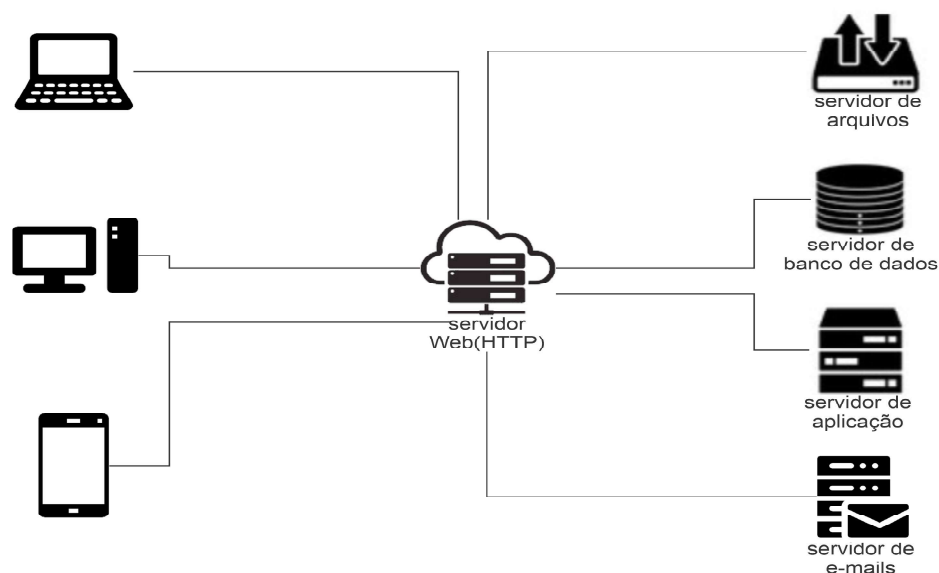
Durante as décadas de 1960 e 1970, os *mainframes*, computadores de grande porte que eram acessados por terminais ”burros”predominavam entre as instituições (Morimoto, 2010). A partir da década de 1980 os computadores de mesa substituíram os *mainframes* e com a popularização das redes e da Internet passou-se a depender cada vez mais de sistemas e serviços em rede.

Surgiu a necessidade de se criar páginas *web* e sistemas distribuídos, onde as aplicações destes sistemas, bem como os dados neles armazenados e manipulados estivessem disponíveis a todo momento. Para suprir tal necessidade utiliza-se até hoje a tecnologia de servidores, computadores dedicados exclusivamente para executar serviços e processos em rede como execução de serviço de e-mails, configuração e distribuição de conexões em diversas redes, disponibilização de dados, hospedagem de páginas e sistemas *web*, entre outros (VMware.com, 2022).

Os servidores são, em sua maioria, computadores com alta capacidade de processamento e armazenamento de dados, dedicados exclusivamente a executar um serviço. Se há a necessidade de se executar mais de um serviço ao mesmo tempo, também se faz necessário a adição de mais um servidor a tal infraestrutura. A gestão de infraestrutura de servidores está se tornando cada vez mais complexa e dispendiosa (Pasin; Perini; Bazzan, 2013). Apresenta-se na Figura 1 uma infraestrutura com arquitetura cliente-servidor convencional.

Esta configuração muitas vezes implica na ociosidade dos recursos de máquina disponíveis, uma vez que como cada servidor é dedicado a um único serviço, se o serviço não utiliza 100% da capacidade computacional daquele *hardware*, há um desperdício dos recursos computacionais que poderiam ser destinados à execução de outras aplicações

Figura 1 – Arquitetura cliente-servidor convencional.



Fonte: Elaborado pelo Autor.

(redhat.com, 2018). Apresenta-se na Figura 2 um exemplo de servidores executando com apenas uma parte de seus recursos sendo utilizados.

Figura 2 – Exemplo de uso da capacidade computacional e ociosidade de recursos.



Fonte: Adaptado de Gonçalves *et al.* (2021).

Com o crescimento de serviços oferecidos e realizados pela Internet, a popularização de dispositivos de informática como computadores pessoais, *notebooks*, *tablets* e principalmente *smartphones*, em conjunto com a modernização dos processos produtivos e comerciais, houve um enorme crescimento pela demanda de serviços de infraestrutura para executarem tais aplicações remotas. Outro fator que deve ser considerado é o consumo energético que essas infraestruturas demandam.

Jones (2018) apontou que os data centers em 2018 consumiam cerca de 200 terawatt-hora(tWh) de eletricidade por ano, superando o consumo de energia de alguns países. A tecnologia de virtualização de servidores trouxe melhorias nos recursos utilizados pelos *data centers*, diminuindo o desperdício de energia já que possibilita a redução das máquinas

físicas em operação (Bhimani *et al.*, 2017).

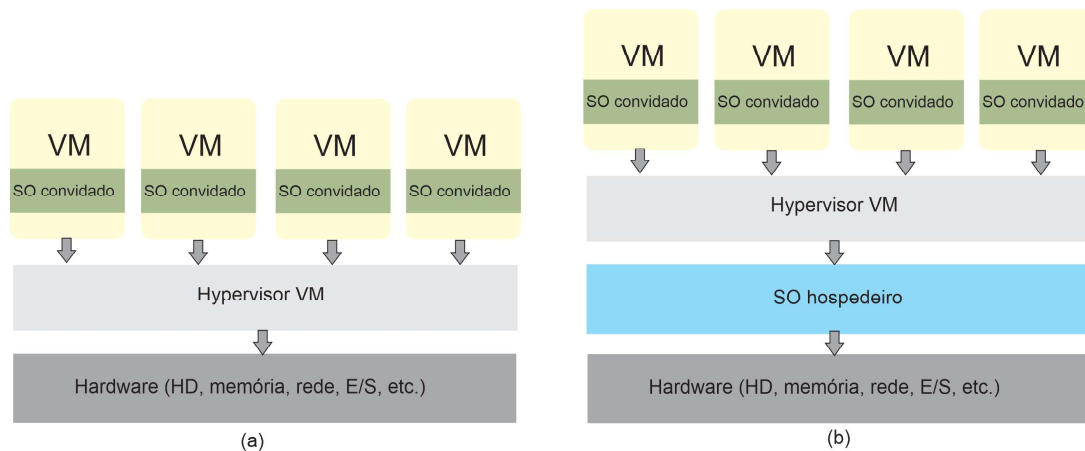
3.2 VIRTUALIZAÇÃO

A virtualização é uma tecnologia que se baseia no compartilhamento dos recursos de *hardware* de uma máquina física para replicar outras múltiplas máquinas virtuais independentes e isoladas entre si (Gonçalves *et al.*, 2021).

A principal função da virtualização é otimizar a utilização dos recursos de *hardware* visando a minimização da ociosidade, distribuindo parte dos recursos disponíveis para múltiplas instâncias ou ambientes, reduzindo os custos na aquisição de novos equipamento e consumo de energia.

Uma vez que a alocação de recursos se torna mais especializada ao tipo de aplicação que se deseja implementar, podem ser implementados diversos ambientes diferentes em um mesmo *hardware* (VMware.com, 2022). Na Figura 3 apresenta-se uma arquitetura de virtualização com *hypervisor* executando múltiplas máquinas virtuais.

Figura 3 – Virtualização Direto em *hardware* (a) e em SO hospedeiro (b).



Fonte: Adaptado de Bhimani *et al.* (2017).

Para gerenciar múltiplas instâncias de máquinas virtuais é necessário a execução de um sistema controlador, que decida qual instancia iniciar e qual finalizar, além de inspecionar elementos como endereços IP (*Internet Protocol*) e portas de acesso de todas as instâncias (Bhimani *et al.*, 2017).

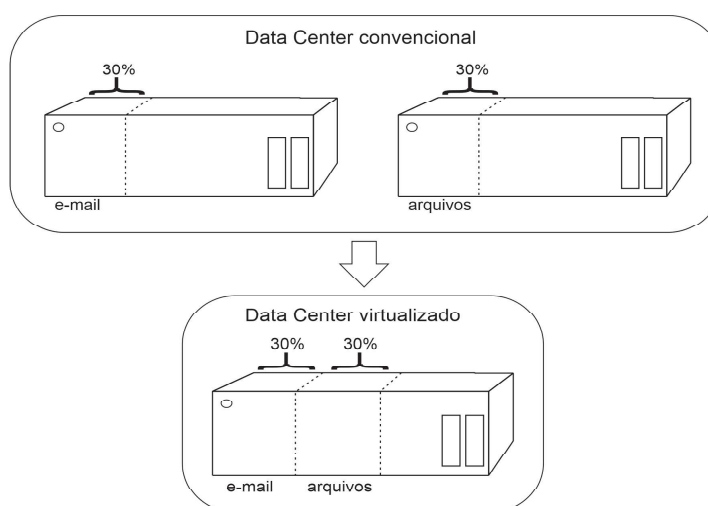
Este sistema controlador é chamado de *hypervisor* e particiona os recursos físicos dos ambientes virtuais que precisem utilizar estes recursos, podendo ser executados em conjunto com um sistema operacional (método tradicional usado em computadores pessoais) ou diretamente no hardware de um servidor (redhat.com, 2018).

Os usuários destes sistemas interagem e executam suas atividades computacionais no

ambiente virtualizado como um usuário *guest* (convidado). Como uma máquina virtual possui um único arquivo de dados, geralmente um arquivo de imagem ISO (*International Organization for Standardization*), a máquina pode ser transferida e replicada em qualquer outro dispositivo compatível e executada da mesma forma. Caso um programa ou usuário necessite da utilização de recursos adicionais na máquina virtual, o *hipervisor* faz a solicitação ao sistema físico e realoca os recursos necessários (Lee; Yoo, 2018).

O gerenciamento de um *data center* também é simplificado, pois em um único servidor físico é possível executar várias máquinas virtuais rodando serviços diferentes e compartilhando os mesmos recursos, proporcionando a automação de tarefas e aumento da disponibilidade dos serviços ou aplicações, otimizando a alocação e gerenciamento destes recursos (Barik *et al.*, 2016). Na Figura 4 pode-se visualizar a utilização da virtualização para executar diversos serviços em um único servidor.

Figura 4 – Processo de virtualização de *data centers*.



Fonte: Adaptado de redhat.com (2018).

Além da virtualização de *data centers* existem vários outros recursos que a tecnologia oferece. Apresenta-se nos itens 1 a 4 outras aplicações que podem ser implementadas através da virtualização:

Virtualização de dados: permite que dados distribuídos ou coletados de várias fontes possam ser consolidados em uma única fonte, permitindo o tratamento dinâmico destes dados, assim os dados são fornecidos as aplicações no formato e tratamento correto (redhat.com, 2018);

Virtualização de SO (sistemas operacionais): proporciona o uso e distribuição paralela de vários sistemas operacionais ao mesmo tempo, como Windows e Linux, reduzindo

o custo com *hardware* e aumentando a segurança uma vez que cada instância pode ser monitorada isoladamente (Lopes *et al.*, 2017);

Virtualização de rede: Yi *et al.* (2018) em seu trabalho cita que a virtualização de rede pode trazer independência de *hardware* pois separa as funções de rede do sistema e as implementa como *software*, assim os serviços ou novos requisitos de rede podem ser atendidos através de uma atualização de *software*;

Virtualização de *desktop*: este tipo de virtualização se difere da virtualização de SO, pois um único *hypervision* pode implementar ambientes *desktop* simulados em várias máquinas de uma vez, mantendo somente uma imagem em memória da MV (Máquina Virtual) (Lopes *et al.*, 2017). Isso permite que um único administrador realize atualizações e verificações de segurança ao mesmo tempo, em todos os *desktops* virtuais.

A virtualização é um recurso extremamente útil quando se deseja replicar uma mesma configuração de *hardware* para diferentes ambientes, quando não há espaço físico para uma infraestrutura de servidores ou até mesmo para criação de um ambiente virtual compartilhado.

Porém, é importante ressaltar que o processo de virtualização pode demandar alto consumo de armazenamento e memória e ocasionar sobrecarga de tarefas, uma vez que um único *hardware* será responsável por executar simultaneamente várias máquinas virtuais. Além disso, caso ocorra algum problema com o *hipervisor* todas as máquinas virtuais poderão ficar vulneráveis ou inativas, causando problemas de acesso aos conteúdos das máquinas virtuais.

3.3 CONTÊINERES

A containerização é uma alternativa ao processo tradicional de virtualização, pois se baseia na criação de várias instâncias virtuais como mecanismos que empacotam arquivos, bibliotecas e dependências de aplicações isoladamente para a execução de determinado sistema ou serviço.

O conceito fundamental da containerização é fazer instâncias virtuais compartilharem um único sistema operacional *host* e bibliotecas, *drivers* ou binários importantes para a execução de alguma aplicação (Barik *et al.*, 2016).

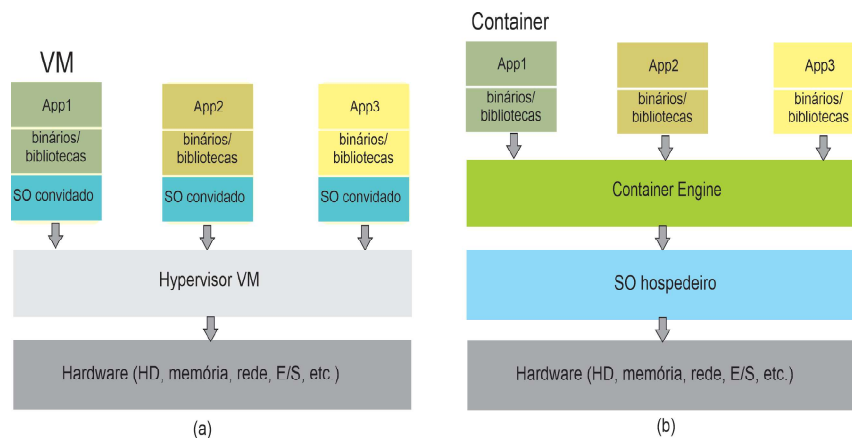
Em contraste com a virtualização de servidores que requer instâncias de sistemas operacionais separadas para máquinas virtuais, vários contêineres compartilham o mesmo sistema operacional. Lee and Yoo (2018) enfatiza em seu trabalho que um contêiner tem

um sistema de arquivos independente com binários e bibliotecas de sistema compartilhados com segurança.

Com base na arquitetura da virtualização baseada em contêiner, os contêineres oferecem várias vantagens em relação às máquinas virtuais, como alto desempenho, eficiência de recursos e ambiente ágil.

A principal diferença em relação às tecnologias tradicionais de virtualização é que nos contêineres não é necessária uma camada de emulação ou um *hypervisor* para serem executados, pois são executados pelo próprio sistema operacional. Além disso, os contêineres armazenam somente os arquivos necessários para a execução de um ou vários processos empacotados em uma imagem, iniciada isoladamente no sistema operacional *host*. Na Figura 5 são apresentadas as diferenças entre a arquitetura de virtualização e de contêineres.

Figura 5 – Comparação entre virtualização (a) e containerização (b).



Fonte: Adaptado de Bhimani *et al.* (2017).

Cada contêiner é executado como um processo separado que compartilha os recursos do sistema operacional *host*, isso permite que os *contêineres* sejam inicializados e interrompidos com facilidade (amazon.com, 2022). A pesquisa de Cuadrado-Cordero, Orgerie and Menaud (2017) avaliou o desempenho entre máquinas virtuais do tipo KVM (*Kernel Virtual Machine*) e contêiner usando a ferramenta Docker analisando eficiência energética e qualidade do serviço e, concluiu-se que o tempo de resposta dos contêineres foi 26% melhor, comprovando que contêineres consomem menos energia do que o KVM.

Gonçalves *et al.* (2021) concluiu em seus testes que a energia consumida nas operações utilizando contêineres geraram redução nos custos por kWh (Kilowatt hora) entre 45 e 73%, confirmando maior eficiência energética dos contêineres.

Barik *et al.* (2016) realizaram uma avaliação de desempenho entre containerização e virtualização através do Oracle Virtual Box e Docker, concluindo que o contêiner superou

a VM na maioria dos experimentos quanto ao uso de CPU (*Central Processor Unit*) , memória e disco rígido.

Por fim, a tecnologia de contêineres aplicada a dispositivos IoT possibilita a execução de diferentes dispositivos IoT simultaneamente, melhorando a utilização dos recursos, pois como cada contêiner é independente, possibilita o compartilhamento dos recursos computacionais do dispositivo IoT com múltiplos contêineres (Lee; Yoo, 2018).

3.4 PLATAFORMAS EM NUVEM PARA VIRTUALIZAÇÃO E *CONTAINERS*

Serviços em nuvem vem sendo adotados por empresas de todos os setores produtivos como uma alternativa às infraestruturas de servidores e virtualização. Diferente das estruturas tradicionais de virtualização e containerização que ainda dependem de uma arquitetura de *hardware* física para serem executados, os serviços em nuvem necessitam apenas que os usuários possuam um dispositivo para acesso e não de um servidor robusto.

Ranjan *et al.* (2020) destaca que a computação em nuvem emergiu como uma das mais poderosas tecnologias para fornecer serviços sob demanda para usuários finais. Os principais pontos fortes da tecnologia são a escalabilidade e distribuição dos recursos computacionais, alta disponibilidade das aplicações e melhor utilização dos recursos computacionais.

A computação em nuvem pode ser vista como um método de aluguel de recurso computacional e armazenamento de data centers de terceiros, devolvendo estes recursos após o uso e pagando somente pelos recursos utilizados (microsoft.com, 2022c).

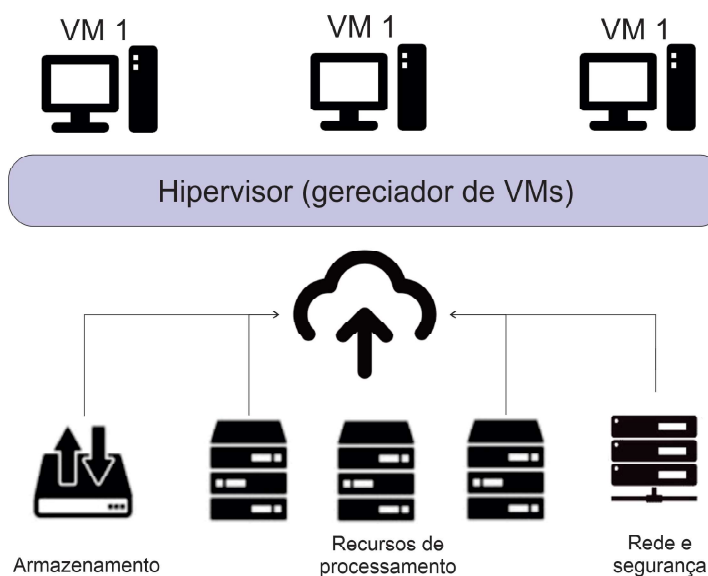
Sistemas em nuvem vem sendo utilizados para disponibilização de ferramentas de *software* em várias áreas da engenharia. Boid *et al.* (2018) apresentaram uma ferramenta educacional em nuvem onde os alunos podem construir protótipos de *hardware* através de interações visuais e sintetizá-los em HDL(*Hardware Description Language*) , onde todo o sistema é executado por meio de uma aplicação em nuvem.

Firestone *et al.* (2018) utilizaram a tecnologia de FPGAs (*field-programmable gate array*) em conjunto com o serviço em nuvem Azure AccelNet para melhorar o processo de gerenciamento de cargas em pilhas de redes, redirecionando as pilhas de conexões para o *hardware* reconfigurável que é programado somente com os elementos essenciais para a aplicação em execução. A medida que as cargas de rede sofrem aumento na pilha de execuções, o dispositivo FPGA é reprogramado para atender aos requisitos de rede.

Dentre os serviços em nuvem presentes no mercado destacam-se as tecnologias de IaaS, PaaS e SaaS. No primeiro serviço em nuvem tem-se a Infraestrutura como Serviço onde os recursos de computação, armazenamento e rede são fornecidos sob demanda e

pagos conforme o uso. Neste tipo de sistema, um conjunto de *data centers* trabalham colaborativamente fornecendo recursos computacionais através de máquinas virtuais, que podem oferecer diferentes configurações de processamento, memória e armazenamento simultaneamente (Li *et al.*, 2012). Na Figura 6 apresenta-se a arquitetura do serviço IaaS.

Figura 6 – Arquitetura do serviço cloud IaaS.



Fonte: Elaborado pelo autor.

Mistry, Mohammad and Bouguettaya (2021) afirmam que as VMs fornecidas em IaaS fornecem aos usuários a capacidade e flexibilidade de escolher a combinação apropriada de recursos para suas aplicações. O serviço foi usado em sua pesquisa para executar uma aplicação de Aprendizagem Sequencial baseada em IaaS, proporcionando maior adaptabilidade da aplicação devido à natureza dinâmica que o serviço IaaS consegue oferecer.

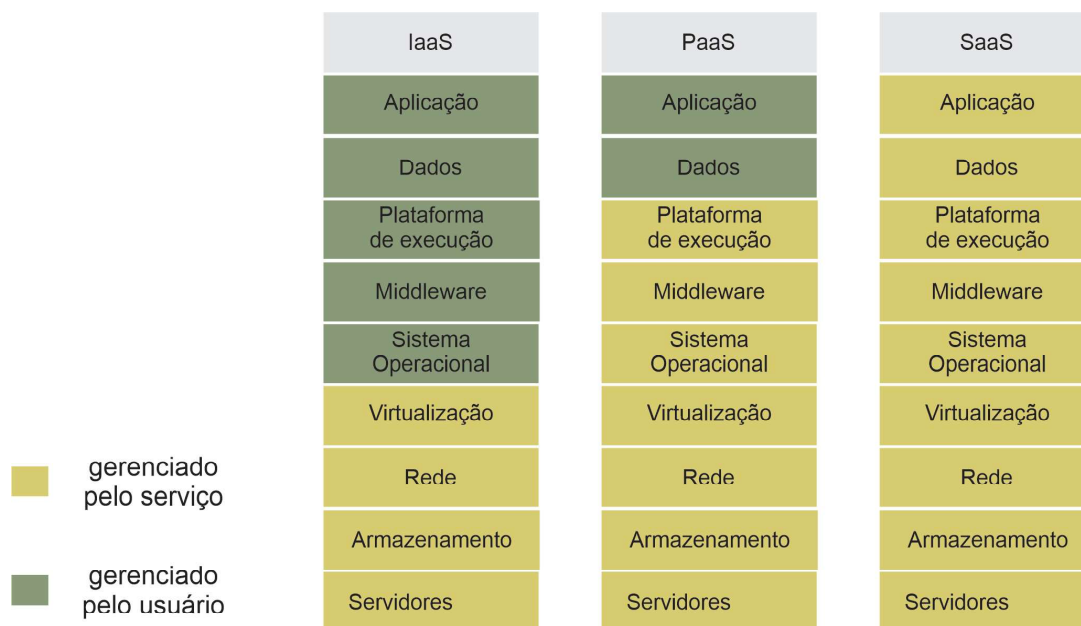
O segundo serviço abordado é chamado PaaS, que além de englobar os recursos presentes na plataforma IaaS também conta com o sistema operacional necessário para a execução das aplicações, bem como as ferramentas de banco de dados, *frameworks* de desenvolvimento e outras ferramentas de hospedagem e manutenção do sistema.

Este recurso é utilizado principalmente por desenvolvedores para implementar, testar e validar suas aplicações. As plataformas PaaS visam facilitar o desenvolvimento e a implantação de aplicativos em nuvem, fornecendo uma plataforma de computação e uma pilha de soluções como serviço (Walraven; Truyen; Joosen, 2014). Em linhas gerais, PaaS é um serviço em nuvem que adiciona ao IaaS as funcionalidades e ferramentas de implementação, versionamento e disponibilização de sistemas em nuvem para o usuário

final.

A plataforma SaaS é o serviço de implementação e utilização de recursos em nuvem mais completo. Neste sistema todos os recursos em nuvem são disponibilizados e gerenciados pelo provedor, ficando a cargo do desenvolvedor somente as tarefas de programação e implementação do sistema. Na Figura 7 apresenta-se um comparativo entre os três serviços em nuvem apresentados.

Figura 7 – Diferença entre os serviços disponíveis nas plataformas IaaS, PaaS e SaaS.



Fonte: Adaptado de (microsoft.com, 2022c).

Com o uso de SaaS, todos os aplicativos são *softwares* hospedados na *web*, ficando disponíveis a qualquer momento, podendo ser acessados por qualquer dispositivo. Além disso, pela arquitetura em nuvem SaaS é possível utilizar toda uma infraestrutura de *data centers* sem a necessidade de aquisição de equipamentos como servidores, componentes de rede, etc.

Kim, Jang and Yang (2017) afirmaram que SaaS fornece serviços de TIC (Tecnologia da Informação e Comunicação) maiores e com maior qualidade a um custo menor que os métodos convencionais, além de melhorar os processos de negócios das instituições. A manutenção é facilitada em aplicações em nuvem, pois quando é realizada uma modificação ou atualização na aplicação, as mudanças ficam imediatamente disponíveis para todos os usuários clientes.

4 FUNDAMENTOS TEÓRICOS E MATERIAIS

Neste capítulo apresentam-se as ferramentas utilizadas no desenvolvimento do projeto, apresentando a plataforma em nuvem Microsoft Azure, abordando o método de assinatura do serviço e os principais serviços, além de apresentar quais serviços foram utilizados no desenvolvimento em nuvem para monitoramento da qualidade do ar.

Serão apresentados os componentes da camada física do projeto: o microcontrolador ESP8266, multiplexador analógico-digital CD74HC4067 e sensores MICS6814, DHT22 e DSM501A.

Por fim, são descritos os protocolos TCP/IP e MQTT usados para a comunicação do dispositivo físico com a nuvem através de WiFi e envio dos dados de telemetria para o serviço em nuvem.

4.1 AMBIENTE DE INSTALAÇÃO DO DISPOSITIVO IOT

Para testes e coleta de dados sobre as condições ambientais da qualidade do ar o projeto foi aplicado em laboratório de informática com capacidade para 20 alunos na escola técnica de Jales (ETEC Jales). O ambiente contém 20 computadores *desktop*, TV de 50 polegadas e quadro branco. A climatização é realizada por meio de ar condicionado com temperatura fixada em 23 °C.

O dispositivo IoT foi instalado sobre um armário para *switch* de rede, localizado próximo à porta de entrada do laboratório. A escolha do local de instalação se justifica por ser um local alto, fora da ventilação do ar-condicionado e longe dos alunos que utilizam o laboratório. Foram lidos dados por 4 semanas, as quintas e sextas-feiras das 8h às 17h, horário de aulas no laboratório, com poucos intervalos onde o ambiente esteve vazio.

A escola fica localizada no centro da cidade, próxima da região comercial, onde o fluxo de veículos e pessoas nos arredores é muito intenso. No período da tarde, a partir das 13h30min, o trânsito apresenta maior volume, tendo como pico de fluxo das 16 às 19h. O laboratório fica localizado na área externa da escola, próximo às ruas de acesso às duas avenidas de acesso principal ao centro da cidade.

4.2 MICROSOFT AZURE

O serviço em nuvem Microsoft Azure foi lançado pela Microsoft em 2008 e oferece serviços nos formatos PaaS, SaaS, IaaS entre outros com alto grau de confiabilidade, escalabilidade e baixo custo de infraestrutura (Rizik *et al.*, 2019).

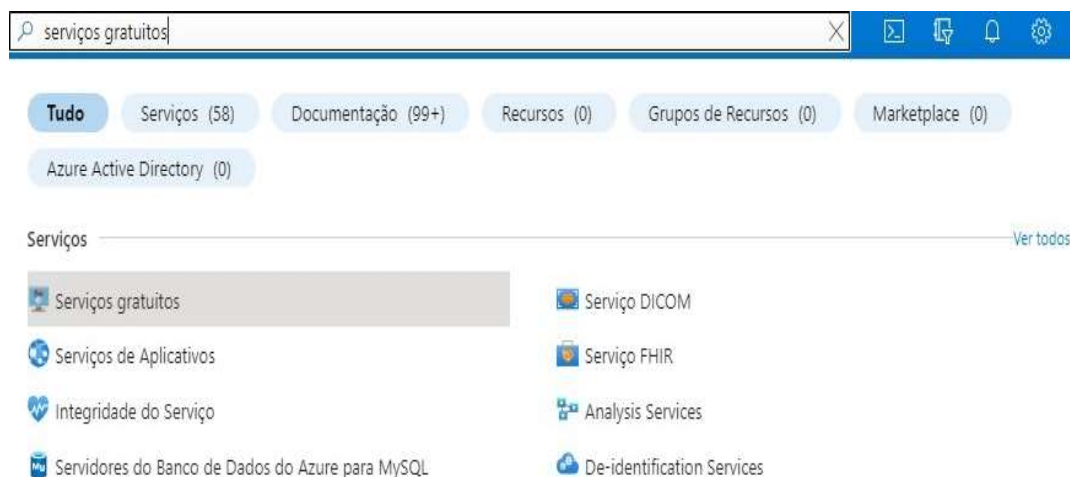
Os mais de 200 produtos e serviços em nuvem disponibilizados pela Azure podem ser categorizados pelos seguintes tópicos:

- **Desenvolvimento de aplicativos:** processos de desenvolvimento e testes, microsserviço, desenvolvimento SaaS, mensageiria;
- **Inteligência Artificial:** análise de imagens e fala, análise de dados e criação de modelos de *machine learning*;
- **Modernização e migração para nuvem:** migração de dados, cargas de trabalho, aplicativos, servidores e serviços de integração;
- **Dados e análise:** serviços de BI (*Business Intelligence*) ;
- **Nuvem híbrida e infraestrutura:** soluções multinuvem, *backup* e recuperação, computação de alto desempenho e servidores;
- **Serviços para IoT:** conexão de dispositivos, análise de dados e automatização de processos;
- **Segurança e governança:** conformidade com a legislação vigente no ambiente de hospedagem e execução e proteção de dados;
- **Soluções da indústria:** como serviços financeiros, energia, mídia e entretenimento entre outros.

Para acessar os serviços Azure é necessário criar uma conta usando seus dados pessoais como documentos, endereço, e-mail, senha para acesso e cartão de crédito para posterior cobrança. Ao criar uma conta Azure, o novo usuário recebe um crédito de US\$ 200 na moeda de cobrança para serem utilizados nos primeiros 30 dias de acesso, uma quantidade de serviços gratuitos por 12 meses e outros serviços gratuitos sempre. Caso o usuário não ultrapasse os limites de uso para os serviços gratuitos, ele não será cobrado (learn.microsoft.com, 2023c).

É possível acessar os serviços gratuitos buscando na página inicial do usuário usando o termo "serviços gratuitos" e ao acessar os serviços gratuitos haverá uma listagem com

Figura 8 – Acesso à listagem detalhada de serviços gratuitos Azure.



Fonte: Retirado de (microsoft.com, 2022a)

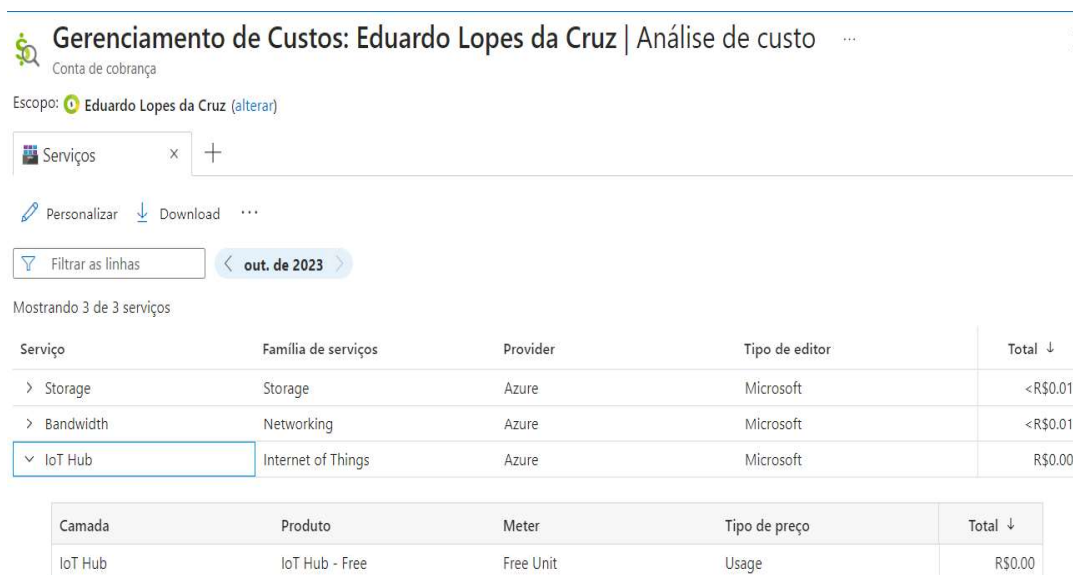
os serviços. Na Figura 8 demonstra-se como acessar a listagem detalhada de serviços gratuitos.

Caso um usuário continue a utilizar um serviço gratuito com prazo de validade após o período de gratuidade, são cobradas taxas conforme uso. A Microsoft envia uma notificação para o e-mail cadastrado sobre os serviços que expiram em data próxima, além da possibilidade de excluir um serviço gratuito ou pago sob uso a qualquer momento.

É possível monitorar os custos provenientes de cada serviço utilizado acessando o serviço de gerenciamento de custos e clicando na opção "análise de custos". Além disso, é possível gerar uma análise prévia de custo mensal, alerta de custos para cada serviço (que avisa quando um serviço atingiu um custo especificado), visualizar custos por consumo e cobranças anteriores. Na Figura 9 ilustra-se a área de gerenciamento de custos do Microsoft Azure (learn.microsoft.com, 2023f).

Cada serviço Azure possui um método de configuração e uma forma de acesso, todos vinculados a conta do usuário. É possível realizar a integração entre os serviços Azure através das ferramentas de configuração e integração disponíveis dentro da plataforma. A integração entre serviços de terceiros pode ser realizada por meio de APIs (*Application Program Interface*) e chave de acesso (learn.microsoft.com, 2023d).

As próximas seções descrevem com mais detalhes quais os serviços Azure foram utilizados para criar a camada de aplicação em nuvem do projeto de monitoramento da qualidade do ar.

Figura 9 – Acesso à listagem detalhada de serviços gratuitos Azure.


Gerenciamento de Custos: Eduardo Lopes da Cruz | Análise de custo

Conta de cobrança

Escopo: **Eduardo Lopes da Cruz** (alterar)

Serviços

Personalizar Download

Filtrar as linhas out. de 2023

Mostrando 3 de 3 serviços

Serviço	Família de serviços	Provider	Tipo de editor	Total ↓
> Storage	Storage	Azure	Microsoft	<R\$0.01
> Bandwidth	Networking	Azure	Microsoft	<R\$0.01
▼ IoT Hub	Internet of Things	Azure	Microsoft	R\$0.00

Camada	Produto	Meter	Tipo de preço	Total ↓
IoT Hub	IoT Hub - Free	Free Unit	Usage	R\$0.00

Fonte: Retirado de (microsoft.com, 2022a)

4.2.1 Azure IoT Hub

A Internet das Coisas (do termo em inglês *Internet of Things* (IoT)) atualmente consiste em uma rede que aporta um número gigantesco de dispositivos heterogêneos implantados de forma generalizada que permite novas e diferentes formas de interação entre coisas e pessoas, que variam de dispositivos altamente restritos, acessados via IP, *smartphones* e nuvem (Cirani *et al.*, 2018).

Tendo em vista estes aspectos, um *Hub* IoT representa uma infraestrutura fundamental para a implantação e gerenciamento de diferentes dispositivos IoT que estejam interconectados ou não. Um *Hub* IoT permite que vários dispositivos diferentes trabalhem em conjunto, possibilitando a criação de novas aplicações e padrões de interação humano-máquina, *machine learning* e análise de dados (Cirani *et al.*, 2018).

A necessidade de interconectar e gerenciar diferentes dispositivos IoT fez com que Abdelqawy *et al.* (2022) em seu trabalho desenvolvessem uma plataforma para computação IoT chamada Hub-OS que gerencia dispositivos e aplicações de diferentes fornecedores, sem a necessidade de modificações nos componentes de *software* do dispositivo ou do Hub-OS.

O Hub IoT Azure é um serviço que atua como *hub* central de mensagens para a comunicação entre os dispositivos IoT e outros serviços ou aplicativos. Há suporte para diversos formatos de mensagens como telemetria, envio de arquivos, monitoramento, criação de métodos de solicitação-resposta para controle de dispositivos

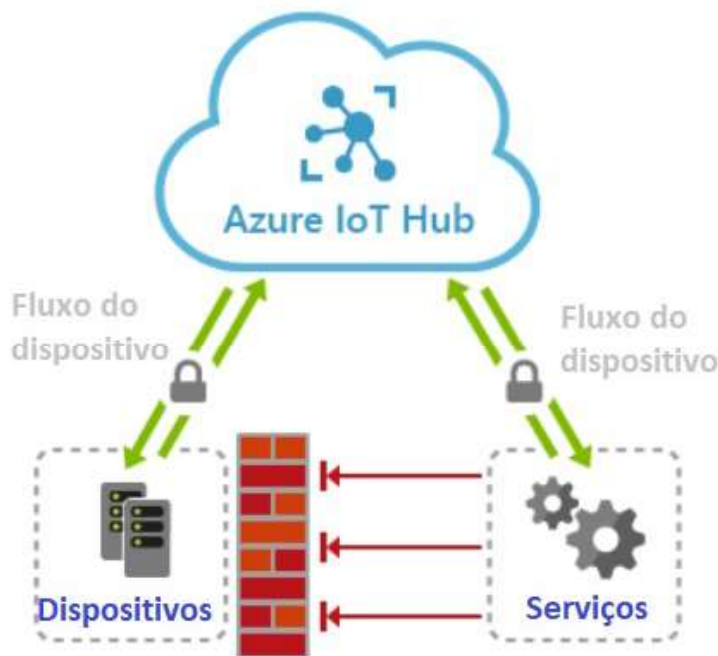
(learn.microsoft.com, 2023b).

Com o Hub IoT é possível permitir conexões entre pontos de extremidade de entrada e saída, simplificando a comunicação do dispositivo IoT com outros serviços, podendo realizar tráfego de dados utilizando diversos protocolos como MQTT, TCP/IP, protocolos proprietários entre outros.

O serviço se comunica com o dispositivo IoT através de um identificador (ID) ao invés de um endereço IP, facilitando a conexão entre o serviço e o dispositivo físico, pois se a configuração de endereço IP em uma rede é privado ou atribuído dinamicamente, isso não afetará na comunicação entre as duas extremidades, além de proteger as informações da rede ao qual o dispositivo IoT está conectado.

Outro ponto a ser destacado é que os demais serviços Azure ou de terceiros não acessam diretamente o dispositivo IoT, criando uma camada adicional de segurança entre as extremidades. Na Figura 10 demonstra-se a comunicação entre o dispositivo IoT, IoT Hub e outros serviços.

Figura 10 – Comunicação entre as extremidades do IoT Hub.



Fonte: Adaptado de learn.microsoft.com.

Ahmed *et al.* (2020) apresentam uma solução de irrigação inteligente utilizando sensores de umidade do solo, temperatura, umidade relativa do ar entre outros, conectados a um Raspberry Pi 3, que funciona como *gateway* IoT coordenando os sensores e agregando os dados coletados.

O Hub IoT Azure é utilizado para provisionar estes dados em nuvem para acesso em

tempo real e disponibilizar para outros serviços em nuvem como Stream Analytics usado para análise dos dados em tempo real e o serviço Event Hub, que permite o acionamento de um sensor/atuator de acordo com um valor recebido ou o resultado da análise dos dados pelo serviço Stream Analytics.

Neste trabalho foi utilizado um serviço em nuvem Azure IoT Hub para realizar o gerenciamento e a comunicação de dispositivos IoT com os serviços em nuvem. O Hub IoT é responsável por realizar a autenticação do dispositivo IoT com a nuvem Azure, receber os dados de telemetria e redirecionar para os serviços em nuvem IoT Central e Blob Storage.

O método de assinatura escolhido para o serviço foi padrão gratuito, que permite a troca de até 8 mil mensagens por dia sem custos. Esta configuração é realizada ao criar um novo IoT Hub no serviço Microsoft Azure. Na próxima seção é apresentado o serviço IoT Central.

4.2.2 Azure IoT Central

Facilitar a disponibilização e visualização de dados coletados por meio de dispositivos IoT de maneira simplificada através da internet é extremamente importante para verificar as condições da qualidade do ar em ambientes internos. Criar uma interface de usuário que exiba as informações em tempo real de forma precisa, organizada e de fácil compreensão é essencial para a tomada de decisões sobre quais equipamentos acionar ou não, visando alterar as condições climáticas e de qualidade do ar de um ambiente conforme o seu estado atual.

Um serviço em nuvem Azure que fornece todas as ferramentas necessárias para atender tais necessidades é o IoT Central, pois fornece um ambiente completo para visualização e compartilhamento das informações coletadas de dispositivos IoT através da Internet por meio de uma interface simples, editável e de fácil compreensão.

Segundo Secker (2020), o serviço IoT Central é uma plataforma simples, escalonável, que mescla os principais serviços IoT em uma única interface, simplificando a implementação de projetos com IoT.

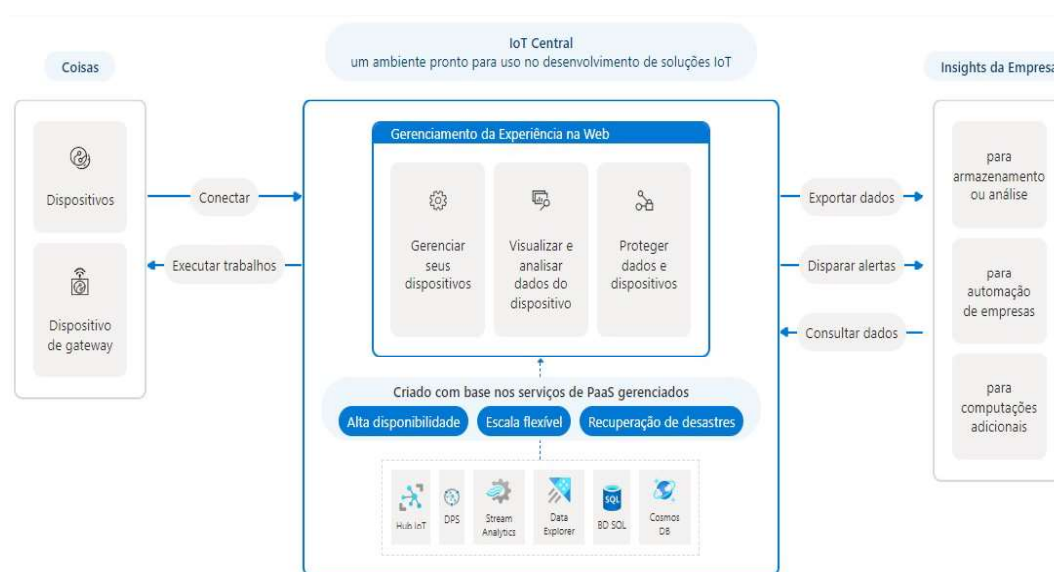
IoT Central fornece um alto nível de segurança na sua conexão com os dispositivos, pois possui um sistema de autenticação baseado em certificado com chave simétrica, onde o dispositivo e o serviço compartilham uma mesma chave privada utilizada para autenticação na nuvem e criptografia das mensagens trocadas (Secker, 2020).

O serviço IoT Central oferece uma infraestrutura com análises integradas que permitem o gerenciamento de informações com menos custos e acesso mais rápido aos

resultados. Possui painéis que demonstram o status do dispositivo, mapas, gráficos e blocos que mostram os dados de telemetria do dispositivo (Sreyes *et al.*, 2022).

Sua infraestrutura como PaaS de aplicativo permite a redução de trabalho no desenvolvimento de interface para o gerenciamento e manutenção de soluções IoT. Facilita o monitoramento das condições do dispositivo, permitindo exportar dados para criar soluções mais elaboradas com base nas informações coletadas e armazenadas (learn.microsoft.com, 2023g). Na Figura 11 apresenta-se a estrutura do serviço Azure IoT Central.

Figura 11 – Estrutura do Azure IoT Central.



Fonte: Retirado de: (microsoft.com, 2022b).

O acesso aos dados coletados e exibidos pelo IoT Central podem ser disponibilizados para outros serviços Azure para análises mais elaboradas, bem como baixados no formato de planilha em arquivo .CSV (*Comma-Separated Values*) ou acessados externamente via API utilizando *tokens* de autenticação fornecidos pelo próprio serviço.

O método de cobrança adotado no serviço IoT Central é dividido em camadas de acordo com a quantidade de mensagens enviadas do dispositivo IoT para o serviço em nuvem.

No plano *Standard 0* é cobrado o valor de R\$ 0,40 por mês e permite o envio de 400 mensagens, no plano *Standard 1* possui custo de R\$1,98 por mês com envio de 5.000 mensagens e por fim o plano *Standard 2* custa R\$ 3,52 por mês e disponibiliza o envio de 30.000 mensagens. Em todos os planos estão incluídas permissão de acesso gratuito ao serviço IoT Central por dois dispositivos (dados verificados em setembro de 2023) (azure.microsoft.com, 2023b).

4.2.3 Azure Blob Storage

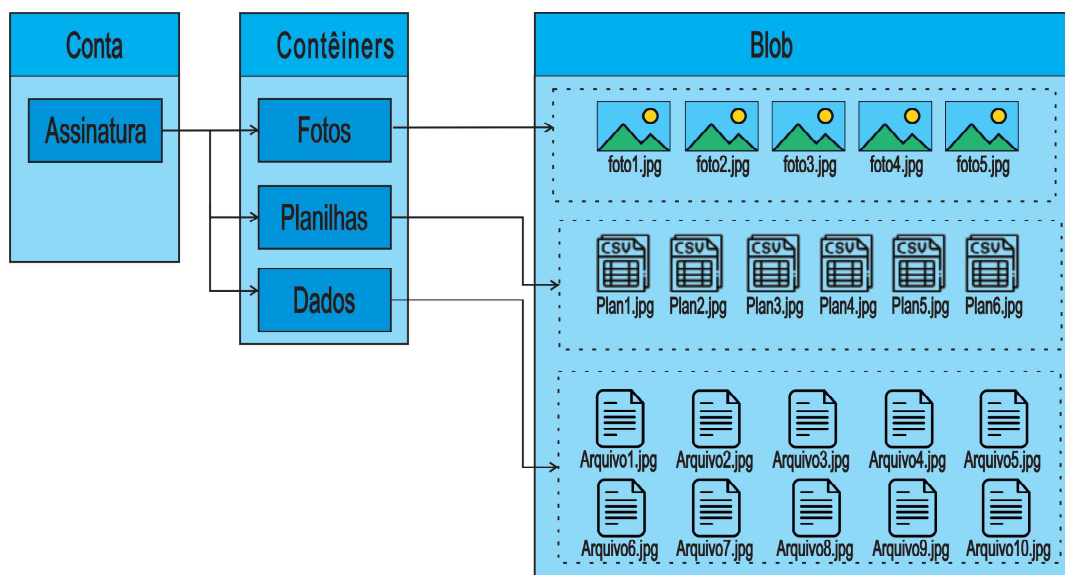
Armazenar os dados coletados por dispositivos IoT permite que análises mais elaboradas possam ser realizadas, disponibilizar esses dados através da Internet ou até mesmo para serem utilizados por algoritmos de aprendizagem de máquina ou outros métodos de inteligência artificial.

Existem inúmeras formas de armazenamento de informações como planilhas, banco de dados, *data lakes* e *data warehouses*, cada um com características distintas e funcionalidades para armazenamento, manipulação e disponibilização de dados.

O serviço em nuvem utilizado para armazenar os dados coletados do dispositivo IoT que mede as condições da qualidade do ar de ambientes é o Azure Blob Storage, que permite o armazenamento de grandes volumes de dados em forma de *data lakes*, que são repositórios que armazenam grandes volumes de dados estruturados ou não em sua forma original (Oracle, 2023).

O serviço permite a elaboração de camadas de armazenamento, organizando tais camadas de forma hierárquica, sendo necessária uma conta de armazenamento principal que gerencie os outros recursos. A partir da conta de armazenamento são criados contêineres de dados, cada contêiner pode armazenar dados de maneiras diferentes como imagens, vídeos, planilhas ou dados no formato de objetos (learn.microsoft.com, 2023e). Na Figura 12 apresenta-se um exemplo de armazenamento com o serviço Blob Storage.

Figura 12 – Estrutura de armazenamento Blob Storage.



Fonte: Adaptado de: (learn.microsoft.com, 2023e).

Os dados armazenados nos contêineres podem ser disponibilizados para outros serviços

em nuvem Azure ou de terceiros através e APIs de acesso via protocolos HTTP (*Hypertext Transfer Protocol*) ou HTTPS (*Hyper Text Transfer Protocol Secure*).

O serviço Blob Storage opera com cobrança conforme uso referente a quantidade de gigabytes (GB) usados mensalmente, total de operações de gravação e leitura dos dados, entre outras operações. Os valores variam entre os planos e operações realizadas. O plano padrão têm custo de R\$0,10 por GB, R\$0,50 para gravação de até 10.000 operações de gravação e R\$25,15 para até 10.000 operações de leitura de dados (com acesso externo via API) (azure.microsoft.com, 2023a).

Os preços de cobrança podem variar conforme a cotação do dólar ou quantidade de uso do serviço. Custos de serviços de rede são cobrados separadamente.

Armazenar dados no formato de blobs é interessante para aplicações que trabalham com grandes volumes de dados não estruturados, que podem ser manipulados por diferentes serviços, devido à flexibilidade na extração de informações para diferentes usos.

4.3 DISPOSITIVOS FÍSICOS

Esta seção descreve os dispositivos utilizados na construção da camada física do projeto, abordando o microcontrolador ESP8266, multiplexador analógico-digital CD74HC4067 e os sensores MICS6814, DHT22 e DSM501A.

4.3.1 Microcontrolador ESP8266

Projetos baseados em IoT demandam que a camada física do projeto seja confiável, compacta, robusta e que permita a escalabilidade e conectividade de fácil configuração.

O microcontrolador ESP8266 oferece processador de 32 bits da série L106 Diamond da Tensilica, adaptador Wi-Fi integrado que opera com o protocolo IEEE 802.11 b/g/n, baixo consumo de eletricidade, compacto e desempenho confiável para ser aplicado na indústria IoT (Espressif, 2023).

Pode funcionar como um elemento independente ou como dispositivo escravo em uma rede de dispositivos IoT. Pode ser aplicado em projetos domésticos, automação residencial, controle industrial sem fio, rede de sensores entre outros. No Quadro 5 apresentam-se suas principais características.

Possui processador RISC (*Reduced Instruction Set Computer*) de consumo elétrico extra baixo, frequência de *clock* que pode variar entre 24 e 52MHz (megahertz). Trabalha com o protocolo TCP/IP e protocolo MAC (*Media Access Control*) para redes tipo WLAN (*Wireless Local Area Network*) 802.11 b/g/n (Espressif, 2023). Na Figura 13 apresenta-se o microcontrolador ESP0266 e seu esquema de pinos.

ou utilizando a ferramenta de programação para Arduino através da linguagem C. Para utilizar a ferramenta Arduino é necessário adicionar a biblioteca de programação de placas ESP nas configurações de conexão a dispositivos da ferramenta Arduino IDE.

Uma característica importante a ressaltar sobre o ESP8266 é sua baixa quantidade de portas analógicas, possuindo somente uma funcional (pino A0), o que pode resultar na necessidade de utilização de um multiplexador ou outro dispositivo microcontrolador que possua mais portas analógicas caso o projeto necessite de sensores ou atuadores que operem com este método de comunicação.

Outro ponto a destacar é seu baixo custo (entre R\$20 e R\$60) e tamanho reduzido se comparado a um Arduino ou Raspberry Pi, facilitando sua instalação em ambientes internos.

4.3.2 Multiplexador analógico-digital CD74HC4067

O multiplexador analógico-digital CD74HC4067 foi utilizado no projeto para suprir a baixa quantidade de portas analógicas do microcontrolador ESP8266 (somente uma entrada analógica). Trabalha como um módulo 16 em 1, onde por meio dos 4 pinos digitais, definidos em código binário (0 - 15), proporciona a utilização de 16 entradas analógicas.

Este multiplexador controla as tensões que podem variar em toda a faixa de alimentação de tensão (2 a 6V). Possui interruptores bidirecionais que permitem qualquer conexão analógica de entrada possa ser utilizada como saída e vice-versa (INSTRUMENTS, 2023).

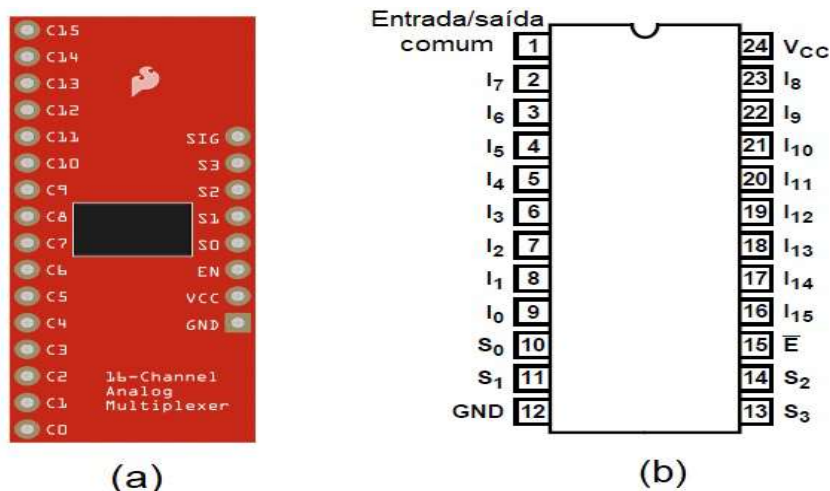
O multiplexador permitiu ao microcontrolador ESP8266 receber dados do sensor de gases MIS6814 que possui 3 saídas analógicas e do sensor de partículas DSM501A que contém uma saída analógica. Na Figura 14 demonstra-se o multiplexador CD74HC4067 e sua configuração de pinos.

O uso do multiplexador CD74HC4067 mostrou-se uma opção viável, de fácil configuração e baixo custo (R\$ 7,00 na data de aquisição), pois supriu a necessidade de expansão de portas analógicas sem comprometer a velocidade de aquisição de dados devido ao método de transferência de dados aplicado no código-fonte gravado no microcontrolador ESP8266.

4.3.3 Sensor detector de gases MICS6814

O sensor de gases MICS6814 Realiza a leitura simultânea dos níveis de concentração dos seguintes gases: monóxido de carbono(CO), dióxido de nitrogênio (NO_2) e

Figura 14 – Multiplexador CD74HC4067 (a) e seu esquema de pinos (b).



Fonte: Adaptado do *software* Fritzing e (INSTRUMENTS, 2023).

amônia(NH_3) entre outros 5 gases. Sua escala de detecção varia entre os diferentes gases na faixa entre 0.05 e 1000 ppm (partículas por milhão) (Sensortec, 2023). No Quadro 6 apresentam-se os gases detectáveis pelo sensor MICS6814 e suas escalas de detecção.

Quadro 6 – Gases detectados pelo sensor MICS6814 e suas escalas

Gás	Símbolo	ppm
Monóxido de carbono	CO	1-1000
Dióxido de nitrogênio	NO_2	0.05-10
Etanol	C_2H_5OH	10-500
Hidrogênio	H_2	1-1000
Amônia	NH_3	1-500
Metano	CH_4	>1000
Propano	C_3H_8	>1000
Iso-butano	C_4H_{10}	>1000

Fonte: Adaptado de (Sensortec, 2023)

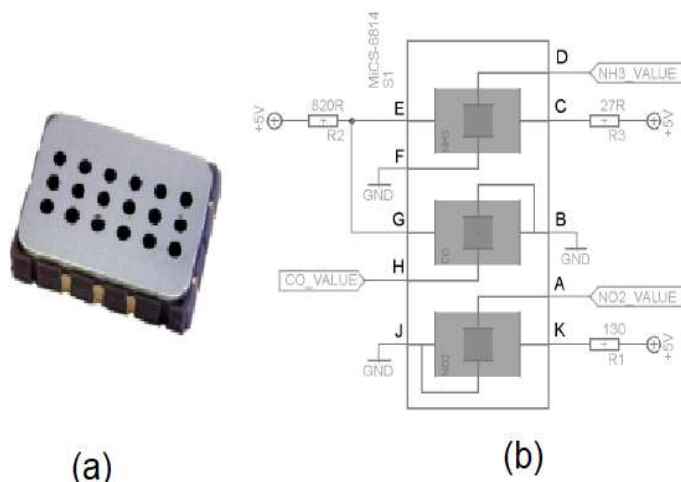
Sua estrutura consiste em um diafragma micro usinado de precisão com um resistor de aquecimento incorporado e uma camada de detecção na parte superior. Inclui também 3 chips sensores com aquecedores independentes com camadas de detecção (Sensortec, 2023).

É possível realizar a leitura dos gases CO_2 , NO_2 e NH_3 de forma direta, sem necessidade da utilização de bibliotecas e protocolos específicos. Caso seja necessário a detecção de outros gases, deve-ser utilizada uma biblioteca I2C para comunicação com o ESP8266 ou outro microcontrolador. Para este projeto os gases selecionados foram CO_2 , NO_2 e NH_3 , gases mais comuns encontrados em ambientes internos escolares.

Além disso, o sensor MICS6814 foi escolhido devido a seu tamanho reduzido e alta

precisão na detecção dos gases, apesar de possuir custo entre R\$90 e R\$230 na data da aquisição. Na Figura 15 apresenta-se o sensor MICS6814.

Figura 15 – Sensor de gases MICS6814 (a) e seu esquema elétrico (b).

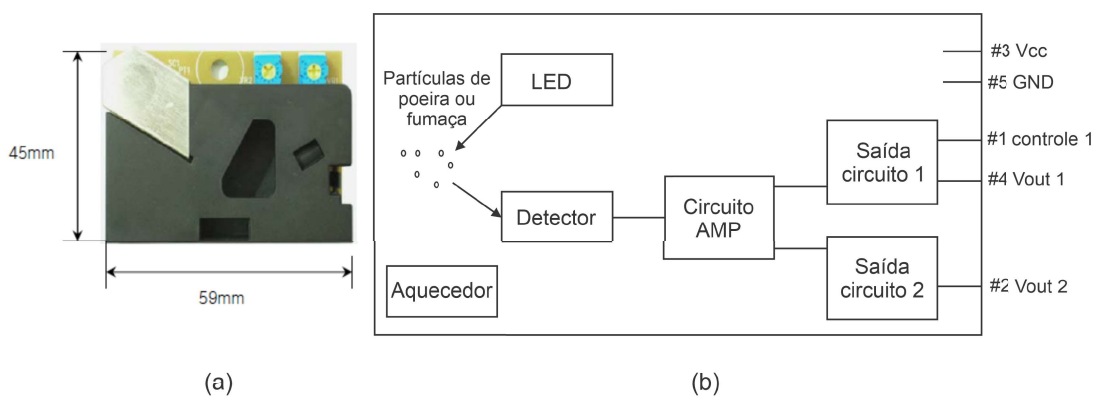


Fonte: Adaptado de (Sensortec, 2023).

4.3.4 Sensor de partículas DSM501A

O sensor DSM501A detecta partículas de tamanho compacto em suspensão no ar como poeira em partículas finas a partir de 1m (micrômetro) conforme os padrões MP 2.5 (Material particulado de diâmetro menor que 2.5m) e MP10 (Material particulado de diâmetro menor que 10m) através do recolhimento do ar pelo sensor, medindo a dispersão refletida nas partículas de luzes (SAMYOUNG, 2023). Na Figura 16 ilustra-se o sensor DSM591A.

Figura 16 – Sensor de partículas DSM591A (a) e seu diagrama de blocos (b).



Fonte: Adaptado de (SAMYOUNG, 2023).

Possui um aquecedor interno que induz a passagem do ar pelo módulo de detecção

enquanto o LED (*Light Emitting Diode*) interno ilumina as partículas para o módulo de detecção realizar a contagem de partículas. O pino #1 (controle) é usado para ajustar a sensibilidade quando o pino de saída #4 (Vout1) é utilizado (SAMYOUNG, 2023).

O pino #2 (Vout2) fornece um PWM (*Pulse With Modulation*) de saída detecção de partículas de 1m. Este é o pino padrão de saída segundo as configurações de fábrica. Já o pino #4 (Vout1) é utilizado quando se deseja ajustar o tamanho mínimo para detecção das partículas, utilizando as chaves presentes no sensor ou adicionando um resistor entre o pino #1 (controle) e pino #4 (Vout1) (SAMYOUNG, 2023).

Se a chave reguladora do pino #4 (Vout1) estiver completamente aberta, sua saída será referente a detecção de partículas acima de 2.5 m. Adicionado um resistor de 100K entre o pino #1 (controle) e pino #4 (Vout1), a saída apresentará sensibilidade média acima de 1.75 m. Se adicionar um resistor de 27K entre o pino #1 (controle) e pino #4 (Vout1) sua saída será igual a do pino 2 (Vout2) (SAMYOUNG, 2023). Neste projeto foram utilizadas ambas as saídas para promover a detecção de partículas até 2.5 m e acima de 2.5 m.

Deve ser instalado na vertical e longe do alcance de correntes de ar artificiais como ventiladores ou ar-condicionados. Também é necessário que ele esteja em local aberto ou se instalado em local fechado (como caixas protetoras) que existam fendas onde o ar possa circular sobre o sensor. Deve evitar seu uso próximo de locais emissores de partículas adesivas como óleos, pois podem causar mau funcionamento ao aderirem na parte óptica do sensor (SAMYOUNG, 2023).

Por fim, o sensor DSM501A demora cerca de um minuto para seu funcionamento pleno (SAMYOUNG, 2023), seu custo pode variar entre R\$40 a R\$70 e apresenta boa leitura de valores, desde que as calibrações e condições de ambientes sejam adequadas para seu funcionamento.

4.3.5 Sensor de temperatura e umidade DHT22

O sensor DHT22 mede temperatura e umidade em um único dispositivo, utiliza um sensor de umidade capacitivo e um termistor para medir o ar circundante, o resultado da medição é um sinal digital (Aosong, 2023).

Mede temperaturas entre -40 °C e 80 °C e umidade do ar nas faixas de 0 a 100%, com uma precisão que varia entre 2 e 5%. Possui tamanho reduzido, baixo consumo energético e transmissão de dados de até 100 metros, opera com tensão de alimentação entre 3.3-6V e seu intervalo de leituras é de aproximadamente cada 2 segundos (Aosong, 2023).

Vapor de materiais químicos e exposição do sensor a luz ultravioleta podem reduzir a

capacidade de medição do sensor. É recomendado que o sensor fique longe de dispositivos que emitam calor, pois podem influenciar na medição da umidade relativa do ar (Aosong, 2023). Na Figura 17 apresenta-se o sensor DHT22.

Figura 17 – Sensor de partículas DSM591A (a) e seu diagrama de blocos (b).



Fonte: Retirado de (Aosong, 2023).

O sensor DHT22 foi adicionado ao projeto para medir temperatura e umidade relativa do ar, pois apresenta custo entre R\$20 a R\$35, sua programação é simples e pode ser aplicado em vários ambientes internos e externos e fornece um bom nível de precisão em suas medições.

4.4 CONEXÃO DE REDE WIFI E PROTOCOLOS DE COMUNICAÇÃO

Nesta sessão serão apresentados os aspectos de comunicação entre os serviços em nuvem utilizados e a camada física do projeto. Será descrito o método de comunicação via WiFi entre o microcontrolador ESP8266 e o serviço Azure IoT Hub, abordando a utilização dos protocolos TCP/IP e MQTT, apresentando suas principais características.

4.4.1 Comunicação WiFi via Protocolos TCP/IP

O padrão IEEE 802.11 para redes locais sem fio WiFi, é uma tecnologia consolidada, com mais de 20 anos de desenvolvimento e padronização. Sua versão mais antiga foi realizada em 1997 como uma alternativa sem fio para redes LANs (*Local Area Network*) com fios existentes na época usando a tecnologia Ethernet. Atualmente as WLANs não são apenas a tecnologia mais comum de acesso à internet como também se expandiram em uma ampla variedade de mercados, como indústria automotiva, segurança e monitoramento, IoT e usuários comuns (Bellalta *et al.*, 2016).

Muitos fatores contribuíram para o sucesso da família de padrões IEEE 802.11, dentre eles a interoperabilidade, facilidade de uso e flexibilidade são as que mais se destacam. Além disso, o baixo custo de implementação de uma rede WiFi usando IEEE 802.11 tornam a implementação de projetos de sistemas baseados em IoT mais acessíveis.

Para suportar ambientes onde o ruído pode ser elevado em determinados momentos, a especificação IEEE 802.11b determina a troca da taxa de transmissão, dinamicamente, dependendo das condições do sinal, sendo essa troca transparente às camadas superiores do protocolo. As velocidades possíveis são: 11Mbps, 5,5 Mbps, 2 Mbps e 1Mbps (IEEE, 2021).

Segundo O'Hara and Patrick (2005), o padrão IEEE 802.11 define uma arquitetura para as redes sem fio, baseada na divisão da área coberta pela rede em células denominadas BSA (*Basic Service Area*). As dimensões da BSA dependem das características do ambiente e da potência dos transmissores/receptores utilizados no ambiente.

Aliado ao padrão IEEE 802.11 temos também o conjunto de protocolos TCP/IP, que possibilita a conexão entre dispositivos e servidores através de rede local ou Internet. O TCP (*Transmission Control Protocol*) é um dos protocolos pelos quais a internet funciona, utilizado em quase todas as redes de transmissão de dados graças a sua capacidade de verificar se os dados são enviados de, na sequência correta e sem erros pela rede (Kanmai, 2020).

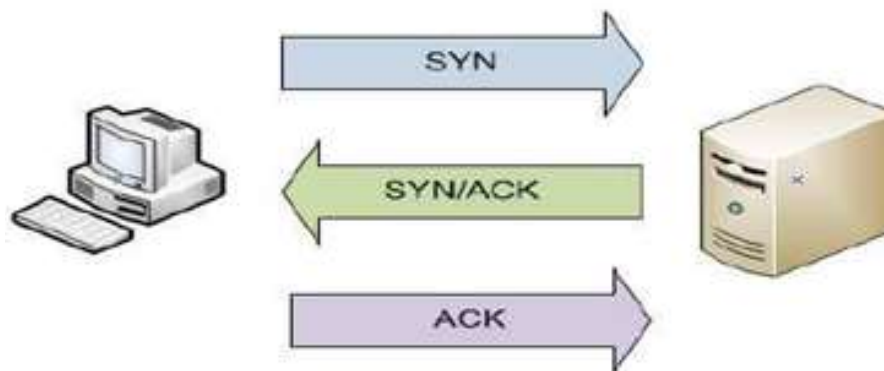
O protocolo TCP determina três fases durante uma conexão: estabelecimento da conexão, transferência e término da conexão. Em uma ligação TCP, existem os elementos denominados cliente e servidor, onde o cliente inicia uma conexão enviando um pacote SYM (*Synchronize*) TCP com um sinalizador ativo e aguarda o servidor permitir a conexão que, envia um pacote de confirmação chamado SYM+ACK (*Acknowledgement*).

Se durante um período este pacote não for recebido ocorre uma pausa e o pacote SYM é reenviado. Este processo é chamado de *Handshake* (aperto de mão), onde ocorre o reconhecimento entre os dispositivos para realizar a comunicação entre eles.

A fase de encerramento da sessão TCP ocorre em quatro fases, onde cada elemento na conexão é responsável pelo encerramento do seu lado da conexão. Quando um elemento decide encerrar a conexão, envia um pacote FIN (*Finalize*) e deve receber um pacote ACK para confirmar o encerramento (Kanmai, 2020). Na Figura 18 demonstra-se o processo de *Handshake*.

Além disso, cada dispositivo conectado à rede é identificado com um endereço virtual chamado IP (*Internet Protocol*). A camada Internet é responsável por adicionar um cabeçalho ao pacote de dados recebidos da camada Transporte, além do endereço IP de

Figura 18 – Processo de *Handshake* do protocolo TCP.



Fonte: Adaptado de Kanmai (2020).

origem e destino do pacote (Tanenbaum, 2021).

O endereçamento de dispositivos conectados a rede por meio de um IP permite que cada dispositivo possua um endereço diferente, que pode ser fixo ou atribuído dinamicamente. Isso faz com que diferentes dispositivos estejam conectados a mesma rede local e, ao mesmo tempo, a internet. Além disso, o protocolo IP garante que os pacotes de dados do endereço de origem sejam encaminhados ao endereço destino (Tanenbaum, 2021).

O dispositivo IoT utiliza configuração de IP fixo para a conexão com o roteador WiFi, evitando conflitos entre outros dispositivos que possam se conectar a rede.

Neste projeto o padrão IEEE 802.11 foi utilizado para conectar o dispositivo IoT através de rede WiFi a um roteador TP-Link AX3000 conectado a Internet. O protocolo TCP permite que o dispositivo IoT se conecte com o serviço em nuvem Azure IoT Hub e envie os dados para autenticação do dispositivo no serviço em nuvem.

Realizada a autenticação no serviço em nuvem, o protocolo TCP mantém a conexão necessária para que o dispositivo IoT possa enviar os dados de telemetria através da Internet usando o protocolo MQTT.

O Protocolo IP é o responsável por endereçar os pacotes que serão enviados com informações do endereço do emissor (dispositivo IoT) e receptor dos dados (serviço em nuvem) através do roteador, que redireciona estes pacotes através da internet.

4.4.2 Protocolo MQTT

A evolução no uso de redes sem fio e os avanços tecnológicos na produção de eletroeletrônicos inteligentes, que possuem capacidade de serem configurados de acordo com necessidades específicas, facilitaram o monitoramento e acionamento remoto de tais

dispositivos, formando um componente integrado da Internet das Coisas (IoT).

Em projetos baseados em IoT, a transferência de mensagens entre os dispositivos é extremamente importante, pois as informações presentes em um dispositivo podem ser utilizadas para gerenciar outros dispositivos inter-conectados (Soni; Makwana, 2017).

Um dos protocolos-base para a comunicação entre dispositivos IoT é o MQTT (*Message Queuing Telemetry Transport*), um protocolo de transporte de dados leve e eficiente na entrega de mensagens instantâneas, planejado para funcionar em equipamentos com baixo consumo energético (Hwang; Park; Shon, 2016).

O protocolo MQTT é um padrão de *publish/subscribe* (publicante/assinante), lançado pela IBM (*International Business Machines*) em 1999 com o intuito de enviar dados de forma confiável em redes com baixa largura de banda e atrasos nas conexões, trabalhando em conjunto com os protocolos TCP/IP (ibm.com, 2023).

Seu funcionamento baseia-se no conceito de *publish/subscribe*, onde um ou diversos publicantes enviam mensagens consumidas pelos assinantes. Os assinantes, por sua vez, se inscrevem em tópicos específicos relacionados a eles, o que lhes permite consumir todas as mensagens publicadas em um tópico. O elemento responsável por gerenciar o recebimento de mensagens enviadas pelos publicantes e redirecioná-las aos assinantes correspondentes é chamado de *broker* (intermediário) (Soni; Makwana, 2017).

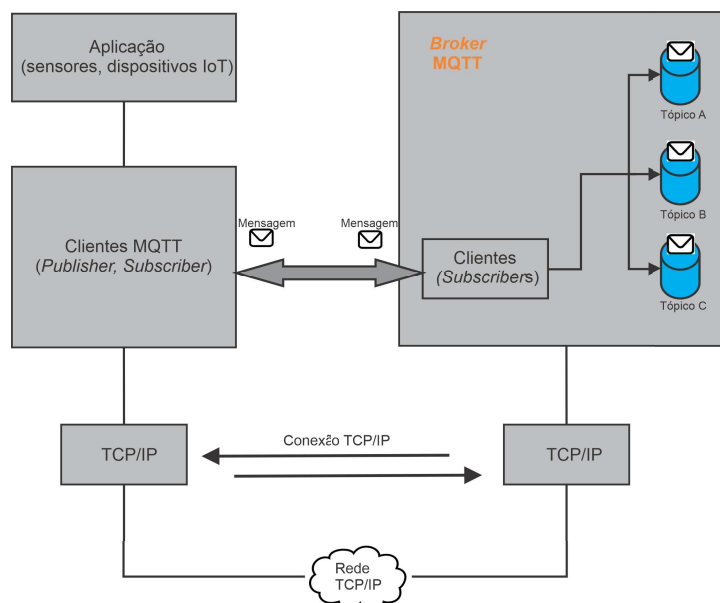
O *broker* é responsável por receber todas as mensagens e controlar a distribuição das informações para os *subscribers* segundo as inscrições em um tópico. Além disso, o *broker* também é responsável por: processar requisições de diferentes *subscribers* e cancelar inscrições em tópicos. Um *subscriber* só consegue visualizar as informações presentes nos tópicos ao qual está inscrito.

As mensagens retidas pelo *broker* MQTT são redirecionadas aos assinantes de um tópico, quando há um novo assinante para um tópico as mensagens são retransmitidas ao novo assinante. Um cliente MQTT pode ser um publicante ou assinante e esta configuração é estabelecida no método de conexão com o servidor (*broker*) (ibm.com, 2023). Na Figura 19 ilustra-se a colaboração entre os protocolos MQTT e TCP/IP.

A conexão entre um *publisher* e o *broker* MQTT é realizada em 4 fases:

1. **Estabelecer conexão:** a conexão do cliente com o servidor (*broker*) MQTT através do TCP/IP é realizada pela troca de pacotes de controle entre o cliente e o servidor. O cliente que deseja se conectar com o MQTT envia um pacote do tipo *CONNECT* com seus dados de identificação, entre outras informações, se o servidor reconhecer o cliente é retornado um pacote *CONNACK* indicando o status da conexão (Kodali; Soratkal, 2016).

Figura 19 – Colaboração entre os protocolos MQTT e TCP/IP.



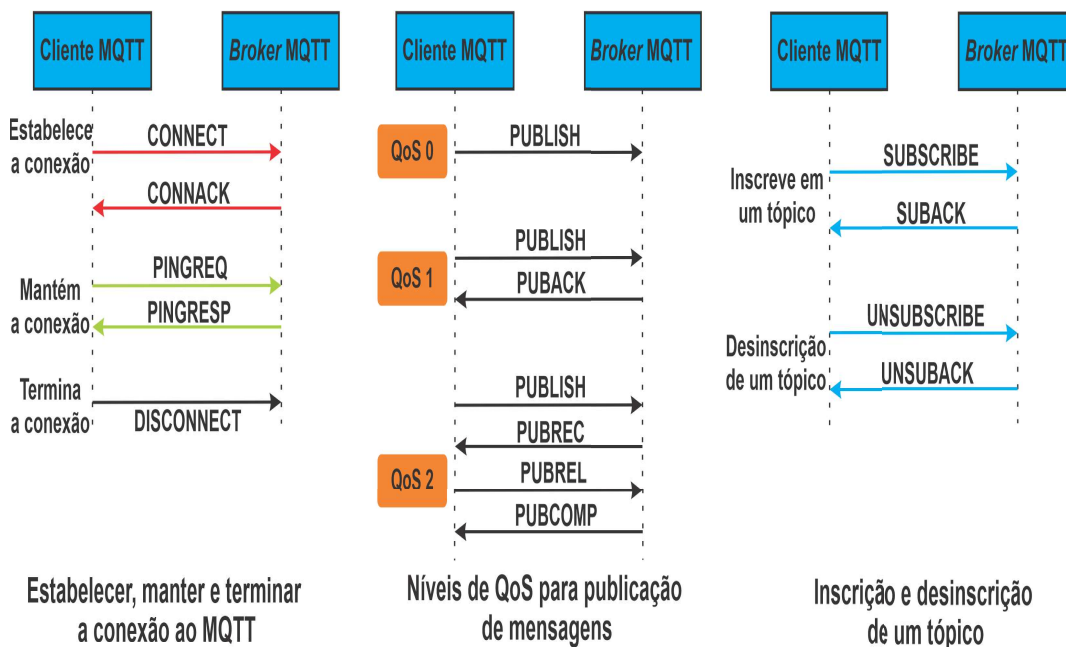
Fonte: Adaptado de Soni and Makwana (2017).

2. **Publicar as mensagens:** se o cliente deseja publicar uma mensagem deve ser enviado um pacote do tipo *PUBLISH* para o servidor. MQTT suporta 3 níveis de *Quality of Service* (QoS) , se a troca de mensagens for realizada no QoS 0, o cliente não recebe nenhum pacote de confirmação. Para o nível QoS 1 o servidor reconhece o pacote do cliente retornando um pacote do tipo *PUBACK*. No nível QoS 2 quatro pacotes são trocados: o servidor reconhece o pacote *PUBLISH* e retorna um pacote *PUBREC*. O cliente então envia um pacote para publicação dos dados *PUBREL* e o servidor retorna um quarto pacote *PUBCOMP* indicando que a publicação da informação no tópico foi concluída (Kodali; Soratkal, 2016).
3. **Inscrição a um tópico:** se um cliente do tipo *subscriber* deseja assinar um tópico MQTT, ele envia um pacote *SUBSCRIBE* contendo o nome do tópico indicado em codificação UTF-8 (*8-bit Unicode Transformation Format*) . Se o servidor reconhecer a assinatura é retornado um pacote *SUBACK* indicando o status da solicitação, assim que a inscrição for realizada serão encaminhadas as mensagens recebidas no tópico com QoS 2. Para cancelar a assinatura de um tópico, o cliente envia um pacote *UNSUBSCRIBE* ao servidor que o reconhece e retorna um pacote *UNSUBACK* (Kodali; Soratkal, 2016).
4. **Manutenção da conexão:** após um tempo, a conexão entre o cliente e o servidor é encerrada. Se o cliente quiser manter a conexão, deve indicar que está ativo enviando

um pacote do tipo *PINGREQ* para o servidor, que por sua vez retorna um pacote do tipo *PINGRESP* para manter a conexão ativa.

Na Figura 20 ilustram-se os métodos de conexão, publicação e inscrição do MQTT.

Figura 20 – Funcionamento do MQTT.



Fonte: Adaptado de Kodali and Soratkal (2016).

O protocolo MQTT é utilizado no projeto para enviar os dados de telemetria lidos dos sensores pelo ESP8266 para o serviço em nuvem Azure IoT Hub. O nível de QoS 1 é oferecido automaticamente pelo serviço em nuvem, podendo ser alterado nas configurações através do *framework* fornecido pelo próprio serviço em nuvem.

Na próxima subseção é descrito as bibliotecas presentes no *framework* fornecido pela Microsoft para conexão ao serviço IoT Hub.

4.4.3 SDK IoT Hub Client ESP8266 NodeMCU

A escolha da linguagem de programação usada para configurar o *hardware* de dispositivos IoT é uma decisão importante para o desempenho e funcionamento do projeto.

Segundo Ferreira (2021), existem várias configurações de dispositivos empregados em projetos IoT, havendo muitas variáveis que influenciam no tempo de processamento e, um projeto de IoT deve considerar estes aspectos.

Linguagens de baixo nível como *Assembly* são utilizadas na programação de microcontroladores por meio de ferramentas de *software*, muitas vezes disponibilizadas

pelos fabricantes dos microchips, onde o foco principal é o acesso direto ao *hardware* através da linguagem de programação. Porém, cada microprocessador possui um conjunto de instruções específicas que devem ser conhecidas pelo programador para extrair o melhor desempenho do microcontrolador.

Linguagens de programação como C/C++, consideradas de nível médio, fornecem acesso direto a alguns recursos de *hardware* (como acesso a memória) e, ao mesmo tempo, possui uma padronização de escrita de código-fonte mais próxima à linguagem humana, facilitando a construção de bibliotecas que podem ser utilizadas em diferentes dispositivos de *hardware*, mas que ainda demandam um bom conhecimento em estruturas de dados e lógica de programação.

Já as linguagens de alto nível como Python, C#, Java, JavaScript entre outras em sua maioria possuem escrita de código-fonte simplificado, possuindo funções e métodos prontos para a realização da maioria da lógica necessária na construção de códigos-fonte. Estas linguagens quando aplicadas na programação de dispositivos de *hardware* específicos são escritas usando ferramentas de tradução, que convertem o código escrito em alto nível para um código correspondente em baixo nível antes de serem embarcadas no dispositivo.

A escolha da linguagem de programação usada na criação do código-fonte da implementação física de um projeto de IoT deve levar em consideração as funcionalidades que ela oferece, o desempenho que se espera conseguir, a compatibilidade com os serviços que serão agregados ao projeto e a afinidade do programador com a linguagem escolhida.

O serviço em nuvem Azure fornece uma série de pacotes de desenvolvimento de *software* chamados SDK (*Software Development Kit*) para os diversos serviços disponibilizados. Para o serviço IoT Hub os SDKs podem ser encontrados nas linguagens Python, C#, Java, Node Js ou C (learn.microsoft.com, 2023h).

Neste projeto foi utilizado o SDK IoT Hub *Client* ESP8266 NodeMCU, fornecido pela própria Microsoft para comunicação entre o microcontrolador e o serviço Hub IoT. Este SDK fornece os métodos de configuração e bibliotecas necessárias para conectar o ESP8266 ao serviço Azure IoT Hub utilizando quatro bibliotecas: *azure-iot*, *azure-iot-pnp*, *azure-iot-dps* e *azure-iot-hub*, todas construídas em linguagem C.

O SDK fornece os métodos necessários para conexão ao serviço em nuvem, autenticação, configuração dos dados de *payload* necessário para envio dos dados e envio dos dados de telemetria usando o protocolo MQTT. As bibliotecas fornecidas auxiliam nos processos de comunicação com o serviço em nuvem, ficando a cargo do programador elaborar os códigos-fonte para implementar a lógica necessária do dispositivo IoT.

Os pré-requisitos necessários para a comunicação entre o dispositivo físico e o serviço

em nuvem são:

- Ter uma conta Azure;
- Possuir um IoT Hub criado;
- Ter um "dispositivo lógico" criado no IoT Hub usando autenticação do tipo chave simétrica.

Para a utilização do SDK na programação do dispositivo físico é necessário:

- Ter uma Interface de programação do Arduino atualizada;
- Suporte a programação do ESP8266 configurado na ferramenta de programação Arduino.

É importante ressaltar que o SDK fornece somente as funções de comunicação com o serviço em nuvem, todo o processo de configuração do ESP8266 para calibração dos sensores e leitura dos dados de telemetria deve ser realizado conforme a lógica necessária para atender as necessidades do projeto.

4.4.4 Considerações finais sobre o capítulo

Neste capítulo foram apresentadas todas as ferramentas utilizadas no desenvolvimento do projeto de medição da qualidade do ar em ambientes internos.

Foram descritos os serviços em nuvem Azure IoT Hub, IoT Central e Blob Storage utilizados para receber, armazenar e disponibilizar os dados de telemetria para visualização através da internet.

Apresentaram-se os componentes físicos que compõe o projeto como o microcontrolador ESP8266 e sensores DSM501A, DHT22 e MICS6814. Uma breve descrição do funcionamento dos protocolos de comunicação TCP/IP e MQTT também foi realizada.

Por fim foram descritas as bibliotecas disponibilizadas pelo serviço em nuvem Azure para comunicação entre o serviço e o microcontrolador.

No próximo capítulo serão descritas as implementações realizadas para interligar as camadas do projeto e comunicação entre os serviços em nuvem.

5 METODOLOGIA

Atualmente, o monitoramento e acionamento de equipamentos eletro-eletrônicos por meio de dispositivos inteligentes pode ser considerado como um avanço tecnológico que estará presente no cotidiano de pessoas e organizações. O uso de controles remotos tradicionais pode ser considerado um exemplo de monitoramento e acionamento manual de dispositivos como ventiladores, ar-condicionados, janelas automatizadas, etc.

O monitoramento adequado das condições climáticas e de qualidade do ar de ambientes internos pode fornecer dados a serem utilizados para acionar remotamente e de maneira automática equipamentos eletrônicos, visando manter as condições do ambiente monitorado as melhores possíveis para os indivíduos ali presentes.

Integrar dispositivos IoT com serviços em nuvem para proporcionar o monitoramento de ambientes internos tem se mostrado uma alternativa interessante quando se visa melhorar as condições de qualidade do ar.

Segundo Ferreira (2021), o uso de serviços em nuvem para processar dados da IoT tem se tornado comum devido os serviços em nuvem disponibilizarem grande capacidade de armazenamento de dados, possibilidade de utilização de vários núcleos de processamento e compartilhamento de informações em tempo real.

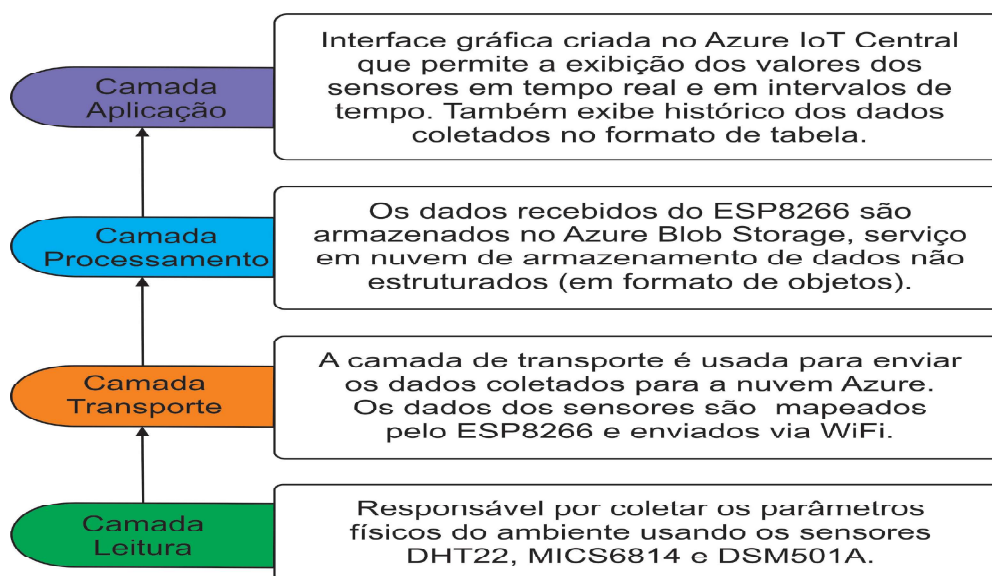
Dispositivos IoT fornecem os recursos de uma solução discreta e de baixo custo, adicionando-os como parte de uma rede, podem interagir com o ambiente sem a necessidade da interação humana. Estes sistemas podem realizar o monitoramento e controle da qualidade do ar interno em tempo real baseando-se em ações pré-definidas em seu código; permitindo que ações preventivas sejam tomadas, como acionar a abertura de uma janela diretamente ou fornecer notificações aos ocupantes para abrir uma janela (Wall *et al.*, 2021).

Na seção 5.1 serão apresentadas as configurações dos sensores e programação do microcontrolador ESP8266, na seção 5.2 a implementação dos serviços em nuvem e a comunicação entre os sensores e o microcontrolador.

5.1 IMPLEMENTAÇÃO FÍSICA

Este projeto consiste na utilização do microcontrolador ESP8266 (Espressif, 2023) conectado a Internet via rede local WiFi para a leitura e envio dos dados coletados pelos sensores MICS6814 (pewatron.com, 2022), DHT22 (Aosong, 2023) e DSM501A (SAMYOUNG, 2023) para o serviço em nuvem Azure. Na Figura 21 apresenta-se as quatro camadas da arquitetura implementada e os detalhes de cada camada.

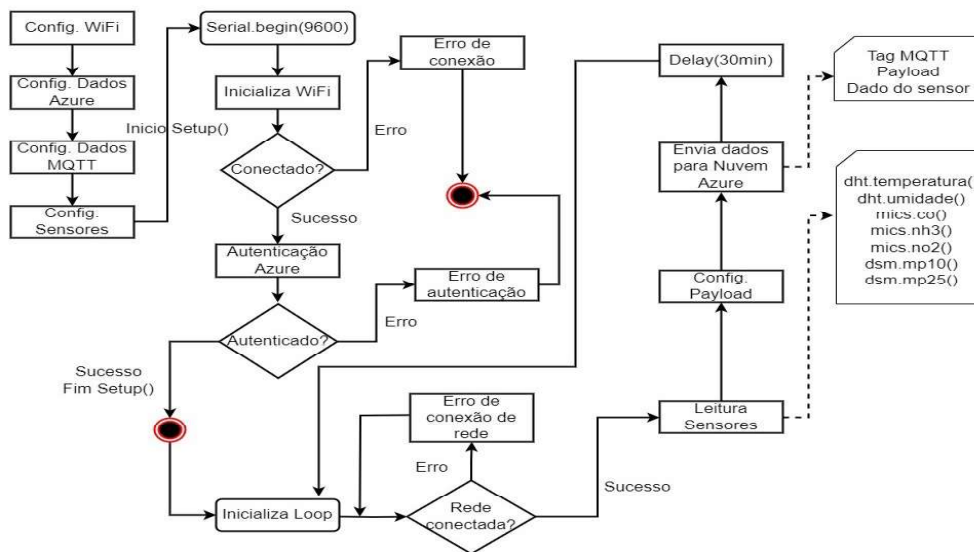
Figura 21 – Camadas do projeto de medição da qualidade do ar.



Fonte: Elaborado pelo autor.

O código-fonte gravado no microcontrolador ESP8266 foi escrito em C para Arduino combinando as bibliotecas fornecidas pelo serviço Azure com as funções de conexão a rede WiFi, configuração dos pinos de entrada e saída, configuração para envio dos dados usando protocolo MQTT, configuração e leitura dos dados do sensor MICS6814 usando o multiplexador CD74HC4067, leitura dos sensores DHT22 e DSM591A e por fim envio dos dados coletados para o serviço em nuvem IoT Hub. Na Figura 22 apresenta-se a estrutura do código-fonte gravado no microcontrolador ESP8266.

O primeiro passo realizado no código-fonte gravado no ESP8266 é a configuração da conexão com a rede WiFi, onde é utilizada a biblioteca em C **ESP8266WiFi.h** indicada pelo próprio fabricante (Espressif, 2023). Devem ser inseridos no código o nome da rede WiFi que se deseja conectar, senha para autenticação e endereço IP fixo. É possível obter um endereço IP dinâmico caso o roteador WiFi esteja configurado para atribuir endereços IP dinamicamente, o que ocorre na maioria das redes WiFi domésticas ou de pequeno porte.

Figura 22 – Estrutura do código-fonte gravado no ESP8266.

Fonte: Elaborado pelo autor.

Para a autenticação no serviço em nuvem devem ser especificados os seguintes parâmetros apresentados no Quadro 7.

Quadro 7 – Parâmetros necessários para autenticação no serviço em nuvem.

Parâmetro	Definição
SCOPE_ID	Identificador de escopo do serviço em nuvem que define a qual dispositivo lógico dentro do serviço IoT Hub os dados serão enviados. Este valor é gerado pelo serviço em nuvem na criação de um dispositivo lógico no IoT Hub.
DEVICE_ID	Identificador do dispositivo. Este valor é atribuído pelo usuário do serviço em nuvem na criação do dispositivo lógico.
DEVICE_KEY	Chave simétrica para comunicação entre o dispositivo físico e o serviço em nuvem. Fornecido pelo serviço em nuvem.

Fonte: Elaborado pelo Autor.

Após configurar os dados de conexão WiFi, é configurada a conexão do protocolo MQTT através de uma função que verifica o *status* da conexão com o serviço em nuvem e exibe uma mensagem, que pode ser vista pelo monitor serial da IDE Arduino e no serviço

em nuvem.

Em caso de sucesso na conexão é preparado o *payload* (conteúdo) que será enviado para o serviço em nuvem, o *payload* recebe valor zero toda vez que um dado novo é enviado e, após o envio, o serviço em nuvem retorna um valor denominado *callbackInfo* indicando se o valor foi recebido pelo serviço em nuvem ou não.

Por fim é realizada a configuração dos pinos de saída digital e entrada analógica do ESP8266 que serão utilizados pelo multiplexador analógico-digital CD74HC4067, utilizado para enviar os dados lidos das 3 saídas analógicas dos sensores MICS6814.

Após todas as configurações iniciais, na função *setup* é inicializada a conexão WiFi com o roteador de internet e após a conexão é realizada a autenticação do dispositivo no serviço em nuvem IoT Hub. Uma mensagem de erro é exibida se em algum destes processos houver alguma falha ou erro de conexão.

Na função *Loop*, o primeiro passo executado é a verificação do status da conexão antes de realizar a leitura dos valores e envio para a nuvem. Se houver conexão com a Internet, o próximo passo executado é a leitura dos dados dos sensores, iniciando pelo sensor MICS6814. Foi criada no código-fonte, uma função para ler os dados de cada saída analógica do sensor MICS6814 e armazená-las em uma estrutura do tipo vetor.

Após todos os dados lidos pelo multiplexador serem armazenados, o vetor é lido para que os dados sejam enviados ao ESP8266 que irá redirecionar para o serviço em nuvem Microsoft Azure IoT Hub. No Algoritmo 1 apresenta-se o código-fonte para configuração do multiplexador e leitura dos dados do sensor MISC6814.

```

1 para ( $i = 0$  ;  $i \leq 2$ ;  $i++$ ) faça
2   |   para ( $j = 0$ ;  $j \leq 2$ ;  $j++$ ) faça
3   |   |   escritaDigital(mix[j], ci[i][j]);
4   |   fim
5   |   val_analog[i] = leituraAnalogica(SIG) ;
6 fim

```

Algoritmo 1: Função de configuração e uso do multiplexador CD74HC4067

São utilizados dois laços de repetição, o primeiro para gerar os índices do vetor onde serão armazenados os dados do sensor e o segundo (interno) para gerar os endereços de saídas de sinal digital que indicam qual entrada analógica do multiplexador será lida.

A calibração do sensor MICS6814 é feita pelo código-fonte em C (Arduino), onde uma função *map()* é usada para especificar os valores mínimos e máximos de cada gás lido suportados pelo sensor e calcular o valor de referência em partículas por milhão (ppm).

Cada um dos gases lidos possui um valor mínimo e máximo que pode ser identificado conforme o Quadro 6. Após a leitura, o sensor retorna um valor entre 0 e 1023 (escala de

leitura do gás). Este valor deve ser remapeado para se obter o valor correspondente em ppm. O cálculo realizado pela função *map()* pode ser observado na Equação 1.

$$(x - in_min) \times (out_max - out_min) \div (in_max - in_min) + out_min \quad (1)$$

onde:

- x: valor lido;
- in_min: valor mínimo do intervalo de entrada;
- in_max: valor máximo do intervalo de entrada;
- out_min: valor mínimo a ser remapeado;
- out_max: valor máximo a ser remapeado;

Nas Equações 2,3 e 4 são apresentados os mapeamentos de valores realizados pela função *map* segundo as especificações presentes no *datasheet* do sensor:

$$CO = map(val_analog[i], 0, 1023, 1, 1000) \quad (2)$$

$$NO2 = map(val_analog[i], 0, 1023, 0.05, 10) \quad (3)$$

$$NH3 = map(val_analog[i], 0, 1023, 1, 300) \quad (4)$$

Finalizada a leitura dos dados do sensor MICS6814, são lidos os dados do sensor DHT22 para temperatura e umidade relativa do ar. A biblioteca **DHT.h** foi utilizada para realizar a leitura através das funções *readTemperature()* e *readHumidity()*, que retornam valores no formato decimal.

Por fim, são lidos os dados do sensor DSM501A para partículas em suspensão PM10 e PM2.5. A biblioteca **DSM501.h** foi empregada na leitura fornecendo os valores em mg/m³, convertidos em ppm antes de serem enviados para a nuvem. A fórmula usada para a conversão considerou as partículas em suspensão como sendo compostos orgânicos voláteis(COV), como descrita na Equação 5:

$$ppm = valor_lido \div 78,9516 \quad (5)$$

Após obter todos os dados de telemetria necessários, é realizado o envio para o serviço em nuvem. Para cada dado de telemetria é necessário criar uma *tag* MQTT equivalente

ao valor que será enviado. O tamanho do valor do *payload* é configurado e são enviados a nuvem a *tag* MQTT, os dados de *payload* e o valor de telemetria correspondente, se os dados forem recebidos pelo serviço em nuvem é ativada uma espera (*delay*) de 30 minutos até a próxima leitura, em caso negativo é ativada uma espera de um minuto e então todo o processo é repetido. No Algoritmo 2 demonstra-se o método de envio dos dados para o serviço em nuvem.

```

1 se (conectado()) então
2   | payload [64]  $\leftarrow$  {0}
3   | multiplexGases()
4   | umi = dht.lerUmidade()
5   | temp = dht.lerTemperatura()
6   | pm25 = dsm.lerPM25()
7   | pm10 = dsm.lerPM10()
8   | enviaDados(payload, \“CO”\, val_analog[0])
9   | enviaDados(payload, \“NO2”\, val_analog[1])
10  | enviaDados(payload, \“NO3”\, val_analog[2])
11  | enviaDados(payload, \“Temperatura”\, temp)
12  | enviaDados(payload, \“Umidade”\, umi)
13  | enviaDados(payload, \“PM25”\, pm25)
14  | enviaDados(payload, \“PM10”\, pm10)
15  se (erro!=0) então
16    | LOG_ERROR(“Erro ao enviar mensagem”,errorCode)
17    | espera(60000)
18  fim
19  senão
20    | MSG(“Envio concluído”)
21    | espera(1800000)
22  fim
23 fim

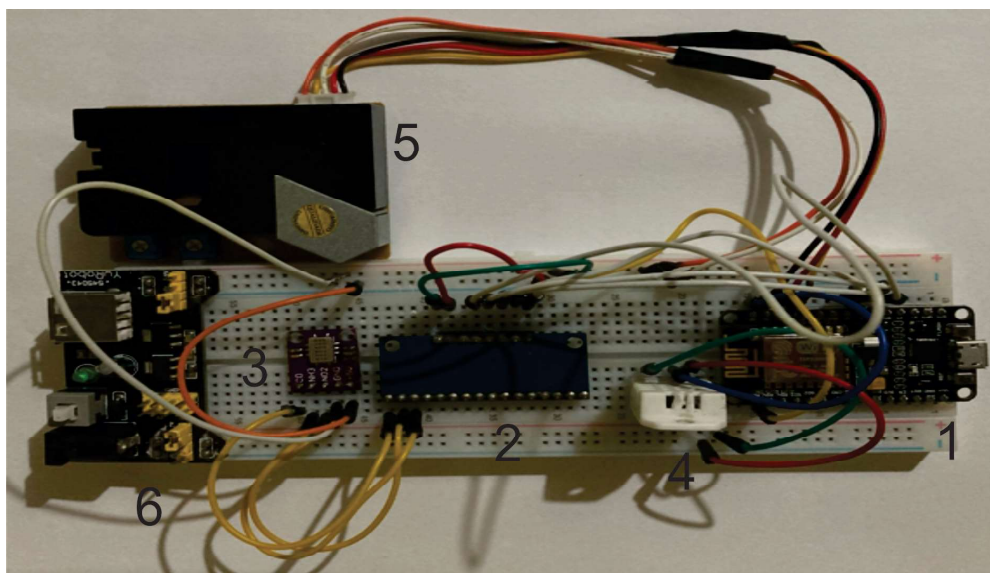
```

Algoritmo 2: Envio de dados para a nuvem Azure

O intervalo de tempo de envio de dados pelo ESP8266 pode ser configurado para atender a necessidade da aplicação e o tempo de recebimento do dado pela nuvem depende da velocidade de conexão a internet. Em teste realizado com conexão a Internet de 600 Mb o tempo de envio dos dados de telemetria pelo ESP8266 e recebimento do serviço em nuvem foi de em média 10 segundos. Nas Figuras 23 e 24 é apresentada a interligação dos componentes físicos do projeto e um esquema elaborado no *software* Fritzing.

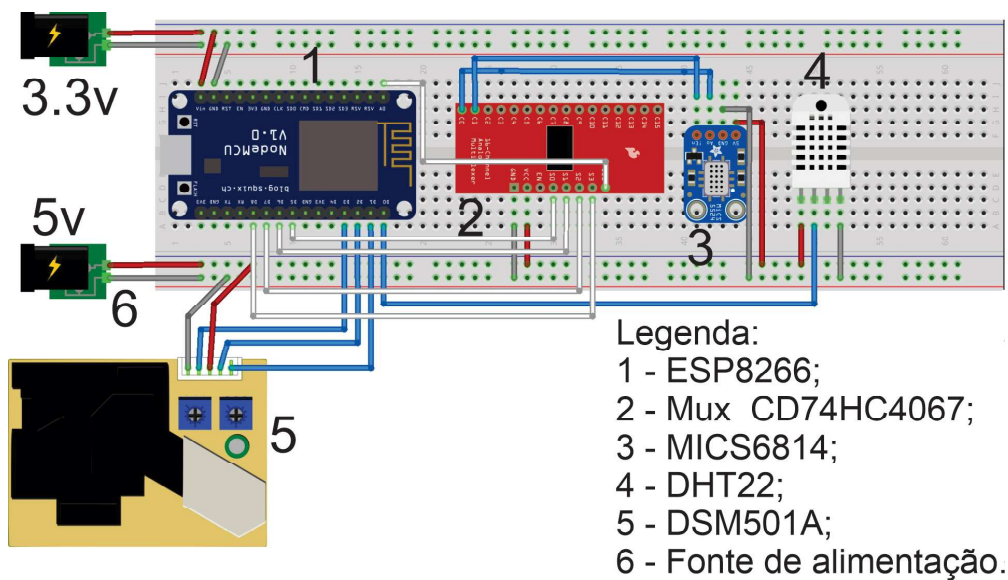
A transferência de dados de telemetria entre o ESP8266 e o IoT Hub utiliza alguns comportamentos do protocolo MQTT, uma vez que o serviço Azure não possui suporte a todos os comportamentos (learn.microsoft.com, 2023a). Nesta aplicação, o MQTT é utilizado para transformar o ESP8266 em um *publisher* (cliente) MQTT para fazer o

Figura 23 – Integração física dos componentes.



Fonte: Elaborado pelo autor.

Figura 24 – Integração física dos componentes elaborado no *software* Fritzing.



Fonte: Elaborado pelo autor.

envio dos dados coletados pelos sensores via tópicos MQTT ao Hub IoT.

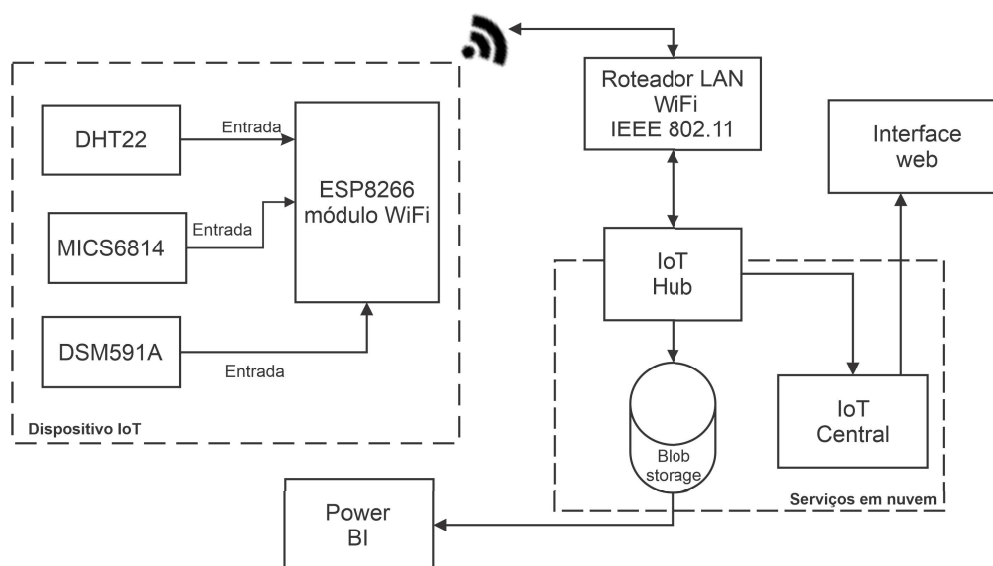
Nesta seção foram abordados os aspectos físicos e de envio de dados para o serviço em nuvem IoT Hub, na próxima seção serão apresentados as configurações realizadas entre os serviços em nuvem Azure, bem como a comunicação entre eles.

5.2 CONFIGURAÇÃO DOS SERVIÇOS EM NUVEM

Os serviços em nuvem Microsoft Azure foram utilizados no projeto para proporcionar uma interface simples e de fácil acesso para armazenamento e visualização dos dados coletados pelo dispositivo IoT. Com o uso dos serviços IoT Hub, IoT Central e Blob Storage foi possível receber os dados do dispositivo IoT de maneira segura, armazenar os dados em um contêiner no formato de objetos, visualizar e elaborar gráficos com análises temporais.

Na Figura 25 ilustra-se a comunicação entre o dispositivo IoT e os serviços em nuvem.

Figura 25 – Integração entre o dispositivo IoT e os serviços em nuvem



Fonte: Elaborado pelo autor.

Todos os serviços utilizados dentro da nuvem Azure devem ser vinculados a um grupo de recursos, que agrega todos os recursos em uma única interface e é criado facilmente na página inicial do Azure.

5.2.1 Configuração IoT Hub

O primeiro serviço criado um novo serviço IoT Hub, que é vinculado a conta Azure e atua como gerenciador de dispositivos IoT e *broker MQTT*.

As configurações especificadas na criação do *hub* determinam sua capacidade de troca de mensagens e custos. O acesso remoto ao IoT Hub pelos dispositivos IoT podem ser por rede pública ou privada, em caso de acesso privado pode ocorrer cobrança adicional, já que o serviço Azure criará uma VPN *Virtual Private Network* para a comunicação entre as duas extremidades da rede.

Para este projeto todas as configurações foram realizadas visando o menor custo possível, por isso o nível de camada, que define a quantidade de mensagens trocadas entre o dispositivo IoT e o serviço em nuvem, foi definido como gratuito, possibilitando o consumo 8.000 mensagens por dia. Mensagens enviadas do dispositivo IoT para a nuvem e vice-versa são contabilizadas igualmente. Na Figura 26 apresenta-se o processo configuração de um novo IoT Hub.

Figura 26 – Configuração para criação de um novo IoT Hub.

Microsoft Azure

Pesquisar recursos, serviços e documentos (G+)

Página inicial > Hub IoT >

Hub IoT

Microsoft

Básico Rede Gerenciamento Complementos Marcas Revisar + criar

Crie um hub IoT para ajudá-lo a se conectar, monitorar e gerenciar bilhões de ativos de IoT. [Saiba mais](#)

Detalhes do projeto

Escolha a assinatura que você usará para gerenciar as implantações e os custos. Use os grupos de recursos como pastas para ajudar você a organizar e gerenciar recursos.

Assinatura * ? AzureEduardoCruz

Grupo de recursos * ? AirStuff [Criar novo](#)

Detalhes da instância

Nome do hub IoT * ? hubAirstuff ✓

Região * ? East US

Camada * Gratuito

i A avaliação gratuita explora o aplicativo com dados dinâmicos. As avaliações não podem ser dimensionadas ou atualizadas posteriormente. [Comparar camadas](#)

Limite diário de mensagens * ? 8.000 (US\$ 0/mês)

Revisar + criar < Anterior Avançar: Rede >

Fonte: Elaborado pelo autor.

Após realizar todas as configurações básicas necessárias, basta realizar uma última revisão nas especificações e clicar no botão criar como mostrado na Figura 27.

A partir da criação, seu novo IoT Hub ficará disponível para acesso na página inicial da assinatura Azure. Dentro do IoT Hub é possível visualizar todas as configurações disponíveis, os custos e também monitorar o número de mensagens trocadas entre o serviço em nuvem e o Dispositivo IoT.

Figura 27 – Validação das configurações e criação do IoT Hub.

Microsoft Azure

Pesquisar recursos, serviços e documentos (G+ /)

Página inicial > Hub IoT >

Hub IoT

Microsoft

Básico Rede Gerenciamento Complementos Marcas **Revisar + criar**

Preços

Hub IoT

US\$ 0 USD
por mês
[Alterar noções básicas](#)

Total de complementos: [Alterar complementos](#)

Básico

Assinatura	AzureEduardoCruz
Grupo de recursos	AirStuff
Nome do hub IoT	hubAirstuff
Região	East US
Recuperação de desastre habilitada	Sim
Nível	Free
Limite diário de mensagens	8,000 (US\$ 0/mês)

Rede

Configuração de conectividade	Acesso público
Conexões de ponto de extremidade	Nenhum

Criar < Anterior: Marcas Avançar > [Opções de automação](#)

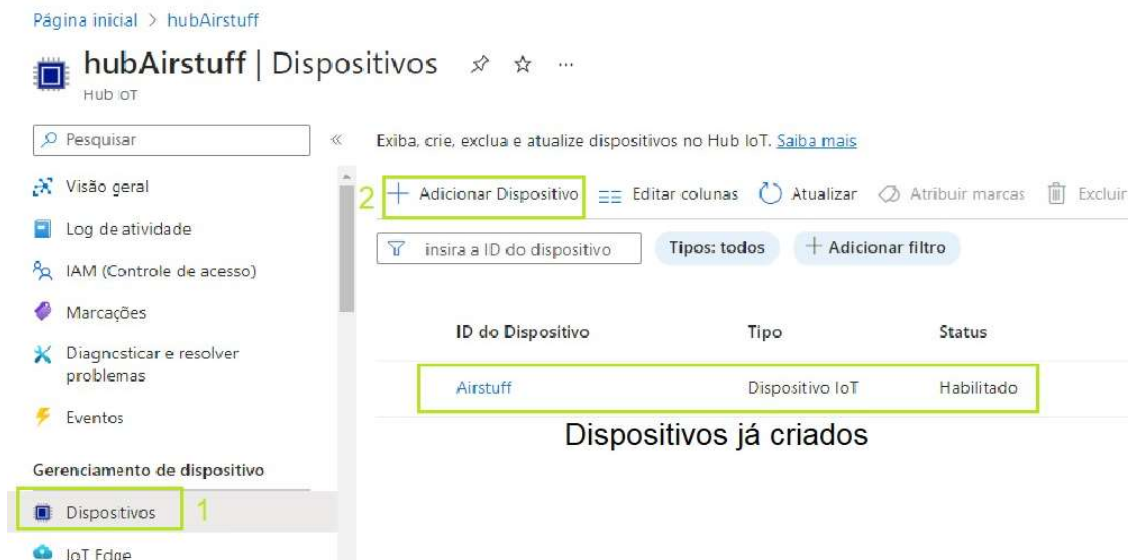
Fonte: Elaborado pelo autor.

Para se comunicar com um dispositivo físico é necessário criar um dispositivo lógico dentro do IoT Hub. Com seu IoT Hub aberto, acesse a opção dispositivos e logo em seguida clique na opção Adicionar Dispositivo como mostrado na Figura 28.

Na página de criação de novos dispositivos é necessário atribuir um ID para o dispositivo válido, selecionar o tipo de autenticação dentre as três formas disponibilizadas pelo serviço em nuvem. A opção chave simétrica é o método mais simples de ser utilizado, pois utiliza um par de chaves para autenticação onde o valor da chave é um código *hash* gerado por meio de um algoritmo de criptografia. As outras duas formas de autenticação utilizam certificados digitais para autenticação e nem todos os dispositivos físicos têm a capacidade de manipular tais certificados. Na Figura 29 demonstra-se a criação de um novo dispositivo lógico.

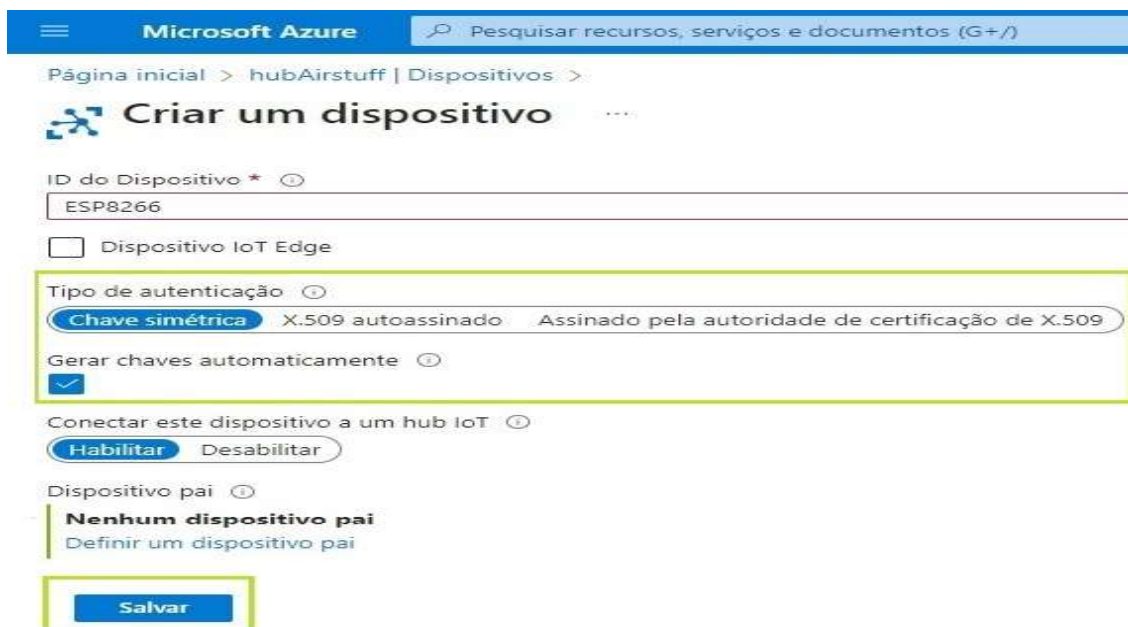
Após realizar todas as configurações para a criação de um novo dispositivo dentro do

Figura 28 – Criação de um novo dispositivo no IoT Hub.



Fonte: Elaborado pelo autor.

Figura 29 – Validação das configurações e criação do IoT Hub.



Fonte: Elaborado pelo autor.

IoT Hub, basta clicar no botão Salvar para finalizar o processo. Assim que criado, o novo dispositivo aparecerá na lista de dispositivos do IoT Hub.

Para iniciar a comunicação entre o dispositivo físico e o serviço em nuvem, ambos devem ter armazenados a mesma chave com código *hash*. Há as opções das chaves serem

geradas automaticamente pelo serviço em nuvem ou serem definidas manualmente pelo usuário do serviço.

Para acessar as chaves geradas pelo serviço em nuvem, basta clicar no seu dispositivo lógico dentro do IoT Hub. O Serviço oferece uma chave primária que é a chave principal de autenticação e uma chave secundária caso haja algum problema no uso da chave primária.

A chave do dispositivo lógico do IoT Hub deve ser armazenada no dispositivo físico, seu valor é utilizado como um dos parâmetros para realizar a autenticação do dispositivo IoT no serviço em nuvem. Na Figura 30 apresenta-se a página de acesso aos dados do dispositivo lógico para visualização da chave simétrica de conexão ao dispositivo IoT.

Figura 30 – Configuração de conexão do dispositivo IoT e Iot Hub.



Fonte: Elaborado pelo autor.

O serviço IoT Hub atua como *broker* (intermediário) MQTT, ficando responsável por coletar as mensagens enviadas pelo ESP8266, redirecionar os dados recebidos para o serviço Central de Aplicativos IoT, que atua como *subscriber* (assinante) MQTT, consumindo os dados de telemetria (hivemq, 2023).

5.2.2 Serviço de armazenamento Blob Storage

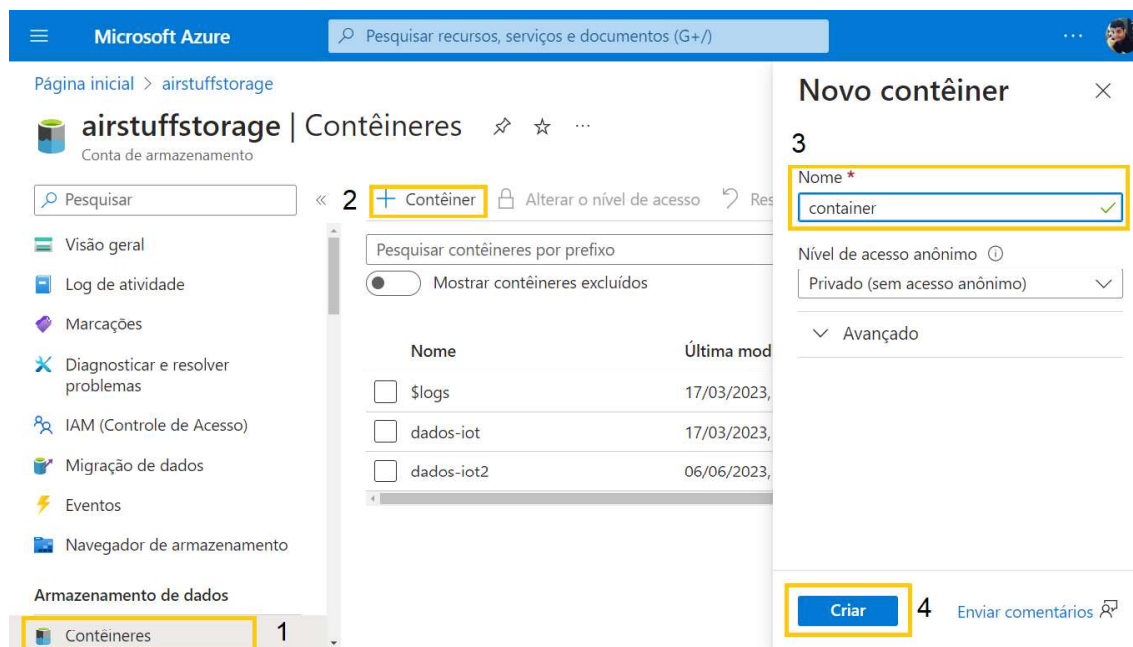
Para armazenar os dados recebidos pelo IoT Hub foi escolhido o serviço Blob Storage, um serviço que armazena dados e arquivos de forma não relacional e em diversos formatos. Para utilizar o serviço Blob Storage é necessário ter uma conta de armazenamento criada

vinculada a uma assinatura Azure. Na opção Contas de Armazenamento na página inicial do Azure é possível criar uma conta informando os seguintes dados:

- Nome da nova conta de armazenamento;
- Região que o serviço será implantado;
- Desempenho: são oferecidos a opção *Standard* (para aplicações tradicionais) ou *Premium* (aplicações que demandam baixa latência de rede);
- Redundância: define como os dados serão replicados para outras regiões disponíveis além da selecionada. Existem quatro níveis de redundância.

É necessário criar um contêiner para cada tipo de armazenamento (imagens, planilhas, arquivos, etc). Na Figura 31 apresenta-se o processo de criação de um novo contêiner.

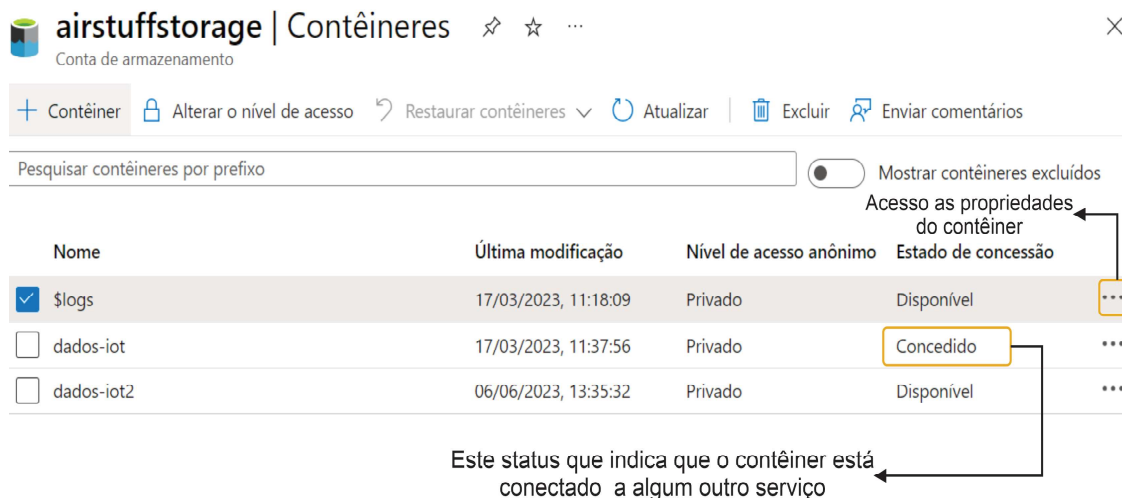
Figura 31 – Criação de um novo contêiner de armazenamento



Fonte: Elaborado pelo autor.

Os contêineres criados podem ser acessados clicando novamente na opção Contêineres. São exibidos dados relacionados a última modificação realizada no contêiner, suas propriedades, gerenciamento de acesso entre outras informações. Uma informação importante exibida é o Estado de concessão, que define se o contêiner está sendo utilizado por outro serviço ou não. Na Figura 32 demonstram-se as propriedades e o Estado de concessão dos contêineres criados em uma conta de armazenamento.

Figura 32 – Acesso aos contêineres vinculados a uma conta de armazenamento.



Fonte: Elaborado pelo autor.

O recebimento de dados de outros serviços a serem armazenados no Blob Storage deve ser configurado no serviço ao qual os dados são recebidos ou criados e cada serviço possui um método de conexão ao Blob Storage. Neste projeto foi configurado a comunicação com o serviço IoT Hub para armazenamento dos dados de telemetria do dispositivo IoT.

Para isso deve-se criar uma rota para um ponto de extremidade no serviço IoT Hub que será responsável por enviar os dados recebidos para um contêiner no Blob Storage ou outro serviço desejado. No IoT Hub, acesse a opção Roteamento de mensagens, em seguida clique em Adicionar para criar um novo ponto de extremidade, isso abrirá a página de configuração para adição de uma nova rota.

Neste passo devem ser configurar as seguintes opções de *Endpoint*:

- **Tipo de ponto de extremidade:** escolha a opção Armazenamento para acessar os contêineres do serviço Blob Storage. Clicar no botão **Escolha um contêiner** e selecionar o contêiner que deseja usar para armazenar os dados. Nesta opção será aberta as conta de armazenamento vinculadas a assinatura Azure, selecione a conta de armazenamento que deseja utilizar e em seguida o contêiner que receberá os dados;
- **Frequência do lote:** define o tempo máximo de espera em segundos para que os dados sejam enviados ao contêiner;
- **Janela de tamanho da parte:** tamanho máximo do arquivo de blobs (entre 10 e 50 MB, valor mínimo definido pelo serviço em nuvem);

- **Codificação:** formato dos dados gravados no blobs (escolhido foi JSON por ser um formato de fácil manipulação);
- **Formato do nome do arquivo:** deve seguir um padrão de informações exigido pelo serviço contendo no nome do arquivo o nome do IoT Hub, nome da partição que serão gravados os dados, ano, mês, dia, hora e minuto que o arquivo foi criado.
- **Tipo de autenticação:** define o método de autenticação entre os serviços, possuindo as opções Baseado em chave e Atribuída pelo usuário. Como ambos serviços encontra-se no mesmo ambiente, a opção Baseado em chave foi a escolhida.

Nas Figuras 33 e 34 apresenta-se a primeira parte de configuração de uma rota entre o IoT Hub e o serviço de armazenamento.

Figura 33 – Configuração de rota do IoT Hub para contêiner do Blob Storage.

Adicionar uma rota ...

hubAirstuff

1 Endpoint 2 Route 3 Enrichment

Crie um ponto de extremidade para sua rota— isso determinará quais serviços do Azure receberão suas mensagens. Você pode ter no máximo 10 pontos de extremidade para cada Hub IoT. [Saiba mais](#)

Tipo de ponto de extremidade ⓘ

Armazenamento

Nome do ponto de extremidade *

dadosEsp8266

[Selecionar existente](#)

Contêiner de armazenamento do Azure *

<https://airstuffstorage.blob.core.windows.net/dados-iot2>

Escolha um contêiner

Frequência do lote * ⓘ

100

Fonte: Elaborado pelo autor.

Terminada a configuração da aba **Endpoint** basta configurar a segunda parte denominada *Route* onde devem ser especificados os seguintes dados:

- **Nome:** especifica o nome para a rota de comunicação entre o IoT Hub e o contêiner do Blob Storage;
- Marcar a opção **Habilitar rota** caso não esteja selecionada;
- **Fonte de dados:** especifica o tipo de dado que será transmitido entre os serviços.

Figura 34 – continuação da configuração de rota do IoT Hub para contêiner do Blob Storage.

Janela de tamanho da parte * ⓘ

 *

Codificação * ⓘ

☐ AVRO

☒ JSON

Formato do nome do arquivo * ⓘ

 *

Se vários arquivos forem criados no mesmo minuto, o formato de nome de arquivo será {iothub}/0/2023/11/15/2-01.json

Tipo de autenticação *

Escolha o tipo de autenticação para esse ponto de extremidade de roteamento. [Saiba mais](#)

☒ Baseado em chave

☐ Atribuída pelo sistema

☐ Atribuída pelo usuário

[Criar + avançar](#)

Fonte: Elaborado pelo autor.

Após realizar todas as especificações apresentadas a rota entre os serviços pode ser implementada, antes de confirmar a criação da rota podem ser realizados testes para verificar se a comunicação entre os serviços está correta no campo **Consulta de roteamento**.

Configurações adicionais podem ser adicionadas no campo *Enrichment*, onde podem ser adicionados até 10 informações adicionais que podem ser anexadas aos dados que serão enviados do IoT Hub para o contêiner do Blob Storage. Na Figura 35 demonstra-se a parte final da configuração da rota.

Ao final, sua nova rota aparecerá listada na opção Roteamento de mensagens dentro das opções de gerenciamento do IoT Hub nas opções Rotas e Pontos de extremidade personalizados como mostrado na Figura 36.

Para acessar os dados em um contêiner basta clicar sobre seu nome na lista de contêineres no serviço Blob Storage e navegar pelos diretórios criados até os dados. Cada contêiner pode possuir sua própria estrutura de diretórios e para este projeto foram definidas pastas para organizar os dados por data e hora da gravação do dado no contêiner. O formato dos dados escolhido foi o de objetos em formato texto, para facilitar o acesso futuro a outros serviços de análises de dados.

Secker (2020) em seu trabalho descreve o serviço Blob Storage como um serviço de

Figura 35 – Configuração final de rota do IoT Hub para contêiner do Blob Storage.

Endpoint **Route** Enrichment

Crie uma rota e adicione uma consulta para filtrar dados antes de encaminhá-los para o ponto de extremidade. [Saiba mais](#)

Nome *

dadosHubToBlob

☒ Habilitar rota

Fonte de dados *

Mensagens de Telemetria do Dispositivo

Consulta de roteamento

1 true

> Teste

Anterior Criar + ignorar enriquecimentos Criar + adicionar enriquecimentos

Fonte: Elaborado pelo autor.

Figura 36 – Visualização das rotas do IoT Hub.

Rotas **Pontos de extremidade personalizados** Enriquecer mensagens

Escolha quais serviços do Azure receberão suas mensagens. Você pode adicionar até 10 pontos de extremidade a um hub IoT.

Sincronizar chaves Alterar tipo de autenticação Excluir Atualizar

▼ Cosmos DB

▼ Hubs de Eventos

▼ Fila do Barramento de Serviço

▼ Tópico do Barramento de Serviço

▲ Armazenamento

Recomendado para o arquivamento de dados.

☐ Nome	Nome do contêiner	Formato de codifica...	Frequência do lote (...)	Formato de nome d...	Tipo de autenticação	Status
☐ dadosEsp8266	dados-iot2	JSON	100	{iothub}/{partition}/{...	Baseado em chave	● Novo

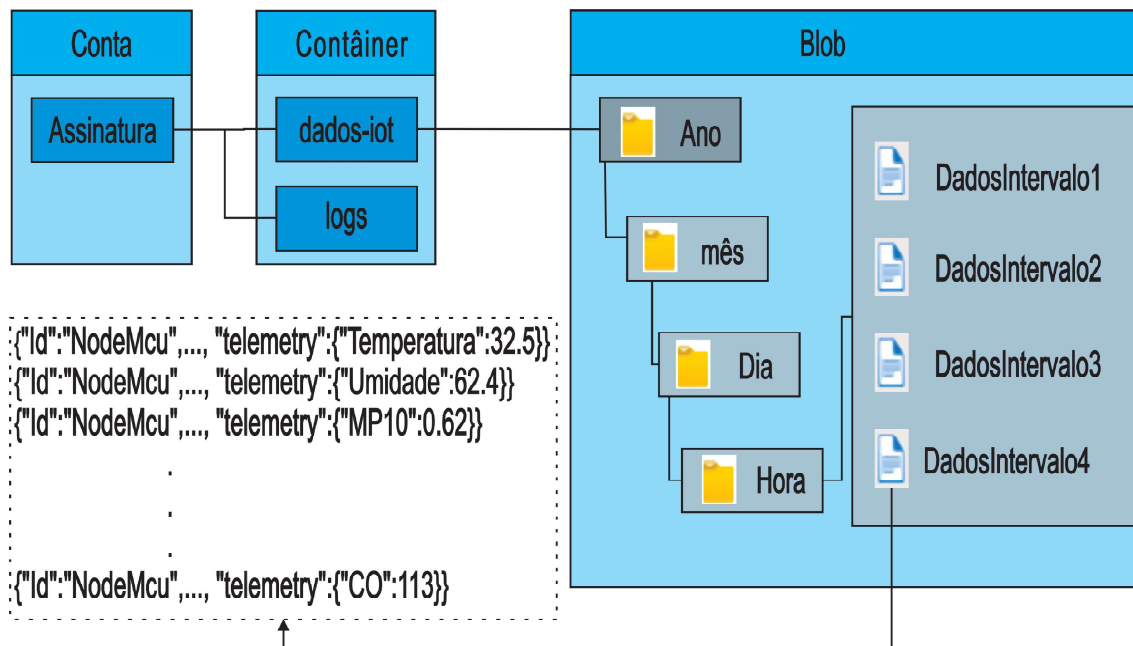
Fonte: Elaborado pelo autor.

armazenamento não estruturado em formato de *data-lake* recomendado para armazenar dados de histórico de IoT de forma barata e com recuperação de dados lenta. Como ele armazena os dados no formato JSON (*JavaScript Object Notation*) em subpastas organizadas por dia, facilita a análise por máquina e a integração a outros bancos de

dados.

Para este projeto foi definido que dentro de cada arquivo contém os dados recebidos a cada 30 minuto com as informações sobre o dispositivo IoT conectado, dados sobre data e hora do armazenamento e dados de telemetria lidos e enviados pelo dispositivo IoT. Na Figura 37 apresenta-se a estrutura aplicada ao serviço Blob Storage.

Figura 37 – Estrutura de armazenamento do serviço Blob Storage.



Fonte: Elaborado pelo autor.

O serviço Blob Storage permite a visualização individual dos arquivos armazenados nos contêineres apresentando os dados sobre a conta Azure, dados do arquivo e seu conteúdo. Na Figura 38 apresenta-se a área de visualização dos arquivos armazenados no Blob Storage enquanto na Figura 39 demonstra-se os dados armazenados no arquivo.

Cada linha presente no arquivo de blobs corresponde a um dado de telemetria recebido pelo IoT Hub do dispositivo IoT, além dos dados de telemetria, na linha também estão presentes os dados referentes a qual dispositivo IoT enviou os dados, data e hora do envio, tipo de dado recebido entre outras informações de configurações dos serviços Azure. Estes dados podem ser lidos diretamente no arquivo de blob ou podem ser criadas funções de extração de dados para selecionar somente os dados de telemetria e data da leitura armazenada.

Tais funções de extração podem ser criadas utilizando serviços Azure de análise de dados ou até mesmo por aplicações externas como sistemas *web* de terceiros, visto que o serviço permite o acesso aos dados através de APIs.

Figura 38 – Visualização dos arquivos de dados e suas propriedades.

es > dados-iot >

Carregar Adicionar Diretório

Método de autenticação: Chave de acesso (Alternar para conta de usuário do Microsoft Entra)

Local: dados-iot / a26f70ce-cf89-4c24-acd9-a92728af84ae / 3 / 2023 / 07 / 21 / 17 / 30

Pesquisar blobs por prefixo (...)

Mostrar os objetos que foram excluídos

Dados sobre o local de armazenamento

Nome

[...] [5nstpgx25hjuw]

Arquivo de dados com nome criptografado

a26f70ce-cf89-4c24-acd9-a92728af84ae/3/2023/07/21/

Blob

Salvar Descartar Baixar Atualizar Excluir Alterar a camada Ad

Visão geral Versões Editar Gerar SAS

Propriedades

URL https://airstuffstorage.b...

ÚLTIMA MODIFICAÇÃO 21/07/2023, 2:31:09 PM

HORA DE CRIAÇÃO 21/07/2023, 2:31:09 PM

IDENTIFICAÇÃO DA VERSÃO -

TIPO Blob de blocos

TAMANHO 3.25 KiB

CAMADA DE ACESSO Principal (Inferidos)

ÚLTIMA MODIFICAÇÃO DA CAMADA DE ACESSO N/D

STATUS DO ARQUIVO -

REIDRATAR A PRIORIDADE -

SERVIDOR CRIPTOGRAFADO true

ETAG 0x8DB8A1045E51396

Dados de criação do arquivo

Dados sobre tamanho do arquivo

Fonte: Elaborado pelo autor.

Figura 39 – Dados presentes no arquivo de blobs.

Salvar Descartar Baixar Atualizar Excluir

Visão geral Versões Editar Gerar SAS

```

1 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"Temperatura":29.5}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
2 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"Umidade":50.6}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
3 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"CO":7}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
4 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"NH3":5}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
5 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"NO2":13}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
6 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"MP10":1.972}, "templateId":"dtmi:urdfss0m:n7vvaznpef"
7 messageSource:"telemetry", "schema":"default@v1", "telemetry":{"MP25":4.728}, "templateId":"dtmi:urdfss0m:n7vvaznpef"

```

Visualização

Fonte: Elaborado pelo autor.

Esta subseção apresentou o serviço de armazenamento de dados Blob Storage, a implantação realizada para este projeto, abordando as principais configurações necessárias

e por fim o método de armazenamento utilizado e o método de visualização dos dados. Na próxima subseção serão apresentadas as principais características do serviço IoT Central, suas configurações e interface.

5.2.3 Configuração IoT Central

Para a disponibilização dos dados de telemetria recebidos do dispositivo IoT foi utilizado o serviço IoT Central, que fornece toda interface para exibir os dados em qualquer dispositivo conectado a Internet.

IoT Central é um serviço PaaS que fornece um conjunto de ferramentas prontas para facilitar o desenvolvimento de interfaces de usuário para disponibilização de dados de dispositivos IoT. Fornece ainda recursos de exportação de dados para outros serviços, criação de percepções sobre os dados e acesso aos dados recebidos por meio de APIs (learn.microsoft.com, 2023g).

Neste projeto os dados recebidos pelo serviço IoT Hub são redirecionados para o IoT Central para a visualização em tempo real através da internet. Para criar um novo aplicativo IoT Central, deve-se clicar na opção Aplicativos da IoT Central na página inicial do Azure para iniciar a configuração.

Na área de aplicativos do IoT Central é possível visualizar ou acessar os aplicativos já criados e criar novos aplicativos clicando no botão Criar, como mostrado na Figura 40.

A configuração de novos aplicativos IoT Central é facilitada, pois requer inicialmente que sejam definidos os parâmetros de assinatura, nome do recurso/aplicativo, URL de acesso, modelo de aplicativo, região que será implementado e tipo de plano de preços. Na Figura 41 apresenta-se a tela de configuração de um novo aplicativo IoT Central.

Deve-se atentar para as opções Modelo e Plano de preços, pois a primeira define uma interface base para a construção do aplicativo, onde cada opção define um formato de dados que serão exibidos. A segunda opção define os recursos disponíveis no aplicativo e seus respectivos preços de cobrança. Para este projeto foram definidas a opção de Aplicativo personalizado para o Modelo e plano *Standard* 0 para a opção Plano de preços devido às necessidades do projeto.

Após a implantação do serviço, é disponibilizada uma área para configurações adicionais e acesso ao recurso criado, basta clicar na URL do aplicativo para ter acesso a sua versão completa e realizar as adequações conforme a necessidade do projeto. Na Figura 42 demonstra-se a área de acesso ao IoT Central.

Na área de configuração do aplicativo deve ser feita a configuração da fonte de dados que será utilizada pelo aplicativo IoT Central, que pode ser a comunicação direta do

Figura 40 – Página principal do IoT Central

Aplicativos da IoT Central ✨ ... X

Etec | Centro Paula Souza (etec.sp.gov.br)

+ Criar Gerenciar a exibição Atualizar Exportar para CSV Abrir a consulta Atribuir marcações

→ Criar novo aplicativo

Filtrar por qualquer ca... Assinatura igual a **tudo** Grupo de recursos igual a **tudo** X Adicionar filtro Mais (1)

Mostrando 1 a 2 de 2 registros. Nenhum agrupamento Exibição de lista

Nome ↑↓	Tipo ↑↓	Grupo de recursos ↑↓	Localização ↑↓	Assinatura ↑↓
airstuffnodemcu	Aplicativo IoT Central	AirStuff	East US	AzureEduardoCruz ...
device-app	Aplicativo IoT Central	AirStuff	East US 2	AzureEduardoCruz ...

→ Aplicativos existentes

< Anterior Página 1 de 1 Próximo > Enviar comentários

Fonte: Elaborado pelo autor.

Figura 41 – Configuração de novo aplicativo IoT Central

Aplicativo IoT Central ...

organize and manage all your resources.

Assinatura * ⓘ

Assinatura AzureEduardoCruz

Grupo de recursos * ⓘ

Grupo de recursos AirStuff

Criar novo

Instance details

Nome do recurso *

Nome do recurso dispositivoiot

URL do Aplicativo *

URL do Aplicativo dispositivoiot

azureiotcentral.com

Modelo * ⓘ

Modelo Aplicativo personalizado

Region *

Region Leste dos EUA

Plano de preços * ⓘ

Plano de preços Standard 0

< Anterior Próxima Examinar + criar

Dados da assinatura Azure

Dados sobre nome do aplicativo e URL de acesso

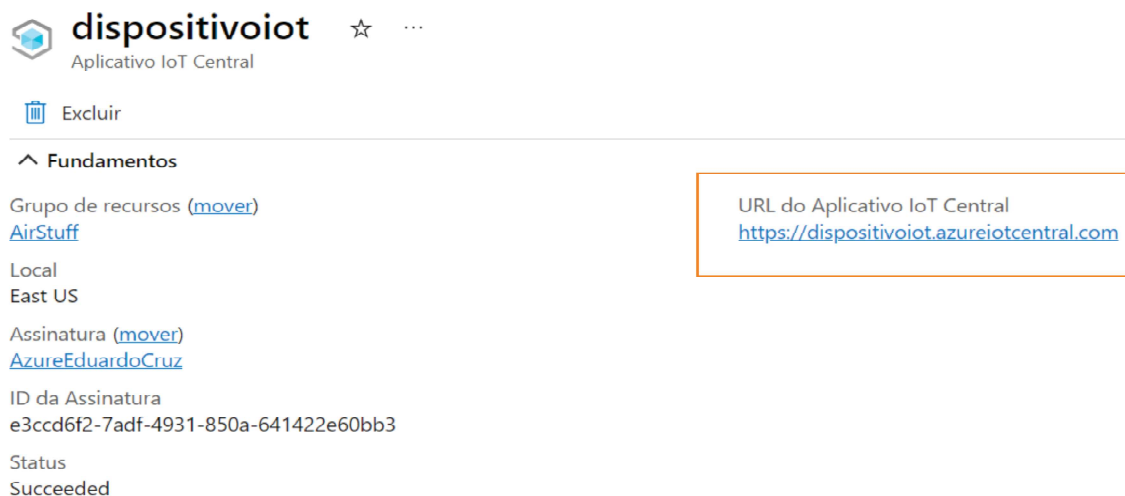
Tipo de modelo base do aplicativo

Região de hospedagem

Tipo de plano de uso e cobrança

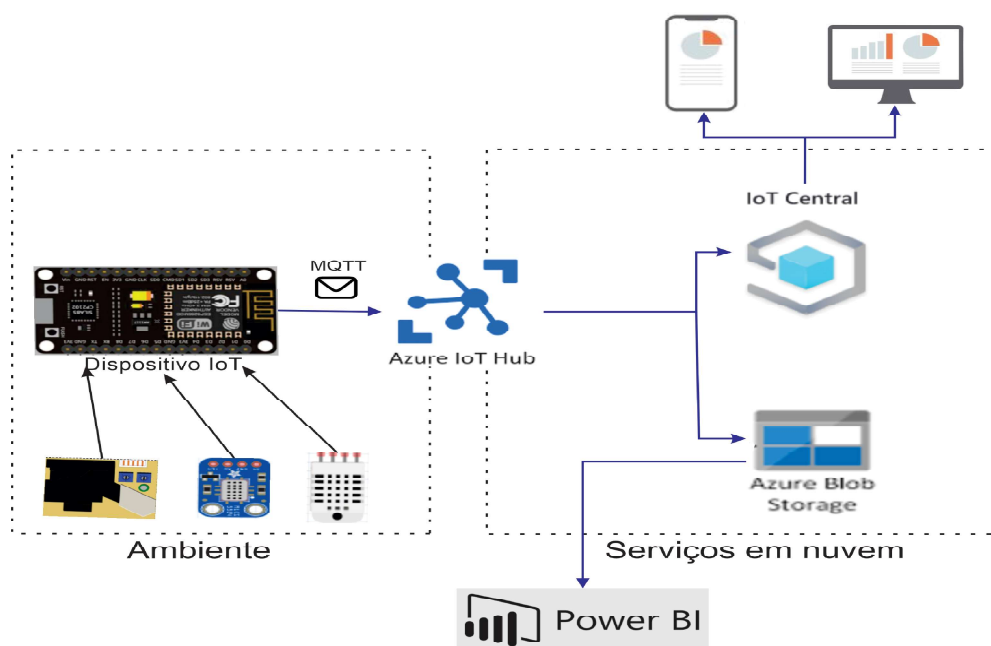
Fonte: Elaborado pelo autor.

aplicativo com um dispositivo IoT ou com outros serviços em nuvem. Para este projeto foi implantada a comunicação do aplicativo com o serviço em nuvem IoT Hub, não havendo

Figura 42 – Configuração de novo aplicativo IoT Central

Fonte: Elaborado pelo autor.

acesso direto entre o aplicativo e o dispositivo IoT como apresenta a Figura 43.

Figura 43 – Comunicação entre as camadas e serviços do projeto.

Fonte: Elaborado pelo autor.

Com o aplicativo IoT Central criado, deve-se criar um dispositivo dentro do aplicativo para que os dados recebidos pelo serviço IoT Hub possam ser exibidos na interface do aplicativo. Por padrão, o serviço em nuvem atribui valores para todas as informações que devem ser fornecidas, é recomendável que tais informações sejam personalizadas para facilitar a criação de novos recursos e alterações nas configurações do aplicativo. Na Figura

44 apresenta-se a criação de um dispositivo dentro do IoT Central.

Figura 44 – Criação de novo dispositivo no IoT Central.

The screenshot shows the 'dispositivoiot' interface. The top bar has a search icon and the text 'Pesquisar por dispositivos'. The main header is 'Dispositivos' with a '+ Novo' button highlighted in a yellow box and an 'Importar' button. Below the header, there's a 'Modelos de dispositivo' dropdown and a 'Criar um novo dispositivo.' title. The form contains the following fields and options:

- Nome do dispositivo ***: Input field with '1tfro60r3na'.
- ID do dispositivo ***: Input field with '1tfro60r3na'.
- Organização ***: Input field with 'dispositivoiot'.
- Modelo de dispositivo ***: Dropdown menu with 'Não atribuído'.
- Deseja simular este dispositivo?**: Toggle switch set to 'Não'.
- Dispositivo Azure IoT Edge?**: Toggle switch set to 'Não'.

At the bottom right, there are 'Criar' and 'Cancelar' buttons.

Fonte: Elaborado pelo autor.

Com o dispositivo criado, agora basta criar as tags MQTT para receber os dados do serviço IoT Hub e exibi-los no aplicativo através da opção Modelos de dispositivo. Nesta opção são definidos quais os dados serão exibidos, seu formato, personalizações visuais para cada dado entre outras configurações.

É possível criar um modelo de exibição de dados totalmente do zero ou importar modelos já criados e reutilizá-lo. Na Figura 45 ilustra-se a área de criação de modelos enquanto na 46 demonstra-se o processo de configuração das tags de exibição dos dados.

Após criar todas as *tags* necessárias no modelo de dispositivo, basta salvar as *tags* criadas que estarão disponíveis na página de visualização dos dados recebidos.

O próximo passo é criar um componente de exibição personalizado para cada um dos dados que serão exibidos na opção Modos de exibição, existem vários componentes prontos para serem adicionados ao aplicativo e configurados de acordo com cada tipo de dado a ser exibido como individual de um dado no seu formato original, gráficos exibindo histórico de um dado ou até a combinação de vários dados em um único componente. Na Figura 47 apresenta-se a adição de componentes de exibição dos dados de telemetria.

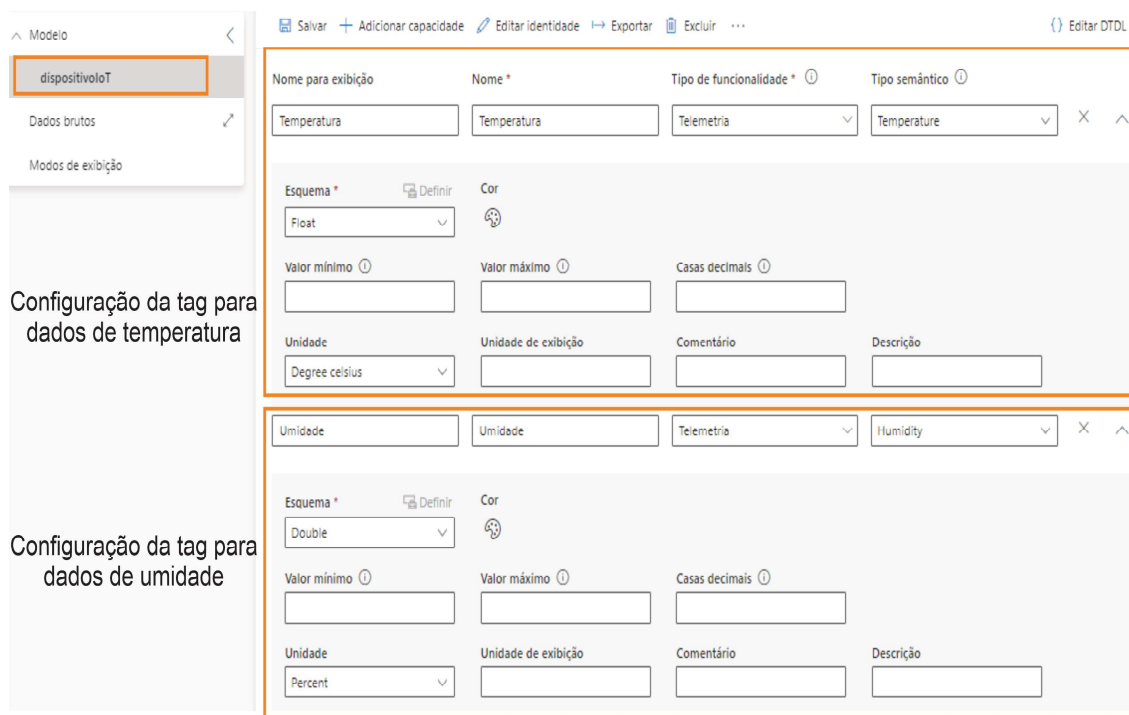
Por fim é necessário realizar a conexão entre o aplicativo criado no IoT Central e a fonte de dados, neste caso o serviço IoT Hub. Acessando a opção Dispositivos, abra o dispositivo adicionado ao IoT Central e clique na opção Conectar no meu superior da

Figura 45 – Área de criação de novos modelos de dispositivo IoT Central



Fonte: Elaborado pelo autor.

Figura 46 – Configuração dos dados a serem exibidos.



Fonte: Elaborado pelo autor.

página.

Copie os dados que aparecerão na janela que se abrirá, pois serão necessários para criar uma rota de roteamento de dados do serviço IoT Hub para o serviço IoT Central

Figura 47 – Configuração dos dados a serem exibidos.

Fonte: Elaborado pelo autor.

como apresentado na subseção 5.2.1. Na Figura 48 apresentam-se os dados necessários para comunicação com o aplicativo IoT Central.

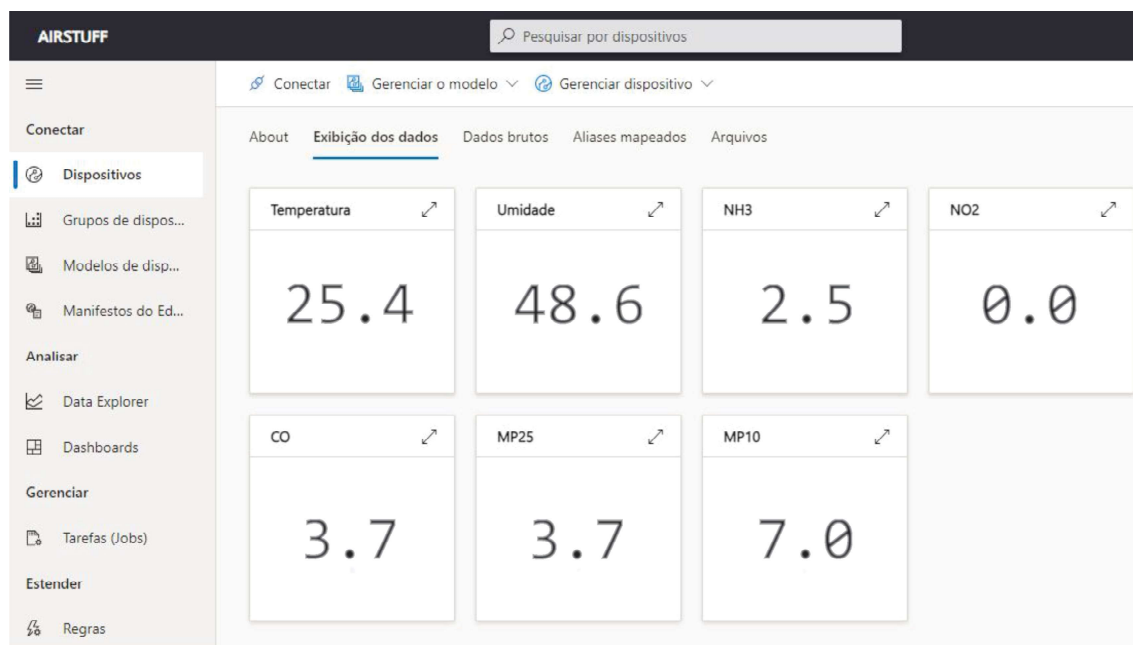
Figura 48 – Dados para a conexão com IoT Central.

Fonte: Elaborado pelo autor.

Após todas as configurações no serviço IoT Central e IoT Hub, é possível visualizar os

dados de telemetria através da Internet por qualquer dispositivo que tenha um navegador de Internet utilizando sua conta Azure. Na Figura 49 apresenta-se a interface de dados do IoT Central.

Figura 49 – Interface de exibição dos dados no IoT Central.



Fonte: Elaborado pelo autor.

Também é possível disponibilizar os dados em forma de aplicativo para celular se necessário, com a possibilidade de utilização gratuita por até 2 dispositivos. A disponibilização do aplicativo para mais dispositivos requer a alteração no plano de cobrança do serviço IoT Central para uma opção mais robusta.

5.3 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO

Este capítulo abordou o desenvolvimento do projeto de qualidade do ar, na seção 5.1 foram apresentadas as configurações dos sensores e programação do microcontrolador ESP8266, apresentando a calibração e mapeamento de dados dos sensores. Na seção 5.2 foi apresentada implementação dos serviços em nuvem e a comunicação entre eles.

No próximo capítulo serão apresentados os resultados relacionados a coleta de dados realizada pelo projeto.

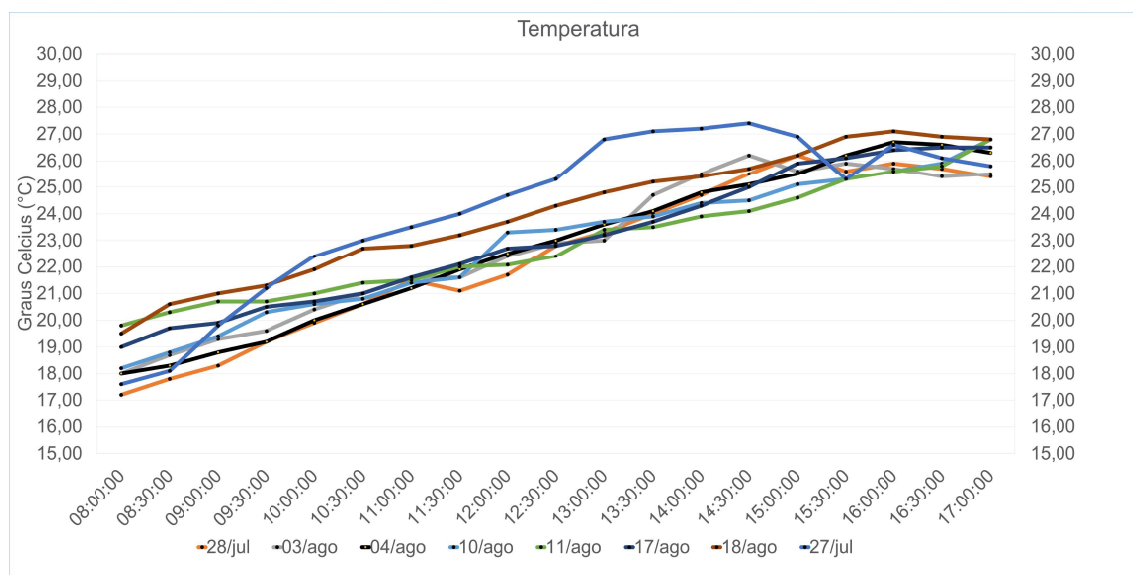
6 RESULTADOS

Os dados coletados e armazenados no serviço Blob Storage foram utilizados pelo *software* Microsoft Power BI para elaboração de gráficos. Para cada valor de telemetria foi gerado um gráfico para comparação entre horário e datas, o intervalo de leitura adotado foi de 30 minutos, iniciando a primeira leitura sempre as 8h e finalizando as 17h.

Os horários com maiores aumentos na temperatura do ambiente foram entre as 12h30min e 13h30min, horário de troca de turnos das turmas do ensino médio e após as 15h devido ao período de intervalo e aumento natural da temperatura.

Em 27 de julho houve a maior alteração de temperatura devido à necessidade de deixar a porta do ambiente aberta para entrada e saída de alunos que estavam realizando uma atividade proposta pela unidade escolar. Na Figura 50 apresenta-se gráfico com as alterações de temperatura do ambiente.

Figura 50 – Dados referentes a temperatura do ambiente.

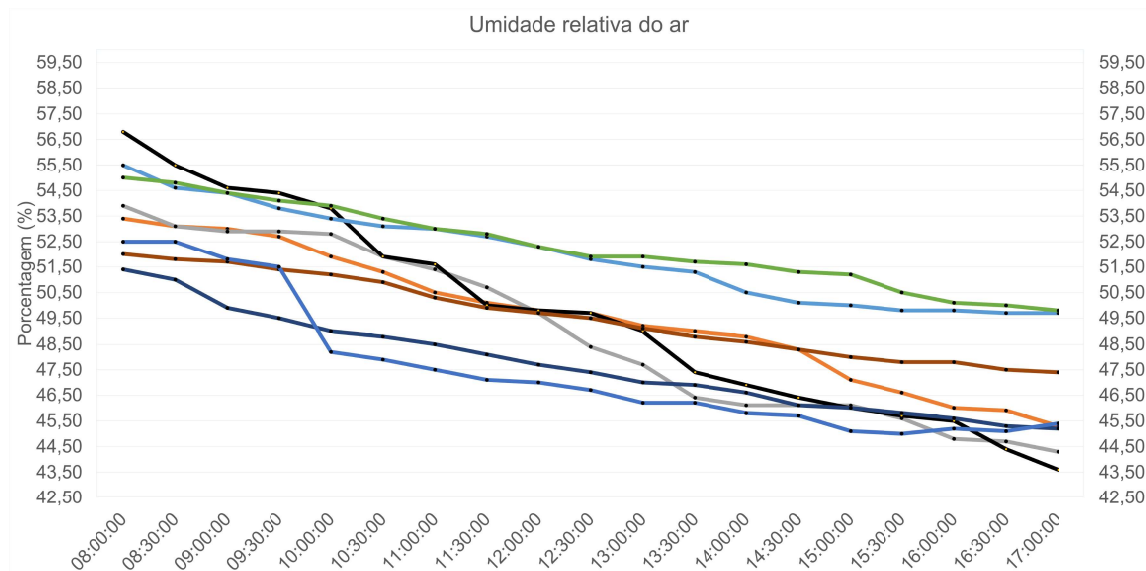


Fonte: Elaborado pelo autor.

Em relação à umidade do ar, os dados apresentaram uma queda na umidade durante todo o período de utilização do laboratório, uma das principais causas é o uso constante do

ar-condicionado e o aumento de temperatura ao longo do dia. Na Figura 51 demonstra-se o gráfico referentes aos dados de umidade relativa do ar coletados.

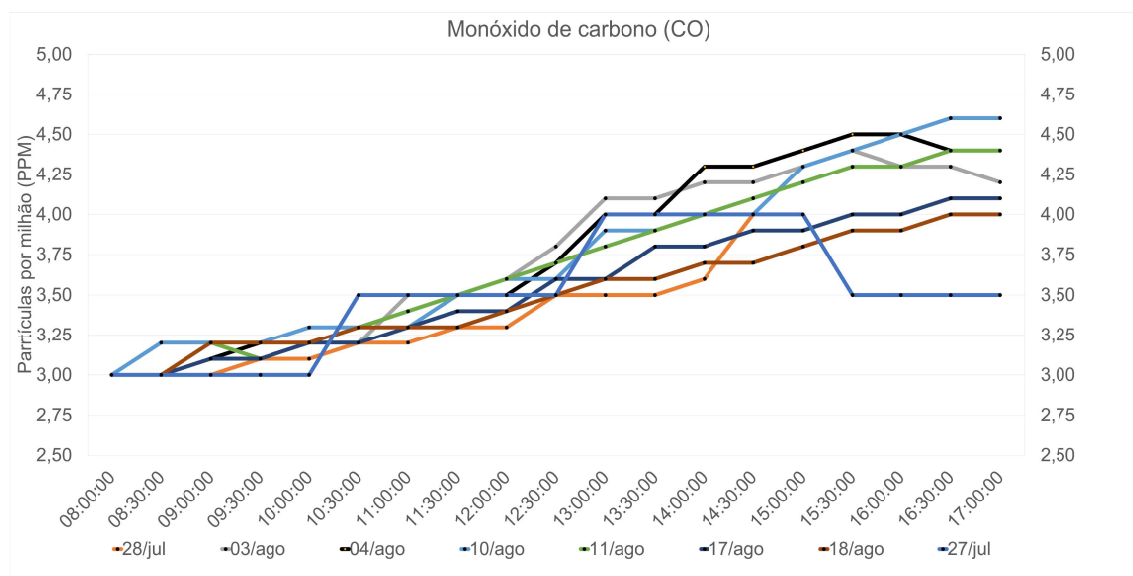
Figura 51 – Dados referentes a umidade relativa do ar do ambiente.



Fonte: Elaborado pelo autor.

Ao longo do dia houve aumento na concentração de CO, uma das causas possíveis é o aumento de fluxo de carros transitando ao longo do dia. Na Figura 52 apresenta-se gráfico com os dados de CO coletados.

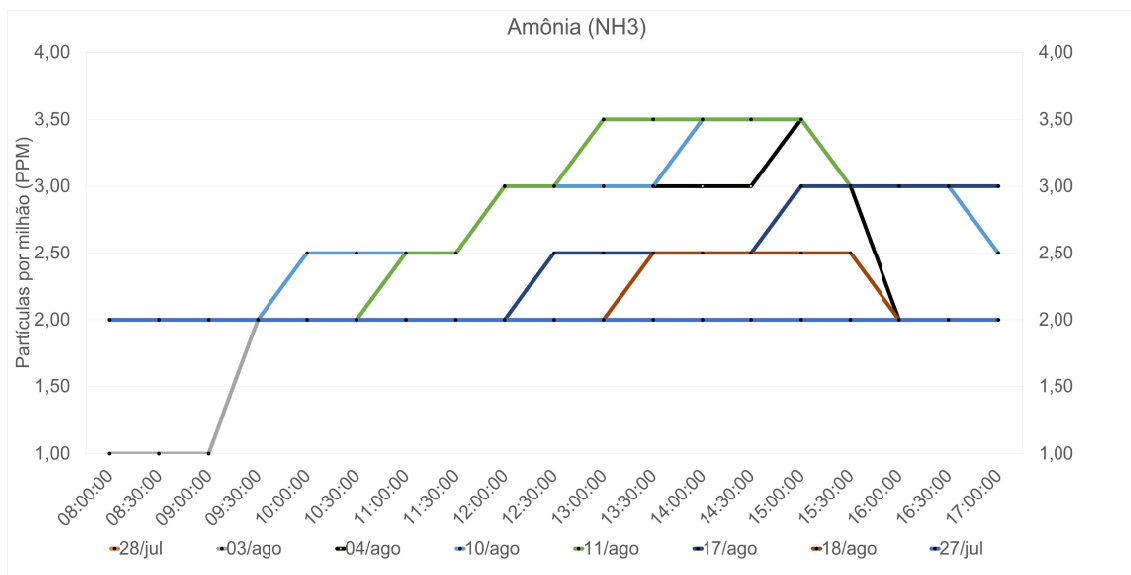
Figura 52 – Dados referentes a monóxido de carbono(CO) do ambiente.



Fonte: Elaborado pelo autor.

Observou-se que entre 12h30min e 15h ocorreram as maiores alterações relacionadas a NH_3 , a chegada de alunos para as aulas do período da tarde pode ser uma das causas da mudança no ambiente. Na Figura 53 apresenta-se gráfico com os dados sobre NH_3 .

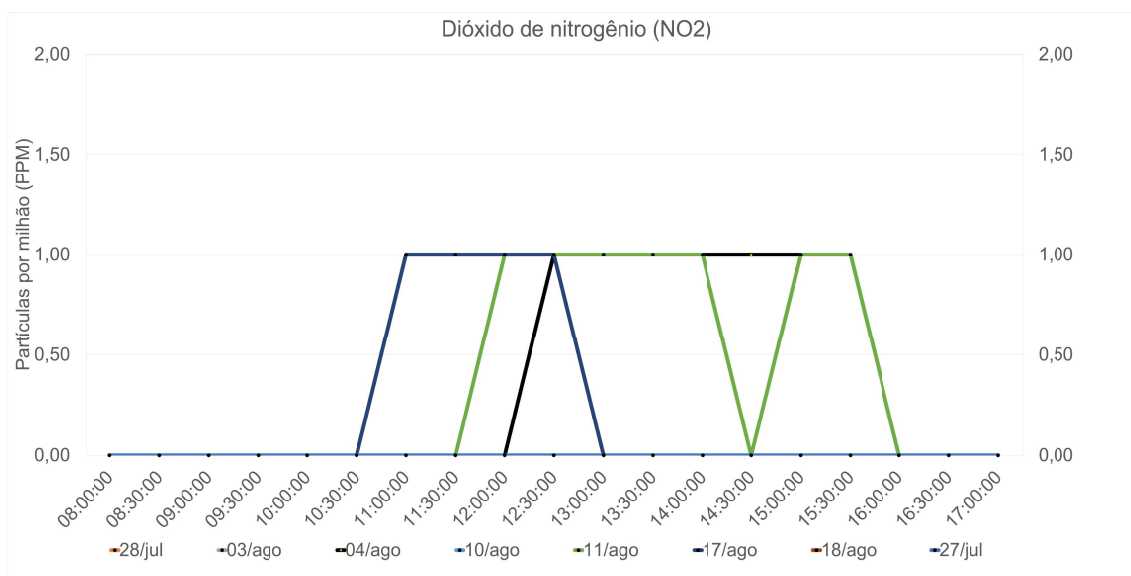
Figura 53 – Dados referentes a amônia (NH_3) do ambiente.



Fonte: Elaborado pelo autor.

Não foram apresentadas alterações significativas relacionadas ao NO_2 conforme apresentado na Figura 54.

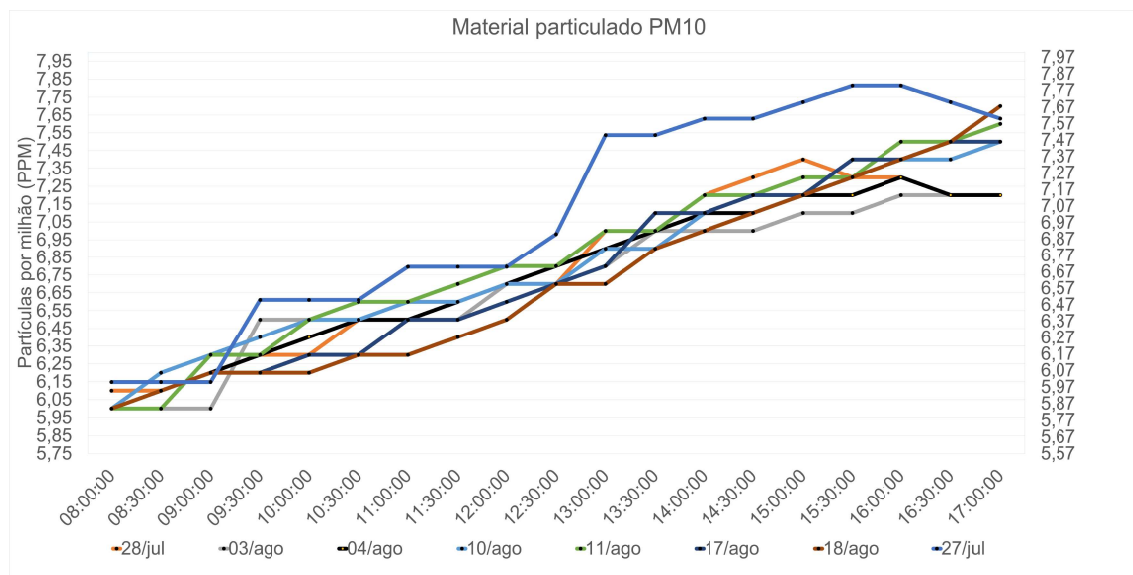
Figura 54 – Dados referentes ao dióxido de nitrogênio (NO_2) do ambiente.



Fonte: Elaborado pelo autor.

Percebeu-se que ao longo do dia houve aumento na quantidade de partículas em suspensão PM10 e em 27 de julho houve o maior aumento conforme demonstrado no gráfico apresenta na Figura 55.

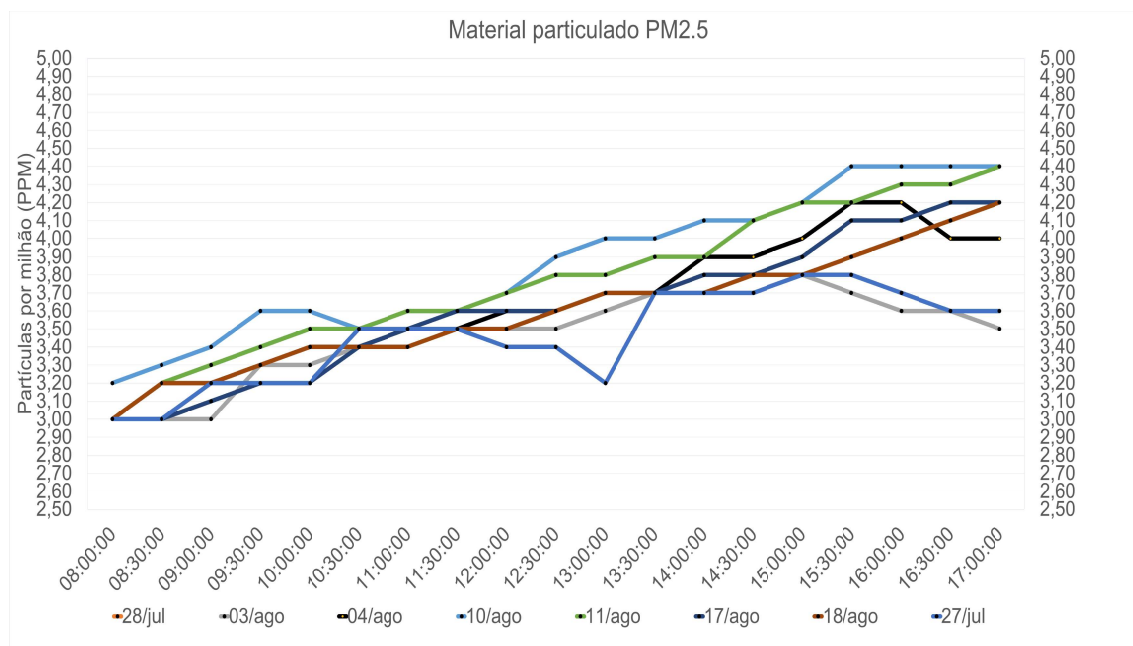
Figura 55 – Dados referentes partículas em suspensão de até 10 m (PM10) do ambiente.



Fonte: Elaborado pelo autor.

Por fim, também houve aumento na quantidade de partículas em suspensão ao longo do período de medições, especialmente em 27 de julho. Na Figura 56 demonstra-se gráfico com os dados sobre PM25.

Figura 56 – Dados referentes a partículas de até 2,5 m (PM25) do ambiente.



Fonte: Elaborado pelo autor.

7 CONCLUSÃO

Este trabalho apresentou uma proposta de monitoramento da qualidade do ar em ambientes internos baseada em dispositivo IoT conectado a um conjunto de serviços em nuvem para gerenciamento, visualização e armazenamento dos dados de telemetria coletados.

O uso do microcontrolador ESP8266 e dos sensores DHT22, DSM501A e MICS6914 apresentou desempenho satisfatório na coleta de dados, visto o bom custo benefício de aquisição dos materiais, fácil programação do microcontrolador e calibração dos sensores. O único ponto negativo observado no microcontrolador ESP8255 do ponto de vista de desempenho foi a necessidade do uso do multiplexador CD74HC4067 para suprir a baixa qualidade de portas analógicas do ESP8266 (somente uma funcional), o que pode influenciar na velocidade de transmissão de dados em projetos que necessitem de alta disponibilidade dos dados.

Em relação aos serviços em nuvem utilizados, foram adotados serviços provenientes do Microsoft Azure devido à familiaridade com o serviço e gratuidade da maioria deles. O serviço IoT Hub apresentou ser uma ótima solução para criação de um MQTT *broker* pois facilita a comunicação entre o dispositivo IoT e a Internet, proporcionando a comunicação do dispositivo físico a outros serviços de *software*.

A disponibilização dos dados para serem visualizados através da internet usando o serviço IoT Central apresentou resultados satisfatórios devido a fácil configuração da interface, adição de novas *tags* MQTT se necessário e a possibilidade de visualizar os dados coletados em diversos formatos como planilha, formato original e comparativos entre os valores.

Um ponto negativo a ser destacado na utilização dos serviços em nuvem é que para reduzir os custos na utilização do serviço IoT Central, somente dois dispositivos podem acessar o serviço em simultâneo através de aplicativo gerado pelo próprio serviço, além disso, o acesso fica restrito a conta Azure ao qual o serviço está vinculado.

O serviço de armazenamento Blob Storage se mostrou uma boa alternativa para projetos que não necessitam de armazenamento de dados de maneira estruturada e

totalmente relacionados, proporcionando o armazenamento dos dados de telemetria em um formato JSON e organizado por arquivos, facilitando o acesso e visualização dos dados lidos. Por outro lado, a não estruturação dos dados armazenados pode gerar a necessidade de criação de algoritmos para realizar a seleção dos dados que serão utilizados, principalmente se forem utilizados por serviços ou *softwares* fora da nuvem Azure.

Por fim, a proposta do projeto foi utilizar a maior quantidade de serviços em nuvem e de forma gratuita para implantação de um projeto IoT para monitoramento da qualidade do ar em ambientes internos. O microcontrolador ESP8266 atendeu as necessidades do projeto e é uma boa alternativa para projetos com IoT devido ao baixo custo e facilidade de conexão a redes sem fio.

Os serviços em nuvem Azure se mostraram uma alternativa interessante para a coleta, armazenamento e visualização dos dados devido a fácil configuração dos serviços utilizados e compatibilidade entre eles.

Para este projeto foram estudadas e encontradas soluções que proporcionaram custo quase zero na utilização dos serviços em nuvem, os maiores custos apresentados no processo foram em testes com os serviços onde o uso de rede excedeu os parâmetros de gratuidade do serviço ou utilização de serviços complementares como Azure Synapse Analytics que além de não apresentar recursos necessários ao projeto, resultou em cobrança sobre o uso do serviço.

Após estudos e testes no serviço em nuvem chegou-se a uma configuração ao qual o maior custo de implementação durante o pico de uso dos serviços foi de R\$2,30 mensal, custo extremamente baixo em relação à infraestrutura que seria necessária se o projeto fosse implementado utilizando equipamentos como um servidor local ou até mesmo uso de um Raspberry Pi.

O projeto se provou viável para aplicações de pequeno porte que utilizem poucos dispositivos IoT, onde a coleta de dados não é constante, o que gera um grande volume de dados e tráfego de rede, visto que a maioria dos serviços em nuvem possui um limite de tráfego de dados e armazenamento gratuito.

A utilização dos serviços em nuvem provou-se uma excelente alternativa quando se deseja evitar trabalhar com linguagens de programação para desenvolvimento *web*, banco de dados, construção de APIs e outras tecnologias que o desenvolvimento de um *software* demanda.

A exclusão da necessidade de aquisição ou uso de infraestrutura física para execução de um sistema *web* é outro atrativo a ser considerado na escolha do uso de serviços em nuvem em projetos de IoT. Os serviços em nuvem podem fornecer toda a infraestrutura

para a execução de um projeto IoT com menor custo que a compra de equipamentos, além da possibilidade de pagamento conforme uso do serviço.

REFERÊNCIAS

- Abdelqawy, D.; El-Korany, A.; Kamel, A.; Makady, S. Hub-os: An interoperable iot computing platform for resources utilization with real-time support. **Journal of King Saud University – Computer and Information Sciences.**, [S. l.], p. 1498–1510, 2022.
- Ahmed, A. A.; Omari, S. A.; Awal, R.; Fares, A.; M.Chouikha. A distributed system for supporting smart irrigation using internet of things technology. **Engineering Reports**, [S. l.], p. 1–13, 2020.
- amazon.com. **O que é um container?** [S. l.]: , 2022. Site. Acesso em: 10.06.2022. Disponível em: <https://aws.amazon.com/pt/containers/>.
- ANS. **Planos de saúde realizaram 1,3 bilhão de procedimentos em 2020.** 2022. Site. Acesso em: 22.10.2022. Disponível em: <https://www.gov.br/ans/pt-br/assuntos/noticias/numeros-do-setor/planos-de-saude-realizaram-1-3-bilhao-de-procedimentos-em-2020>.
- ANVISA. **Constituição (2003). Resolução nº 09, de 2003.** 2022. Site. Acesso em: 20.10.2022. Disponível em: http://antigo.anvisa.gov.br/documents/10181/2718376/RE_-09_-2003_-pdf/8ccafc91-1437-4695-8e3a-2a97deca4e10.
- Aosong, E. **Digital-output relative humidity temperature sensor/module DHT22.** 2023. Site. Acessado em: 28.03.2023. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.
- Ayode, J. O. **Introdução a climatologia para os trópicos.** : Bertrand Brasil, 1994. 350 p.
- azure.microsoft.com. **Preços do Armazenamento de Blobs do Azure.** 2023. Site. Acesso em: 24.09.2023. Disponível em: <https://azure.microsoft.com/pt-br/pricing/details/storage/blobs/>.
- azure.microsoft.com. **Preços do Azure IoT Central.** 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://azure.microsoft.com/pt-br/pricing/details/iot-central/>.
- Barik, R. K.; Lenka, R. K.; Rao, K. R.; Ghose, D. Performance analysis of virtual machines and containers in cloud computing. **International Conference od Computing, Communication and Automation ICCCA**, [S. l.], p. 1–7, 2016.
- Bauerová, P.; Indelarová, A.; Rychlik, S.; Novak, Z.; Keder j. Low-cost air quality sensors: one-year field comparative measurement of different gas sensors and particle counters with reference monitors at tusimice observatory. **Atmosphere**, [S. l.], p. 492–506, 2020.
- Bellalta, B.; Bononi, L.; Bruno, R.; Kassler, A. Next generation ieee 802.11 wireless local area networks: current status, future directions and open challenges. **Computer Communications**, [S. l.], p. 1–25, 2016.
- Bhimani, J.; Yang, Z.; Leeser, M.; Mi, N. Accelerating big data applications using lightweight virtualization framework on enterprise cloud. **IEEE Conference on High Performance Extreme Computing (HPEC)**, [S. l.], 2017.

Boid, A.; Callaly, F.; Canavan, D.; O'Loughlin, D.; Audiger, J.; Boyer, Y.; Timlin-Canning, N.; Vianes, A.; Da-Costa, C.; Morgan, F.; Bakó, L.; Hajdú, S. Iccapt: Online design capture and hdl generation, with pynq soc prototyping in the cloud. **29th Irish Signals and Systems Conference (ISSC)**, [S. l.], 2018.

Castelhano, F. J. Riscos e vulnerabilidades sob a perspectiva da qualidade do ar. **Finisterra**, [S. l.], p. 118, 2021.

CETESB. **Padrões de qualidade do ar**. 2022. Site. Acesso em: 31.10.2022. Disponível em: <https://cetesb.sp.gov.br/ar/padroes-de-qualidade-do-ar>.

Chen, R. C.; Guo, H. Y.; Lin, M. P.; Lin, H. S. The carbon dioxide concentration detection using mobile phones combine bluetooth and qr codes. In: **Proceedings of the 6th IEEE International Conference on Awareness Science and Technology (iCAST)**. [S. l.] Anais [...]: , 2014. p. 6th IEEE International Conference on Awareness Science and Technology (iCAST).

Cirani, S.; Ferrari, G.; Mancin, M.; Picone, M. Virtual replication of iot hubs in the cloud: A flexible approach to smart object management. **Journal of Sensor and Actuator Networks**, [S. l.], p. 1–20, 2018.

CONAMA. **Resolução CONAMA nº 491, de 19/11/2018**. 2020. Site. Acesso em: 20.10.2022. Disponível em: <https://cetesb.sp.gov.br/ar/wp-content/uploads/sites/28/2021/05/Apendice-7-Qualidade-do-Ar-2020.pdf>.

Cuadrado-Cordero, I.; Orgerie, A.; Menaud, J. Comparative experimental analysis of the quality-of-service and energy-efficiency of vms and containers' consolidation for cloud applications. **International Conference on Software, Telecommunications and Computer Networks (SoftCOM)**, [S. l.], 2017.

Cunha, L. J.; Valin, P. R. O. Rede de sensores sem fio para monitoramento de variáveis de ambiente. **XI COMPUTER ON THE BEACH**, [S. l.], p. 1–3, 2020.

Dean, J.; Kiendler-Schar, A.; Mengis, N.; Rudich, Y.; Schepanski, K.; Zimmermann, R. Above us only sky. **Communications Earth Environment**, [S. l.], p. 1–4, 2021.

Dias, F. L. T.; Mendonça, F. D.; Pinto, G. M.; Borges, I. S. C.; Oliveira, S. V. Doenças respiratórias no triângulo mineiro: Análise epidemiológica e projetiva com a pandemia de covid-19. **Journal of Health Biological Sciences**, [S. l.], p. 1–6, 2020.

Espressif. **ESP8266EX Datasheet**. 2023.

Ferreira, W. A. P. **Rede neural ARTMAP Fuzzy implementada em hardware aplicada na previsão da qualidade do ar em ambiente interno**. Dissertação (Mestrado) — Universidade Estadual Paulista - UNESP - Campus de Ilha Solteira, 2021.

Firestone, D.; Putnam, A.; Mundkur, S.; Chiou, D.; Dabagh, A.; Andrewartha, M.; Angepat, H.; Bhanu, V.; Caulfield, A.; Chung, E.; Chandrappa, H. K.; Chaturmohta, S.; Humphrey, M.; Lavier, J.; Lam, N.; Liu, F.; Ovtcharov, K.; Padhye, J.; Popuri, G.; Raindel, S.; Sapre, T.; Shaw, M.; Silva, G.; Sivakumar, M.; Srivastava, N.; Verma, A.; Zuhair, Q.; Bansal, D.; Burger, D.; Vaid, K.; Maltz, D. A.; Greenberg, A. Azure accelerated networking: Smartnics in the public cloud. In: . [S. l.] Anais [...]: , 2018. p. 50–64.

Gabel, P.; Koller, C.; Hertig, E. Development of air quality boxes based on low-cost sensor technology for ambient air quality monitoring. **MDPI**, [S. l.], p. 1–24, 2022.

Gonçalves, C. F.; Andrade, E.; Callou, G.; Nogueira, B. Evaluation of performance, energy consumption and cost for environments baes on containers and virtual machines. **Revista Brasileira de Computação aplicada**, [S. l.], v. 13, n. 13, p. 11–26, 2021.

Hewitt, C. N.; Ashworth, K.; Mackenzie, A. R. Using green infrastructure to improve urban air quality (gi4aq). **Springer Science and Business Media LLC**, [S. l.], p. 62–73, 2019.

hivemq. **Publish Subscribe - MQTT Essentials: Part 2**. 2023. Site. Acesso em: 31.03.2023. Disponível em: <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>.

Hwang, H. C.; Park, J.; Shon, J. G. Design and implementation of a reliable message transmission system based on mqtt protocol in iot. **Wireless Personal Communications** **91**, [S. l.], p. 1765–1777, 2016.

ibm.com. **Protocolo MQTT**. 2023. Site. Acesso em: 07.06.2023. Disponível em: <https://www.ibm.com/docs/pt-br/ibm-mq/7.5?topic=ssfsj-7-5-0-com-ibm-mq-pro-doc-q002870--htm>.

IEEE. Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - redline. **IEEE 802.11 Handbook: A Designer’s Companion**, [S. l.], p. 1–7524, 2021.

INSTRUMENTS, T. **High-Speed CMOS Logic 16-Channel Analog Multiplexer/Demultiplexer**. 2023.

Jones, A. P. Indoor air quality and health. **Atmospheric Environment**, [S. l.], p. 4535–4564, 1999.

Jones, N. How to stop data centers from globbling up the world’s electricity. **Springer Nature Limited** **561**, [S. l.], 2018. Acesso em: 03.06.2022.

Kanmai, Z. Tcp/ip protocol security problems and defenses. **2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)**, [S. l.], p. 117–120, 2020.

Karaoghlanian, N.; Nouredine, B.; Saliba, N.; Shihadeh, A.; Lakkis, I. Low cost air quality sensors “purpleair” calibration and inter-calibration dataset in the context of beirut, lebanon. **Data In Brief**, [S. l.], p. 1–9, 2022.

Kelvin, A. A.; Halperin, S. Covid-19 in children: the link in the transmission chain. **The Lancet Infectious Diseases**, [S. l.], p. 633–634, 2020.

Kim, S. H.; Jang, S. Y.; Yang, K. H. Analysis of the determinants of software-as-a-service adoption in small businesses: Risks, benefits, and o rganizational and environmental factors. **Journal of Small Business Managment**, [S. l.], v. 35, p. 303–325, 2017.

- Kodali, R. K.; Soratkal, S. Mqtt based home automation system using esp8266. **IEEE Region 10 Humanitarian Technology Conference (R10-HTC)**, [S. l.], 2016.
- learn.microsoft.com. **Comunicar-se com o Hub IoT usando o protocolo MQTT**. 2023. Site. Acesso em: 25.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/iot/iot-mqtt-connect-to-iot-hub>.
- learn.microsoft.com. **Conceitos de Hub IoT do Azure**. 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/iot-hub/iot-concepts-and-iot-hub>.
- learn.microsoft.com. **Evite cobranças com a conta gratuita Azure**. 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/cost-management-billing/manage/avoid-charges-free-account>.
- learn.microsoft.com. **Integração aos ganchos de serviços**. 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/devops/service-hooks/overview?view=azure-devops>.
- learn.microsoft.com. **Introdução ao Armazenamento de Blobs do Azure**. 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/storage/blobs/storage-blobs-introduction>.
- learn.microsoft.com. **Início Rápido: Começar a usar a Análise de custos**. 2023. Site. Acesso em: 23.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/cost-management-billing/costs/quick-acm-cost-analysis>.
- learn.microsoft.com. **O que é Azure IoT Central?** 2023. Site. Acesso em: 22.09.2023. Disponível em: <https://learn.microsoft.com/pt-br/azure/iot-central/core/overview-iot-central>.
- learn.microsoft.com. **SDKs do Hub IoT do Azure**. 2023. Site. Acesso em: 25.09.2023. Available on: <https://learn.microsoft.com/pt-br/azure/iot-hub/iot-hub-devguide-sdks>.
- Lee, K.; Yoo, Y. K. and C. The impact of container virtualization on network performance of iot devices. **Hindawi Mobile Information Systems**, [S. l.], p. 1–6, 2018.
- Li, J.; Qiu, M.; Ming, Z.; Quan, G.; Qin, X.; Gu, Z. Online optimization for scheduling preemptable tasks on iaas cloud systems. **Journal of Parallen and Distributed Computing**, [S. l.], 2012.
- Liu, Y.; Pang, Z.; Karlsson, M.; Gong, S. Anomaly detection based on machine learning in iot-based vertical plant wall for indoor climate control. **Building and environment**, [S. l.], p. 107–212, 2020.
- Ljubojevic, M.; Simic, M.; Babic, Z.; Zoric, M. Quality of life context influence factors improvement using houseplants and internet of things. In: **Proceedings of the 2016 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)**. [S. l.] Anais [...]: , 2016. p. 107–212.
- Lopes, P. A.; Preguiça, N.; Medeiros, P.; Martins, M. M. icdb: Uma infraestrutura baseada nos clientes para execuÇÃO de desktops virtuais. In: . [S. l.] Anais [...]: , 2017. p. 13–18.

microsoft.com. **Listagem de serviços gratuitos**. 2022. Site. Acesso em: 22.06.2022. Disponível em: portal.azure.com/#home.

microsoft.com. **Listagem de serviços gratuitos**. 2022. Site. Acesso em: 22.06.2022. Disponível em: <https://apps.azureiotcentral.com/home>.

microsoft.com. **O que é a computação em nuvem?** 2022. Site. Acesso em: 21.06.2022. Disponível em: <https://docs.microsoft.com/pt-br/learn/modules/intro-to-azure-fundamentals/what-is-cloud-computing>.

Mistry, S.; Mohammad, S.; Bouguettaya, A. Sequential learning-based iaas composition. **ACM Transactions on the Web**, [S. l.], v. 15, n. 3, p. 1–37, 2021.

Morimoto, C. E. **Servidores Linux guia prático**. 1. ed. : Sul Editores, 2010.

Moura, H. M. **Química inorgânica**. : Editora e Distribuidora Educacional S.A, 2018. 212 p.

Mumtaz, R.; Zaidi, S. M. H.; Shakir, M. Z.; Shafi, U.; Malik, M. M.; Haque, A.; Mumtaz, S.; Zaidi, S. A. R. Internet of things (iot) based indoor air quality sensing and predictive analytic—a covid-19 perspective. **Eletronics**, [S. l.], p. 184–210, 2021.

O'Hara, B.; Patrick, A. 2 ieee 802.11: First international standard for wlans. **IEEE 802.11 Handbook: A Designer's Companion**, [S. l.], 2005.

Oracle. **O que é um Data Lake?** 2023. Site. Acesso em: 25.09.2023. Disponível em: <https://www.oracle.com/br/big-data/data-lake/what-is-data-lake/>.

Pasin, M.; Perini, F. S.; Bazzan, A. L. C. Infraestrutura autônoma descentralizada como suporte para aplicações com demanda elástica. **iSys - Revista Brasileira de Sistemas de Informação**, [S. l.], p. 99–135, 2013.

Perdrix, A.; Parat, S.; Liaudy, S.; Maître, A. Syndrome des bâtiments malsains (sbm). **Revue Francophone des Laboratoires**, [S. l.], p. 67–72, 2005.

Peter, G.; Stonier, A. A.; Gupta, P.; Gavilanes, D.; Vergara, M. M.; Sin, J. L. Smart fault monitoring and normalizing of a power distribution system using iot. **energies**, [S. l.], p. 1–22, 2022.

pewatron.com. **Datasheet-MICS-6814**. 2022. Site. Acesso em: 07.12.2022. Disponível em: https://www.usinainfo.com.br/index.php?controller=attachment&id_attachment=505.

Postolache, O. A.; Pereira, J. D.; Girão, P. S. Smart sensors network for air quality monitoring applications. **IEEE Trans. Instrum. Meas**, [S. l.], p. 3253–3262, 2009.

Ranjan, R.; Thakur, I. A.; Aujla, G. S.; Kumar, N.; Zomaya, A. Y. Energy-efficient workflow scheduling using container-based virtualization in software-defined data centers. **IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS**, [S. l.], v. 16, 2020.

redhat.com. **O que é virtualização?** 2018. Site. Acesso em: 01.06.2022. Disponível em: <https://www.redhat.com/pt-br/topics/virtualization/what-is-virtualization>.

Ribeiro, J. C. J.; Custodio, M. M.; Pereira, D. H. Covid-19: reflexões sobre seus impactos na qualidade do ar e nas modificações climáticas. **Veredas do Direito: Direito Ambiental e Desenvolvimento Sustentável**, [S. l.], p. 265–296, 2020.

Rizik, M. H.; Sayyed, A.; A., H. W.; AlJarrah, B. A. M. I.; Obeid, N.; Y, A. O. Investigation of microsoft azure and amazon web services from users' perspectives. **ijet**, [S. l.], p. 217–240, 2019.

SAMYOUNG, S. **Dust Sensor Module DSM501 Specifications**. 2023. Site. Acessado em: 27.03.2023. Disponível em: <https://www.samyongsnc.com>.

Saúde, M. da. **Poluição atmosférica na óptica do sistema único de saúde: vigilância em saúde ambiental e qualidade do ar**. 2021. Site. Acesso em: 25.10.2022. Disponível em: https://bvsms.saude.gov.br/bvs/publicacoes/poluicao_atmosferica_SUS_saude_-ambiental.pdf.

Secker, B. Development of an iot system for environmental monitoring: Software. **Victoria University of Wellington - School of Engineering and Computer Science**, [S. l.], p. 1–53, 2020.

Sensortec, S. **MiCS 6814 Datasheet**. 2023.

Silva, D. R.; Viana, V. P.; Muller, A. M.; Livi, F. P.; Dalcin, P. Respiratory viral infections and effects of meteorological parameters and air pollution in adults with respiratory symptoms admitted to the emergency room. **Influenza and Other respiratory Viruses**, [S. l.], p. 42–52, 2013.

Soni, D.; Makwana, A. Survey on mqtt: A protocol of internet of things(iot). **ResearchGate**, [S. l.], p. 1–5, 2017.

Sreyes, K.; Xavier, K. A.; Davis, D.; Jayapandian, N. Internet of things and cloud computing involvement microsoft azure platform. **e International Conference on Edge Computing and Applications (ICECAA 2022)**, [S. l.], p. 603–609, 2022.

SUS. **Plano de Dados Abertos**. 2022. Site. Acesso em: 22.10.2022. • :
<https://www.gov.br/saude/pt-br/acao-a-informacao/dados-abertos/pda>.

Tanenbaum, A. S. **Redes de computadores**. : Bookman, 2021.

USP. **FUNGOS**. 2022. Site. Acesso em: 23.10.2022. • : http://www.icb.usp.br/~crpmicol/materiais/apostila_fungos.pdf.

VIGGIANO, G. **Alergia: o que é, quais as reações e como chegar ao diagnóstico**. 2022. Site. Acesso em: 23.10.2022. • : <https://revistagalileu.globo.com/Ciencia/noticia/2017/09/alergia-o-que-e-quais-reacoes-e-como-chegar-ao-diagnostico.html>.

VMware.com. **O que é virtualização?** 2022. Site. Acesso em: 01.06.2022. • :
<https://www.vmware.com/br/solutions/virtualization.html>.

Wall, D.; McCullagh, P.; Cleland, I.; Bond, R. Development of an internet of things solution to monitor and analyse indoor air quality. **Internet of Things**, [S. l.], p. 1–16, 2021.

Walraven, S.; Truyen, E.; Joosen, W. Comparing paas offerings in light of saas development: A comparison of paas platforms based on a practical case study. **Computing**, [*S. l.*], v. 96, n. 96, p. 669–724, 2014.

Wazwaz, A. A.; Amin, K. M.; Semari, N. A.; Ghanem, T. F. Enhancing human activity recognition using features reduction in iot edge and azure cloud. **Decision Analytics Journal**, [*S. l.*], p. 1–8, 2023.

Yi, B.; Wang, X.; Li, K.; Das, S. K.; Huang, M. A comprehensive survey of network function virtualization. **Computer Network**, [*S. l.*], p. 212–262, 2018.

APÊNDICE A - CÓDIGO FONTE GRAVADO NO MICROCONTROLADOR ESP8266

Figura 57 – Código-fonte em C gravado no ESP8266 - Parte 01.

```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license.

#include <ESP8266WiFi.h>
#include «src/iotc/common/string_buffer.h"
#include «src/iotc/iotc.h"
#include «DHT.h"

/* INCLUIR CONFIGURAÇÕES DOS DEMAIS SENSORES */
#define DHTPIN D1
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

//CONFIGURAÇÃO SENSOR GASES E MUX
// Mapeamento dos pinos
#define SIG A0 // pino analógico
#define S0 D8
#define S1 D7
#define S2 D6
#define S3 D5
// o pino EN do mux vai no GND

const int S_analog = 1023.0; //PARA MAPEAR O MAX VALUE LIDO EM
1024

int mix[4]={S3,S2,S1,S0}; //define os pinos digitais usados

int valores_analogicos[4];

// Valores em binario para a leitura dos valores
byte ci[4][4]={

    {0,1,0,1}, // 5 em decimal
    {0,1,1,0}, // 6 em decimal
    {0,1,1,1}, // 7 em decimal
    {1,0,0,0}, // 8 em decimal

};

//FIM CONFIGURAÇÃO DO SENSOR DE GASES E DO MUX
```

Fonte: elaborado pelo autor.

Figura 58 – Código-fonte em C gravado no ESP8266 - Parte 02.

```

int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;

float voMeasure = 0;
float calcVoltage = 0;
float dustDensity = 0;

//INICIO DSM501A

unsigned long duration1;
unsigned long duration2;
unsigned long duration3;

unsigned long starttime;
unsigned long sampletime_ms = 1000;//tempo de amostra em 1s ;
unsigned long lowpulseoccupancy1 = 0;
unsigned long lowpulseoccupancy2 = 0;
unsigned long lowpulseoccupancy3 = 0;
float ratio1 = 0;
float ratio2 = 0;
float ratio3 = 0 ;
float part25 = 0;
float part10 = 0;
float fumaca = 0;

//FIM DSM501A

/* CONFIGURAÇÃO DA REDE WIFI*/
#define WIFI_SSID "BRK"
#define WIFI_PASSWORD "rstm*26_vt"

```

Fonte: elaborado pelo autor.

Figura 59 – Código-fonte em C gravado no ESP8266 - Parte 03.

```

/* ESTES DADOS DO AZURE - IoT Hub ou Aplicativo IoT Central */

/* Config conexao com a nuvem */
const char* SCOPE_ID = "0ne00987E8F"; //id do escopo AZURE
const char* DEVICE_ID = "NodeMcu"; //id do dispositivo no AZURE
const char* DEVICE_KEY =
"mu9qFVjBIUCK6OJkCUkOMSdu39OyjB1rPJ4PhhnAe9s="; //chave de
conexao do Azure

void on_event(IOTContext ctx, IOTCallbackInfo* callbackInfo);
#include «src/connection.h"

void on_event(IOTContext ctx, IOTCallbackInfo* callbackInfo) {
//configuração e verificação da conexão com o broker MQTT
// Status da conexao
if (strcmp(callbackInfo->eventName, "ConnectionStatus") == 0) {
    LOG_VERBOSE("Is connected ? %s (%d)",
        callbackInfo->statusCode == IOTC_CONNECTION_OK ? "YES"
: "NO",
        callbackInfo->statusCode);
    isConnected = callbackInfo->statusCode == IOTC_CONNECTION_OK;
    return;
}

// Se payload(valor útil) do buffer não possuir final nulo (dados da
callback)
// adiciona outro buffer nulo para enviar mais dados
AzureIoT::StringBuffer buffer;
if (callbackInfo->payloadLength > 0)
{
    buffer.initialize(callbackInfo->payload, callbackInfo->payloadLength);
}

LOG_VERBOSE("- [%s] event was received. Payload => %s\n",
    callbackInfo->eventName, buffer.getLength() ? *buffer :
"EMPTY");

if (strcmp(callbackInfo->eventName, "Command") == 0) {
    LOG_VERBOSE("- Command name was => %s\r\n", callbackInfo->tag);
}
}

```

Fonte: elaborado pelo autor.

Figura 60 – Código-fonte em C gravado no ESP8266 - Parte 04.

```
void setup() {  
  Serial.begin(9600);  
  
  pinMode(D0, OUTPUT);  
  
  dht.begin();  
  
  //inicio pino do mux e sensor de gases  
  pinMode(S0,OUTPUT);  
  pinMode(S1,OUTPUT);  
  pinMode(S2,OUTPUT);  
  pinMode(S3,OUTPUT);  
  pinMode(SIG, INPUT);  
  //fim pino do mux e sensor de gases  
  
  connect_wifi(WIFI_SSID, WIFI_PASSWORD);  
  connect_client(SCOPE_ID, DEVICE_ID, DEVICE_KEY);  
  
  if (context != NULL) {  
    lastTick = 0; // define tempo para inicializar a primeira leitura  
  }  
  
}
```

Fonte: elaborado pelo autor.

Figura 61 – Código-fonte em C gravado no ESP8266 - Parte 05.

```

void loop() {

    //LEITURA DHT22
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    // LEITURA SENSOR DE PARTÍCULAS

    duration1 = pulseIn(D2, LOW);
    duration2 = pulseIn(D3, LOW);
    duration3 = pulseIn(D4, LOW);

    lowpulseoccupancy1 = lowpulseoccupancy1+duration1;
    lowpulseoccupancy2 = lowpulseoccupancy2+duration2;
    lowpulseoccupancy3 = lowpulseoccupancy3+duration3;

    ratio1 = lowpulseoccupancy1/(samptime_ms*10.0); // Integer
    percentage 0=>100
    part25 = 1.1*pow(ratio1,3)-3.8*pow(ratio1,2)+520*ratio1+0.62; // using
    spec sheet curve

    ratio2 = lowpulseoccupancy2/(samptime_ms*10.0); // Integer
    percentage 0=>100
    part10 = 1.1*pow(ratio2,3)-3.8*pow(ratio2,2)+520*ratio2+0.62; //

    ratio3 = lowpulseoccupancy3/(samptime_ms*10.0); // Integer
    percentage 0=>100
    fumaca = 1.1*pow(ratio3,3)-3.8*pow(ratio3,2)+520*ratio3+0.62; //

    //LEITURA MUX DO SENSOR DE GASES
    multiplexGases();

```

Fonte: elaborado pelo autor.

Figura 62 – Código-fonte em C gravado no ESP8266 - Parte 06.

```

if (isConnected) {

    unsigned long ms = millis();
    if (ms - lastTick > 1800000) { // config para enviar os dados a cada 30
MINUTOS
        char msg[64] = {0};
        int pos = 0, errorCode = 0;

        lastTick = ms;
        if (loopId++ % 2 == 0) { // envia dados de telemetria

            //inicio dados DHT22
            pos = snprintf(msg, sizeof(msg) - 1, "{\"Temperatura\": %.2f}",t);
            errorCode = iotc_send_telemetry(context, msg, pos);

            pos = snprintf(msg, sizeof(msg) - 1, "{\"Umidade\": %.2f}",h);
            errorCode = iotc_send_telemetry(context, msg, pos);
            //fim dados DHT22

            //INICIO LEITURA DO SENSOR DE GASES
            for (int x = 0; x <= 4; x++)
            {
                float gas =0;

                Serial.print("X:");
                Serial.println(x);
                Serial.print("valor:");
                Serial.println(valores_analogicos[x]);

                if(x==0) //mapeamento de CO
                {
                    gas = map(valores_analogicos[x],0, S_analog, 1, 1000);
                    pos = snprintf(msg, sizeof(msg) - 1, "{\"CO\": %.2f}",gas);
                    errorCode = iotc_send_telemetry(context, msg, pos);
                }
            }
        }
    }
}

```

Fonte: elaborado pelo autor.

Figura 63 – Código-fonte em C gravado no ESP8266 - Parte 07.

```

else if(x==1) //Mapeamento de NH3
{
    gas = map(valores_analogicos[x],0, S_analog, 1, 300);
    pos = snprintf(msg, sizeof(msg) - 1, "{\"NH3\": %.2f}",gas);
    errorCode = iotc_send_telemetry(context, msg, pos);
}
else if(x==2)//mapeamento de NO2
{
    gas = map(valores_analogicos[x],0, S_analog, 0.05, 10);
    pos = snprintf(msg, sizeof(msg) - 1, "{\"NO2\": %.2f}",gas);
    errorCode = iotc_send_telemetry(context, msg, pos);
}
else if(x==3) //Valores sensor partículas
{
    pos = snprintf(msg, sizeof(msg) - 1, "{\"Poeira\": %.2f}",
valores_analogicos[x]);
    errorCode = iotc_send_telemetry(context, msg, pos);
}
}
//FIM SENSOR GASES
else
{ // send property
}

msg[pos] = 0;

if (errorCode != 0) {
    LOG_ERROR("Sending message has failed with error code %d",
errorCode);
}
}

iotc_do_work(context); // do background work for iotc
}
else
{
    iotc_free_context(context);
    context = NULL;
    connect_client(SCOPE_ID, DEVICE_ID, DEVICE_KEY);
}
}
}

```

Fonte: elaborado pelo autor.

Figura 64 – Código-fonte em C gravado no ESP8266 - Parte 08.

```
//=====FUNÇÕES=====
void multiplexGases(){

    for(int x=0; x <= 3; x++){
        for (int y = 0; y <= 3; y++){
            digitalWrite(mix[y], ci[x][y]);
        }

        if(x < 3)
        {
            valores_analogicos[x] = analogRead(SIG);
        }
        else if (x == 3)
        {
            digitalWrite(D0, LOW);
            delayMicroseconds(samplingTime);

            voMeasure = analogRead(SIG);
            //Serial.print("voMeasure: ");
            //Serial.println(voMeasure);

            delayMicroseconds(deltaTime);
            digitalWrite(D0, HIGH);
            delayMicroseconds(sleepTime);

            calcVoltage = voMeasure * (5.0 / 1024.0); //mapeia 5v para valores
entre 0 e 1023
            dustDensity = 170 * calcVoltage -0.1 ;
            valores_analogicos[x] = dustDensity;
            Serial.print("PoeiraMux: ");
            Serial.println(valores_analogicos[x]);
        }
        delay(1000);
    }
} // fim do multiplex
```

Fonte: elaborado pelo autor.

APÊNDICE B - DADOS DE LEITURAS FEITAS PELO DISPOSITIVO IOT ARMAZENADOS NO SERVIÇO BLOB STORAGE

Quadro 8 – Dados lidos em 27 de julho de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	19,50	52,00	3,00	0,00	2,00	6,00	3,00
08:30	20,60	51,80	3,00	0,00	2,00	6,10	3,20
09:00	21,00	51,70	3,20	0,00	2,00	6,20	3,20
09:30	21,30	51,40	3,20	0,00	2,00	6,20	3,30
10:00	21,90	51,20	3,20	0,00	2,00	6,20	3,40
10:30	22,70	50, 90	3,30	0,00	2,00	6,30	3,40
11:00	22,80	50,30	3,30	0,00	2,00	6,30	3,40
11:30	23,20	49,90	3,30	0,00	2,00	6,40	3,50
12:00	23,70	49,70	3,40	0,00	2,00	6,50	3,50
12:30	24,30	49,50	3,50	0,00	2,00	6,70	3,60
13:00	24,80	49,10	3,60	0,00	2,00	6,70	3,70
13:30	25,20	48,80	3,60	0,00	2,50	6,90	3,70
14:00	25,40	48,6	3,70	0,00	2,50	7,00	3,70
14:30	25,70	48,30	3,70	0,00	2,50	7,10	3,80
15:00	26,20	48,00	3,80	0,00	2,50	7,20	3,80
15:30	26,90	47,80	3,90	0,00	2,50	7,30	3,90
16:00	27,10	47,80	3,90	0,00	2,00	7,40	4,00
16:30	26,90	47,50	4,00	0,007	2,00	7,50	4,10
17:00	26,80	47,40	4,00	0,00	2,00	7,70	4,20
Média	24,00	49,54	3,51	0,00	2,13	6,72	3,60

Fonte:Elaborado pelo Autor.

Quadro 9 – Dados lidos em 28 de julho de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	17,20	53,40	3,00	0,00	2,00	6,10	3,00
08:30	17,80	53,10	3,00	0,00	2,00	6,10	3,00
09:00	18,30	53,00	3,00	0,00	2,00	6,20	3,20
09:30	19,20	52,70	3,10	0,00	2,00	6,30	3,20
10:00	19,90	51,90	3,10	0,00	2,00	6,30	3,20
10:30	20,60	51,30	3,20	0,00	2,00	6,50	3,50
11:00	21,50	50,50	3,20	0,00	2,00	6,50	3,50
11:30	21,10	50,10	3,30	0,00	2,00	6,50	3,50
12:00	21,70	49,80	3,30	0,00	2,00	6,60	3,40
12:30	22,80	49,70	3,50	0,00	2,00	6,70	3,40
13:00	23,30	49,20	3,50	0,00	2,00	7,00	3,20
13:30	24,00	49,00	3,50	0,00	2,00	7,00	3,70
14:00	24,70	48,80	3,60	0,00	2,00	7,20	3,70
14:30	25,50	48,30	4,00	0,00	2,00	7,30	3,70
15:00	26,20	47,10	4,00	0,00	2,00	7,40	3,80
15:30	25,60	46,60	3,50	0,00	2,00	7,30	3,80
16:00	25,90	46,00	3,50	0,00	2,00	7,30	3,70
16:30	25,70	45,90	3,50	0,00	2,00	7,20	3,60
17:00	25,40	45,30	3,50	0,00	2,00	7,20	3,60
Média	22,44	49,56	3,38	0,00	2,00	6,77	3,46

Fonte:Elaborado pelo Autor.**Quadro 10** – Dados lidos em 03 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	18,00	53,9	3,00	0,00	1,00	6,00	3,00
08:30	18,70	53,1	3,00	0,00	1,00	6,00	3,00
09:00	19,30	52,9	3,10	0,00	1,00	6,00	3,00
09:30	19,60	52,9	3,10	0,00	2,00	6,50	3,30
10:00	20,40	52,8	3,20	0,00	2,00	6,50	3,30
10:30	21,00	51,9	3,20	0,00	2,00	6,50	3,40
11:00	21,60	51,4	3,50	0,00	2,00	6,50	3,40
11:30	21,60	50,7	3,50	0,00	2,00	6,50	3,50
12:00	22,40	49,7	3,60	0,00	2,00	6,70	3,50
12:30	22,90	48,4	3,80	0,00	2,00	6,70	3,50
13:00	23,00	47,7	4,10	0,00	2,00	6,80	3,60
13:30	24,70	46,4	4,10	0,00	2,00	7,00	3,70
13:30	25,50	46,1	4,20	0,00	2,00	7,00	3,70
14:00	26,20	46,1	4,20	0,00	2,00	7,00	3,80
14:30	25,60	46,1	4,30	0,00	2,00	7,10	3,80
15:00	25,90	45,6	4,40	0,00	2,00	7,10	3,70
15:30	25,70	44,8	4,30	0,00	2,00	7,20	3,60
16:00	25,40	44,7	4,30	0,00	2,00	7,20	3,60
16:30	25,50	44,3	4,20	0,00	2,00	7,20	3,50
17:00	22,79	48,92	3,74	0,00	1,84	6,71	3,47

Fonte:Elaborado pelo Autor.

Quadro 11 – Dados lidos em 04 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	18,00	56,80	3,00	0,00	2,00	6,00	3,00
08:30	18,30	55,50	3,00	0,00	2,00	6,10	3,20
09:00	18,80	54,60	3,10	0,00	2,00	6,20	3,20
09:30	19,20	54,40	3,20	0,00	2,00	6,30	3,30
10:00	20,00	53,80	3,30	0,00	2,00	6,40	3,40
10:30	20,60	51,90	3,30	0,00	2,00	6,50	3,40
11:00	21,20	51,60	3,40	0,00	2,50	6,50	3,50
11:30	21,90	50,00	3,50	0,00	2,50	6,60	3,50
12:00	22,50	49,80	3,50	0,00	3,00	6,70	3,60
12:30	23,00	49,70	3,70	1,00	3,00	6,80	3,60
13:00	23,60	49,00	4,00	1,00	3,00	6,90	3,70
13:30	24,10	47,40	4,00	1,00	3,00	7,00	3,70
14:00	24,80	46,90	4,30	1,00	3,00	7,10	3,90
14:30	25,10	46,40	4,30	1,00	3,00	7,10	3,90
15:00	25,50	46,00	4,40	1,00	3,50	7,20	4,00
15:30	26,20	45,70	4,50	1,00	3,00	7,20	4,20
16:00	26,70	45,50	4,50	0,00	2,00	7,30	4,20
16:30	26,60	44,40	4,40	0,00	2,00	7,20	4,00
17:00	26,30	43,60	4,40	0,00	2,00	7,20	4,00
Média	22,76	49,63	3,78	0,37	2,50	6,75	3,65

Fonte:Elaborado pelo Autor.**Quadro 12** – Dados lidos em 10 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	18,20	55,50	3,00	0,00	2,00	6,00	3,20
08:30	18,80	54,60	3,20	0,00	2,00	6,20	3,30
09:00	19,40	54,40	3,20	0,00	2,00	6,30	3,40
09:30	20,30	53,80	3,20	0,00	2,00	6,40	3,60
10:00	20,60	53,40	3,30	0,00	2,50	6,50	3,60
10:30	20,80	53,10	3,30	0,00	2,50	6,50	3,50
11:00	21,40	53,00	3,30	0,00	2,50	6,60	3,60
11:30	21,60	52,70	3,50	0,00	2,50	6,60	3,60
12:00	23,30	52,30	3,60	0,00	3,00	6,70	3,70
12:30	23,40	51,80	3,60	0,00	3,00	6,70	3,90
13:00	23,70	51,50	3,90	0,00	3,00	6,90	4,00
13:30	23,90	51,30	3,90	0,00	3,00	6,90	4,00
14:00	24,40	50,50	4,00	0,00	3,50	7,10	4,10
14:30	24,50	50,10	4,00	0,00	3,50	7,20	4,10
15:00	25,10	50,00	4,30	0,00	3,50	7,20	4,20
15:30	25,30	49,80	4,40	0,00	3,00	7,30	4,40
16:00	25,60	49,80	4,50	0,00	3,00	7,40	4,40
16:30	25,90	49,70	4,60	0,00	3,00	7,40	4,40
17:00	26,80	49,70	4,60	0,00	2,50	7,50	4,40
Média	22,79	51,95	3,76	0,00	2,74	6,81	3,86

Fonte:Elaborado pelo Autor.

Quadro 13 – Dados lidos em 11 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	19,80	55,00	3,00	0,00	2,00	6,00	3,00
08:30	20,28	54,80	3,00	0,00	2,00	6,00	3,20
09:00	20,68	54,40	3,20	0,00	2,00	6,30	3,30
09:30	20,74	54,10	3,10	0,00	2,00	6,30	3,40
10:00	21,00	53,90	3,20	0,00	2,00	6,50	3,50
10:30	21,46	53,40	3,30	0,00	2,00	6,60	3,50
11:00	21,48	53,00	3,40	1,00	2,50	6,60	3,60
11:30	21,95	52,80	3,50	1,00	2,50	6,70	3,60
12:00	22,07	52,30	3,60	1,00	3,00	6,80	3,70
12:30	22,44	51,90	3,70	1,00	3,00	6,80	3,80
13:00	23,36	51,90	3,80	1,00	3,50	7,00	3,80
13:30	23,49	51,70	3,90	1,00	3,50	7,00	3,90
14:00	23,91	51,60	4,00	1,00	3,50	7,20	3,90
14:30	24,14	51,30	4,10	1,00	3,50	7,20	4,10
15:00	24,65	51,20	4,20	1,00	3,50	7,30	4,20
15:30	25,34	50,50	4,30	1,00	3,00	7,30	4,20
16:00	25,65	50,10	4,30	0,00	3,00	7,50	4,30
16:30	25,88	50,00	4,40	0,00	3,00	7,50	4,30
17:00	26,77	49,80	4,40	0,00	3,00	7,60	4,40
Média	22,90	52,30	3,71	0,53	2,76	6,85	3,77

Fonte:Elaborado pelo Autor.**Quadro 14** – Dados lidos em 17 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	19,00	51,40	3,00	0,00	2,00	6,00	3,00
08:30	19,70	51,00	3,00	0,00	2,00	6,10	3,00
09:00	19,90	49,90	3,10	0,00	2,00	6,20	3,10
09:30	20,50	49,50	3,10	0,00	2,00	6,20	3,20
10:00	20,70	49,00	3,20	0,00	2,00	6,30	3,20
10:30	21,00	48,80	3,20	0,00	2,00	6,30	3,40
11:00	21,60	48,50	3,30	1,00	2,00	6,50	3,50
11:30	22,10	48,10	3,40	1,00	2,00	6,50	3,60
12:00	22,70	47,70	3,40	1,00	2,00	6,60	3,60
12:30	22,80	47,40	3,60	1,00	2,50	6,70	3,60
13:00	23,20	47,00	3,60	0,00	2,50	6,80	3,70
13:30	23,70	46,90	3,80	0,00	2,50	7,10	3,70
14:00	24,30	46,60	3,80	0,00	2,50	7,10	3,80
14:30	25,00	46,10	3,90	0,00	2,50	7,20	3,80
15:00	25,90	46,00	3,90	0,00	3,00	7,20	3,90
15:30	26,10	45,80	4,00	0,00	3,00	7,40	4,10
16:00	26,40	45,60	4,00	0,00	3,00	7,40	4,10
16:30	26,50	45,30	4,10	0,00	3,00	7,50	4,20
17:00	26,50	45,20	4,10	0,00	3,00	7,50	4,20
Média	23,03	47,67	3,55	0,21	2,39	6,77	3,62

Fonte:Elaborado pelo Autor.

Quadro 15 – Dados lidos em 18 de agosto de 2023.

Hora	Temperatura	Umidade	CO2	NO2	NH3	PM10	PM25
08:00	19,50	52,00	3,00	0,00	2,00	6,00	3,00
08:30	20,60	51,80	3,00	0,00	2,00	6,10	3,20
09:00	21,00	51,70	3,20	0,00	2,00	6,20	3,20
09:30	21,30	51,40	3,20	0,00	2,00	6,20	3,30
10:00	21,90	51,20	3,20	0,00	2,00	6,20	3,40
10:30	22,70	50,90	3,30	0,00	2,00	6,30	3,40
11:00	22,80	50,30	3,30	0,00	2,00	6,30	3,40
11:30	23,20	49,90	3,30	0,00	2,00	6,40	3,50
12:00	23,70	49,70	3,40	0,00	2,00	6,50	3,50
12:30	24,30	49,50	3,50	0,00	2,00	6,70	3,60
13:00	24,80	49,10	3,60	0,00	2,00	6,70	3,70
13:30	25,20	48,80	3,60	0,00	2,50	6,90	3,70
14:00	25,40	48,60	3,70	0,00	2,50	7,00	3,70
14:30	25,70	48,30	3,70	0,00	2,50	7,10	3,80
15:00	26,20	48,00	3,80	0,00	2,50	7,20	3,80
15:30	26,90	47,80	3,90	0,00	2,50	7,30	3,90
16:00	27,10	47,80	3,90	0,00	2,00	7,40	4,00
16:30	26,90	47,50	4,00	0,00	2,00	7,50	4,10
17:00	26,80	47,40	4,00	0,00	2,00	7,70	4,20
Média	24,00	49,54	3,51	0,00	2,13	6,72	3,60

Fonte:Elaborado pelo Autor.