UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MESTRADO ACADÊMICO EM COMPUTAÇÃO

ROGERIO LOPES VIEIRA CESAR

UMA ARQUITETURA DE CONTROLE DE ACESSO PARA INTERNET DAS COISAS

QUIXADÁ

2022

ROGERIO LOPES VIEIRA CESAR

UMA ARQUITETURA DE CONTROLE DE ACESSO PARA INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Marcio Espíndola Freire Maia

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C416a Cesar, Rogerio Lopes Vieira.
Uma arquitetura de controle de acesso para internet das coisas / Rogerio Lopes Vieira Cesar. – 2022.
76 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Quixadá, Programa de Pós-Graduação em Computação, Quixadá, 2022.
Orientação: Prof. Dr. Marcio Espíndola Freire Maia.

1. Internet das coisas - Controle de Acesso. 2. Internet das Coisas. 3. Confiabilidade. 4. Disponibilidade. I. Título.

CDD 005

ROGERIO LOPES VIEIRA CESAR

UMA ARQUITETURA DE CONTROLE DE ACESSO PARA INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Computação do Programa de Pós-Graduação em Computação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em computação. Área de Concentração: Ciência da Computação

Aprovada em: ___/___/_____

BANCA EXAMINADORA

Prof. Dr. Marcio Espíndola Freire Maia (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Arthur de Castro Callado
Universidade Federal do Ceará (UFC)

Prof. Dr. Windson Viana de Carvalho
Universidade Federal do Ceará (UFC)

Profa. Dra. Rossana Maria de Castro Andrade
Universidade Federal do Ceará (UFC)

Dedico esta dissertação ao Dr. Laurivan da Silva Diniz (in memorian), meu primeiro mentor em pesquisa e referência profissional, e às vítimas da COVID-19.

AGRADECIMENTOS

Agradeço, inicialmente, a todos que me proporcionaram percorrer essa jornada, de forma direta ou indireta.

Em especial, agradeço aos meus pais: Manoel Ubirajara Sarmento Cesar e Maria Lopes Vieira Cesar, por me apoiarem em minhas escolhas. Sempre serão minhas inspirações.

Agradeço aos meus colegas de curso: Gilberto, Raehl, José Neto, Warlles, Niltemberg, Leonardo, Caetano, Junior pelo companheirismo, compartilhando os dias de luta e de glória desta trajetória.

Obrigado ao amigo, Luis Ilderlandio (Lucas), pelas discussões sobre o trabalho e escuta de lamentações. Enfim, obrigado pela amizade.

Agradeço ao meu orientador, Márcio Espíndola Freire Maia pelas discussões e direcionamentos para a pesquisa.

Obrigado à minha esposa, Ariel Barbosa Gonçalves por me aceitar e me entender nesse caminho. Compartilhar a vida com você é mais leve!

Gratidão, da forma mais sincera.

“A dúvida é o preço da pureza, e é inútil ter certeza’

(Humberto Guessiguer)

RESUMO

A Internet das Coisas é uma abordagem que considera a comunicação entre diversos dispositivos, serviços, software, objetos e "coisas", onde há compartilhamento e troca de informação. A ideia do paradigma de IoT se mostra multidisciplinar e heterogênea, o que torna desafiador garantir a confiabilidade em ambientes que utilizam tal tecnologia. Dessa forma, estudar como o controle de acesso se comporta em relação à disponibilidade torna-se fundamental para o fornecimento de serviços eficientes e confiáveis por meio do provimento de recursos, quando necessário, pelo tempo necessário e para quem de direito. Neste trabalho, propomos uma abordagem que busca melhorar a disponibilidade em ambientes IoT, por meio do uso de técnicas de controle de acesso e utilizando uma arquitetura descentralizada baseada em edge computing. Assim, o objetivo é regionalizar e otimizar a troca de dados usando uma linguagem de consulta baseada em GraphQL para reduzir a busca excessiva de dados. Apresentamos uma análise de desempenho na qual comparamos o uso de recursos computacionais (CPU e Memória), latência, throughput e taxa de erros em solicitações a recursos, e os resultados apontam para uma melhoria na disponibilidade quando é utilizado mecanismo de acesso em comparação a não utilização na maioria dos quadros investigados, sendo o cenário que adota 100 políticas de acesso o que obteve melhores efeitos, em relação aos cenários que utilizam 10, 500 ou 1000 políticas.

Palavras-chave: Controle de Acesso; *Internet of Things*; Disponibilidade; Análise de Desempenho.

ABSTRACT

The Internet of Things is an approach that considers communication between various devices, services, software, objects, and "things"where information is shared and exchanged. The idea of the IoT paradigm is multidisciplinary and heterogeneous, which makes it challenging to ensure reliability in environments that use such technology. Thus, studying how access control behaves in relation to availability becomes critical to providing efficient and reliable services by providing resources when needed, for the time needed, and to the right people. In this paper, we propose an approach that seeks to improve availability in IoT environments through the use of access control techniques and using a decentralized edge computing-based architecture. Thus, the goal is to regionalize and optimize data exchange using a GraphQL-based query language to reduce excessive data searching. We present a performance analysis in which we compare computational resource usage (CPU and Memory), latency, throughput and error rate in resource requests, and the results point to an improvement in availability when access mechanism is used compared to not using it in most of the investigated frameworks, being the scenario that adopts 100 access policies the one that obtained better effects, compared to the scenarios that use 10, 500 or 1000 policies.

Keywords: Access Control; Internet of Things; Availability; Performance Analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IoT	*Internet Of Things*
QoS	*Quality Of Service* /
HVAC	*Heating, Ventilation e Air - Conditioning* / distância lateral
TCP	*Transmission Control Protoco*
UDP	*User Datagram Protocol* / distância lateral
LAN	*Local Area Network* / falha de blindagem
CC	*Common Criteria*
PAP	Ponto de Acesso da Política
PEP	Ponto de Execução da Política
PDP	Ponto de Decisão Política
PIP	Ponto de Informações da Política
ACL	*Access Control List*
ABAC	*Attribute-based Access Control*
RBAC	*Role-based Access Control* / falha de blindagem
CapBAC	*Capability-based Access Control*
UCON	*Usage Access Control* /
OrBAC	*Organization-based Access Control*
TBAC	*Trust-Based Access Control*
IBAC	*Identity-Based Access Control*
SDN	*Software-Defined Networking* / distância lateral
ARM	Ar*Achitectural Reference Mode*
XACML	*eXtensible Access Control Markup Language* / distância lateral
ECC	*Elliptic-curve cryptography*
API	*Application Programming Interface*
UD	*User Device* / falha de blindagem
TRR	*Things Registration Repository*
CMS	*Central Management System*
ADAC	*Atribute-Based Distributed Access Control*
DAC	*Discretionary Access Control*
AS	Atributo de Sujeito

AO	Atributos de Objeto
AR	Atributos de Recurso
AC	*Access Control*
MQTT	*Message Queuing Telemetry Transport* / distância lateral
REST	*Representational State Transfer* / distância lateral
JWT	*JSON web Token*
HTTP	*Hypertext Transfer Protocol* / falha de blindagem

SUMÁRIO

1 INTRODUÇÃO

A Internet das Coisas (*Internet Of Things* (IoT)) tem como características fundamentais a conexão e comunicação entre diferentes dispositivos computacionais, que variam desde pequenos sensores até máquinas mais robustas, em termos de recursos de processamento e memória, como um *smartphone* ou dispositivo vestível. Nela, estima-se que até 2025 o número de dispositivos IoT conectados no mundo cresça para 21,5 bilhões (IoT Analytics Research, 2018), a impactar na economia global com a movimentação de mais de 11 trilhões de dólares.

A construção dessa rede consiste em introduzir uma arquitetura de hardware e software nos objetos do cotidiano, como veículos, casas, edifícios, cidades entre outros itens. Assim, esses objetos passam a atuar como sensores e atuadores, monitorando o ambiente e alterando o seu estado. Se por um lado essas características trazem benéficos para a economia e bem estar de indivíduos e cidades, por outro, são introduzidos desafios em aspectos técnicos e sociais (DESHMUKH; SONAVANE, 2017).

Na Internet das Coisas, parte das aplicações possui como requisito fundamental o processamento de dados em tempo real (LIU *et al.*, 2020b). Isso significa que ela coleta dados utilizando sensores presentes no ambiente, processa e agrega esses dados para gerar informações para as aplicações e possivelmente atuar no ambiente. Por exemplo, (KUMAR *et al.*, 2021) apresentam um estudo de caso para monitorar a qualidade da água em lençóis freáticos e (THIAM *et al.*, 2021) realiza um estudo sobre a confiabilidade em aplicações IoT em agricultura. Em resumo, uma característica comum nesses estudos é a preocupação com a disponibilidade, onde o processamento e a entrega de resultados devem ser efetuados de forma confiável e segura, diante das restrições de tempo e recursos computacionais, respondendo aos eventos com segurança.

Essa é uma preocupação que deve estar presente durante a construção das aplicações IoT, considerando o crescimento do seu uso na sociedade, como citado anteriormente. Assim, estudos que investiguem ou proponham mecanismos e soluções que atendam atributos de confiabilidade são fundamentais para a adesão e implantação deste tipo de tecnologia. Entretanto, observa-se que há uma lacuna entre os modelos teóricos e práticos, dificultando a utilização de conceitos de confiabilidade em aplicações reais (MAITA, 2020).

Uma definição mais formal afirma que a confiabilidade representa a competência do sistema em manter a continuidade do serviço de forma correta (AVIZIENIS *et al.*, 2004). Esse conceito está relacionado à diversos atributos, como disponibilidade, manutenibilidade e integridade. Ela é um requisito relevante às aplicações em IoT, como elemento para medir o

desempenho do serviço, bem como a satisfação das partes interessadas (*stakeholders*) (NUAIMI, 2017). Para (MA *et al.*, 2019), padrões atuais podem não ser suficientes para atender aplicações em ascensão que necessitam de confiabilidade.

Considerando mais especificamente a comunicação entre dispositivos, controlar o acesso é papel fundamental na execução das aplicações com restrições de segurança, já que dispositivos não autorizados ou mal-intencionados podem afetar toda a cadeia de funcionamento. Essa características está diretamente ligada ao conceito de disponibilidade. Por exemplo, uma falha ou acesso não autorizado constitui uma grande ameaça, com o possível vazamento de informações sobre residências, interrupção de equipamentos que realizam a manutenção da vida, ou elementos que controlam uma linha de produção de uma indústria.

O controle de acesso é definido como a metodologia que lida com os direitos de acesso a dados e serviços, por um usuário ou dispositivo autorizado. Seu objetivo é gerenciar privilégios, de forma a limitar, permitir ou negar obtenção de serviços ou informações, garantindo a proteção dos dados (BATE *et al.*, 2017). Considerando a natureza crítica dos sistemas IoT, a confiabilidade de elementos de segurança se torna primordial em qualquer projeto de aplicação.

1.1 Motivação

O modelo tradicional de computação em nuvem possui dificuldades em atender demandas de aplicações que exijam altos requisitos de tempo real e consumo de banda, como em IoT (LAN *et al.*, 2019). Mesmo considerando o grande fluxo de dados nas proximidades dos dispositivos *(edge)*, concentrar os dados em um elemento central *(cloud)* pode ser desnecessário e gerar alta latência (KAWAGUCHI; BANDAI, 2020). Nesse sentido, a utilização de uma arquitetura descentralizada por meio de técnicas que melhorem a disponibilidade pode expor uma melhor confiabilidade.

O fluxo e processamento de informações entre as entidades envolvidas no ecossistema IoT podem estar sob ameaças que diminuem a confiabilidade, e geralmente, estão associadas a dois tipos de problemas: os **técnicos** que estão relacionados a aspectos como condições do ambiente, contexto de implantação, transmissão prejudicada, forma de entrega de mensagens e *Quality Of Service* / (QoS) - *Quality Of Service*; e os **comportamentais**, onde a relação entre a execução efetiva de tarefas e o ambiente no qual está inserido, é causal, com objetivo de diminuir os riscos de que entidades maliciosos possam agir e causar danos. (FILHO, 2019).

Nsse sentido, a diminuição de riscos em serviços que utilizem o ambiente IoT, sem

reconhecer o contexto da aplicação, pode acarretar em soluções ineficientes, aumentando a probabilidade de contratempos. Por esta razão, este trabalho propõe uma infraestrutura para controle de acesso em IoT, cujo objetivo é a melhoria na confiabilidade dos serviços. A proposta baseia-se em computação em borda e tecnologias de encaminhamento de mensagens sobre a linguagem de consultas baseadas em grafos (GraphQL) e MQTT, além de repositório de políticas de acesso baseadas em contexto. Para análise de atributos de confiabilidade, foi realizada uma avaliação de desempenho estruturada seguindo a metodologia proposta por (BUKH, 1992) .

1.2 Objetivos

De acordo com as discussões apresentadas anteriormente, a seguinte questão de pesquisa norteia a presente dissertação de mestrado:

Como o controle de acesso para aplicações em ambiente IoT implementado em *edge computing* e políticas baseadas em contexto pode melhorar a disponibilidade do sistema?

1.2.1 Objetivo Geral

Como objetivo geral, busca-se elaborar um modelo de controle de acesso baseado em *edge computing* que melhore a confiabilidade, por meio do atributo de disponibilidade, das aplicações em ambientes IoT.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral da pesquisa, listamos os seguintes objetivos específicos:

- Levantar requisitos para controle de acesso, no contexto de IoT;
- Elaborar modelo de controle de acesso baseado em contexto a partir dos requisitos para o ambiente IoT;
- Implementar mecanismo de controle de acesso verificando seu comportamento na borda da rede;
- Investigar a performance do modelo através de uma análise de desempenho;

1.3 Metodologia

Os seguintes passos foram seguidos para a realização desta pesquisa:

- **Revisão da Literatura:** Uma revisão de conceitos relacionados à *Internet Of Things*, arquiteturas de referência, bem como o estado da arte sobre o controle de acesso nesse tipo de paradigma foi realizada. Esse estudo foi importante para investigar a respeito dos desafios de pesquisa na área e a relação entre confiabilidade em sistemas IoT, assim como arquiteturas propostas e tecnologias emergentes nesse contexto.
- **Requisitos para Controle de Acesso em IoT:** Considerando as especificidades do paradigma IoT, foram reunidos nessa etapa os requisitos para o desenvolvimento de mecanismos de controle de acesso a serem implementados nesse tipo de ambiente. Para tal, os requisitos definidos por (RAVIDAS *et al.*, 2019b) foram utilizados como base para o estudo.
- **Trabalhos Relacionados**: Com objetivo de comparação e complementação da pesquisa, cinco trabalhos foram escolhidos seguindo os critérios de serem propostas de frameworks de controle de acesso em ambiente IoT. A seleção foi realizada através de pesquisas em bases da *IEEE Xplorer, ACM Digital Library, Springer e ScienceDirect*, combinando as palavras-chave *"Access Control" AND ("IOT" OR "Internet Of Things") AND "Authorization"*.
- **Modelo**: Neste passo, foram modeladas a estrutura das políticas a partir de atributos e expressões contextuais bem como a arquitetura proposta para o mecanismo de controle de acesso. Em seguida, definidas as tecnologias de implementação, seguindo a estrutura da proposta por meio dos diagramas desenvolvidos ainda nessa fase.
- **Implementação**: De posse dos requisitos básicos e do modelo de políticas constituído, a implementação do serviço alvo do estudo foi desenvolvida. O mecanismo atua gerenciando a autorização de acesso ao recurso, podendo utilizar token JWT para autenticação de credenciais de identidade.
- **Avaliação e Análise**: A análise de desempenho foi realizada a partir de um estudo empírico, por meio de prova de conceito que detém os elementos da arquitetura proposta.

1.4 Organização do Trabalho

O trabalho segue organizado por capítulos. O seguinte (2) expõe a revisão de literatura, contemplando: a) o conceito de *Internet Of Things*, suas arquiteturas de referência, estudo de confiabilidade aplicado, arquiteturas de aplicação; b) Controle de acesso, suas arquiteturas, mecanismos e requisitos relacionados. O capítulo 3 trata de trabalhos relacionados e

em conformidade com, pelo menos, um dos requisitos de aplicação de controle de acesso, além de uma comparação destes com o presente estudo. Na seção seguinte, expõe-se o modelo da abordagem proposta, contemplando um caso de uso, a especificação do modelo de políticas adotado, a arquitetura proposta, diagrama de sequência e o algoritmo. Em sequência, o quinto capítulo apresenta a avaliação e análise de experimentos e, por fim, tem-se as considerações finais na última seção.

2 REVISÃO DA LITERATURA

Este capítulo faz uma apresentação sobre os conceitos envolvendo o tema tratado nesta dissertação de mestrado, como Internet das Coisas e arquiteturas de referência, discutindo como *Cloud, Fog e Edge* permitem a criação de uma infraestrutura descentralizada. Depois, alguns requisitos não-funcionais como confiabilidade e disponibilidade são apresentados. Por fim, técnicas, mecanismos e aspectos relevantes para a implementação de uma solução baseada em controle de acesso são discutidos.

2.1 Internet das Coisas

Desde a proposição do termo em 1999 por Kevin Ashton, são várias as definições concebidas para Internet das Coisas. Em um contexto mais atual, (KOME *et al.*, 2018) define como sendo o paradigma que permite a composição de objetos endereçáveis de maneira única, podendo realizar ações de identificação, detecção ou atuação, compartilhando recursos de processamento e cooperando entre si, a fim de se atingir determinado objetivo.

Essa ideia nos remete a considerar que um sistema de IoT não pode ser visto como um conjunto de sistemas individuais e sim como uma infraestrutura crítica e integrada no qual os objetos se comunicam e executam vários serviços e aplicações. Esses serviços são providos de forma personalizada aos usuários finais, sejam indivíduos em seu cotidiano ou cidades inteiras (MAHYOUB *et al.*, 2017). Em sua essência, a IoT é vista como um paradigma multidisciplinar abrangendo várias áreas, como: computação, saúde, transporte, nas mas diversas aplicações.

As aplicações, geralmente têm objetivos de coletar dados do mundo real, que podem inferir atividades de indivíduos ou grupos, os quais podem gerar informações valiosas que melhorem a qualidade das pessoas. A confiabilidade relativa aos dados adquiridos é crucial. Assim, ao selecionar o sensor, objeto inteligente ou arquitetura adequados deve-se levar em consideração os requisitos da aplicação alvo (MAITA, 2020).

Em vários setores da sociedade, os benefícios da utilização de dispositivos se comunicando em rede IoT são percebidos, como em comércios, indústrias, casas, prédios e aplicações em hospitais. Devido às próprias características do paradigma, os requisitos para aplicações em IoT, tal como o controle de acesso, devem respeitar o domínio de cada aplicação. Como segue, são listadas categorias ou domínios de aplicação:

– **Veículos Conectados:** A perspectiva de futuro enquanto IoT envolve também uma infra-

estrutura onde veículos e outros elementos que fazem uso da infraestrutura de trânsito (semáforos, unidades de coletas de dados em rodovias, por exemplo) possam se comunicar e compartilhar dados do referido contexto (RAVIDAS *et al.*, 2019b).

- **Casas Inteligentes:** Consistem em prover recursos em ambientes domésticos e equipamentos inteligentes que facilitem a vida das pessoas. Esses objetos podem ser trancas automáticas, controladores de temperatura e luminosidade, equipamentos inteligentes (smart tv, cafeteira inteligente, fogão inteligente, geladeira inteligente, etc), que podem ser controlados remotamente, usando smartphones ou outros equipamentos que possam auxiliar o controle e supervisão. Esse tipo de tecnologia proporcionou um cresimento na utilização por parte de pessoas idosas ou com alguma deficiencia, podendo controlar esses eletrodometiscos remotamente (VISHWAKARMA *et al.*, 2019).
- **Prédios Inteligentes:** Em aplicações de prédios inteligentes há um direcionamento em aplicações relacionadas a equipamentos inteligentes de *Heating, Ventilation e Air - Conditioning* / distância lateral (*HVAC*) *(Heating, Ventilation e Air Conditioning)*. Além disso, iluminação inteligente e sistemas de segurança e alarme contra incêndios são foco de aplicações em IoT, tendo a intenção de facilitar e melhorar a vida das pessoas , podendo economizar energia (BINDRA *et al.*, 2019).
- **Saúde:** Aplicações que envolvem saúde, no ambiente IoT, são basicamente equipamentos médicos que podem ser controlados remotamente ou aplicados a sensores que supervisionam pacientes (BAGDASARYAN *et al.*, 2019).
- **Indústria:** Na indústria, aplicações que envolvam sistemas de controle de máquinas industriais, robótica, atuadores e sensores que ofereçam suporte a sistemas de tempo real, podem ser vistos como utilização de IoT nesse contexto. Além disso, ferramentas que possam monitorar, por meio de sensores, a saúde de máquinas ou mesmo a gerência da produção (RAVIDAS *et al.*, 2019b).
- **Dispositivos Vestíveis (*wearable)*:** Geralmente são utilizados para monitoramento da saúde de pessoas. Sensores captam informações pessoais de saúde de pacientes e podem enviar para médicos, enfermeiros ou outro profissional da saúde para análise (ARFAOUI *et al.*, 2020).
- **Smart Cities:** As cidades inteligentes objetivam proporcionar um melhor uso dos recursos públicos, através de tecnologias de informação, melhorando a qualidade dos serviços oferecidos aos cidadãos e reduzindo o custo operacional para o público e também a

administração pública (MEDINA *et al.*, 2017).

2.1.1 Arquiteturas de Referência em IoT

As aplicações para IoT demandam requisitos para sistemas cada vez mais complexos diante da gama de possíveis aplicações. Assim sendo, a arquitetura a qual é implantada o sistema influencia diretamente nos objetivos para qual a aplicação foi concebida. Há diferentes propostas de arquiteturas que visam principalmente confiabilidade, qualidade do serviço (QoS) e integridade dos dados (MAITA, 2020).

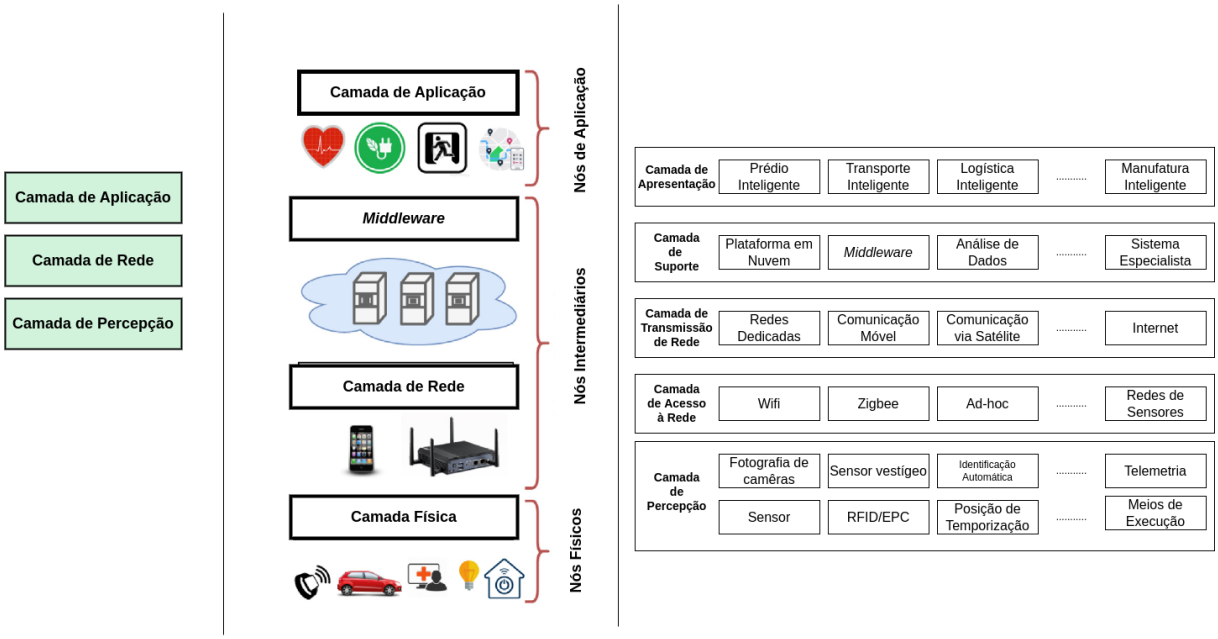
Inicialmente, a arquitetura usada por pesquisadores foi a baseada na indústria de comunicação móvel (MAITA, 2020), utilizando 3 camadas (*perception, network e application*), como exemplo a proposta por (WU *et al.*, 2010).

Para (RAVIDAS *et al.*, 2019b), a separação entre as camadas de *middleware* e comunicação permite um bom entendimento sobre o compartilhamento de recursos entre nós e usuários finais, aumentando a vantagem de abstração sobre os protocolos usado no compartilhamento de mensagens. Em seu estudo, a adoção de arquitetura com 4 (*physical, network, middleware and application*) camadas oportuniza a vantagem citada anteriormente sobre o estudo mecanismos de autorização em IoT.

Autores também propõem a representação de aplicações em IoT apresentando 5 camadas. (KHAN *et al.*, 2012) relata que as aplicações em IoT devem abordar fatores como interoperabilidade, confiabilidade e QoS, sendo que o fato de todos os componentes estão conectados e e todos podem trocar informações, o tráfego e o armazenamento podem aumentar exponencialmente, sendo guiados pelo progresso da tecnologia e o modelo de negócios. Analisando as tecnologias em *gateway* a partir da visão de IoT, (ZHONG *et al.*, 2015) apresenta também o modelo de 5 camadas (*perception, network access,network,application support e presentation*).

Como há uma ampla variedade de domínios de aplicativo para IoT, seus requisitos exigem sistemas cada vez mais complexos. Esta situação afeta diretamente o design das arquiteturas para IoT, produzindo um conjunto de propostas com diferentes camadas e funcionalidades, e um conjunto de diferentes terminologias. Pesquisadores e organizações de padronização têm orientado seus esforços para propor arquiteturas de IoT que visam confiabilidade, confidencialidade, Qualidade de Serviço (QoS) e integridade (MAITA, 2020).

Figura 1 – Exemplos de arquiteturas em camadas em IoT. (a) 3 Camadas, (b) 4 camadas e (c) 5 Camadas.



Fonte: (WU *et al.*, 2010; RAVIDAS *et al.*, 2019b; ZHONG *et al.*, 2015). (adaptados)

2.2 Disponibilidade em IoT

A disponibilidade em IoT versa sobre a prontidão de serviços ou dados, garantindo o acesso imediato ao recursos, pelas partes interessadas. Este conceito está relacionado com a recuperação e confiabilidade do sistema, incluindo os recursos a níveis de software, que indica que o serviços estão sendo providos para quem de direito acessá-lo, e hardware, quando há compatibilidade com as aplicações e protocolos de IoT, fornecidos para benefícios de usuários. Dadas as circunstancias, muitos objetos em IoT estão com conectividade total ou parcial à internet, sendo estes podem ser expostos à falhas e até a invasores (CHANAL; KAKKASAGERI, 2020).

Em IoT, uma definição de disponibilidade é expressa como a proporção em que um dispositivo é afetado por uma falha ou operação anormal, impossibilitando o funcionamento. As estratégias definidas para alcançar esse requisito são implementadas tanto pela perspectiva de hardware, como de software, sendo este último, o grande gargalo (ABBAS *et al.*, 2021). Assim, há a necessidade de que esses dispositivos ofereça o devido serviço em tempo adequado sob a perspectiva abrangente de aplicações.

Dessa forma, como afirma (GUPTA *et al.*, 2021), parâmetros como dependabilidade, confiabilidade e disponibilidade na rede IoT dependem de fatores como equipamentos

de hardware (poder e capacidade computacional), serviços de software (estruturas de dados e protocolos) e fatores humanos (direitos de acesso e usabilidade), atestando que apenas o equipamento funcionando, não se trata de confiabilidade ou disponibilidade, haja visto que, em muitas vezes, acontece a observação do funcionamento normal do equipamento de hardware, mas o serviço de rede pode estar indisponível devido ao congestionamento da rede, erros de software ou falhas de dispositivos.

Quando se trata de comunicação em rede, como é o caso de IoT, de um ponto de vista do usuário, a confiabilidade está relacionada à mínima interrupção, ou seja disponibilidade do serviço e do ponto de vista do provedor do serviço é suportar ou tolerar falhas ou ataques sistemáticos, sem impactar diretamente na experiência do usuário. Para tanto, há múltiplas possibilidade de utilização do termo "confiabilidade", como no caso de robustez da aplicação, resistência a problemas de segurança, autoadaptação e usabilidade a longo prazo (MAITA, 2020).

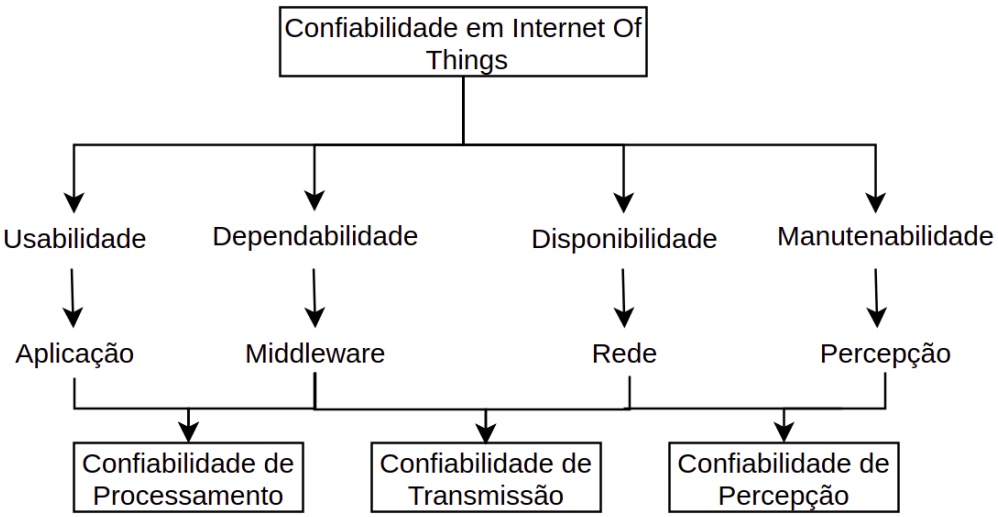
O ambiente complexo em IoT requer uma visão de confiabilidade aplicada em cada camada sob diferentes aspectos, dados os diferentes requisitos das aplicações. Assim sendo, a arquitetura a qual é implantada o sistema influencia diretamente nos objetivos para qual a aplicação foi concebida. Há diferentes propostas de arquiteturas que visam principalmente confiabilidade, qualidade do serviço (QoS) e integridade dos dados (MAITA, 2020).

Considerando uma arquitetura em 3 camadas, (KEMPF *et al.*, 2011) aponta que, na camada link, a confiabilidade requer o uso da pilha *Transmission Control Protoco* (TCP) para transmissão entre aplicativos na internet e deve-se considerar problemas de percas de pacotes por congestionamento. Na camada de transmissão e roteamento, a comunicação deve preferir o protocolo *User Datagram Protocol* / distância lateral (*UDP*) devido a restrições computacionais e de energia, porem é necessário tratar retransmissões. Na camada de aplicação, a indicação do correto funcionamento e estado operacional do aplicativo é importante para confiabilidade do sistema. Um outro elemento importante é o controle de acesso, cujo objetivo é gerenciá-lo de forma a garantir a continuidade e corretude da aplicação.

No estudo apresentado por ((XING, 2020), os desafios relativos ao estudo de confiabilidade em IoT são analisados sob a perspectiva de arquitetura em 4 camadas (percepção, comunicação, suporte e aplicação). O autor relata que a computação em borda ganha popularidade por melhorar o tempo de resposta (latência) em comparação ao processamento centralizado na nuvem. O trabalho condensa estudo sobre confiabilidade em IoT, apontando tópicos relevantes, categorizados pelas camadas: percepção, comunicação, suporte e aplicação.

Um modelo de avaliação de confiabilidade em sistemas IoT pode ser encontrado em (THOMAS; RAD, 2017), que se baseia em métricas fundamentais de qualidade para sistemas de alto desempenho, como no caso de sistemas IoT. Nesse sentido, a figura 2 mostra a relação entre essas métricas e a arquitetura em 4 camadas em sistemas IoT. A **Usabilidade** está diretamente relacionada com à construção de uma interface para uso conveniente e satisfação do usuário. A **Dependabilidade** refere-se ao sistema funcionar de acordo com suas especificações, a relacionar concepção, projeto e objetivos. Um dos elementos que entregaria essa característica é o de redução de latência, que pode comprometer a entrega do serviço. A **Disponibilidade** é o parâmetro quantitativo que associa a probabilidade do sistema estar disponível para o uso em um determinado período. A tomada de decisão entre o dispositivo IoT e o usuário final deve ser feito em tempo prontamente acessível. A **Manutenabilidade** se refere à capacidade do sistema ser facilmente desacoplando, atualizado e consertado, conservando as funcionalidades do sistema sem obstrução.

Figura 2 – Modelo de Avaliação de Confiabilidade em IoT.



Fonte: (THOMAS; RAD, 2017)(Adaptado)

Ainda em análise da figura 2, é possível relacionar o controle dos direitos de acesso ao recurso na rede IoT com a disponibilidade, onde a adoção de mecanismo adequado de controle de acesso pode reter requisições melhorando o processamento computacional e a vazão da rede.

2.3 *Cloud, Fog e Edge Computing*

O conceito de Cloud Computing está consolidado há alguns anos. Sua concepção é dada por tudo que esteja hospedado na Internet, sendo possível alocar recursos, serviços ou dados e que estejam disponibilizados para o usuário, quando solicitado, podendo ainda compor serviços mais robustos (BISWAS; GIAFFREDA, 2014). As principais características apresentadas por (BISWAS; GIAFFREDA, 2014) são: oferta de serviços sob demanda, acesso onipresente, *pool* de recursos e elasticidade.

Essas características podem convergir com o ambiente para aplicações em IoT, possibilitando a arquitetura de computação em Nuvem mais viável em alguns casos. Mas o crescimento da utilização de dispositivos em ambiente IoT proporciona, consequentemente, o risco de aumento da largura da banda para comunicação. Além disso, também expõe riscos de segurança e privacidade (ALWARAFY *et al.*, 2020).

Quando se trata de *Fog Computing*, (OMONIWA *et al.*, 2018) elata que o objetivo principal para introdução da utilização do paradigma é estender serviços (armazenamento, operação em base de dados, integração de dados, gerenciamento de end-devices em IoT e segurança, por exemplo) que até então eram disponibilizados na nuvem até a borda da rede. Nesse sentido, há uma movimentação da inteligência para o nível de *Local Area Network* / falha de blindagem (LAN), em que os dados são processados em um *gateway* IoT.

Todas essas características foram bem vistas tanto no âmbito acadêmico quanto na indústria, em uma perspectiva que não visa substituir a Computação em Nuvem, como está disposta atualmente, mas complementar e melhorar a execução e o processamento de serviços, sendo realizados localmente (OMONIWA *et al.*, 2018). O mesmo autor cita algumas vantagens da abordagem em Fog Computing:

- Elementos distribuídos geograficamente;
- Suporte à redes de sensores em grande escala e nós finais;
- Melhor resposta em relação ao tempo se comparado ao modelo em nuvem;
- Suporte à heterogeneidade e interoperabilidade; e
- Análise online e possível interação com nuvem.

Em um modelo de costume de computação, as aplicações seguem um padrão em que a computação é processada no servidor em nuvem e executada nos dispositivos do usuário. Porém este mecanismo possui algumas desvantagens, como grande acúmulo de tarefas a serem processadas e alto atraso na transmissão de dados. Para tentar solucionar esses problemas,

pode-se fazer o uso da *Edge Computing*, cujo objetivo é executar determinadas tarefas de processamento e armazenamento de dados na borda da rede, possibilitando a execução de tarefas o mais próximo dos usuários finais (XUE *et al.*, 2020).

Além de características como consciência de localização e acesso em tempo real, (XUE *et al.*, 2020) cita outras características chaves encontradas na arquitetura de *Edge Computing*:

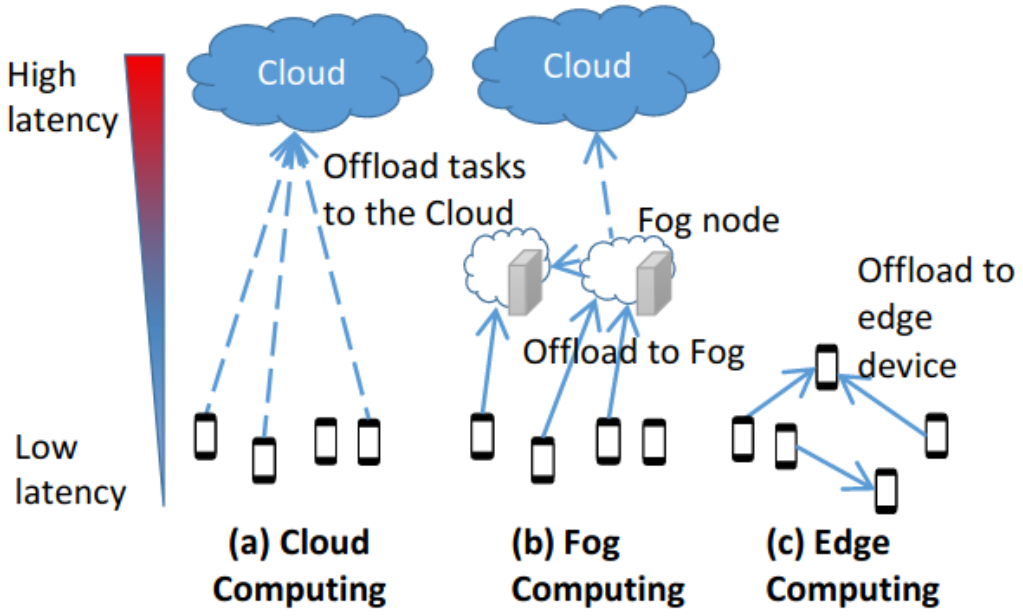
- **Distribuição Intensiva**: Colocando os serviços de computação na borda da rede, análises de *big data* podem ser feitas mais rapidamente e com melhor eficiência, oportunizando análises em tempo real e em grande escala;
- **Baixa Latência**: Devido a localização próxima aos usuários finais, a latência observada nas transmissões de mensagens proporciona uma boa experiencia com o usuário;
- **Alta Largura de Banda**: Há possibilidade de explorar toda a largura de banda entre o dispositivo servidor de borda e o dispositivo inteligente do usuário final;
- **Identificação de localização**: Esse fator pode fazer com que os dispositivos efetivamente percebam e mudem a forma de como realizar o processamento das informações quando muda-se a localização dos mesmos;
- **Proximidade com Usuário**: Essa características pode melhorar a experiência do usuário, pois os serviços de processamento de dados estão mais próximos dos mesmos.

Segundo (MECHALIKH *et al.*, 2019), as definições de *Edge e Fog Computing* ainda são percebidas como o mesmo paradigma. Nesse contexto, o autor afirma que a diferença entre as abordagens é dada pela não utilização da *cloud*, no caso da computação em borda, como mostrado na figura 3.

O conceito que adotamos para o paradigma *edge computing* nessa dissertação, é dado por (SHI *et al.*, 2016) como modelo de tecnologias que concedem computação ou armazenamento na borda da redes, a partir de dados de serviços relacionados a nuvem (*downstream*) e serviços IoT (*upstream*). Com isso, a visão de computação na borda sugere uma "ponte" entre os dispositivos IoT e o dispositivo computacional físico mais próximo, facilitando o desenvolvimento de novas aplicações que executem nos dispositivos que estão na imediação do ambiente (AVASALCAI *et al.*, 2020).

A figura 4 mostra que, por exemplo, um computador, desktop, laptop, tablet, smatphone ou outro dispositivo pode ser capaz de de processar o fluxo de dados localmente, armazenando por tempo limitado, sendo capaz de se comunicar com dispositivos de IoT (hierar-

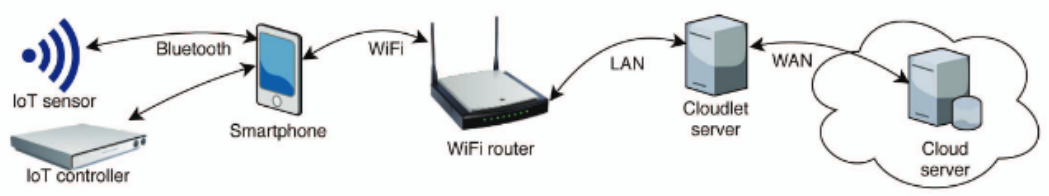
Figura 3 – Arquiteturas Cloud, Fog e Edge Computing.



Fonte: (MECHALIKH *et al.*, 2019)

quia inferior) e *cloudlets*, servidor de borda móvel ou data center em nuvem (hierarquia superior) (GUSEV; DUSTDAR, 2018).

Figura 4 – Solução de *edge computing* usando IoT e dispositivos de borda.



Fonte: (GUSEV; DUSTDAR, 2018)

2.4 Controle de Acesso

O controle de acesso é definido como a metodologia que lida com os direitos de acesso a dados e serviços, por um usuário ou dispositivo autorizado. Seu objetivo é gerenciar privilégios, de forma a limitar, permitir ou negar obtenção de serviços ou informações, garantindo a proteção dos dados (BATE *et al.*, 2017).

O controle de acesso é considerado como a combinação de 3 conceitos: Autenticação que estabelece a identificação onde a entidade fornece as credenciais e essas são verificadas e validadas; Autorização sendo o processo de definição de privilégios de acesso aos recursos por parte da entidade solicitante; e *Accountability* ou auditoria que é o processo de garantia de

que as operações ou atividades realizadas pelos usuário, processos ou sistema são identificadas, possibilitando um mapeamento entre a entidade e o processo realizado (ALKHRESHEH *et al.*, 2020).

Basicamente, podemos resumir a proposta de controle de acesso sentenciando em "**quem e quando pode acessar o quê**". Nesse sentido, a partir de políticas próprias, deve-se decidir pela permissão ou recusa de acesso a determinado recurso, seja ele digital ou físico. A definição apresentada por *Common Criteria* (*Common Criteria* (CC)), estabelece que o AC (*Access Control*) pode ser representado como uma ou mais regras, procedimentos, práticas ou diretrizes de segurança impostas por uma organização à suas operações (KALAM *et al.*, 2018).

Para o desenvolvimento de mecanismos de controle de acesso, (HU *et al.*, 2013) define 3 principais componentes que podem ser abstraídos, sendo eles: **Políticas** , o **Modelo** e **Mecanismo**.

Em nível mais alto, encontram-se as **Políticas** que são basicamente os "requisitos" que especificam como o gerenciamento de quem acessa determinado conteúdo, sob determinadas circunstâncias, pode ser processado. São as regras impostas pelo ambiente o qual o controle de acesso irá agir.

O **Modelo** é responsável por traduzir as especificações das políticas a serem implementadas para os usuários. Ele representa, de maneira formal, as políticas de segurança, estabelecendo as limitações teóricas do sistema bem como a possibilidade de análise e avaliação do mesmo.

O **Mecanismo** traduz as operações entre o usuário e suas requisições baseadas na estrutura proposta pelas políticas e modelo. Há uma gama de mecanismos propostos pela literatura e algumas serão discutidas na seção 2.4.2

O gerenciamento do acesso ao meio em IoT não é trivial e são vários os fatores que influenciam essa complexidade: heterogeneidade de dispositivos, poder computacional, dinamicidade e diversidade do ambiente onde o sistema está inserido (escopo), recursos concorrentes, diversidade de usuários e suas respectivas necessidades, tempo para o acesso, dentre outras (DONG *et al.*, 2018). Nas seções que seguem serão discutidas algumas características de controle de acesso.

2.4.1 Arquitetura

2.4.1.1 Baseado em Políticas

Pode-se visualizar uma arquitetura através de blocos quando o controle de acesso é baseado em políticas, na figura 5. No primeiro bloco, uma entidade solicita, através de uma requisição, o acesso para o Ponto de Acesso da Política (PAP) (Ponto de Acesso da Política). Essa requisição é então interceptada por um bloco chamado de Ponto de Execução da Política (PEP) (Ponto de Execução da Política), para depois ser encaminhada para o Ponto de Decisão Política (PDP) (Ponto de Decisão Política). O PDP envia solicitação ao Ponto de Informações da Política (PIP) (Ponto de Informações da Política) para obter informações relacionadas aos atributos necessários para a execução.

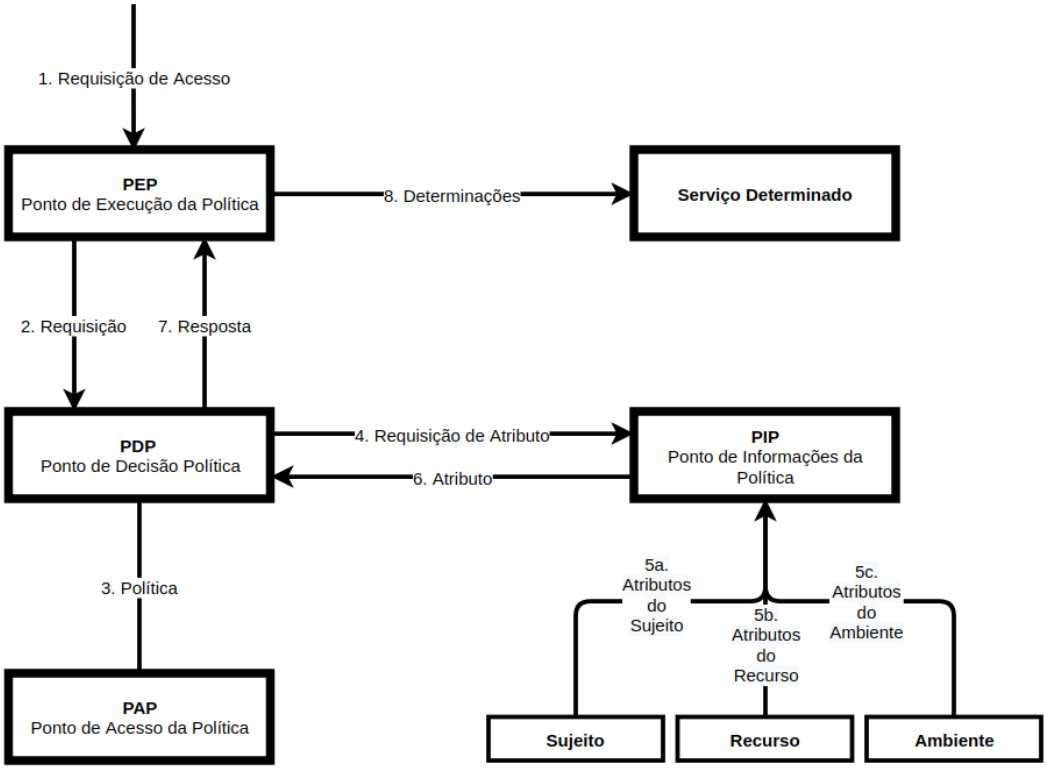
Esses atributos podem estar relacionados ao sujeito que solicitou, o recurso requerido pelo sujeito e do ambiente ou contexto a ser analisado pela política. Com os atributos, o PDP avalia a solicitação tendo como princípios estabelecidos no PAP e o contexto dos atributos. Cabe ao bloco PDP enviar ao PEP a decisão tomada de permitir ou não o acesso ao meio/recurso solicitado, com as merecidas determinações. Por último, o PEP aplica as decisões, cumprindo as determinações enviadas anteriormente pelo PDP.

2.4.1.2 Baseado em Token

Para suprir algumas necessidades de arquiteturas distribuídas em IoT, surge a arquitetura baseada em token (RAVIDAS *et al.*, 2019a). Basicamente, o fluxo é inicializado a partir da requisição de um usuário por um token, codificando os direitos e deveres, fornecidas por um servidor de autorização que, posteriormente, será usado para fornecer o acesso ao recurso requisitado.

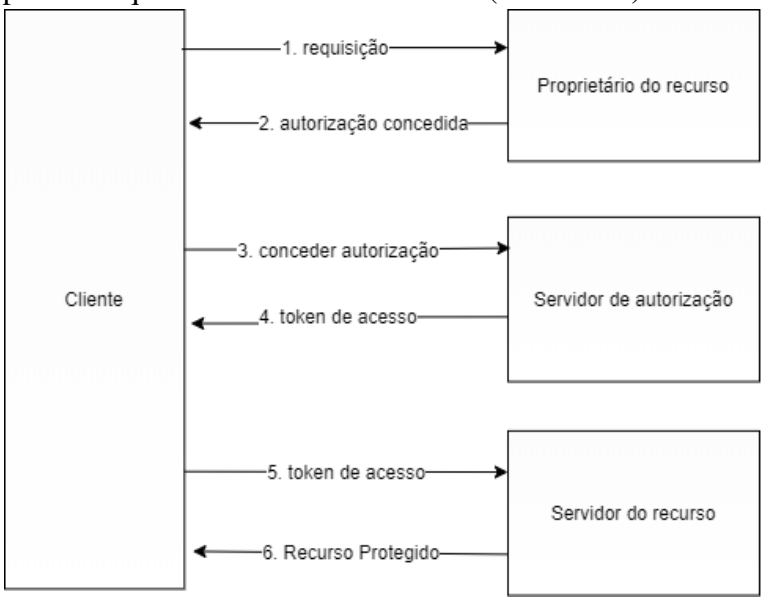
A figura 6 (DENNISS; BRADLEY, 2016) apresenta um exemplo de uso do OAuth 2.0. Como passo 1, há uma requisição de acesso ao proprietário de um recurso. Este, passo 2, devolve a requisição com as autorizações possíveis. No passo 3, o cliente solicita um token de autenticação para o servidor, apresentando para tal, a concessão recebida pelo proprietário do recurso. O servidor devolve o token no passo 4, caso validado os direitos de acesso. O passo 5, o cliente, de posse do token, solicita o acesso ao recurso pretendido e caso o token seja validado pelo servidor de recursos, entrega ou não o recurso protegido (passo 6).

Figura 5 – Fluxograma Controle de Acesso Baseado em Políticas.



Fonte: (HU *et al.*, 2013) (Adaptado)

Figura 6 – Exemplo de Arquitetura baseada em Token (OAuth 2.0)



Fonte: (DENNISS; BRADLEY, 2017)) (Adaptado)

2.4.1.3 Híbridas

Alguns mecanismos se beneficiam das duas arquiteturas como mostra (RAVIDAS *et al.*, 2019a). Os autores propõem utilizar uma arquitetura em que o servidor avalia as políticas e gera o token baseado na avaliação, de forma offline, sem participação do usuário nessa

fase. Essa combinação fornece flexibilidade e desempenho considerável, tendo em vista a natureza dinâmica. Sua aplicação visa o ambiente de sistema Inteligente de Transportes (C-ITS - *Cooperative Intelligent Transport System*).

2.4.2 Mecanismos de Controle de Acesso

O mecanismo realiza as operações de restrições e permissões de acesso em um sistema, a partir das políticas e do modelo associados ao controle de acesso. Sua função é garantir que os ativos sejam acessados por usuários devidamente credenciados, seguindo determinados critérios. Assim, especificando-se os usuários, define quais são seus direitos e privilégios sobre determinados recursos (ALRAMADHAN; SHA, 2017). São vários os mecanismos propostos pela literatura. A seguir, são apresentados uma visão geral, sob óptica de aplicações em IoT:

- ***Access Control List* (ACL) (*Access Control List*):** esse tipo de mecanismo usa uma lista, orientada por coluna, para determinar os direitos dos usuários. A lista possui objetos relacionando um recurso a diferentes pares de sujeitos, usuários ou processos, ligados aos seus direitos perante o sistema (ALRAMADHAN; SHA, 2017). O principal problema desses modelos empregados em IoT é a restrição computacional, intrínseca ao ambiente e a necessidade de arquitetura totalmente centralizada. Requisitos como a escalabilidade ficam comprometidos em resposta à implementação de mecanismo. (ALRAMADHAN; SHA, 2017).
- ***Attribute-based Access Control* (ABAC) (*Attribute-based Access Control*):** procura usar atributos de usuários, de sujeitos/objetos e do ambiente de IoT, para gerenciar o acesso. Os atributos podem ser considerados qualquer informação a ser caracterizada pelo *host*. O que pode definir a liberação ou não de acesso será a política que avaliará essas características. Nesse sentido, todas as funções, capacidades ou recursos são usados como atributos para definir os direitos e permissões ao usuário. (HEMDI; DETERS, 2016) relata o fato de que o controle de acesso e ambientes IoT deve ser direcionado para as características específicas do usuário, dispositivos e ambiente (contexto).
- ***Role-based Access Control* / falha de blindagem (RBAC) (*Role-based Access Control*):** esse tipo de mecanismo tem como foco o papel ou função do usuário, dentro do sistema, como a credencial para o controle de acesso ao meio (SANDHU *et al.*, 1996). Como relata (WANG *et al.*, 2016), a utilização desse mecanismo é vantajosa quando há a possibilidade de ser empregado em ambientes onde há estruturas hierárquicas (ALRAMADHAN; SHA,

2017). Esse mecanismo pode se tornar relativamente inflexível e de baixo poder de detalhamento de informações (granularidade) tendo em vista um ambiente onde haja diversidade de usuários. (DONG *et al.*, 2018)

- ***Capability-based Access Control* (CapBAC) (*Capability-based Access Control*):** (DENNIS; HORN, 1966) explica que a ideia fundamental para aplicação desse modelo está no conceito de *"capability"*. Esse, por sua vez, trata-se de uma chave ou *"token"* responsável, por fornecer a permissão para acesso ao objeto, entidade ou sistema. Esse elemento carrega informações que relacionam a identidade do usuário ou dispositivo, relacionando-a com as permissões e direitos de acesso sobre as operações e recursos do sistema. Pode ter uma boa aplicabilidade em sistemas IoT por ter uma boa escalabilidade, uma maior flexibilidade e usabilidade (KARIMIBIUKI *et al.*, 2018).
- ***Usage Access Control* / (UCON) (*Usage Access Control*):** Esse modelo foi considerado uma incógnita por (PARK; SANDHU, 2002), porém é visto com bons olhos para a nova geração de modelos de CA por (BOUANANI *et al.*, 2019). Este último relata que, no modelo de acesso baseado em uso, há uma forma de contínua da autorização de acesso, ou seja, a avaliação é realizada antes, durante e depois da solicitação do acesso. Nesse sentido, há a vantagem que, caso algum determinado atributo seja alterado durante o processo de tomada de decisão, o sistema pode revogar um acesso concedido anteriormente e o uso cancelado. Os métodos tradicionais consideram que todos os usuários já são conhecidos, o que é chamado de sistema fechado. O mecanismo de controle UCON, considera que os usuários estão parcialmente autenticados ou ainda desconhecidos, além dos já autenticados (LEE *et al.*, 2015).

Outros modelos são propostos - *Organization-based Access Control* (OrBAC) (*Organization-based Access Control*) (BOUIJ-PASQUIER *et al.*, 2015); *Trust-Based Access Control* (TBAC) (*Trust-Based Access Control*) (BERNABE *et al.*, 2016); *Identity-Based Access Control* (IBAC) (*Identity Based Access Control*) (LI *et al.*, 2016) - tendo como base os modelos citados anteriormente, possuindo uma extensão ou adaptação ao paradigma de IoT, sendo esse uma possibilidade de desafio para pesquisadores.

Outras propostas utilizam tecnologias emergentes como blockchain (BINDRA *et al.*, 2019), fornecendo a vantagem de uma abordagem descentralizada baseada em consenso e *Software-Defined Networking* / distância lateral (*SDN*) (*Software-Defined Networking*) (NGU *et al.*, 2016) que facilita a comunicação entre os dispositivos diante do problema da dinamicidade

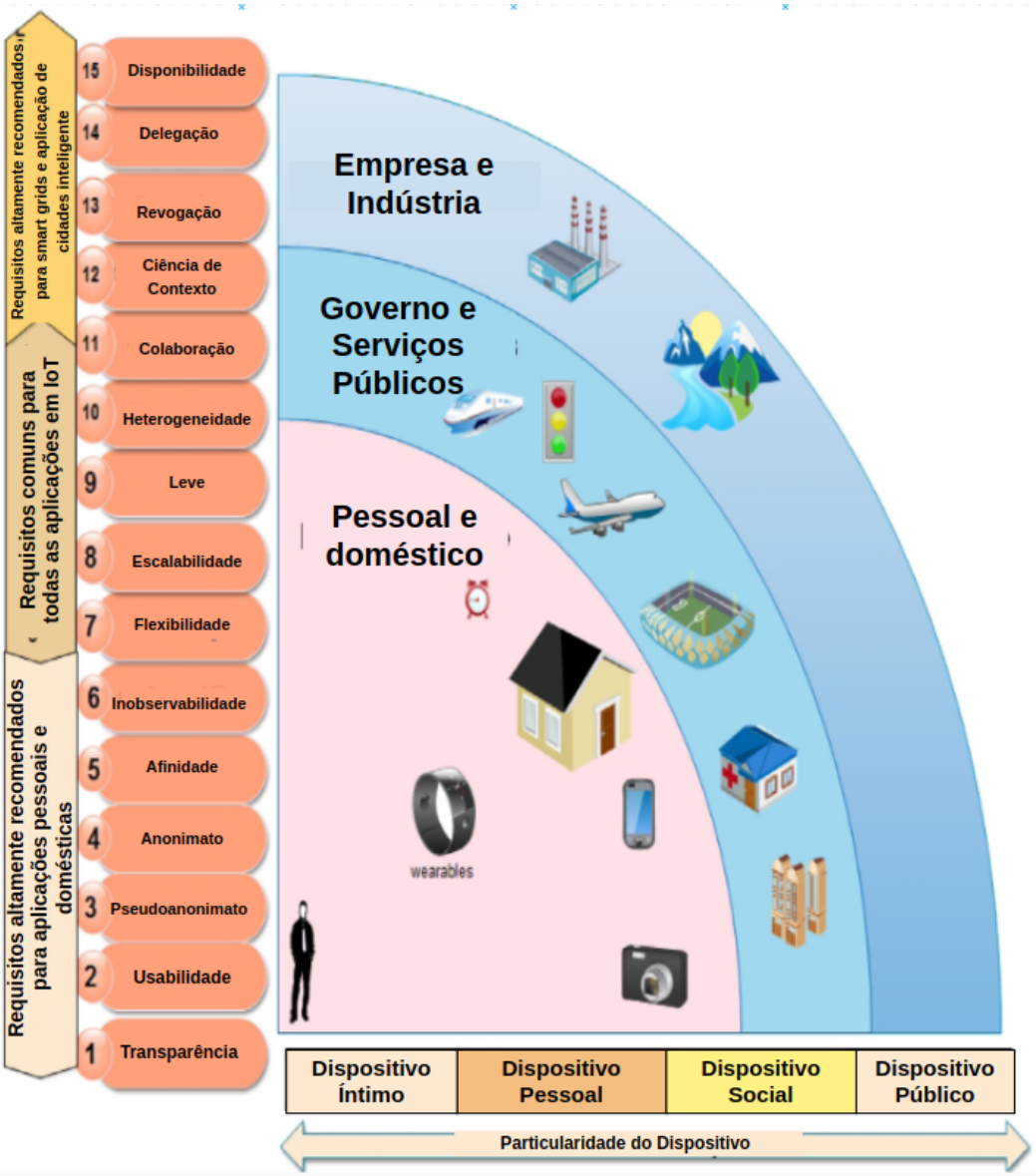
em IoT (PAL *et al.*, 2020).

2.4.3 Requisitos para Controle de Acesso em Ambiente IoT

Requisitos para aplicações em IoT são estudados pela comunidade e há consenso entre vários requisitos e características que diferenciam de sistemas tradicionais. Nesse sentido, (OUADDAH *et al.*, 2017) ffornece uma extensa revisão do estado da arte de diferentes soluções de controle de acesso em IoT. Assim, este estudo realiza uma análise de objetivos.

A seguir, a figura 7 faz um resumo dos requisitos de segurança e os domínios de aplicações em IoT:

Figura 7 – Taxonomia de aplicações em IoT e seus requisitos de segurança.



Fonte: (OUADDAH *et al.*, 2017) (Adaptado)

(AHMAD; RANISE, 2018) analisa os requisitos para validação de sistemas de controle de acesso:

- **Granularidade:** especificação de políticas com granularidade fina;
- **Administração:** fácil administração
- **Portabilidade:** aplicação independente da plataforma;
- **Extensibilidade:** apoiar a aplicação de restrições de segurança;
- **Latência:** projeto de acordo com a latência necessária conforme as características do ambiente IoT;
- **Confiabilidade:** fornecer decisão confiável em vários diferentes cenários;
- **Escalabilidade:** desenvolvimento de mecanismo capaz de lidar com o aumento de dispositivos e quantidade de dados.

No estudo apresentado por (RAVIDAS *et al.*, 2019b), ão apresentados requisitos e princípios para design a serem considerados no desenvolvimento de sistemas de controle de acesso em IoT. Os requisitos são categorizados em 3 aspectos: especificação de política, gerenciamento de políticas e avaliação e aplicação de políticas.

- **Especificação de políticas:**
  - especificação de políticas de controle de acesso refinadas (granularidade).
  - capacidade de lidar com a dinamicidade de nós e ambientes IoT (dinamicidade).
- **Gerenciamento de políticas:**
  - capacidade de lidar com a complexidade dos ambientes de IoT.
  - prover facilidade para usuários no gerenciamento de políticas.
  - permitir o gerenciamento de políticas em vários domínios administrativos.
- **Avaliação e aplicação de políticas:**
  - decisão de acesso automatizada.
  - deve ser leve a ponto de não afetar significativamente os recursos de computação e comunicação de dispositivos com restrição de recursos.
  - não devem afetar o desempenho do sistema IoT.
  - deve ser coerente em vários domínios administrativos (contexto).
  - devem estar sempre operacionais (disponibilidade).

2.5 Conclusões

Este capítulo apresenta elementos conceituais e teóricos como base para realização do estudo aqui exposto. Concepções acerca de Internet das Coisas em relação às arquiteturas de referências, protocolos de comunicação da camada de aplicação além de suas aplicações foram apresentadas de forma a proporcionar uma visão estrutural de como o controle de acesso pode e deve atuar com a aplicação. Aspectos relacionados à confiabilidade em ambiente IoT também foram alvo do estudo.

Princípios sobre arquitetura de computação em nuvem, na névoa e na borda da rede foram necessários para direcionar a pesquisa com relação a questões de confiabilidade em ambiente IoT. Assim, a escolha de aplicação em edge computing foi definida por garantir características como acesso em tempo real, baixa latência e alta largura de banda, fatores esses básicos para confiabilidade da aplicação em IoT.

Princípios de controle de acesso proporcionaram uma análise e definição de arquitetura da ferramenta. A possibilidade de usar modelo de políticas baseadas em contexto favorece a utilização da arquitetura apoiada em computação em borda e confiabilidade.

Durante o estudo, foi constatada a necessidade de utilização de requisitos específicos para controle de acesso para o ambiente IoT. Devido à natureza do paradigma, foi identificado a necessidade de adequação da proposta de controle de acesso à restrições impostas pelo paradigma de IoT.

3 TRABALHOS RELACIONADOS

O objetivo deste capitulo é relacionar trabalhos acadêmicos investigados que possuam correlações com a pesquisa aqui apresentada. Os estudos compreendem propostas de frameworks de autorização para acesso em ambientes IoT, apresentando arquitetura e infraestrutura de comunicação. Desta forma, trabalhos que investigam prioritariamente mecanismos de autenticação e/ou auditoria (*accountability*) não fazem parte do escopo deste trabalho.

O procedimento de coleta e análise das pesquisas aqui relacionados foi efetuado a partir de uma pesquisa bibliográfica realizada seguindo as bases da *IEEE Xplorer, ACM Digital Library, Springer e ScienceDirect*. Para tal, o conjunto de combinações de palavras-chave foi inserido nas plataformas listadas:*"Access Control" AND ("IOT" OR "Internet Of Things") AND "Authorization"*. Com base na leitura dos resumos e em seguida a leitura transversal das propostas, foram selecionados cinco trabalhos que abordam a conformidade e discussão de pelo menos um dos requisitos citados na seção 2.4.3.

3.1 SAFIR: Secure Access Framework for IoT-enabled Services on Smart Buildings

O trabalho de (HERNÁNDEZ-RAMOS *et al.*, 2015) apresenta a proposta de um *framework* de segurança IoT compatível com a arquitetura de referência Ar*Achitectural Reference Mode* (ARM) (*Architectural Reference Mode*) e analisa sua viabilidade em cenários de edifícios inteligentes. A base para o controle de autorização, são aspectos contextuais, integrando-os como componente fundamental para a estrutura implementada. Além disso, módulos que funcionam como gerenciador de níveis de confiança e gerenciamento de chaves e identidade são considerados na estrutura.

Para o autor, aplicações que envolvam *smart cities e smart buildings* são centradas no usuário, pois as pessoas passam o maior tempo nesses locais. Como objetivo, o trabalho se propõe a proteger os recursos por meio de mecanismos de autenticação e autorização, como forma de alcançar uma abordagem holística. Como forma de relacionar contexto, o uso de dados de localização é base para acesso aos recursos do prédio, sendo, para isso, necessária aplicação centrada no usuário.

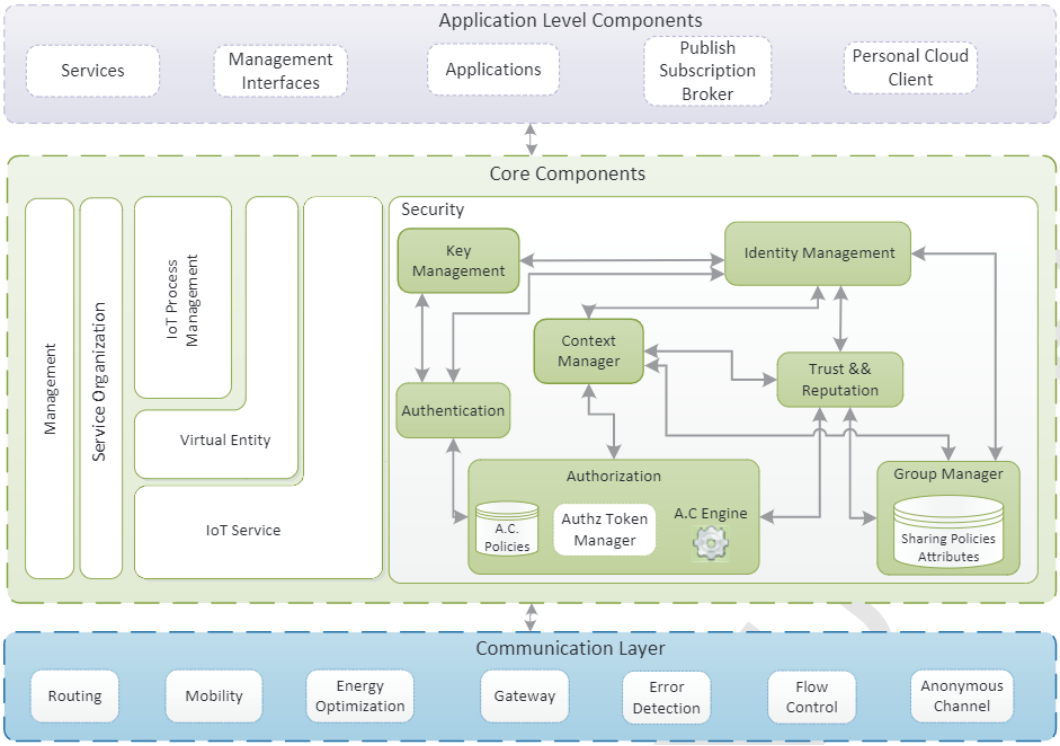
A implementação é uma instância do projeto chamado *"City Explorer"*, cujo objetivo volta-se às aplicações em cidades inteligentes. O autor destaca que as interações em ambiente de IoT são, geralmente, regradas por associações fracas e pouco duradouras (voláteis), em que não

há, entre as partes, um nível de confiança previamente estabelecido.

Nesta pauta, é necessária uma flexibilidade para gerenciar essas relações, dada a dinamicidade do ambiente o qual trata no trabalho, e para isso elabora um gerenciador de contexto e de grupo de usuários. O mecanismo de autorização sugerido pelo autor utiliza arquitetura híbrida, em que além de geração de políticas por meio padrão usando *eXtensible Access Control Markup Language* (*eXtensible Access Control Markup Language* / distância lateral (*XACML*)) é utilizado o mecanismo baseado em token, esta última dispõe de estrutura de criptografia leve *Elliptic-curve cryptography* (*Elliptic-curve cryptography* (ECC)).

Uma visão geral e demais componentes da estrutura do trabalho, podem ser vistos na figura 8.

Figura 8 – Arquitetura da proposta por (HERNÁNDEZ-RAMOS *et al.*, 2015).



Fonte: (HERNÁNDEZ-RAMOS *et al.*, 2015)

Como forma de avaliação da performance da estrutura, o autor realiza uma análise de desempenho com relação à descoberta de serviços, a energia consumida pelo funcionamento, bem como as interações que exijam segurança: geração e acesso ao token, autenticação e autorização. O quadro de utilização de dispositivos com alto poder computacional pode ser inviável em alguns contextos.

3.2 Access Control Framework for API-Enabled Devices in Smart Buildings

A proposta de (BANDARA *et al.*, 2016) consiste em um framework para controle de acesso em prédios inteligentes, como suplemento para *Application Programming Interface* (API) em edifícios inteligentes. Nesse caso, o gerenciador de segurança é considerado um terceiro sistema, diante da interação entre o usuário e a rede de sensores. O objetivo é facilitar a gerência da segurança, realizar cálculos complexos de segurança (suporte a ambientes com restrição computacional). Para garantir que apenas usuários autorizados possam acessar o dispositivo, este é forçado a solicitar ao gerente de segurança para avaliar solicitações.

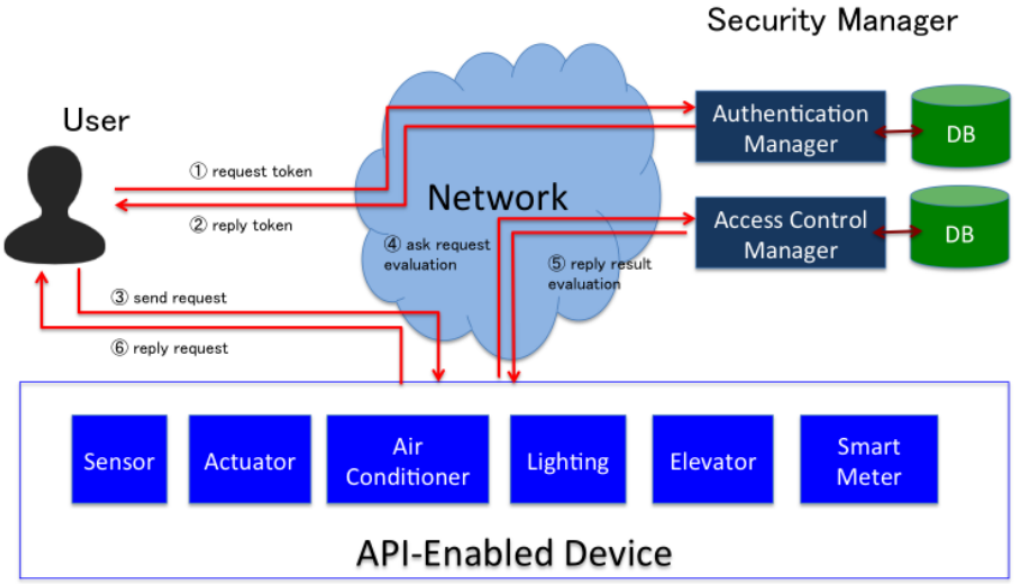
O gerenciador de segurança é dividido em dois módulos: de autenticação e outro de autorização. O agente de autenticação é responsável por verificar as credenciais de usuários e geração de tokens de acesso. Além de permitir que o usuário envie informações de suas credenciais como parâmetros para o acesso, ele pode optar por uma autenticação baseada em confiança (*"trust"*), a qual é medida a partir do uso. O nível de confiança medido restringe assim os níveis de acesso que o determinado usuário possui. Podemos notar a flexibilidade do sistema com relação à operação de autenticação.

O gerenciador de autorização utiliza-se de políticas implementadas pelo padrão XACML, e, segundo o autor, por prédios inteligentes estarem em condições complexas no âmbito de dispositivos, usuários e contextos, as políticas baseadas em atributos são utilizadas. Nessa estrutura, o administrador do recurso elabora a política desejável e armazena em um repositório, o que facilita a flexibilidade diante do contexto de prédios inteligentes.

O administrador ou o proprietário do recurso define a política de segurança e salva no repositório de políticas. Com o controle de acesso baseado em políticas, vários modelos de controle de acesso podem ser adotados com flexibilidade. Além disso, a política no mundo físico também pode ser facilmente criada. Em sua estrutura, um usuário verificado enviará uma solicitação diretamente aos recursos. Quando um recurso recebe uma chamada de um usuário, a existência e a expiração do token de acesso serão verificadas. Se o token de acesso existir e não expirar, o lado do recurso será forçado a solicitar ao gerente de controle de acesso a avaliação da chamada aceita. .

Caso uma entidade deseje acesso a algum recurso, este deve possuir um token válido. Caso exista, uma chamada para o gerenciador de controle de acesso é requisitada e anexada uma pontuação para nível de confiança, que varia de acordo com as políticas preestabelecidas pelo administrador de políticas. A arquitetura proposta pelo autor é mostrada na figura 9.

Figura 9 – Arquitetura da proposta por (BANDARA *et al.*, 2016)



Fonte: (BANDARA *et al.*, 2016)

Para avaliação da proposta, o autor realiza testes em ambiente com mais de 300 dispositivos, medindo seu desempenho por meio de experimentos, verificando aspectos relacionados à segurança, usabilidade, escalabilidade e performance. Embora tenham-se elencados aspectos interessantes para o estudo, os experimentos são acercados à latência.

3.3 Context-aware Automatic Access Policy Specification for IoT Environments

(ALKHRESHEH *et al.*, 2018) propõe, diante das técnicas tradicionais de especificações de políticas de controle de acesso, automatizar a geração delas de acordo com o contexto, objetivando mais flexibilidade e adaptabilidade, diante dos ambientes de IoT dinâmicos.

Para o serviço de autorização, o autor utiliza atributos das entidades envolvidos na requisição de acesso para a geração de políticas e o contexto a qual estarão submetidas. O contexto então é definido pelo administrador das políticas, visando restrições para acesso, a qual é definida como *"context guard"*.

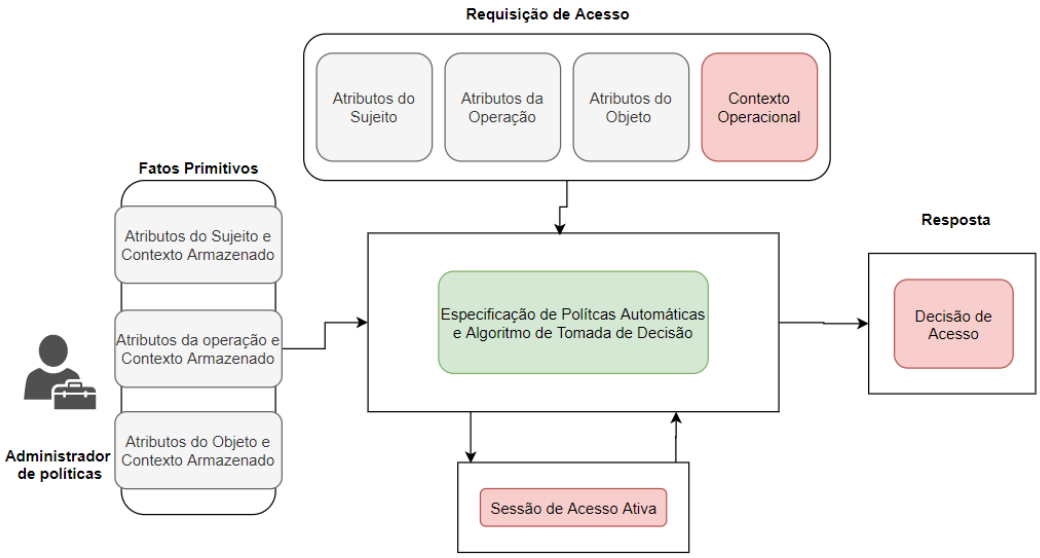
Uma outra estrutura relacionada ao contexto, chamada de *"operational context"*, define um conjunto de associações correlacionando o objeto a ser acessado, com o sujeito que solicita o acesso. Essas duas estruturas trabalham em conjunto de maneira que, se o *"operational context"* for correspondente ao *"context guard"*, o requisitante poderá realizar as ações permitidas para o contexto em questão..

Para a geração de políticas automáticas, o sistema recebe uma requisição que pode ser

interpretada como uma consulta com 3 parâmetros: o sujeito que deseja acessar, a operação que o sujeito deseja realizar sobre o objeto a ser acessado. Assim, durante a requisição, os atributos destes 3 parâmetros são extraídos e juntamente com parâmetros do *"operational context"*, são avaliados para permissão ou não de acesso.

(ALKHRESHEH *et al.*, 2018) utiliza como estudo de caso acesso aos recursos em uma universidade, fazendo a análise e comparando o seu algoritmo de geração de políticas automáticas com outro que utiliza políticas estáticas. A seguir, a arquitetura do projeto pode ser visualizada na figura 10.

Figura 10 – Arquitetura da proposta por (ALKHRESHEH *et al.*, 2018) (Adaptado).



Fonte: (ALKHRESHEH *et al.*, 2018) (Adaptado

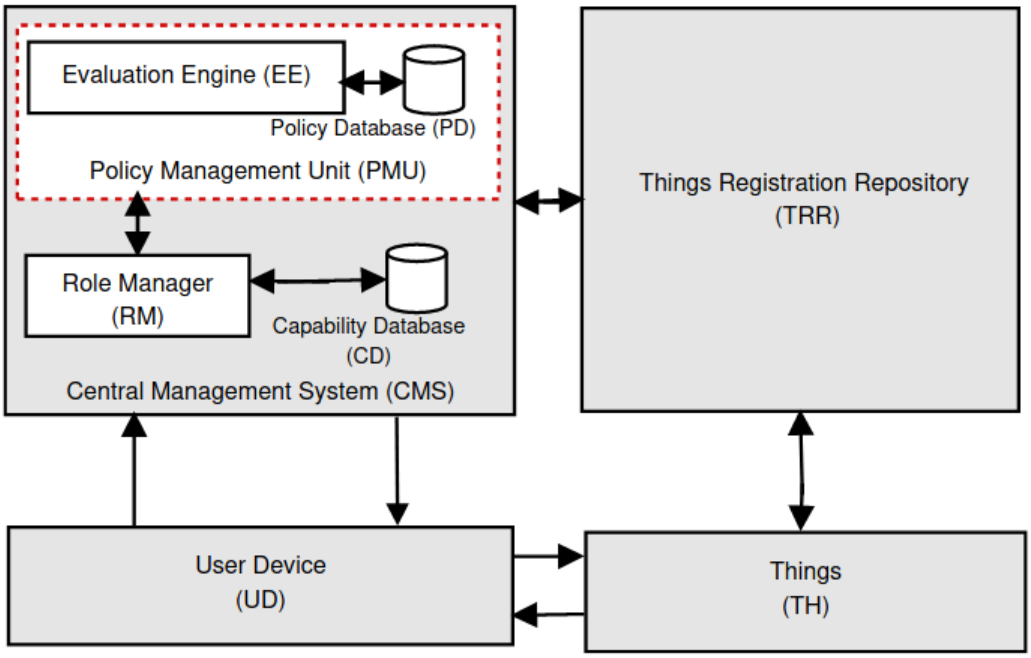
A comparação é realizada sob uma perspectiva de gerenciamento de políticas, inclusão de novos usuários, inclusão de novas regras de acesso e resolução de conflitos. Apesar da modelagem ser definida em amplos aspectos do controle de acesso, nenhum tipo de experimentação ou análise de desempenho é realizada.

3.4 On Design of A Fine-Grained Access Control Architecture for Securing IoT-Enabled Smart Healthcare Systems

Tendo em vista os problemas de segurança de dados provenientes do uso de tecnologias IoT em saúde, mais especificamente a necessidade de proteção de "coisas" inteligentes contra acessos não autorizados, (PAL *et al.*, 2017) propõe uma arquitetura para controle de

acesso que melhora o gerenciamento de políticas de forma a reduzi-las e provê um controle de acesso refinado para o domínio de *Smart Healthcare*. Em seu modelo, há uma abordagem híbrida em que são utilizados como base atributos, funções e recursos. Os atributos são utilizados para associar uma entidade a uma determinada função e também para avaliar as permissões. Com base em atributos adicionais, os recursos especificados por políticas podem receber solicitações de acesso e a partir desta, o acesso pode ser concedido ou não.

Figura 11 – Arquitetura da proposta por (PAL *et al.*, 2017)



Fonte: (PAL *et al.*, 2017)

A arquitetura da proposta (figura 11) possui basicamente 4 blocos: *User Device* (*User Device* / falha de blindagem (UD)) que representa o dispositivo o qual pertence a um usuário (médico, por exemplo, *Things* (TH) que representa um ojeto inteligente, por exemplo um sensor em um paciente, *Things Registration Repository* (*Things Registration Repository* (TRR)) onde serve de repositório para informações sobre os objetos inteligentes e *Central Management System* (*Central Management System* (CMS)) que é o elemento principal que verifica atributos e emite as possibilidades de recursos a serem acessados pelo usuário, gerenciando assim o mapeamento de permissões de acordo com os papéis e políticas pré-definidas.

Os autores avaliam a proposta por meio experimentos, em que se verificam tempos de resposta de comunicação para acesso ou não do usuário (UD) e o objeto Inteligente (TH). Foram analisados os tempos em vários cenários de utilização: quando a solicitação do recurso

é rejeitada por tempo expirado para o acesso; quando o acesso está disponível por um período válido, mas o recurso não está disponível para o usuário; quando o tempo e o recurso estão compatíveis e o tempo de verificação de uma condição simples de acesso. Além disso, foram analisadas as falhas nesses cenários. A comunicação entre os componentes é feita utilizando protocolos convencionais, que podem resultar em problemas para ela mesma.

3.5 An Attribute-Based Distributed Access Control for Blockchain-enabled IoT

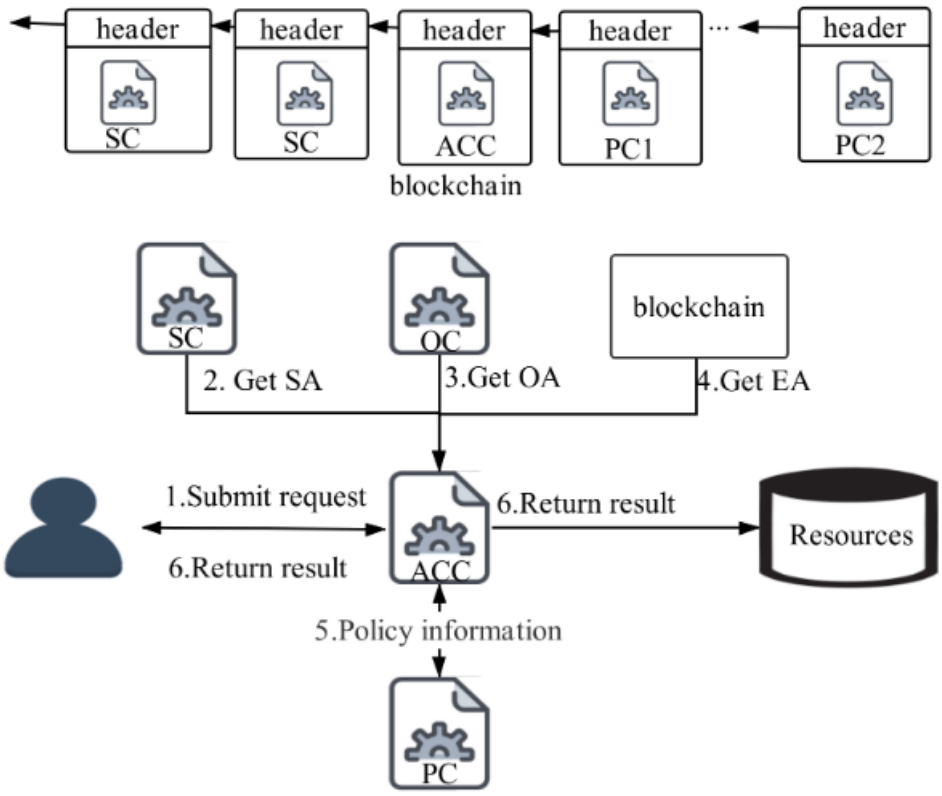
(WANG *et al.*, 2019) Propõe o *framework* de controle de acesso distribuído baseado em atributos chamado *Atribute-Based Distributed Access Control* (ADAC) (*Atribute-Based Distributed Access Control*), que associa a tecnologia blockchain e atributos de dispositivos IoT.

Para o autor, atributos específicos dos objetos, como fabricante, por exemplo, são considerados na proposta com objetivo de refinação nas especificações das políticas. Em sua estrutura, o sistema possui um contrato inteligente que inclui um contrato para sujeito/usuário (SC) que gerencia as contas dos fabricantes, dispositivos e outras informações dos sujeitos; contrato para o objeto (OC), que gerencia informações sobre atributos dos objetos de cada dispositivo; um contrato para o controle de acesso (ACC), que é usado para determinar o acesso das requisições estão de acordos com as políticas pré-cadastradas; e vários contratos que representam as políticas (PCs), que são contratos criados vinculados aos dispositivos dos usuários.

A figura 12 mostra os elementos e o fluxo para o processo de requisição e acesso ao recurso. Na proposta, depois de receber a solicitação de acesso do usuário, o ACC obtém atributos do SC, OC e da cadeia *blockchain* (por exemplo, *timestamp*), obtendo em seguida as políticas fornecidas pelo PC, finalizando o processo realizando o cálculo de permissões ou proibições de acesso ao recurso sob determinada operação.

Os autores avaliaram a proposta através de um estudo de caso especificando uma arquitetura e utilizando no experimento a rede *Ropsten*, essa oficialmente fornecida pela *Etherium*. Além disso, a simulação contou com um laptop que simula um dispositivo leve (cliente) que requisita o acesso e outro que representa um gateway para o acesso aos dispositivos IoT. Apesar do estudo ter contado com uma considerável avaliação por meio da simulação, não foi identificada uma estruturação de análise de desempenho relacionados a consumo computacional.

Figura 12 – Arquitetura da proposta por (WANG *et al.*, 2019).



Fonte: (WANG *et al.*, 2019) (Adaptado

3.5.1 Comparação entre trabalhos

Podemos visualizar uma comparação entre os estudos mencionados neste trabalho por meio da Tabela 1. Dentre as características, destaca-se a realização de um estudo sobre a confiabilidade em IoT e a realização de análise de desempenho. Em relação ao modelo de políticas, consideramos mais adequado a utilização de contexto como base para a produção de políticas, aproveitando o conceito de *edge computing*.

Os modelos tradicionais de controle de acesso, o *Discretionary Access Control* (DAC) - *Discretionary Access Control*, por exemplo atuam de forma manual, onde as propriedas são manipuladas por um administrador do sistema. No entrando, os mecanismos mais flexíveis, como o ABAC, tornam-se mais aplicáveis em ambientes IoT, como é vistos nos trabalhos citados.

A possibilidade de utilizar elementos do contexto também aumenta o refinamento para indicações de acesso, facilitando ainda seu gerenciamento. O presente trabalho considera que um modelo de controle de acesso baseado em contexto torna-se mais adequado se considerarmos a aplicação em ambiente IoT.

Os domínios de aplicações são elementares para a determinação do modelo de

manuseamento de acesso. Nesse sentido, é necessário determinar para qual âmbito será a atuação da aplicação. Assim, os trabalhos aqui apresentados, com exceção do (WANG *et al.*, 2019), especificam um cenário de aplicação, sendo o mais recorrente, Prédios Inteligentes, em que este também definido com caso de uso deste trabalho.

Estudos em que a confiabilidade é abordada não são tratados nos trabalhos aqui expostos, com ressalvas ao trabalho de (PAL *et al.*, 2017). Mas a análise se resume à investigação de comportamento do atributo de latência em diferentes cenários de controle de acesso.

A estruturação de uma avaliação de desempenho da aplicação do controle de acesso pode apontar elementos importantes para implantação da abordagem. Os estudos evidenciados não aplicam uma metodologia que possa evidenciar dados do funcionamento do serviço/sistema, auxiliando em pareceres sobre ele. Esta dissertação apresenta uma metodologia proposta por (BUKH, 1992), a qual cita que o desempenho é critério chave para projetos, aquisição e uso de sistemas computacionais, sendo necessário um conhecimento inerente ao sistema a ser analisado afim de comparar diferentes alternativas que melhor abrigam os seus requisitos.

Tabela 1 – Trabalhos Relacionados

Trabalho	**Modelo de Políticas**	**Domínio de Aplicação**	**Confiabilidade**	**Análise de Desempenho Estruturada**
Hernández-Ramos et al. 2015	Atributos	Prédios Inteligentes	Não	Não
Bandara et al. 2016	Atributos	Prédios Inteligentes	Não	Não
Alkhresh et al. 2018	Contexto	Prédios Inteligentes	Não	Não
Pal et al. 2017	*Capability*	Saúde	Sim	Não
Wang et al. 2019	Atributos	Não Específica	Não	Não
Este Trabalho	**Contexto**	**Prédios Inteligentes**	**Sim**	**Sim**

Fonte: Próprio Autor

3.6 Conclusões

Este capítulo expos uma discussão e uma comparação entre trabalhos relacionados que propõem arquitetura para controle de acesso em ambiente IoT. Foi possível identificar que a

maioria dos trabalhos não realizam uma análise de desempenho que possa favorecer investigações sobre confiabilidade.

É importante notar que a maioria dos trabalhos apresenta experimentos, modelagem e implementação, mas não há foco em confiabilidade. Além disso, os trabalhos determinam um domínio de aplicação específico, sendo na sua maioria, prédios inteligentes.

4 CONTROLE DE ACESSO EM INTERNET DAS COISAS BASEADO EM EDGE COMPUTING

Este capítulo apresenta as principais contribuições trazidas por esta dissertação. Inicialmente, um cenário motivador é apresentando. A seguir, a técnica de controle de acesso baseada em políticas aqui construída é discutida. Por fim, a arquitetura geral do trabalho e os principais módulos são apresentados.

4.1 Introdução

A disponibilidade em IoT é capacidade com a qual um sistema provê acesso aos recursos quando requisitados. Este conceito inclui acesso aos recursos de software, que indica que o serviços estão sendo providos para quem tem direito de acessá-los, e hardware, quando há compatibilidade com as aplicações e protocolos de IoT, fornecidos para benefícios de usuários. Assim, um modelo de controle de acesso tem por objetivo garantir que apenas dispositivos e usuários autorizados devem ter acesso aos recursos em um dado momento.

A definição de quando um dispositivo ou um usuário deve ter acesso a determinado recurso é parte das políticas de segurança de uma organização. Elas utilizam informações sobre esses usuários e dispositivos, como ainda informações de contexto. Por exemplo, sensores podem capturar sinais físicos como temperatura, luminosidade, umidade e converter em sinais digitais, assim como câmeras podem captar imagens do mundo real e fornecer dados de video ou fotos e microfones que capturam sons do ambiente entregam dados de áudio. Esses dados precisam ser entregues em tempo real para usuários autorizados (LIU *et al.*, 2020a).

O mecanismo de controle de acesso deve decidir em tempo de execução e de forma não supervisionada quem pode utilizar determinando serviço. Ao controlar o acesso aos recursos em um ambiente dinâmico, a disponibilidade do sistema como um todo no sistema IoT tende a melhorar.

4.2 Cenário Motivador

Para demonstrar e ilustrar os aspectos e arquitetura deste trabalho, um caso de uso para o domínio de prédio inteligente é descrito, contemplando modelo e arquitetura.

Para o estudo, consideramos o cenário de um prédio inteligente. O conceito de prédios inteligentes, segundo (WALLACE; MARSHALL, 2003), compreende as estruturas que

integram sistemas, comunicação e controle para criar um ambiente que seja flexível, eficiente, confortável e seguro. Essas infraestruturas propiciam diversas funcionalidades, ao mesmo tempo que promovem sustentabilidade, uso consciente de tecnologia e ambiente saudável.

De acordo com o contexto no qual o prédio se apresenta, outros serviços são passíveis de serem monitorados e controlados, como o controle de umidade do ar, temperatura, luminosidade, acesso físico por intermédio de catracas, monitoramento de gás, ângulo da posição de janelas, dentre outros. A presença automatizada desses serviços através de integração de redes de sensores de maneira consciente e confiável pode garantir conforto e sustentabilidade para o usuário da tecnologia.

Dentro de um prédio inteligente, diferentes tipos de usuários podem ser caracterizados. Podemos relacionar algumas categorias de usuários: i. *Administrador*, quando a entidade exerce a função de gerenciar e supervisionar o prédio na sua totalidade; ii. *habitantes*, quando um prédio possui fins residenciais; iii. *colaborador*, para pessoas que realizam suas atividades de trabalho ou não de forma regular e frequente, ocupando o espaço por um longo período, por exemplo; iv. *visitantes*, para ambientes residenciais, mas por pessoas que não moram e frequentam o ambiente; v. clientes, usuários que possam realizar serviços ou visitas de curta duração. Nesse aspecto, reconhecer esses grupos e gerenciar o acesso, se baseando nos privilégios que cada um possa ter, requer um reconhecimento do contexto o qual a aplicação ou o usuário está inserido.

Diante deste cenário, poderíamos definir as seguintes políticas:

1. Qualquer morador do prédio que tenha apartamento no *BlocoA*, tem acesso às câmeras e aos dados monitorados por elas;
2. Quando a temperatura de uma sala está abaixo de 20º, nenhum usuário pode acionar outro resfriador de ambiente;
3. No período entre 12h e 17h, as luzes devem ser desligadas, sendo acionadas apenas pelo administrador;
4. Um usuário visitante tem acesso por 30 minutos ao sistema de controle de umidade.
5. Os funcionários do setor de manutenção da rede elétrica possuem acesso à todos os equipamentos relacionados à sua função, mas apenas no período em que se encontrem no prédio.
6. Todos os usuários podem receber notificações sobre o sistema de alarme contra incêndios.

4.3 Modelo de Políticas

Considerando os exemplos de políticas citados, a utilização de atributos de contexto é necessária para o suporte de expressões de políticas refinadas, dinamicidade do ambiente e a confiabilidade para o modelo de autorização.

Atributos de contexto são utilizados para a gerência do acesso, uma vez que descrevem as entidades envolvidas representando os elementos físicos e situacionais. Podem ser representados por estruturas de dados mais complexas (ontologias) ou mais simples (chave-valor) e podem ser acoplados em categorias, como dispositivos de uma sala, usuários comuns, habitantes do prédio, sensores de temperatura, por exemplo. (BROSSARD *et al.*, 2017).

Os atributos descrevem as entidades envolvidas na requisição, sendo elas:

- **Sujeito**: entidade que executa ação de solicitar o acesso a um recurso. Pode ser um usuário de uma aplicação ou a própria aplicação. Os atributos dessa entidade ajudam a caracterizar e definir a identidade podendo ser relacionados com um ID único, nome, grupo, função, etc.
- **Operação**: indica qual procedimento foi solicitado pelo sujeito em relação ao recurso. Os atributos descrevem a ação que foi solicitada, por exemplo: leituras, escritas, atualizações e exclusões;
- **Recurso**: relaciona o alvo a ser atingido pelo sujeito. Como exemplo, pode ser serviço de relatório de telemetria, sensor ou grupo de sensores específicos, relatório de dados armazenados, atuador ou grupo de atuadores, componente específico do sistema, etc. Caso a aplicação esteja relacionada a uma Smart Home, uma lâmpada pode possuir atributos como identificação única, proprietário, subsistema o qual pertence (Luminosidade), data de instalação, etc.
- **Contexto**: os atributos de contexto estão relacionados às características do ambiente e às circunstâncias da operação. Horário e data correntes, dia da semana e local são exemplos de atributos relacionados ao contexto.

A partir do modelo de requisição, os atributos são extraídos para análise e execução das políticas. O conjunto de atributos de Sujeito (Atributo de Sujeito (AS)), de Objetos (Atributos de Objeto (AO)), de Recursos (Atributos de Recurso (AR)) e Contexto (*Access Control* (AC))

armazenam os dados em uma estrutura definida através de arranjos do tipo “chave-valor", .

$$Atributos = \begin{cases} AS_{\mathrm{i}}, & \{as_{\mathrm{j}} : \text{“}valor\text{”}, as_{\mathrm{j+1}} : \text{“}valor\text{”}, \ ..., \ as_{\mathrm{n\text{-}1}} : \text{“}valor\text{”}, \ as_{\mathrm{n}} : \text{“}valor\text{”}\} \\ AO_{\mathrm{i}}, & \{ao_{\mathrm{j}} : \text{“}valor\text{”}, ao_{\mathrm{i+1}} : \text{“}valor\text{”}, \ ..., \ ao_{\mathrm{n\text{-}1}} : \text{“}valor\text{”}, ao_{\mathrm{n}} : \text{“}valor\text{”}\} \\ AR_{\mathrm{i}}, & \{ar_{\mathrm{j}} : \text{“}valor\text{”}, ar_{\mathrm{j+1}} : \text{“}valor\text{”}, \ ..., \ ar_{\mathrm{n\text{-}1}} : \text{“}valor\text{”}, ar_{\mathrm{n}} : \text{“}valor\text{”}\} \\ AC_{\mathrm{i}}, & \{ac_{\mathrm{j}} : \text{“}valor\text{”}, ac_{\mathrm{i+1}} : \text{“}valor\text{”}, \ ..., \ ac_{\mathrm{n\text{-}1}} : \text{“}valor\text{”}, ac_{\mathrm{n}} : \text{“}valor\text{”}\} \end{cases} \tag{4.1}$$

De acordo com o domínio da aplicação, são estabelecidas faixas com possíveis valores que podem ser atribuídos ao respectivo atributo. A tabela 2 mostra uma possível descrição das entidades e faixas de valores para os atributos.

Tabela 2 – Atributos em Requisição (Próprio autor)

Entidade	**Atributo**	**Faixa de Valores**
Sujeito	Nome	indiferente
	Tipo	*{ morador, visitante, administrador, colaborador }*
	Idade	*{18 ou mais }*
	Departamento	*{manutenção, hidráulica, elétrico, térmico}*
	Síndico	*{verdadeiro, falso}*
Objeto	Nome	*{aquecedor, lampada, termômetro, sensor de presença}*
	Localização	{*lazer*, *área de convivência*, *recepção* , quarto 1}
	Fabricante	{ *fabricante01, fabricante02, fabricante03* }
	Categoria	*{térmico, eletrodoméstico, iluminação }*
Operação	Tipo	{*leitura, escrita, alteração, exclusão* }
Contexto	Localização	{*lazer*, *área de convivência*, *recepção* }
	Dia da semana	{ *dom, seg, ter, qua, qui, sex, sáb* }
	Tempo de uso	{ *24 horas* }
	Ocupação	*{comercial, residencial, museu, governamental}*

Fonte: Próprio Autor

O administrador dos recursos define quais as políticas devem ser aplicadas, especificando as possibilidades de valores dos atributos. Essa especificação estabelece quais atributos caracterizam as entidades, elencando também a faixa de valores que podem ser associados àqueles atributos.

Os atributos então são utilizados para compor as expressões contextuais que definem os modelos das políticas. As expressões **(Ex)** associam e comparam os atributos e seus valores, na requisição de acesso, através de operadores relacionais **($\mathbf{op_r}$)** (“ $>$ ”,“ $<$ ”,“ $>$ ”,“ $\leq$ ”,“ $\geq$ ”,“ $=$ ”,“ $\neq$ ”) . As expressões seguem a estrutura:

$$EX_{\mathrm{i}} = A(S,O,P,C)_{\mathrm{i}} < \boldsymbol{op_r} > \text{“}valor\text{”} \tag{4.2}$$

De acordo com as necessidades de restrições de acesso, a composição de expressões contextuais definem as políticas **P** de controle de acesso, através de operadores lógicos **$\mathbf{op_l}$** (“$\wedge$”,“$\vee$”,“$\neg$”),

Tabela 3 – Conjunto de políticas para cenário de prédios inteligentes

Requisição	Operação	Objeto	Políticas	Permissão
Req01	**Leitura**	**Ar condicionado**	P_1 : Tipo = "*morador*" P_2 : Tipo = "*visitante*" $\wedge$ *localizacao* = "*recepcao*" P_2 : síndico = *True* $\vee$ *Departamento* = "*manutencao*" $\wedge$ *horario* = "6 *s* 18"	✔ ✔ ✔
Req02	**Acender**	**Lâmpada**	P_4 : Tipo = ("*morador*") $\wedge$ *localizacao* = "*casa*" P_5 : *Tipo* = "*visitante*" $\wedge$*localizacao* = "*recepcao*" $\wedge$ idade $\geq$ 18 anos	✔ ✔
Req03	**Notificação**	**Alarme Incêndio**	P_6 : Tipo = qualquer $\wedge$*localizacao* = "*nopredio*" $\wedge$ *durao* = 30*min* P_7 : *Tipo* = "*administrador*" $\vee$: *sindico* = "*true*"	✔ ✔

Fonte: Próprio Autor

resultando em permissão ou proibição.

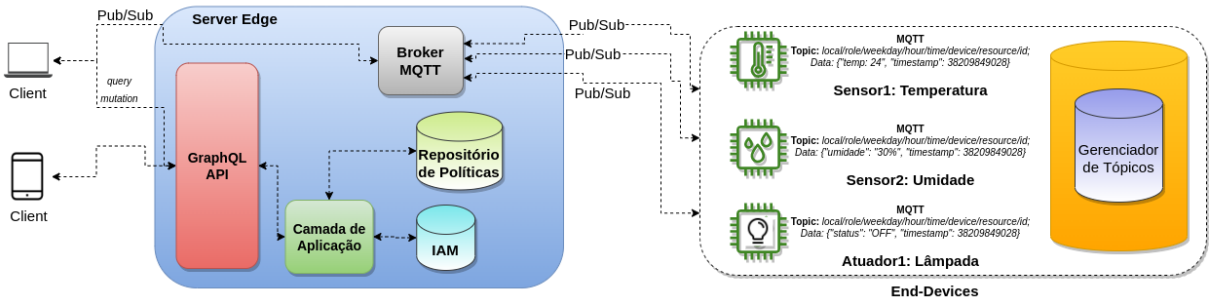
$$P = Ex_{\mathrm{i}} < \boldsymbol{op_l} > Ex_{(\mathrm{i+1})} < \boldsymbol{op_l} > Ex_{(\mathrm{i+2})} < \boldsymbol{op_l} > \ \ldots Ex_{(\mathrm{n\text{-}2})} < \boldsymbol{op_l} > Ex_{(\mathrm{n\text{-}1})} < \boldsymbol{op_l} > Ex_{\mathrm{n}} \quad (4.3)$$

Considerando um cenário de prédios inteligentes como exemplo, a tabela 3 mostra um conjunto de políticas que podem ser especificadas seguindo o modelo descrito anteriormente.

4.4 Arquitetura

O sistema de controle de acesso dessa proposta é baseado em atributos de contexto e utiliza a tecnologia *edge computing*. O gerenciamento das requisições e dos dados é executado através da linguagem de consulta GraphQL. A figura 13 mostra os quatro elementos básicos que compõe o modelo: Cliente, *Server Edge*, Broker *Message Queuing Telemetry Transport* / distância lateral (*MQTT*) e End Devices.

Figura 13 – Arquitetura do modelo Proposta.



Fonte: Próprio Autor

4.4.1 Cliente

O **cliente** é a entidade que possui e manipula o dispositivo/aplicação cliente e que envia a requisição de acesso a um determinado recurso dentro do ambiente IoT. De acordo com a linguagem de consulta GraphQL, a requisição pode ser de dois tipos: *query* que realiza uma consulta de dados ou *mutation* que executa ações de alteração nos dados (por exemplo inserção, atualização ou exclusão). Na arquitetura apresentada, esse tipo de tecnologia também pode efetuar conexões do tipo *websocket* por meio de *subscriptions e publish*, às quais permitem a comunicação bidirecional, notificações em alterações e percepção em tempo real, com objetivo de gerenciar a disponibilidade dos recursos.

4.4.2 Server Edge

O ***Server Edge*** é o elemento responsável por receber as requisições e prepará-las para acesso aos recursos. Essa entidade é composta por 3 camadas: A **camada de aplicação**, que gerencia as requisições em um *endpoint - GraphQL API*, sendo responsável por preparar e aplicar as políticas de acesso. A utilização do GraphQL determina um único ponto para o acesso aos recursos, fornecendo todas as informações necessárias para as aplicações cliente. Essa configuração sustenta também a redução de ocorrências de *overfetching* (quando há uma consulta de dados além dos necessários), e *underfetching* (quando a requisição retorna menos dados que o essencial), características essas presentes em arquiteturas do tipo *Representational State Transfer* / distância lateral (*REST*). O **Broker MQTT** realiza o acesso aos recursos via protocolo MQTT, orquestrando a comunicação de acordo com o controle de acesso, por meio de tópicos.

Na ocasião de utilização do *Server Edge*, há a possibilidade de uso de um elemento para verificação de identidades **IAM**, que realiza o processo de autenticação utilizando decodificação de token JWT (*JSON web Token*), o qual é estabelecido em (JONES *et al.*, 2015) como sendo um método seguro para representação e compartilhamento de informações entre duas partes, preservando a autenticidade entre a interação.

Repositório de políticas é ponto que armazena e gerencia as políticas especificadas, sendo gerenciado por meio de um banco de dados NoSQL, sendo escolhido por possuir alto desempenho e escalabilidade, agrupando estruturas de dados em formato JSON em semântica de fácil leitura e interpretação, conferindo também flexibilidade aos registros.

4.4.2.1 End Devices

Representam os dispositivos IoT, que podem ser sensores ou atuadores. Esses dispositivos possuem pouca capacidade de processamento, armazenamento e, possivelmente, restrições de energia, onde alguns funcionam através de baterias duráveis. Estes são recursos possíveis para acesso aps usuários.

4.5 Diagrama de Sequência e Algoritmo

A ilustração da sequência de solicitação de acesso a um recurso, por um cliente, mostrada abaixo, na figura 14. O cliente é o componente que, de acordo com seu interesse, solicita acesso ao recurso, enviando suas credenciais de identidade (login e senha) para o endpoint GraphQL API. Para verificar as credenciais, a Camada de Aplicação analisa a estrutura dos dados e realiza a consulta ao IAM. Este componente, de posse das credenciais, retorna os dados para a Camada de Aplicação que realiza a validação do usuário (autenticação). Nesse momento, caso as credenciais sejam válidas, o token JWT é criado e enviando-o para o cliente. Caso contrário, o acesso será negado.

De posse do token JWT e da identificação do recurso, o cliente pode realizar a requisição de acesso, junto à camada de aplicação, que realiza a verificação e validação do mesmo para gerenciar o acesso. Dentre as possibilidades de certificação, é possível averiguar se o tempo do token está expirado. Em caso de token validado, as políticas de acesso são consultadas junto ao repositório de políticas e, este último, realiza o reconhecimento de direitos, permitindo a realização da operação ou negando o acesso nesse ponto. O tópico é armazenado e usado para inscrição e publicação de dados.

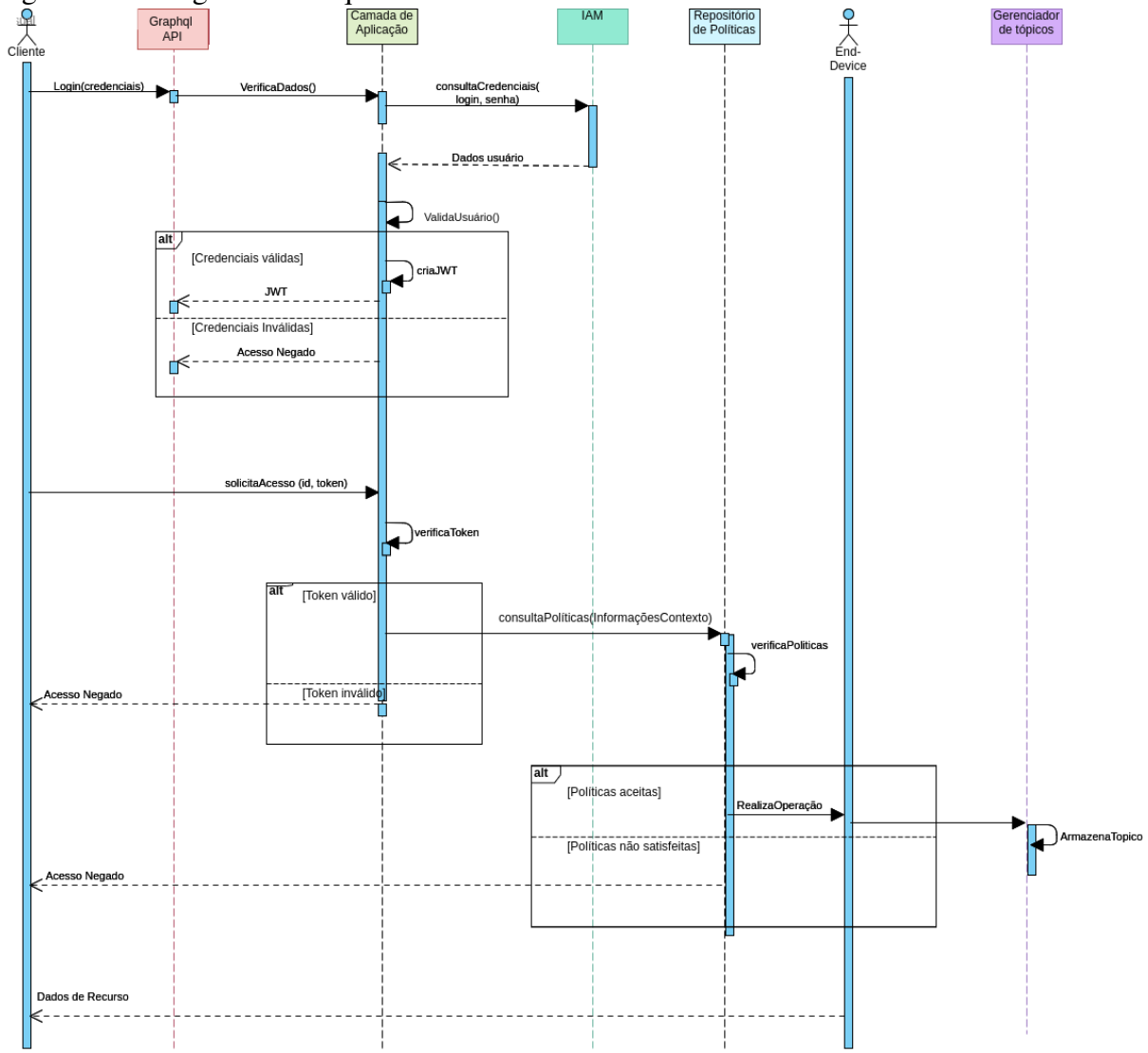
4.5.1 Implementação

A implementação é baseada na arquitetura proposta na figura 13. A construção do *server edge* é feita em linguagem JavaScript no ambiente de execução *nodejs*[1]. A estrutura para a utilização do *schema* GraphQL é apoiada na biblioteca *graphql-yoga*[2].

Para a camada de persistência, foi usado o MongoDB[3]. Este, por sua vez, é um banco de dados multiplataforma orientado a documentos que oferece alto desempenho e fácil

[1] https://nodejs.org/
[2] https://github.com/dotansimha/graphql-yoga
[3] https://www.mongodb.com/

Figura 14 – Diagrama de sequência.



Fonte: Próprio Autor

escalabilidade.

Para o broker central de mensagens, é utilizada o *Eclipse Mosquitto* [4], que habilita as conexões via protocolo MQTT, sendo leve, de alto desempenho e com código aberto.

Uma possibilidade de utilização de mecanismos de autenticação é proposta por esse trabalho usando o token *JSON web Token* (JWT), sendo seu uso restrito à perspectiva de gerenciamento de identidades para acesso ao recurso. O cliente envia suas credenciais (*login* e senha) para consulta e confirmação e, caso sejam aceitas, o token é então construído e devolvido para o usuário. Do contrário, o acesso é negado, como mostra o algoritmo 1.

O procedimento para autorização é descrito no algoritmo 2. Este, aponta como o mecanismo funciona a partir da requisição realizada pelo cliente. A mensagem de requisição é

[4] https://mosquitto.org/

Algoritmo 1: Autenticação

```
Entrada: credenciais (login e senha)
início
    Recebe credenciais
    Consulta credenciais no IAM
    if credenciaisValidas() then
        criaJWT()
        retorna o JWT para o cliente
    else
        retorna "Acesso negado"
fim
```

Fonte: Próprio Autor

Algoritmo 2: Autorização

```
Entrada: (id, token)
início
    verifica token
    nItensPolíticas <- 0
    if token==valido then
        recupera políticas no repositório
        repeat
            if politica.item[n] == contexto.item[n] or politica.item[n] == null then
                count++;
                if (count == nItensPolíticas) then
                    realiza operação
                    insere tópico em repositório
                    se inscreve no tópico
        until todas as políticas;
    else
        retorna “Acesso Negado”
    if topicoExpirado then
        retorna “Acesso Negado”
fim
```

Fonte: Próprio Autor

composta pelo tipo de operação (*query ou mutation*), carregando como parâmetro a identificação do recurso alvo e o token JWT em seu cabeçalho. Ao recebê-la, é iniciado o processo de verificação de assinatura do token, certificando a identidade do usuário. Nesse momento, caso o token não seja validado ou tenha o tempo de utilização expirado, o cliente terá seus direitos revogados. Ao passo que o token é reconhecido, há a verificação de políticas disponíveis para o recurso solicitado, através de uma varredura e correlação entre as unidades de contexto e os itens das políticas. Cada correspondência entre estes termos é contada e caso atinja o valor determinado por um administrador (dez itens nesse trabalho), a solicitação é aceita e a operação pode ser realizada no *end-device*. Em caso contrário, o acesso é negado.

4.6 Conclusões

Neste capítulo, foram apresentados os componentes básicos que guiaram o desenvolvimento do artefato aplicação do presente estudo. Foram mostrados os elementos da arquitetura, um cenário de utilização, diagrama de sequência e algoritmos usados. O conceito de prédios inteligentes e a explanação de possíveis políticas para o acesso a recursos serviram de guia para a definição do modelo e especificação de políticas.

A arquitetura apresenta os componentes relativos ao quadro de comunicação relacionado ao estudo. Foi demonstrado como um cliente se comunica com o servidor da borda, por meio de uma API Graphql, realizando o manejo de acesso através das políticas inseridas no repositório oportuno. Além do mais, a arquitetura mostrada aponta a utilização de um broker que realiza a orquestração do acesso, via protocolo MQTT e as tecnologias usadas para implementação, como a linguagem de programação, bibliotecas e o banco de dados usado no repositório.

As interações que ocorrem no ambiente são demonstradas por meio de um diagrama de sequência, o qual apresenta os diálogos entre os mecanismos, apresentando no momentos em que o acesso é negado e permitido, juntamente com os algoritmos desenvolvidos.

5 AVALIAÇÃO DE DESEMPENHO

O atual capítulo expõe a avaliação e análise da performance da abordagem de estudo, elencando o desenho dos experimentos, os dispositivos e tecnologias utilizadas.

Este experimento tem como objetivo apresentar o desempenho da arquitetura aqui proposta, entendendo como a disponibilidade é afetada quando o mecanismo de controle de acesso é utilizado.

Como ferramenta para produção de carga de trabalho, foi utilizado o Apache JMeter[1], o qual é um aplicativo que realiza a produção e carregamento de dados para execução de funções diversas, analisando o desempenho do sistema.

Para verificar as características do mecanismo de controle de acesso, foi utilizado o collectl[2], que se trata de um utilitário leve de coleta de dados estatísticos em relação ao sistema operacional utilizado, como por exemplo: utilização de CPU, memória, disco, dentre outros.

Em relação ao hardware, foram utilizadas 3 máquinas: A primeira representa o cliente, a qual possui sistema operacional Ubuntu 21.10 com 64 bits, processador 11º Ger Intel® Core™ i7-1165G7 @ 2.80GHz × 8 com 7,5GiB de memória; a segunda com o papel de servidor com *server edge*, sendo um placa Raspberry Pi 4 modelo B[3] com sistema operacional Raspberry Pi OS de 64 bits, com processador quad-core Cortex-A72, e memória de 2GB; e a terceira que representa o *end-device* sendo o Módulo WiFi ESP8266 NodeMcu ESP-12, que contém um chip ESP8266 com microprocessador Tensilica Xtensa 32-bit LX106 RISC e 128 KB de RAM e 4 MB de memória Flash para armazenamento de dados e programas (MACHESO *et al.*, 2021)

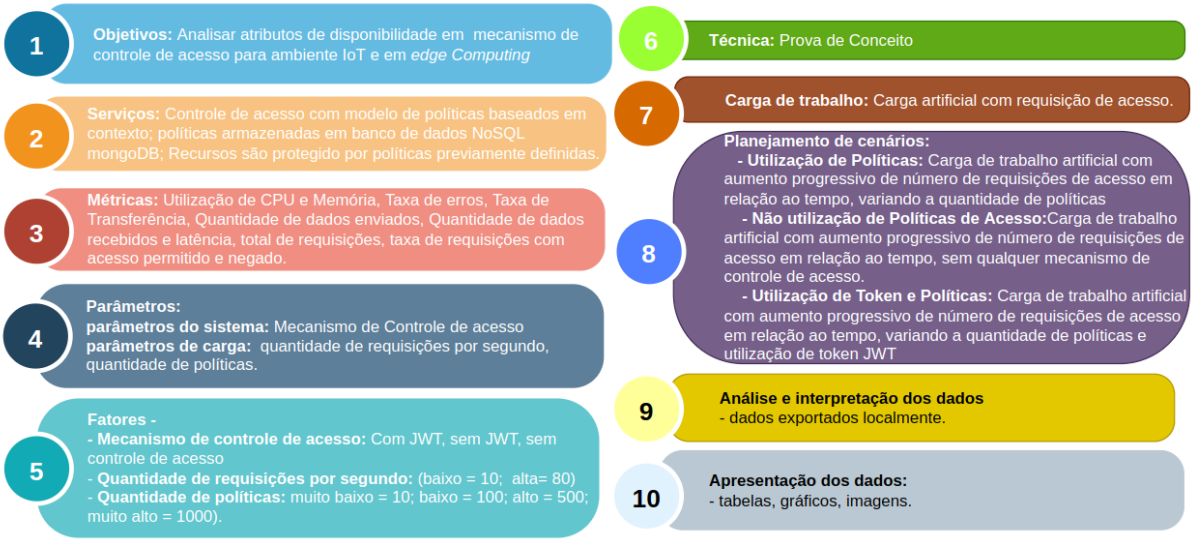
A execução dos experimentos sucedeu-se em três etapas que consistem na observação do comportamento de elementos de confiabilidade no processo de requisição de acesso, sendo a primeira com envio de token e verificação de politicas de permissões; a segunda, sem envio de token, apenas com verificação de politicas de permissões; a terceira, sem nenhum tipo de controle de acesso ao recurso.

A metodologia de avaliação de desempenho foi seguida utilizando elementos propostos por (BUKH, 1992), buscando moderar vieses que por ventura a estrutura do estudo possa conter. A figura 15, especifica os pontos da abordagem.

– **Objetivos:** Estudar como a confiabilidade da prestação de um serviço em IoT é assegurada ao usuário, por meio da verificação do atributo de disponibilidade usando mecanismo de

[1] https://jmeter.apache.org/
[2] https://collectl.sourceforge.net/
[3] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

Figura 15 – Estrutura da Avaliação de Desempenho (BUKH, 1992).



Fonte: Próprio Autor

controle de acesso;

– **Serviços:** o serviço a ser analisado trata de controle de acesso com modelo de políticas baseadas em contexto para autorização. Nesse sentido, é utilizado banco de dados NoSQL mongoDB e o ponto de solicitação de acesso refere-se a um endpoint GraphQL.
– **Métricas:** para analisar a continuidade do serviço, dados de utilização do CPU e Memória, taxa de erros, quantidade de dados recebidos e enviados, *throughput* e latência são verificados. Se os valores de desempenho computacional no *server-edge* forem intensos, a continuidade do serviço de acesso pode ser prejudicada. Isso também acontece quando a latência e *throughput* são acentuados, avariando a disponibilidade do mecanismo de acesso à recursos ;
– **Parâmetros:** em ambiente IoT, os elementos possuem poucos recursos computacionais, desta forma, em relação ao parâmetro de sistema, a utilização ou não de mecanismo de controle de acesso serve de comparativo para verificação de sobrecarga na comunicação e hardware envolvido, o que pode reduzir ou mesmo interromper o serviço; para parâmetros de carga, a quantidade de requisições variando, amplia a possibilidade de estudos em sobrecarga, bem como a presença de uma grande quantidade de políticas pode influenciar no desempenho computacional do sistema, assim como uma pequena quantidade pode não ser suficiente para o refinamento de acesso à recursos. A possibilidade de utilização de mecanismo de autenticação também pode afetar a performance;
– **Fatores:** A variação de fatores busca analisar o comportamento da disponibilidade e

continuidade do serviço. Os valores para tal, foram definidos de acordo com o panorama em que há pouca ou muita sobrecarga no serviço, acarretando em falhas ou interrupção do mesmo. Para a análise do parâmetro "mecanismo de controle de acesso", associamos os fatores de utilização ou não de controle de acesso e aplicação de token JWT e políticas de acesso. A quantidade de requisições por segundo varia continuamente tendo como valores mínimo (baixa) de 10 req/seg e máximo (alta) 80 req/seg. Outro fator que pode auxiliar a verificar o parâmetro de carga é a quantidade de políticas. Nesse estudo, foram utilizadas uma quantidade muito baixa (10), baixa (100), alta (500) e muito alta (1000);

- **Técnica:** Executamos os experimentos através de uma prova de conceito com os elementos descritos nas seções anteriores;
- **Carga de trabalho:** A carga de trabalho trata de uma requisição através de uma operação de *mutation* e envio de parâmetro com a identificação do recurso a ser acessado. O parâmetro com a identificação do recurso varia progressivamente para verificação em diferentes cenários;
- **Planejamento de cenários:** Considerando o cenário motivador descrito na seção 4.2, foram projetados cenários, considerando o número de requisições e as diferentes operações do controle de acesso;
- **Análise e interpretação dos dados:** Os dados obtidos através das ferramentas definidas para o estudo (*JMeter* e *collectl*) são exportados localmente em arquivos em extensão .CSV para leitura e compilação posterior. A relação das métricas obtidas possibilitam a identificação em que o cenário apresenta confiabilidade;
- **Apresentação dos dados:** Os dados obtidos são compilados em tabelas e gráficos que sistematizam os resultados dos experimentos.

5.1 Autorização

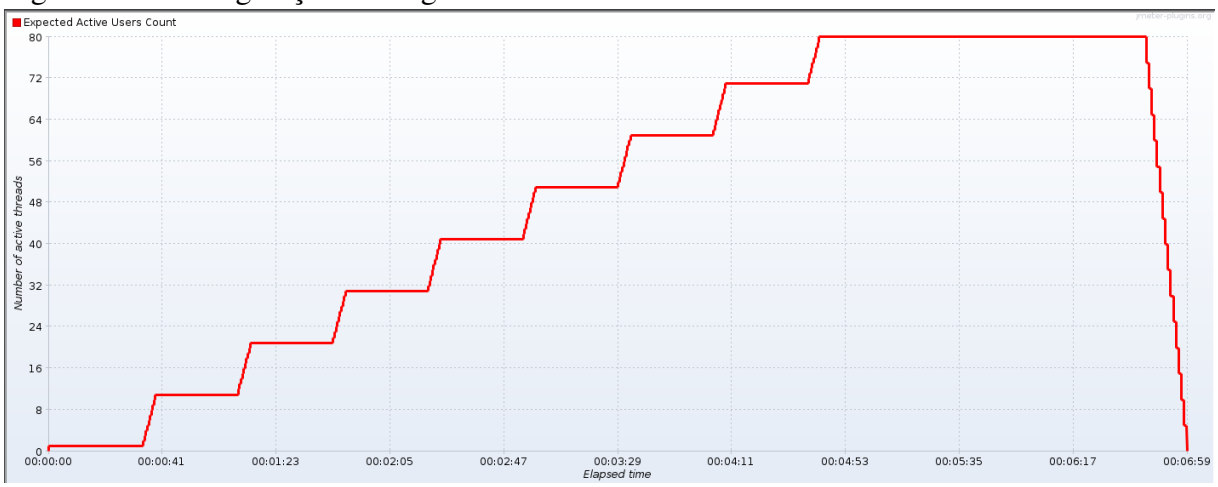
Inicialmente, estabeleceu-se um modelo para a realização das requisições, enquanto carga de trabalho, haja visto que essas, são parâmetros para o estudo. Dessa forma, foram determinados valores mínimos de 10 requisições por segundo (baixa quantidade) e 80 requisições por segundo (alta quantidade), seguindo um modelo de progressão crescente entre essas regras. Esse padrão foi adotado visando uma simulação em um ambiente real, segundo o cenário descrito na seção 4.2.

A ferramenta utilizada foi o Apache JMeter, com a aplicação do *plugin jp@gc* -

Stepping Thread Group[4] que utiliza um algoritmo de agendamento e concatenamento de *threads*. Dentre outras características, esse *plugin* oferece: um gráfico com visualização de carga estimada; combinação de grupo de *threads* para várias atividades, podendo aumentar a carga por porções (usuários); e tempo de espera configurável após todos os *threads* serem iniciados. As requisições foram configuradas de acordo com a figura 16, a saber:

- Inicia o teste com 1 *thread*;
- Após os 30 primeiros segundos, são adicionadas 9 *threads*. Em seguida, a cada 30 segundos, são adicionadas 10 novas *threads*;
- A carga de requisições é mantida por 30 segundos;
- Em carga máxima (80 *threads*), as requisições são mantidas por 2 minutos (120 segundos);
- A carga útil é amortizada com a diminuição de 10 *threads* a cada 1 segundo.

Figura 16 – Configuração da carga de trabalho.

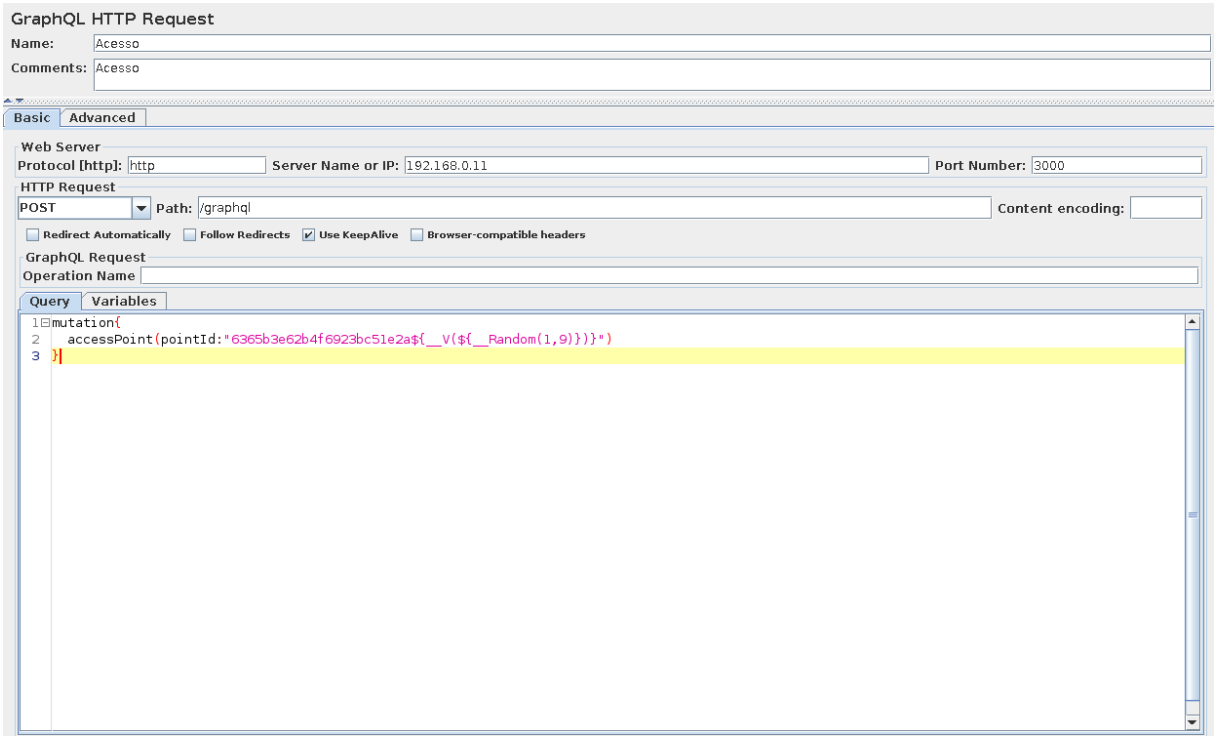


Fonte: Próprio Autor

A requisição é enviada através de protocolo *Hypertext Transfer Protocol* / falha de blindagem (HTTP), usando GraphQL Request, onde, no corpo da mensagem, é informada a referência de recurso (número de identificação), usando uma operação de *mutation*, denominada de “accessPoint”. Enquanto parâmetro, a referência do recurso utiliza uma função que gera um número aleatório. Dessa forma, é possível verificar se a requisição, de acordo com a referência do recurso e a suas políticas de acesso, possui direitos de acesso ou não. A figura 17 mostra o formato da requisição, dentro do Apache JMeter.

[4] https://jmeter-plugins.org/wiki/SteppingThreadGroup/

Figura 17 – Formato da requisição.



Fonte: Próprio Autor

5.1.1 Resultados

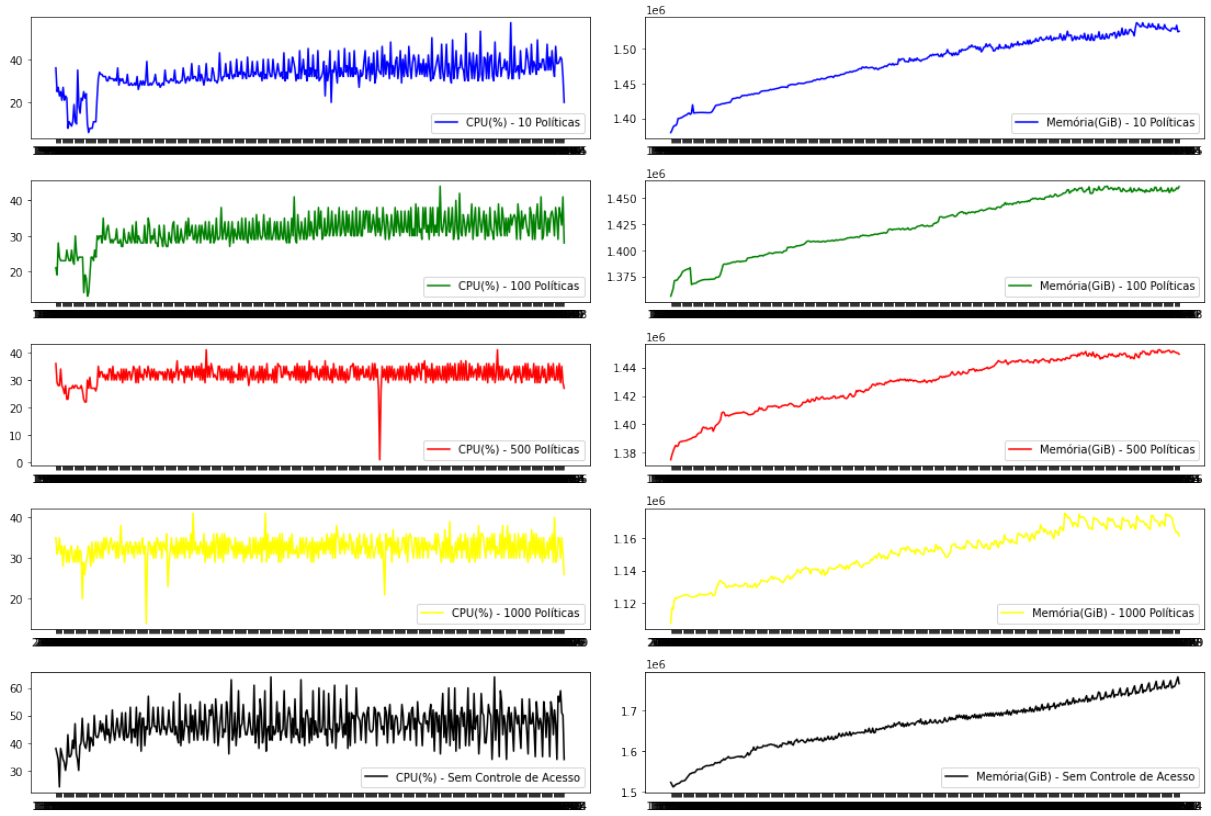
5.1.1.1 Hardware

Para estudo da disponibilidade e continuidade do serviço, foi necessário uma avaliação do desempenho computacional. Nesse sentido, a performance de utilização de CPU e memória foram do *broker* (*server edge*). As figuras 18 e 19, apresentam o comportamento do uso de recursos computacionais do *server edge*.

O comportamento do uso de CPU e Memória foi estudado para verificar uma possível sobrecarga no desempenho computacional. Podemos verificar que em todos os cenários em que é utilizado controle de acesso, há uma redução na taxa de utilização de CPU. Essa característica melhora o desempenho e disponibilidade do serviço de acesso ao recurso, possibilitando a alocação de CPU para outros tipos de aplicações que por ventura possam existir no ecossistema IoT. Também foi apurado uma redução na utilização de Memória quando comparados cenários com e sem controle de acesso.

Dentre os cenários do estudo com controle de acesso e não uso de autenticação (Sem JWT), o que obteve melhores métricas em relação ao uso de CPU foi o cenário que possui 100 políticas, com média de utilização de CPU 31,30%. O melhor resultado para memória foi o que

Figura 18 – Comportamento de Utilização de CPU e Memória em Controle de Acesso sem JWT.



Fonte: Próprio Autor

estabelece 1000 políticas de acesso, com máximo de utilização registrado 1175692 GiB. Pode-se verificar que a quantidade de políticas influencia na quantidade de Memória.

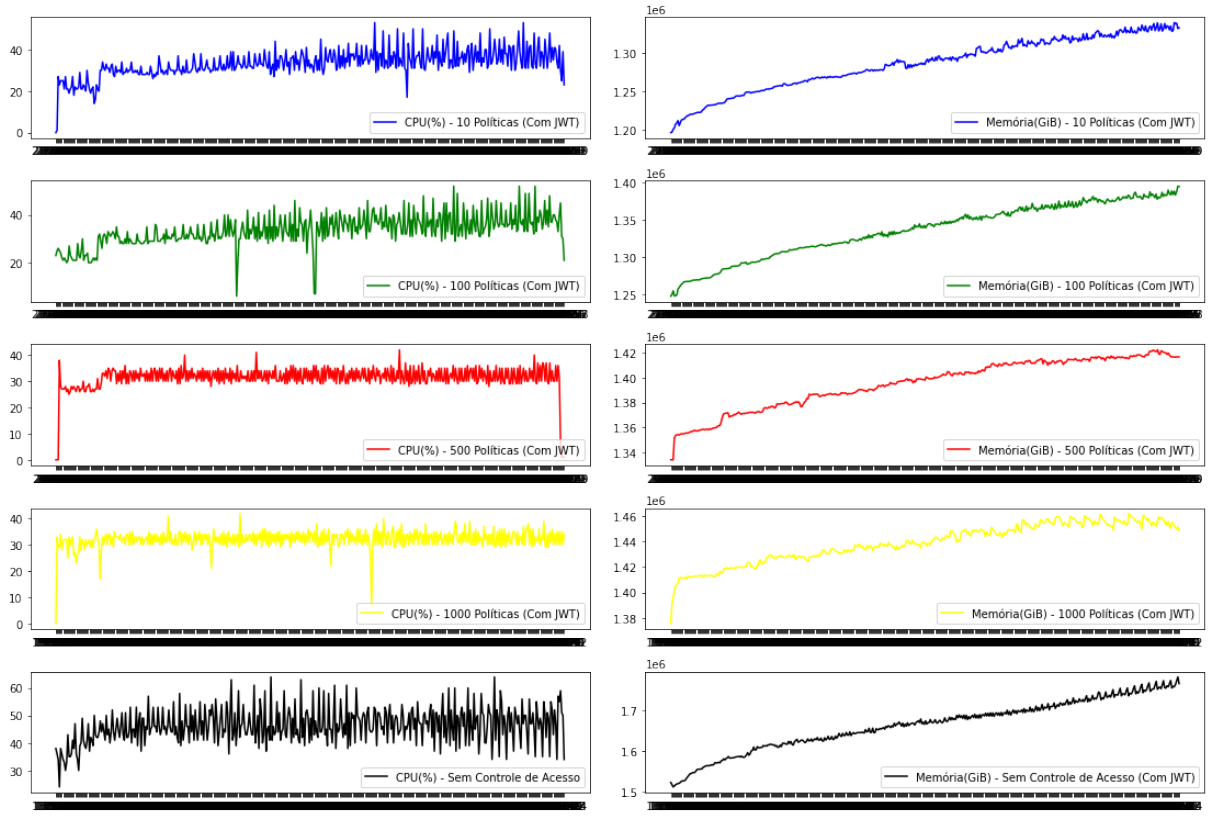
Para os cenários em que são utilizados tanto o mecanismo de autenticação (com JWT) e autorização, foi constado, em média, como melhor resultado ao uso de CPU, cenário em que há 500 políticas e para Memória, com 10 políticas (1339376GiB), podendo-se observar que, quando o sistema não possui controle de acesso, há uma grande utilização de CPU e Memória, em relação aos outros cenários, como mostra a tabela 4.

Foi aferido que a quantidade de políticas não influencia no desempenho de uso de CPU, mas influencia diretamente no uso de Memória. Um novo desenho de ensaio em que se tenha uma maior quantidade de execução de experimentos, poderia possibilitar uma melhor análise estatística.

5.1.2 Rede

A seguinte seção apresenta os resultados referentes à disponibilidade e continuidade do serviço, com relação à conectividade. Métricas como latência, *througput* e taxa de erros foram coletadas durante a realização dos experimentos.

Figura 19 – Comportamento de Utilização de CPU e Memória em Controle de Acesso com JWT.



Fonte: Próprio Autor

Tabela 4 – Valores de taxa de utilização média de CPU e máximos valores registrados de Memória.

Cenário	**Média CPU (%)**	**Máximo Mem (GiB)**
10 Políticas sem JWT	33.15	1537388
100 Políticas sem JWT	31.30	1461492
500 Políticas sem JWT	31.88	1452504
1000 Políticas sem JWT	32.49	1175692
10 Políticas com JWT	33.24	1339376
100 Políticas com JWT	33.48	1395576
500 Políticas com JWT	31.35	1422392
1000 Políticas com JWT	32.14	1462068
Sem Controle de Acesso	46.25	1781872

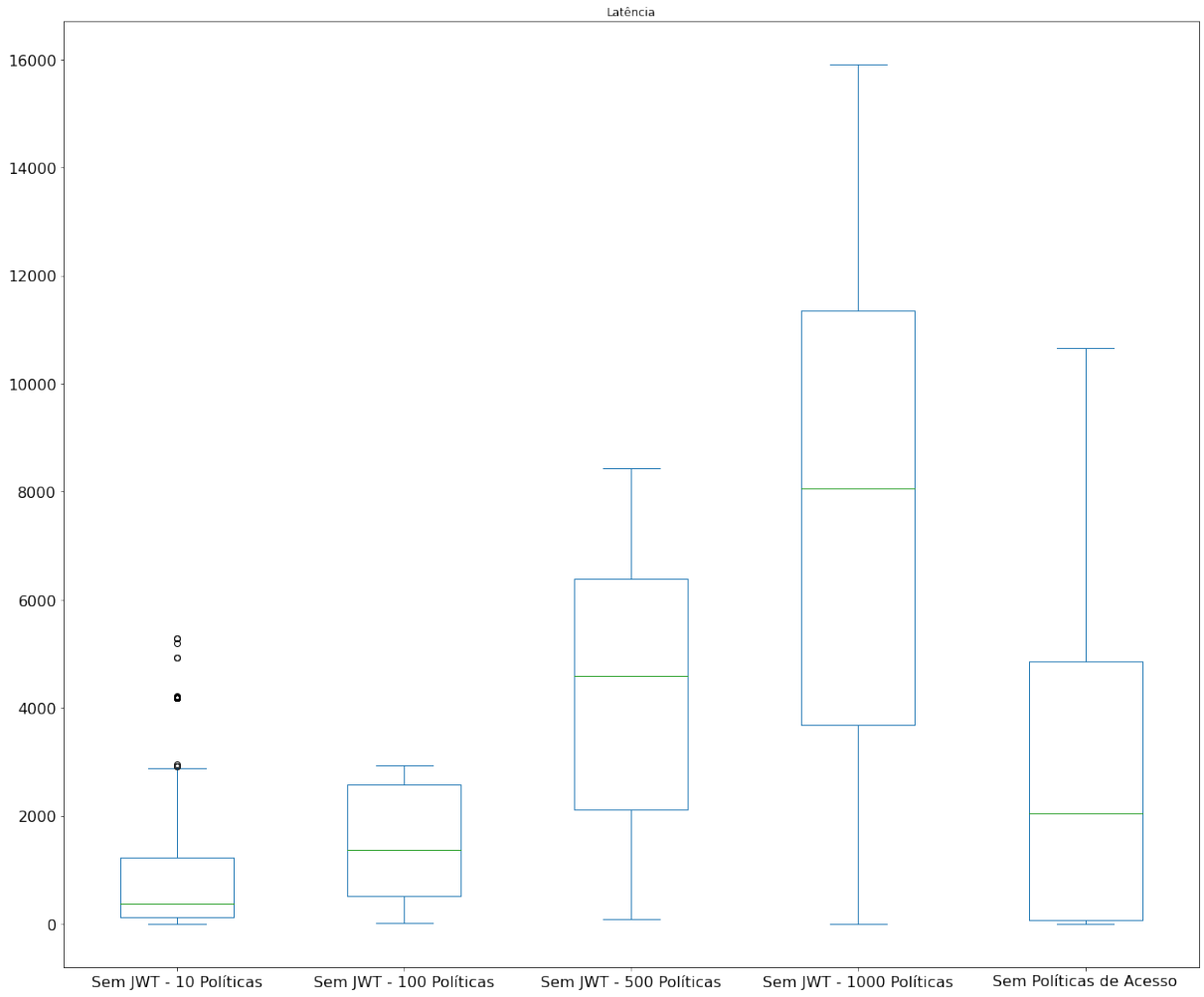
Fonte: Próprio Autor

A figura 20 mostra uma análise estatística dos valores de latência para as requisições de acesso a um *end-device*, para os cenários em que há controle de acesso sem autenticação (sem jwt) e com autenticação. Os resultados mostram que, em média, sem controle de acesso, uma maior latência pode ser verificada, em relação aos cenários que apresentam 10 ou 100 políticas de acesso, o que não ocorre com 500 e 1000 políticas. Esses dados constatam que a quantidade de políticas podem influenciar os níveis de latência, e a alta quantidade pode diminuir

a confiabilidade do serviço.

O tempo que o *server edge* leva para realizar a busca pela política de acesso pode influenciar na latência e, nesse caso, intervir no tempo de execução. Considerando a conjuntura de funcionamento do sistema em uma situação real (vide seção 4.2), entendemos que a quantidade de políticas de acesso (100) é suficiente para melhorar os níveis de confiabilidade.

Figura 20 – Amostra de Latências sem JWT.

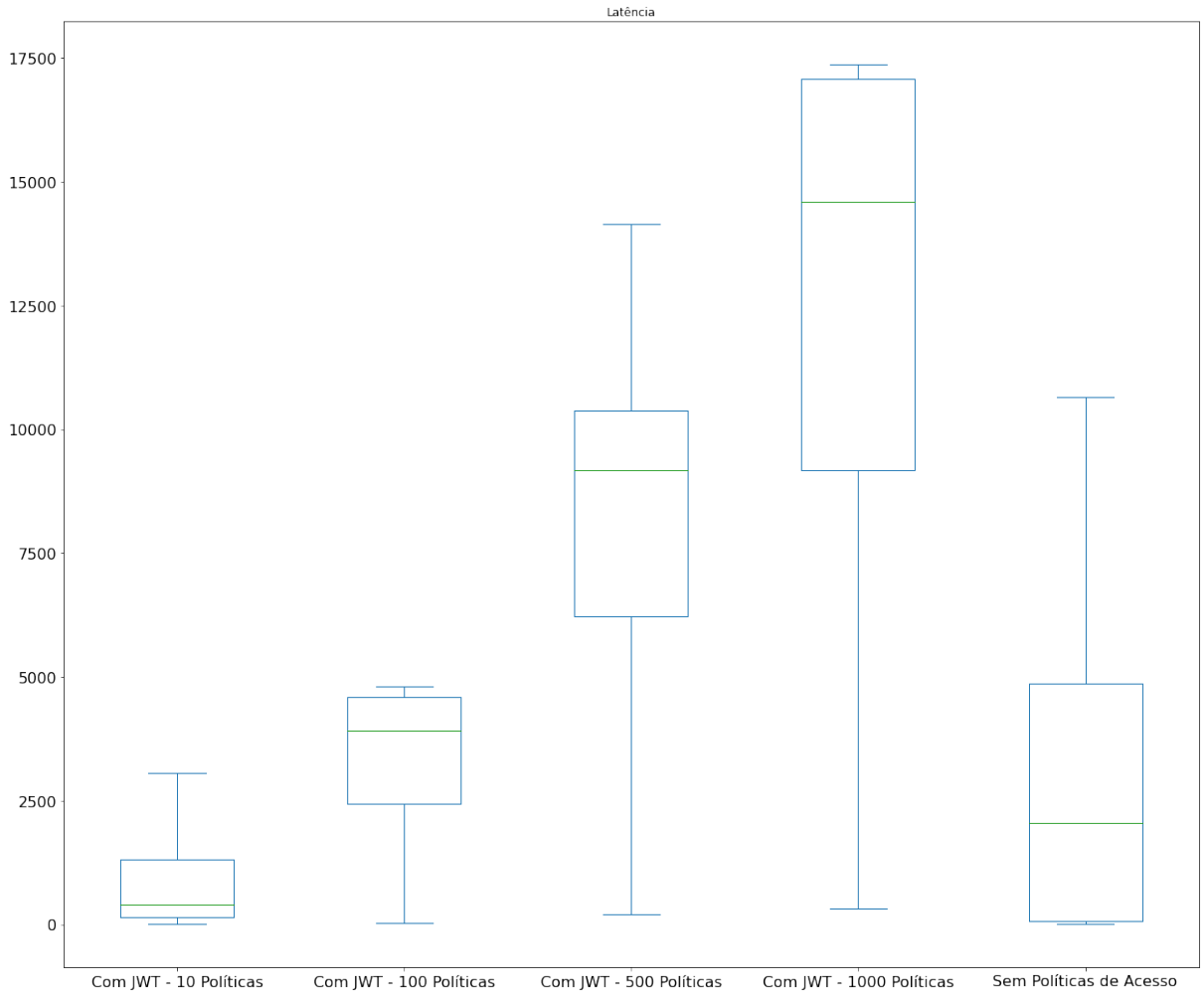


Fonte: Próprio Autor

Para os cenários em que há a utilização de mecanismo de autenticação via JWT, como mostra a figura 21, obteve-se valores de latência, em média menores para o cenário em que são utilizadas 10 políticas de acesso. Nos outros cenários, há uma diferença relevante, quando há grande quantidade de políticas. Em hipótese de aplicação de 100 políticas, valores consideráveis obtidos refletem uma boa porção dos valores registrados para o cenário sem políticas de acesso (terceiro quartil). É plausível perceber que, mesmo com uma baixa quantidade de políticas e o emprego de autenticação, a confiabilidade do serviço é melhorada em relação a não utilização de

nenhum controle de acesso.

Figura 21 – Amostra de Latências com JWT.



Fonte: Próprio Autor

Para (SUCIU *et al.*, 2018), as análises de confiabilidade são estimadas através de indicadores quantitativos, sendo um deles *throughput*. Este se define como a quantidade de dados recebidos com sucesso, em meio a uma solicitação de acesso.

Os valores registrados para *throughput*, assim como dados recebidos e enviados para os nove cenários analisados são apresentados na tabela 5. É possível verificar maiores valores para *throughput* quando há poucas políticas de acesso, mesmo com a utilização de instrumento de autenticação, quando comparados ao cenário "Sem Controle de Acesso", apontando os primeiros como aprimoramento de confiabilidade. Em situação em que ocorre um grande número de políticas, esse fato mostrou uma redução da taxa de transferência, situação que pode ocasionar perda de pacotes e redução da disponibilidade.

No tocante aos dados recebidos e enviados, observou-se que nos mesmos cenários

Tabela 5 – Valores registrados para *throughput*, dados recebidos e enviados

Cenário	Throughput (Tran/seg)	Dados Recebidos (KB/seg)	Dados Enviados (KB/seg)
10 Políticas sem JWT	66.94	18.00	19.15
100 Políticas sem JWT	43.51	11.70	12.45
500 Políticas sem JWT	12.36	3.32	3.54
1000 Políticas sem JWT	6.50	1.75	1.86
10 Políticas com JWT	76.63	20.60	52.01
100 Políticas com JWT	76.53	20.58	51.94
500 Políticas com JWT	12.17	3.27	8.26
1000 Políticas com JWT	5.96	1.60	4.04
Sem Controle de Acesso	28.82	7.82	8.22

Fonte: Próprio Autor

em que a propriedade de *Throughput* foi mais elevada também houveram mais transações por segundo, garantindo uma maior quantidade de dados para processamento e gerenciamento, fato que confere uma melhora da confiabilidade ao serviço.

Quando considerado o total de requisições efetuadas durante os experimentos, constatou-se que em cenários com controle de acesso e baixo número de políticas, o total de requisições é superior aos dados do cenário sem controle de acesso. No entanto, as requisições permitidas são menores quando o controle de acesso é presente em confronto a sua ausência. Esse fato garante a melhoria da confiabilidade no que diz respeito a segurança, certificando que apenas as requisições que atendem ao menos uma das políticas, estarão aptas à usufruir do recurso solicitado.

Este estudo buscou examinar a taxa de erros em relação ao sucesso em realização de requisição e sucesso no cálculo da permissão de acesso. Não foram registrados erros durante a execução dos experimentos em nenhum cenário analisado, dispensando assim a necessidade de apresentação dos resultados.

Tabela 6 – Quantitativo de requisições permitidas e negadas.

Cenário	Total requisições permitidas	Total requisições negadas	Total requisições
10 Políticas sem JWT	3663	28999	32662
100 Políticas sem JWT	1974	16106	18080
500 Políticas sem JWT	542	4490	5032
1000 Políticas sem JWT	295	2458	2753
10 Políticas com JWT	3598	28431	32029
100 Políticas com JWT	1965	15943	17908
500 Políticas com JWT	581	4539	5120
1000 Políticas com JWT	300	2331	2631
Sem Controle de Acesso	12345	0	12345

Fonte: Próprio Autor

6 CONSIDERAÇÕES FINAIS

Este capítulo expõe as considerações finais da pesquisa realizada, elencando limitações encontradas e apresentando sugestões para futuros trabalhos pertinentes ao tema principal.

A proposta do estudo utilizou uma arquitetura para controle de acesso fundamentada em políticas baseadas em contexto. Esta consistiu na utilização da borda da rede (*edge computing*) por meio do uso de um *gateway* que funciona como *broker* e controlador de acesso. As políticas foram definidas por meio de expressões contextuais e gerenciadas em banco de dados NoSQL. Nesta mesma arquitetura, o usuário cliente solicita o acesso por meio de uma requisição via GraphQL, informando a referência do recurso a ser atendido e, após a análise da requisição, a mensagem é entregue aos destinatários via protocolo MQTT.

O objetivo fundamental da pesquisa consistiu na elaboração e análise quantitativa e qualitativa de uma arquitetura de controle de acesso baseado em *edge computing*, em ambientes IoT, com vistas no melhoramento da disponibilidade. Sendo assim, consideramos que o objetivo central foi alcançado, pois são apresentados aspectos referentes ao conceito de disponibilidade sob a óptica de uma abordagem de avaliação de desempenho estruturada, expondo cenários em que a presença de um mecanismo de controle de acesso melhora este atributo.

Constatou-se que a utilização de controle de acesso melhora a disponibilidade e continuidade do serviço. Porém, o emprego de modelos de políticas que sejam aplicáveis ao ambiente específico de IoT é de difícil gerenciamento, pois deve haver um balanceamento entre a complexidade do modelo, quantidade de opções e possibilidade de tornar o ambiente mais seguro por meios de garantias de autenticidade.

A aplicação de poucas políticas de acesso são suficientes para a melhoria da disponibilidade, que se confirma quando analisamos métricas de latência, uso de CPU e Memória e taxas de transferência. Quando esse número é expandido, os resultados mostram que o efeito oposto acontece, agravando a comunicação. Além disso, a presença de um aparato que gerencie a autenticação não interfere consideravelmente nos números.

Os objetivos complementares trataram da investigação de requisitos específicos para aplicações de controle de acesso em ambientes IoT, sendo atingidos por meio de pesquisa bibliográfica que abordou os trabalhos que descrevem tais requisitos, estando retratados no capítulo 2. Os objetivos de elaboração, implementação e análise de desempenho de um mecanismo de controle de acesso, guiado por paradigma IoT, foram atingidos aplicando técnica de modelo de políticas com expressões contextuais e tecnologias de comunicação de dados via GraphQL e

protocolos de comunicação específicos para ambientes IoT, no caso MQTT. Esses últimos são mostrados nos capítulos 4 e 5.

Em um panorama de aplicação de mecanismo de controle de acesso baseado em expressões de contexto para prédios inteligentes, o qual esse trabalho contemplou, assimilou-se que uma quantidade de políticas de acesso (100), é razoável para captar níveis de confiabilidade aceitáveis no que tange limitação de uso de recursos computacionais e vazão de dados.

A concepção do paradigma de IoT ser multidisciplinar e heterogênea, torna desafiador o problema de garantir a confiabilidade em ambientes que utilizam tal tecnologia. Para tanto, em decorrência da demanda de utilização de dispositivos IoT na sociedade atual e as estimativas de crescimento, torna-se fundamental o fornecimento de serviços eficientes e confiáveis (provimento de recursos quando necessário, pelo tempo necessário e para quem de direito).

6.1 Limitações

As limitações deste trabalho de mestrado encontram-se na escolha dos elementos para controle de acesso e no desenho dos experimentos, que são listados a seguir:

- **Único mecanismo de autenticação**: o mecanismo de controle de acesso adotou a possibilidade de apenas um mecanismo de autenticação (token JWT), desconsiderando outras técnicas, como: biometria, OAuth2.0 ou mesmo gerenciamento de chaves via OSCORE que implicariam na realização de um número maior de experimentos, prejudicando a qualidade das análises estatísticas, considerando o tempo para a execução deste estudo;
- **Escolha dos Protocolos**: Outros protocolos de comunicação projetados para ambientes com restrições computacionais, como o CoAP e XMPP poderiam ter sido discutidos, no entanto a abordagem resultaria em uma vasta quantidade de experimentos, inviáveis para o tempo desta pesquisa;
- **Análise de consumo energético**: A melhoria da confiabilidade está relacionada à redução do consumo energético. Devido a falta de ferramenta adequada, essa métrica não foi adotada para avaliação de desempenho;
- **Realização de experimentos em ambientes reais**: este trabalho considera que uma das contribuições é a avaliação de desempenho por meio de prova de conceito. Uma análise mais realista poderia ser realizada com a implementação do controle de acesso em um ambiente real, como um prédio inteligente, porém, essa estrutura é de difícil acesso na região onde o trabalho foi desenvolvido.

6.2 Produção Bibliográfica

No percurso do desenvolvimento desse trabalho, um artigo intitulado **Controle de Acesso na Internet das Coisas baseado em Edge Computing com foco em Confiabilidade** foi produzido e apresentado no VI Workshop de Computação Urbana (CoUrb), no ano de 2022.

6.3 Trabalhos Futuros

Considerando que esta pesquisa trata de uma abordagem geral para estudar atributos de confiabilidade através de controle de acesso em IoT, listo abaixo tópicos que podem ampliar a visão sobre o tema em futuros trabalhos:

- Adição de outras técnicas de autenticação, diferentes modelos de políticas, ou mesmo outros protocolos de comunicação para efeitos de comparação podem ser alvos de pesquisas futuras;
- Investigar o conceito de confiabilidade sob a óptica de outras métricas que ficaram fora do escopo deste trabalho;
- Aplicação da técnica de simulação de experimentos através de ferramentas computacionais para abranger uma maior quantidade de cenários de ensaios;
- Uso em uma aplicação real, dentro de um domínio específico de sistemas IoT, para análise de métricas específicas;
- Estudo do acoplamento da arquitetura em uma aplicação já em uso e/ou *frameworks*;
- Aprofundar a análise sobre a definição da estrutura de dados para a aplicação de políticas de acesso, que seja adequada ao paradigma IoT.
- Estudo sobre ameaças à confiabilidades em relação ao comportamento das entidades em IoT (aspectos sociais).

REFERÊNCIAS

ABBAS, S.; NAZ, M.; ANWAAR, Z.; FAROOQ, M. U.; KHAN, F. U. Availability testing of iot-based health care devices: A survey. In: IEEE. **2021 International Conference on Innovative Computing (ICIC)**. [*S. l.*], 2021. p. 1–6.

AHMAD, T.; RANISE, S. Validating requirements of access control for cloud-edge iot solutions (short paper). In: SPRINGER. **International Symposium on Foundations and Practice of Security**. [*S. l.*], 2018. p. 131–139.

ALKHRESHEH, A.; ELGAZZAR, K.; HASSANEIN, H. S. Context-aware automatic access policy specification for iot environments. In: IEEE. **2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)**. [*S. l.*], 2018. p. 793–799.

ALKHRESHEH, A.; ELGAZZAR, K.; HASSANEIN, H. S. Daciot: Dynamic access control framework for iot deployments. **IEEE Internet of Things Journal**, IEEE, v. 7, n. 12, p. 11401–11419, 2020.

ALRAMADHAN, M.; SHA, K. An overview of access control mechanisms for internet of things. In: IEEE. **2017 26th International Conference on Computer Communication and Networks (ICCCN)**. [*S. l.*], 2017. p. 1–6.

ALWARAFY, A.; AL-THELAYA, K. A.; ABDALLAH, M.; SCHNEIDER, J.; HAMDI, M. A survey on security and privacy issues in edge computing-assisted internet of things. **IEEE Internet of Things Journal**, IEEE, 2020.

ARFAOUI, A.; BOUDIA, O. R. M.; KRIBECHE, A.; SENOUCI, S.-M.; HAMDI, M. Context-aware access control and anonymous authentication in wban. **Computers & Security**, Elsevier, v. 88, p. 101496, 2020.

AVASALCAI, C.; MURTURI, I.; DUSTDAR, S. Edge and fog: A survey, use cases, and future challenges. **Fog Computing: Theory and Practice**, Wiley Online Library, p. 43–65, 2020.

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. **IEEE transactions on dependable and secure computing**, IEEE, v. 1, n. 1, p. 11–33, 2004.

BAGDASARYAN, E.; BERLSTEIN, G.; WATERMAN, J.; BIRRELL, E.; FOSTER, N.; SCHNEIDER, F. B.; ESTRIN, D. Ancile: Enhancing privacy for ubiquitous computing with use-based privacy. In: **Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society**. [*S. l.*: *s. n.*], 2019. p. 111–124.

BANDARA, S.; YASHIRO, T.; KOSHIZUKA, N.; SAKAMURA, K. Access control framework for api-enabled devices in smart buildings. In: IEEE. **2016 22nd Asia-Pacific Conference on Communications (APCC)**. [*S. l.*], 2016. p. 210–217.

BATE, K. O.; KUMAR, N.; KHATRI, S. K. Framework for authentication and access control in iot. In: IEEE. **2017 2nd International Conference on Telecommunication and Networks (TEL-NET)**. [*S. l.*], 2017. p. 1–6.

BERNABE, J. B.; RAMOS, J. L. H.; GOMEZ, A. F. S. Taciot: multidimensional trust-aware access control system for the internet of things. **Soft Computing**, Springer, v. 20, n. 5, p. 1763–1779, 2016.

BINDRA, L.; LIN, C.; STROULIA, E.; ARDAKANIAN, O. Decentralized access control for smart buildings using metadata and smart contracts. In: IEEE. **2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)**. [*S. l.*], 2019. p. 32–38.

BISWAS, A. R.; GIAFFREDA, R. Iot and cloud convergence: Opportunities and challenges. In: IEEE. **2014 IEEE World Forum on Internet of Things (WF-IoT)**. [*S. l.*], 2014. p. 375–376.

BOUANANI, S. E.; KIRAM, M. A. E.; ACHBAROU, O.; OUTCHAKOUCHT, A. Pervasive-based access control model for iot environments. **IEEE Access**, IEEE, v. 7, p. 54575–54585, 2019.

BOUIJ-PASQUIER, I.; OUAHMAN, A. A.; KALAM, A. A. E.; MONTFORT, M. O. de. Smartorbac security and privacy in the internet of things. In: IEEE. **2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)**. [*S. l.*], 2015. p. 1–8.

BROSSARD, D.; GEBEL, G.; BERG, M. A systematic approach to implementing abac. In: **Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control**. [*S. l.*: *s. n.*], 2017. p. 53–59.

BUKH, P. N. D. **The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling**. [*S. l.*]: JSTOR, 1992.

CHANAL, P. M.; KAKKASAGERI, M. S. Security and privacy in iot: a survey. **Wireless Personal Communications**, Springer, v. 115, n. 2, p. 1667–1693, 2020.

DENNIS, J. B.; HORN, E. C. V. Programming semantics for multiprogrammed computations. **Communications of the ACM**, ACM, v. 9, n. 3, p. 143–155, 1966.

DENNISS, W.; BRADLEY, J. Oauth 2.0 for native apps. **Internet Engineering Task Force, Internet-Draft draft-ietf-oauthnative-apps-05**, 2016.

DENNISS, W.; BRADLEY, J. **OAuth 2.0 for Native Apps**. 2017. Disponível em: https://tools.ietf.org/html/rfc6749. Acesso em: 2017.

DESHMUKH, S.; SONAVANE, S. Security protocols for internet of things: A survey. In: IEEE. **2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)**. [*S. l.*], 2017. p. 71–74.

DONG, Y.; WAN, K.; HUANG, X.; YUE, Y. Contexts-states-aware access control for internet of things. In: IEEE. **2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))**. [*S. l.*], 2018. p. 666–671.

FILHO, N. B. **Trumit-gerenciador de confiança para IOT**. 72 p. Dissertação (Mestrado) – Programa de PósGraduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná., Curitiba - PR, 2019.

GUPTA, A.; FERNANDO, X.; DAS, O. Reliability and availability modeling techniques in 6g iot networks: A taxonomy and survey. **2021 International Wireless Communications and Mobile Computing (IWCMC)**, IEEE, p. 586–591, 2021.

GUSEV, M.; DUSTDAR, S. Going back to the roots—the evolution of edge computing, an iot perspective. **IEEE internet Computing**, IEEE, v. 22, n. 2, p. 5–15, 2018.

HEMDI, M.; DETERS, R. Using rest based protocol to enable abac within iot systems. In: IEEE. **2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)**. [*S. l.*], 2016. p. 1–7.

HERNÁNDEZ-RAMOS, J. L.; MORENO, M. V.; BERNABÉ, J. B.; CARRILLO, D. G.; SKARMETA, A. F. Safir: Secure access framework for iot-enabled services on smart buildings. **Journal of Computer and System Sciences**, Elsevier, v. 81, n. 8, p. 1452–1463, 2015.

HU, V. C.; FERRAIOLO, D.; KUHN, R.; FRIEDMAN, A. R.; LANG, A. J.; COGDELL, M. M.; SCHNITZER, A.; SANDLIN, K.; MILLER, R.; SCARFONE, K. *et al.* Guide to attribute based access control (abac) definition and considerations (draft). **NIST special publication**, Citeseer, v. 800, n. 162, 2013.

IoT Analytics Research. **tate of the IoT 2018: Number of IoT devices now at 7B -Market accelerating**. 2018. Disponível em: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b. Acesso em: 04 Março 2021.

JONES, M.; MICROSOFT; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. [*S. l.*], 2015. 1-25 p. Disponível em: https://www.rfc-editor.org/rfc/rfc7519.

KALAM, E.; ABOU, A.; OUTCHAKOUCHT, A.; ES-SAMAALI, H. Emergence-based access control: New approach to secure the internet of things. In: ACM. **Proceedings of the 1st International Conference on Digital Tools & Uses Congress**. [*S. l.*], 2018. p. 15.

KARIMIBIUKI, M.; AGGARWAL, E.; PATTABIRAMAN, K.; IVANOV, A. Dynpolac: Dynamic policy-based access control for iot systems. In: IEEE. **2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)**. [*S. l.*], 2018. p. 161–170.

KAWAGUCHI, R.; BANDAI, M. Edge based mqtt broker architecture for geographical iot applications. In: IEEE. **2020 International Conference on Information Networking (ICOIN)**. [*S. l.*], 2020. p. 232–235.

KEMPF, J.; ARKKO, J.; BEHESHTI, N.; YEDAVALLI, K. Thoughts on reliability in the internet of things. In: **Interconnecting smart objects with the Internet workshop**. [*S. l.*: *s. n.*], 2011. v. 1, p. 1–4.

KHAN, R.; KHAN, S. U.; ZAHEER, R.; KHAN, S. Future internet: the internet of things architecture, possible applications and key challenges. In: IEEE. **2012 10th international conference on frontiers of information technology**. [*S. l.*], 2012. p. 257–260.

KOME, M. L.; CUPPENS, F.; CUPPENS-BOULAHIA, N.; FREY, V. Coap enhancement for a better iot centric protocol: Coap 2.0. In: IEEE. **2018 Fifth international conference on internet of things: systems, management and security**. [*S. l.*], 2018. p. 139–146.

KUMAR, G. R.; KISHORE, D.; KUMAR, G. V. M. G.; AVILA, J.; THENMOZHI, K.; AMIRTHARAJA, R.; PRAVEENKUMAR, P. Waste contamination in water—a real-time water quality monitoring system using iot. In: IEEE. **2021 International Conference on Computer Communication and Informatics (ICCCI)**. [*S. l.*], 2021. p. 1–4.

LAN, L.; SHI, R.; WANG, B.; ZHANG, L. An iot unified access platform for heterogeneity sensing devices based on edge computing. **IEEE access**, IEEE, v. 7, p. 44199–44211, 2019.

LEE, Y.-k.; LIM, J.-d.; JEON, Y.-s.; KIM, J.-n. Technology trends of access control in iot and requirements analysis. In: IEEE. **2015 International Conference on Information and Communication Technology Convergence (ICTC)**. [*S. l.*], 2015. p. 1031–1033.

LI, F.; HAN, Y.; JIN, C. Practical access control for sensor networks in the context of the internet of things. **Computer Communications**, Elsevier, v. 89, p. 154–164, 2016.

LIU, H.; HAN, D.; LI, D. Fabric-iot: A blockchain-based access control system in iot. **IEEE Access**, IEEE, v. 8, p. 18207–18218, 2020.

LIU, Y.; PENG, M.; SHOU, G.; CHEN, Y.; CHEN, S. Toward edge intelligence: Multiaccess edge computing for 5g and internet of things. **IEEE Internet of Things Journal**, IEEE, v. 7, n. 8, p. 6722–6747, 2020.

MA, Z.; XIAO, M.; XIAO, Y.; PANG, Z.; POOR, H. V.; VUCETIC, B. High-reliability and low-latency wireless communication for internet of things: challenges, fundamentals, and enabling technologies. **IEEE Internet of Things Journal**, IEEE, v. 6, n. 5, p. 7946–7970, 2019.

MACHESO, P.; MANDA, T. D.; CHISALE, S.; DZUPIRE, N.; MLATHO, J.; MUKANYILIGIRA, D. Design of esp8266 smart home using mqtt and node-red. In: IEEE. **2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)**. [*S. l.*], 2021. p. 502–505.

MAHYOUB, M.; MAHMOUD, A.; SHELTAMI, T. An optimized discovery mechanism for smart objects in iot. In: IEEE. **2017 8th IEEE annual information technology, electronics and mobile communication conference (IEMCON)**. [*S. l.*], 2017. p. 649–655.

MAITA, S. L. S. **New Models of Reliability in the New Generation of Internet of Things**. Tese (Doutorado) – 00500:: Universidade de Coimbra, 2020.

MECHALIKH, C.; TAKTAK, H.; MOUSSA, F. Pureedgesim: a simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments. In: IEEE. **2019 International Conference on High Performance Computing & Simulation (HPCS)**. [*S. l.*], 2019. p. 700–707.

MEDINA, C. A.; PEREZ, M. R.; TRUJILLO, L. C. Iot paradigm into the smart city vision: a survey. In: IEEE. **2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)**. [*S. l.*], 2017. p. 695–704.

NGU, A. H.; GUTIERREZ, M.; METSIS, V.; NEPAL, S.; SHENG, Q. Z. Iot middleware: A survey on issues and enabling technologies. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 1, p. 1–20, 2016.

NUAIMI, N. A. D. E. A. Managing qos in iots: a survey. In: **ICC '17: Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing**. [*S. l.*: *s. n.*], 2017. p. 1–7.

OMONIWA, B.; HUSSAIN, R.; JAVED, M. A.; BOUK, S. H.; MALIK, S. A. Fog/edge computing-based iot (feciot): Architecture, applications, and research issues. **IEEE Internet of Things Journal**, IEEE, v. 6, n. 3, p. 4118–4149, 2018.

OUADDAH, A.; MOUSANNIF, H.; ELKALAM, A. A.; OUAHMAN, A. A. Access control in the internet of things: Big challenges and new opportunities. **Computer Networks**, Elsevier, v. 112, p. 237–262, 2017.

PAL, S.; HITCHENS, M.; VARADHARAJAN, V. Access control for internet of things—enabled assistive technologies: an architecture, challenges and requirements. In: **Assistive Technology for the Elderly**. [*S. l.*]: Elsevier, 2020. p. 1–43.

PAL, S.; HITCHENS, M.; VARADHARAJAN, V.; RABEHAJA, T. On design of a fine-grained access control architecture for securing iot-enabled smart healthcare systems. In: **Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services**. [*S. l.*: *s. n.*], 2017. p. 432–441.

PARK, J.; SANDHU, R. Towards usage control models: beyond traditional access control. In: ACM. **Proceedings of the seventh ACM symposium on Access control models and technologies**. [*S. l.*], 2002. p. 57–64.

RAVIDAS, S.; KARKHANIS, P.; DAJSUREN, Y.; ZANNONE, N. An authorization framework for cooperative intelligent transport systems. In: SPRINGER. **International Workshop on Emerging Technologies for Authorization and Authentication**. [*S. l.*], 2019. p. 16–34.

RAVIDAS, S.; LEKIDIS, A.; PACI, F.; ZANNONE, N. Access control in internet-of-things: A survey. **Journal of Network and Computer Applications**, Elsevier, v. 144, p. 79–101, 2019.

SANDHU, R. S.; COYNE, E. J.; FEINSTEIN, H. L.; YOUMAN, C. E. Role-based access control models. **Computer**, IEEE, v. 29, n. 2, p. 38–47, 1996.

SHI, W.; CAO, J.; ZHANG, Q.; LI, Y.; XU, L. Edge computing: Vision and challenges. **IEEE internet of things journal**, IEEE, v. 3, n. 5, p. 637–646, 2016.

SUCIU, I.; VILAJOSANA, X.; ADELANTADO, F. An analysis of packet fragmentation impact in lpwan. In: IEEE. **2018 IEEE Wireless Communications and Networking Conference (WCNC)**. [*S. l.*], 2018. p. 1–6.

THIAM, F.; MBAYE, M.; WYGLINSKI, A. M. Generic reliability analysis model of iot: Agriculture use case. In: IEEE. **2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)**. [*S. l.*], 2021. p. 1–5.

THOMAS, M. O.; RAD, B. B. Reliability evaluation metrics for internet of things, car tracking system: a review. **Int. J. Inf. Technol. Comput. Sci.(IJITCS)**, v. 9, n. 2, p. 1–10, 2017.

VISHWAKARMA, S. K.; UPADHYAYA, P.; KUMARI, B.; MISHRA, A. K. Smart energy efficient home automation system using iot. In: IEEE. **2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU)**. [*S. l.*], 2019. p. 1–4.

WALLACE, B.; MARSHALL, S. Technology roadmap for intelligent buildings. 2003.

WANG, J.; HUANG, J.; CHEN, W.; LIU, J.; XU, D. Design of iot-based energy efficiency management system for building ceramics production line. In: IEEE. **2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)**. [*S. l.*], 2016. p. 912–917.

WANG, P.; YUE, Y.; SUN, W.; LIU, J. An attribute-based distributed access control for blockchain-enabled iot. In: IEEE. **2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**. [*S. l.*], 2019. p. 1–6.

WU, M.; LU, T.-J.; LING, F.-Y.; SUN, J.; DU, H.-Y. Research on the architecture of internet of things. In: IEEE. **2010 3rd international conference on advanced computer theory and engineering (ICACTE)**. [*S. l.*], 2010. v. 5, p. V5–484.

XING, L. Reliability in internet of things: Current status and future perspectives. **IEEE Internet of Things Journal**, IEEE, v. 7, n. 8, p. 6704–6721, 2020.

XUE, H.; HUANG, B.; QIN, M.; ZHOU, H.; YANG, H. Edge computing for internet of things: A survey. In: IEEE. **2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)**. [*S. l.*], 2020. p. 755–760.

ZHONG, C.-L.; ZHU, Z.; HUANG, R.-G. Study on the iot architecture and gateway technology. In: IEEE. **2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)**. [*S. l.*], 2015. p. 196–199.