UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL BRUNO PEREIRA NEGRI
Orientador: Prof. Dr. Carlos Frederico Marcelo da Cunha Cavalcanti

VULNERABILIDADES EM IOT: DISPOSITIVOS ECHO

Ouro Preto, MG
2025

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

GABRIEL BRUNO PEREIRA NEGRI

VULNERABILIDADES EM IOT: DISPOSITIVOS ECHO

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Carlos Frederico Marcelo da Cunha Cavalcanti

Ouro Preto, MG
2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Gabriel Bruno Pereira Negri

Vulnerabilidades em Iot: Dispositivos Echo

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 3 de Abril de 2025.

Membros da banca

Doutor Carlos Frederico Marcelo da Cunha Cavalcanti (Orientador) - Universidade Federal de Ouro Preto
Doutor Ricardo Augusto Rabelo Oliveira (Examinador) - Universidade Federal de Ouro Preto
Doutor Fernando Cortez Sica (Examinador) - Universidade Federal de Ouro Preto

Carlos Frederico Marcelo da Cunha Cavalcanti, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 16/04/2025.

Documento assinado eletronicamente por **Carlos Frederico Marcelo da Cunha Cavalcanti**, **CHEFE DO DEPARTAMENTO DE COMPUTAÇÃO**, em 22/04/2025, às 19:04, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0886265** e o código CRC **7B35D38E**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Dedico este trabalho a todos aqueles que tenham interesse no assunto relacionado a segurança de dispositivos IoT, e desejam dar continuidade a estes estudos.

Agradecimentos

Agradeço primeiramente a Deus por este ciclo que se encerra.

Também à minha família, **Valéria**, **Bruno** e **Ricardo** pelos suportes e incentivos ao longo de toda minha graduação, vocês foram muito essenciais! Amo muito vocês!

Agradeço à três antigos colegas de trabalho, **Maycon**, **Alex** e **Wanderley**, que me mostraram sobre esse mundo de *Network, Edge/Cloud Computing*, o que causou um grande interesse em fazer este trabalho num tema relacionado.

Agradeço à **UFOP (Universidade Federal de Ouro Preto)**, pelo ensino gratuito e de qualidade.

Agradeço ao meu orientador **Carlos Frederico Marcelo da Cunha Cavalcanti**, pelo apoio ao longo de todo o desenvolvimento deste projeto.

E por fim mas não menos importante, agradeço à cada pessoa que me encorajou a não desistir do curso, cada apoio valeu muito a pena para que esta etapa fosse concluída.

“(...) se tivéssemos computadores que conhecessem tudo o que existe para se saber sobre as coisas reais - usando dados que eles mesmos agrupem, sem nossa ajuda – nós poderíamos, por exemplo, acompanhar tudo, o que reduziria imensamente o desperdício, perdas e custos.”(Projetos, 2024)

Resumo

Este estudo, uma pesquisa bibliográfica, concentra-se na segurança de dispositivos *IoT* (Internet das Coisas), especialmente sobre os dispositivos *Echo*, da Amazon, que tem como assistente virtual a Alexa, e são produtos que são *Edge Computing*. O objetivo principal é investigar sobre as diversas formas de ataques cibernéticos em dispositivos *IoT*, com ênfase na proteção de dados sensíveis. A pesquisa identifica ameaças (nos dispositivos *IoT* de forma geral e também aos dispositivos *Echo*, da *Alexa*) será apresentado um mecanismo de autenticação e controle de acesso em um sistema feito de forma genérica, como forma de propôr recomendações práticas para aprimorar a segurança. Além disso, o estudo aborda os componentes dos sistemas *Cyber-Physical (CPS)*. A pesquisa destaca a importância da segurança nos dispositivos *Echo*, e a eficiência em redes industriais e sistemas de automação. As conclusões deste trabalho fornecerão orientações cruciais para proteger dados sensíveis em um ambiente *IoT* na *Edge Computing*, contribuindo para a integridade e confidencialidade das informações em um mundo cada vez mais conectado.

Palavras-chave:IoT. *Edge Computing*, Segurança de dados, *Cyber-Physical Systems (CPS)*, *Alexa*, dispositivos *Echo*.

Abstract

This study, a bibliographical survey, focuses on the security of IoT (Internet of Things) devices, especially Amazon's Echo devices, whose virtual assistant is Alexa, and which are Edge Computing products. The main objective is to investigate the various forms of cyber attacks on IoT devices, with an emphasis on protecting sensitive data. The research identifies threats to IoT devices in general and also to Alexa's Echo devices. A generic authentication and access control mechanism will be presented as a way of proposing practical recommendations for improving security. In addition, the study looks at the components of Cyber-Physical Systems (CPS). The research highlights the importance of security in Echo devices and efficiency in industrial networks and automation systems. The conclusions of this work will provide crucial guidelines for protecting sensitive data in an IoT environment on Edge Computing, contributing to the integrity and confidentiality of information in an increasingly connected world.

Keywords:IoT, Edge Computing, Data security, Cyber-Physical Systems (CPS), Alexa, Echo devices.

Lista de Ilustrações

Sumário

1 Introdução

A *Edge Computing* e a Internet das Coisas *IoT* têm revolucionado a forma como interagimos com o mundo digital e físico. A convergência dessas tecnologias deu origem a um ecossistema altamente interconectado, onde dispositivos *IoT* desempenham um papel central na coleta, processamento e transmissão de dados em tempo real. Ao mesmo tempo, a *Edge Computing*, como uma extensão da computação em nuvem, trouxe a capacidade de processamento mais próximo da fonte de dados, criando um ambiente onde decisões críticas podem ser tomadas instantaneamente.

Um exemplo de estudo de caso que utiliza *IoT* é a implementação de cidades inteligentes (*Smart Cities*). Nesse contexto, a *IoT* é empregada para coletar e analisar dados em tempo real a partir de sensores distribuídos pela cidade. Esses sensores podem incluir dispositivos de monitoramento de tráfego, sensores ambientais, câmeras de vigilância, medidores inteligentes e outros dispositivos conectados.

Imaginemos uma cidade que busca otimizar o tráfego e aprimorar a gestão de recursos. Um estudo de caso consistiria na instalação de sensores de tráfego em pontos estratégicos, conectados à dispositivos *IoT* para coleta contínua de dados sobre volume, padrões de movimentação e tempos de espera. Com base nessas informações, algoritmos adaptativos de controle de tráfego poderiam ser implementados em tempo real, otimizando os tempos de semáforos e reduzindo congestionamentos. Além disso, os dados coletados permitiriam identificar padrões de tráfego ao longo do tempo, orientando o planejamento urbano e a implementação de melhorias na infraestrutura viária.

Um exemplo de estudo de caso é a instalação de sensores de tráfego em pontos estratégicos de uma cidade para coleta de dados em tempo real. Esses dados podem ser usados por algoritmos adaptativos para otimizar semáforos, reduzir congestionamentos e apoiar o planejamento urbano, identificar padrões de tráfego ao longo do tempo, auxiliando no planejamento urbano e na implementação de melhorias na infraestrutura viária. Esses sensores seriam conectados à *IoT*, permitindo a coleta contínua de dados sobre o volume de tráfego, padrões de movimento, tempos de espera e congestionamentos.

A *Edge Computing*, também conhecida como "Computação de Borda", diferentemente da computação em nuvem tradicional, leva o processamento e o armazenamento de dados para mais perto dos dispositivos *IoT* e dos pontos de coleta de informações. Isso permite uma redução significativa na latência, tornando-a ideal para aplicações em tempo real, como automação industrial, cidades inteligentes, saúde digital e muito mais. No entanto, essa proximidade aos dispositivos *IoT* também introduz desafios complexos de segurança.

A crescente proliferação de dispositivos *IoT*, conectados a redes industriais e residenciais,

exige uma abordagem minuciosa para garantir a segurança dos dados sensíveis que eles coletam e transmitem. A interseção entre *Edge Computing* e *IoT* é onde os riscos e as oportunidades da segurança de dados se tornam mais evidentes. Neste contexto, este trabalho aborda a segurança de dados em dispositivos *IoT* na abordagem *Edge Computing*, explorando as melhores práticas, os desafios e as soluções necessárias para proteger informações sensíveis nesse ambiente complexo. Ao abordar o contexto de sistemas embarcados e *IoT*, os seguintes elementos são cruciais a serem considerados:

Um dos dispositivos *IoT* que são amplamente usados em lares e comércios em todo o mundo é o *Amazon Echo* que, na realidade, é uma família de dispositivos *IoT*, geralmente denominados simplesmente pelo nome de *"Echo"* ou até mesmo de *Alexa*, porém *Alexa* é a assistente virtual da *Amazon*, enquanto o *Echo* é um alto-falante inteligente que usa a *Alexa*. Apesar de *Alexa* e *Echo* serem distintos, muitas vezes os termos são usados como sinônimos pois todos os dispositivos Echo usam Alexa mas nem todo dispositivo que usa *Alexa* é um *Echo*, a exemplo de *Smart TVs* equipados com *Alexa*. Dispositivos *Amazon Echo*, assim como a maioria de dispositvos *IoT*, também apresentam vulnerabilidades no que diz respeito à segurança, a exemplo de falsificação de comandos de voz, abuso de habilidades de terceiros e escuta passiva, expõem os usuários a riscos de invasões e perda de dados, entre outros. Este estudo busca analisar quais são as principais vulnerabilidades associadas aos dispositivos *IoT* e aos dispositivos Echo, e apresentar um estudo de caso, criando-se uma *skill* "dormente"(inativa) que vai ser usada e posteriormente manipulada. As *skills* da *Alexa* são aplicativos ativados por voz que adicionam recursos ao dispositivo compatível com a *Alexa*.

1.1 Justificativa

A realização deste trabalho é intrinsecamente ligada à crescente importância da segurança de dispositivos *IoT* na *Edge Computing* em um mundo cada vez mais conectado. A *Edge Computing*, ao permitir o processamento próximo à fonte de dados, aprimora a eficiência e a capacidade de resposta em tempo real, sendo usada em inúmeras aplicações, desde a automação industrial até a assistência médica digital. No entanto, essa proximidade aos dispositivos *IoT* também traz consigo desafios significativos de segurança.

A proliferação massiva de dispositivos *IoT* e, até mesmo para este estudo que se trata dos dispositivos Echo, pode-se observar que coletam e transmitem dados sensíveis em uma variedade de setores da vida humana, o que intensifica a necessidade de medidas rigorosas de segurança. Os ataques cibernéticos[1] direcionados a esses dispositivos são uma realidade, com implicações que vão desde a violação da privacidade até a interrupção de operações críticas. A segurança dos dispositivos *IoT* na *Edge Computing* é essencial para garantir a integridade, confidencialidade e disponibilidade dos dados nesse ambiente altamente conectado.

[1] Um ciberataque, também denominado como ataque cibernético, é qualquer tentativa de expor, desativar, roubar, alterar, destruir, roubar ou obter acesso não autorizado ou fazer uso não autorizado de um dispositivo.

Este trabalho tem como proposta abordar questões críticas de segurança, identificando ameaças, avaliando técnicas de proteção, e propondo melhores práticas de segurança que possam ser aplicadas em dispositivos *IoT*. Além de um estudo prático com o Dispositivo Echo. Ao compreender a justificativa intrínseca a este tema, este trabalho busca contribuir para um ambiente digital mais seguro, protegendo dados sensíveis e permitindo que a *Edge Computing* e a *IoT* continuem a impulsionar a inovação em diversas áreas, mantendo a integridade e a confiabilidade das operações.

1.2 Objetivos

Este trabalho tem como objetivo principal investigar e analisar as possíveis e principais vulnerabilidades em dispositivos *IoT* de forma geral e em específico à Amazon Echo, sobre a proteção de dados sensíveis em um ambiente de rápida evolução tecnológica e tem como objetivos específicos:

- Apontar **ameaças de segurança em dispositivos *IoT***: Realizar uma análise abrangente das principais ameaças à segurança que afetam dispositivos *IoT* na *Edge Computing*, bem como os dispositivos Echo, levando em consideração o ambiente complexo e altamente conectado em que operam.
- Apresentar um estudo prático sobre um sistema capaz de realizar a autenticação do usuário de forma segura, comprovando que ele é, de fato, quem afirma ser no momento do cadastro.
- Definir um processo válido e seguro, utilizando um dispositivo Echo usando Alexa da Amazon, para ser o validador do processo de autenticação,
- Implementar o processo de autenticação definido, como PoC (*Proof of Concept*), através de ***Alexa Skills Kit (SDK)***, com a realização de testes voltados à retenção de dados e à autenticação do usuário pelo próprio dispositivo.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma: o capítulo 1 contextualiza o trabalho mostrando os objetivos a serem alcançados, o capitulo 2 é uma revisão bibliográfica com embasamento teórico sobre o tema (contextualizando primeiramente de forma mais geral e depois especificando para a questão dos próprios dispositivos Echo), no capítulo 3, o Desenvolvimento, que abordará quais serão os materiais e métodos utilizados para o estudo, além dos resultados apresentados e o capítulo 4, a conclusão com considerações finais.

2 Revisão Bibliográfica

Este capítulo é dedicado à revisão bibliográfica onde será apresentada a contextualização da pesquisa com um resumo das discussões já realizadas por outros autores sobre o tema "vulnerabilidades em dispositivos *IoT*."

2.1 Trabalhos Relacionados

Nesta seção, serão destacados os principais pontos levantados por autores anteriores sobre o tema de vulnerabilidades em dispositivos *IoT*. Alguns autores já abordaram essa questão, enfocando aspectos relevantes que veremos abaixo.

A Internet das Coisas (*Internet of Things - IoT)* é um conjunto de dispositivos com sensores, capacidade de processamento, software e outras tecnologias que conectam e trocam dados com outros dispositivos e sistemas pela Internet ou outras redes de comunicação. Muitos consideram o termo inadequado pois dispositivos *IoT* podem estar ligados a outras redes e não necessariamente à Internet.(Ekanayake; Halgamuge; Syed, 2018)

Com a aumento do poder computacional e a diminuição do custo dos componentes eletrônicos, observa-se um crescimento explosivo de dispositivos *IoT* conectados e controlados via Internet. *IoT* consolidou-se como uma tecnologia habilitadora (*enabler tecnology*) com o propósito de revolucionar processos industriais, automação e gerenciamento inteligente em diversas áreas

O uso da *Fog Computing* (Computação em Névoa) em uma planta com dispositivos *IoT* é uma abordagem promissora para enfrentar desafios de segurança e privacidade. A *Fog Computing* destaca-se por oferecer proximidade e capacidade de processamento necessárias para suportar aplicativos em tempo real, ao mesmo tempo em que fortalece a segurança dos dispositivos *IoT*. A computação em névoa permite que serviços de ponta de próxima geração forneçam ampla gama de aplicações para dispositivos de Internet das Coisas (*IoT*). Ele também suporta agilidade, localização, heterogeneidade, escalabilidade, baixa latência e distribuição geográfica. Em termos gerais, o objetivo do paradigma da computação em neblina é reduzir a quantidade de dados e tráfego para servidores em nuvem, reduzir a latência e melhorar a qualidade do serviço (QoS). Em suma, a *Fog Computing* visa aumentar a eficiência e as capacidades de serviço. Ele oferece benefícios como redução da quantidade de dados enviados para servidores em nuvem e redução da latência.(Alrawais *et al.*, 2017)

A diferenciação fundamental entre *Fog* e *Edge computing* reside na localização e na hierarquia dos recursos computacionais. No *Fog Computing*, o processamento ocorre em nós intermediários — como *gateways* e servidores dedicados — que ficam entre os dispositivos

finais e a nuvem central, permitindo uma análise agregada dos dados em vários níveis. Já o *Edge Computing* desloca o processamento para a borda da rede, onde os dispositivos geram os dados, assegurando respostas em tempo real, porém com recursos computacionais geralmente mais limitados.

Em termos de aplicação, *Fog Computing* é ideal para cenários que exigem uma distribuição equilibrada do processamento e uma abordagem hierárquica, como em cidades inteligentes e sistemas industriais complexos. Por outro lado, *Edge Computing* é preferido em situações que demandam latência mínima, como veículos autônomos e aplicações críticas em tempo real. Assim, a escolha entre os paradigmas depende dos requisitos específicos de desempenho e da infraestrutura disponível

A aplicação de dispositivos e *frameworks*[1], dentro de uma arquitetura de *IoT* tem sido cada vez mais adotada nas infraestruturas de redes, devido às vantagens operacionais que oferece segundo (Soares, 2022). Devido a crescente demanda por soluções inteligentes, eficientes e altamente interoperáveis, a integração de métodos, sistemas e soluções, sejam novos ou legados, tornou-se essencial tanto para negócios públicos quanto privados

As redes de dispositivos *IoT* têm se tornado uma presença funcional significativa em diversos segmentos, incluindo na implementação de casas inteligentes (*smart homes*), para sistemas de automação e gerenciamento de energia, sistemas de saúde para monitoramento de condições físicas através de sensores, agropecuária para avaliação de parâmetros temporais por meio de sensores e fábricas para gerenciamento de produtividade e monitoramento de equipamentos.

Para lidar com a grande quantidade de dados na comunicação de dispositivos *IoT*, são utilizadas arquiteturas de Internet das Coisas, como *Cloud*, *Fog* e *Edge Computing*. O *Edge Computing*, em particular, torna possível o tratamento de dados para fins de visualização e integração, além de processamento computacional de redes externas, enquanto o *Cloud Computing* é amplamente utilizado para servidores com menor custo computacional e maior eficiência no armazenamento de dados (Soares, 2022)

As redes *IoT* enfrentam desafios significativos, incluindo questões de segurança, privacidade, interoperabilidade, escalabilidade de serviços e disponibilidade de equipamentos (Soares, 2022).

A segurança é fundamental, já que os dados dos dispositivos *IoT* são transmitidos e armazenados externamente, exigindo criptografia, controle de acesso e backup para garantir a confidencialidade, integridade e disponibilidade. A escalabilidade é um desafio importante, pois as redes *IoT* precisam ajustar recursos conforme a demanda, exigindo arquiteturas flexíveis e alocação eficiente de recursos. A interoperabilidade é crítica, mas a falta de padrões dificulta a comunicação entre diferentes dispositivos e plataformas, necessitando de interfaces comuns para

[1] *frameworks* são conjuntos de softwares que provem uma determinada facilidade, uma determinada abstração, capazes de automatizar processos cotidianos tem se destacado

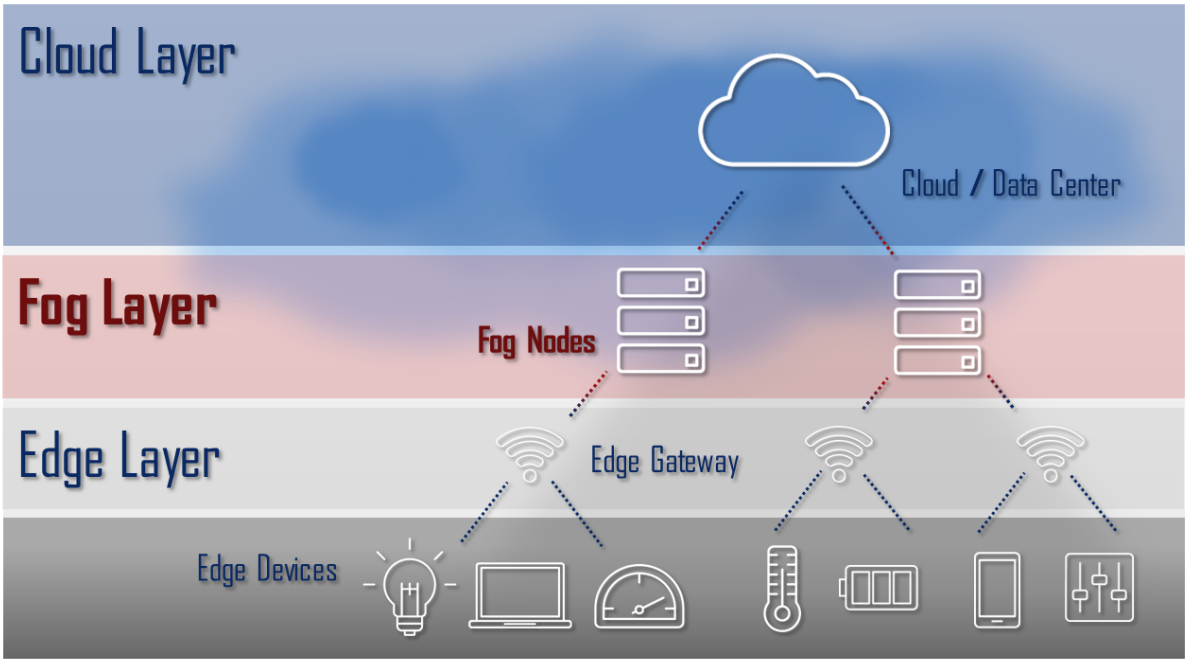

Figura 2.1 – Ilustração de *Cloud*, *Fog* e *Edge Computing*. Fonte: Ionos (2024).

garantir flexibilidade. A confiabilidade é essencial, com a necessidade de backup e redundância para garantir que os dispositivos funcionem mesmo em caso de falhas. Por fim, a disponibilidade contínua dos equipamentos é crucial, sendo necessário o uso de arquiteturas distribuídas e mecanismos de tolerância a falhas para evitar interrupções. (Felix, 2023)

Os autores referenciados anteriormente oferecem uma base teórica sólida para a presente investigação sobre vulnerabilidades em dispositivos *IoT* no contexto da *Edge Computing*. Seus trabalhos evidenciam tanto a relevância das arquiteturas *IoT*, *Edge*, *Fog* e *Cloud Computing*, quanto os desafios associados à segurança e à privacidade nesses ambientes distribuídos. Este trabalho tem como objetivo aprofundar a análise dessas vulnerabilidades, com ênfase nos dispositivos *IoT* e, especialmente, nos dispositivos Echo da Amazon. A proposta inclui ainda a realização de um experimento que permitirá demonstrar, na prática, a presença e os impactos dessas vulnerabilidades.

Para isso é interessante citar o trabalho de (Su *et al.*, 2020), aonde o estudo foca em revelar vulnerabilidades de segurança e privacidade no ecossistema de *skills* da Alexa. Os pesquisadores exploraram como desenvolvedores mal-intencionados podem abusar da forma como a Amazon revisa e publica essas habilidades (*skills*).

Eles descobriram que, embora o front-end de uma *skill* (parte que define os comandos de voz e interações) seja revisado, o *back-end* (onde o código realmente roda) permanece invisível para a Amazon. Isso permite que desenvolvedores modifiquem o código do *back-end* após a *skill* ser aprovada, sem que essas mudanças sejam detectadas. Com isso, eles podem, por exemplo, coletar informações sensíveis dos usuários, como número de telefone e endereço, ou até mesmo

alterar a função da *skill* para algo malicioso após a publicação.

Na arquitetura de sistemas modernos, há uma distinção crescente entre computação realizada perto da borda (*near Edge*) e aquela feita mais distante dela (*far Edge*). Esse posicionamento estratégico afeta diretamente a eficiência operacional, a latência dos dados e a capacidade de resposta dos sistemas às demandas dos usuários e dispositivos conectados.

- Elementos perto da *Edge*: integrados à infraestrutura entre a *Edge* mais longe e as camadas de nuvem, sistemas próximos à borda podem coexistir com a infraestrutura de operadoras de WAN, como a instalação de hardware em torres de celular e estações de comutação celular. Nesta camada, é viável hospedar serviços computacionalmente complexos.(Lea, 2020)
- Elementos distantes à *Edge*: envolvem dispositivos de processamento com capacidade de comunicação, gerenciamento e troca de dados com a nuvem e/ou dispositivos próximos à *Edge*. Essa camada está situada mais distante da nuvem em geral, mas ainda mantém uma conexão com a nuvem e seus componentes próximos à *Edge*. Ela se encontra mais próxima dos usuários finais ou sistemas de sensores, tendo requisitos específicos, como design de tempo real rigoroso ou crítico para a segurança, e pode atuar como uma gateway para grandes redes PAN[2] (Lea, 2020).

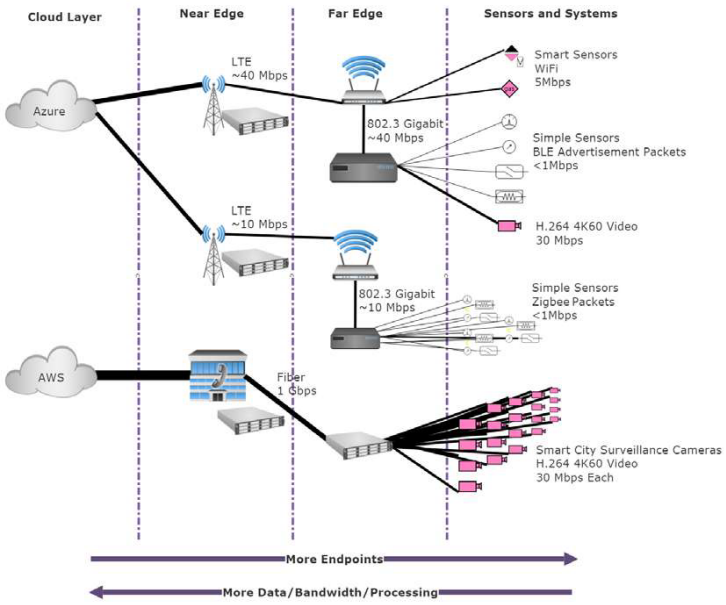

Figura 2.2 – Diferença na proximidade e nos recursos entre computação próxima à *Edge* (*near Edge* e longe da *Edge* (*far Edge*.) Fonte: Lea (2020).

Com base em estudos recentes a partir de 2021, pode-se observar que a diferenciação na localização dos recursos na arquitetura de sistemas modernos gera impactos significativos em

[2] PAN é o acrônimo de *Personal Area Network*, referindo-se a redes de curto alcance, geralmente usadas para comunicação entre dispositivos pessoais.

Figura 2.3 – Microcontrolador ESP32. Fonte: Savarati(2025)

aspectos como a latência, a eficiência operacional e a capacidade de resposta dos dispositivos. Elementos localizados perto da *Edge*, por exemplo, costumam se integrar à infraestrutura de operadoras, sendo instalados em torres de celular ou estações de comutação, o que possibilita a hospedagem de serviços computacionalmente intensos com menor latência. Em contraste, elementos distantes da *Edge* envolvem dispositivos que se conectam à nuvem ou a outros dispositivos próximos, atendendo a requisitos rigorosos de *real-time* e segurança, atuando muitas vezes como gateways para redes maiores.

No contexto específico dos microcontroladores ESP32, identificou-se a existência de vulnerabilidades que podem comprometer a segurança dos dispositivos. Estudos apontam casos de exploração devido à implementação inadequada de funcionalidades de segurança, como o *secure boot* e algoritmos de criptografia, facilitando ataques que podem resultar em execução arbitrária de código ou roubo de dados. Esses casos evidenciam a necessidade de melhorias tanto em nível de hardware quanto de software para mitigar tais riscos, permitindo que os dispositivos atendam com segurança às crescentes demandas do ambiente IoT (Al-Mashhadani; Shujaa, 2022).

Em comparação com os trabalhos anteriores, a pesquisa atual se concentra em um tópico específico de grande relevância: as vulnerabilidades em dispositivos *IoT* na *Edge*. Acredita-se que a pesquisa pode complementar os resultados existentes, oferecendo uma visão mais aprofundada e focada em um aspecto crítico da segurança na *IoT*. Dessa forma, espera-se contribuir para o desenvolvimento do conhecimento e fornecer *insights* valiosos para a segurança desses dispositivos na *Edge Computing*.

2.2 Fundamentação Teórica

Nesta seção, será fornecida uma fundamentação teórica para explicar os conceitos-chave discutidos ao longo do texto. O objetivo é situar o leitor, especialmente aquele que pode não estar familiarizado com os termos técnicos, para que esteja preparado para a leitura do capítulo de metodologia e para entender a pesquisa em sua totalidade.

A Internet das Coisas (*IoT*) representa um avanço significativo na conectividade digital, onde objetos do dia a dia se integram à Internet para coletar e trocar dados. Sensores, câmeras, dispositivos médicos, veículos e uma variedade de outros dispositivos *IoT* permitem automação,

monitoramento contínuo e tomada de decisões baseadas em dados em tempo real.

O *Edge Computing* surge como uma abordagem inovadora na computação distribuída, movendo o processamento de dados para mais próximo da fonte de origem, ou seja, nos dispositivos de borda. Esta proximidade reduz a latência e permite ações imediatas, essenciais para aplicações sensíveis ao tempo como sistemas de controle industrial e cidades inteligentes.

Em contrapartida, o *Cloud Computing* oferece escalabilidade e eficiência ao armazenar e processar dados em servidores remotos acessados pela Internet. Essa infraestrutura é fundamental para suportar aplicações globais e demandas computacionais intensas.

O *Fog Computing* expande ainda mais o conceito de *Edge Computing* ao introduzir uma camada intermediária entre os dispositivos de borda e a nuvem. Esta camada possui capacidade de processamento adicional em comparação aos dispositivos de borda, facilitando uma distribuição otimizada de tarefas e análise de dados em tempo real.

O *Alexa Skills Kit (ASK)* é um conjunto de ferramentas da *Amazon* que permite aos desenvolvedores criar habilidades personalizadas para a Alexa. Essas habilidades ampliam suas funções, como controlar dispositivos, fornecer informações e realizar serviços.

A segurança da *IoT* na *Edge Computing* é crucial para proteger dispositivos *IoT* e os dados sensíveis que eles manipulam. Medidas como autenticação robusta, criptografia forte e detecção de intrusões são implementadas para mitigar vulnerabilidades, como falhas de segurança no software e configurações inadequadas, que podem ser exploradas por invasores.

Identificar e abordar vulnerabilidades em dispositivos *IoT* é fundamental para garantir a integridade e privacidade dos sistemas, evitando potenciais ataques cibernéticos e mantendo a confiança nas infraestruturas digitais em evolução.

Ao fornecer essa fundamentação teórica, espera-se que o leitor tenha uma compreensão sólida dos conceitos essenciais relacionados à *IoT* e à *Edge Computing*, bem como à segurança desses dispositivos. Isso preparará o leitor para a leitura do capítulo de metodologia, onde será detalhada a abordagem de pesquisa utilizada para abordar as vulnerabilidades em dispositivos *IoT*.

2.2.1 As principais vulnerabilidades em dispositivos *IoT*

À medida que avançamos para uma era em que a *Edge Computing* se torna a espinha dorsal para processamento de dados descentralizado, é imperativo examinar as vulnerabilidades técnicas que permeiam dispositivos *IoT* na *Edge*.

Com base na imagem acima e também no artigo de (Mukhtar *et al.*, 2023), serão explicadas as principais vulnerabilidades em dispositivos *IoT*, e seguindo o tema deste estudo, será abordado as vulnerabilidades em dispositivos Echo.

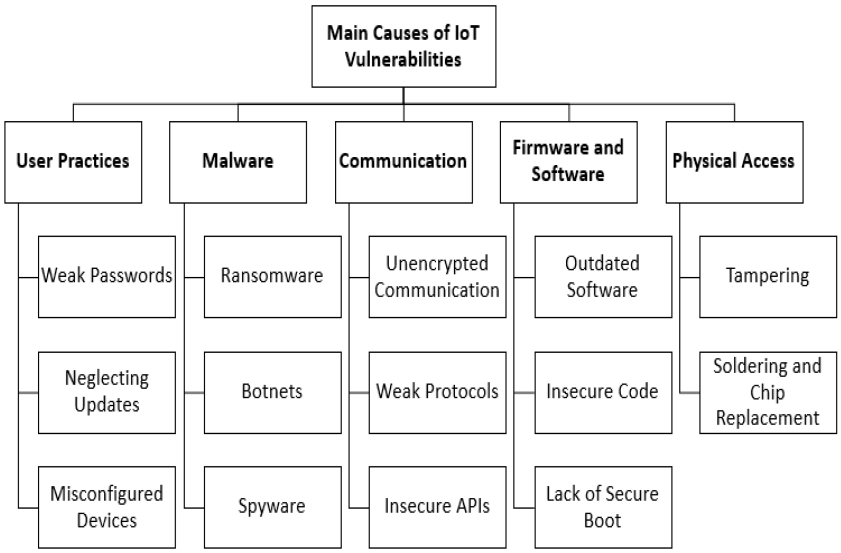

Figura 2.4 – Ilustração das principais causas de vulnerabilidades em dispositivos *IoT*. Fonte: Mukhtar (2023).

2.2.1.1 Práticas do Usuário (*User Practices*)

Vulnerabilidades de segurança muitas vezes têm origem em falhas humanas, seja por desconhecimento, desatenção ou práticas inadequadas. Três das causas mais recorrentes estão relacionadas ao uso de senhas frágeis, à negligência com atualizações de software e à má configuração dos dispositivos.

O uso de senhas fracas continua sendo um dos principais vetores de ataque. Quando usuários optam por senhas simples, previsíveis ou reutilizam credenciais padrão de fábrica (isto é, credenciais que vêm quando o aparelho ou software é inicializado pela primeira vez), aumentam significativamente o risco de comprometimento. A ausência de complexidade e a falta de atualização periódica também contribuem para tornar essas senhas vulneráveis a ataques de força bruta ou "de dicionário". Além disso, o uso de informações pessoais nas senhas, como datas de nascimento ou nomes de familiares, facilita ataques direcionados (*spear phishing*).

Outro fator crítico é a negligência na aplicação de atualizações de segurança. Muitas vulnerabilidades exploradas por atacantes já possuem correções disponibilizadas pelos fabricantes, mas continuam ativas devido à inércia dos usuários. A ausência de atualizações não apenas compromete a segurança, mas também pode afetar a estabilidade do sistema e sua conformidade com políticas de segurança.

Por fim, a má configuração dos dispositivos, especialmente em ambientes IoT ou edge, representa um risco considerável. A manutenção de configurações padrão, a exposição desnecessária de portas e serviços, e a ausência de mecanismos de autenticação robustos ou de criptografia adequada tornam os dispositivos alvos fáceis para exploração automatizada por agentes maliciosos.

2.2.1.2 *Malware*

Malware é uma das maiores ameaças cibernéticas, projetado para comprometer sistemas, roubar dados ou interromper operações. Ele se dissemina por e-mails, downloads e explora falhas de segurança. Entre os danos estão a coleta de dados sensíveis, espionagem e interrupção de serviços. Uma forma grave de *malware* é a *botnet*, onde dispositivos infectados atuam em rede para realizar ataques coordenados.

- ***Ransomware***: Malware que criptografa dados e exige resgate para desbloqueá-los. Frequentemente disseminado por phishing e vulnerabilidades, o ransomware impacta gravemente as operações e a integridade dos dados, além de causar prejuízos financeiros.
- ***Botnets***: Redes de dispositivos comprometidos, controladas por cibercriminosos para realizar ataques coordenados, como *DDoS* e *spam*. A proliferação de dispositivos *IoT* torna a formação de *botnets* uma ameaça crescente.

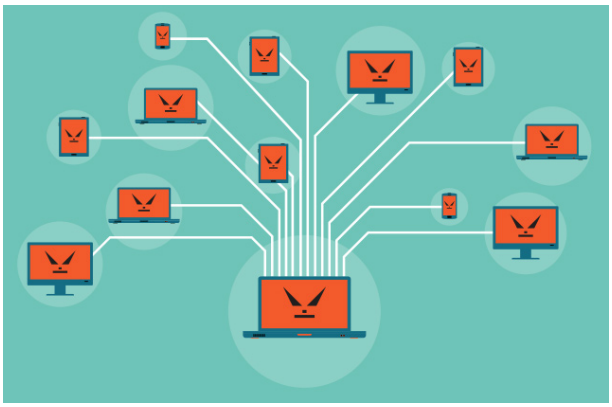

Figura 2.5 – Ilustração de ataque de *Botnets*. Fonte: Fisher (2013).

- ***Spyware***: Coleta informações do usuário sem consentimento. Monitoramento de dados sensíveis e comunicações com servidores remotos são características principais. Pode prejudicar a performance do sistema e ser persistente após reinicializações.

2.2.1.3 Comunicação

Falhas de comunicação, como a transmissão não criptografada, expõem dados à interceptação e ataques *MITM*, *Man-in-the-Middle*[3], comprometendo a integridade e a confidencialidade dos dados.

- **Comunicação não Criptografada (*Unencrypted Communication*)**: A falta de criptografia torna os dados vulneráveis a interceptações e manipulações durante a transmissão.

[3] Ataques MITM, *Man-in-the-Middle* é uma forma de ataque em que os dados trocados entre duas partes são de alguma forma interceptados, registrados e, possivelmente, alterados pelo atacante sem que as vitimas se apercebam.

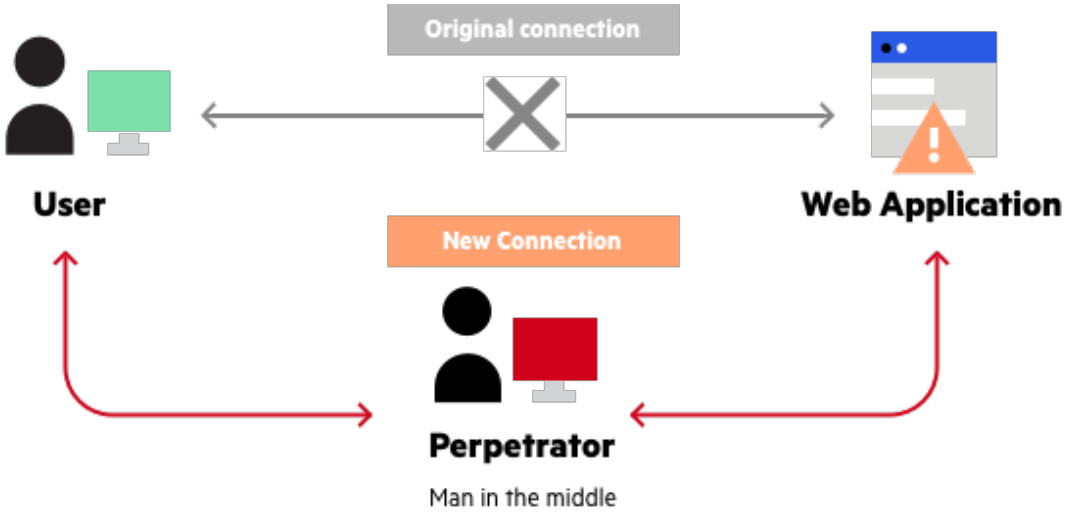

Figura 2.6 – Ilustração de ataque *Man-in-the-Middle*. Fonte: Hasson (2023).

- **Protocolos Fracos**: Protocolos vulneráveis facilitam a interceptação de dados, ataques de *replay* e falhas de autenticação. Protocolos robustos são essenciais para garantir a segurança da comunicação.
- ***APIs* inseguras**: APIs mal protegidas podem ser exploradas por falta de autenticação, transmissão não segura de dados e validação inadequada de entradas, aumentando os riscos de acessos indevidos e ataques de injeção.

2.2.1.4 *Firmware e Software*

- **Software Desatualizado**: Manter o software atualizado é fundamental para evitar falhas de segurança que possam ser exploradas por invasores.
- **Código Inseguro**: A presença de vulnerabilidades no código permite que atacantes comprometam a integridade do sistema.
- **Ausência de Inicialização Segura (*Lack of Secure Boot*)**: Dispositivos que não implementam inicialização segura são mais suscetíveis à execução de código malicioso durante o processo de boot.

2.2.1.5 Acesso Físico

Crackers[4] podem manipular fisicamente dispositivos *IoT*, alterando seus componentes ou realizando substituições de chips, comprometendo a integridade e a segurança do hardware.

[4] Um *cracker* é um criminoso cibernético que usa suas habilidades tecnológicas para violar sistemas de segurança e obter vantagens pessoais. Conceitualmente, *crackers* são diferentes dos *hackers*, que buscam falhas de segurança para ajudar empresas a resolver vulnerabilidades. Coloquialmente, ambos os termos são usados como sinônimos, mas conceitualmente são abordagens diferentes.

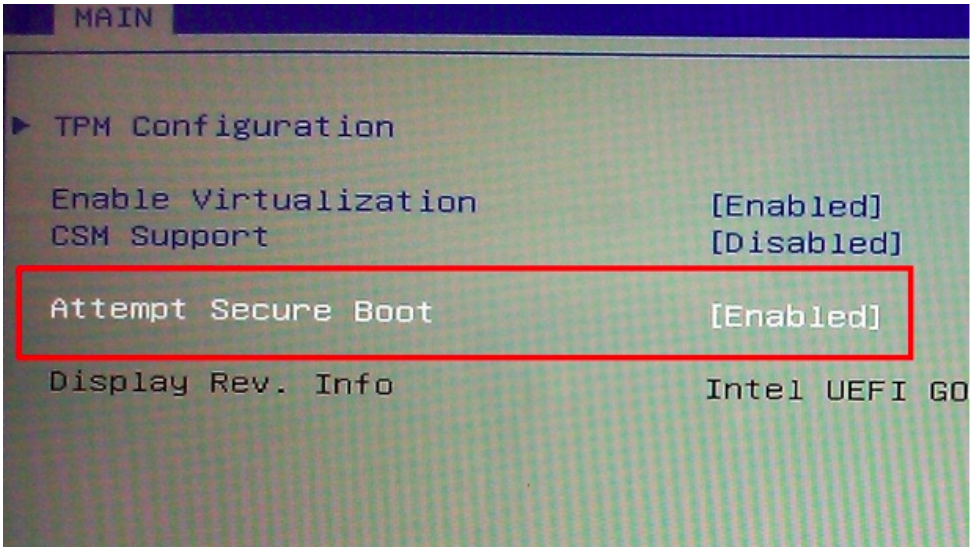

Figura 2.7 – Tela para habilitação do *Secure Boot* em Windows 10. Fonte: Staff (2015).

- **Manipulação Física (*Tampering*)**: Alteração de hardware ou componentes para comprometer o dispositivo.
- **Substituição de Chips e Soldagem (*Soldering and Chip Replacement*)**: Criminosos podem substituir componentes para tomar controle total sobre o dispositivo.

Sendo assim, é possível afirmar que desde a utilização de senhas fracas até práticas inadequadas de atualização, as ameaças abrangem uma ampla gama de áreas, comprometendo a autenticidade, integridade e confidencialidade dos dados. Além disso, a presença de *malware*, falhas na comunicação, vulnerabilidades em *firmware* e *software*, e as ameaças físicas, como a manipulação de hardware, contribuem para um panorama complexo e multifacetado de riscos.

Agora se tratando dos produtos *Echo*, da *Amazon*, há uma vertente mais específica, que é através da captação de voz da Alexa. Será abordado no próximo tópico sobre as vulnerabilidades encontradas no dispositivo *Echo*.

2.2.2 Dispositivos *Echo*: uma análise na segurança

Os dispositivos *Echo*, desenvolvidos pela *Amazon*, são uma linha de alto-falantes inteligentes controlados por voz, equipados com a assistente virtual *Alexa*. E a *Alexa* tem diversas funções que faz através de comandos de voz dados pelo usuário, algumas dessas funções foram citadas abaixo com a referência da (Intelbras, 2024).

- **Controlar a casa inteligente:** Os dispositivos Echo, assim como muitos dispositivos *IoT*, conseguem se conectar a diversas coisas da casa como, janelas, cortinas, portas, iluminação, eletrônicos, piscina, sauna.

- **Fazer ligações/mandar mensagem**: Ligações e mensagens de texto que você pode pedir para fazer apenas por comando de voz, seja do seu celular, ou se for por exemplo uma Echo Show, pode-se fazer através do próprio dispositivo.
- **Rastrear pedidos:** É possível pedir para a *Alexa* acompanhar cada etapa do pedido que você fez, até sua entrega. Na *Alexa* é possível ativar a *skill* "Cadê meu pacote", e então você estará habilitado a colocar o código de rastreio do pedido feito.
- **Lista de compras**: Adicione, remova ou consulte itens em listas de compras facilmente por voz e as veja no próprio app da *Alexa*, no celular.
- **Ouvir livros:** *Alexa* lê e-books da sua biblioteca Kindle, permitindo ouvir suas leituras.
- **Organizar agenda:** Crie listas de tarefas, rotinas e configure lembretes e compromissos.
- **Pedir Uber ou iFood:** Solicite transporte ou comida ativando as *skills* de serviços como Uber e iFood.
- **Atualizações:** Receba notícias, previsões do tempo, trânsito e informações personalizadas em tempo real.
- **Escolher música ou filme:** Controle músicas no Spotify ou escolha filmes em serviços de *streaming* com comandos de voz.
- **Fazer faxina:** Comande robôs aspiradores inteligentes para limpar a casa ou voltar à base de carregamento.

Figura 2.8 – Echo Show 5 de 3ª geração. Fonte: Amazon (2023).

Baseado no estudo de (Hernandez *et al.*, 2022), será citado abaixo algumas das principais vulnerabilidades dos dispositivos Echo, de forma geral.

Falsificação de comandos de voz: A *Alexa* pode ser enganada por vozes que soam semelhantes à do usuário autorizado. Atacantes podem usar gravações de voz ou até mesmo simular a voz de alguém para emitir comandos à *Alexa*, realizando ações como desbloquear portas inteligentes, controlar dispositivos da casa ou fazer compras não autorizadas.

Preocupações com privacidade de dados: A *Alexa* grava e armazena interações com o usuário para melhorar a qualidade do serviço. No entanto, há preocupações de que essas gravações possam ser acessadas por terceiros ou usadas de maneira inadequada. Isso expõe o

risco de vazamento de informações sensíveis, como conversas pessoais, hábitos de consumo e dados bancários.

Abuso de habilidades (*skills*): A *Alexa* permite que desenvolvedores de terceiros criem habilidades, que são como aplicativos para a assistente. Habilidades maliciosas ou mal configuradas podem solicitar informações pessoais ou realizar ações não desejadas, como controlar dispositivos conectados de maneira inadequada ou interceptar dados sem o conhecimento do usuário.

Escuta passiva: A *Alexa* está constantemente em modo de espera, "ouvindo"o ambiente para detectar o comando de ativação ("*Alexa*"). Embora esse recurso seja necessário para o funcionamento do dispositivo, ele levanta preocupações de privacidade, pois pode captar conversas não intencionais ou ser ativado acidentalmente, expondo o ambiente a um monitoramento contínuo.

Conexões *IoT* inseguras: A *Alexa* se conecta a vários dispositivos *IoT* (Internet das Coisas), como lâmpadas, fechaduras e câmeras. Se esses dispositivos ou as conexões entre eles não forem protegidos adequadamente, podem se tornar alvos fáceis para hackers, que podem explorar essas vulnerabilidades para acessar redes domésticas ou controlar dispositivos sem permissão.

Falhas de autenticação: Em algumas situações, a *Alexa* não exige autenticação robusta para realizar certas ações, como fazer compras ou ajustar configurações importantes. Isso pode permitir que qualquer pessoa com acesso ao dispositivo emita comandos críticos, potencialmente resultando em uso indevido dos serviços ou compromissos financeiros.

As vulnerabilidades mencionadas acima, mostram que dispositivos *Alexa* enfrentam riscos significativos, desde falsificação de comandos de voz até escuta passiva. Além disso, conexões *IoT* inseguras e falhas de autenticação podem expor os dispositivos a ataques, permitindo controle não autorizado e acesso a informações sensíveis.

3 Desenvolvimento

Este capítulo descreve o desenvolvimento prático do estudo, que consiste na criação e teste de uma skill personalizada para dispositivos Amazon Echo, com foco na autenticação segura do usuário. A metodologia adotada é exploratória e analítica, e utiliza ferramentas da Amazon, como o *Alexa Skills Kit (ASK)*, *AWS Lambda, CloudWatch* e *DynamoDB*. O experimento visa demonstrar como uma *skill*, inicialmente inativa, pode ser modificada para capturar dados sensíveis do usuário e, posteriormente, utilizá-los para autenticação por SMS.

O experimento foi testado inicialmente no dispositivo Echo Show 5. Como alternativa para validação e comparação dos resultados, o ambiente de desenvolvimento do *Amazon Developer Consol*e, juntamente com o *Alexa Simulator*, também foi utilizado. Além disso, ferramentas de logging e monitoramento, como o ***AWS CloudWatch***, foram empregadas para acompanhar as interações entre o dispositivo Echo e o backend, verificando se a *skill* capturava corretamente os dados sensíveis fornecidos pelo próprio usuário.

Cabe destacar que este estudo não tem como objetivo apresentar um tutorial de criação de uma *skill*, mas sim demonstrar o desenvolvimento de um sistema funcional de autenticação de usuário por meio da Alexa. Para contextualizar o processo e permitir o entendimento dos resultados obtidos, serão apresentados a seguir os principais passos realizados durante a implementação da *skill* e a análise do comportamento da assistente virtual.

Para aqueles que desejam ir diretamente ao ponto em que o objetivo do estudo é plenamente alcançado, recomenda-se a leitura da Subseção 3.0.2, onde é apresentada a versão final da *skill* implementada.

Como este é um estudo voltado à viabilização da coleta de dados sensíveis, serão descritas a seguir as etapas da simulação, incluindo as diferentes tentativas realizadas até se alcançar uma forma de autenticação que fosse ao mesmo tempo simples e segura.

3.0.1 Criando a *Skill - Amazon Developer*

O estudo prático teve início com a criação de uma conta na plataforma *Amazon Developer*. Após a conclusão do cadastro, foi possível iniciar a criação e configuração da *skill*, de acordo com os propósitos específicos deste trabalho. Por se tratar de um projeto acadêmico, a *skill* foi configurada como local, com a sincronização para outras regiões desabilitada.

Conforme mencionado anteriormente, o *AWS Lambda Console* foi utilizado para realizar a manipulação do *backend* da *skill*. Considerando que o dispositivo Echo Show 5 utilizado nos testes estava inteiramente configurado para o idioma português, todas as interações da *skill* com a Alexa ocorreram nesse mesmo idioma.

Para que a coleta de dados ocorresse, foi necessário que o usuário interagisse verbalmente com a Alexa, utilizando comandos previamente definidos na *skill* dormente. Foram selecionadas cinco frases distintas, cada uma relacionada a um tipo específico de dado sensível: nome, país, rua, cidade e telefone.

As frases utilizadas para ativar a *skill* no momento da interação foram:

- Cadastre meu nome.
- Cadastre meu país.
- Cadastre minha rua.
- Cadastre minha cidade.
- Cadastre meu telefone.

Essas frases foram vinculadas a diferentes tipos de dados sensíveis, de acordo com os recursos disponíveis na plataforma de desenvolvimento da Amazon.

É importante ressaltar que nem todos os tipos de dados puderam ser analisados, pois a *skill* limita-se aos tipos de entrada (*Slot Types*) reconhecidos e disponibilizados pela própria Amazon. Os tipos selecionados foram definidos com base nessa disponibilidade — como, por exemplo, o *AMAZON.NameStreet* — e na sua relevância para fins de identificação e autenticação de usuários, conforme ilustrado na Figura 3.1.

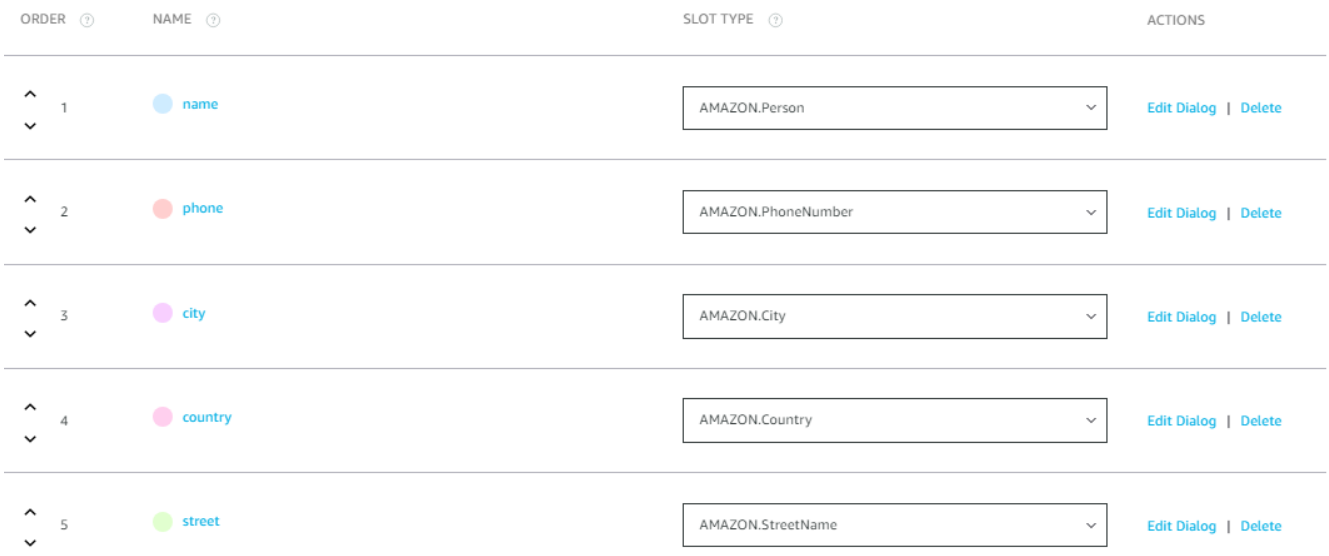

Figura 3.1 – Alexa Skills Kit, criação da *skill*. Captura de tela pessoal (2024).

Cada interação definida na *skill* possui um identificador chamado *name*, associado ao tipo de dado que se deseja coletar. O *name* pode ser interpretado como uma variável, enquanto o *Slot Type* representa o tipo dessa variável, conforme reconhecido pela plataforma da Amazon.

Observação importante: Esta *skill* não foi configurada com o recurso de *Dialog Delegation Strategy*, uma funcionalidade que define como a responsabilidade pela condução do diálogo é distribuída entre a Alexa e a própria *skill*. Essa estratégia permite, por exemplo, que a Alexa assuma temporariamente o controle da conversa ou delegue completamente à *skill*, dependendo do fluxo desejado. Neste estudo, optou-se por não utilizar esse recurso, mantendo a lógica de diálogo mais direta.

Durante a configuração, foi disponibilizada uma opção para que a Alexa confirmasse, ao final da interação, se o usuário realmente desejava validar os dados informados. No entanto, como se trata de uma simulação acadêmica, essa confirmação foi desabilitada. Assim, os dados fornecidos pelo usuário são automaticamente considerados como confirmados e registrados no sistema.

Antes de implementar o código responsável pelas interações, foi necessário configurar a função de backend na *AWS Lambda*, vinculando-a à *skill* criada no Alexa Developer Console. Um dos dados obrigatórios nesse processo é o *ID da skill*, que permite a correta identificação e associação entre os dois componentes.

Como podemos ver abaixo, a *skill* está conectada com a função Lambda, arbitrariamente chamada *"EchoSkill"*.

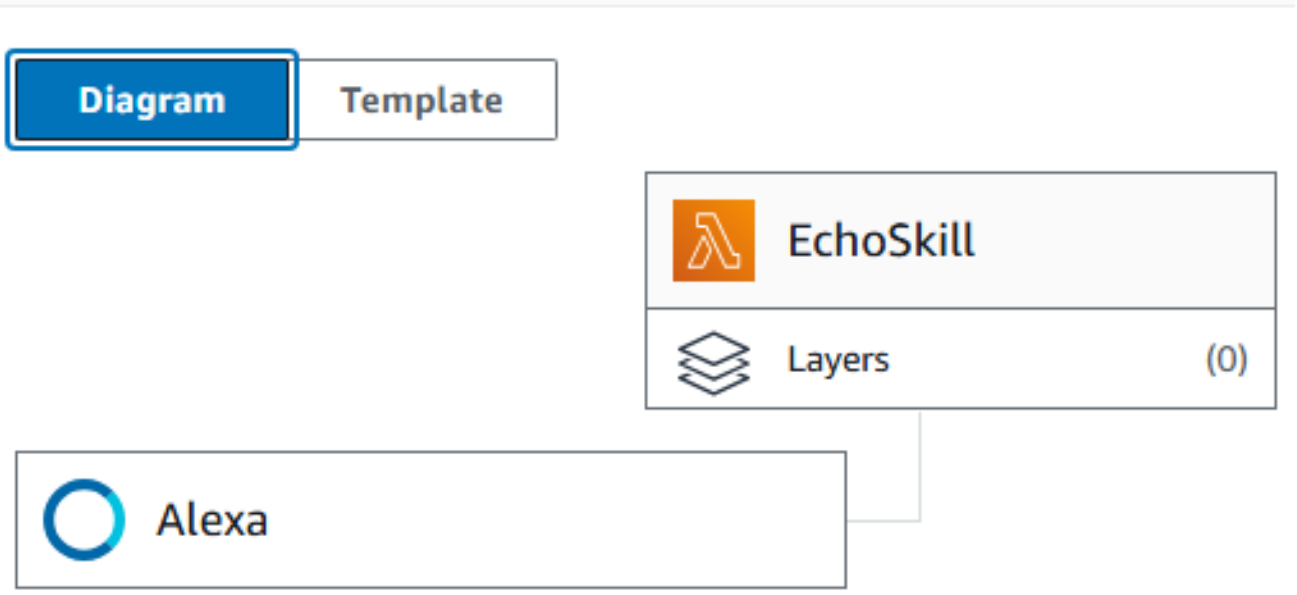

Figura 3.2 – Pequeno diagrama demostrando a conexão da função Lambda com a *skill* criada. Captura de tela pessoal (2024).

Após a conexão feita, o código é desenvolvido em *Python* 3.13. Esse código serve para a coleta dos dados sensíveis logo em que as *intents* são acionadas da *skill*.

3.0.1.1 Código e algumas considerações

Ao abordar o código implementado no backend da *skill*, alguns aspectos cruciais merecem destaque:

- Embora o diálogo com a *Alexa* pudesse ser aprimorado com interações mais complexas, o que exigiria uma *skill* mais elaborada e possivelmente mais custosa, este estudo tem como foco exclusivo a análise da coleta de dados sensíveis. Por isso, cada execução da *skill* contempla apenas uma interação isolada, em que um único campo é fornecido pelo usuário e respondido pela assistente. Dessa forma, não há um fluxo de diálogo contínuo, mas sim a abertura da *skill* e o registro pontual de um dado por vez.

- Para cada campo preenchido pelo usuário, a *Alexa* retorna o valor informado. Caso o campo não seja preenchido ou a informação não seja compreendida, o retorno padrão é "não informado".

- Embora o diálogo com a *Alexa* possa ser aprimorado por meio de uma *skill* mais robusta, o que implicaria maior complexidade e custo, este estudo tem como foco a análise da coleta de dados sensíveis e do comportamento da assistente frente a essas interações. Por esse motivo, optou-se por um modelo baseado em interações unitárias: cada execução da *skill* corresponde à abertura do aplicativo no dispositivo e ao fornecimento de um único dado, resultando em uma resposta direta da *Alexa*, sem continuidade dialogal.

- Todas as respostas obtidas no *Alexa Simulator* foram idênticas às geradas pelo dispositivo *Echo Show* 5. Ainda assim, foi realizada uma comparação inicial entre ambos com o intuito de verificar possíveis variações de comportamento. Como não foram identificadas diferenças, o estudo comportamental foi conduzido com base nas interações realizadas no **Echo Show 5**, utilizando o *Alexa Simulator* como ferramenta complementar. Dessa forma, qualquer falha observada em um dos ambientes também se reproduz no outro, sem divergências de resultado.

- A cada modificação realizada no código, é necessário executar um novo *deploy* pela plataforma *AWS Lambda*, o que envia automaticamente o código atualizado para a *Alexa*, seguido de um *Build* da *skill* para que as alterações sejam aplicadas. Somente após essas etapas é possível realizar novos testes.

- A cada nova informação fornecida à *Alexa*, os dados coletados na interação anterior são recuperados e exibidos novamente, juntamente com o novo dado inserido, formando uma sequência cumulativa de respostas — como ilustrado na Figura 3.3.

Observação importante: Todos os tipos de dados devem ser definidos em inglês, uma vez que os *Slot Types* reconhecidos pela *Alexa* seguem essa convenção. Na primeira tentativa de

execução, a *Alexa* não reconheceu os dados fornecidos porque as variáveis no código estavam nomeadas de forma diferente das reconhecidas oficialmente. Portanto, para evitar inconsistências, é fundamental que os nomes das variáveis coincidam exatamente com os nomes utilizados pela própria Alexa.

O código a seguir demonstra como são tratados os dados fornecidos pelo usuário, bem como os casos em que os campos permanecem nulos — seja por ausência de entrada ou falha de interpretação da Alexa, tanto no dispositivo Echo Show 5 (3ª geração) quanto no *Alexa Simulator*.

```
# Captura os valores dos slots usando os nomes corretos configurados no Alexa Devel
name = slots.get('name', {}).get('value', 'não informado')
phone = slots.get('phone', {}).get('value', 'não informado')
city = slots.get('city', {}).get('value', 'não informada')
country = slots.get('country', {}).get('value', 'não informado')
street = slots.get('street', {}).get('value', 'não informada')

# Monta a resposta da Alexa
speech_output = f"Dados coletados: Nome: {name}, Telefone: {phone}, Cidade: {city},
```

Figura 3.3 – Parte do código apresentando os dados coletados e nulos, se não informados pelo usuário. Captura de tela pessoal (2024).

Após o código ser finalizado, é preciso também inserir o *ARN (Amazon Resource Name)*, que é um identificador único utilizado para referenciar recursos específicos dentro da *Amazon Web Services (AWS)*. Neste caso, ele será utilizado no campo de *Endpoint* da *skill* criada. Ao definir a *Default Region* com o número de identificação do *ARN*, a *skill* é executada novamente por meio do comando *Build Skill*, ficando pronta para a simulação.

3.0.1.2 Alexa - Uma análise comportamental

Não foi mencionado anteriormente, mas o nome da *skill* pode ser diferente do nome utilizado para acioná-la por comando de voz. No caso da *skill* criada neste estudo, seu nome é Datacase, e a chamada utilizada é: "Alexa, abrir dados sensíveis". Para garantir o correto reconhecimento da *skill* pela Alexa, essa frase — "abrir dados sensíveis"— foi incluída juntamente com as demais intents responsáveis pela coleta de dados. Dessa forma, evita-se qualquer ambiguidade quanto ao comando necessário para iniciar a *skill*.

A seguir, será apresentada uma análise aprofundada do comportamento da *Alexa* em relação aos dados fornecidos pelos usuários.

No começo, a *Alexa* não estava coletando os dados quando eram informados, sua resposta sempre era: "Dados coletados: Nome: não informado, Telefone: não informado, Cidade: não informada, País: não informado, Rua: não informada.". A solução foi o que foi falado acima, sobre as variáveis estarem com nomes diferentes (sendo as do *Lambda* e da *Alexa*).

O código precisou passar por certas alterações, pois outro problema identificado foi que a *Alexa* só é capaz de captar um dado por vez em cada interação.

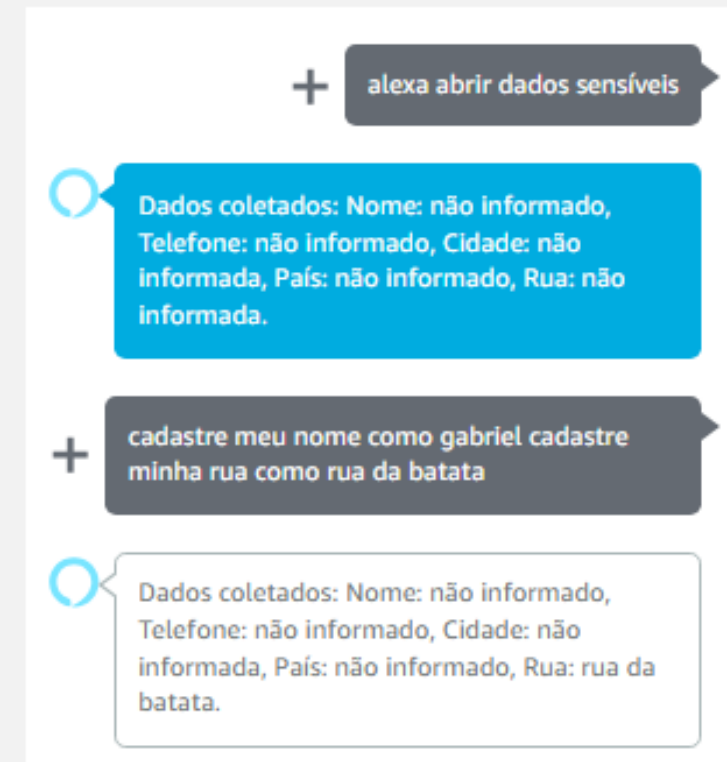

Figura 3.4 – Interação entre usuário e a Alexa, pela *skill* criada. Captura de tela pessoal (2024).

Como podemos observar na Figura 3.4, ao solicitar que a Alexa abra a *skill*, a resposta exibida já apresenta os dados fornecidos anteriormente. Esse comportamento não era o esperado, e acredita-se que tenha ocorrido porque, na interação anterior, o comando "*Alexa*, sair" não foi utilizado. Assim, a *skill* permaneceu ativa e a última entrada registrada continuou sendo considerada.

Outro ponto mostrado na figura é que apenas um dado do cadastro é processado por vez. Quando dois dados são informados na mesma interação, a Alexa armazena apenas o último. Isso pode estar relacionado à forma como a assistente interpreta pausas na fala: ao identificar uma pausa, ela entende que a interação foi concluída e responde com base no que foi compreendido até aquele momento, o que pode resultar em perda parcial ou total das informações fornecidas.

Ao interagir com qualquer *skill*, frases mais longas são processadas integralmente apenas se não houver interrupções perceptíveis. Caso haja uma pausa durante a fala, a Alexa tende a considerar que o comando foi finalizado, processando apenas a parte anterior à pausa. Esse comportamento limita a inserção de múltiplos dados em uma única interação e compromete a experiência do usuário.

Diante disso, considerando que a Alexa registra apenas um dado por vez, o próximo passo foi ajustar a *skill* para permitir uma coleta de dados mais eficaz, respeitando essa limitação.

Para atingir esse objetivo, o código foi ajustado com foco em três metas principais:

- Evitar que a *Alexa* repita todas as frases sempre que um dado for coletado. **Nível: Fácil**.
- Permitir que todos os dados sejam cadastrados em uma única interação com a *Alexa*. **Nível: Médio**.
- Garantir que a *Alexa* armazene corretamente os dados coletados, sem perdas. **Nível: Difícil**.

Para isso, foram definidas novas *Sample Utterances*, com o objetivo de permitir que a Alexa aceite múltiplos dados em uma única interação e retorne apenas aqueles que foram efetivamente cadastrados.

Para isso foram colocados novas *Sample Utterances*, que façam com que a *Alexa* aceite vários dados de uma vez e repita somente as que foram cadastradas. Essas são as *sample utterances*:

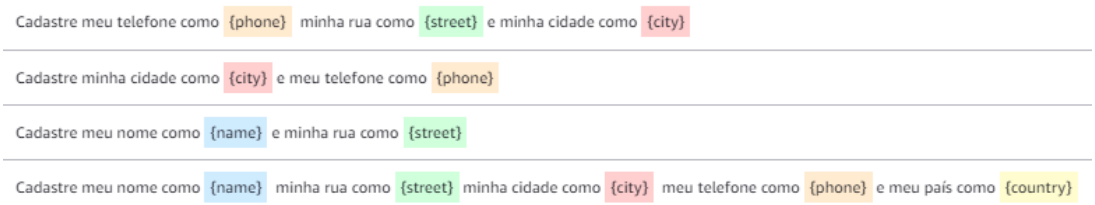

Figura 3.5 – Novas Sample Utterances criadas. Captura de tela pessoal (2024).

É interessante destacar que desta forma, a *Alexa* capta vários dados numa única interação, independente da ordem, conforme é mostrado na próxima figura.

Infelizmente, ao informar um novo dado — como o país, por exemplo — a *Alexa* já não reconhecia mais os dados fornecidos anteriormente, indicando que eles não estavam sendo armazenados entre as interações.

Para evitar esse problema de perda de informações, foi necessário implementar uma forma de armazenamento persistente. Assim, mesmo que o usuário não forneça todos os dados em uma única interação, a *Alexa* seria capaz de registrar cada entrada individualmente, mantendo o histórico até a conclusão do cadastro e a autenticação do usuário.

3.0.2 Banco de Dados Dynamo DB, e ajustes na *Alexa*

Como forma de autenticação, foi adotada a utilização do serviço ***SNS*** (Simple Notification Service), da *AWS*, configurado para o envio de mensagens SMS aos usuários. Através dele foi feita a configuração para que o SMS seja enviado com sucesso ao usuário.

É importante enfatizar que, para que o envio de mensagens funcione com números arbitrários, é necessário solicitar uma autorização à *AWS* e configurar o serviço em modo de produção. No contexto deste estudo científico/acadêmico, foi utilizado o número pessoal do autor, previamente autorizado, para a realização dos testes. Ressalta-se ainda que há um limite gratuito

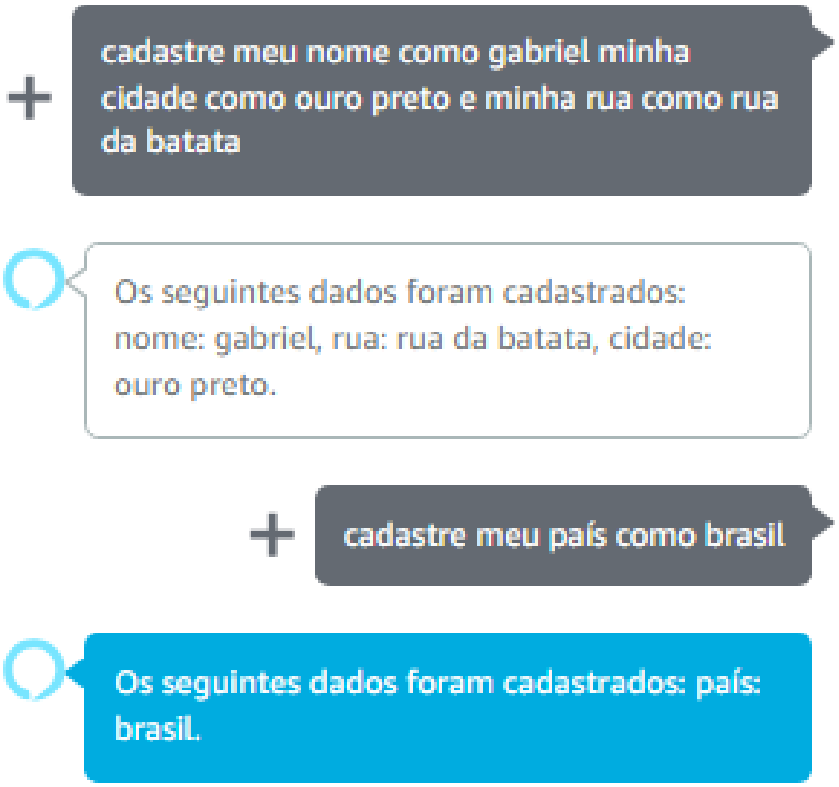

Figura 3.6 – Interação entre usuário e a *Alexa*, pela *skill* criada. Captura de tela pessoal (2024).

```
# Atualizar dados no DynamoDB
updates = []
try:
    update_expression = "SET "
    expression_values = {}
    expression_names = {}  # Para lidar com palavras reservadas no DynamoDB

    if name:
        updates.append(f"nome: {name}")
        update_expression += "#name = :name, "
        expression_names["#name"] = "name"
        expression_values[":name"] = name
    if street:
        updates.append(f"rua: {street}")
        update_expression += "street = :street, "
        expression_values[":street"] = street
```

Figura 3.7 – Parte do código sobre os updates dos dados. Captura de tela pessoal (2024).

de envio de mensagens via SMS, estipulado em até 1 dólar por mês, o qual foi respeitado durante toda a execução do experimento.

Adicionalmente, a região da AWS foi alterada para São Paulo (**sa-east-1**), com o objetivo de evitar atrasos ou falhas no envio das mensagens, utilizando servidores mais próximos geograficamente e, portanto, potencialmente mais estáveis.

Para armazenar os dados, foi utilizado o *DynamoDB*, que é um banco de dados NoSQL[1], disponibilizado pela própria *AWS*. Foi criada uma tabela denominada *UserData*, cuja *Partition Key* foi definida como *userID*. No código, foi estabelecida uma conexão direta com o banco

[1] NoSQL (Not Only SQL) é um modelo de banco de dados que utiliza modelos flexíveis de armazenamento, como chave-valor, documentos, colunas ou grafos, oferecendo escalabilidade e desempenho em aplicações que lidam com grandes volumes de dados ou dados não estruturados.

e com essa tabela, permitindo que os dados fornecidos pelo usuário fossem automaticamente coletados e armazenados a cada novo cadastro.

Além da própria conexão do código Lambda com a tabela, há também a atualização da mesma caso necessário, podendo atualizar diretamente no *DynamoDB*, conforme mostra a Figura 3.7

O teste foi realizado utilizando apenas os seguintes dados: nome, cidade e número de telefone, conforme ilustrado na figura a seguir.

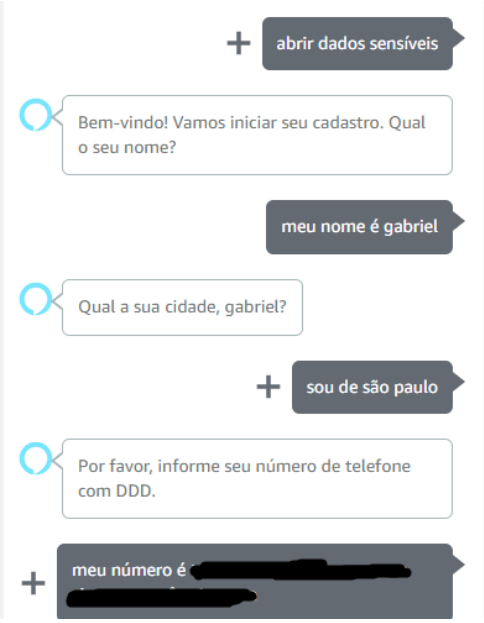

Figura 3.8 – Imagem que mostra os dados que são coletados do usuário. Captura de tela pessoal (2025).

Como a *Alexa* não consegue interpretar o caractere "+"(referente ao DDD do país), optou-se por solicitar ao usuário apenas o DDD regional seguido do número de telefone. Dessa forma, o prefixo "+55"foi adicionado manualmente no backend, sendo armazenado junto ao número informado e permitindo o envio do SMS sem erros.

```
def format_phone_number(number):
    if not number.startswith("+"):
        number = "+55" + number
    return number
```

Figura 3.9 – Código colocando o +55 pré-definido para ser antecessor ao número fornecido. Captura de tela pessoal (2025).

Também foi implementado um *hash* alfanumérico de 6 dígitos, gerado automaticamente após o cadastro, com o objetivo de garantir a unicidade do usuário no sistema. Assim, além

```
# Pré-computar os hashes SHA-256 para cada palavra
HASH_MAP = {word: hashlib.sha256(word.encode()).hexdigest() for word in WORDS_LIST}
```

Figura 3.10 – Função que gera o hash. Captura de tela pessoal (2025)

userID ...	city	ID Hash	name	number	password
user1	são paulo	ODoVMj	gabriel	[redacted]	866bc378ba438b210f6d3...

Figura 3.11 – Apresentação de como ficam os dados de cada usuário cadastrado no banco. Captura de tela pessoal (2025)

do *user ID*, que já é único, o usuário passa a ter um identificador adicional, referido como seu "localizador".

Como os dados já estão sendo armazenados automaticamente no banco, uma medida adicional de segurança foi implementada ao final do cadastro: uma palavra-chave é enviada automaticamente ao celular do usuário **via SMS**. Em seguida, o sistema gera um *hash* dessa palavra utilizando o algoritmo *SHA-256*[2], que é então armazenado juntamente com os demais dados fornecidos.

Essa abordagem garante que, mesmo que o *hash* seja acessado indevidamente, não seja possível descobrir a palavra original a partir dele, assegurando a confidencialidade da informação e permitindo a autenticação bem-sucedida do usuário.

O ***Hash Map*** é uma estrutura que associa cada palavra ao seu respectivo *hash* gerado com o algoritmo *SHA-256*, permitindo que o sistema funcione de maneira segura e eficiente, sem a necessidade de manipular senhas em texto claro. Essa abordagem é fundamental para a autenticação baseada em *hashes*.

É importante destacar que este sistema não tem como foco o modo de uso pelo usuário final, mas sim demonstrar que o indivíduo que realiza o cadastro é, de fato, o próprio usuário que será autenticado / cadastrado no sistema.

As palavras não são armazenadas ou comparadas em texto claro. Durante o cadastro, o *hash* da senha gerada é armazenado no banco de dados. Na autenticação, o *hash* da senha fornecida pelo usuário é **comparado com o *hash* armazenado**. Isso garante que, mesmo que o banco de dados seja comprometido, as senhas dos usuários permaneçam seguras.

No banco, os dados ficarão desta forma, como mostrado na figura Figura 3.11.

Podemos observar que o *hash* é extremamente longo, sendo uma ótima autenticação feita pelo usuário. Após a autenticação feita com sucesso, a *Alexa* gera a resposta, como mostrada figura Figura 3.12.

É importante destacar que essa autenticação está relacionada ao processo de cadastro do

[2] SHA-256 é um algoritmo de *hash* criptográfico da família SHA-2, que gera uma saída de 256 bits a partir de uma entrada, garantindo integridade e segurança de dados.

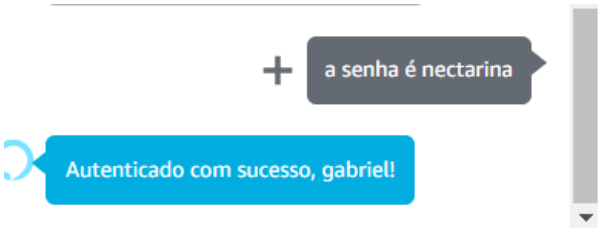

Figura 3.12 – Autenticação da *Alexa*. Captura de tela pessoal (2025)

usuário no sistema, e não à segurança do uso contínuo da aplicação. Dessa forma, o objetivo proposto foi alcançado com êxito, demonstrando a viabilidade da autenticação do usuário por meio da *Alexa*. Ressalta-se que o procedimento é compatível tanto com dispositivos da linha Echo quanto com o simulador disponível no *Alexa Console*.

4 Considerações Finais

4.1 Conclusão

Neste trabalho, foi conduzido um estudo voltado à segurança de dispositivos *IoT* no contexto da *Edge Computing*, com ênfase nos dispositivos Echo e suas interações por meio do *Alexa Skills Kit (ASK)*. A metodologia adotada envolveu a criação de uma *skill* personalizada e a manipulação de *intents*[1], com o objetivo de analisar o comportamento da assistente virtual na coleta e tratamento de dados sensíveis fornecidos pelos usuários.

Este estudo buscou explorar os desafios de segurança em dispositivos *IoT*, com foco prático no *Amazon Echo (Alexa)*, analisando como a autenticação por voz pode ser implementada de forma mais segura. A pesquisa partiu da premissa de que, embora assistentes virtuais como a Alexa facilitem interações cotidianas, sua arquitetura apresenta vulnerabilidades — especialmente no tratamento de dados sensíveis, como senhas. A solução proposta, baseada na utilização de *hashes SHA-256*, mostrou-se eficaz para proteger informações, substituindo o armazenamento de senhas em texto claro por representações criptográficas únicas. No entanto, o processo evidenciou limitações inerentes ao dispositivo, como a dificuldade da Alexa em reconhecer palavras ou números com precisão, sobretudo em ambientes ruidosos ou diante de pronúncias similares, o que pode comprometer tanto a experiência do usuário quanto a confiabilidade do sistema.

Um dos pontos críticos observados foi a capacidade da Alexa em lidar com os dados fornecidos pelo usuário. Por se tratar de um dispositivo com poder de processamento inferior ao de modelos avançados de IA (Inteligência Artificial), especialmente aqueles baseados em *machine learning*, a Alexa enfrenta dificuldades na interpretação de contextos mais complexos ou na adaptação a variações na fala. Essa limitação impacta diretamente a viabilidade de implementar métodos de autenticação mais sofisticados, como perguntas dinâmicas ou análises comportamentais, frequentemente utilizados em sistemas de segurança de alto nível. Ainda assim, o estudo demonstrou que, mesmo diante dessas restrições técnicas, é possível desenvolver mecanismos seguros, desde que as interações sejam planejadas cuidadosamente e a estrutura de dados esteja bem definida.

A comparação com outras inteligências artificiais disponíveis no mercado reforçou a importância de compreender as limitações da Alexa enquanto dispositivo *IoT*. Enquanto sistemas como o *ChatGPT* ou assistentes integrados a plataformas empresariais operam com recursos computacionais robustos e algoritmos sofisticados, a Alexa é projetada com foco na praticidade e no baixo custo, o que a torna menos eficiente em tarefas que exigem análises profundas ou respostas altamente contextualizadas. Essa diferença evidencia a necessidade de ajustar

[1] *Intent* ou intenção é uma representação da ação que atende a uma solicitação falada do usuário para a *Alexa*.

expectativas ao desenvolver soluções para dispositivos amplamente utilizados, buscando um equilíbrio entre segurança e usabilidade sem exceder as capacidades técnicas da plataforma.

A pesquisa também evidenciou a importância de estudos contínuos na área de segurança aplicada a dispositivos *IoT*. Apesar de suas limitações, a Alexa representa um campo promissor para a exploração de inovações, como a integração de autenticação multifatorial ou o uso combinado de biometria vocal e tokens físicos. Além disso, melhorias no tratamento de dados, como algoritmos de reconhecimento de fala mais robustos ou a adoção de criptografia de ponta a ponta, poderiam mitigar riscos sem comprometer a experiência do usuário. O avanço em direção a dispositivos *IoT* mais seguros depende não apenas da evolução tecnológica, mas também de uma compreensão aprofundada de suas vulnerabilidades atuais.

Por fim, este trabalho reforça que, mesmo em dispositivos com capacidades limitadas, a segurança não deve ser negligenciada. A Alexa, inserida em um ecossistema *IoT* em constante expansão, representa uma oportunidade concreta para testar e aprimorar soluções acessíveis, passíveis de adaptação a diferentes contextos. A jornada aqui iniciada abre espaço para pesquisas futuras, como a integração com tecnologias de *blockchain* para auditoria de dados ou o uso de redes neurais leves, voltadas a dispositivos com baixo poder computacional. Dessa forma, cada avanço, ainda que incremental, contribui para a construção de um futuro em que conveniência e segurança coexistam de maneira equilibrada.

Referências

AL-MASHHADANI, M.; SHUJAA, M. Iot security using aes encryption technology based esp32 platform. **The International Arab Journal of Information Technology (IAJIT)**, v. 19, n. 02, p. 214–223, 2022.

ALRAWAIS, A. *et al.* Fog computing for the internet of things: Security and privacy issues. **IEEE Internet Computing**, IEEE, v. 21, n. 2, p. 1, 2, 2017.

CARVALHO, M. de. **Construindo o saber: técnicas de metodologia científica**. [S.l.]: Papirus Editora, 1989. ISBN 9788530800710.

EKANAYAKE, B. N.; HALGAMUGE, M. N.; SYED, A. Security and privacy issues of fog computing for the internet of things (iot). **Cognitive Computing for Big Data Systems Over IoT: Frameworks, Tools and Applications**, Springer, p. 139–174, 2018.

FELIX, M. **Desafios e vantagens da computação em nuvem**. 2023. Disponível em: <https://www.dio.me/articles/desafios-e-vantagens-da-computacao-em-nuvem>.

HERNANDEZ, K. *et al.* Emerging ai and cyber-physical technologies for industry 5.0: Early trends and open challenges. **Engineering Reports**, Wiley, v. 4, n. 2, p. e12592, 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667295222000393>.

INTELBRAS, B. **10 funções da Alexa que você precisa conhecer**. 2024. Disponível em: <https://blog.intelbras.com.br/funcoes-da-alexa/>.

LEA, P. **IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security**. [S.l.]: Packt Publishing Ltd, 2020. 311, 312 p.

MUKHTAR, B. I. *et al.* Iot vulnerabilities and attacks: Silex malware case study. **Symmetry**, MDPI, v. 15, n. 11, p. 1978, 2023.

PROJETOS, F. F. de Estudos e. Revista de inovação. **Revista de Inovação**, FINEP, Rio de Janeiro, v. 18, p. 5–8, 2024. Disponível em: <http://www.finep.gov.br/images/revista/revista18/index.html>. Acesso em: 2 set. 2024.

RAMPAZZO, L. **Metodologia científica**. [S.l.]: Edições Loyola, 2005. ISBN 9788515024988.

SOARES, S. C. M. Arquitetura de detecção de intrusão por anomalias com federated learning em redes iot. 2022.

SU, D. *et al.* "are you home alone? yes"disclosing security and privacy vulnerabilities in alexa skills. **arXiv preprint arXiv:2010.10788**, 2020.