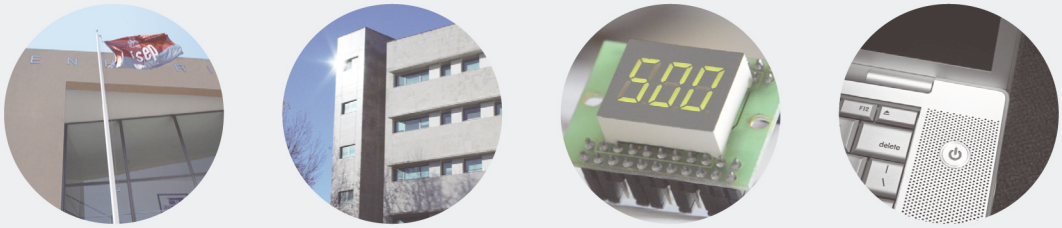
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



AMBIENTE DIDÁTICO NA NUVEM PARA INTEGRAÇÃO DE DISPOSITIVOS IOT

GABRIELE SOTO SANTOS
Agosto de 2024

POLITÉCNICO
DO PORTO

Politécnico do Porto
Instituto Superior de Engenharia do Porto

AMBIENTE DIDÁTICO NA NUVEM PARA INTEGRAÇÃO DE DISPOSITIVOS IOT

Gabriele Soto Santos

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

Agosto, 2024

Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.

Candidato: Gabriele Soto Santos, Nº 1230103, 1230103@isep.ipp.pt

Orientação Científica: João Miguel Leitão, jml@isep.ipp.pt

Coorientação Científica: Alex Lopes de Oliveira, alex.lopes@mackenzie.br

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200–072 Porto

Agosto, 2024

Dedico este trabalho a Deus. Sem ele nada seria possível.

Agradecimentos

A Deus por me proporcionar perseverança durante toda a minha vida.

Quero agradecer à Universidade Presbiteriana Mackenzie (UPM) e todo o seu corpo docente, pela possibilidade e apoio para realização desta Dupla Titulação.

Ao Instituto Superior de Engenharia do Porto (ISEP), toda a diretoria e professores que me ajudaram e desenvolveram durante este processo.

Agradeço aos meus orientadores, João Miguel Leitão e Alex Lopes de Oliveira cuja dedicação e atenção foram essenciais para que este trabalho fosse concluído.

Agradeço aos meus pais e irmã, sua presença e amor incondicional na minha vida sempre. Esta tese é a prova de que os esforços deles pela minha educação não foram em vão e valeram a pena.

Agradeço ao meu namorado que esteve ao meu lado me apoiando durante todo este período.

A todos os meus amigos do curso de graduação que compartilharam dos inúmeros desafios que enfrentamos, sempre com o espírito colaborativo.

Expresso minha gratidão a cada um de vocês, que colaboraram tanto diretamente quanto inderetamente para meu desenvolvimento pessoal, acadêmico e profissional. Muito obrigada por fazerem parte desta jornada.

Resumo

O progresso tecnológico está sempre em evolução, introduzindo uma diversidade de ferramentas e dispositivos que se integram cada vez mais à nossa rotina diária. A computação em nuvem e a *Internet of Things* (IoT) acompanham essa trajetória, abrindo novas oportunidades de conexão e armazenamento de dados. A computação em nuvem possui diversas aplicabilidades, sendo uma delas o auxilio da utilização destes dispositos. Empresas líderes nesse setor, como a *Amazon Web Services* (AWS) disponibilizam uma vasta gama de serviços em nuvem, incluindo recursos para IoT. Esta interconexão entre tecnologias e serviços na nuvem promove maior eficiência e flexibilidade nas aplicações, estimulando a inovação em diversos setores, desde corporativos até os contextos pessoal e educacional.

Esta tese apresenta uma análise sobre a viabilidade da integração desses dispositivos com a plataforma da AWS, delineando de maneira concisa e minuciosa o procedimento para essa integração. Ela destaca a conexão com outros serviços disponíveis na plataforma, bem como diversos parâmetros de aplicabilidade desses serviços.

Palavras-Chave: AWS, IoT Core, S3, Conexão IoT.

Abstract

Technology is always evolving, introducing a variety of tools and devices that are increasingly integrated into our daily routine. Cloud computing and IoT follow this trajectory, opening up new opportunities for connecting and storing data. Cloud computing has several applications, one of which is helping to use these devices. Leading companies in this sector, such as AWS, provide a wide range of cloud services, including IoT resources. This interconnection between cloud technologies and services promotes greater efficiency and flexibility in applications, stimulating innovation in various sectors, from corporate to personal and educational contexts.

This thesis presents an analysis of the possibilities for integrating these devices with the AWS platform, showing in a simple and detailed way the procedure for this integration. It highlights the connection with other services available on the platform, as well as various applicability parameters of these services.

Keywords: AWS, IoT Core, S3, IoT Connection.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

AMQP	*Advanced Message Queuing Protocol*
API	*Application Programming Interface*
ARPANET	*Rede de Agências de Projetos de Pesquisa Avançada*
AWS	*Amazon Web Services*
AZs	*Availability Zones*
CA	*Autoridade Certi- ficadora*
CI/CD	*Continuous Integration/ Continuous Delivery*
CLI	*Command Line Interface*
CoT	*Cloud of Things*
CSR	*Certificate Signing Request*
EPN	*Event Processing Networks*
GCP	*Google Cloud Plataform*
HTTPS	*Hypertext Transfer Protocol Secure*
I2C	*Inter-Integrated Circuit*
IA	*Inteligência Artificial*
IaaS	*Infraestructure as a Service*
IAM	*Identity and Access Management*
ICTAI	*International Conference on Technological Advancements and Innovations*
IDC	*International Data Corporation*
IDE	*Integrated Development Environment*
IEEE	*Institute of Electrical and Electronic Engineers*

IoT	*Internet of Things*
ISEP	Instituto Superior de Engenharia do Porto
JSON	*JavaScript Object Notation*
LoRaWAN	*Long Range Wide Area Network*
M2M	*Machine to Machine*
MQTT	*Message Queuing Telemetry Transport*
MSK	*Managed Streaming para Kafk*
NFC	*Near Field Communication*
NIST	*National Institute of Standards and Technology*
NoSQL	*No Structured Query Language*
NTC	*Negative Temperature Coefficient*
OASIS	*Organization for the Advancement of Structured Information Standards*
PaaS	*Plataform as a Service*
QoS	*Quality of Service*
S3	*Simple Storage Service*
SaaS	*Software as a Service*
SPI	*Serial Peripheral Interface*
SQL	*Structured Query Language*
SSID	*Service Set Identifier*
SSL	*Secure Sockets Layer*
TI	*Tecnologia da Informação*
TLS	*Transport Layer Security*
UART	*Universal Asynchronous Receiver/Transmitter*
UPM	*Universidade Presbiteriana Mackenzie*
WSS	*Web Security Service*
YAML	*YAML Ain't Markup Language*

Capítulo 1

Introdução

O avanço tecnológico vem em uma crescente exponencial e ocorre em diversos âmbitos. Com este crescimento, ferramentas e aparelhos novos são disponibilizados no mercado, e cada vez mais, são utilizados em nosso dia a dia. Neste novo contexto em que vivemos, a disputa por lançamentos de novas tecnologias é progressiva e acelerada, para auxiliar esse crescimento são desenvolvidos diariamente ferramentas e *softwares*. A computação em nuvem é uma das tecnologias que vem acompanhando esse crescimento, a fim de auxiliar e facilitar trabalhos como, infraestrutura de *Tecnologia da Informação* (TI), criação de *softwares*, plataformas, além armazenamento de dados, entre outras milhares de funções [1].

Aparelhos eletrônicos ligados à internet, conhecido como aparelhos de *Internet of Things* (IoT), estão em ascensão devido à crescente tecnológica. São dispositivos conectados à internet que permitem conexão entre si e entre sistemas, podendo armazenar informações nas redes ou utilizá-las para tomadas de decisão. Os dispositivos IoT podem ser feitos de diversos materiais e são aplicados em diversas funções, para isto eles usam sensores, chips, antenas, rastreadores, entre outros aparelhos [2]. Com esta estrutura, ele é válido para coleta e geração de dados, a partir destes dados coletados é feito uma análise para que ocorra uma tomada de decisão a depender do uso deste aparelho. De forma geral estes aparelhos são utilizados como sistemas de controles e redes de comunicações.

Para tecnologia de comunicação são utilizados o wi-fi, bluetooth ou até mesmo o recente *Near Field Communication* (NFC). Por conta disto alguns deles possuem alcances limitados, não só em relação a comunicação, mas também em relação ao

armazenamento de seus dados coletados. Alguns dispositivos utilizam apenas dados instantâneos que não precisam ser armazenados, porém outros necessitam de memórias ou locais de sistema de controles, onde estes dados serão processados e armazenados. Muitas vezes estes locais de armazenamento podem ser compartilhados entre outros aparelhos formando assim uma rede de aparelhos e dispositivos, o que os tornam cada vez mais úteis e eficientes. Tais aplicações de aparelhos que utilizam um sistema de controle compartilhado, são aparelhos usados em *smart cities*, *smart homes*, controles industriais, de plantações, entre outras aplicabilidades.

A computação em nuvem é uma nova tecnologia desenvolvida para diversas funcionalidades que envolvem TI. De forma geral a *cloud computing* (computação em nuvem) são recursos de TI disponibilizados como forma de serviços na internet remotamente [3]. Estes serviços disponibilizam basicamente três tipos de computação, sendo elas a *Infraestructure as a Service* (IaaS), *Plataform as a Service* (PaaS) e *Software as a Service* (SaaS). É dividido também o modelo de implementação necessário, sendo eles quatro modelos, as nuvens públicas, nuvens privadas, nuvens híbridas e nuvens comunitárias. Com estas divisões pessoas e empresas conseguem contratar um ou mais destes serviços e utilizá-los para o apoio e desenvolvimento de seu produto [4].

Os serviços de computação em nuvem são usados para reduzir custos e tempo utilizado para processos de TI, assim pessoas e empresas conseguem disponibilizar mais seus esforços em resultados e decisões finais, do que em algo que pode ser "terceirizado" [5]. A computação em nuvem disponibiliza diversos recursos computacionais de uma forma agrupada, para que todas as necessidades possam ser atendidas em um só lugar, além de conseguir atender demandas diversificadas. A computação vem se desenvolvendo conforme as necessidades e tecnologias vem crescendo, de forma que consegue se adaptar às demandas atuais. Isto gera uma competição de mercado entre as desenvolvedoras desta tecnologia, que cada vez mais buscam aprimorar seus serviços para expansão no mercado.

A *Amazon Web Services* (AWS) é uma das grandes empresas que disponibilizam o serviço de computação em nuvem, sendo a pioneira neste ramo. Em sua plataforma pode-se encontrar serviços de IaaS, PaaS e SaaS, o que a torna atraente para diversas aplicações, desde em grandes empresas, *startups* e até mesmo para uso educacional ou pessoal. A AWS é uma plataforma de serviços em nuvem abrangente e escalável que oferece uma variedade de serviços, incluindo computação, armazenamento, banco de dados, análise, inteligência artificial e IoT [6].

A plataforma da Amazon disponibiliza diversos serviços para serem aplicados juntos a dispositivos IoT, sendo os três principais: *AWS IoT Core*, *AWS IoT Analytics* e *AWS IoT Greengrass*. Estes serviços direcionados a dispositivos IoT são todos relacionados ao gerenciamento dos aparelhos e o relacionamento deles com os dados

e seguranças necessárias para suas aplicações. Os serviços disponibilizados para aparelhos IoT podem ser usados de forma separada ou em conjunto com outros serviços disponibilizados na plataforma, o que torna o site mais adaptável a cada aplicação e necessidade pessoal ou profissional.

1.1 Contextualização

O conceito de IoT, surgiu em 1999 e foi criado por um pesquisador britânico Kevin Ashton [7]. Desde então, com o impulsionamento da internet e de novas tecnologias, os aparelhos IoT vem tomando espaço no mercado e no cotidiano das pessoas. Cada vez mais é notório aplicações utilizadas no cotidiano que possuem tal tecnologia [8]. Com essa mudança de hábitos e consequentemente do mercado, as empresas e universidades buscam desenvolver novas aplicações e soluções. Sendo de extrema importância o seu desenvolvimento para a criação de novas soluções em áreas médicas, ou até mesmo de segurança, além de outras diversas aplicações.

Os espaços *smart* (inteligentes) são capazes de executar diversas tarefas que possuem importância para nós em diversos âmbitos. Produtos *smarts* serão responsáveis por futuras oportunidades de negócio e investimento. Segundo [9], para os próximos anos existem tecnologias e tendências emergentes com maior potencial para redefinir e transformar o mercado. Estas tecnologias são divididas em quatro principais temas, sendo eles: o mundo inteligente, revolução na produtividade, segurança transparente e universal, e facilitadores essenciais.

O desenvolvimento e criação de aparelhos IoT vem se dando de uma forma exponencial com o passar do tempo, o mercado acredita que aparelhos voltados a *Machine to Machine* (M2M) e IoT só tendem a crescer conforme observado na figura 1.1. Estima-se que até 2025 cerca de 27 bilhões de dispositivos estejam conectados [10]. A área de desenvolvimento de aparelhos e soluções IoT vem recebendo cada vez mais investimentos no âmbito corporativo e acadêmico.

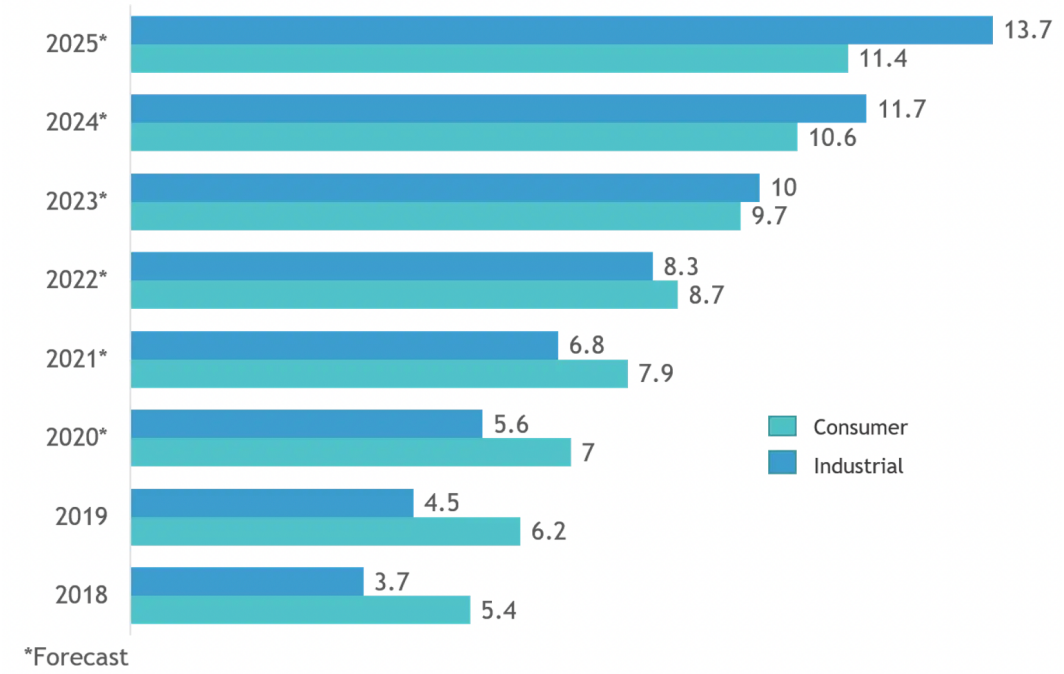


Figura 1.1: Número de objetos industriais e de consumo conectados à IoT, em bilhões [8]

O crescente número de aparelhos conectados a sistemas inteligentes que podem compartilhar, processar, armazenar e analisar dados entre si terá como resultado a conexão de bilhões de máquinas e outros dispositivos a redes e a criação de ainda mais dados. Dessa forma, são necessárias técnicas inteligentes de gestão e análise de dados para extrair *insights* significativos [11]. É fundamental o desenvolvimento de diversos setores associados à tecnologia, tais como telecomunicações, serviços de computação em nuvem e análise de dados (*Analytics*).

Big Data é um conceito muito utilizado atualmente, devido a grande quantidade de dados que são gerados. Diariamente são gerados cerca de 2,5 quintilhões de dados [12]. Estes dados gerados são indispensáveis para organizações, por isso saber obter e avaliar estes dados é algo indispensável.

À medida que o número de dispositivos IoT aumenta, a quantidade de dados gerados por ele também cresce. A *International Data Corporation* (IDC) estima que em 2025 os dispositivos conectados à IoT gerarão cerca de 79,4 zettabytes (ZB), eles projetam que o crescimento anual de dados criados por estes dispositivos será de 28,7% [13].

Estes dados gerados por aparelhos IoT também entram na categoria de metadados. Os metadados possuem um valor significativo, pois podem ser utilizados para informar sistemas inteligentes, orientar a personalização e contextualizar conjuntos de dados ou cenários aparentemente aleatórios. Em suma, os metadados são candidatos primordiais para serem inseridos em bancos de dados *No Structured*

Query Language (NoSQL), a fim de conferir estrutura a conteúdos não estruturados ou alimentar sistemas cognitivos, resultando em níveis avançados de compreensão, inteligência e organização em ambientes aparentemente caóticos [13].

1.2 Definição do Problema

O desenvolvimento tecnológico anda em uma crescente exponencial. Cada vez mais a busca por novas tecnologias vem dominando o mercado e a realidade atual. Estudantes, pesquisadores e empresas dedicam seus esforços e conhecimentos para o aprimoramento tecnológico em diversas áreas. As universidades são grandes polos para a criação destas novas tecnologias, pois buscam desenvolver e preparar alunos para o mercado que demandam destas inovações.

Os cursos voltados para áreas de tecnologia, informática, engenharia, ciência entre outros, incentivam a inovação. Estas, que são comumente implementadas nas universidades, ocorrem por meio de dispositivos IoT. Porém, muitos alunos que implementam estes dispositivos não possuem conhecimento de computação em nuvem ou de ferramentas voltadas à análise e processamento dos dados, fazendo com que o processo de aprendizagem se perca em uma etapa que é essencial para o mercado de trabalho, a coleta e o tratamento destes dados.

Além de que a otimização do tempo de coleta e tratamento de dados é de extrema importância, porém demandam conhecimentos prévios de práticas já concebidas. Para isto este trabalho propõe a utilização da computação em nuvem para criar a prática de gerenciamento de dados nos alunos e prepará-los para os requisitos do mercado de trabalho [14].

A escolha da plataforma mais adequada ao meio profissional atual é essencial. Portanto, a plataforma AWS foi a selecionada, por ser a mais utilizada e conciliar eficiência a um preço mais atrativo em relação as outras plataformas [15], [16].

O presente trabalho busca implementar e viabilizar a utilização do sistema AWS em ambientes universitários. Trazer para os alunos a orientação necessária para conectar um ou uma série de dispositivos à plataforma. Fazer com que os dados sejam coletados e armazenados simultaneamente, de forma a acelerar o processo do tratamento de dados, e consequentemente, ambientar os alunos à plataforma.

As práticas realizadas levaram em consideração a realidade dos alunos do curso de engenharia da *Universidade Presbiteriana Mackenzie* (UPM). Podendo vir a ser implementado por alunos e professores de outras universidades.

1.2.1 Objetivos

Desenvolver um ambiente na nuvem que permita a fácil integração de dispositivos de IoT, para fins didáticos nas disciplinas relacionadas ao tema oferecidas no curso

de Engenharia Elétrica da Escola de Engenharia da UPM e do Instituto Superior de Engenharia do Porto (ISEP), podendo ser replicado em outras universidades.

- Levantamento do estado da arte no âmbito da análise das plataformas disponíveis para aplicação;
- Desenvolvimento de uma metodologia de comunicação em nuvem que possa ser replicada em diversos contextos de aplicação IoT;
- Criação de uma infraestrutura de comunicação que possa ser utilizada de forma facilitada por alunos e professores das instituições envolvidas;
- Preparação de um vídeo explicativo que facilite a utilização da infraestrutura de comunicação disponibilizada por utilizadores sem conhecimentos avançados de computação em nuvem.

1.3 Plano de Trabalho

Para o desenvolvimento desta tese a princípio foi definido o tema geral que seria abordado. Após esta definição foi realizado pesquisas e estudos sobre o tema e feito o levantamento de estudos que deram embasamento para o desenvolvimento da tese e para a escolha correta e adequada da plataforma utilizada e dos serviços necessários. Com isto, começou a ser realizado os testes para conexão na plataforma escolhida, foram feitos testes de conexão com aparelhos IoT, testes de conectividade dos dados enviados para os serviços dentro da plataforma e por fim testes de conexão utilizando códigos. Feito todo o desenvolvimento prático da tese, foi realizado o desenvolvimento da parte teórica e escrita afim de detalhar e documentar todo o desenvolvimento feito durante este período.

1.4 Organização da Dissertação

Este relatório encontra-se estruturado em seis capítulos:

- No capítulo 1, “Introdução”, contextualiza-se o trabalho desenvolvido, realçando a necessidade da utilização de computação em nuvem nas universidades, apresentam-se os principais objetivos e a metodologia seguida.
- O capítulo 2, “Estado da Arte”, apresentação dos dispositivos IoT, suas aplicações e funções na sociedade atual. Os conceitos fundamentais associados a *Cloud Computing*, a sua definição, as características que o tornam inovador, sua arquitetura e os modelos de disponibilização de serviços e implementação. Assim como as plataformas disponíveis e seus respectivos benefícios.

- O capítulo 3, "Tecnologias usadas", apresentação detalhada da plataforma escolhida para o desenvolvimento do projeto, juntamente com as características e especificações escolhidas para a comunicação.
- No capítulo 4, “Metodologia e Desenvolvimento”, inicia-se com a descrição da metodologia adotada para conexão dos dispositivos com a plataforma, após isto a integração de armazenamento dos dados na nuvem, o desenvolvimento do vídeo explicativo para alunos e por fim testes com códigos.
- No capítulo 5, “Análise de Resultados”, apresenta-se a análise feita após as conexões e simulações, observando seus resultados esperados.
- No capítulo 6, “Conclusões e Desenvolvimentos Futuros”, são apresentadas as principais conclusões do trabalho desenvolvido, como também traçadas linhas de investigação para desenvolvimentos futuros.

Capítulo 2

Estado da Arte

Nesta secção, serão explorados os principais temas necessários para o entendimento técnico das ferramentas disponíveis. A fundamentação teórica deste trabalho baseou-se na consulta de obras de renomados autores, através de extensa revisão literária, a fim de garantir que o desenvolvimento ocorresse de maneira exemplar. Considerando a diversidade das áreas abordadas, torna-se imprescindível um entendimento claro e preciso dos tópicos discutidos.

Temas como dispositivos IoT e suas vantagens e importância, computação em nuvem e sua relação com os dispositivos IoT e algumas das maiores plataformas de computação em nuvem atualmente.

2.1 Dispositivos IoT

A importância dos dispositivos IoT vai além de suas funcionalidades práticas, alcançando um impacto transformador nas indústrias e na sociedade como um todo. No contexto urbano, a IoT desempenha um papel vital no desenvolvimento de cidades inteligentes, facilitando a integração de sensores e dispositivos em infraestruturas urbanas que aprimoram a qualidade de vida, segurança pública e sustentabilidade ambiental. Além disso, a IoT é fundamental para impulsionar a quarta revolução industrial, conhecida como Indústria 4.0, ao automatizar processos de produção e habilitar sistemas de manufatura mais flexíveis e eficientes, o que não apenas aumenta a competitividade industrial, mas também possibilita a personalização em massa.

Os dispositivos IoT representam uma confluência crucial entre o mundo físico e o digital, agindo como catalisadores para inovações significativas que promovem eficiência, sustentabilidade e desenvolvimento econômico. Essas tecnologias expandem as possibilidades em diversos campos e indústrias, redefinindo constantemente os limites da tecnologia e interação humana.

No âmbito educacional, a IoT também assume um papel transformador, introduzindo novas demandas que envolvem a digitalização dos processos educativos. Equipamentos e laboratórios dotados de dispositivos IoT facilitam a coleta e análise de dados em tempo real, o que melhora a precisão experimental e expande as possibilidades de descobertas científicas em áreas como engenharia, biologia e química. Essa transformação está alinhada com a chamada Educação 4.0, que visa preparar estudantes para aplicar novas tecnologias na sociedade digital e enfatiza habilidades de criação e inovação.

Contudo, a implementação de IoT na educação enfrenta desafios, incluindo preocupações com a segurança dos dados, privacidade e a necessidade de investimentos substanciais em infraestrutura tecnológica. Além disso, o aprendizado dos conceitos e ferramentas de IoT por parte de docentes e administradores exige programas de treinamento extensivos para garantir que o potencial dessas tecnologias seja plenamente alcançado.

Em resumo, a incorporação crescente de dispositivos IoT nas universidades está moldando uma nova era na educação, onde a eficiência operacional e a experiência de aprendizado são significativamente aprimoradas. No entanto, para que essas tecnologias atinjam seu pleno potencial, é essencial enfrentar os desafios associados à sua implementação, capacitando escolas e universidades para que elas incentivem o desenvolvimento de seus alunos.

Uma abordagem vinculada a este tema envolve o emprego de dispositivos IoT que já estão disponíveis comercialmente, exemplificado por um estudo realizado por Fu et al. [17]. Neste estudo, conduzido em um ambiente de ensino superior, pulseiras inteligentes foram utilizadas em um grupo de controle para facilitar a entrega de microaulas sobre técnicas de basquete. Este método foi comparado com uma turma que recebia aulas tradicionais, visando avaliar melhorias na técnica e no desempenho dos alunos.

Uma abordagem adicional sobre o uso da IoT no contexto educacional se refere ao emprego de práticas pedagógicas que não só utilizam, mas também incentivam a criação de artefatos tecnológicos por parte dos estudantes. Este método é detalhado em um estudo realizado por Tortoriello et al. [18], que descreve uma experiência com alunos do ensino médio. Nesse projeto, os alunos foram envolvidos na elaboração de protótipos que utilizavam tecnologias IoT para abordar desafios ambientais e melhorar a qualidade de vida na comunidade. Além disso, o projeto visava aprimorar competências dos estudantes em disciplinas acadêmicas fundamentais, tais como

matemática, física, biologia e ciências da computação, demonstrando o potencial multidisciplinar da IoT na educação.

2.2 Computação em Nuvem

A *Cloud Computing* teve seu início por volta de 1950 e, em 1960 foi ganhando forma com McCarthy, que discutiu o uso de computadores compartilhados por duas pessoas de forma simultânea, ideia esta que foi chamada de *Utility Computing*. Anos depois a *Rede de Agências de Projetos de Pesquisa Avançada* (ARPANET) foi elaborada com a ajuda de Joseph Carl Robnett. Em 1977 foi realizado pela primeira vez, pelo professor de sistemas de informação, Ramnath Chellappa, o contexto de *Cloud Computing* [19].

O conceito de computação em nuvem possui diversas definições e continua sofrendo alterações, isto ocorre devido às constantes inovações em torno desta tecnologia [20]. Segundo o *National Institute of Standards and Technology* (NIST) a computação em nuvem é um modelo que permite o acesso a recursos de computação de qualquer lugar por meio da rede, possui um conjunto partilhado destes recursos e que podem ser configuráveis e rapidamente provisionado e libertado, demandando o mínimo esforço e interação de gestão e do prestador de serviço [21].

Computação em nuvem é uma tecnologia que disponibiliza recursos de computação através da internet, como uma forma de serviço, possuindo diversas utilizações. TI é um serviço utilizado, sem necessariamente precisar de informações especializadas sobre esta tecnologia e infraestrutura, o serviço é disponibilizado para suportar tais necessidades através da internet. Este conceito, portanto, está sendo utilizado a fim de integrar novas tecnologias, como *software* como serviço, *web* entre outras soluções [14].

A computação em nuvem está cada dia mais presente no cotidiano dos usuários e das empresas, que a utilizam para reduzir custos de aquisição, manutenção, infraestrutura e até mesmo com equipes. Além de melhorar o desempenho no serviço, com uma infraestrutura com maior escalabilidade, confiabilidade e disponibilidade [22].

A integração da computação em nuvem com o IoT facilita uma gestão mais eficaz dos vastos volumes de dados gerados pelos dispositivos IoT, enquanto amplia as capacidades de armazenamento e processamento, essenciais para o tratamento e a análise desses dados.

A fusão, frequentemente referida como *Cloud of Things* (CoT), é reconhecida por potencializar a produtividade e melhorar o desempenho dos sistemas, com sua adoção se expandindo por diversas indústrias. A combinação da IoT com a computação em nuvem cria uma plataforma robusta que não apenas resolve questões

de escala e acessibilidade, mas também abre novas possibilidades para inovações em eficiência energética e automação em larga escala.

Estudos salientam a necessidade de pesquisas contínuas para superar esses obstáculos, visando aprimorar a integração das tecnologias e maximizar seu potencial para aplicações em tempo real, tais como em edifícios sustentáveis, cidades inteligentes, veículos conectados e automação industrial [23]. Portanto, a fusão da IoT com a computação em nuvem é vista não apenas como uma estratégia técnica, mas como uma evolução necessária para o futuro da tecnologia digital e da interação humana com os ambientes urbanos e industriais.

Segundo NIST a computação em nuvem é composta por quatro modelos de implantação e três modelos de serviços, assim como detalhado abaixo:

2.2.1 Modelos de Implementação

A computação em nuvem é uma evolução da internet onde é disponibilizado ferramentas e serviços que permitem a migração de uma computação com infraestrutura física para uma forma virtual [24]. Existem várias formas de fazer a implantação do serviço desejado com base em suas necessidades. Os modelos disponíveis são nuvens privadas, públicas, comunitárias e híbridas [4].

- **Nuvem Privada:** Constitui-se de uma infraestrutura estabelecida para o uso interno. Este tipo de implantação realiza o funcionamento dentro dos *firewalls* de uma organização, o que lhes permite gerir as infraestruturas internas de TI de forma eficaz e fornece serviços para utilizadores locais. A nuvem privada é adequada para empresas que desejam uma maior segurança para suas informações [3].

- **Nuvem Pública:** Uma nuvem pública pode ser gerenciada por uma empresa ou por um meio acadêmico [25]. Seus serviços são escalados e prestados de forma dinâmica através da internet e um fornecedor de terceiros, sendo assim o fornecedor divide os recursos e envia a fatura, que é calculada numa base de utilidade [14].

- **Nuvem Comunitária:** Esta infraestrutura é indicada para consumidores exclusivos de comunidades e organizações que partilham preocupações. Pode ser gerida e operada por mais de uma organização, ou terceiro, ou combinações deles [25].

- **Nuvem Híbrida:** Consiste na junção de nuvem privada e pública, dando liberdade ao consumidor escolher o que é necessário que fique de forma privada e o que pode ser utilizado de forma pública, além de ser a escolha mais popular entre as empresas [3].

2.2.2 Modelos de Serviço

Os serviços de computação disponíveis se dividem em basicamente três categorias para sua utilização e função, sendo eles: IaaS, PaaS e SaaS, todos eles com uma proposta de pagamento apenas em cima daquilo que utilizar [24].

- **Infraestrutura como Serviço:** É um serviço que possui recursos essenciais de computação, como armazenamento e *networking*. Oferece recursos de TI com maior flexibilidade e escalabilidade, ajudando a providenciar novas aplicações. Dá ao usuário maior disponibilidade de serviços, assim ele pode utilizá-lo para diversas atividades. Possui benefícios como redução ou isenção da manutenção do *datacenter* e consequentemente dos custos de hardware e servidores físicos [1].

- **Plataforma como Serviço:** É um ambiente/plataforma de desenvolvimento e implementação completa na nuvem, que disponibiliza diversos recursos, assim o consumidor consegue desenvolver aplicações simples ou complexas. Sendo utilizado para desenvolvimento, testes, implementação, gestão e atualização. Com acesso pela internet de forma segura, o programador consegue desenvolver e gerir as aplicações que deseja. Sendo a parte de infraestrutura e outros recursos de TI de responsabilidade da empresa de computação em nuvem contratada [1].

- ***Software* como Serviço:** O prestador de serviço disponibiliza de forma direta o *software* a ser utilizado, garantindo assim a disponibilidade e a segurança da aplicação e dos dados. Os consumidores contratam aplicações já desenvolvidas pelo prestador, como exemplo: e-mail, calendário, ferramentas de escritório, entre diversos outros *softwares* prontos para utilização [1].

2.3 Plataformas de Computação em Nuvem

A computação em nuvem é um serviço na internet que possibilita ao usuário acessar serviços de qualquer lugar. Este modelo de serviço de TI de forma virtual disponibiliza às empresas mais praticidade, segurança, flexibilidade, disponibilidade de recursos, menores custos, escalabilidade, recuperação de dados, entre diversos outros benefícios [5]. Ao disponibilizar serviços de tecnologia da informação de forma virtual, as plataformas auxiliam as empresas com seus serviços mais práticos, além da praticidade de apenas contratar um serviço pelo qual é usado, o que facilita seu gerenciamento e reduz custos que são significativamente altos para instalação e manutenção de um *datacenter* físico.

Atualmente quatro empresas representam 70% do mercado de *clouding*: Amazon, Microsoft, Alphabet (Google) e Alibaba, conforme a figura 2.1 abaixo. Outros provedores notáveis incluem Tencent, IBM, Salesforce e Oracle [15].

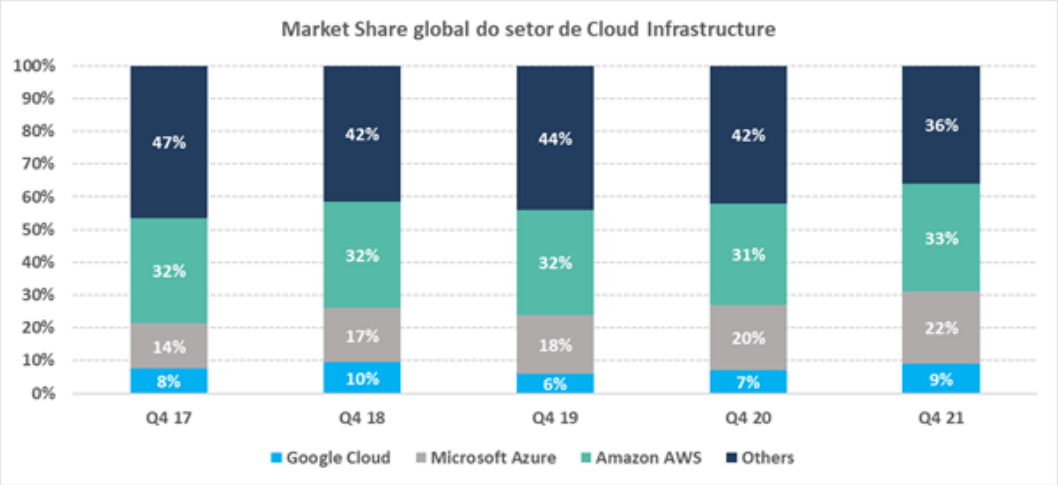


Figura 2.1: Divisão do mercado de empresas de computação em nuvem [15]

No trabalho desenvolvido por Kaushik publicado pela *Institute of Electrical and Electronic Engineers* (IEEE) na *International Conference on Technological Advancements and Innovations* (ICTAI), foi analisado três plataformas, sendo elas a AWS, Azure e *Google Cloud Plataform* (GCP), e fizeram um comparativo de preços de acordo com a demanda nestas três plataformas. A tabela 2.1 demonstra os preços para as instâncias estudadas das plataformas. A análise mostra que entre a maioria das instâncias a AWS possui o menor custo, portanto tem melhor custo-benefício para seus clientes e usuários.

Tabela 2.1: Custo das plataformas de nuvem (por hora) [16].

Tipo de instância	AWS	GCP	Azure
Propósito geral	m6g.xlarge ($0.154)	e2-standard-4 ($0.156)	B4MS ($0.166)
Computação otimizada	c6g.xlarge ($0.136)	c2-standard-4 ($0.235)	F4sv2 ($0.169)
Memória otimizada	r6g.xlarge ($0.202)	m1-ultramem-40 ($6.303)	E4av4 ($0.252)
Computação acelerada	p2.xlarge ($0.90)	a2-highcpu-1g ($3.839)	NC4asT4v3 ($0.526)

2.3.1 Amazon Web Services

A Amazon com seu lançamento da *cloud* AWS em 2006, foi a primeira a entrar no segmento de computação em nuvem, sendo um dos motivos para ter a posição de

liderança neste mercado. Possui uma base consolidada de clientes de todos os setores e tamanhos, como *startups*, corporações e organizações do setor público [1]. Além disto, muitos profissionais de TI que estão no mercado dominam o provedor.

A AWS oferece uma grande quantidade de serviços como tecnologias de infraestrutura, computação, armazenamento e banco de dados, tecnologias emergentes como *machine learning* e *Inteligência Artificial* (IA), *data lakes*, análises e IoT. Este provedor possui alto nível de segurança, contando com mais de 300 recursos de serviços de segurança, além de ser compatível com 98 normas de segurança e certificações de conformidade, e todos os produtos de armazenamento disponibilizados por ele oferecem criptografia de dados [6].

A nuvem da AWS abrange 31 regiões, com 99 zonas de disponibilidade e mais de 410 pontos de presença, com vários *datacenters* espalhados pelo mundo, conforme apresentado na figura 2.2, assim o usuário pode escolher o mais próximo de sua localidade. Cada *datacenter* é instalado em uma zona de disponibilidade chamada *Availability Zones* (AZs), uma delas fica localizada em São Paulo que é a única disponível no Brasil atualmente [26].

Figura 2.2: Mapa de infraestrutura global da AWS [26]

Serviços Disponíveis

A AWS disponibiliza um vasto portfólio de serviços integráveis com dispositivos da IoT, oferecendo soluções escaláveis e robustas para a coleta, processamento, análise e ação sobre os dados gerados por esses dispositivos. Esses serviços facilitam a administração de dispositivos IoT, aprimoram a segurança e permitem uma integração eficiente dos dados em aplicações extensivas. A seguir, apresentam-se alguns dos principais serviços da AWS que se destacam por sua aplicabilidade em implementações IoT:

- ***AWS IoT Core*:** Facilita a conexão de dispositivos IoT à nuvem da AWS, permitindo interações seguras com outras aplicações e dispositivos. Este serviço suporta milhões de dispositivos e bilhões de mensagens, processando e redirecionando essas mensagens para *endpoints* da AWS e outros dispositivos de forma segura e confiável [27].

- ***AWS IoT Analytics*:** Destinado à análise de grandes volumes de dados provenientes de dispositivos IoT, este serviço elimina a necessidade de infraestrutura de hardware externo, fornecendo *insights* detalhados sobre padrões de dados, o que auxilia na otimização de operações e na formulação de estratégias decisivas baseadas em dados concretos [28].

- ***AWS IoT Device Management*:** Auxilia na gestão de dispositivos IoT em larga escala, oferecendo ferramentas para o registro, organização e monitoramento remoto dos dispositivos. Este serviço simplifica a integração e atualização de software dos dispositivos de maneira segura, independentemente de sua localização [29].

- ***AWS IoT Greengrass*:** Permite a execução de computação local, processamento de mensagens e sincronização de dados de forma segura entre dispositivos IoT e a nuvem. Inclui funcionalidades locais do *AWS Lambda*, que permitem aos dispositivos responder rapidamente a eventos locais, operar com latência reduzida e comunicar-se eficazmente com outros dispositivos, mesmo offline [30].

- ***AWS Lambda*:** Facilita a execução de código em dispositivos IoT sem a necessidade de gerenciar servidores, permitindo a execução de funções em resposta a eventos, reduzindo a complexidade e o custo no desenvolvimento de aplicações interativas com dispositivos IoT [31].

- ***AWS IoT 1-Click*:** Este serviço possibilita a criação de gatilhos para ações específicas em dispositivos IoT com apenas um clique, sendo ideal para situações que demandam respostas rápidas, como em emergências ou na automação de tarefas [32].

Estes serviços compõem um ecossistema integrado que não só armazena e processa dados de IoT, mas também habilita análises avançadas e ações automatizadas baseadas nos dados coletados. A integração de serviços como o *AWS IoT Core*, *AWS Lambda* e *AWS IoT Analytics* possibilita às empresas explorar as potencialidades da IoT para transformar seus processos, aumentar a eficiência operacional e explorar novas oportunidades de mercado.

2.3.2 Microsoft Azure

O Microsoft Azure, também conhecido apenas como Azure, foi lançado em 2010 e possui código aberto. É uma plataforma de computação em nuvem operada pela Microsoft que oferece mais de 600 serviços como gerenciamento e desenvolvimento de aplicativos [33].

Os principais recursos do Azure são classificados basicamente em quatro categorias, sendo elas: computação, rede, armazenamento e banco de dados. Entre os serviços disponíveis estão o Azure *active directory*, CDN, *data factory*, *Structured Query Language* (SQL), *function*, cosmosDB, DevOps, *backup*, *machine*, entre outros. A Azure também disponibiliza seus serviços mais populares de forma gratuita por 12 meses, além de ter outros 55 serviços sempre gratuitos, assim os novos clientes conseguem aprender sobre a plataforma de forma gratuita e após esta experiência contratar seus serviços de forma fixa [33].

Serviços Disponíveis

A Microsoft Azure disponibiliza um conjunto diversificado de serviços integrados projetados para otimizar o desenvolvimento e a implementação de soluções para a IoT. Estes serviços proporcionam a conectividade e as ferramentas analíticas essenciais para o gerenciamento eficaz de dispositivos IoT, além de assegurar a segurança, escalabilidade e eficiência na gestão de dados. Seguem alguns dos principais serviços da Azure destinados ao IoT:

- ***Azure IoT Hub*:** Este serviço gerenciado serve como um portal central para comunicação bidirecional entre dispositivos IoT e a nuvem Azure, suportando diversos protocolos de mensagem, como *Message Queuing Telemetry Transport* (MQTT), *Hypertext Transfer Protocol Secure* (HTTPS) e *Advanced Message Queuing Protocol* (AMQP). Ele facilita a coleta de dados e o envio de comandos de maneira segura e escalável [34].
- ***Azure IoT Central*:** Uma solução SaaS que simplifica a configuração e o gerenciamento de aplicações IoT, oferecendo uma interface de usuário intuitiva que permite conectar, monitorar e gerenciar dispositivos sem exigir conhecimento especializado em programação de nuvem. Isso permite às empresas acelerar o desenvolvimento de soluções reduzindo custos e complexidade [35].
- ***Azure IoT Edge*:** Permite o processamento de dados próximo aos dispositivos, no limite da rede, minimizando latência e reduzindo a dependência de conexões contínuas com a nuvem. Integrando-se ao *IoT Hub*, habilita a execução de modelos de IA e análise de dados diretamente nos dispositivos, melhorando as operações IoT [36].

- ***Azure Sphere*:** Focado na segurança, o *Azure Sphere* é projetado para proteger dispositivos IoT em larga escala. Combina microcontroladores certificados, um sistema operacional baseado em Linux otimizado para segurança e um serviço de segurança na nuvem para proteger cada dispositivo durante seu ciclo de vida [37].
- ***Azure Time Series Insights*:** Fornece análise avançada e visualização de dados temporais de dispositivos IoT, auxiliando organizações a identificar tendências, anomalias e padrões para fundamentar decisões estratégicas com base em dados históricos e em tempo real [38].
- ***Azure Stream Analytics*:** Um serviço de processamento de eventos em tempo real que integra dados de dispositivos IoT, oferecendo *insights* instantâneos. É utilizado para filtrar, classificar, agregar e transformar milhões de eventos por segundo de múltiplos dispositivos antes de encaminhar os dados para armazenamento ou aplicações subsequentes [39].

Esses serviços constituem um ecossistema abrangente que apoia desde o desenvolvimento inicial de uma aplicação IoT até sua implementação e administração em larga escala. A incorporação destes serviços na plataforma Azure viabiliza que as empresas desenvolvam soluções IoT robustas, seguras e escaláveis, que são meticulosamente adaptadas às suas necessidades específicas, ao mesmo tempo que exploram as capacidades avançadas de computação em nuvem.

2.3.3 *Google Cloud*

A GCP foi criada em 2008 pela Google com código aberto para seus desenvolvedores, ele possui um *workspace* muito conhecido e utilizado por diversas pessoas e empresas, com diversas aplicações e uso. Possui um alto nível de segurança, utilizando a mesma tecnologia que é aplicada em sites do Google [40]. A nuvem da Google possui diversos serviços, entre os principais estão: os serviços de computação, armazenamento, IA e *machine learning*, banco de dados, análise de dados, rede e ferramentas para desenvolvedores [40].

Serviços Disponíveis

A *Google Cloud* proporciona uma gama completa de serviços projetados para apoiar o desenvolvimento e a gestão de soluções de IoT. Com uma plataforma sólida e integrada, a *Google Cloud* simplifica a conexão, o controle e a análise de dispositivos IoT em uma escala ampla. A seguir, apresentamos alguns dos serviços mais relevantes da *Google Cloud* para IoT:

- ***Google Cloud* Pub/Sub:** Auxilia na captação de fluxos de dados em tempo real provenientes de dispositivos IoT. O Pub/Sub é um serviço de mensagens

assíncronas que separa os produtores de eventos dos consumidores, proporcionando escalabilidade e flexibilidade no gerenciamento desses eventos [41].

- ***Google Cloud Dataflow*:** Serviço de processamento de fluxos contínuos e em lote que permite modificar e enriquecer dados em tempo real, ideal para manipular as informações coletadas de dispositivos IoT. O *Dataflow* facilita a criação de pipelines de dados complexos, essenciais para análises avançadas [42].

- ***Google BigQuery*:** Um armazém de dados completamente gerenciado e escalável, apto para realizar análises rápidas em SQL de grandes conjuntos de dados. *BigQuery* é ideal para executar análises profundas de dados volumosos originados de dispositivos IoT, possibilitando a integração de dados de múltiplas fontes para obtenção de *insights* detalhados [43].

- ***Google Cloud Functions*:** Permite a execução de código em resposta a eventos de qualquer parte da infraestrutura da *Google Cloud*, incluindo as mensagens de dispositivos IoT via *Cloud* Pub/Sub. Este ambiente de execução gerenciado oferece escalabilidade automática e processamento orientado por eventos [44].

- ***Google Cloud Firestore*:** Um banco de dados NoSQL para armazenar e sincronizar dados em tempo real, útil para aplicações IoT que requerem sincronização de estado em tempo real entre dispositivos e a nuvem [45].

Juntos, estes serviços constituem um ecossistema potente que não somente facilita a coleta e o processamento de dados de IoT, mas também promove análises detalhadas e a integração com aplicações e dispositivos. A plataforma *Google Cloud* possibilita que as empresas explorem as capacidades da IoT para transformar suas operações, incrementar a eficiência e gerar novas oportunidades de negócio de maneira segura e escalável.

Capítulo 3

Tecnologias Usadas

3.1 Dispositivos

No presente capítulo, procede-se a exploração minuciosa das tecnologias e conceitos empregados neste projeto, abrangendo tanto os componentes de hardware quanto de software.

Dispositivos como o microcontrolador ESP32, o sensor de umidade DHT11 e os resistores constituem elementos básicos no ramo da prototipagem de sistemas geração e tratamento de dados, pelo seu baixo custo e performance escalonável.

A conjugação desses dispositivos possibilita a elaboração de sistemas funcionais, que se estendem desde a automação residencial e dispositivos portáteis até ao alcance de sistemas industriais de ponta. O domínio sobre a integração e aplicação desses componentes revela-se fundamental para professores e estudantes de eletrônica, e constituiu a base para a realização deste projeto.

3.1.1 Microcontrolador

O ESP32 demonstrado figura 3.1, criado pela Espressif Systems, representa um avanço significativo no campo dos microcontroladores devido ao seu custo reduzido e alta funcionalidade. Este microcontrolador é dotado de uma arquitetura de dois núcleos e oferece uma vasta gama de interfaces de comunicação, como Wi-Fi, Bluetooth, *Universal Asynchronous Receiver/Transmitter* (UART), *Serial Peripheral Interface* (SPI) e *Inter-Integrated Circuit* (I2C) [46], que o tornam extremamente

adaptável para aplicações em IoT. A placa de desenvolvimento Doit ESP32 DevKit v1, que se baseia no ESP32, inclui recursos adicionais como um regulador de tensão, um conector micro-USB para alimentação e programação, botão de reset e um LED indicador de energia, características que simplificam o uso em projetos de IoT devido à integração facilitada de periféricos diversos [47].

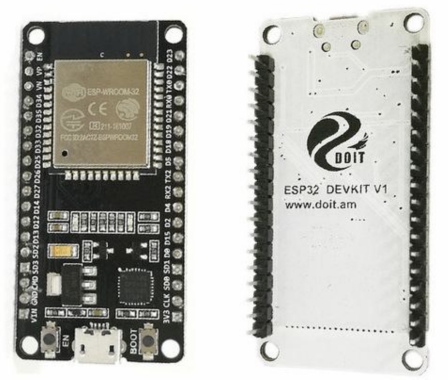


Figura 3.1: Placa ESP32 [47]

Adicionalmente, o ESP32 incorpora os componentes fundamentais de um microcomputador, incluindo memórias voláteis e não voláteis, bem como portas de entrada e saída, adequando-se a projetos que não exigem armazenamento extensivo de dados, como em automações residenciais, prediais, industriais e embarcadas. Equipado com um microprocessador Xtensa Dual-Core 32-bit LX6, o módulo possui memória Flash programável de 4 MB, RAM de 520 KBytes, ROM de 448 KBytes e opera com uma frequência de até 240 MHz. Com 25 pinos digitais e módulos Wireless 802.11 que funcionam na frequência de 2,4 GHz, além de Bluetooth *Low Energy* versão 4.2, o ESP32 é uma escolha robusta para conectividade e desenvolvimento IoT [46].

No contexto de desenvolvimento, tanto o ESP32 com a utilização do Arduino proporcionam plataformas ricas com *Integrated Development Environment* (IDE), onde é possível programar utilizando linguagens como C++, oferecendo ampla liberdade para personalização de *firmware* [48]. Esta capacidade permite que os processadores se dediquem inteiramente aos programas em execução, otimizando a capacidade de processamento e aumentando significativamente a eficiência dos projetos.

Estas tecnologias são ideais para a criação de uma variedade de projetos de IoT, desde sistemas de monitoramento ambiental até dispositivos de controle remoto e sensores de presença. A versatilidade do ESP32, juntamente com sua facilidade de configuração, manutenção e custo reduzido, torna-o frequentemente a escolha preferencial para projetos inovadores de dispositivos IoT. Tais características destacam a importância dessas tecnologias não apenas como ferramentas educacionais ou de prototipagem, mas também como componentes vitais em aplicações industriais e

comerciais, abrindo novas possibilidades para automação e sistemas inteligentes em diversos setores.

3.1.2 Sensor DHT11

O sensor DHT11 é empregado para efetuar aferições de temperatura e humidade, emitindo um sinal digital como resultado. Este dispositivo incorpora um microcontrolador de 8 bits e um resistor *Negative Temperature Coefficient* (NTC), que é responsável pela medição da temperatura. O sensor realiza leituras a cada 2 segundos, operando em uma faixa de temperatura de 0 a 50°C e medindo a humidade relativa do ar em percentual, com uma faixa de 20 a 80% [49].

Este sensor é considerado de baixo custo, funcionando com uma tensão de alimentação de 3 a 5 volts e consumindo uma corrente de 2,5 mA durante as medições. A precisão do sensor é de 5% para humidade e a variação pode alcançar até 2°C para temperatura. As dimensões físicas do sensor são 15,5 mm de altura, 12 mm de largura e 5,5 mm de profundidade, e ele é equipado com quatro pinos. O primeiro pino, da esquerda para a direita, é destinado à conexão da alimentação (Vcc), o segundo transmite os dados (sinal), o terceiro pino não é utilizado, e o quarto é conectado ao terra (*Ground*) [49]. A disposição desses pinos pode ser observada na figura 3.2.

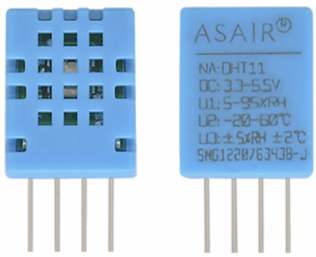


Figura 3.2: Sensor DHT11 [47]

3.1.3 Arduino

O Arduino IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto, amplamente empregado para a programação de microcontroladores da série Arduino, incluindo o ESP32. Este ambiente proporciona uma interface gráfica intuitiva, facilitando o acesso a programadores novatos. O software baseia-se na linguagem C++, utilizando um editor de texto para escrita de código, que pode ser verificado e compilado dentro do próprio IDE, convertendo-o em um arquivo pronto para ser transferido para o microcontrolador via USB [50].

Além disso, o Arduino IDE é enriquecido com uma vasta biblioteca e exemplos práticos disponíveis, contribuindo para uma rápida aprendizagem e implementação

de projetos variados. Essas características tornam o Arduino uma plataforma valiosa para o desenvolvimento de objetos interativos, tanto autônomos quanto conectados a computadores, sem a necessidade de aquisição de licenças.

A flexibilidade do Arduino é ampliada pelo uso de diversas placas de entrada/-saída, que podem ser adquiridas já montadas ou configuradas manualmente, cada uma adaptada para diferentes tipos de projetos [48]. Portanto, o Arduino IDE não apenas serve como uma ferramenta essencial para a programação do ESP32, mas também como um recurso crucial que permite a programadores de todos os níveis desenvolverem e implementarem suas ideias com eficácia e eficiência.

3.2 Serviços utilizados da AWS

A AWS é uma plataforma de serviços em nuvem que se destaca pela oferta extensa e integrada de soluções para infraestrutura digital. Este ecossistema abrangente inclui computação, armazenamento, bancos de dados, análises, redes, aprendizado de máquina, IoT, segurança, identidade e conformidade, entre outras categorias [6]. Disponibilizados sob demanda e com uma estrutura de pagamento conforme o uso, esses serviços permitem a organizações de todos os tamanhos — desde *startups* até grandes empresas e entidades do setor público — crescerem e adaptarem-se rapidamente às mudanças nos requisitos empresariais.

Com mais de 200 serviços disponíveis, a AWS possibilita que novos recursos sejam provisionados de maneira rápida e sem despesas fixas iniciais, ajudando a reduzir custos de TI e aumentar a agilidade operacional. Os benefícios oferecidos pela AWS não se limitam apenas à redução de custos e flexibilidade, mas também incluem a capacidade de escalar globalmente em questão de segundos, destacando-se como uma líder no fornecimento de soluções de nuvem que atendem a uma vasta gama de necessidades empresariais. Esta plataforma é uma ferramenta essencial para empresas que buscam eficiência e inovação tecnológica, permitindo-lhes explorar os avanços em tecnologia de nuvem para otimizar suas cargas de trabalho[51].

3.2.1 *AWS IoT Core*

O *AWS IoT Core* é uma plataforma avançada projetada para a integração e gerenciamento de dispositivos de IoT na computação em nuvem. Esta solução elimina a necessidade de provisionamento e gerenciamento de servidores, facilitando uma conexão segura e eficiente dos dispositivos utilizando uma variedade de protocolos de comunicação, incluindo MQTT, HTTPS, MQTT sobre *Web Security Service* (WSS) e *Long Range Wide Area Network* (LoRaWAN) [27].

O serviço é notável por sua habilidade em gerenciar "*device shadows*"ou sombras de dispositivos, que são representações virtuais dos dispositivos físicos, permitindo a continuidade da interação e gestão dos dispositivos mesmo quando estão offline.

O *AWS IoT* Core pode suportar milhões de dispositivos simultaneamente, salvando seu estado atual como um perfil sombra e atualizando-o continuamente.

A funcionalidade de segurança é uma pedra angular do *AWS IoT Core*, com autenticação robusta e gestão de políticas de segurança utilizando certificados para garantir conexões seguras. Esta segurança reforçada é vital para a integridade e fiabilidade das operações IoT em ambientes industriais e comerciais. O *Core* proporciona uma infraestrutura que suporta criptografia de ponta a ponta, oferecendo elasticidade, escalabilidade e a capacidade de conectar bilhões de dispositivos [52].

Além disso, o *AWS IoT Core* permite aos usuários conectar-se facilmente a outros serviços da AWS, assim como pode ser observado na figura 3.3, como AWS S3, *AWS Lambda*, *AWS Kinesis* e *AWS IoT Analytics*, que fornecem uma gama de capacidades de computação, armazenamento, rede, gerenciamento e análise, permitindo a criação de soluções complexas e interconectadas sem a necessidade de desenvolver infraestrutura do zero. A AWS oferece uma infraestrutura escalável de baixo custo como modelo de serviço, com uma presença global massiva e políticas de preços que incluem um nível gratuito de entrada, cobrança por uso e praticamente nenhuma taxa fixa.

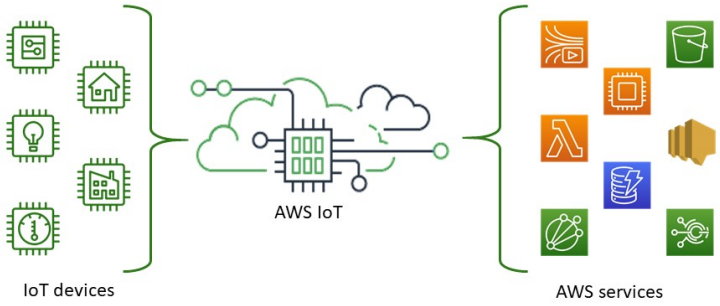


Figura 3.3: Arquitetura AWS IoT [53]

Em resumo, o *AWS IoT Core* apresenta-se como uma solução eficiente e segura para a expansão de aplicações IoT em vários setores, promovendo uma plataforma confiável que suporta a vasta integração de dispositivos IoT em sistemas mais amplos, facilitando a inovação tecnológica em uma escala global. Seu ambiente gerenciado elimina a necessidade de atualização e gerenciamento de servidores, garantindo segurança e confiabilidade na transferência de dados.

3.2.2 Amazon S3

O Amazon *Simple Storage Service* (S3), oferecido pela AWS, constitui-se como uma solução de armazenamento em nuvem notavelmente robusta, desenhada para ser uma opção de armazenamento de objetos altamente escalável e de confiança elevada. Operando sob um modelo econômico de "pague pelo que usar", permite aos usuários

armazenar volumes ilimitados de dados sem encargos iniciais fixos, cobrando exclusivamente pela capacidade de armazenamento efetivamente utilizada. As unidades de armazenamento no S3 são organizadas em "*buckets*", que servem como contêineres principais onde os objetos, isto é, arquivos e pastas juntamente com seus metadados descritivos, são alocados [54].

Tais "*buckets*"proporcionam uma configurabilidade extensiva no que tange ao controle de acesso, possibilitando a definição de permissões tanto no nível do "*bucket*"quanto do objeto individual. Esta adaptabilidade assegura que os dados sejam gerenciados com uma escalabilidade e disponibilidade elevadas, fortalecidas por funcionalidades como controle de acesso robusto, ferramentas de replicação versáteis e uma visibilidade completa para a organização. Ademais, o S3 é reconhecido por sua eficácia em minimizar custos relacionados a investimentos preliminares e ciclos de atualização de hardware, mantendo simultaneamente a segurança dos dados por meio de conformidade e recursos de auditoria avançados [55].

O S3 assegura uma resiliência de dados de 99,9%, adaptando os recursos de armazenamento para satisfazer as necessidades variáveis de seus usuários. Este serviço não apenas substitui as tradicionais unidades de armazenamento físico encontradas nos centros de dados, como também oferece uma alternativa mais econômica e confiável [55]. A funcionalidade de arrastar e soltar para carregar dados, suportada em navegadores como Google Chrome e Mozilla Firefox, simplifica significativamente o processo de transferência de arquivos para os "*buckets*", independente do tipo de arquivo, seja ele musical, cinematográfico, textual, fotográfico ou arquivístico [54].

3.2.3 *Amazon kinesis*

O *Amazon Kinesis Data Analytics* é uma plataforma robusta e integrada projetada para a manipulação e análise de fluxos de dados em tempo real, ideal para implementar modelos de *Event Processing Networks* (EPN). A capacidade do *Amazon Kinesis* de integrar-se com o *Apache Flink* realça sua aptidão para processar eficientemente grandes volumes de dados de diversos tipos, incluindo vídeos, áudios, logs de aplicativos, clickstreams de sites e dados de telemetria de IoT [56]. Este processamento é essencial para a tomada de decisão rápida e informada dentro das empresas, permitindo uma ação imediata baseada em *insights* gerados a partir dos dados analisados.

O *Amazon Kinesis* possui escalabilidade, que permite ajustar os recursos automaticamente conforme a demanda sem intervenção manual, e a flexibilidade, que facilita a conexão com diferentes fontes de dados. É uma ferramenta valiosa para organizações que precisam de um processamento de dados em tempo real, econômico e eficaz, fornecendo a infraestrutura necessária para suportar aplicações de *machine*

learning e análises avançadas [57]. Tais capacidades consolidam o *Amazon Kinesis* como uma solução estratégica para empresas que buscam otimizar processos e melhorar a compreensão operacional através de dados continuamente atualizados.

3.2.4 *Amazon Kinesis Firehose*

O *Amazon Kinesis Data Firehose* é uma plataforma de destaque para aquisição, transformação e entrega de fluxos de dados para lagos de dados, armazéns de dados e serviços de análise de forma rápida e eficiente. Este serviço permite configurar um fluxo com uma origem e um destino específicos, juntamente com as transformações necessárias, processando continuamente os dados, escalando automaticamente com base no volume disponível e entregando-os em questão de segundos [58].

Funciona permitindo que os usuários selecionem a fonte para seu fluxo de dados, que pode ser um tópico no Amazon *Managed Streaming para Kafk* (MSK), um fluxo no *Kinesis Data Streams* ou dados escritos usando a *Application Programming Interface* (API) *Firehose Direct PUT*. O *Firehose* é integrado a mais de 20 serviços AWS, facilitando a configuração de fluxos a partir de fontes variadas como Amazon *CloudWatch Logs*, *AWS WAF web ACL logs*, e outros [58].

Além disso, os usuários podem especificar transformações de dados opcionais, como a conversão de fluxos de dados para formatos como Parquet ou ORC, descompressão de dados ou realização de transformações de dados personalizadas usando funções *AWS Lambda*. Estas transformações permitem o particionamento dinâmico de registros de entrada baseado em atributos para entrega em diferentes localizações [59].

O destino do fluxo pode ser configurado para várias plataformas como Amazon S3, *Amazon OpenSearch Service*, *Amazon Redshift*, entre outros, permitindo uma flexibilidade significativa no gerenciamento e análise de dados [58]. Este sistema não apenas facilita a captura e transformação de dados, mas também destaca-se pela capacidade de reduzir a carga administrativa através de uma gestão simplificada dos fluxos de dados, garantindo alta durabilidade, disponibilidade e segurança na entrega de dados aos destinos configurados [59].

O *Amazon Kinesis Data Firehose* emerge, portanto, como uma solução integral para empresas que buscam otimizar a análise de dados em tempo real, proporcionando um meio econômico, escalável e seguro para integrar e analisar grandes volumes de dados em vários setores industriais e comerciais.

3.2.5 Protocolo IAM

O AWS *Identity and Access Management* (IAM) é um componente crítico do ecossistema AWS, essencial para o gerenciamento eficaz de identidades e acessos. Este serviço confere controle meticuloso sobre os acessos aos recursos e serviços da AWS,

permitindo aos administradores estabelecer e administrar as permissões com exatidão. O IAM viabiliza a criação de políticas de segurança detalhadas, assegurando que somente entidades autorizadas — usuários, grupos e papéis dentro de uma organização — tenham acesso a operações e dados específicos sob condições claramente estabelecidas [60].

Ademais, o IAM é reconhecido como um instrumento crucial para a preservação da segurança dos dados e das operações dentro da AWS, integrando-se a outros serviços para oferecer uma gestão de segurança unificada. É aconselhável adotar as melhores práticas de segurança, como implementar o princípio do privilégio mínimo e realizar auditorias frequentes de direitos de acesso, para potencializar a proteção oferecida pelo IAM aos recursos armazenados na nuvem [61].

3.2.6 CLI

A AWS *Command Line Interface* (CLI) é uma ferramenta que unifica a gestão de serviços da AWS a partir da linha de comando. Ela foi desenvolvida para oferecer uma maneira eficiente e programática de interagir com os serviços da AWS, sendo essencial para automatizar tarefas e integrar serviços em processos de desenvolvimento e operações (DevOps). Compatível com diversos sistemas operacionais, a AWS CLI garante que desenvolvedores e administradores de sistemas possam utilizá-la independentemente da plataforma escolhida.

Um dos principais benefícios da AWS CLI é a capacidade de automatizar tarefas repetitivas através de scripts, permitindo a criação, configuração e gerenciamento de recursos na nuvem AWS de maneira eficiente. A ferramenta suporta múltiplos perfis de usuário, facilitando a gestão de credenciais e configurações para diferentes ambientes, como desenvolvimento, teste e produção. Documentação abrangente e exemplos de uso para cada comando tornam sua aprendizagem acessível mesmo para iniciantes.

A CLI suporta entrada e saída de dados em formatos *JavaScript Object Notation* (JSON) e *YAML Ain't Markup Language* (YAML), facilitando a integração com outras ferramentas de automação e gerenciamento de configuração. Ela também permite que os usuários configurem políticas de segurança detalhadas, definindo permissões específicas para diferentes ações e recursos, garantindo conformidade com as políticas de segurança da organização. A capacidade de scriptabilidade facilita a automação de tarefas, garantindo que processos complexos possam ser executados de forma consistente e reproduzível, essencial para a gestão de infraestrutura em grande escala e práticas DevOps.

Em resumo, a AWS CLI oferece controle detalhado sobre os recursos na nuvem, permitindo a execução de comandos ajustados para atender a necessidades específicas. Pode ser facilmente integrada em *pipelines* de *Continuous Integration/*

Continuous Delivery (CI/CD), contribuindo para ciclos de desenvolvimento mais rápidos e lançamentos de software mais frequentes e confiáveis. Suportando operações em larga escala, a CLI facilita a administração de grandes infraestruturas na nuvem de forma eficiente e eficaz, sendo uma ferramenta essencial para administradores de sistemas, desenvolvedores e engenheiros DevOps.

3.3 Protocolo MQTT

O Protocolo MQTT é reconhecido como um padrão consagrado pela *Organization for the Advancement of Structured Information Standards* (OASIS), especificamente concebido para a IoT. Distintivo por sua eficiência em ambientes onde a simplicidade do código e a economia de largura de banda são primordiais, o MQTT é amplamente aplicado em contextos IoT e de comunicação M2M em uma multiplicidade de setores [62]. Este protocolo, eminentemente projetado para a transmissão de mensagens de caráter leve, opera sobre o modelo de publicação e assinatura, facultando aos clientes alternar entre o papel de publicadores e assinantes. Na posição de publicador, o cliente propaga dados para o intermediário, conhecido como *broker*, sob a forma de tópicos, enquanto que no papel de assinante, ele se inscreve para recepcionar dados emanados de um ou mais tópicos específicos [63].

A Tabela 3.1 mostra o cabeçalho fixo das mensagens MQTT, e o comprimento restante que pode ter entre um e quatro bytes. O primeiro byte contém parâmetros que especificam o tipo da mensagem, indicam se é uma mensagem duplicada, marcam a qualidade do serviço atribuída à mensagem e indicam a retenção, determinando se o broker deve manter essa informação até receber um comando para deletar ou substituir a mensagem [63].

Tabela 3.1: Cabeçalho de mensagem MQTT [62]

Bit	7	6	5	4	3	2	1	0
Byte 1	Tipo da mensagem				Flag DUP	Nível QoS		RETAIN
Byte 2...	Comprimento restante							

A partir do segundo byte, é determinado o "comprimento restante", que indica o tamanho total da mensagem, excluindo o "*Fixed Header*". Este comprimento é codificado como um inteiro de 1, 2, 3 ou 4 bytes, possibilitando extensões adicionais de até 2^{28} -1 Bytes (̃256MB). Nesse byte, são também revelados parâmetros opcionais, como o intervalo máximo permitido para desconexão entre cliente e *broker*, conhecido como "*Keep Alive*", e o tempo predefinido para o envio de uma requisição do tipo Ping, que antecede a desconexão em caso de ausência de comunicação. Além disso, podem estar incluídos contornos da "Última Mensagem de Testamento"(*Last Will Message*) para situações de desligamento abrupto, seguidos pelo conteúdo da

própria mensagem, denominado *payload*. Após o cabeçalho fixo, uma mensagem MQTT pode ainda conter um cabeçalho adicional (*variable header*) e uma seção de dados (*payload*), ambos opcionais e dependentes do tipo de mensagem [63].

O protocolo MQTT é revestido de atributos distintivos quando confrontado com outros protocolos, sendo exaltado como paradigmático para desenvolvimento com dispositivos IoT. Sua adoção é amplamente incentivada para dispositivos remotos, devido sua notória qualidade de serviço, contribui como mecanismos de diagnóstico e avaliação de conectividade, tais como o *Keep Alive* e o *Ping*. Acrescente-se a isso, a sua reputação em segurança e fiabilidade é amplificada pelo emprego de métodos de criptografia avançados, a exemplo do *Transport Layer Security* (TLS) e *Secure Sockets Layer* (SSL).

3.3.1 Fiabilidade do protocolo MQTT

O protocolo em questão estipula três níveis de *Quality of Service* (QoS) [64], as quais são delineadas da seguinte maneira:

QoS0: Nesta camada, a mensagem é expedida unicamente uma vez em direção ao seu destinatário, eliminando a possibilidade de subsequente retransmissão.

QoS1: Neste nível, a mensagem é reiteradamente transmitida até a confirmação de sua receção pelo destinatário ser assegurada, acarretando na eventualidade de recebimento duplicado da mensagem.

QoS2: Neste patamar, a mensagem é despachada de maneira singular com a certeza de seu recebimento, em virtude da implementação de um procedimento de *handshaking* composto por quatro etapas, o qual inclui a autenticação mediante credenciais de usuário e senha.

Deste modo, o MQTT aborda a questão da fiabilidade por meio da retransmissão de pacotes. Este procedimento eleva a fiabilidade da transmissão de dados ao custo de um incremento no consumo dos recursos de processamento e energia. A aptidão do MQTT para administrar de modo eficaz as retransmissões e assegurar a entrega das mensagens mediante distintos níveis de QoS consolida sua posição como uma alternativa robusta para aplicações em IoT que demandam variados patamares de fiabilidade [65].

3.3.2 Segurança da informação do protocolo MQTT

Para que exista segurança nas informações que são transmitidas é utilizado o princípio de confidencialidade, onde somente usuários autorizados podem ter acesso à informação. Outra forma que é utilizada para a confiabilidade é a criptografia das mensagens trocadas entre o emissor e receptor, onde ocorre o embaralhamento das

mensagens enviadas. O protocolo possui também integridade, onde o sistema detecta e impede a modificação ou corrupção dos conteúdos que estão sendo enviados [62].

Estudos não só enfatizam a eficácia inerente ao protocolo MQTT, mas igualmente realçam a necessidade de uma ponderação criteriosa entre a segurança e a performance. Esta dualidade assegura que as implementações no âmbito da IoT preservem a integridade das informações, ao passo que não oneram excessivamente a funcionalidade dos dispositivos. Tais dispositivos são frequentemente caracterizados por suas limitações em termos de energia disponível e capacidade de transmissão de dados, impondo assim a necessidade de uma abordagem equilibrada para a manutenção da sua efetividade operacional. [66].

Capítulo 4

Metodologia e Desenvolvimento

Neste capítulo, são detalhados os procedimentos metodológicos empregados no desenvolvimento do dispositivo IoT, abrangendo desde sua montagem até a elaboração do código para a coleta de dados no arduino. Ressaltando que foi feito a montagem de um dispositivo básico apenas para ensaios experimentais, porém as conexões com a plataforma podem ser feitas através de qualquer dispositivo IoT com as mesmas instruções. Ademais, é apresentado o processo de integração dos dispositivos IoT com a plataforma da AWS, selecionada com base em estudos prévios. Este processo engloba a utilização dos serviços de conexão *AWS IoT Core*, bem como os serviços de armazenamento S3 e *Kinesis Firehose*.

Durante o desenvolvimento do projeto a arquitetura montada será construida conforme a figura 4.1, sendo composta inicialmente pela Esp32 e precedida pelas intermediações e serviços *Iot Core*, *Kineses Firehose* e S3, disponibilizados pela própria plataforma.

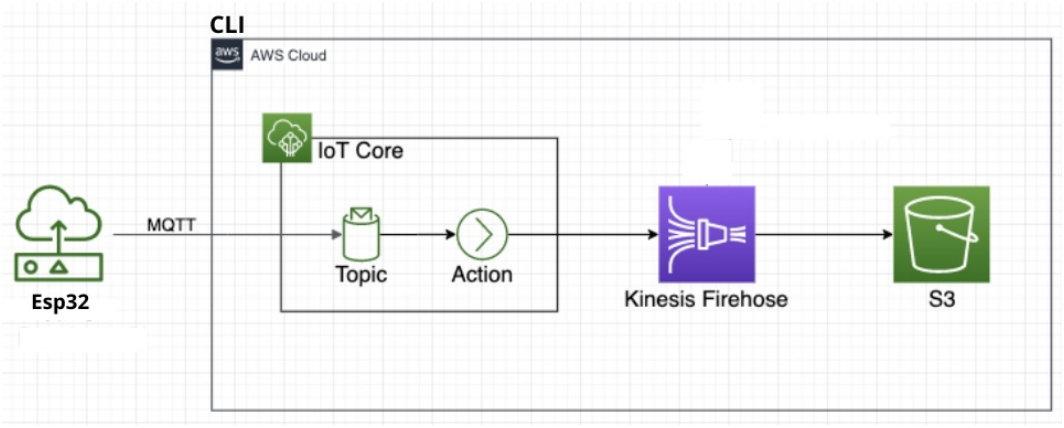


Figura 4.1: Arquitetura usada [Autor]

Ao longo deste capítulo, cada etapa é minuciosamente explicada, visando fornecer um guia abrangente para a utilização deste documento em trabalhos e aplicações futuras. Além disso, um tutorial em vídeo desta integração é disponibilizado através de um link no YouTube [67], proporcionando uma ferramenta adicional que pode ser empregada e compartilhada pelas instituições de ensino com seus futuros alunos.

4.1 Montagem do dispositivo

Para conduzir as simulações requeridas neste projeto, foi imprescindível desenvolver um dispositivo IoT básico. Para essa finalidade, optou-se pela utilização do sensor de humidade e temperatura DHT11, fornecido pela instituição acadêmica. Contudo, é relevante salientar que o propósito desta tese é demonstrar as conexões de dispositivos IoT de maneira abrangente. Deste modo, os procedimentos apresentados a seguir são aplicáveis a qualquer conexão entre dispositivos e a plataforma AWS.

Ao definir o sensor utilizado para os testes, procedeu-se com a montagem do circuito, composto por uma protoboard, o sensor de humidade e temperatura DHT11, um resistor, uma ESP32 e a programação no arduino. Este processo envolveu uma série de etapas cuidadosas.

Primeiramente, é fundamental assegurar que todos os componentes necessários estejam disponíveis e em perfeitas condições. Esta verificação inclui a integridade física dos componentes, como fios, cabos USB, resistores, a própria protoboard e a ESP32, além da funcionalidade do sensor DHT11.

A protoboard foi empregada como plataforma para a montagem do circuito, fornecendo uma abordagem conveniente e flexível para conectar os componentes eletrônicos. A disposição específica dos componentes na protoboard pode ser visualizada na figura 4.2.

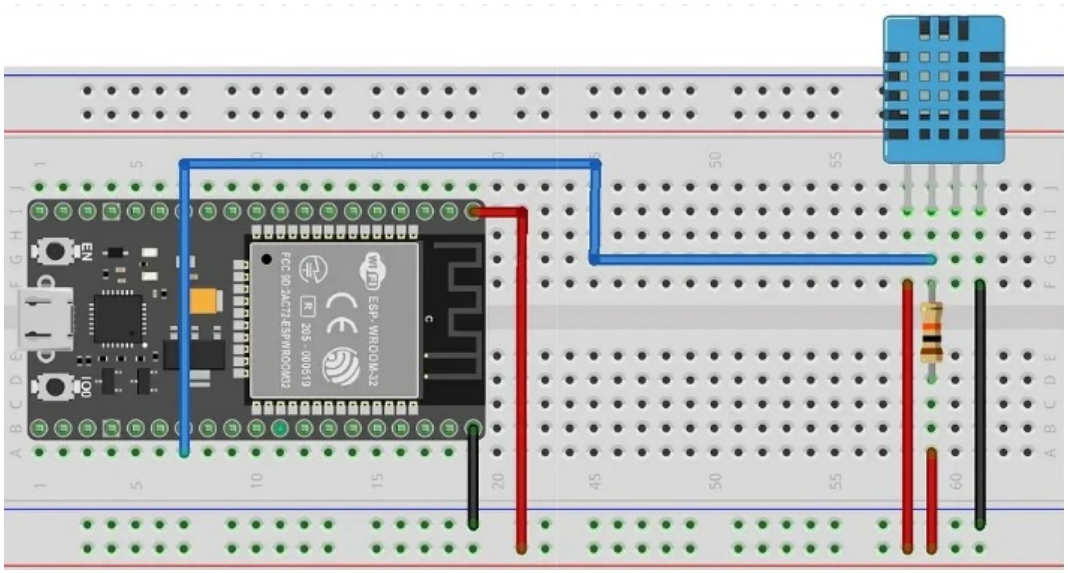


Figura 4.2: Dispositivo IoT [Autor]

O sensor DHT11 desempenha um papel fundamental no sistema, sendo responsável pela medição da humidade e temperatura do ambiente. O dispositivo possui geralmente quatro pinos: VCC (5V), GND (terra), DATA (para comunicação de dados) e um pino não utilizado. Na presente configuração, o pino VCC do sensor foi conectado à posição e4 da protoboard, o pino GND foi associado à posição e7 da protoboard, e o pino de dados foi ligado à posição e5, restando o pino não utilizado na posição e6 da protoboard.

Um resistor é comumente empregado em conjunto com o sensor DHT11 para assegurar a estabilidade na leitura dos dados. Este resistor é conectado entre o pino de dados (DATA) do sensor (posição b5 da protoboard) e o pino de alimentação (VCC) da ESP32 (posição b4 da protoboard), configurando-o como um resistor de *pull-up*.

A ESP32 assume o papel de microcontrolador central no circuito, incumbida de gerir o funcionamento do sensor DHT11 e processar os dados obtidos. O modelo específico de ESP32 utilizado neste estudo é o ESP32-WROOM-32D, conforme fornecido pela instituição acadêmica. A conexão da ESP32 à protoboard é realizada de acordo com os demais dispositivos já conectados. Neste contexto, os fios macho e fêmea foram utilizados para conectar o VCC do sensor ao VCC da ESP32 (posição a4 da protoboard), o GND do sensor ao GND da ESP32 (posição a7 da protoboard) e o pino de dados ao pino D14 (posição a5 da protoboard), configurado como GPIO14.

4.1.1 Elaboração do código no arduino

Para a elaboração do código foi utilizado a IDE que é uma plataforma de código aberto que permite programar microcontroladores como o ESP32 de maneira simples e eficiente. A utilização do arduino IDE para o ESP32 é bastante popular devido à sua facilidade de uso, ampla comunidade de suporte e extensa biblioteca de exemplos e tutoriais.

Para programar o ESP32 usando a IDE, você precisa adicionar o suporte ao ESP32 na IDE. Abra o arduino IDE e vá para *File* e depois *Preferences*. No campo "*Additional Board Manager URLs*", adicione o link [68] para ESP e depois clique em OK.

Em *Tools* e depois em *Board* selecione o modelo do ESP32 que você está usando (por exemplo, "*ESP32 Dev Module*"). Novamente em *Tools* e depois em *Port* selecione a porta COM à qual o ESP32 está conectado, após isto já é possivel fazer o upload do código no ESP32.

O código foi dividido em duas abas, sendo que cada uma delas tem um objetivo geral distinto. A primeira aba denominada "Main.ino" foi utilizada para colocar o código geral para o funcionamento do dispositivo. A segunda aba denominada "Secrets.h" foi utilizada para colocar os certificados, chaves obtidos na plataforma da AWS.

Main.ino

O programa desenvolvido para um dispositivo ESP32 tem como finalidade coletar dados de um sensor de temperatura e humidade do tipo DHT11 e enviá-los para o serviço *AWS IoT* através da internet, utilizando o protocolo MQTT.

A estrutura do programa compreende inclusões de bibliotecas necessárias, definições de constantes, declarações de variáveis (humidade h e temperatura t), inicializações de objetos e definições de funções. As principais bibliotecas incluídas são:

- WiFiClientSecure.h: Para comunicação segura pela internet;
- PubSubClient.h: Para comunicação usando o protocolo MQTT;
- ArduinoJson.h: Para manipulação de dados em formato JSON;
- WiFi.h: Para configuração e controle da conexão Wi-Fi;
- DHT.h: Para leitura dos sensores DHT.

A função connectAWS() é responsável por estabelecer a conexão com a *AWS IoT*, ela é uma função que não faz parte de uma biblioteca padrão do arduino, o que significa que ela precisa ser desenvolvida pelo programador com base nas

necessidades do projeto. Esta função conecta o ESP32 à rede Wi-Fi, configura os certificados de segurança necessários para autenticação com o *AWS IoT*, configura e conecta o cliente MQTT ao servidor *AWS IoT*, inscrever o cliente a um tópico MQTT para receber mensagens, garante que todo o processo seja monitorado e que mensagens de status sejam impressas no monitor serial para facilitar a depuração e o acompanhamento do processo de conexão. Estes detalhes são descritos abaixo:

- WiFi.mode(WIFI_STA): Define o ESP32 para operar no modo de estação (STA), permitindo que ele se conecte a uma rede Wi-Fi;
- WiFi.begin(WIFI_SSID, WIFI_PASSWORD): Inicia a conexão à rede Wi-Fi usando o *Service Set Identifier* (SSID) e a senha fornecidos;
- Serial.println("Connecting to Wi-Fi"): Imprime uma mensagem no monitor serial indicando que a conexão Wi-Fi está sendo estabelecida;
- While (WiFi.status() != WL_CONNECTED): Espera até que o ESP32 esteja conectado à rede Wi-Fi. Durante a espera, imprime um ponto no monitor serial a cada 500 ms para indicar progresso;
- Net.setCACert(AWS_CERT_CA): Define o certificado da *Autoridade Certificadora* (CA) raiz da AWS para o cliente seguro (WiFiClientSecure). Este certificado valida a autenticidade do servidor *AWS IoT*;
- Net.setCertificate(AWS_CERT_CRT): Define o certificado do dispositivo, que identifica de forma única o dispositivo ESP32 para o *AWS IoT*;
- Net.setPrivateKey(AWS_CERT_PRIVATE): Define a chave privada do dispositivo, que é usada junto com o certificado do dispositivo para autenticação segura;
- Client.setServer(AWS_IOT_ENDPOINT, 8883): Define o *endpoint* do servidor MQTT do *AWS IoT* e a porta (8883 para comunicação segura);
- Client.setCallback(messageHandler): Define a função *messageHandler* como o *callback* para processar mensagens recebidas do servidor MQTT. Esta função será chamada sempre que uma mensagem for recebida em um tópico ao qual o dispositivo está inscrito;
- Serial.println("Connecting to AWS IOT"): Imprime uma mensagem no monitor serial indicando que a conexão com o *AWS IoT* está sendo estabelecida;
- While (!client.connect(THINGNAME)): Tenta conectar ao servidor MQTT do *AWS IoT* usando o *THINGNAME* como o ID do cliente MQTT. Continua tentando até que a conexão seja estabelecida. Durante a tentativa, imprime um ponto no monitor serial a cada 100 ms para indicar progresso;

- If (!client.connected()): Verifica se a conexão foi estabelecida. Se não, imprime uma mensagem de *timeout* e retorna;
- Client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC): Inscreve o dispositivo a um tópico MQTT especificado em AWS_IOT_SUBSCRIBE_TOPIC, permitindo que ele receba mensagens publicadas nesse tópico;
- Serial.println("AWS IoT Connected!"): Imprime uma mensagem no monitor serial indicando que a conexão com o *AWS IoT* foi estabelecida com sucesso.

A função publishMessage() é responsável por compor um objeto JSON contendo os valores de temperatura e humidade lidos pelo sensor e publica este objeto no tópico MQTT especificado, assim como especificado abaixo:

- StaticJsonDocument<200> doc: Declaração de um objeto StaticJsonDocument com capacidade para 200 bytes. Este objeto é utilizado para criar e manipular a estrutura JSON que conterá os dados;
- Doc["humidity"]=h e doc["temperature"]=t: Adiciona um campo chamado "*humidity*"e "*temperature*"ao documento JSON e atribui a ele o valor da variável h e t, que representa a leitura de umidade e temperatura obtida do sensor;
- Char jsonBuffer[512]: Declaração de um buffer de caracteres com capacidade para 512 bytes. Este buffer é utilizado para armazenar a representação serializada (em formato de *string*) do documento JSON;
- SerializeJson(doc, jsonBuffer): A função serializeJson() converte o documento JSON (doc) em uma *string* JSON e armazena essa *string* no jsonBuffer. Essa conversão é necessária para enviar a mensagem JSON como uma cadeia de caracteres pelo protocolo MQTT;
- Client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer): Publica a mensagem JSON (armazenada em jsonBuffer) no tópico MQTT especificado por AWS_IOT_PUBLISH_TOPIC. O cliente MQTT (client) envia essa mensagem ao *AWS IoT*.

Já a função messageHandler() é chamada quando mensagens são recebidas do *AWS IoT*, imprimindo essas mensagens na porta serial. Esta função é chamada automaticamente pelo cliente MQTT sempre que uma nova mensagem é recebida. Os datalhes estão a seguir:

- Serial.print("incoming: "): Imprime a *string* "incoming: "no monitor serial;
- Serial.println(topic): Imprime o nome do tópico no monitor serial, seguido de uma nova linha. Isso ajuda a identificar em qual tópico a mensagem foi recebida;

- StaticJsonDocument<200> doc: Declara um objeto StaticJsonDocument com capacidade para 200 bytes. Este objeto será usado para armazenar e manipular os dados JSON recebidos;

- DeserializeJson(doc, payload): Converte o payload (que é uma sequência de bytes) em um documento JSON e armazena o resultado em doc. Esta função analisa a mensagem recebida e preenche o documento JSON com os dados;

- *Const char message = doc["message"]**: Extrai o valor associado à chave "*message*"no documento JSON e armazena em um ponteiro *message*. Assume-se que a mensagem JSON contém um campo chamado "*message*";

- Serial.println(message): Imprime o valor da mensagem no monitor serial. Isso permite visualizar o conteúdo da mensagem recebida.

No bloco setup(), a comunicação serial é inicializada e a função connectAWS() é chamada para estabelecer a conexão com a *AWS IoT*. Além disso, o sensor DHT11 é inicializado para a leitura dos dados de temperatura e humidade.

No loop principal (loop()), os valores de temperatura e umidade são lidos do sensor DHT11. Se a leitura for bem-sucedida, os valores são impressos na porta serial e enviados para a *AWS IoT* utilizando a função publishMessage(). O programa continua a receber e processar mensagens MQTT enquanto aguarda 1 segundo antes de repetir o ciclo.

Secrets.h

O código desenvolvido tem como objetivo configurar o dispositivo ESP32 para estabelecer uma conexão segura com o serviço *AWS IoT*. Ele inclui informações sensíveis, como credenciais de rede Wi-Fi e certificados de segurança necessários para a autenticação na *AWS IoT*.

Inicialmente, são importadas as bibliotecas necessárias, como pgmspace.h, que permite lidar com dados armazenados em *flash*. Em seguida, são definidas constantes, como o nome do dispositivo, o SSID da rede Wi-Fi, a senha da rede Wi-Fi e o *endpoint* da *AWS IoT*.

O código incorpora os certificados de segurança essenciais para a autenticação na *AWS IoT*: o certificado da autoridade certificadora raiz da Amazon, o certificado do dispositivo e a chave privada do dispositivo. Esses certificados são armazenados na memória *flash* do dispositivo ESP32 utilizando o modificador PROGMEM, permitindo que dados constantes sejam armazenados na memória *flash* em vez da RAM.

No bloco *setup* do dispositivo, as configurações de rede Wi-Fi são inicializadas e a conexão com a *AWS IoT* é estabelecida utilizando os certificados e o *endpoint* previamente configurados.

É relevante ressaltar que os dados como nome do dispositivo, SSID da rede Wi-Fi, senha, certificados e chaves são exclusivos para cada projeto e devem ser substituídos conforme as especificidades de cada usuário e dispositivo.

Em resumo, o código proporciona uma configuração abrangente para permitir que um dispositivo ESP32 se conecte de forma segura ao serviço *AWS IoT*, garantindo autenticação e comunicação seguras entre o dispositivo e a plataforma em nuvem da AWS. As informações fornecidas neste trecho são utilizadas na seção para o correto funcionamento do dispositivo e sua conexão.

4.2 Conexão com o AWS IoT Core

Para concretizar a integração do dispositivo IoT com a plataforma da Amazon, adotou-se uma abordagem prática, dividida em várias fases. Inicialmente, foi imprescindível proceder com a criação da conta na AWS. Esta conta é pessoal e cada usuário deve criar sua própria conta para acessar os serviços da AWS. Para os propósitos desta pesquisa, foi aproveitado o período gratuito oferecido pela plataforma por um período de um ano, permitindo o acesso a uma variedade de serviços dentro de um limite de utilização gratuito. Assim, para as simulações realizadas, foi viável utilizar exclusivamente os serviços disponibilizados de forma gratuita. Apesar de disponibilizar este período gratuito, ao criar a conta é preciso cadastrar um cartão caso seja utilizado mais serviços do que os gratuitos.

4.2.1 Configuração do dispositivo

Após a conclusão do processo de criação da conta nos serviços disponíveis, procede-se à configuração no *AWS IoT Core*, onde um novo dispositivo é criado. Um dispositivo IoT fica associado à conta específica na qual foi criado, garantindo que os dados e configurações estejam vinculados a essa conta. Para realizar esta etapa, é preciso acessar o serviço da *AWS IoT Core*, a página principal deste serviço pode ser observada na figura 4.3. Depois no menu lateral no campo "Gerenciar", localizar "Todos os dispositivos"e, posteriormente, "Coisas". Em seguida, é selecionada a opção "Criar itens"no botão destacado em laranja. Nesse ponto, surgem duas alternativas, sendo possível criar uma única coisa ou várias coisas. No presente caso, optou-se por "Criar um único Item".

Figura 4.3: AWS IoT Core [Autor]

4.2.2 Definição das propriedades

Posteriormente, são definidas as propriedades do dispositivo, conforme figura 4.4 abaixo, incluindo o nome da coisa, bem como as configurações referentes à sombra do dispositivo, mantendo-se as opções padrão das configurações adicionais e mantendo também sem sombra.

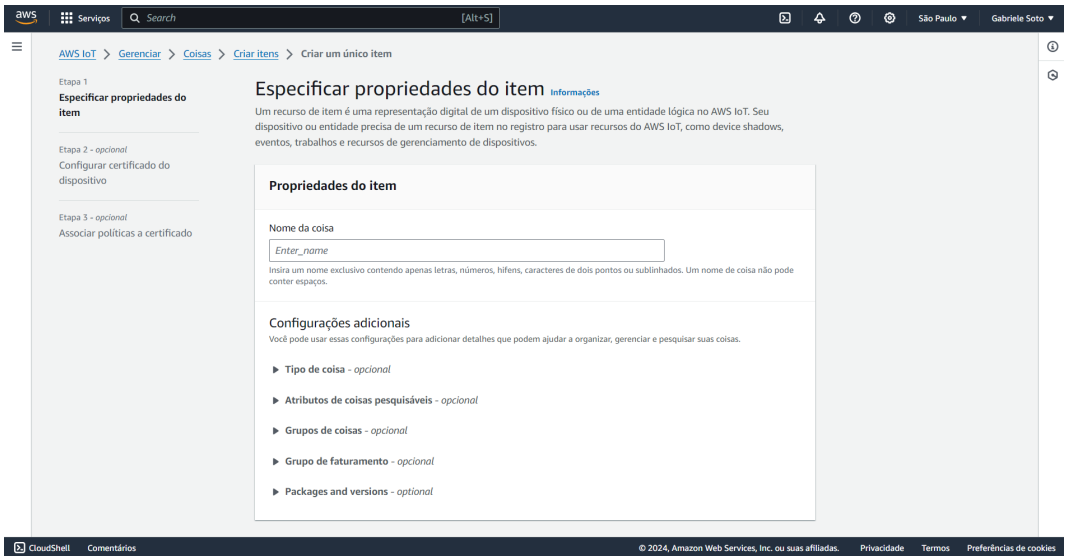


Figura 4.4: AWS IoT Core - nome do dispositivo [Autor]

A comunicação dos dispositivos com sombras, por meio de mensagens MQTT é considerada o método preferencial para a interação dos dispositivos com o serviço *AWS IoT Device Shadow*, com opções demonstradas na figura 4.5. Esse tipo de comunicação com as sombras emula um modelo de solicitação/resposta, utilizando o

padrão de comunicação de publicação/assinatura do MQTT. Cada ação relacionada à sombra compreende um tópico de solicitação, um tópico de resposta bem-sucedida (*accepted*) e um tópico de resposta de erro (*rejected*).

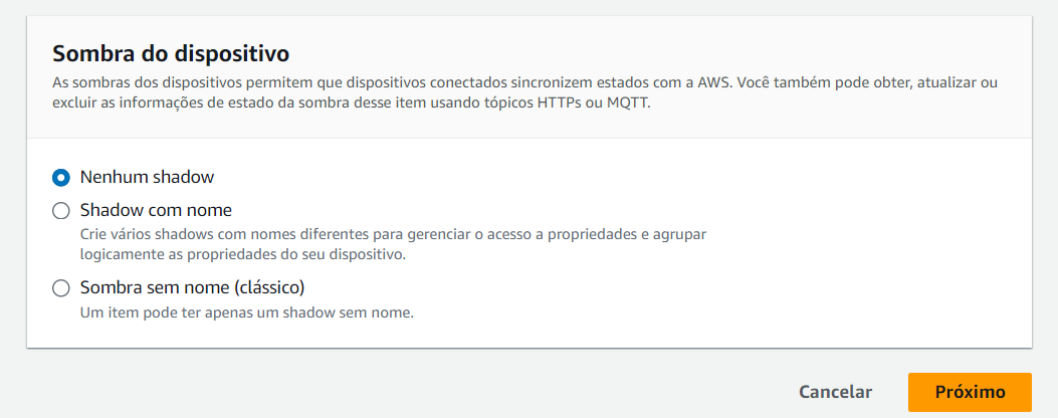


Figura 4.5: AWS IoT Core - Shadow [Autor]

4.2.3 Configuração dos certificados

O próximo procedimento consiste na configuração do certificado do dispositivo, no qual é viável utilizar um certificado previamente criado ou realizar o upload do *Certificate Signing Request* (CSR). No entanto, a recomendação da AWS é a geração de um novo certificado. Estas opções são apresentadas de acordo com a figura 4.6. Ao optar por essa alternativa, uma tela será apresentada para a vinculação de uma política a este novo certificado, para a qual é necessário criar uma nova política clicando no botão localizado na parte superior direita da tela.

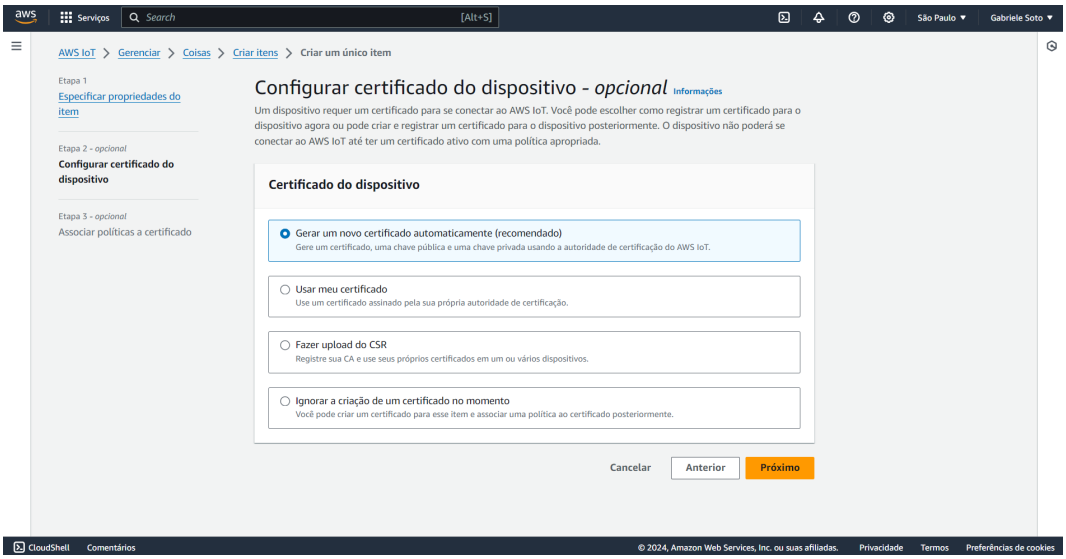


Figura 4.6: AWS IoT Core - Certificado [Autor]

4.2.4 Criação de políticas

Para criar uma política, é necessário atribuir um nome a esta política. Em seguida, na seção de declarações de política, é imprescindível definir quatro ações (publicar, subscrever, conectar e receber) para a política. Estas ações devem ser criadas na tela conforme figura 4.7 abaixo.

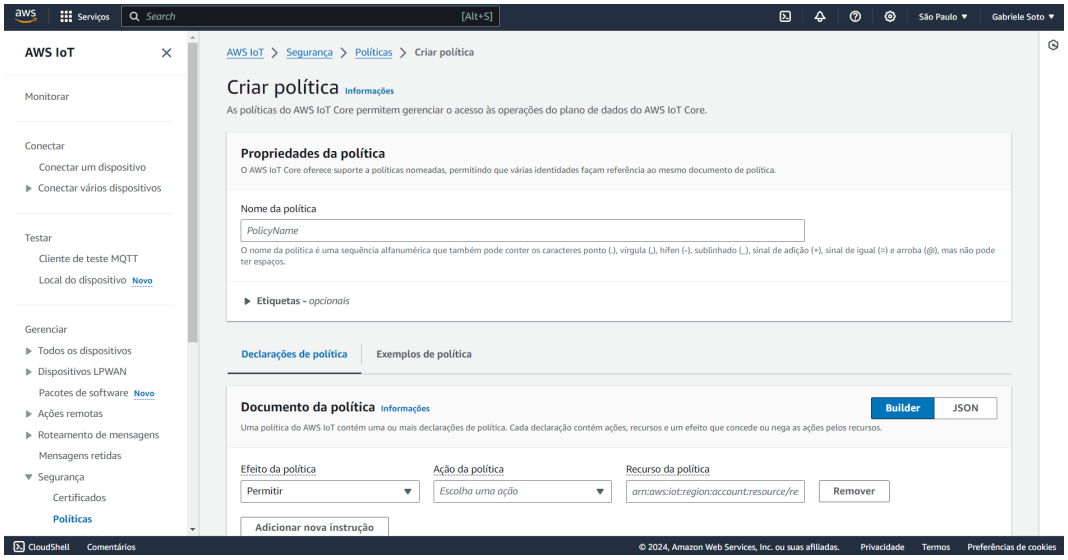


Figura 4.7: AWS IoT Core - Política [Autor]

A primeira ação é "iot:connect", sendo essencial que o recurso da política esteja vinculado ao seu usuário. Para isso, preenchemos a secção final com o ID do cliente, seguindo o formato: "arn:aws:iot:sa-east-1:xxxxxxxxxxxx:client/clientID". Em "clientID", substitua pelo nome do dispositivo, por exemplo.

Na parte "sa-east-1", substitua conforme necessário para a sua localização, identificada no canto superior direito, próximo ao seu nome. Para "xxxxxxxxxxxx", substitua pelo ID da sua conta, encontrado no mesmo local onde aparece o seu nome.

A segunda ação é "iot:Publish", e no recurso da política, preencha com "arn:aws:iot:sa-east-1:xxxxxxxxxxxx:topic/replacewithatopic". Para "*replacewithatopic*", coloque o nome do tópico definido no seu código arduino em "AWS_IOT_PUBLISH_TOPIC".

A terceira ação é "iot:Subscribe", e no recurso da política, preencha com "arn:aws:iot:sa-east-1:xxxxxxxxxxxx:topic/replacewithatopicfilter". Para "*replacewithatopicfilter*", substitua pelo nome do tópico definido no seu código Arduino em "AWS_IOT_SUBSCRIBE_TOPIC".

A quarta ação é "iot:Receive", e no recurso da política, preencha com "arn:aws:iot:sa-east-1:xxxxxxxxxxxx:topic/replacewithatopic". Para "*replacewithatopic*", utilize o mesmo tópico de subscrição definido na ação anterior.

Para todas as políticas considere as seguintes substituições. Na parte "sa-east-1", substitua conforme necessário para a sua localização, identificada no canto superior direito, próximo ao seu nome. Para "xxxxxxxxxxxx", substitua pelo ID da sua conta, encontrado no mesmo local onde aparece o seu nome.

No efeito da política, deve-se selecionar "permitir"para todas as ações.

Finalmente, basta clicar em "criar política"para concluir o processo, assegurando assim a criação bem-sucedida da política.

4.2.5 Vinculação da política e download dos certificados

Após a criação da política, é possível retornar à aba anterior de criação do dispositivo, onde a política recém criada estará disponível para ser vinculada ao certificado. Para realizar este procedimento, selecione a política criada e clique em "criar coisa". Posteriormente, uma tela será exibida para efetuar o download do certificado e das chaves geradas.

Neste momento, é de suma importância realizar o download de quatro arquivos distintos, que aparecem conforme figura 4.8 abaixo. São eles: o certificado do dispositivo, renomeado como "*Device Certificate*"; o arquivo da chave pública, renomeado como "*Public Key*"; o arquivo da chave privada, renomeado como "*Private Key*"; por último, faça o download do Certificado CA Raiz.

É crucial ressaltar que todos esses arquivos devem ser salvos na mesma pasta no computador.

Figura 4.8: AWS IoT Core - Downloads [Autor]

4.2.6 Verificação final

Após a conclusão desses procedimentos, é possível clicar em "concluído", e a "coisa" estará devidamente criada. Para verificar se todos os passos foram executados corretamente, é aconselhável revisar cada criação em suas respectivas abas no menu lateral. A "coisa" estará registrada na secção de "Todos os dispositivos-> "Coisas", onde será possível visualizar o dispositivo criado. Na secção de "Segurança->

"Certificados" e "Políticas", é possível observar o certificado e a política criados, respectivamente. Além disso, pode-se verificar se estão ativos, conforme o esperado.

É importante ressaltar que um dispositivo IoT pode se comunicar com outros dispositivos através de tópicos MQTT, permitindo a troca de informações e coordenação entre dispositivos. No entanto, essa comunicação é mediada através do *AWS IoT Core*, que atua como um intermediário, gerenciando a comunicação e a segurança.

Os servidores do *AWS IoT Core* são compartilhados entre os dispositivos associados à mesma conta, mas a plataforma garante que cada conta tenha suas próprias configurações, políticas e dados de dispositivo isolados e seguros. Isso significa que os dispositivos de uma conta podem se comunicar entre si, mas estão logicamente separados dos dispositivos de outras contas.

4.3 Armazenamento de dados

O armazenamento dos dados gerados pelos dispositivos IoT, pode ser feito através da utilização de serviços disponíveis na plataforma da AWS, como o Amazon S3. Para coleta de dados em maior volume, optou-se pelo *Amazon Kinesis Firehose*, exigindo a adesão a um procedimento meticuloso e bem definido. Abaixo, descreve-se minuciosamente os passos necessários para estabelecer essa conexão.

Para iniciar esse processo, é necessário acessar o serviço *AWS IoT Core*. Em seguida, no menu lateral, na seção "Gerenciar", deve-se localizar e acessar "Roteamento de mensagens" e, posteriormente, "Regras", conforme ilustrado na figura 4.9. Ao acessar as regras, seleciona-se a opção "Criar regras" no botão destacado em laranja.

Figura 4.9: AWS IoT Core - Roteamento de mensagens [Autor]

4.3.1 Especificando as propriedades da regra

Inicialmente, é necessário especificar as propriedades da regra, o que inclui definir o nome da regra e fornecer uma descrição. Essa descrição pode ser uma repetição do nome da regra ou uma explicação do propósito e das funcionalidades da regra, facilitando assim a compreensão para outros usuários que possam precisar utilizá-la ou para identificação futura. Após esta etapa, é possível visualizar a figura 4.10, que ilustra a tela que deverá ser exibida.

Figura 4.10: AWS IoT Core - Regras [Autor]

4.3.2 Estabelecendo as instruções SQL

Na próxima página, são estabelecidas as instruções do SQL. A versão do SQL pode ser mantida conforme o padrão. Na secção do código SQL, define-se o atributo que será selecionado, podendo ser apenas parte dos dados gerados ou todos os dados, dependendo das necessidades de cada aplicação. É especificado o tópico do qual esses dados serão obtidos, indicando, portanto, o dispositivo do qual serão provenientes os dados, e também pode ser adicionada uma condição, caso haja necessidade de aplicar filtros nos dados coletados. Neste projeto, foram armazenadas todas as informações provenientes do dispositivo de teste, utilizando-se, portanto, o código "SELECT * FROM 'esp32/pub'". No caso de ser necessário atributos e condições especiais, deve-se seguir o código exemplificado na figura 4.11, fazendo apenas as alterações necessárias.

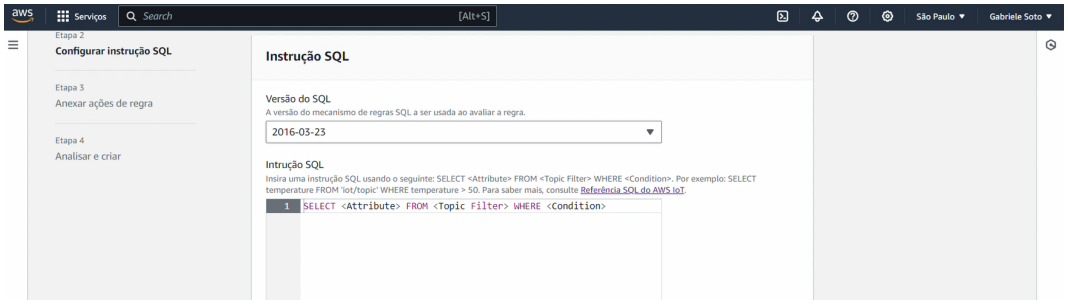


Figura 4.11: AWS IoT Core - Definição de regra por SQL [Autor]

4.3.3 Anexando ações à regra

Na página subsequente, são anexadas as ações da regra. Existem diversas ações disponíveis na plataforma, entretanto, para este projeto, foi selecionada a "*Amazon Data Firehose Stream*", a qual encaminha uma mensagem para o *stream* do *Amazon Data Firehose*, conforme evidenciado na figura 4.12.

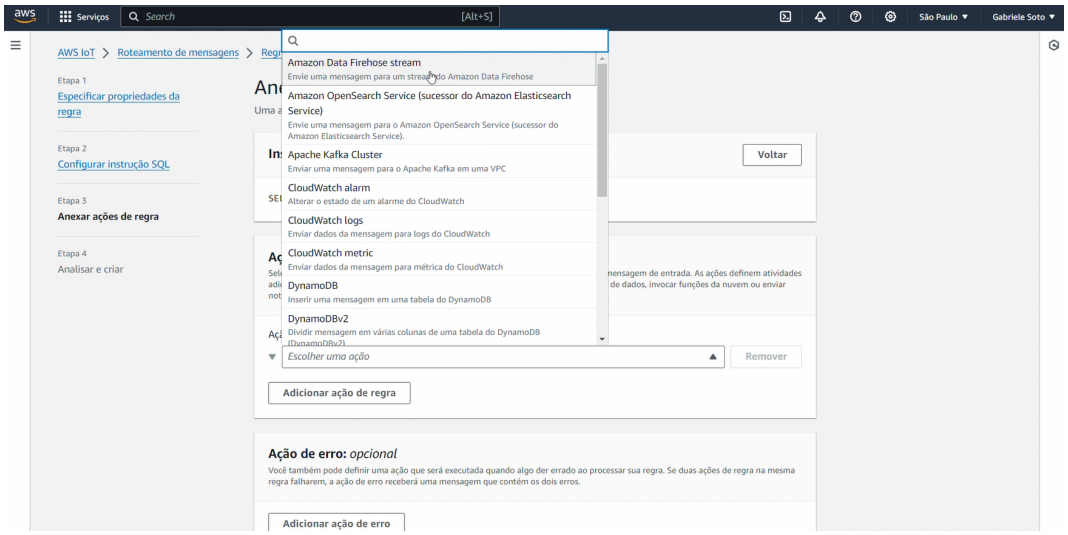


Figura 4.12: AWS IoT Core - Data Firehose Stream [Autor]

4.3.4 Criando o stream do Amazon Kinesis Firehose

É necessário criar o *Stream* do Amazon que será utilizado. Para isso, seleciona-se entre as opções disponíveis a origem do fluxo, que oferece três alternativas: *Amazon Kinesis Data Stream*, Amazon MSK e *Direct PUT*. Neste caso, utiliza-se o *Direct PUT* e, para o destino do fluxo, pode-se escolher entre diversas opções, porém, neste caso específico, opta-se pelo Amazon S3. O nome do fluxo é atribuído automaticamente e, nas demais configurações, pode-se manter as opções padrão que forem apresentadas, sendo o nome semelhante ao da figura 4.13.

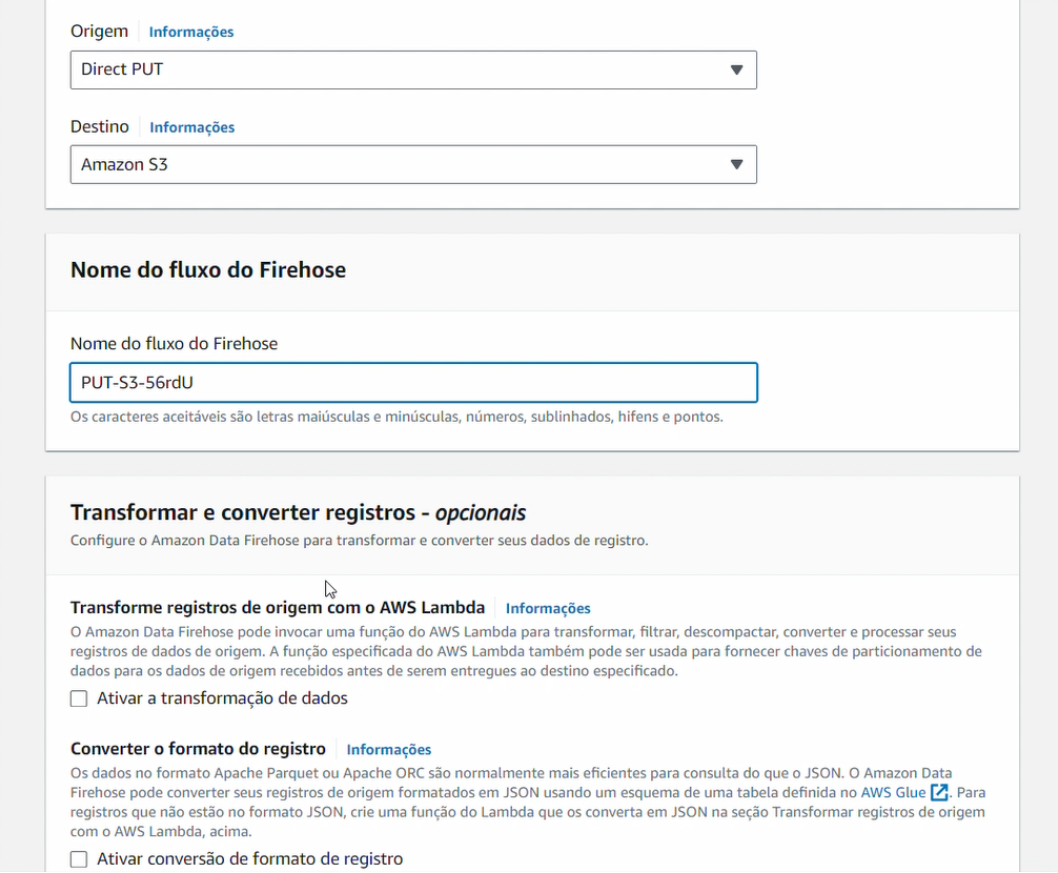


Figura 4.13: AWS IoT Core - Fluxo do Firehose [Autor]

4.3.5 Configurando o Bucket

Na configuração do destino, é necessário criar o *bucket* do S3. Para tal, deve-se clicar no botão ao lado de "criar", o que resultará na abertura de uma nova página. É então definido um nome para o *bucket*, e todas as demais configurações podem ser mantidas no padrão que são apresentadas. Ao clicar em "Criar *bucket*", será possível visualizar o *bucket* já criado e disponível, acompanhado de uma mensagem de confirmação da criação destacada em verde, conforme demonstrado na figura 4.14.

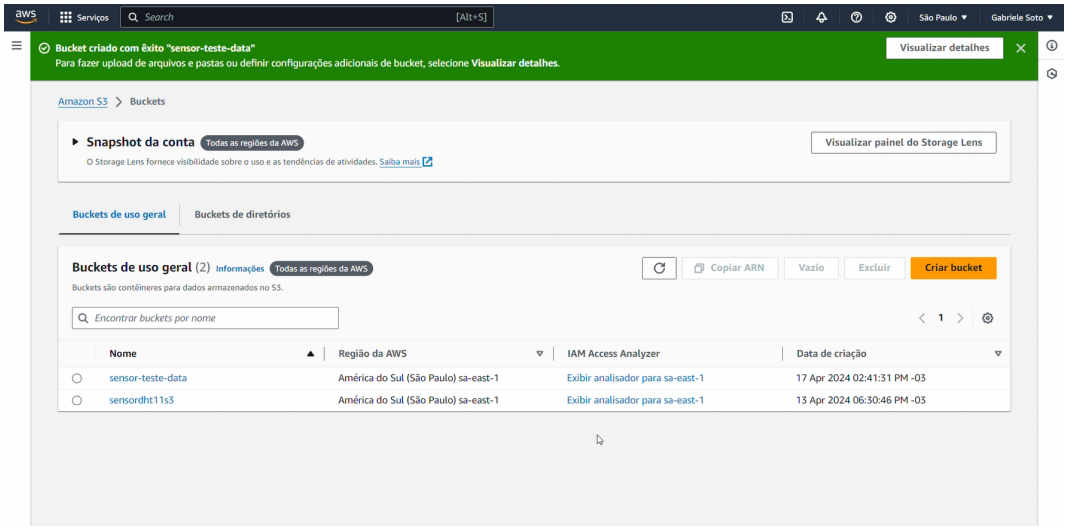


Figura 4.14: AWS IoT Core - Criação do Bucket [Autor]

4.3.6 Finalizando a configuração do Amazon Data Firehose

Após a criação do *bucket*, é possível retornar à aba do *Amazon Data Firehose* e clicar em "Procurar" para selecionar o *bucket* que foi criado anteriormente. Logo abaixo, no campo "Prefixo do *bucket* S3", pode-se definir um prefixo a ser aplicado aos dados quando forem armazenados, este campo pode ser preenchido conforme o interesse de cada projeto. Em "Dicas de buffer, compressão, extensão de arquivo e criptografia", é necessário configurar o tamanho do buffer, neste caso, 1MiB, e o intervalo do buffer, neste caso, 120 segundos. As demais configurações podem ser mantidas como padrão. Após completar essas etapas, é possível criar o fluxo do *Firehose*, e uma tela semelhante à figura 4.15 fornecida deverá ser exibida após a conclusão dessa criação.

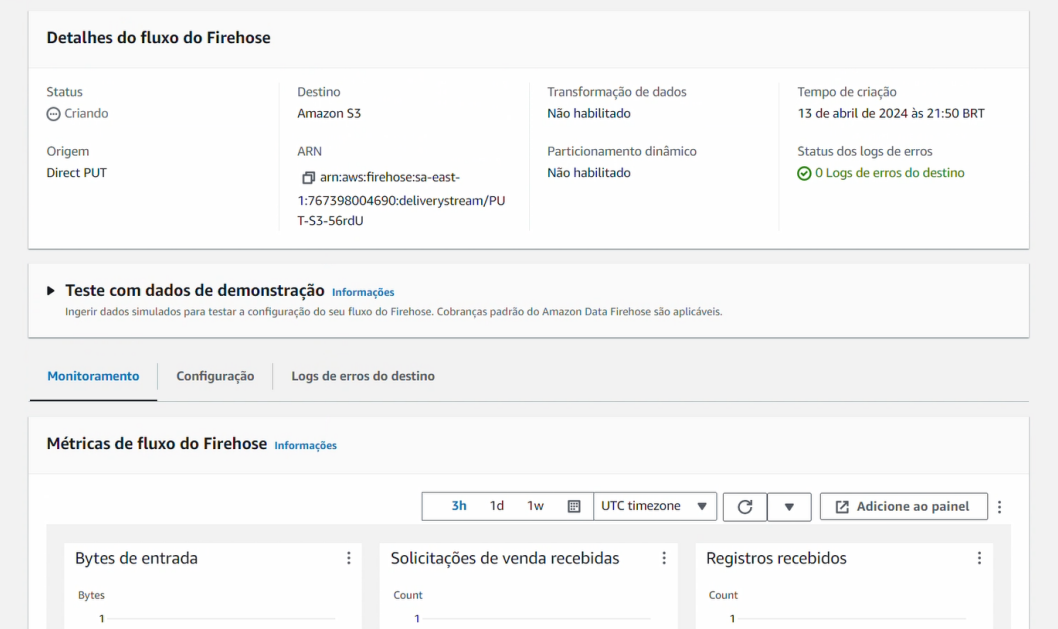


Figura 4.15: AWS IoT Core - Criação do fluxo [Autor]

4.3.7 Criando e associando à função IAM

Após a criação do fluxo do *Firehose*, retorna-se à aba de criação da regra e seleciona-se o *stream* do *Amazon Data Firehose* criado. Para o armazenamento de dados desta forma, também é necessário criar uma função do IAM para conceder ao AWS acesso ao seu *endpoint*. Para isso, basta clicar em "criar nova função" e, em seguida, nomeá-la. Após esta etapa, é possível relacionar esta função IAM com a regra, prosseguir para a próxima página de revisão dos campos preenchidos e, por fim, criar a regra. Ao criar a regra, uma página semelhante à figura 4.16 deverá ser exibida.

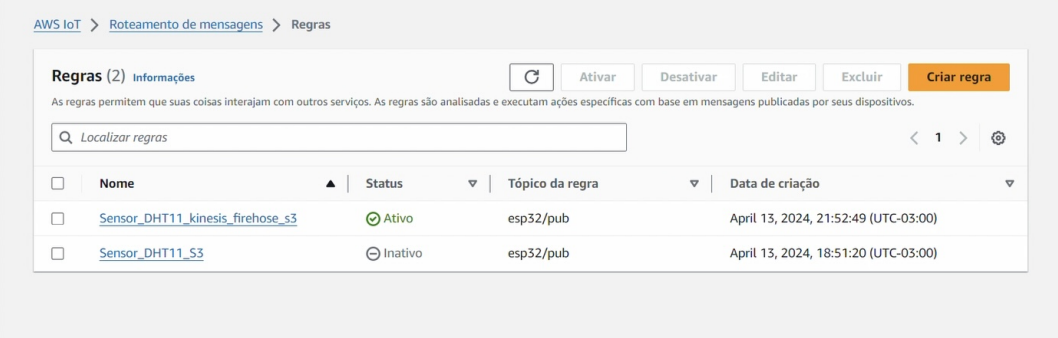


Figura 4.16: AWS IoT Core - Criação da regra [Autor]

4.4 Script para conexão

Ao utilizar a plataforma da AWS, é possível realizar os passos descritos anteriormente por meio da ferramenta CLI disponibilizada pela própria plataforma. Inicialmente,

é necessário acessar a conta da AWS criada e navegar até o serviço IAM para criar um usuário. Na barra lateral, selecione "Usuários" e clique em "Adicionar usuário". Insira o nome desejado para o usuário. Na etapa de definição de permissões, selecione "Anexar políticas diretamente" e escolha a política "*Administrator Access*". Opcionalmente, adicione uma tag de identificação e, em seguida, crie o usuário.

Após a criação do usuário, acesse seus detalhes assim como demonstrado na figura 4.17 , clique em "Credenciais de segurança" e selecione CLI. Confirme a seleção ao final da página. Em seguida, defina uma etiqueta de descrição, se necessário. Isso gerará uma chave de acesso e uma chave de acesso secreta, que devem ser armazenadas com segurança, pois serão utilizadas posteriormente para autenticar no CLI. Ressalta-se que essas chaves são pessoais e não devem ser compartilhadas.

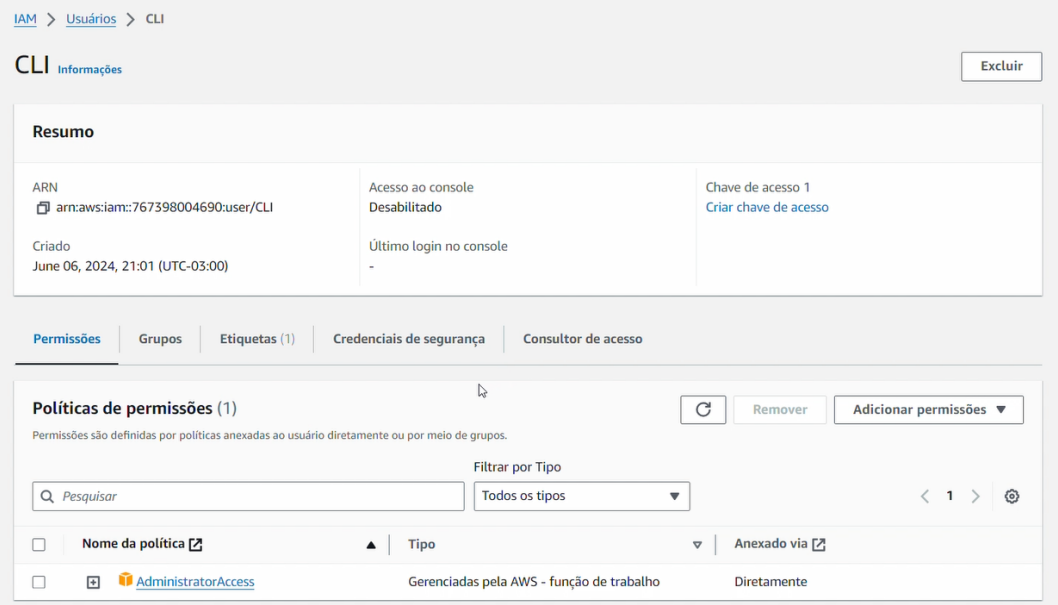


Figura 4.17: AWS IAM - permissões [Autor]

Para baixar o CLI, acesse o site *AWS CLI* [69] e baixe a versão compatível com seu sistema operacional. Após a instalação, abra o *prompt* de comando do computador e navegue até a pasta de instalação da *AWS CLI*. Utilize os comandos cd.. para retornar ao diretório raiz e, em seguida, navegue até os diretórios "*Program Files*", "Amazon" e "AWSCLIV2". Uma vez na pasta correta, execute o comando "*aws configure*". Insira a chave de acesso fornecida e, em seguida, a chave de acesso secreta. Também será solicitado que você informe a região padrão utilizada na plataforma e o formato de saída preferido, que pode ser definido como "*text*". Após a configuração, pode-se iniciar as configurações necessárias para o dispositivo utilizando o *AWS CLI*.

4.4.1 Criação do dispositivo

Da mesma forma que os procedimentos realizados no console, os passos subsequentes têm os mesmos objetivos, mas são executados de maneira mais prática para facilitar o uso por futuros usuários através do CLI. Os comandos utilizados são detalhados a seguir:

- Criar *thing*:

```
aws iot create-thing --thing-name NomeDaThing
```

- Criar certificado:

```
aws iot create-keys-and-certificate --set-as-active
--output json > certs.json
```

- Extrair informações do certificado:

```
powershell -Command "((Get-Content certs.json |
ConvertFrom-Json).certificateArn)" > cert_arn.txt &&
powershell -Command "((Get-Content certs.json |
ConvertFrom-Json).certificateId)" > cert_id.txt && set
/p CERTIFICATE_ARN=<cert_arn.txt && set /p
CERTIFICATE_ID=<cert_id.txt
```

- Associar o certificado à *thing*:

```
aws iot attach-thing-principal --thing-name
NomeDaThing --principal %CERTIFICATE_ARN%
```

- Criar a política:

```
aws iot create-policy --policy-name NomeDaSuaPolicy
--policy-document {"Ver-sion": "2012-10-17",
"Statement": [{"Effect": "Allow", "Action":
"iot:Connect", "Resource": "arn:aws:iot:sa-east-
1:xxxxxxxxxxx:client/NomeDaThing"}, {"Effect": "Allow",
"Action": "iot:Publish", "Resource": "arn:aws:iot:sa-
east-1:xxxxxxxxxxx:topic/replacewithatopic"},
```

```
{"Effect": "Allow", "Action": "iot:Subscribe",
"Resource": "arn:aws:iot:sa-east-
1:xxxxxxxxxxx:topic/replacewithatopicfilter"},
{"Effect": "Allow", "Action": "iot:Receive", "Resource":
"arn:aws:iot:sa-east-1:xxxxxxxxxxx:topic/
replacewithatopic"}]}
```

- Anexar a política ao certificado:

```
aws iot attach-policy --policy-name NomeDaSuaPolicy
--target %CERTIFICATE_ARN%
```

4.4.2 Configuração para o armazenamento

Assim como a criação do dispositivo, o armazenamento dos dados gerados por este também pode ser realizado de maneira simplificada através do CLI. Os comandos utilizados para este propósito são detalhados a seguir:

- Criar *bucket*:

```
aws s3api create-bucket --BucketName --region
us-east-1/BucketName
```

- Criar a Função IAM com a Política de Confiança:

```
aws iam create-role --role-name RoleName --assume-role-
policy-document "{\"Version\": \"2012-10-17\",
\"Statement\": [{\"Effect\": \"Allow\", \"Principal\":
{\"Service\": \"firehose.amazonaws.com\"}, \"Action\":
\"sts:AssumeRole\"}]}"
```

- Criar a Política de Permissões para o *Kinesis Firehose*:

```
aws iam create-policy --policy-name RoleNamePolicy --
policy-document "{\"Version\": \"2012-10-17\",
\"Statement\": [{\"Effect\": \"Allow\", \"Action\":
[\"s3:PutObject\", \"s3:GetBucketLocation\",
\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
[\"arn:aws:s3:::BucketName\",
```

```
\"arn:aws:s3:::BucketName/*\"]}, {\"Effect\":
\"Allow\", \"Action\": [\"firehose:PutRecord\",
\"firehose:PutRecordBatch\"], \"Resource\":
\"arn:aws:firehose:us-east-
1:xxxxxxxxxxx:deliverystream/StreamName\"}]}"
```

- Anexar a Política de Permissões à Função IAM:

```
aws iam attach-role-policy --role-name RoleName
--policy-arn arn:aws:iam::xxxxxxxxxxx:policy/
RoleNamePolicy
```

- Criar o *Kinesis Firehose Delivery Stream*:

```
aws firehose create-delivery-stream --delivery-stream-
name StreamName --s3-destination-configuration "
{\"RoleARN\":\"arn:aws:iam::xxxxxxxxxxx:role
/RoleName\", \"BucketARN\": \"arn:aws:s3:::BucketName\",
\"Prefix\": \"prefixo/\", \"BufferingHints\":
{\"SizeInMBs\": 1, \"IntervalInSeconds\": 120},
\"CompressionFormat\": \"UNCOMPRESSED\"}"
```

- Criar a Função IAM com a Política de Confiança:

```
aws iam create-role --role-name RoleName2 --assume-role-
policy-document "{\"Version\": \"2012-10-17\",
\"Statement\": [{\"Effect\": \"Allow\", \"Principal\":
{\"Service\": \"iot.amazonaws.com\"}, \"Action\":
\"sts:AssumeRole\"}]}"
```

- Criar a Política de Permissões:

```
aws iam create-policy --policy-name RoleName2Policy --
policy-document "{\"Version\": \"2012-10-17\",
\"Statement\": [{\"Effect\": \"Allow\", \"Action\":
[\"firehose:PutRecord\", \"firehose:PutRecordBatch\",
\"s3:PutObject\", \"s3:GetBucketLocation\",
\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
```

```
[\"arn:aws:firehose:us-east-1:xxxxxxxxxxx:
deliverystream/StreamName\", \"arn:aws:s3:::
BucketName\", \"arn:aws:s3:::BucketName/*\"]}]}"
```

- Anexar a Política de Permissões à Função IAM:

```
aws iam attach-role-policy --role-name RoleName2 --
policy-arn arn:aws:iam::xxxxxxxxxxx:policy/
RoleName2Policy
```

- Criar a Regra no *AWS IoT Core*:

```
aws iot create-topic-rule --rule-name "RuleName"
--topic-rule-payload "{\"sql\": \"SELECT * FROM
'replacewithatopic'\", \"description\": \"Descrição da
regra\", \"actions\": [{\"firehose\": {\"roleArn\":
\"arn:aws:iam::xxxxxxxxxxx:role/RoleName2\",
\"deliveryStreamName\": \"StreamName\"}}]}"
```

Com todas as etapas concluídas, é possível verificar a correta realização das conexões tanto no console quanto no CLI. A verificação pode ser realizada enviando uma mensagem e verificando se os dados foram armazenados no *bucket* utilizando os seguintes comandos:

- Publicar uma Mensagem de Teste no Tópico IoT:

```
aws iot-data publish --topic replacewithatopic --payload
"{\"teste\": \"valor\"}" --cli-binary-format raw-in-
base64-out
```

- Listar Objetos no *Bucket* S3:

```
aws s3 ls s3://BucketName/prefixo/
```

- Baixar um Arquivo Específico do S3:

```
aws s3 cp s3://BucketName/prefixo/FileName
```

Capítulo 5

Análise de Resultados

Na primeira parte das simulações, após a criação do código no arduino IDE, é possível observar os dados captados pelo sensor na própria aba do arduino IDE, assim como mostrado na Figura 5.1. Nas primerias linhas aparecem informações relacionadas ao WiFi, o que demonstra que a conexão com o AWS foi feita.

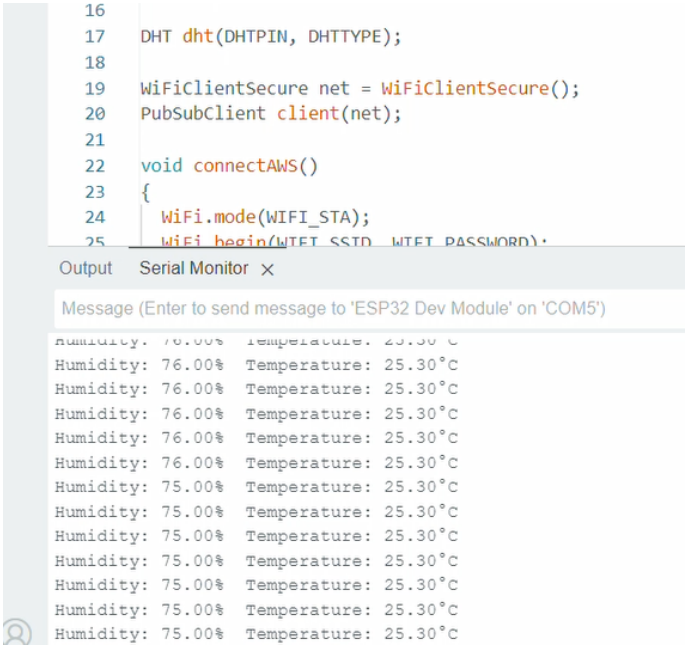


Figura 5.1: Monitor Serial Arduino [Autor]

Já na segunda parte, após fazer toda a conexão com a plataforma da Amazon, ao entrar na parte de teste na própria página do *AWS IoT Core* há uma aba de testes, em clientes de teste MQTT, foi colocado o mesmo nome do tópico que foi definido no código no arduino IDE e inscreve-se neste tópico. Com isto foram obtidos na plataforma AWS os dados captados pelo sensor assim como pode ser observado na Figura 5.2.

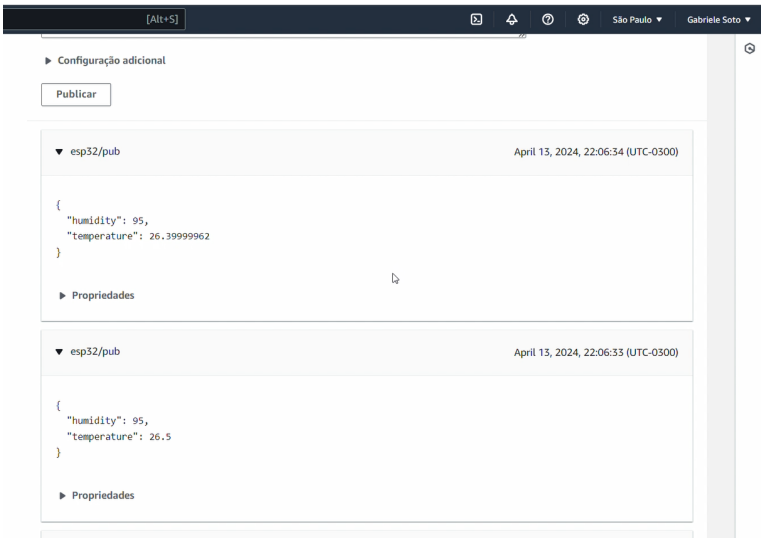


Figura 5.2: Dados no AWS IoT Core [Autor]

Após a criação do *bucket* e o sensor estar funcionando a alguns minutos, é necessário recarregar a página para que os arquivos de armazenamento apareçam. A cada dois minutos um novo arquivo era criado com os dados armazenados, assim como definido em etapas anteriores. Para obter os dados armazenados foi feito o download de um dos arquivos e ao abri-lo observasse os dados devidamente armazenados, assim como na Figura 5.3.

File Edit View

{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":73,"temperature":25.20000076}
{"humidity":92,"temperature":25.39999962}

Figura 5.3: Dados armazenados [Autor]

A plataforma AWS permitiu que os dispositivos IoT fossem conectados de forma segura, fornecendo uma variedade de serviços que facilitam a comunicação e o gerenciamento de dispositivos. Além disso, o potencial da AWS para processar e analisar grandes quantidades de dados produzidas por dispositivos IoT é demonstrado pelo armazenamento de dados em serviços como o Amazon S3 e pela análise de fluxos de dados em tempo real com o *Amazon Kinesis Firehose*.

Ao utilizar a AWS este projeto teve como objetivo auxiliar futuros alunos em seus trabalhos e também agrega-lo um conhecimento base sobre uma das plataformas de computação em nuvem mais utilizadas por empresas atualmente.

Para resumir, a integração da tecnologia de IoT com a plataforma AWS forneceu uma solução escalável e robusta para aplicações IoT, Além de fornecer ferramentas essenciais para a educação em ambientes industriais e acadêmicos. Demonstrando enfim a viabilidade da AWS como plataforma para o desenvolvimento de dispositivos IoT em ambientes universitários.

Capítulo 6

Conclusões e Desenvolvimentos Futuros

Este trabalho focou na metodologia para a criação de um ambiente didático na nuvem que facilitará o processo de aprendizagem da coleta e tratamento de dados provenientes de dispositivos IoT, cada etapa foi minuciosamente explicada, visando fornecer um guia abrangente para a utilização deste documento em trabalhos e aplicações futuras. Além disso, um tutorial em vídeo desta integração é disponibilizado através de um link no YouTube e o código no GitHub [67], proporcionando uma ferramenta adicional que pode ser empregada e compartilhada pelas instituições de ensino.

Ao utilizar a CLI da AWS para projetos e em aulas universitárias, os benefícios e vantagens são amplos e significativos. A ferramenta permite a automação de tarefas repetitivas, facilitando a gestão eficiente de recursos na nuvem e promovendo práticas de infraestrutura como código. A interação programática com os serviços da AWS reduz a dependência de interfaces gráficas, acelerando o desenvolvimento de projetos e permitindo um controle granular e flexível sobre os recursos. Além disso, a CLI é uma ferramenta essencial para ensinar conceitos de DevOps, administração de sistemas e desenvolvimento de aplicações em nuvem, graças à sua compatibilidade multiplataforma e documentação abrangente, que a tornam acessível e prática tanto para estudantes quanto para educadores, enriquecendo o processo de aprendizagem das tecnologias de nuvem.

Ressalva que para aplicações deste tipo de conexão no campos Higienópolis Mackenzie é preciso utilizar uma conexão de internet própria ou rede que o laboratório, do seu respectivo curso, possui para estas utilizações, como feito no laboratório de engenharia do prédio 6 do Mackenzie. Por motivos de que a internet disponibilizada pelo campus traz barreiras entre algumas conexões, inclusive entre o dispositivo e a plataforma. Com isto é importante ressaltar que para conexões como a descrita acima é precisso verificar os bloqueios do Wi-Fi e utilizar redes sem bloqueios, para que todas as conexões funcionem devidamente.

6.1 Trabalho Futuro

Para trabalhos futuros, propõe-se implementar uma expansão da prática feita neste projeto para um desenvolvimento em larga escala, a fim de simular um ambiente de sala de aula. Este teste permitirá analisar o desempenho da plataforma AWS quando há múltiplos usuários integrados simultaneamente. É importante avaliar como a plataforma gerência a comunicação e processamento de dados quando há diversas aplicações ocorrendo paralelamente e a diferente quantidade de informações sendo compartilhadas entre os usuários. É possível analisar também o gerenciamento desses dispositivos com diferentes acessos, simulando acessos como de aluno e professor de formas diferentes

Com este intuito é possível começar por criações de diferentes níveis de acesso com o IAM, nele é criado uma camada adicional de segurança e gerenciamento. Por meio do IAM, é possível definir permissões específicas, garantindo que cada dispositivo ou usuário tenha acesso apenas aos recursos necessários para sua função. Isso pode ser usado para criação de diferentes níveis de acesso para alunos e professores em uma sala de aula, onde os alunos têm permissões limitadas para visualizar e interagir com certos conteúdos, enquanto os professores têm controle total sobre o ambiente e os materiais de ensino.

Para o funcionamento de uma sala de aula, múltiplos alunos podem ser simultaneamente conectados, cada um utilizando seu próprio dispositivo para compartilhar informações entre si e com o professor. Na *AWS IoT Core*, quando esses dispositivos estabelecem conexão com a plataforma, eles passam por um processo de autenticação e registro, que garante a singularidade e a segurança de cada um.

Após estabelecerem a conexão, os dispositivos IoT podem comunicar-se entre si e com a infraestrutura na nuvem. Na *AWS IoT Core*, essa comunicação é facilitada pelo protocolo MQTT, que é utilizado para permitir que os dispositivos publiquem e assinem tópicos específicos, atuando como canais de comunicação, possibilitando o envio e recebimento de mensagens em tempo real. Para a utilização da sala de aula e troca de informações entre os alunos e professores, pode ser adicionado um id

para cada sensor e *timestamp* em cada mensagem mensagem enviada para melhor controle das informações, isto é feito nas configurações da mensagem enviada.

Adicionalmente, a *AWS IoT Core* disponibiliza um conjunto abrangente de ferramentas para armazenamento, monitoramento e controle dos dispositivos conectados. A plataforma permite a definição de regras e políticas de acesso, determinando quais dispositivos podem interagir entre si, além de permitir a automação de respostas a eventos predefinidos, como alertas ou ações específicas, como por exemplo para poupar banda é possível criar uma regra para serem enviadas apenas amostras diferentes das anteriores.

Assim, a *AWS IoT Core*, em conjunto com o IAM, se destaca como uma solução extremamente eficaz para a conexão, o gerenciamento e a troca de informações entre diversos dispositivos IoT em larga escala. Tornando-a uma escolha adequada para a implementação de soluções IoT no ambiente educacional e porfissional.

Esta aplicações e testes trará mais *insights* e considerações sobre a escalabilidade e capacidade da solução no ambito educacional.

Referências

[1] L. F. S. Brienze JR, "Análise da implementação de tecnologias da nuvem Amazon web services para aplicação backend em java," *Repositorio UNESP*, Mar. 2022. [Citado nas páginas 1, 13 e 15]

[2] MASTER, "Dispositivos IoT: O que é, tipos, vantagens e exemplos." Available at `https://master.org.br/noticias/dispositivos-iot/`, 2023. (Last accessed in 23/03/2024). [Citado na página 1]

[3] G. C. Rodrigues, L. R. Galdino, and J. M. F. A. Neto, "Aplicação da computação em nuvem em pequenas e médias empresas: revisão sistemática," *Prospectus (ISSN: 2674-8576)*, vol. 1, no. 1, 2019. [Citado nas páginas 2 e 12]

[4] F. T. B. Oliveira, "Uma análise da qualidade de um serviço de armazenamento baseado em métricas de monitoramento da AWS," *Repositorio Institucional UFC)*, 2022. [Citado nas páginas 2 e 12]

[5] E. V. F. G. d. SILVA, "Os desafios e oportunidades da integração e migração de empresas com cloud computing." `https://www.cin.ufpe.br/~tg/2019-2/TG_CC/tg_evfgs.pdf`, Dec. 2019. Trabalho de Graduação - Universidade Federal de Pernambuco, Centro de Informática. [Citado nas páginas 2 e 13]

[6] A. Amazon Web Service, "AWS – O que é AWS." Available at `https://aws.amazon.com/pt/what-is-aws/`, 2024. (Last accessed in 23/03/2024). [Citado nas páginas 2, 15 e 24]

[7] TechTudo, "Internet das coisas: o que é, como funciona e exemplos de uso." Available at `https://encurtador.com.br/bfF5P`, 2023. (Last accessed in 23/03/2024). [Citado na página 3]

[8] M. Intelligence, "Mercado de IoT para consumidores - crescimento, tendências, impacto da covid-19 e previsões (2023 - 2028)." Available at `https://www.mordorintelligence.com/pt/industry-reports/consumer-iot-market`. (Last accessed in 23/03/2024). [Citado nas páginas vii, 3 e 4]

[9] T. H. N. Gartner, "30 emerging technologies that will guide your business decisions." Available at `https://www.gartner.com/en/articles/30-emerging-technologies-that-will-guide-your-business-decisions`, Feb. 2024. (Last accessed in 23/03/2024). [Citado na página 3]

[10] S. Sinha, "State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally." Available at `https://iot-analytics.com/number-connected-iot-devices/`, May 2023. (Last accessed in 23/03/2024). [Citado na página 3]

[11] M. Khalil, A. Khomonenko, and M. Matushko, "Measuring the effect of monitoring on a cloud computing system by estimating the delay time of requests," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 3968–3972, 2022. [Citado na página 4]

[12] ICMP, "Qual o volume de dados criados e qual é o futuro dos dados?." Available at `https://encurtador.com.br/NrjlM`, Feb. 2023. (Last accessed in 23/03/2024). [Citado na página 4]

[13] H. N. Security, "41.6 billion IoT devices will be generating 79.4 zettabytes of data in 2025." Available at `https://www.helpnetsecurity.com/2019/06/21/connected-iot-devices-forecast/`, June 2019. (Last accessed in 23/03/2024). [Citado nas páginas 4 e 5]

[14] M. Taghipour, M. Soofi, M. Mahboobi, and J. Abdi, "Application of cloud computing in system management in order to control the process," *Management*, vol. 3, no. 3, pp. 34–55, 2020. [Citado nas páginas 5, 11 e 12]

[15] M. A. I. Vinhaes, Roberto; Viuge, "Amazon, Microsoft, Google ou Alibaba: quem vai vencer a disputa na nuvem?." Available at `https://encurtador.com.br/CnEgB`, Aug. 2022. (Last accessed in 23/03/2024). [Citado nas páginas vii, 5 e 14]

[16] P. Kaushik, A. M. Rao, D. P. Singh, S. Vashisht, and S. Gupta, "Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft Azure, and Google Cloud," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 268–273, IEEE, 2021. [Citado nas páginas ix, 5 e 14]

[17] D. Fu, L. Chen, and Z. Cheng, "Integration of wearable smart devices and internet of things technology into public physical education," *Mobile Information Systems*, vol. 2021, pp. 1–10, 2021. [Citado na página 10]

[18] F. S. Tortoriello and I. Veronesi, "Internet of things to protect the environment: a technological transdisciplinary project to develop mathematics with ethical effects," *Soft Computing*, vol. 25, no. 13, pp. 8159–8168, 2021. [Citado na página 10]

[19] IPM, "História da computação em nuvem: como surgiu a cloud computing?." Available at `https://www.ipm.com.br/blog/administracao-geral/`

historia-da-computacaoem-nuvem-como-surgiu-a-cloud-computing/, Nov. 2023. (Last accessed in 23/03/2024). [Citado na página 11]

[20] M. E. L. d. Silva, "Modelagem de ecossistemas de software das plataformas de computação em nuvem: Amazon web services e Google cloud platform," *Repositorio Institucional UFC)*, 2022. [Citado na página 11]

[21] E. Simmon *et al.*, "Evaluation of cloud computing services based on NIST SP 800-145," *NIST Special Publication*, vol. 500, no. 322, pp. 500–322, 2018. [Citado na página 11]

[22] L. de Paula and M. de Oliveira Dian, "Computação em nuvem: os desafios das empresas ao migrar para a nuvem," *Revista Interface Tecnológica*, vol. 18, no. 2, pp. 304–315, 2021. [Citado na página 11]

[23] M. Ansari, S. A. Ali, and M. Alam, "Internet of things (IoT) fusion with cloud computing: current research and future direction," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 97, p. 1812, 2022. [Citado na página 12]

[24] Microsoft, "O que é computação em nuvem?." Available at `https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-cloud-computing`, 2024. (Last accessed in 23/03/2024). [Citado nas páginas 12 e 13]

[25] DSSOF/CCUEC/UNICAMP, "O que é computação em nuvem?." Available at `https://suporte.nuvem.unicamp.br/sobre/o_que_e.html`, 2020. (Last accessed in 23/03/2024). [Citado na página 12]

[26] Amazon Web Service, "AWS – Infraestrutura global da AWS." Available at `https://aws.amazon.com/pt/about-aws/global-infrastructure/?p=ngi&loc=1`, 2024. (Last accessed in 23/03/2024). [Citado nas páginas vii e 15]

[27] Amazon Web Service, "AWS – IoT Core." Available at `https://aws.amazon.com/pt/iot-core/`, 2024. (Last accessed in 23/03/2024). [Citado nas páginas 16 e 24]

[28] Amazon Web Services, "AWS IoT analytics." Available at `https://aws.amazon.com/pt/iot-analytics/`, 2024. (Last accessed in 22/04/2024). [Citado na página 16]

[29] Amazon Web Services, "AWS IoT device management." Available at `https://aws.amazon.com/pt/iot-device-management/`, 2024. (Last accessed in 22/04/2024). [Citado na página 16]

[30] Amazon Web Services, "AWS greengrass." Available at https://aws.amazon.com/pt/greengrass/, 2024. (Last accessed in 22/04/2024). [Citado na página 16]

[31] Amazon Web Service, "AWS – Lambda." Available at https://aws.amazon.com/pt/lambda/?nc2=type_a, 2024. (Last accessed in 23/03/2024). [Citado na página 16]

[32] Amazon Web Services, "AWS IoT 1-click." Available at https://aws.amazon.com/pt/iot-1-click/, 2024. (Last accessed in 22/04/2024). [Citado na página 16]

[33] Microsoft, "Microsoft azure." Available at https://azure.microsoft.com/pt-br/free/search/, 2024. (Last accessed in 23/03/2024). [Citado na página 17]

[34] Microsoft Azure, "Azure IoT hub." Available at https://azure.microsoft.com/pt-br/products/iot-hub/, 2024. (Last accessed in 22/04/2024). [Citado na página 17]

[35] Microsoft Azure, "Azure IoT central." Available at https://azure.microsoft.com/pt-br/products/iot-central/, 2024. (Last accessed in 22/04/2024). [Citado na página 17]

[36] Microsoft Azure, "Azure IoT edge." Available at https://azure.microsoft.com/pt-br/products/iot-edge/, 2024. (Last accessed in 22/04/2024). [Citado na página 17]

[37] Microsoft Azure, "Azure sphere." Available at https://azure.microsoft.com/pt-br/products/azure-sphere/, 2024. (Last accessed in 22/04/2024). [Citado na página 18]

[38] Microsoft Azure, "Documentação da análise de séries temporais do azure." Available at https://learn.microsoft.com/pt-br/azure/time-series-insights/, 2024. (Last accessed in 22/04/2024). [Citado na página 18]

[39] Microsoft Azure, "Azure stream analytics." Available at https://azure.microsoft.com/pt-br/products/stream-analytics/, 2024. (Last accessed in 22/04/2024). [Citado na página 18]

[40] Google, "Google – Cloud." Available at https://cloud.google.com/?hl=pt-br, 2024. (Last accessed in 23/03/2024). [Citado na página 18]

[41] Google Cloud Platform, "Overview of the pub/sub service." Available at https://cloud.google.com/pubsub/docs/pubsub-basics, 2024. (Last accessed in 22/04/2024). [Citado na página 19]

[42] Google Cloud Platform, "Google cloud dataflow." Available at `https://cloud.google.com/dataflow?hl=pt_br`, 2024. (Last accessed in 22/04/2024). [Citado na página 19]

[43] Google Cloud Platform, "Bigquery overview." Available at `https://cloud.google.com/bigquery/docs/introduction`, 2024. (Last accessed in 22/04/2024). [Citado na página 19]

[44] Google Cloud Platform, "Google cloud functions documentation." Available at `https://cloud.google.com/functions/docs?hl=pt-br`, 2024. (Last accessed in 22/04/2024). [Citado na página 19]

[45] Google Cloud Platform, "Google cloud firestore." Available at `https://cloud.google.com/firestore?hl=pt_br`, 2024. (Last accessed in 22/04/2024). [Citado na página 19]

[46] E. Systems, "Esp32 series." Available at `https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf`, 2024. (Last accessed in 18/04/2024). [Citado nas páginas 21 e 22]

[47] DOIT, "Doit esp32 devkit v1." Available at `https://olddocs.zerynth.com/r2.5.2/official/board.zerynth.doit_esp32/docs/index.html`, 2024. (Last accessed in 18/04/2024). [Citado nas páginas vii, 22 e 23]

[48] B. Perumal, J. Deny, K. Alekhya, V. Maneesha, and M. Vaishnavi, "Air pollution monitoring system by using Arduino IDE," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 797–802, IEEE, 2021. [Citado nas páginas 22 e 24]

[49] RoboCore, "Sensor DHT11 de temperatura e umidade." Available at `https://www.robocore.net/sensor-ambiente/sensor-de-temperatura-dht11`, 2022. (Last accessed in 18/04/2024). [Citado na página 23]

[50] A. S. Ismailov, Z. B. Jo'Rayev, *et al.*, "Study of arduino microcontroller board," *Science and Education*, vol. 3, no. 3, pp. 172–179, 2022. [Citado na página 23]

[51] Amazon Web Services, "Overview of Amazon web services." Available at `https://docs.aws.amazon.com/whitepapers/latest/aws-overview/introduction.html`, 2024. (Last accessed in 18/04/2024). [Citado na página 24]

[52] Y. Tran, "Integrate AWS IoT core and device shadow in embedded instrument implementation with paho MQTT libraries," *THESEUS*, April 2023. [Citado na página 25]

[53] Amazon Web Services, "What is AWS IoT?." Available at `https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html`, 2024. Last accessed in 23/07/2024. [Citado nas páginas vii e 25]

[54] P. Khot, S. Padalkar, V. Raghvani, and A. Agarwal, "Cloud migration with Google cloud storage and AWS (S3 service)," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 10, pp. 555–558, 2020. [Citado na página 26]

[55] Amazon Web Service, "Amazon S3." Available at `https://aws.amazon.com/pt/s3/?nc2=type_a`, 2024. (Last accessed in 23/03/2024). [Citado na página 26]

[56] A. Koschel, I. Astrova, A. Pakosch, C. Gerner, C. Schulze, and M. Tyca, "Is Amazon kinesis data analytics suitable as core for an event processing network model?," in *16th International Conference on Agents and Artificial Intelligence, ICAART 2024, Rome*, pp. 1036–1043, SciTePress, 2024. [Citado na página 26]

[57] Amazon Web Service, "Amazon Kinesis." Available at `https://aws.amazon.com/pt/kinesis/?nc2=type_a`, 2024. (Last accessed in 23/03/2024). [Citado na página 27]

[58] Amazon Web Services, "Amazon data firehose." Available at `https://aws.amazon.com/pt/firehose/`, 2024. (Last accessed in 18/04/2024). [Citado na página 27]

[59] R. K. Indla, "An overview on Amazon rekognition technology," *CSUSB*, May 2021. [Citado na página 27]

[60] Amazon Web Services, "AWS identity and access management (IAM)." Available at `https://aws.amazon.com/pt/iam/`, 2023. (Last accessed in 18/04/2024). [Citado na página 28]

[61] B. E. L. Silva, "Serviços de segurança em computação em nuvem usando a plataforma Amazon web service (AWS)," *Repositorio PUC Goias*, June 2023. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação). [Citado na página 28]

[62] F. Sousa, R. Soares, and R. Campos, "IIOT utilizando protocolo MQTT," *UNA Pouso Alegre*, 2021. [Citado nas páginas ix, 29 e 31]

[63] OASIS, "Mqtt version 5.0." Available at `https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html`, 2019. Last accessed in 2024-08-12. [Citado nas páginas 29 e 30]

[64] M. Pinto, G. Júnior, and G. Viajante, "Supervisório para controle e monitoramento de aplicações industriais utilizando o protocolo de comunicação MQTT,"

in *XVII CEEL – Conferência de Estudos em Engenharia Elétrica*, 11 2019. [Citado na página 30]

[65] M. A. Spohn and W. B. Genero, "Análise experimental dos protocolos MQTT e MQTT-SN," *Revista Brasileira de Computação Aplicada*, vol. 15, no. 1, pp. 22–33, 2023. [Citado na página 30]

[66] W. dos Reis Bezerra and C. B. Westphall, "Avaliação de desempenho de protocolos de mensagens com arquitetura publish/subscribe no ambiente de computação em nevoeiro: um estudo sobre desempenho do MQTT, AMQP e STOMP," in *Anais do XI Workshop de Pesquisa Experimental da Internet do Futuro*, (Chengdu, China), pp. 7–12, SBC, 2020. [Citado na página 31]

[67] G. Soto, "Conexão com AWS iot core." `https://github.com/GabrieleSoto/Conexao-com-AWS-IoT-Core`, 2024. (Last accessed in 18/05/2024). [Citado nas páginas 34 e 61]

[68] E. Systems, "Esp32 package index." `https://dl.espressif.com/dl/package_esp32_index.json`. (Last accessed in 18/04/2024). [Citado na página 36]

[69] Amazon Web Services, "AWS command line interface." `https://aws.amazon.com/pt/cli/`, 2024. Last accessed in 23/07/2024. [Citado na página 52]