



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Avaliação de Fog Computing para suporte a sistemas de localização interna de equipamentos hospitalares

Dissertação de Mestrado

Hélio Cardoso de Moura Filho



São Cristóvão – Sergipe

2022

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Hélio Cardoso de Moura Filho

**Avaliação de Fog Computing para suporte a sistemas de
localização interna de equipamentos hospitalares**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador: Rubens de Souza Matos Júnior

São Cristóvão – Sergipe

2022

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

M929a Moura Filho, Hélio Cardoso de
Avaliação de Fog Computing para suporte a sistemas de
localização interna de equipamentos hospitalares / Hélio Cardoso
de Moura Filho ; orientador Rubens de Souza Matos Júnior. - São
Cristóvão, 2022.
67 f.; il.

Dissertação (mestrado em Ciência da Computação) –
Universidade Federal de Sergipe, 2022.

1. Internet das coisas. 2. Hospitais - Administração. 3.
Computação. I. Matos Júnior, Rubens de Souza orient. II.
Título.

CDU 004

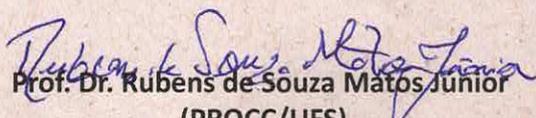


UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do
Curso de Mestrado em Ciência da Computação-UFS.
Candidato: HÉLIO CARDOSO DE MOURA FILHO

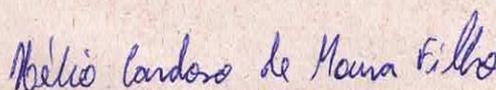
Em 29 dias do mês de agosto do ano de dois mil e vinte dois, com início às 9h, realizou-se na Sala de Seminários do PROCC da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Hélio Cardoso de Moura Filho**, que desenvolveu o trabalho intitulado: “**Avaliação de Fog Computing para suporte a sistemas de localização interna de equipamentos hospitalares**”, sob a orientação do Prof. Dr. **Rubens de Souza Matos Júnior**. A Sessão foi presidida pelo Prof. Dr. **Rubens de Souza Matos Júnior** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **Wanderson Roger Azevedo Dias** (IFRo) e, em seguida, o Prof. Dr. **Jean Carlos Teixeira De Araujo** (PROCC/UFS). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) aprovado “(aprovado/reprovado)”. Atendidas as exigências da Instrução Normativa 05/2019/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 04/2021/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária “Prof. José Aloísio de Campos”, 29 de agosto de 2022.


Prof. Dr. Rubens de Souza Matos Júnior
(PROCC/UFS)
Presidente


 Jean Carlos T. de Araujo
Professor Adjunto
SIAPE 2001286
Prof. Dr. Jean Carlos Teixeira De Araujo
(PROCC/UFS)
Examinador Interno


Prof. Dr. Wanderson Roger Azevedo Dias
(UFRO)
Examinador Externo


HÉLIO CARDOSO DE MOURA FILHO
Candidato

Agradecimentos

Primeiramente agradeço a Deus pela minha saúde que mesmo diante da pandemia do COVID-19 e acometido por essa doença me deu forças para continuar essa difícil jornada.

A minha esposa Ilmara, que diante das minhas incertezas, medos e ansiedade procurou sempre me apoiar e me acalmar, me incentivando a continuar nessa jornada. Continua sempre incentivando meu crescimento profissional e pessoal.

Aos meus pais, Hélio e Nautília, que na minha ausência, entenderam minhas dificuldades e incentivaram a continuar no caminho escolhido por mim.

Um agradecimento especial ao meu orientador Prof. Dr. Rubens Matos, que pacientemente me passou um pouco do seu conhecimento que foi fundamental na conclusão do mestrado.

Enfim, agradeço aos amigos do setor do qual trabalho e faço parte da equipe pelo incentivo e paciência. E também à secretária do Departamento de Computação da UFS, Elaine, pela disponibilidade e por tirar todas as dúvidas em relação ao programa. Agradeço a todos por terem contribuído nesta conquista pessoal e profissional. Minha eterna gratidão!

*A tarefa não é tanto ver aquilo que ninguém viu
mas pensar o que ninguém ainda pensou
sobre aquilo que todo mundo vê
(Arthur Schopenhauer)*

Resumo

O método tradicional de gestão hospitalar é difícil de realizar alocação de recursos para toda a área de forma unificada, o que acarreta em uma grande redundância de mão de obra e desperdício de recursos neste aspecto. A Internet das Coisas, ou Internet of Things (IoT), surgiu como um paradigma de comunicação que envolve sistemas organizados de dispositivos de computação, dispositivos mecânicos e digitais, objetos, coisas vivas ou não vivas que são capazes de transferir informações sem envolver interação humano-humano ou homem-máquina. Dessa forma, o hospital inteligente baseado em sistemas IoT pode proporcionar uma gestão muito mais eficiente e refinada para pessoas e regiões da área hospitalar. A automação hospitalar pode ser oferecida por uma das duas alternativas, seja por meio de computação centralizada ou computação distribuída, acrescentam que os sistemas centralizados não são desejados principalmente porque ao receber uma grande quantidade de dados teriam dois grandes inconvenientes: alta latência e alto uso da rede. No entanto, os dispositivos de borda geralmente têm poderes computacionais menores do que os datacenters na nuvem, o que força a estratégia de inserir no sistema uma camada intermediária, dando origem ao conceito de computação em névoa (*Fog Computing*). Diante dessa problemática e dos resultados observados por meio do mapeamento sistemático, resolvemos propor e analisar uma arquitetura baseada em *Fog Computing* para auxiliar na gestão de recursos hospitalares dando suporte na localização de equipamentos móveis (como desfibrilador, reanimador cardiopulmonar, ecocardiograma, entre outros) internamente. Simulamos dois modelos de aplicação (*display after storage* e *display before storage*) em três cenários (*Edge, Fog e Cloud Computing*), aumentando progressivamente a carga da rede com o aumento de dispositivos transmitindo dados a cada 1s, 2s, 5s e 10 segundos. Foram observados por meio do simulador iFogSim, que o Modelo 1 (*display after storage*), no cenário 2 (*Cloud Computing*) obteve melhor resultado para latência da aplicação. Já no Modelo 2 (*display before storage*), o menor tempo apresentado foi o cenário 3 (*Fog Computing*). Este trabalho também observou que alterar a capacidade de processamento em MIPS dos equipamentos localizados na *Fog Computing* não houve ganhos significativos para a latência.

Palavras-chave: Internet das Coisas, Sistema de Posicionamento Interno, Gerenciamento de Equipamentos Hospitalares, Computação em Névoa.

Abstract

The traditional method of hospital management is difficult to allocate resources to the entire area in a unified way, which leads to a great redundancy of manpower and waste of resources in this aspect. The Internet of Things (IoT), emerged as a communication paradigm that involves organized systems of computing devices, mechanical and digital devices, objects, living or non-living things that are capable of transferring information without involving interaction human-human or human-machine. In this way, the smart hospital based on IoT systems can provide much more efficient and refined management for people and regions in the hospital area. Hospital automation can be offered by one of two alternatives, either through centralized computing or distributed computing, add that centralized systems are not wanted mainly because when receiving a large amount of data they would have two large drawbacks: high latency and high network usage. However, edge devices generally have less computing power than cloud datacenters, which forces the strategy of inserting an intermediate layer into the system, giving rise to the concept of Fog Computing. Faced with this problem and the results observed through systematic mapping, we decided to propose and analyze an architecture based on *Fog Computing* to assist in the management of hospital resources, supporting the location of mobile equipment (such as a defibrillator, cardiopulmonary resuscitator, echocardiogram, among others) internally. We simulate two application models (display after storage and display before storage) in three scenarios (*Edge, Fog and Cloud Computing*), progressively increasing the network load with the increase of devices transmitting data every 1s, 2s, 5s and 10 seconds. Using the iFogSim simulator, it was observed that Model 1 (display after storage), in scenario 2 (*Cloud Computing*) obtained the best result for application latency. In Model 2 (display before storage), the shortest time presented was scenario 3 (*Fog Computing*). This work also observed that changing the processing capacity in MIPS of the equipment located in the *Fog Computing* did not cause significant latency gains.

Keywords: Internet of Things, IoT, Internal Positioning System, Hospital Equipment Management, Fog Computing.

Lista de figuras

Figura 1 – Arquitetura IoT de 5 (cinco) camadas	21
Figura 2 – Relação de dispositivos na camada Edge, Fog e Cloud Computing	23
Figura 3 – Porcentagem de artigos por fonte de dados	29
Figura 4 – Categorias de aplicações IoT na área da saúde	31
Figura 5 – Arquiteturas em camadas das aplicações IoT em Saúde	31
Figura 6 – Diagrama da solução proposta	37
Figura 7 – Proposta da arquitetura em camadas da solução proposta	38
Figura 8 – Modelos da aplicação proposta	40
Figura 9 – Mapa da localização física dos gateways	41
Figura 10 – Arquivo de configuração da topologia física	41
Figura 11 – Topologia física da solução proposta	42
Figura 12 – Etapas dos processos realizados em cada cenário do Modelo 1 - <i>Display after Storage</i>	44
Figura 13 – Etapas dos processos realizados em cada cenário do Modelo 2 - <i>Display before Storage</i>	45
Figura 14 – Média do atraso de <i>loop</i> do aplicativo para o cenário 1 (<i>Edge</i>) no Modelo 1 (<i>display after storage</i>)	46
Figura 15 – Média do atraso de <i>loop</i> do aplicativo para o cenário 2 (<i>Cloud</i>) no Modelo 1 (<i>display after storage</i>)	46
Figura 16 – Média do atraso de <i>loop</i> do aplicativo para o cenário 3 (<i>Fog</i>) no Modelo 1 (<i>display after storage</i>)	47
Figura 17 – Média do atraso de <i>loop</i> do aplicativo para o cenário 1 (<i>Edge</i>) no Modelo 2 (<i>display before storage</i>)	48
Figura 18 – Média do atraso de <i>loop</i> do aplicativo para o cenário 2 (<i>Cloud</i>) no Modelo 2 (<i>display before storage</i>)	48
Figura 19 – Média do atraso de <i>loop</i> do aplicativo para o cenário 3 (<i>Fog</i>) no Modelo 2 (<i>display before storage</i>)	49
Figura 20 – Comparativo do <i>Application Loop Delay</i> no Modelo 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>)	49
Figura 21 – Comparativo do <i>Application Loop Delay</i> , Modelo 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>), no cenário de <i>Fog</i> com retransmissão de 10 segundos	50
Figura 22 – Comparativo do <i>Application Loop Delay</i> nos cenários de <i>Cloud</i> e <i>Fog</i> do Modelo 2 (<i>display before storage</i>)	50
Figura 23 – Média do <i>Network Usage</i> para o cenário 1 (<i>Edge</i>) no Modelo 1 (<i>display after storage</i>)	52

Figura 24 – Média do <i>Network Usage</i> para o cenário 2 (<i>Cloud</i>) no Modelo 1(<i>display after storage</i>)	52
Figura 25 – Média do <i>Network Usage</i> para o cenário 3 (<i>Fog</i>) no Modelo 1 (<i>display after storage</i>)	53
Figura 26 – Média do <i>Network Usage</i> para o cenário 1 (<i>Edge</i>) no Modelo 2 (<i>display before storage</i>)	53
Figura 27 – Média do <i>Network Usage</i> para no cenário 2 (<i>Cloud</i>) no Modelo 2 (<i>display before storage</i>)	54
Figura 28 – Média do <i>Network Usage</i> para o cenário 3 (<i>Fog</i>) no Modelo 2 (<i>display before storage</i>)	54
Figura 29 – Comparativo do <i>Network Usage</i> no Modelo 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>)	55
Figura 30 – Média do <i>Application Loop Delay</i> com alteração no MIPS	58
Figura 31 – Média do <i>Network Usage</i> com alteração no MIPS	58
Figura 32 – Média do <i>Execution Time</i> com alteração no MIPS	59

Lista de tabelas

Tabela 1 – Fontes de pesquisa utilizadas no mapeamento sistemático	28
Tabela 2 – Relação da quantidade de artigos e suas respectivas fontes de pesquisa	28
Tabela 3 – Critérios de inclusão e critérios de exclusão aplicados nos artigos	30
Tabela 4 – Relação de artigos selecionados por fonte de pesquisa	30
Tabela 5 – Resumo dos trabalhos relacionados categorizados quanto ao tipo de gerenciamento de recursos, arquitetura, segurança e tecnologia utilizada pelos autores selecionados	35
Tabela 6 – Configuração dos Dispositivos <i>Fog</i> para aplicação da solução proposta	42
Tabela 7 – Descrição dos links de rede para o aplicativo de rastreamento	43
Tabela 8 – Tabela geral da latência média da aplicação nos cenários de <i>Edge</i> , <i>Cloud</i> e <i>Fog</i> nos Modelos 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>)	51
Tabela 9 – Tabela geral do <i>network usage</i> nos cenários de <i>Edge</i> , <i>Cloud</i> e <i>Fog</i> nos Modelos 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>)	56
Tabela 10 – Tabela geral do <i>execution time</i> nos cenários de <i>Edge</i> , <i>Cloud</i> e <i>Fog</i> nos Modelos 1 (<i>display after storage</i>) e Modelo 2 (<i>display before storage</i>)	57
Tabela 11 – Configuração dos Dispositivos <i>Fog</i> com alteração do MIPS	57

Lista de códigos

Código 1 – Código para alteração de cenário de simulação	44
--	----

Lista de abreviaturas e siglas

BLE	<i>Bluetooth Low Energy</i>
CSV	<i>Comma Separated Values</i>
DTLS	<i>Datagram Transport Layer Security</i>
HF	<i>High Frequency</i>
IoT	<i>Internet of Things</i>
LAN	<i>Local Area Network</i>
LoRa	<i>Long Range</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
NFC	<i>Near Field Communication</i>
PC	<i>Personal Computer</i>
QoS	<i>Quality of Service</i>
RFID	<i>Radio-Frequency Identification</i>
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
TLS	<i>Transport Layer Security</i>
UHF	<i>Ultra High Frequency</i>
Wi-Fi	<i>Wireless Fidelity</i>

Sumário

1	Introdução	13
1.1	Problemática e Hipótese	14
1.2	Justificativa	15
1.3	Objetivos	16
1.4	Metodologia de Pesquisa	17
1.4.1	Classificação da Pesquisa	17
1.4.2	Etapas da Pesquisa	18
1.5	Organização do Trabalho	18
2	Fundamentação Teórica	19
2.1	Internet das Coisas	19
2.2	<i>Fog Computing</i>	21
2.2.1	iFogSim	22
2.3	<i>Smart Healthcare (e-Health)</i>	24
2.4	Gerenciamento de Recursos Hospitalares	25
2.5	Considerações Finais	26
3	Revisão da Literatura	27
3.1	Mapeamento Sistemático	27
3.1.1	Questões de Pesquisa	27
3.1.2	Estratégia de Busca e Seleção de Fontes	28
3.1.3	Critérios de Seleção	28
3.1.4	Análise de Resultados	29
3.2	Análise dos Trabalhos Relacionados	32
3.3	Solução Proposta	34
3.4	Considerações Finais	35
4	Avaliação da Fog Computing para suporte a sistemas de localização interna	36
4.1	Arquitetura da Aplicação	37
4.2	Estudo de Caso	39
4.3	Descrição dos Cenários e Parâmetros da Simulação	41
4.4	Resultados	45
4.4.1	Latência Média do Loop de Aplicativo	45
4.4.2	Uso da Rede	51
4.4.3	<i>Execution Time</i>	55
4.4.4	MIPS 4000 vs MIPS 22400	56

5	Conclusão	60
5.1	Contribuições alcançadas	61
5.2	Dificuldades e Limitações	61
5.3	Trabalhos Futuros	61
5.4	Publicações Relacionadas à Dissertação	62
5.4.1	Artigo Aprovado	62
5.4.2	Artigo Submetido	62
	Referências	63

1

Introdução

A Internet das Coisas, ou *Internet of Things* (IoT), é um paradigma de comunicação que envolve sistemas organizados de dispositivos de computação, dispositivos mecânicos e digitais, objetos, coisas vivas ou não vivas que são capazes de transferir informações sem envolver interação humano-humano ou homem-máquina (VIJAYALAKSHMI; ABISHEK; STEPHEN, 2019). Isso significa que permite a conexão com qualquer pessoa, qualquer coisa, a qualquer hora, em qualquer lugar, qualquer serviço e qualquer rede sem a necessidade de ter alguém (humano) cadastrando ou alterando as informações no sistema, sendo feito de forma automática através de sensores.

Segundo Mohanty, Kougianos e Guturu (2018), uma implementação útil da IoT em geral, e da saúde inteligente em particular, apresenta vários desafios, incluindo a eficiência energética dos componentes conectados ao sistema; segurança das comunicações; identificação única de usuários que garante a privacidade das informações dos usuários; desempenho computacional, confiabilidade e flexibilidade do sistema; análise eficiente de grandes conjuntos de dados em tempo real; e processamento e armazenamento de enormes quantidades de dados para estudos de acompanhamento e para descoberta de conhecimento em estágios de meta-análise.

Em relação à gestão hospitalar, o método tradicional é difícil de realizar uma gestão unificada e alocação de recursos para toda a área hospitalar, o que acarreta em uma grande redundância de mão de obra e desperdício de recursos neste aspecto. Para isso, o hospital inteligente baseado em sistemas IoT pode proporcionar uma gestão muito mais eficiente e refinada para pessoas e regiões da área hospitalar (LENG; LIN; WANG, 2020).

A automação pode ser oferecida por uma das duas alternativas, seja por meio de computação centralizada ou computação distribuída, acrescentam que os sistemas centralizados não são desejados principalmente porque ao receber uma grande quantidade de dados teriam dois grandes inconvenientes: alta latência e alto uso da rede. No entanto, os dispositivos de borda geralmente têm poderes computacionais menores do que os datacenters na nuvem, o que

força a estratégia de inserir no sistema uma camada intermediária, dando origem ao conceito de computação em névoa (*Fog Computing*). Se um dispositivo tem poder computacional suficiente para processar o módulo de aplicação em mãos, ele fará isso, caso contrário, enviará a tarefa para o dispositivo de névoa superior até atingir a nuvem, se necessário (HAGGAG; OUF, 2020).

O novo paradigma de computação *Edge/Fog* abrange as instalações de computação em toda a rede próxima e permite que saúde inteligente, cidade inteligente, Indústria 4.0 e outros sistemas ciber físicos habilitados para IoT executem o software de aplicativo necessário nas proximidades da fonte de dados. Os sistemas ciber físicos (ou CPS, na sigla em inglês para *Cyber-Physical Systems*) são integrações que envolvem computação, comunicação e controle através de redes e processos físicos. Por intermédio desses sistemas, as empresas têm a oportunidade de representar a realidade do mundo físico em ambientes digitais. Sendo assim, conseguem fazer simulações, testes, predições de desgastes, entre muitas outras possibilidades que a tecnologia oferece (OLIVEIRA, 2020). Assim, a computação *Edge/Fog* garante a entrega de serviços de aplicativos para os sistemas ciber físicos habilitados para IoT com atraso de propagação e transmissão de dados reduzida, diminuindo a possibilidade de congestionamento dentro da infraestrutura de rede central e inibindo a transferência de uma grande quantidade de dados brutos para os data centers em nuvem (MAHMUD et al., 2021).

Diante do exposto acima, o trabalho realizado nessa dissertação propõe e avalia uma arquitetura com *Fog Computing* para localização interna em tempo real de equipamentos hospitalares incrementando ao método tradicional de gestão hospitalar uma gestão unificada de toda a área hospitalar, minimizando a redundância de mão de obra e desperdício de recursos neste aspecto.

1.1 Problemática e Hipótese

Localizar equipamentos móveis, como por exemplo, desfibrilador, reanimador cardiopulmonar, ecocardiograma, cadeira de rodas, entre outros equipamentos de uso geral do hospital é uma tarefa morosa e custosa. Visando minimizar tal impacto, alguns gestores alocam recursos de pessoal para de tempos em tempos passar nos setores com uma planilha em mãos anotando a localização desses equipamentos que precisam ser monitorados. Espera-se que ao automatizar essa tarefa com uso de IoT e *Fog Computing*, haja uma melhoria na gestão da localização desses equipamentos além de desafogar recursos de pessoal para emprego em outras tarefas.

Na construção de um sistema de localização interna de equipamentos por meio de recursos da IoT, diferentes tipos de infraestrutura computacionais para processamento e armazenamento dos dados podem ser empregados, desde uma comunicação direta com servidores em nuvem, até o uso de computação em névoa para redução da latência e do tráfego de dados de forma externa à rede do ambiente hospitalar. A hipótese principal deste trabalho é de que o uso de *Fog Computing* para suporte a sistemas desse tipo pode trazer benefícios quanto às métricas

citadas, porém esse benefício precisa ser mensurado de acordo com as características de rede e do conjunto de sensores existentes na infraestrutura hospitalar, para correto dimensionamento dessa infraestrutura, evitando problemas de desempenho ou gastos desnecessários.

1.2 Justificativa

Além da tarefa de localizar/monitorar equipamentos hospitalares ser muito custosa e morosa, principalmente numa área extensa (o próprio hospital), e tendo que sobrecarregar recursos humanos, vários projetos de tecnologia da informação incentivam o uso de IoT, incluindo aplicativos corporais, casa inteligente, cidade inteligente e projetos de ambiente inteligente. Devido ao uso extensivo da tecnologia IoT em todos os campos da vida, seu uso nas ciências da saúde também é uma entidade importante para a vida humana e tem um grande impacto na economia (PASHA; SHAH, 2018).

Portanto, diferentes projetos de serviços de saúde inteligentes baseados em IoT estão sendo iniciados em todo o mundo. Vários tipos de serviços de saúde inteligentes estão sendo fornecidos ao público. Isso pode incluir o seguinte: monitoramento remoto da saúde do paciente, tratamento do paciente em uma emergência, lembretes de exames médicos e de rotina, prescrições remotas do paciente e buscas pelos recursos de saúde mais próximos do paciente, como médicos, paramédicos, medicamentos, serviços de ambulância, e muitos outros recursos de saúde (PASHA; SHAH, 2018). Ao utilizar sistemas de gestão de materiais informatizados, o Hospital Universitário da Universidade de São Paulo (HU-USP) diagnosticou uma redução de 8,13% no consumo de materiais, diminuição na quantidade de itens no estoque em 26,22% e minimização de 12,46% nos custos do setor (PASCHOAL; CASTILHO, 2010).

A pesquisa TIC Saúde vem apontando, desde 2013, um aumento da presença de computadores e Internet nos estabelecimentos de saúde brasileiros (NIC.BR, 2019). Segundo a Sociedade Brasileira de Informática em Saúde (SBIS), entre 20 e 25% dos hospitais brasileiros utilizavam sistemas de informação, no ano de 2014 (SOGIMIG, 2014). Os resultados demonstraram, ao longo dos últimos anos, um crescente acesso às TIC (em especial, computadores e Internet) nesses estabelecimentos de saúde. Em 2021, verificou-se um aumento no percentual de estabelecimentos de saúde que usaram notebook e tablet. O uso de tablets já apresentava uma tendência de aumento desde 2017, quando estava presente em 7% dos estabelecimentos de saúde, chegando a 29%, em 2021. Já o uso de notebooks ocorreu em 61% dos estabelecimentos em 2021, proporção que era de 48% em 2019. O uso dos sistemas informatizados em saúde possibilita o acesso e a troca de dados de maneira ágil e coordenada e contribui para priorizar a atenção, o controle, as análises de resultados e os desfechos clínicos (NIC.BR, 2021).

Além disso, entende-se que o uso de sistemas automatizados de coleta de informações sem a necessidade de intervenção direta dos profissionais de saúde pode ser benéfico, evitando a existência de dados incompletos, otimizando tratamentos dando autonomia ao paciente e

segurança à conduta médica, assim como, aumentando a eficácia e reduzindo desperdícios financeiros e de pessoal. Por exemplo, os prontuários eletrônicos de pacientes facilitam a transferência mais eficiente de informação entre os setores, aumenta a qualidade e a exatidão dos dados, e o apoio à tomada de decisão (RATHERT et al., 2019), assim como, indicadores-chave tais como a utilização de leitos e o tempo médio de ocupação dos mesmos podem ser coletados por meio de sensores, obtendo mais um benefício da automatização de procedimentos hospitalares.

O Hospital é um local com grande fluxo de pessoas e vários tipos de pessoal. O Hospital inteligente baseado em sistemas IoT pode proporcionar uma gestão muito mais eficiente e refinada para pessoas e regiões da área hospitalar, além de atender às necessidades de informações médicas de saúde como uso de equipamentos e suprimentos médicos e gestão inteligente e monitoramento da segurança de saúde pública. Não há dúvida de que a tecnologia IoT tem grande potencial na área médica podendo ajudar os hospitais a realizar um tratamento médico inteligente e a administrar as coisas (LENG; LIN; WANG, 2020; LU et al., 2019).

Desta forma, com objetivo de implementar os benefícios supracitados, o presente trabalho propõe a análise de uma arquitetura em 5 (cinco) camadas (camada de sensores; camada de rede; camada de pré-processamento; camada de armazenamento; e camada de aplicação) com uso da *Fog Computing* para gerenciar/monitorar os equipamentos hospitalares, tendo como propósito facilitar a localização interna desses equipamentos e consequentemente reduzir custos e esforços que poderiam ser empregados em outras atividades de atendimento de saúde.

1.3 Objetivos

Este trabalho visa analisar o desempenho de uma arquitetura computacional proposta com base na tecnologia IoT e no uso da *Fog* e *Cloud Computing* para auxiliar na gestão de recursos dos ambientes de saúde. Para possibilitar a realização do objetivo geral, enumeramos os seguintes objetivos específicos:

- propor uma arquitetura para agregar confidencialidade e integridade dos dados trafegados e armazenados;
- simular e analisar o desempenho da IoT com uso da *Fog Computing* para suporte à localização interna de equipamentos;
- avaliar o desempenho da arquitetura proposta com uso da IoT nos cenários de *Edge*, *Fog* e *Cloud Computing*;
- propor mecanismos para acompanhamento em tempo real do uso, consumo e localização de equipamentos.

1.4 Metodologia de Pesquisa

Segundo Falbo (2013), Revisão Sistemática (RS) e Mapeamento Sistemático (MS) são tipos de estudos secundários que seguem um processo de pesquisa metodologicamente bem definido para identificar, analisar e interpretar as evidências disponíveis relacionadas a um particular conjunto de questões de pesquisa, tópico ou fenômeno de interesse, de uma maneira não tendenciosa e, até certo grau, repetível.

Nesse sentido, a pesquisa foi iniciada com a realização de um mapeamento sistemático da literatura, tendo por finalidade encontrar pesquisas sobre Internet das Coisas nas Universidades com o objetivo de conhecer o estado da arte da aplicação de IoT para *Smart Campus*. Nesse estudo, foi possível identificar tecnologias IoT (*Internet of Things*) e ambientes inteligentes além de identificar estudos realizados na área da saúde.

Devido aos resultados da pesquisa acima, decidiu-se focar na área de saúde, realizando-se um segundo mapeamento sistemático da literatura para identificar lacunas e tendências da aplicação de Internet das Coisas na área da saúde, sendo apresentado mais à frente, no Capítulo 3, apenas esse estudo. Em posse das informações do segundo mapeamento sistemático e diante do cenário atual de pandemia do COVID-19 foi proposto o tema deste trabalho visando permitir um melhor gerenciamento dos recursos hospitalares, em especial, a localização interna de equipamentos móveis.

1.4.1 Classificação da Pesquisa

Será adotado a abordagem de pesquisa quantitativa para a realização da pesquisa como forma de avaliar e melhorar a qualidade dos processos utilizados atualmente em ambientes de saúde. Segundo Gerhardt et al. (2009), a pesquisa quantitativa se centra na objetividade. Influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno, as relações entre variáveis, etc.

Quanto à natureza, a pesquisa será desenvolvida sobre uma perspectiva de natureza aplicada, tendo em vista o interesse da aplicação do conhecimento gerado. Ainda segundo Gerhardt et al. (2009), a pesquisa de natureza aplicada objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos e envolve verdades e interesses locais. O objetivo da pesquisa apresenta caráter exploratório e descritivo. A pesquisa exploratória tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. Quanto ao objetivo descritivo, esse tipo de estudo pretende descrever os fatos e fenômenos de determinada realidade.

Quanto ao procedimento, a pesquisa será desenvolvida sobre uma perspectiva de pesquisa experimental. Segundo Gil et al. (2002), a pesquisa experimental consiste em determinar um

objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto. Segundo [Fonseca \(2002\)](#), a pesquisa experimental seleciona grupos de assuntos coincidentes, submete-os a tratamentos diferentes, verificando as variáveis estranhas e checando se as diferenças observadas nas respostas são estatisticamente significantes.

1.4.2 Etapas da Pesquisa

Os seguintes passos compõem a metodologia para obtenção do trabalho final:

- Realização de mapeamentos e revisões sistemáticas;
- Estudos sobre tecnologias IoT e ambientes inteligentes;
- Identificação de tecnologias IoT para gerenciamento de recursos hospitalares;
- Identificação de potenciais aplicações de IoT em ambientes hospitalares locais;
- Definição da arquitetura para o trabalho proposto;
- Seleção de aplicações para testes em ambiente simulados;
- Construção e simulação do ambiente de teste;
- Realização de simulações para validação da proposta;
- Análise dos resultados no ambiente de simulação;

1.5 Organização do Trabalho

Este trabalho está organizado em 5 (cinco) capítulos conforme explicados à seguir:

- No [Capítulo 2](#) é apresentado a fundamentação teórica para embasamento deste trabalho;
- No [Capítulo 3](#) é apresentado o mapeamento sistemático com a definição do planejamento, fases e questões de pesquisa adotados para realização deste trabalho;
- No [Capítulo 4](#) é apresentado o experimento realizado no ambiente de simulação;
- No [Capítulo 5](#) é apresentado a conclusão da solução proposta;

2

Fundamentação Teórica

Neste capítulo, é apresentada a fundamentação teórica necessária para a realização deste trabalho. A seguir serão detalhados os conceitos referentes a Internet das Coisas, *Fog Computing*, *Smart Healthcare (e-health)* e Gerenciamento de recursos hospitalares.

2.1 Internet das Coisas

O termo Internet das Coisas (ou IoT, na sigla em inglês para *Internet of Things*) é creditado a Kevin Ashton porque, em 1999, ele iniciou uma apresentação intitulada “That ‘Internet of Things’ Thing”. A partir daí, enormes contribuições, como segurança, conectividade, eficiência energética e muito mais, foram feitas nessa área, convergindo diferentes visões e nomenclaturas para o termo Internet das Coisas. A IoT pode ser descrita como um termo sofisticado para um cenário onde qualquer coisa pode ser inserida em uma rede, ser identificada de maneira única e interagir com o mínimo de intervenção humana. Essas coisas podem pertencer ao mundo real (coisas físicas) tanto partes inanimadas quanto a animais vivos, ou ao mundo virtual (“coisas” virtuais) que só existe em um ambiente de simulação. Para esclarecer, coisa é um dispositivo comum que pode ser identificado de maneira única e conectado à Internet. Então, se os usuários ou aplicativos têm acesso às informações e se comunicam com essas coisas (objetos) através da Internet, pode ser considerado um cenário de IoT (CRUZ et al., 2018).

Segundo Tejonidhi, Sunitha e Swathi (2019), a Internet das Coisas (IoT) é uma rede de dispositivos, composta por celulares, eletrodomésticos e veículos, que permite a troca de informações e a interação entre si. A IoT permite que esses dispositivos sejam monitorados e controlados remotamente pela Internet. A IoT encontrou vastas aplicações no campo de espaços de consumo, comerciais, industriais e de infraestrutura, abrangendo uma ampla gama de domínios de residências, cidades, meio ambiente, agricultura, saúde, etc.

Segundo Villegas-Ch et al. (2019), a Internet deixou de ser uma rede de computadores e

servidores para incluir uma variedade de dispositivos que interagem entre si e com os usuários. Os dispositivos têm a capacidade de gerar dados e transferi-los por meio de uma rede automaticamente, sem a indispensável interação de pessoas ou computadores. Todos os dispositivos que compõem a IoT capturam dados do ambiente real e os enviam para serem processados e proporcionam uma melhor experiência do usuário. O principal objetivo da IoT é a digitalização do mundo físico para que todos os dispositivos tradicionais estejam conectados à rede e sincronizados entre si, proporcionando um serviço melhor e eficiente ao usuário.

A Internet das Coisas promove a integração efetiva do mundo real e do mundo digital, permitindo que os objetos da vida cotidiana sejam conectados em qualquer lugar a qualquer momento, para interagir uns com os outros e trocar dados e conhecimentos (AMADEO et al., 2019). A IoT fornece aos usuários uma variedade de serviços inteligentes tais como o monitoramento do ambiente e as tecnologias de detecção pessoal. Ele coleta informações por meio da identificação por radiofrequência, NFC (comunicação de campo próximo), ZigBee e outras redes sem fio e com fio, e estabelece um ambiente de Internet, a partir dos três níveis “sensoriamento, rede e aplicação”, para promover um design de ambiente de vida inteligente e automatizado (CHANG; LAI, 2020).

A IoT visa tornar nossa vida cotidiana mais confortável por meio de dispositivos equipados com recursos computacionais e de comunicação (BONAFINI et al., 2019). Como os ambientes baseados em IoT permitem a interação humana com os objetos e entidades do mundo físico, ele encontrou suas aplicações em quase todas as esferas de um ecossistema de cidade inteligente. Uma dessas áreas de aplicação que cresceu nos últimos anos é a saúde. O crescimento das TIC e tecnologias relacionadas levou ao surgimento de um ecossistema de saúde inteligente, onde pacientes, médicos, equipamentos médicos e hospitais estão interconectados para criar um sistema de informações médicas inteligente (TRIPATHI; AHAD; PAIVA, 2020).

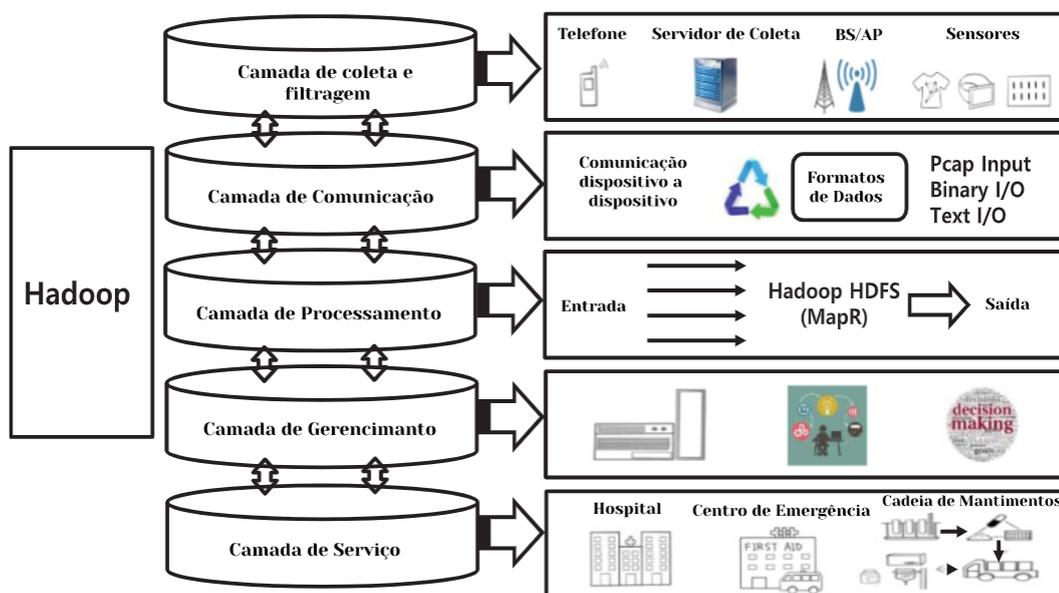
Segundo Tejonidhi, Sunitha e Swathi (2019), os recursos fundamentais de uma arquitetura IoT que geram bons resultados são funcionalidade, escalabilidade, disponibilidade e facilidade de manutenção. Na Figura 1 é apresentada pelo autor Rathore et al. (2017) uma solução para sistemas de saúde em IoT. Ela consiste em cinco camadas funcionais: (1) uma camada de coleta de dados, (2) uma camada de comunicação, (3) uma camada de processamento, (4) uma camada de gerenciamento, e (5) uma camada de serviço.

Na arquitetura de Rathore et al. (2017), a camada de coleta de dados (camada 1) fornece funcionalidade de coleta de dados, incluindo detecção de dados usando sensores médicos, aquisição de dados e armazenamento em *buffer* de dados.

A camada de comunicação (camada 2) oferece conectividade de ponta a ponta aos diversos dispositivos envolvidos nas aplicações de saúde. É responsável pela transmissão dos dados dos sensores para o coordenador por meio de tecnologia *Zigbee* e *Bluetooth*.

A camada de processamento (camada 3) é um componente fundamental que recebe os

Figura 1 – Arquitetura IoT de 5 (cinco) camadas



Fonte: Adaptado de (RATHORE et al., 2017)

arquivos da unidade de coleta. Ele processa os dados enquanto executa os cálculos estatísticos necessários com base na natureza dos dados. Em seguida, os resultados de cada peça são adquiridos novamente e mantidos na área de armazenamento de resultados para análise futura.

A camada de gerenciamento (camada 4) é a camada inteligente do sistema, composta por vários sistemas médicos especialistas que examinam os resultados da camada de processamento e recomendam as ações correspondentes.

Finalmente, a camada de serviço (camada 5) fornece conectividade ao usuário final para acessar várias instalações, como hospitais (por exemplo, suporte médico remoto, exames médicos de rotina), tratamento de emergência (por exemplo, pressão alta), ambulâncias (por exemplo, pressão arterial abaixo de um limite pré-definido, o que pode causar problemas fatais) e delegacias de polícia. Além disso, os médicos também podem monitorar o paciente por meio da análise contínua de seu histórico médico. Esses serviços permitem que um médico se conecte a uma instalação para obter o estado de saúde atual de um paciente.

2.2 Fog Computing

Segundo Subahi (2019) a ascensão da computação em nuvem e da *Fog Computing* trouxe muitos benefícios que incentivaram o setor de saúde a se envolver e adotar esses tipos de tecnologias, juntamente com seus benefícios, para fornecer serviços de saúde modernizados. Assim como a computação em nuvem, a *Fog Computing* também surgiu e se difundiu amplamente para atender a diversos requisitos, como confiabilidade, segurança, latência e eficiência de desempenho trazendo uma nova estratégia de solução de problemas e tratamento de aplicações,

processamento de dados em pequenos servidores de borda, localizados na borda da rede ou em um data center remoto.

Segundo Kim e Lim (2019), os dados de detecção são encaminhados para os servidores de nuvem de borda em vez de diretamente para os servidores de nuvem centrais. A lógica dessa arquitetura é otimizar os sistemas de computação em nuvem, levando o controle de aplicativos de computação, dados e serviços do núcleo para a borda.

Segundo Gupta et al. (2017) a *Fog Computing* é um paradigma de computação distribuída que estende os serviços fornecidos pela nuvem até a borda da rede. Ela permite o aproveitamento contínuo de recursos de nuvem e de borda junto com sua própria infraestrutura, facilita o gerenciamento e a programação de serviços de computação, rede e armazenamento entre data centers e dispositivos finais. Oferece também suporte à mobilidade, heterogeneidade de recursos e interface, interação com a nuvem e análise de dados distribuídos para atender aos requisitos de aplicativos que precisam de baixa latência com uma distribuição geográfica ampla e densa.

Segundo Shukla et al. (2019), a *Fog Computing* tem seis dimensões, são elas: aplicativos distribuídos geograficamente, baixa latência, serviços personalizados baseados em localização, componente de controle local, borda mais próxima do usuário e comunicação máquina para humano. Essas dimensões a tornam mais útil e proposital para os requisitos dos usuários finais na IoT de saúde.

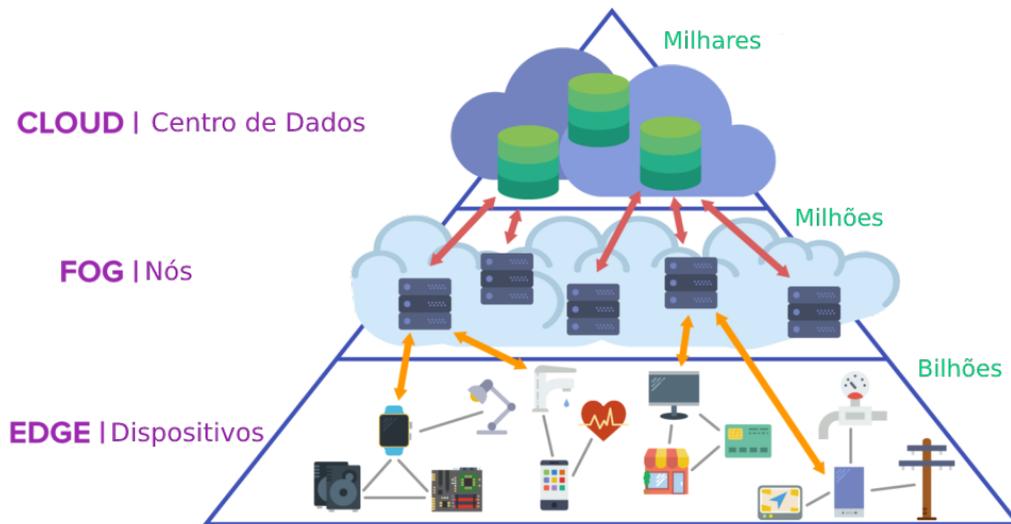
Segundo Gupta et al. (2017), o primeiro benefício é a redução do tráfego de rede, pois o aumento descontrolado do tráfego de rede pode levar ao congestionamento e resultar em aumento da latência. Ela fornece uma plataforma para filtragem e análise dos dados gerados pelos sensores utilizando recursos de dispositivos de borda. Isso reduz drasticamente o tráfego enviado para a nuvem, permitindo a colocação de operadores de filtragem próximos à fonte de dados. A redução considerável na latência de propagação é a próxima vantagem importante da utilização do paradigma de *Fog Computing*, especialmente para aplicativos de missão crítica que exigem processamento de dados em tempo real.

Como mostra a Figura 2, é na camada *Edge* que estão localizados os sensores. Dessa forma, bilhões de dispositivos captura dados do ambiente. Na camada intermediária, encontram-se milhões de nós *Fog*. Essa camada intermediária, possui infraestrutura descentralizada e distribuída, é responsável por minimizar o volume de dados enviados para a nuvem, conservando a largura de banda e minimizando a latência das aplicações. Por fim, a camada de *Cloud Computing*, onde está localizado milhares de data centers com alto poder de processamento.

2.2.1 iFogSim

O simulador usado para modelar e avaliar computação em nuvem e *Fog Computing* continuamente deve ter certas características e recursos, como medição de latência, consumo de energia e uso de largura de banda de rede para representar totalmente ambientes realistas que

Figura 2 – Relação de dispositivos na camada Edge, Fog e Cloud Computing



Fonte: Adaptado de (LOPES, 2018)

refletem as saídas em um ambiente complexo (LAWAL; OLANIYI; GIBSON, 2021).

O iFogSim é um conjunto de ferramentas de software de simulação baseado em tecnologia Java projetado para experimentos de cenário de ambiente de computação *Fog* na avaliação de gerenciamento de recursos, políticas de agendamento, incluindo impacto no consumo de energia, latência e medição de uso de largura de banda da rede. Suporta várias classes Java que correspondem a aplicativos, sensores, atuadores, tuplas e dispositivos *Fog*, nós *Fog* e nós heterogêneos de grande escala. Além disso, o iFogSim foi estendido para otimizar os posicionamentos de dados em dispositivos IoT e nós *Fog* para garantir o gerenciamento de posicionamentos de dados em relação a vários objetivos, como melhorar o consumo de energia, latência e uso de largura de banda da rede (LAWAL; OLANIYI; GIBSON, 2021).

A estrutura do iFogSim foi projetada de forma a torná-la capaz de avaliar as políticas de gerenciamento de recursos aplicáveis a ambientes *Fog* com relação ao seu impacto na latência, consumo de energia, congestionamento de rede e custos operacionais. Ele simula dispositivos de borda, data centers em nuvem e links de rede para medir as métricas de desempenho. Nesses modelos, sensores publicam dados em redes IoT, aplicativos executados em dispositivos *Fog* assinam e processam dados provenientes de sensores e, finalmente, os insights obtidos são traduzidos em ações encaminhadas aos atuadores (GUPTA et al., 2017).

O iFogSim tem como principal objetivo fornecer uma ferramenta que simule ambientes IoT e *Fog* em grande escala, permitindo a modelagem de componentes como sensores, atuadores, dispositivos *Fog* e *Cloud*. Esse *framework* possibilita aos usuários a implementação de soluções próprias para o gerenciamento de recursos, através da modelagem e verificação de políticas de alocação de serviços (FAMA; SANTOS; PERKUSICH, 2018).

2.3 *Smart Healthcare (e-Health)*

Nos últimos tempos, os profissionais e estabelecimentos na área de saúde estão aproveitando cada vez mais as tecnologias de informação para fornecer sistemas inteligentes que visam acelerar o diagnóstico e o tratamento de saúde. Tais sistemas fornecem serviços inteligentes de monitoramento de saúde e automação médica em diferentes contextos e ambientes (hospitais, consultórios, casa, em movimento...), permitindo assim uma redução substancial dos custos da consulta médica e uma melhoria geral da qualidade do cuidado do paciente (GRECO et al., 2020).

Segundo Tian et al. (2019), *Smart Healthcare* nasceu do conceito de “Smart Planet” proposto pela IBM (Armonk, NY, EUA) em 2009. Simplificando, *Smart Planet* é uma infraestrutura inteligente que usa sensores para perceber informações, transmitir informações através da Internet das Coisas (IoT) e processar as informações usando supercomputadores e computação em nuvem. Pode coordenar os sistemas sociais e integrá-los para realizar a gestão dinâmica e refinada da sociedade humana. Nesse sentido, *Smart Healthcare* é um sistema de serviço de saúde que usa tecnologia como dispositivos vestíveis, IoT e internet móvel para acessar dinamicamente informações, conectar pessoas, materiais e instituições relacionadas à saúde e, em seguida, gerencia ativamente e responde às necessidades do ecossistema médico em uma maneira inteligente. Ainda segundo Tian et al. (2019), a saúde inteligente pode promover a interação entre todas as partes no campo da saúde, garantir que os participantes recebam os serviços de que precisam, ajudar as partes a tomar decisões informadas e facilitar a alocação racional de recursos.

Segundo Sanchez-Gallegos et al. (2020), os dispositivos IoT de saúde representavam 40% do total de dispositivos IoT. Esta tecnologia também está se tornando fundamental para que os usuários desses dispositivos adquiram dados importantes (ou seja, sinais vitais e nível de atividade) e tomem decisões para melhorar seus cuidados de saúde. Além disso, usuários, profissionais de saúde e organizações podem obter insights para melhorar os procedimentos de tomada de decisão usando e processando dados extraídos de dispositivos de IoT e também de outras fontes, como imagens médicas, estudos, documentos históricos de expedientes e registros de saúde.

Vários dispositivos médicos, sensores e dispositivos de diagnóstico e imagem podem ser vistos como dispositivos inteligentes ou objetos que constituem uma parte central da IoT. Os serviços de saúde baseados em IoT devem reduzir custos, aumentar a qualidade de vida e enriquecer a experiência do usuário. Da perspectiva dos provedores de saúde, a IoT tem o potencial de reduzir o tempo de inatividade do dispositivo por meio do fornecimento remoto. Além disso, a IoT pode identificar corretamente os tempos ideais para reabastecimento de suprimentos para vários dispositivos para sua operação suave e contínua. Além disso, a IoT fornece o agendamento eficiente de recursos limitados, garantindo seu melhor uso e atendimento a mais pacientes (ISLAM et al., 2015).

A IoT fornece a base para uma saúde inteligente com a assistência de sensores heterogêneos, como sensor de batimento cardíaco, sensor de temperatura, glicosímetro e assim por diante. A principal obrigação da IoT é a capacidade de colocar o sensor em todos os tipos de pontos para coletar dados (DIN; PAUL, 2019). Utilizando as informações coletadas e a análise de dados auxiliando na tomada de decisões que percebem informações para diferentes pessoas, tal inovação poderia criar um efeito drástico nos serviços médicos em todo o mundo e definitivamente reduzir os custos dos serviços humanos e melhorar a velocidade e análise (ADARSH; VIJAYAKUMAR, 2019).

2.4 Gerenciamento de Recursos Hospitalares

Segundo Oueida et al. (2018), o campo da saúde inteligente surge da necessidade de melhorar a gestão do setor de saúde, utilizar melhor seus recursos e reduzir seus custos, mantendo ou mesmo melhorando seu nível de qualidade. Os recursos no setor de saúde podem ser classificados em recursos consumíveis e recursos não consumíveis. Os recursos consumíveis incluem aqueles recursos que se deterioram e expiram com o tempo, como todos os recursos e ferramentas médicas. Os recursos não consumíveis, por outro lado, são aqueles recursos que não expiram com o tempo. Entre os recursos não consumíveis encontram-se os recursos humanos como médicos, enfermeiros e todo o capital humano envolvido no processo de assistência à saúde.

Segundo Oueida et al. (2018), a pesquisa em saúde inteligente pode ser classificada em duas categorias principais: categoria relacionada ao paciente e categoria relacionada ao processo. A categoria relacionada ao paciente inclui, mas não se limita a, pesquisas que visam dispositivos vestíveis para coletar dados sobre pacientes a serem relatados a instituições médicas. A segunda categoria é a pesquisa relacionada ao processo. Nesta categoria, a pesquisa preocupa-se com o aprimoramento de políticas que assegurem diversos aspectos no setor de saúde. Entre esses aspectos estão a utilização de recursos, programação de recursos, qualidade de serviço e muitos outros aspectos relacionados à definição e gerenciamento de processos.

Uma das razões para aumentar o nível de automação em tarefas de baixa e média complexidade é reduzir o risco de erros cometidos pelos funcionários. Em setores como a saúde, isso é especialmente crítico, pois tem um impacto direto no estado de saúde dos pacientes. Por exemplo, erros de medicação são responsáveis por 22.300 mortes por ano no Serviço Nacional de Saúde do Reino Unido (NHS - *National Health System*), tendo também um impacto econômico significativo (LÓPEZ et al., 2018).

No trabalho clínico de prevenção de SRAGs (Síndrome Respiratória Aguda Grave), leitores de RFID (*Radio-Frequency Identification*) podem ser instalados em robôs e adesivos UHF (*Ultra High Frequency*) podem ser lidos quando um medicamento é carregado no robô para confirmar a distribuição do medicamento. Uma vez que um robô ativa seu leitor RFID,

ele pode obter as informações adequadas e a quantidade de todos os medicamentos em uma gaveta e, por meio da combinação precisa das informações dos medicamentos, um medicamento pode ser enviado para as unidades médicas de infecção respiratória aguda grave corretamente. O Geisinger Medical Center em Danville, Pensilvânia, EUA, adotou robôs com leitores RFID embarcados para garantir que um medicamento seja realmente entregue a todas as unidades de uso, com imagens transmitidas instantaneamente. A tecnologia RFID também pode ser usada para construir um sistema de gerenciamento de resíduos médicos, que pode monitorar e rastrear facilmente todo o processo de resíduos médicos da infecção respiratória aguda grave, incluindo geração, reciclagem, transporte e tratamento (SONG et al., 2020).

2.5 Considerações Finais

Este capítulo apresentou a fundamentação teórica que forneceu embasamento para este trabalho de mestrado. Foram abordados os conceitos de Internet das Coisas, *Fog Computing* e a ferramenta de simulação iFogSim utilizada para análise. Em seguida, foram apresentados o conceito, assim como alguns avanços da saúde inteligente (*smart healthcare*) e suas vantagens na gestão de recursos.

3

Revisão da Literatura

Como indicado na [seção 1.4](#), um dos passos iniciais para realização deste trabalho foi a busca por trabalhos na literatura científica que abordassem a temática de IoT na área de saúde. Desta forma, procedeu-se à realização de um mapeamento sistemático da literatura conforme apresentado neste capítulo.

3.1 Mapeamento Sistemático

Segundo [Falbo \(2013\)](#), um Mapeamento Sistemático é uma revisão ampla dos estudos primários existentes em um tópico de pesquisa específico que visa identificar a evidência disponível nesse tópico. O método de pesquisa desse trabalho baseou-se na metodologia definida por [Kitchenham e Charters \(2007\)](#) seguindo o processo que envolve três fases principais: Planejamento da Revisão, Condução da Revisão e Publicação dos Resultados.

Com isso, este capítulo foi subdividido em quatro partes com atividades definidas na fase de planejamento da revisão: questões de pesquisa, estratégias de busca e seleção das fontes, critérios de inclusão e exclusão, e resultados.

3.1.1 Questões de Pesquisa

As seguintes questões de pesquisa foram definidas para alcançar os objetivos desse estudo:

- **QP1.** Quais aplicações são propostas para uso da Internet das Coisas na área da saúde?
- **QP2.** Quais arquiteturas são utilizadas para uso da Internet das Coisas na área da saúde?

3.1.2 Estratégia de Busca e Seleção de Fontes

Para responder às questões de pesquisa do tópico anterior definimos cinco fontes de pesquisa indicadas por pesquisadores e profissionais experientes da área conforme mostrado na [Tabela 1](#).

Tabela 1 – Fontes de pesquisa utilizadas no mapeamento sistemático

FONTE DE PESQUISA	URL
ScienceDirect	https://www.sciencedirect.com/
IEEE Xplore	https://ieeexplore.ieee.org/
ACM Digital Library	https://dl.acm.org/
Web of Science	https://www.isiknowledge.com
Scopus	https://www.scopus.com

Fonte: Autor.

Além disso, a seguinte *string* de busca foi formulada utilizando palavras-chave que remetem ao tema proposto: (*IoT OR “internet of things”*) *AND* (*processor OR architecture OR hardware*) *AND* *health**

Após aplicar a *string* de busca acima em todas as fontes de pesquisa citadas na [Tabela 1](#), foram encontrados um total de 1631 (um mil seiscentos e trinta e um) artigos. A [Tabela 2](#) mostra a relação da quantidade de artigos encontrados e suas respectivas fonte de pesquisa.

Tabela 2 – Relação da quantidade de artigos e suas respectivas fontes de pesquisa

FONTE DE PESQUISA	ARTIGOS
ScienceDirect	159
IEEE Xplore	403
ACM Digital Library	574
Web of Science	252
Scopus	243
TOTAL	1631

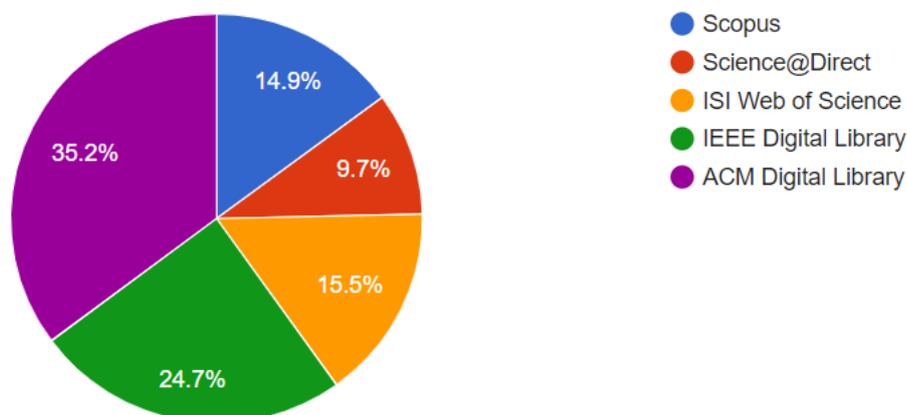
Fonte: Autor.

A [Figura 3](#) mostra a porcentagem dos dados listados na [Tabela 2](#).

3.1.3 Critérios de Seleção

Foi definido que seriam selecionados apenas os artigos do período de tempo entre os anos de 2015 a 2020 (devido ao mapeamento ter sido feito no início de 2021), que estejam em inglês ou português e também com acesso aberto ou disponível no portal periódicos CAPES que contiver em seu título, resumo ou palavras-chave os termos da *string* de busca. A [Tabela 3](#) descreve os critérios de inclusão e exclusão utilizados na seleção dos artigos.

Figura 3 – Porcentagem de artigos por fonte de dados



Fonte: Autor.

Após aplicação dos critérios de inclusão e exclusão, 147 artigos foram selecionados para leitura. Na fase de leitura, mais 43 artigos foram excluídos por não atenderem aos critérios mencionados, restando 104 artigos selecionados. A distribuição de estudos selecionados é exibida na [Tabela 4](#).

3.1.4 Análise de Resultados

Respondendo à **QP1** mencionada na [subseção 3.1.1](#), a [Figura 4](#) mostra a porcentagem das aplicações separadas por categorias específicas: propostas de arquitetura/modelo em geral (com ou sem aplicação específica); monitoramento/coleta de dados vitais ou do ambiente; segurança; eficiência energética; detecção/precisão/previsão de doenças; rastreamento/localização de ativos ou pessoal; e outras categorias. Tais aplicações podem se encaixar em mais de uma categoria.

Na categoria arquitetura ou modelo foram categorizados 37 artigos; monitoramento ou coleta de dados, 47 artigos; segurança, 22 artigos; eficiência energética, 10 artigos; detecção ou precisão ou previsão de doenças, 5 artigos; 4 (quatro) artigos foram categorizados como rastreamento ou localização de ativos ou pessoal; e 9 artigos se encaixaram em outras categorias.

Respondendo à **QP2** mencionada na [subseção 3.1.1](#), a [Figura 5](#) mostra um resumo das arquiteturas em camadas encontradas nas aplicações de saúde IoT. Conforme a [Figura 5](#), a maioria das aplicações possuem 3 (três) camadas. As três camadas comumente utilizadas pelos autores são: camada de infraestrutura (sensores/captura dos dados), camada de processamento (computação em borda ou em nuvem) e camada de comunicação (redes com fio e sem fio).

Foi possível identificar um autor que utilizou 2 camadas; 40 (quarenta) autores definiram suas arquiteturas em 3 (três) camadas; 26 (vinte e seis) autores utilizaram arquitetura em 4 (quatro) camadas; 13 (treze) autores utilizaram arquitetura em 5 (cinco) camadas; e 18 (dezoito) autores não informaram de forma clara a quantidade de camadas utilizadas.

Tabela 3 – Critérios de inclusão e critérios de exclusão aplicados nos artigos

TIPO	ID	DESCRIÇÃO
Critérios de Inclusão	CI1	O título do artigo mencionar IoT (<i>Internet of Things</i>)
	CI2	O título do artigo mencionar Processador, Arquitetura ou Hardware
	CI3	O título do artigo mencionar Saúde
	CI4	O abstract mencionar IoT (<i>Internet of Things</i>)
	CI5	O abstract mencionar Processador, Arquitetura ou Hardware
	CI6	O abstract mencionar Saúde
	CI7	O artigo deve mencionar os tipos de processadores utilizados na Internet das Coisas na área da saúde
	CI8	O artigo deve mencionar as arquiteturas utilizadas na Internet das Coisas na área da saúde
	CI9	O artigo deve mencionar aplicações de Internet das Coisas na área da saúde
	CI10	O artigo deve mencionar as vantagens e desvantagens das tecnologias usadas com Internet das Coisas na área da saúde
Critérios de Exclusão	CE1	Trabalhos duplicados
	CE2	Trabalhos que não estejam disponíveis integralmente nas bases de dados
	CE3	Trabalhos cujo idioma não seja inglês ou português
	CE4	Trabalhos que não atender aos critérios de inclusão

Fonte: Autor.

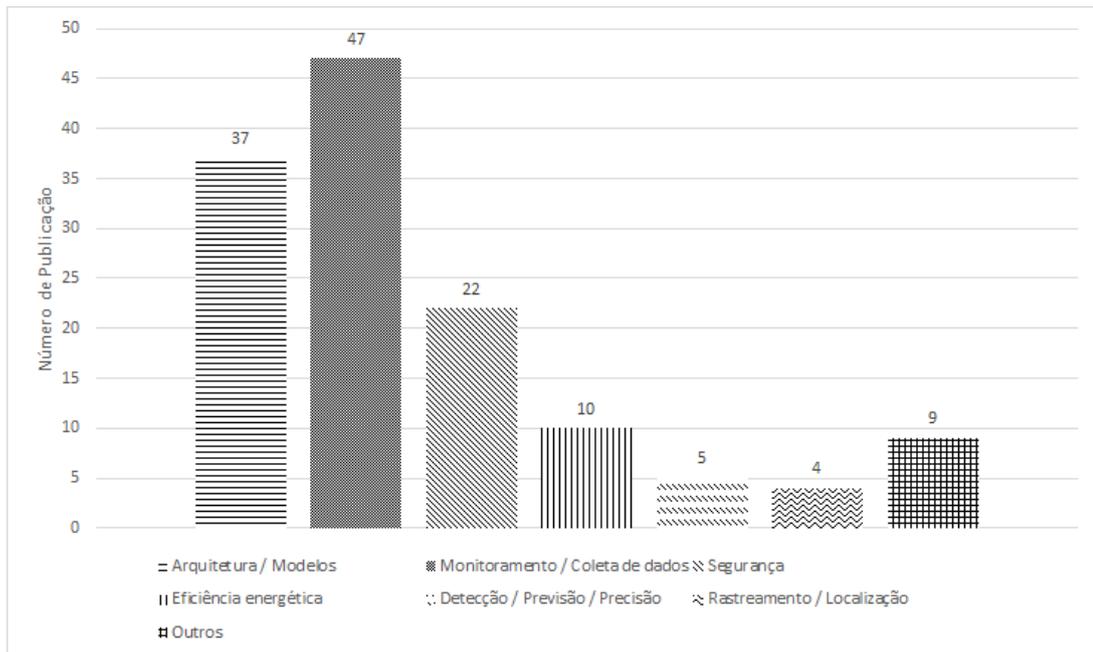
Tabela 4 – Relação de artigos selecionados por fonte de pesquisa

FONTE DE PESQUISA	ARTIGOS	SELECIONADOS
ScienceDirect	159	4
IEEE Xplore	403	11
ACM Digital Library	574	14
Web of Science	252	21
Scopus	243	54
TOTAL	1631	104

Fonte: Autor.

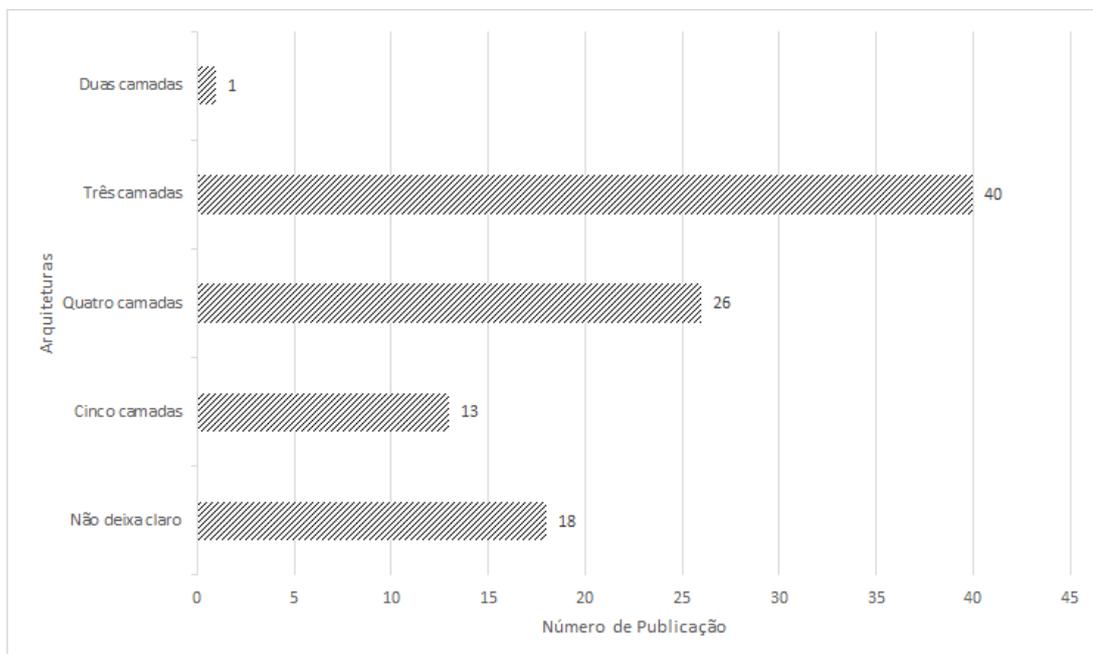
Com diversos autores utilizando arquiteturas diferentes e com definições não homogêneas, o autor [Venčkauskas et al. \(2016\)](#) definiu a arquitetura utilizada em 2 camadas separando apenas como camada superior e camada inferior. Já o autor [Shi-Lin et al. \(2015\)](#) definiu a arquitetura em 3 camadas chamadas de camada de percepção, camada de rede e camada de aplicação; enquanto o autor [Rubí e Gondim \(2019\)](#) definiu as 3 camadas como camada de integração de dispositivos, camada de integração de dados e camada de visualização de dados ou extração de conhecimentos. Para 4 camadas, o autor [Ungurean e Brezulianu \(2017\)](#) definiu as camadas como sensoriamento, *gateway de middleware*, armazenamento e aplicativo. Já o autor [Subahi \(2019\)](#), definiu as 4

Figura 4 – Categorias de aplicações IoT na área da saúde



Fonte: Autor.

Figura 5 – Arquiteturas em camadas das aplicações IoT em Saúde



Fonte: Autor.

camadas como sensibilidade de saúde, névoa, comunicação e camada de nuvem. Em [Tseng, Wu e Lai \(2019\)](#), as 5 camadas foram definidas como registro do usuário, consulta de dados, plataforma de gerenciamento, coleta de dados do sensor e coleta de dados abertos.

3.2 Análise dos Trabalhos Relacionados

A seguir, cada parágrafo descreve resumidamente alguns trabalhos selecionados utilizando-se os critérios de trabalhos que mencionaram gerenciamento de recursos hospitalares; trabalhos que agregaram algum princípio de segurança de aplicações, como confidencialidade e integridade; e também aqueles que utilizaram a tecnologia *Cloud* ou *Fog Computing* para agregar principalmente escalabilidade, disponibilidade e menor latência das aplicações.

Com o principal objetivo de diminuir o tempo que os enfermeiros levam para prestar cuidados de saúde aos pacientes, os autores [Kanan e Elhassan \(2016\)](#), desenvolveram um sistema, nos Emirados Árabes Unidos, de chamada de enfermagem hospitalar que combina dois tipos de sistemas de localização interna. O objetivo do primeiro sistema é localizar pacientes, enquanto o segundo é localizar enfermeiras equipadas com seus smartphones. Em momentos de urgência, o sistema realiza uma comparação entre as coordenadas do enfermeiro e do paciente, e o enfermeiro mais próximo recebe um alarme.

Já o estudo realizado na China por [Shi-Lin et al. \(2015\)](#), desenvolveu um *gateway* IoT para gerenciamento de parâmetros fisiológicos utilizando tecnologias de detecção de portadora, rede automática e programação multi-thread. O *gateway* da Internet das Coisas (IoT) é uma parte importante de todos os tipos de aplicativos IoT na indústria de saúde. O *gateway* de hardware e o servidor de *gateway* possui funções de tradução de protocolo, gerenciamento de roteamento e controle de segurança de dados. O *gateway* de parâmetros fisiológicos, que é o dispositivo de aplicação chave na enfermagem de IoT médica, é projetado para cumprir a coleta inteligente e convergência abrangente de parâmetros fisiológicos diários.

O trabalho realizado na Índia por [Adarsh e Vijayakumar \(2019\)](#), descreve o uso da IoT em dados de assistência à saúde de uma forma clássica para coletar dados de múltiplos sensores e filtrar informações úteis para identificar o estado de saúde de uma pessoa. Além disso, os dados de saúde são compartilhados em uma plataforma segura e utiliza nós autenticados dentro de sua infraestrutura IoT.

Já o trabalho de [López et al. \(2018\)](#), descreve um sistema RFID básico para rastreamento e gerenciamento de ativos em hospitais, com o objetivo de fornecer detalhes adicionais sobre os aspectos de implementação que devem ser considerados para garantir a funcionalidade adequada do sistema. Embora o escopo do sistema RFID descrito nesta contribuição seja restrito a uma pequena área do hospital, a arquitetura é totalmente escalável para cobrir as necessidades dos diferentes serviços médicos no hospital. A tecnologia RFID de ultra-alta frequência (UHF) é selecionada em vez da mais estendida tecnologia RFID de comunicação de campo próximo (NFC) e de alta frequência (HF) para minimizar a infraestrutura de hardware. Em particular, UHF RFID também torna a área de cobertura/leitura mais fácil usando diferentes tipos de antenas. As informações são armazenadas em um banco de dados, que é acessado a partir de dispositivos móveis do usuário final (*tablets, smartphones*) onde são exibidos a posição e o status dos ativos

a serem rastreados. O RFID demonstrou ser de grande ajuda para melhorar o rastreamento de pacientes, medicamentos e ativos médicos em hospitais, onde a digitalização dessas operações melhora sua eficiência e segurança.

O estudo de [Lachtar, Val e Kachouri \(2020\)](#), teve como objetivo desenvolver um sistema especial de rastreamento e monitoramento de idosos, conectado diretamente a uma bengala. O projeto proposto envolve um nó transmissor com invólucro, permitindo enviar dados relacionados à posição e ao estado dos idosos para uma estação base, por meio da implementação da tecnologia LoRa (protocolo de comunicação de baixo consumo de energia e longo alcance para dispositivos de radiofrequência). O último, então, aplicaria o protocolo de transporte de telemetria de enfileiramento de mensagens (MQTT) para interagir com o ambiente assim que comprovar a ocorrência de uma queda.

No artigo de [Polenta et al. \(2020\)](#), é apresentado um sistema chamado BubbleBox que conta com um dispositivo dedicado para realizar o rastreamento de contatos. O BubbleBox é um conjunto de tecnologias IoT e de *softwares* integrados para detectar e limitar novos surtos de infecção COVID-19 por meio de rastreamento de contato. O sistema é composto por um dispositivo vestível (uma pulseira) para rastrear anonimamente todos os contatos sob a distância social segura entre a pessoa que a usa e as demais pessoas; um aplicativo web opcional que permite aos usuários emparelhar seu dispositivo com sua identidade, usando um *smartphone*, *tablet* ou PC; um *backend* de dados, que armazena esses dados para disponibilizá-los ao pessoal de saúde autorizado e, de forma anônima, monitorar a propagação da infecção. A pulseira é responsável por rastrear os contatos - ela gera uma identificação única e, graças ao seu módulo Wi-Fi, é capaz de detectar outras pulseiras BubbleBox dentro de 20m.

Em [Kim e Lim \(2019\)](#), é apresentado uma arquitetura de detecção de dados corporais baseada em nuvem para computação de inteligência artificial. A primeira camada é semelhante à rede de área corporal sem fio que fornece um monitoramento contínuo da saúde das pessoas sem qualquer restrição em suas atividades normais de vida diária. Os dados de detecção são encaminhados para os servidores de nuvem de borda em vez de diretamente para os servidores de nuvem centrais. A lógica de ter a arquitetura de três camadas é otimizar os sistemas de computação em nuvem, levando o controle de aplicativos de computação, dados e serviços do núcleo para a borda. Na arquitetura proposta, o volume de dados que deve ser transferido dos dispositivos sensores para o servidor de nuvem central é reduzido com a implantação de dispositivos de borda entre os dispositivos de sensor e o servidor de nuvem central.

Em [Subahi \(2019\)](#), é apresentado uma arquitetura conceitual, design e recomendações para o sistema de gerenciamento de saúde baseado no IoT Edge. A arquitetura sugerida visa distribuir a carga de trabalho de desempenho do sistema (serviços eletrônicos de saúde), incluindo monitoramento, diagnóstico, previsão, bem como gerenciamento e arquivamento de dados médicos de pacientes em diferentes pontos do sistema (na borda e na nuvem).

Em [Saha et al. \(2019\)](#), é apresentado uma estrutura de e-saúde que trata de registros

médicos eletrônicos (RMEs) que preserva as questões de privacidade. A nova abordagem baseada em consenso usada na estrutura garantiu a confiabilidade do solicitante para visualizar o RME.

Em Hou e Yeh (2015), foi proposto uma arquitetura de comunicação baseada em sensores (ou tags de sensores) para futuros sistemas de serviços de saúde baseados em IoT. Um esquema de autenticação baseado em single sign-on seguro e um protocolo de prova de coexistência robusto para sistemas de saúde baseados em IoT são propostos.

Em Trigo et al. (2020), os autores abordam sobre rastreamento de pacientes em um complexo hospitalar de vários edifícios. Os autores afirmam que os pacientes internados em Unidades de Terapia Intensiva (UTI) são transportados de e para outras unidades. Conhecer sua localização é estratégico para um bom planejamento dos transportes intra-hospitalares, bem como para a gestão de recursos. Embora projetado para o contexto de pacientes internados em UTI, o sistema pode ser facilmente extrapolado para outros casos de uso. Para minimizar as necessidades computacionais do aplicativo, os processos de localização e rastreamento são realizados na infraestrutura em nuvem e o aplicativo se limita a apresentar adequadamente os dados por ele fornecidos.

Em Schenfeld et al. (2017), é apresentado uma arquitetura de sistema para dispositivos IoT capaz de habilitar o processamento de dados nos próprios dispositivos ou o mais próximo deles, criando a camada de Edge e *Fog Computing* que podem ser aplicadas em diferentes domínios, melhorando a Qualidade dos Serviços (QoS) e autonomia na tomada de decisão, mesmo se os dispositivos estiverem temporariamente desconectados da rede (*offline*). Os testes revelaram resultados satisfatórios, melhorando os tempos de conexão, processamento e entrega das informações às aplicações, redução do volume de comunicação entre dispositivos e *core middleware*, além de melhorar a segurança nas comunicações.

3.3 Solução Proposta

A Tabela 5 mostra um resumo dos trabalhos relacionados, fazendo uma classificação quanto aos aspectos de tipo de gerenciamento de recursos hospitalares, quantidade de camadas utilizadas na arquitetura, segurança e tecnologia de *Cloud* e/ou *Fog Computing* utilizadas pelos autores comparando com a solução proposta.

O principal diferencial da solução proposta é adicionar a camada de *Fog Computing* visando minimizar a latência da aplicação na localização interna de equipamentos hospitalares, podendo também ser utilizado na localização de profissionais da saúde para minimizar o tempo de atendimento em situações de emergência. A localização interna de equipamentos baseada em sistemas IoT pode proporcionar uma gestão muito mais eficiente contribuindo para salvar vidas através da agilidade das informações da localização dos equipamentos hospitalares.

Tabela 5 – Resumo dos trabalhos relacionados categorizados quanto ao tipo de gerenciamento de recursos, arquitetura, segurança e tecnologia utilizada pelos autores selecionados

Autor	Gerenciamento de Recursos										
	Localização / Rastreamento			Parâmetros Fisiológicos	Arquitetura			Segurança		Tecnologia	
	Pessoas	Ativos	Contato		3 cam.	4 cam.	5 cam.	Confid.	Integr.	Cloud	Fog
Kanan e Elhassan (2016)	✓					✓					
Shi-Lin et al. (2015)				✓	✓			✓			
Adarsh e Vijayakumar (2019)				✓		✓		✓			✓
López et al. (2018)		✓				✓					
Lachtar, Val e Kachouri (2020)	✓				✓						
Polenta et al. (2020)	✓		✓		✓			✓			
Kim e Lim (2019)					✓						✓
Subahi (2019)						✓					✓
Saha et al. (2019)					✓			✓	✓	✓	✓
Hou e Yeh (2015)					✓			✓	✓	✓	
Solução proposta	✓	✓						✓	✓	✓	✓

Fonte: Autor.

3.4 Considerações Finais

Assim, pretende-se realizar o monitoramento quanto ao uso ou localização de equipamentos ou insumos hospitalares em determinados ambientes hospitalares. Para isso, a revisão sistemática agregou que uma das possíveis soluções é utilizar sensores/leitores de identificador por radiofrequência (RFID) posicionados em pontos estratégicos capaz de cobrir uma determinada área, e TAGs utilizadas em forma de pulseiras, chaveiros, adesivos ou etiquetas fixadas nos equipamentos, que são automaticamente lidas por esses sensores/leitores ao adentrar na região coberta pelo sinal do radio. Os dados lidos e tratados podem ser exibidos em um painel digital, *tablet*, PC ou *smartphone*, ou ficar disponível apenas para os gestores.

4

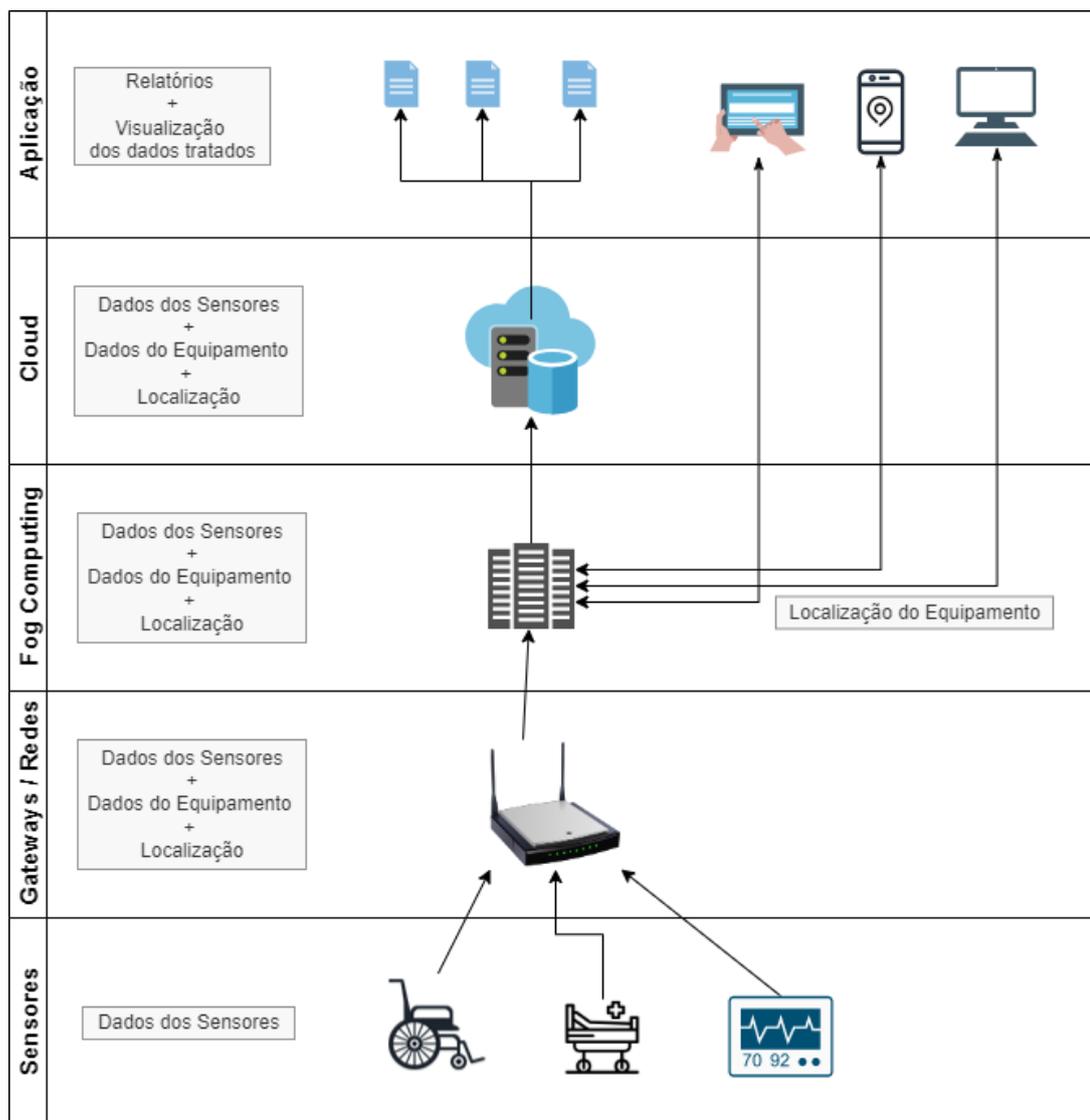
Avaliação da Fog Computing para suporte a sistemas de localização interna

Neste capítulo, apresentamos a descrição da arquitetura de uma aplicação voltada ao gerenciamento da localização de equipamentos hospitalares, que servirá como estudo de caso para a avaliação de questões de desempenho referentes ao uso de *Edge*, *Fog* e *Cloud Computing* em distintas configurações desse ambiente.

A proposta da aplicação de localização interna de equipamentos hospitalares envolve localizar equipamentos móveis (por exemplo, eletrocardiograma, desfibrilador, bomba de oxigênio, monitor cardíaco, cadeira de rodas, entre outros) que se locomovem dentro do hospital entre setores (ala amarela, vermelha, entre outros) e que médicos e enfermeiros necessitam fazer uso desses equipamentos em determinado momento. Essas informações são enviadas de tempos em tempos através de sensores acoplados nesses equipamentos. Em cada setor/sala há um *gateway*, posicionado estrategicamente, responsável por capturar os dados enviados pelos equipamentos ao adentrar numa área de cobertura. Após a captura dos dados, é enviado para um servidor local (*Fog Computing*) com maior capacidade de processamento para tratar as informações recebidas. Ao tratar as informações, os dados são enviados para a nuvem para armazenamento. Na tela de um aplicativo, acessado via dispositivo móvel ou via web em um navegador de um computador pessoal, será mostrado a localização do equipamento. A [Figura 6](#) exemplifica o contexto mencionado.

Para atender à QoS desejada, são necessárias técnicas eficientes de gerenciamento de serviço com reconhecimento de mobilidade. Considerando esta questão, modelamos um estudo de caso de simulação orientada à mobilidade baseado em *Fog Computing* no iFogSim2, pois tal ferramenta tem certas características e recursos, como medição de latência, consumo de energia e uso de largura de banda de rede para representar totalmente ambientes realistas. Os detalhes do modelo de aplicação, parâmetros de simulação, comparando as políticas de gerenciamento de mobilidade e seus desempenhos para este estudo de caso são discutidos a seguir.

Figura 6 – Diagrama da solução proposta



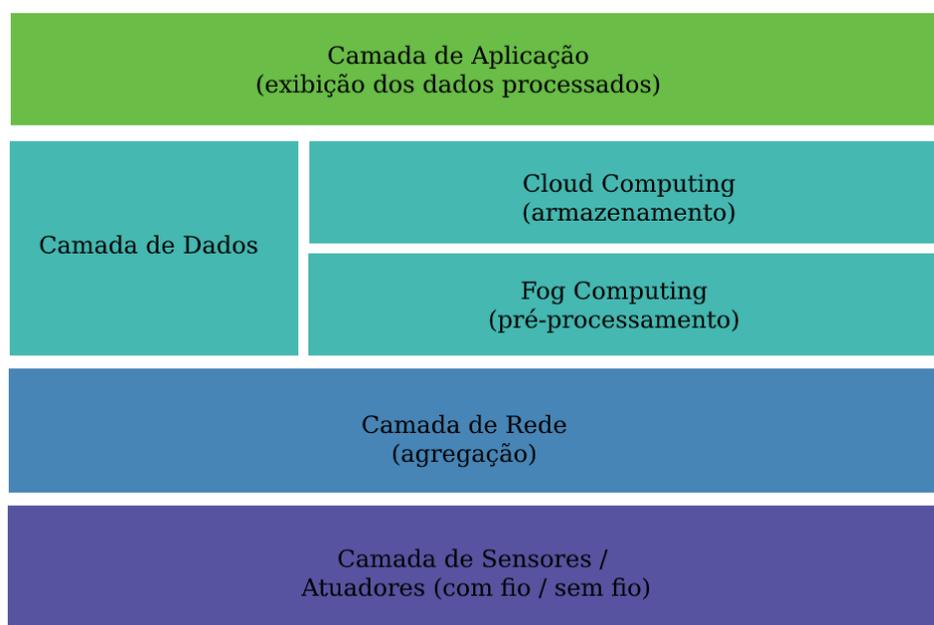
Fonte: Autor.

4.1 Arquitetura da Aplicação

Visando atender o objetivo de agregar confidencialidade e integridade dos dados trafegados e armazenados, como citado na seção 1.3, propõe-se aqui uma arquitetura em cinco camadas (camada de sensores - captura dos dados; camada de rede - comunicação/tráfego dos dados capturados; camada de dados - subdividida em pré-processamento e armazenamento dos dados; e camada de aplicação - camada de acesso aos usuários) com a utilização da *Fog Computing* e *Cloud Computing* conforme mostrado na Figura 7.

Camada de sensores/atuadores - camada responsável por reunir os sensores e atuadores capazes de capturar os dados dos objetos ou ambientes fazendo com que tais objetos passem a existir também no mundo digital. Dentre os dispositivos que podem compor estão os identificadores por radiofrequência (tags RFID), leitores RFID, sensor infravermelho passivo (PIR), sensor

Figura 7 – Proposta da arquitetura em camadas da solução proposta



Fonte: Autor.

de temperatura, sensor de distância ultrassônico, sensor biométrico, entre outros. Através dos atuadores é possível, por exemplo, acender ou apagar a luz, travar ou destravar a porta, ligar ou desligar o ar condicionado, ativar sirenes de alarmes, etc.

Camada de rede - camada responsável por transmitir os dados capturados da camada de sensores/atuadores para outras camadas através de redes com fio ou sem fio como Bluetooth, Wi-Fi, ZigBee, etc. Com a utilização de protocolos de transporte (TLS - *Transport Layer Security*, DTLS – *Datagram Transport Layer Security*) que possuem artifícios de segurança, passa a garantir a confidencialidade e integridade dos dados.

Camada de dados - camada responsável pelo armazenamento e processamento dos dados, subdividida em duas tecnologias: *Cloud Computing* e *Fog Computing*. Com uso da *Cloud Computing*, o armazenamento centralizado permite acessar os dados de forma segura em qualquer lugar e possui alto poder de processamento. Já a *Fog Computing*, é uma infraestrutura descentralizada e distribuída que permite um processamento e análise de dados mais eficiente reduzindo a quantidade de dados enviados para a nuvem, baixa latência e conserva largura de banda da rede.

Camada de aplicação - camada responsável por exibir os dados processados e tratados. Nessa camada será possível exibir relatórios e gráficos específicos para auxiliar os gestores na tomada de decisão através da visualização de gráficos que facilitem, por exemplo, a leitura da temperatura de determinados ambientes que precisam manter-se refrigerados (estoque de vacinas, centro cirúrgicos, etc.); visualizar a evolução do consumo de determinados insumos; visualizar horário de maior fluxo; visualizar o grupo de pessoas por sexo, faixa etária; entre outros.

Essa camada poderá conter também uma implementação de recursos de registro e notificação de eventos em nuvem, a exemplo do AWS IoT Events ([Amazon Web Services, 2017](#)), que é um serviço totalmente gerenciado que facilita a detecção e a resposta a eventos de sensores e aplicativos de IoT. Podendo ser utilizado para enviar alertas quando a temperatura dos centros cirúrgicos ou sala de condicionamento de vacinas atingir valores críticos.

Outro serviço que poderá ser utilizado é o ThingSpeak ([THINGSPEAK, 2021](#)) que é um serviço de plataforma de análise IoT que permite agregar, visualizar e analisar fluxos de dados ao vivo na nuvem. Ao enviar dados dos dispositivos para ThingSpeak, a plataforma poderá criar visualizações instantâneas dos dados ao vivo e enviar alertas usando serviços da web. Com a análise MATLAB dentro do ThingSpeak, poderá escrever e executar o código MATLAB para realizar pré-processamento, visualizações e análises. Thingspeak é uma das mais conhecidas tecnologias de análise de IoT, que é usada para fins de visualização dos dados coletados. Ele faz a tarefa básica de coletar, analisar e fornecer ações com base nos resultados derivados ([VIJAYALAKSHMI; ABISHEK; STEPHEN, 2019](#)).

Portanto, para atender à necessidade, a aplicação proposta será organizada da seguinte forma: na camada de sensores, estarão localizados os leitores e TAGs RFIDs responsáveis por identificar unicamente um objeto/pessoa; a camada de redes será responsável por trafegar todos os dados capturados dos sensores agregando os dados; a camada de dados irá processar os dados que serão necessários para armazenar e apresentar de forma objetiva aos gestores; e por último, a camada de aplicação, responsável pela exibição dos dados tratados.

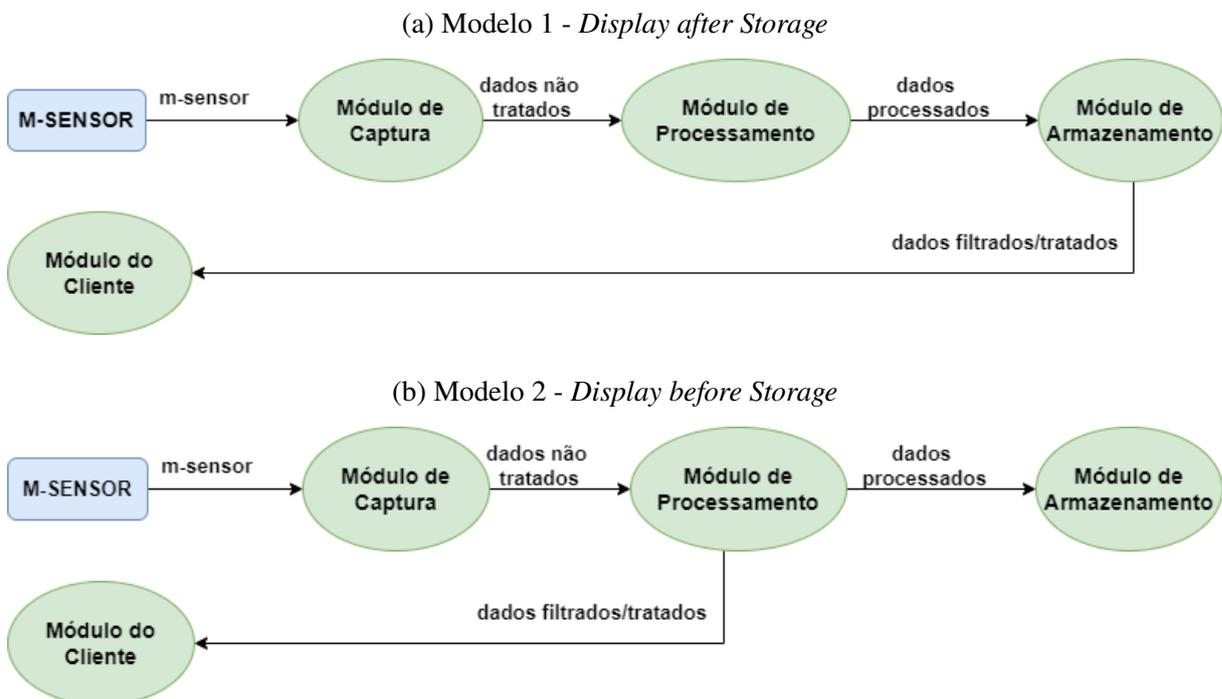
4.2 Estudo de Caso

Conforme ilustrado na [Figura 8](#), a aplicação de localização interna de equipamentos, proposta e utilizada em nosso estudo de caso, é composta por quatro módulos principais que realizam a captura, o processamento, o armazenamento e a exibição do resultado - Módulo de Captura, Módulo de Processamento, Módulo de Armazenamento e Módulo do Cliente, respectivamente. O diferencial do Modelo “Display after Storage” representado na [Figura 8a](#), é que o módulo do cliente é executado após o armazenamento de dados na camada *Cloud Computing*, diferente da [Figura 8b](#) (*Display before Storage*), em que os dados são exibidos para o cliente logo após o processamento dos dados na camada *Fog Computing*.

Os módulos do aplicativo são modelados no iFogSim usando a classe AppModule. Existem dependências de dados entre os módulos, essas dependências são modeladas usando a classe AppEdge no iFogSim2. Por fim, a malha de controle de interesse para aplicação proposta é modelada no iFogSim2 usando a classe AppLoop. O aplicativo recebe sinais de sensores por um sensor RFID ou BLE (*Bluetooth Low Energy*) e um dispositivo móvel com display exibe a localização do equipamento. As funções dos módulos acima mencionados são as seguintes:

1. Módulo de Captura: representado pelos *gateways*, posicionados estrategicamente em cada sala, captura os sinais brutos advindos dos sensores acoplados nos equipamentos móveis;
2. Módulo de Processamento: um servidor local, representado na topologia física por “*Fog*”, com maior poder de processamento faz o tratamento dos dados brutos. Após o processamento, envia os dados processados para a nuvem;
3. Módulo de Armazenamento: representado na topologia física pelo componente *Cloud*, é responsável por armazenar todos os dados já processados;
4. Módulo do Cliente: é o dispositivo móvel capaz de exibir as informações processadas para o usuário final;

Figura 8 – Modelos da aplicação proposta



Fonte: Autor.

Para realizar o estudo de caso mencionado, consideramos uma topologia física com 1 (um) dispositivo na camada *Fog* e 20 (vinte) *gateways* responsáveis pela captura dos dados. A [Figura 9](#) mostra os pontos da localização dos *gateways* no Hospital Universitário de Lagarto, que foi escolhido por ter fornecido informações importantes para a formulação da solução aqui proposta. Os pontos em azul, escolhidos arbitrariamente no Google Maps, simulam os *gateways* fixos nas salas/alas em que os equipamentos irão se locomover. Já o ponto em verde, simula o equipamento hospitalar que será movido aleatoriamente. Esses pontos arbitrários são configurados num arquivo CSV, conforme mostra a [Figura 10](#), contendo as informações do ID (identificador

da sala/ambiente), Latitude, Longitude, Block (Bloco/Prédio), Level (Andar), Parent (Setor “Pai” na hierarquia), State (Estado/Condição do ambiente) e Details (descrição/nome do equipamento).

Figura 9 – Mapa da localização física dos gateways



Fonte: Autor.

Figura 10 – Arquivo de configuração da topologia física

	A	B	C	D	E
1	ID,Latitude,Longitude,Block,Level,Parent,State,Details				
2	0,-37.8136,144.9631,0,0,-1,VIC,DataCenter				
3	1,-37.813946,144.954633,1,1,0,VIC,Proxy Server				
4	3,-10.92332,-37.66017,1,2,1,VIC,Gateway 01				
5	4,-10.92330,-37.66009,1,2,1,VIC,Gateway 02				
6	5,-10.923172,-37.66001,1,2,1,VIC,Gateway 03				
7	6,-10.92294,-37.65994,1,2,1,VIC,Gateway 04				

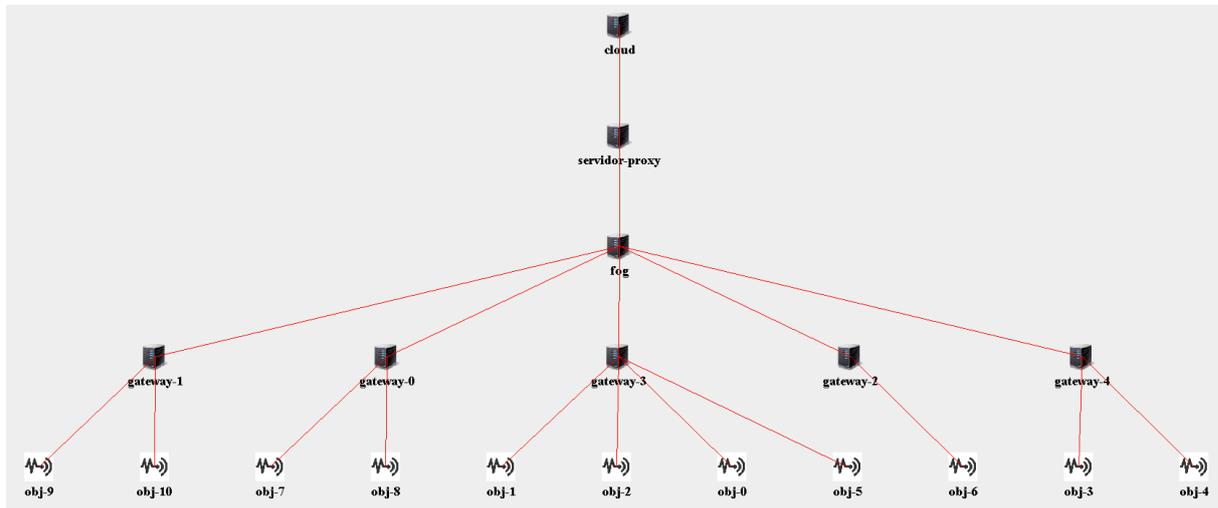
Fonte: Autor.

4.3 Descrição dos Cenários e Parâmetros da Simulação

A Tabela 6 ilustra as configurações dos diferentes tipos de dispositivos Fog utilizados na topologia proposta. O link de rede escolhido em cada camada foi baseado na infraestrutura atual do Hospital Universitário de Lagarto. Além disso, é possível notar que o poder de processamento, em MIPS, é crescente no sentido bottom-up das camadas da arquitetura. A topologia física do estudo de caso é modelada no iFogSim2 através das classes FogDevice, Sensor e Actuator. A Figura 11 mostra a topologia física da aplicação modelada no ambiente gráfico do iFogSim. A Tabela 7, descreve a latência dos links de rede utilizados no simulador. É importante destacar que a latência nas conexões dos níveis mais baixos é menor, por tratar-se de rede local e distâncias

menores, enquanto as conexões *Fog* e *Cloud* são colocadas com latências maiores por envolver em alguns casos conexões à internet ou que passam por diferentes níveis de encaminhamento de pacotes dentro da hierarquia da rede.

Figura 11 – Topologia física da solução proposta



Fonte: Autor.

Tabela 6 – Configuração dos Dispositivos *Fog* para aplicação da solução proposta

Atributo	Cloud	Servidor Proxy	Fog	Gateway	Obj. Móvel
MIPS	44800	2800	4000	2800	500
RAM (MB)	40960	4096	10240	512	20
Uplink (Mbps)	100	10000	10000	100	1000
Downlink (Mbps)	10000	10000	10000	100	270
Hierarquia	0	1	2	3	4

Fonte: Autor.

O valor dos parâmetros de simulação dentro de um intervalo específico é determinado por um gerador de números pseudo-aleatórios que encerra a simulação ao atingir o valor 60000 (sessenta mil) milissegundos, esse valor é definido na classe *Config.java* alterando o valor da variável *MAX_SIMULATION_TIME*.

Além disso, o ambiente de computação é definido para ser hierárquico com objetos móveis no nível mais alto. O nó *Cloud* é marcado como nível 0, em seguida o nó proxy-server, no nível 1, interliga o servidor *Cloud* com o servidor *Fog* localizado no nível 2. O nível 3, é onde se encontram os *gateways* e em seguida o nível 4 localizam-se os objetos móveis. Um notebook configurado com processador Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz com 8 GB de RAM foi utilizado para executar o script de simulação e realizar os experimentos.

Com o objetivo de testar o desempenho em diferentes tamanhos de topologia, variamos o número de objetos móveis, mantendo constante o tempo de transmissão dos sensores. Com

Tabela 7 – Descrição dos links de rede para o aplicativo de rastreamento

Origem	Destino	Latência (em ms)
sensor	obj-móvel	6
obj-móvel	gateway	2
gateway	fog	4
fog	proxy	100
proxy	cloud	100

Fonte: Autor.

os sensores transmitindo a cada 1 (um) segundo, variamos a quantidade de objetos de 1 para 2, em seguida de 2 para 4, depois 8, 16, 32, 64 e 128 objetos. Para a mesma variação de objetos anterior, variamos o tempo de transmissão para 2 (dois) segundos, 5 (cinco) segundos e 10 (dez) segundos. Além disso, executamos o simulador em 3 (três) cenários diferentes, são eles:

1. Cenário 1 (*Edge*): Os dados dos sensores (dos objetos móveis) são enviados para os *gateways*, onde são **capturados e processados**, em seguida enviados para a nuvem armazenar, sem passar por um servidor local;
2. Cenário 2 (*Cloud*): Os dados dos sensores (dos objetos móveis) são enviados para os *gateways*, onde são **capturados e enviados para a nuvem**. Somente ao chegar na nuvem os dados são **processados e armazenados**, sem passar por um servidor local;
3. Cenário 3 (*Fog*): Os dados dos sensores (dos objetos móveis) são enviados para os *gateways*, onde são **capturados e enviados para um servidor local (*Fog*)** para serem processados e em seguida enviados para a nuvem armazenar.

Para alteração dos cenários mencionados foi desenvolvido o [Código 1](#).

Enquanto a [Figura 12](#) detalha os processos realizados por cada cenário mencionado para o Modelo 1 ([Figura 8a](#) - *Display after storage*), a [Figura 13](#), detalha os processos realizados por cada cenário para o Modelo 2 ([Figura 8b](#) - *Display before storage*).

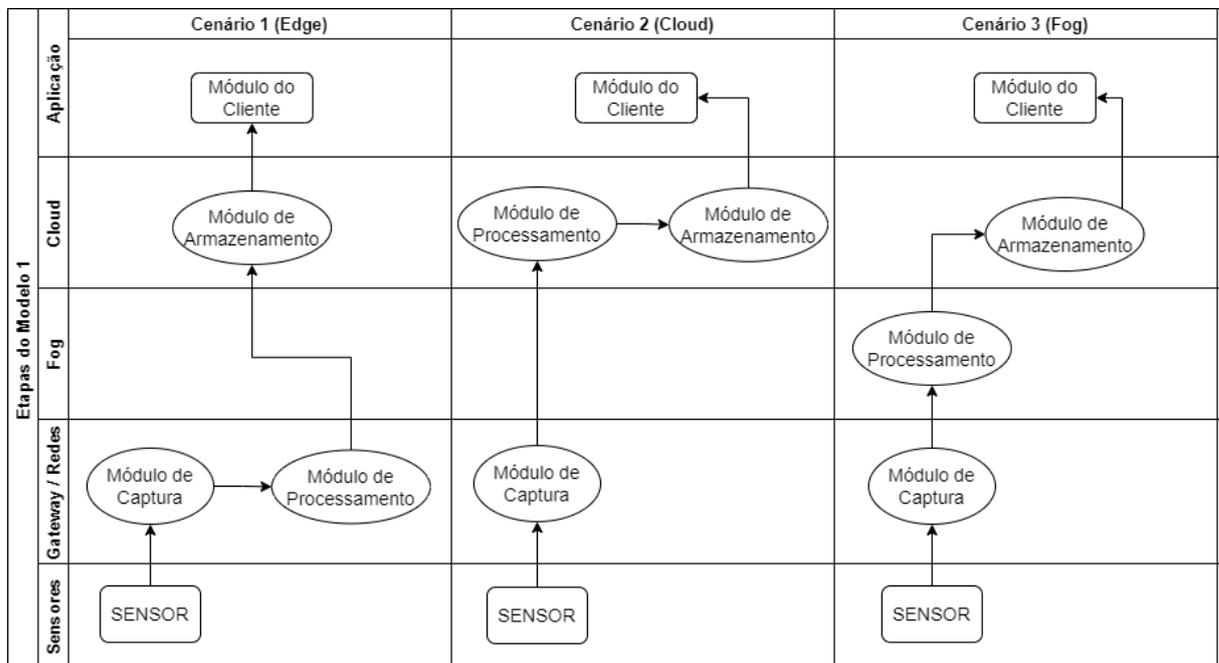
Para cada combinação de parâmetros e cenários mencionados aqui, foram coletados os dados das métricas: “*application loop delay*”, que consiste na latência total observada para a aplicação; “*network usage*”, que mede a quantidade de dados trafegados na rede; e “*execution time*”, que indica o tempo total para execução da simulação. Os resultados para essas métricas são apresentados na [seção 4.4](#).

Código 1 – Código para alteração de cenário de simulação

```

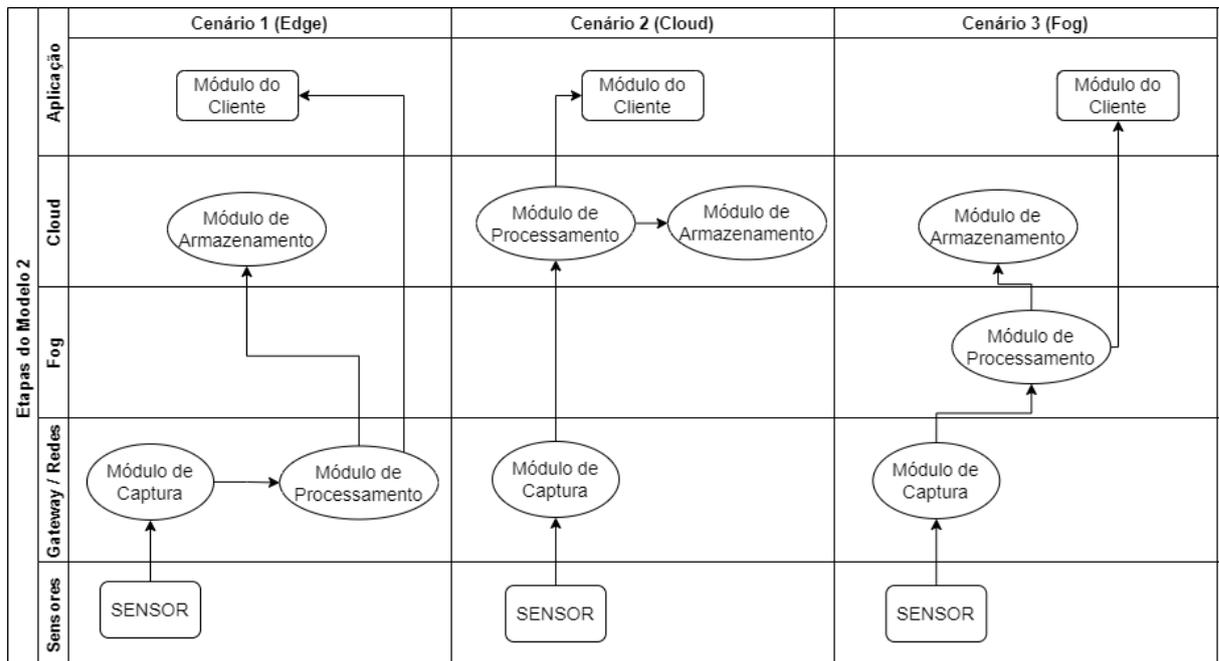
1 //se TRUE: transfere o processingModule para camada da nuvem
2 static boolean CLOUD = false;
3
4 //se TRUE: recupera o config com 5 niveis (objeto (4) -> gateway (3)
5 //se FALSE: recupera o config com 4 niveis (objeto (3) -> gateway
6 static boolean IS_WITH_LOCAL_SERVER = false;
7
8 //se TRUE: sensor -> captureModule -> processingModule ->
9 //se FALSE: sensor -> captureModule -> processingModule ->
10 static boolean RETURN_FILTERED_DATA_FROM_CLOUD = true;
11
12 // tempo de transmissao em Milissegundos
13 static double SENSOR_TRANSMISSION_TIME = 1000;
14
15 // numero de objetos moveis transmitindo suas informacoes
16 static int numberOfMobileObj = 4;
    
```

Figura 12 – Etapas dos processos realizados em cada cenário do Modelo 1 - *Display after Storage*



Fonte: Autor.

Figura 13 – Etapas dos processos realizados em cada cenário do Modelo 2 - *Display before Storage*



Fonte: Autor.

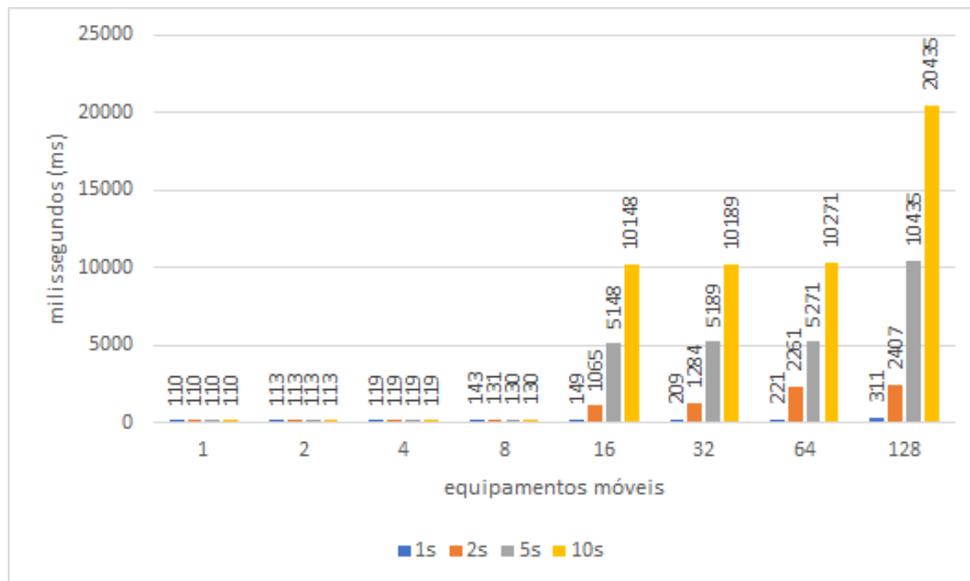
4.4 Resultados

4.4.1 Latência Média do Loop de Aplicativo

O *loop* de aplicativo - em termos de latência de resposta - é o loop responsável por transformar o estado dos sensores em informações úteis na tela do dispositivo do cliente (*smartphone, tablet, computador pessoal*). Isso requer comunicação em tempo real entre os dispositivos móveis do cliente e o dispositivo que hospeda o módulo de processamento (no caso do Modelo 2) e o dispositivo que hospeda o módulo de armazenamento (no caso do Modelo 1), juntamente com um processamento eficiente. Um grande atraso nesse *loop* prejudicará a experiência do usuário, pois afeta as entidades com as quais o usuário interage diretamente. Ou seja, a latência no Modelo 1, é a soma do tempo que leva para percorrer *captureModule*, *processingModule*, *storageModule* e em seguida *clientModule*. Já no Modelo 2, é o tempo para percorrer *captureModule*, *processingModule* e em seguida *clientModule*. A Figura 14 mostra a média do atraso de *loop* para o aplicativo do Modelo 1 no cenário 1 (*Edge*). Nesse modelo, pode-se observar que durou 20435 ms com a transmissão de 128 equipamentos transmitindo de 10 em 10 segundos. E 10435 ms ao transmitir de 5 em 5 segundos. Para necessidade de acompanhamento em tempo real, transmitindo a cada 1 (um) segundo, tivemos em média 311 (trezentos e onze) milissegundos o resultado da localização do equipamento ao utilizar 128 (cento e vinte e oito) equipamentos móveis.

A Figura 15 mostra a média do atraso de *loop* para o aplicativo do Modelo 1 no cenário

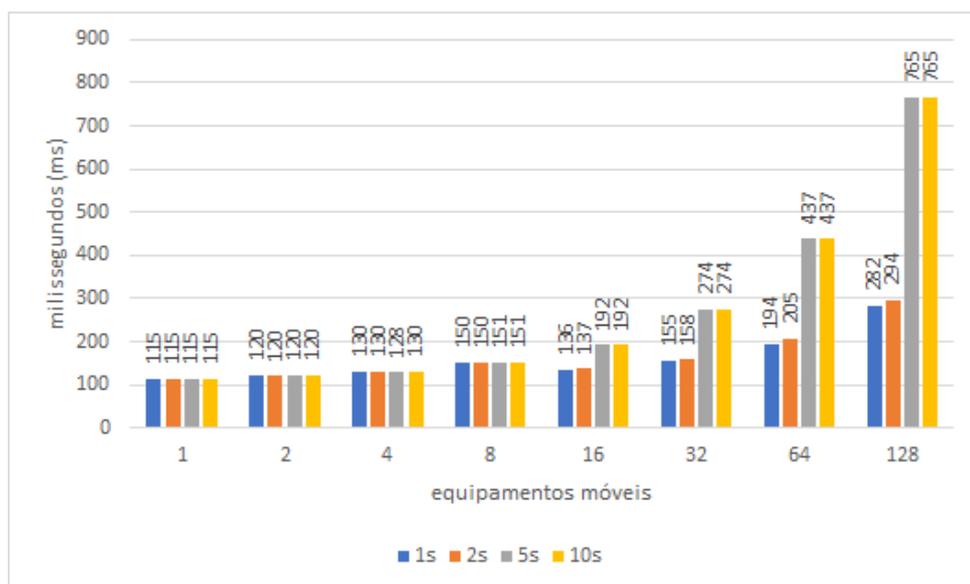
Figura 14 – Média do atraso de *loop* do aplicativo para o cenário 1 (*Edge*) no Modelo 1 (*display after storage*)



Fonte: Autor.

2 (*Cloud*). Nesse modelo, pode-se observar que durou 765 ms com a transmissão de 128 equipamentos transmitindo de 10 em 10 segundos e o mesmo resultado transmitindo de 5 em 5 segundos.

Figura 15 – Média do atraso de *loop* do aplicativo para o cenário 2 (*Cloud*) no Modelo 1 (*display after storage*)

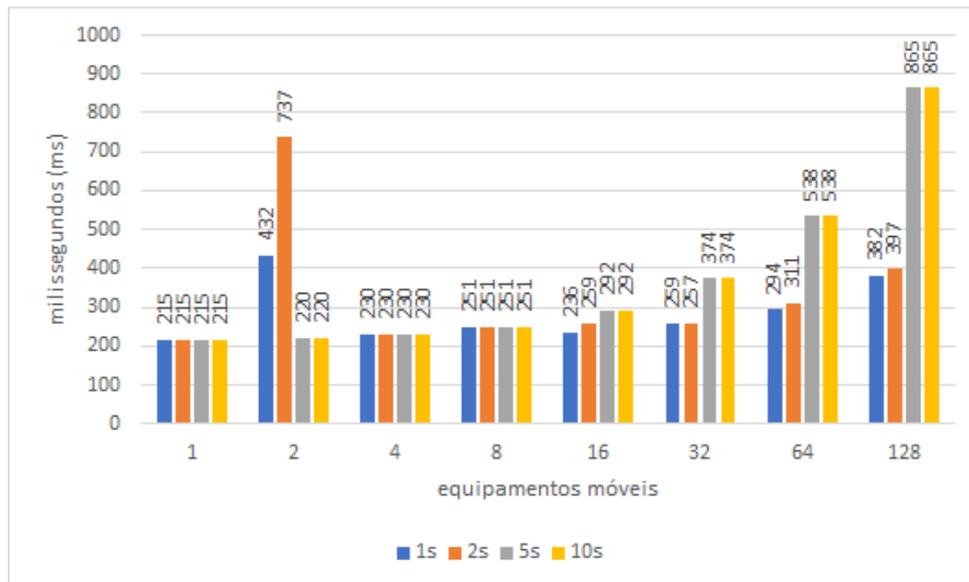


Fonte: Autor.

A Figura 16 mostra a média do atraso de *loop* para o aplicativo do Modelo 1 no cenário 3 (*Fog*). Nesse modelo, pode-se observar que durou 865 ms com a transmissão de 128 equipamentos transmitindo de 10 em 10 segundos e o mesmo resultado transmitindo de 5 em 5 segundos. Porém

ao utilizar 2 equipamentos transmitindo a cada 10 segundos, resultou em 220 ms e transmitindo a cada 2 segundos, resultou em 737 ms. Esse resultado pode ser considerado um *outlier* resultante de flutuações inesperadas do processo de simulação.

Figura 16 – Média do atraso de *loop* do aplicativo para o cenário 3 (*Fog*) no Modelo 1 (*display after storage*)

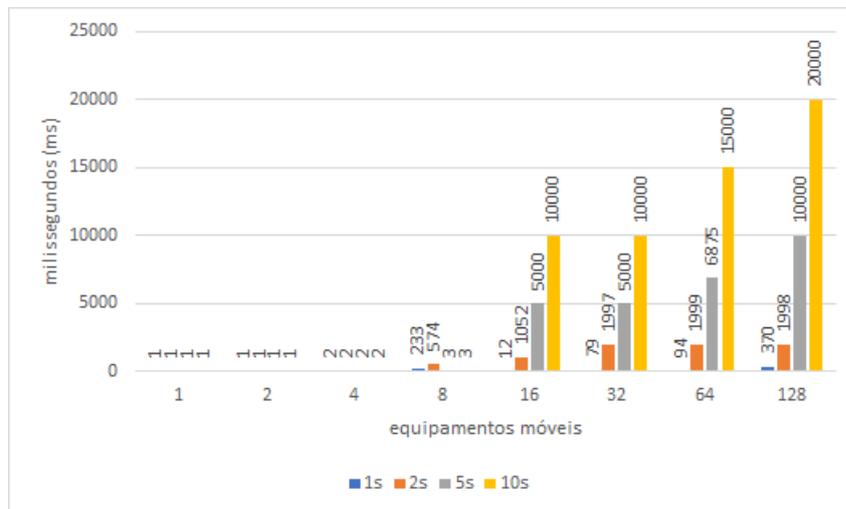


Fonte: Autor.

Já no Modelo 2 (*display before storage*), a Figura 17, Figura 18 e Figura 19 mostram a média do atraso de loop para os três cenários. Na Figura 17 (cenário 1 - *Edge*), observa-se que durou 20000 ms com a transmissão de 128 equipamentos transmitindo a cada 10 segundos e 10000 ms ao transmitir a cada 5 segundos no cenário 1. Na Figura 18 (cenário 2 - *Cloud*), observa-se que durou 765 ms com a transmissão de 128 equipamentos transmitindo a cada 5 e 10 segundos. Já na Figura 19 (cenário 3 - *Fog*), observa-se que durou 665 ms com a transmissão de 128 equipamentos transmitindo a cada 5 e 10 segundos.

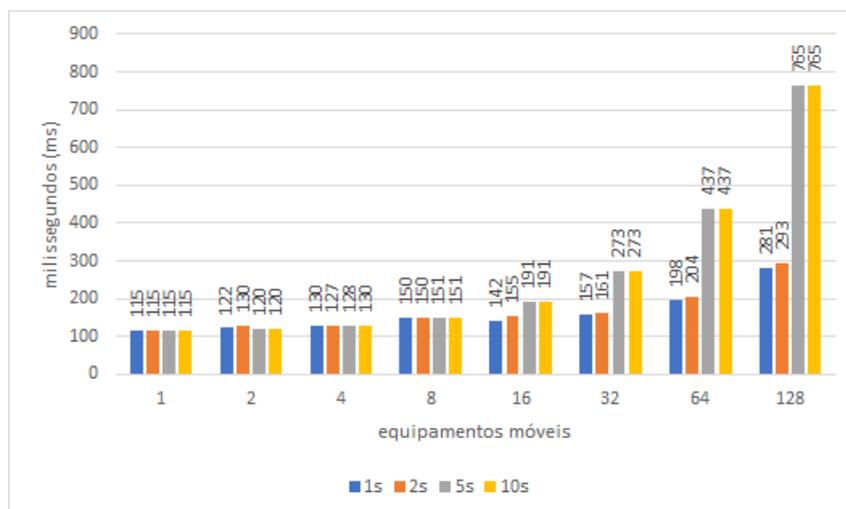
A Figura 20 mostra o comparativo dos cenários 1, 2 e 3 comparando o Modelo 1 com o Modelo 2 utilizando 64 (sessenta e quatro) e 128 (cento e vinte e oito) equipamentos móveis, transmitindo a cada 10 (dez) segundos. Observa-se que o cenário 3 (*Fog Computing*), Modelo 2 (*display before storage*), com 128 equipamentos, reduziu aproximadamente 23,12% em relação ao mesmo cenário no Modelo 1 (*display after storage*). Já para sessenta e quatro equipamentos, a redução foi de aproximadamente 37,36%. No Modelo 1, o cenário 3 aumentou aproximadamente 13,07% em relação ao cenário 2 (*Cloud Computing*) com 128 equipamentos e aumentou aproximadamente 23,11% com 64 equipamentos transmitindo a cada 10 segundos. Já no Modelo 2, o cenário 3 reduziu o tempo do *application loop delay* em aproximadamente 13,07% para 128 equipamentos e reduziu aproximadamente 22,88% com 64 equipamentos, quando comparados ao cenário 2 (*Cloud Computing*). O cenário 1 (*Edge Computing*) nos dois modelos apresentou o pior desempenho em relação aos outros dois cenários (*Cloud* e *Fog Computing*).

Figura 17 – Média do atraso de *loop* do aplicativo para o cenário 1 (*Edge*) no Modelo 2 (*display before storage*)



Fonte: Autor.

Figura 18 – Média do atraso de *loop* do aplicativo para o cenário 2 (*Cloud*) no Modelo 2 (*display before storage*)

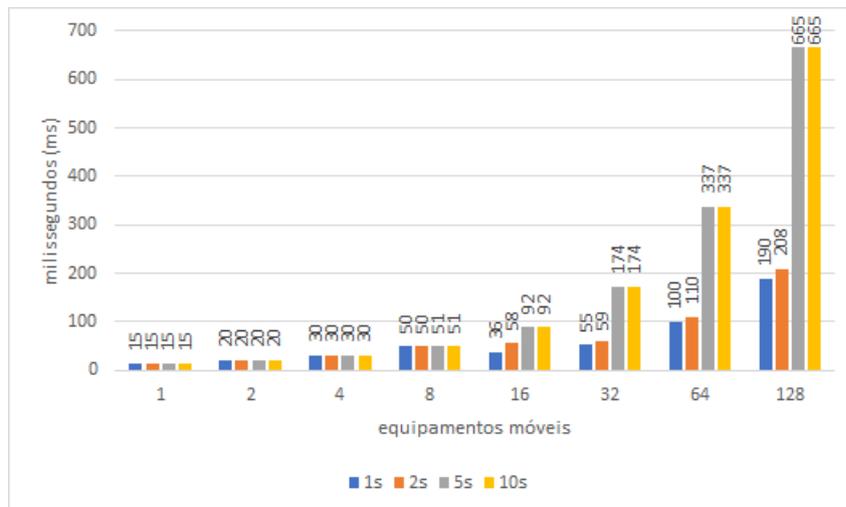


Fonte: Autor.

É importante ressaltar, que para todos os intervalos de envio de dados, a combinação do cenário 3 (*Fog Computing*) com o Modelo 2 (*display before storage*), foi a que obteve os melhores resultados do atraso de *loop* da aplicação. Esse resultado torna essa configuração a mais promissora para ser aplicada em uma implementação do sistema proposto em ambientes reais.

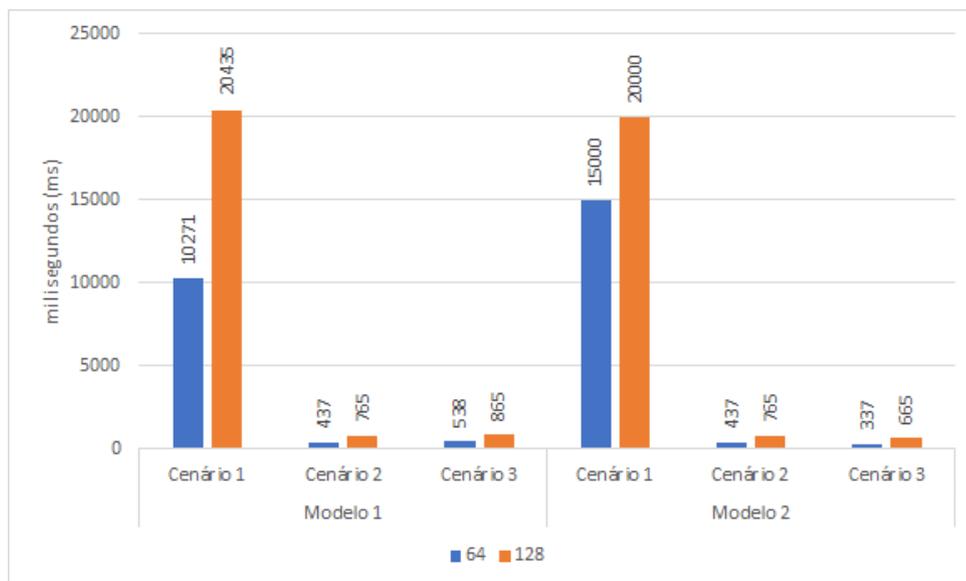
A Figura 21 mostra o comparativo do latência média da aplicação no cenário de *Fog Computing*, comparando o Modelo 1 (*display after storage*) com o Modelo 2 (*display before storage*), com os sensores retransmitindo a cada 10 segundos. É possível notar que o Modelo 2 obteve melhor desempenho, reduzindo 37% na latência da aplicação para 64 equipamentos e reduzindo 23% ao utilizar 128 equipamentos.

Figura 19 – Média do atraso de *loop* do aplicativo para o cenário 3 (Fog) no Modelo 2 (*display before storage*)



Fonte: Autor.

Figura 20 – Comparativo do *Application Loop Delay* no Modelo 1 (*display after storage*) e Modelo 2 (*display before storage*)

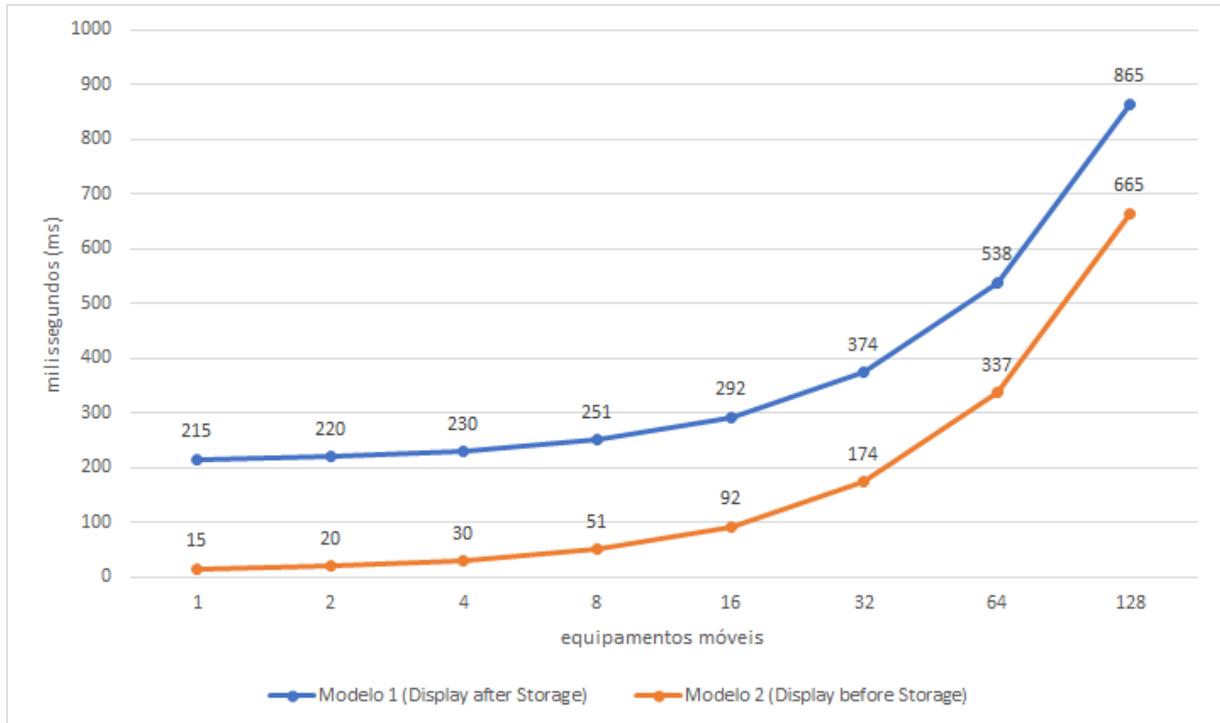


Fonte: Autor.

Já a [Figura 22](#), mostra o comparativo dos cenários *Cloud Computing* e *Fog Computing* no Modelo 2 (*display before storage*). É possível notar que a *Fog Computing* obteve melhor desempenho na latência da aplicação, com redução de 23% com 64 equipamentos e redução de 13% com 128 equipamentos retransmitindo a cada 10 segundos.

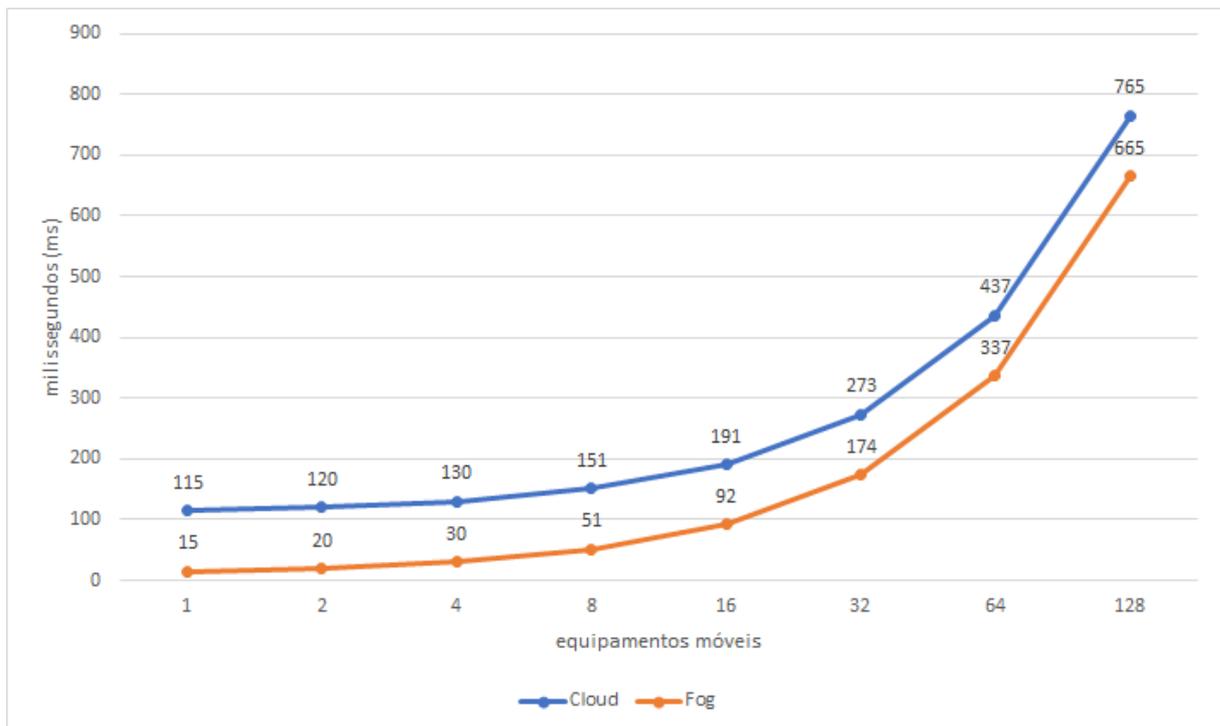
A [Tabela 8](#) mostra os resultados da latência média da aplicação, em milissegundos, nos cenários de *Edge*, *Cloud* e *Fog Computing* nos Modelos 1 (*Display after Storage*) e Modelo 2 (*Display before Storage*).

Figura 21 – Comparativo do *Application Loop Delay*, Modelo 1 (*display after storage*) e Modelo 2 (*display before storage*), no cenário de *Fog* com retransmissão de 10 segundos



Fonte: Autor.

Figura 22 – Comparativo do *Application Loop Delay* nos cenários de *Cloud* e *Fog* do Modelo 2 (*display before storage*)



Fonte: Autor.

Tabela 8 – Tabela geral da latência média da aplicação nos cenários de *Edge*, *Cloud* e *Fog* nos Modelos 1 (*display after storage*) e Modelo 2 (*display before storage*)

Cenário	Modelo	Ret.	Equipamentos móveis (sensores)							
			1	2	4	8	16	32	64	128
Edge	Mod. 1	1s	110	113	119	143	149	209	221	311
	Mod. 2		1	1	2	233	12	79	94	370
	Mod. 1	2s	110	113	119	131	1065	1284	2261	2407
	Mod. 2		1	1	2	574	1052	1997	1999	1998
	Mod. 1	5s	110	113	119	130	5148	5189	5271	10435
	Mod. 2		1	1	2	3	5000	5000	6875	10000
	Mod. 1	10s	110	113	119	130	10148	10189	10271	20435
	Mod. 2		1	1	2	3	10000	10000	15000	20000
Cloud	Mod. 1	1s	115	120	130	150	136	155	194	282
	Mod. 2		115	122	130	150	142	157	198	281
	Mod. 1	2s	115	120	130	150	137	158	205	294
	Mod. 2		115	130	127	150	155	161	204	293
	Mod. 1	5s	115	120	128	151	192	274	437	765
	Mod. 2		115	120	128	151	191	273	437	765
	Mod. 1	10s	115	120	130	151	192	274	437	765
	Mod. 2		115	120	130	151	191	273	437	765
Fog	Mod. 1	1s	215	432	230	251	236	259	294	382
	Mod. 2		15	20	30	50	36	55	100	190
	Mod. 1	2s	215	737	230	251	259	257	311	397
	Mod. 2		15	20	30	50	58	59	110	208
	Mod. 1	5s	215	220	230	251	292	374	538	865
	Mod. 2		15	20	30	51	92	174	337	665
	Mod. 1	10s	215	220	230	251	292	374	538	865
	Mod. 2		15	20	30	51	92	174	337	665

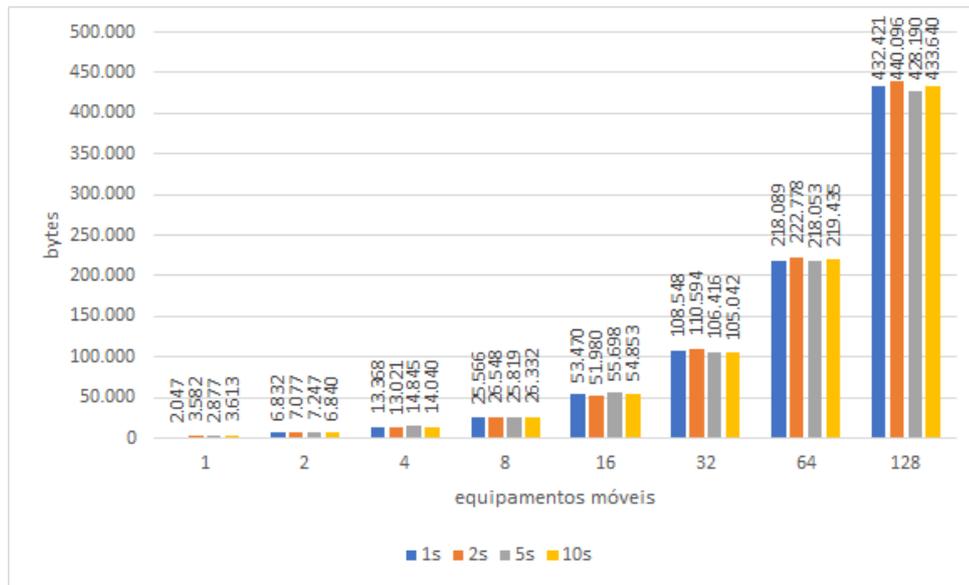
Fonte: Autor.

4.4.2 Uso da Rede

A [Figura 23](#), [Figura 24](#) e [Figura 25](#) mostram a média do uso da rede para os três cenários do Modelo 1. A [Figura 23](#) mostra o uso da rede do aplicativo para o cenário 1. O aumento do número de dispositivos conectados aumenta significativamente a carga na rede conforme esperado. Transmitindo a cada 10 (dez) segundos, utilizando 64 equipamentos o resultado foi de 219.435 bytes ao passo que ao utilizar 128 equipamentos, retornou 433.640 bytes, um aumento de aproximadamente 97,61%.

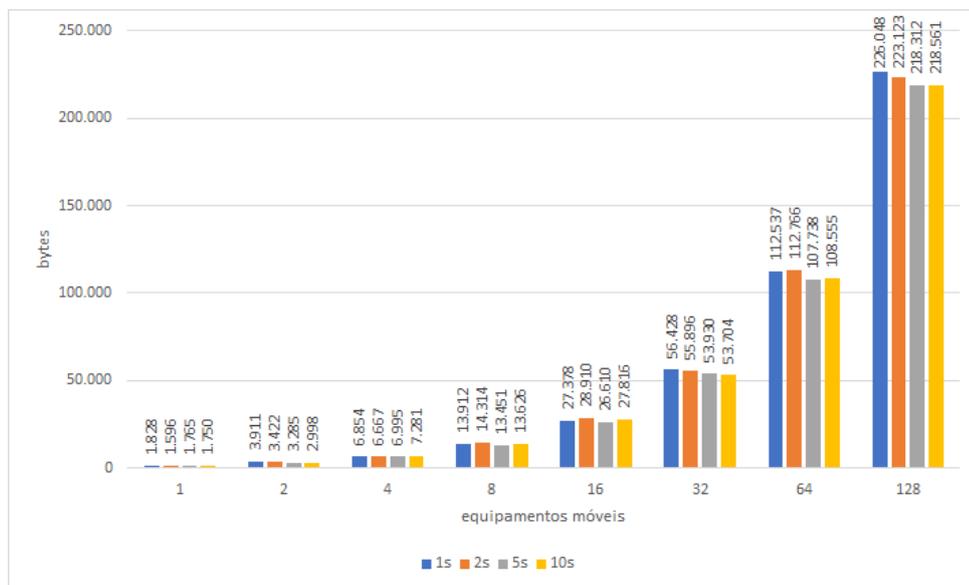
A [Figura 24](#) mostra o uso da rede do aplicativo proposto para o Modelo 1 no cenário 2. Transmitindo a cada 5 (cinco) segundos e 128 (cento e vinte e oito) equipamentos, o simulador retornou 218.312 bytes, enquanto ao transmitir a cada 10 segundos retornou 218.561, um aumento negligível em termos práticos, de aproximadamente 0,11%.

Figura 23 – Média do *Network Usage* para o cenário 1 (*Edge*) no Modelo 1 (*display after storage*)



Fonte: Autor.

Figura 24 – Média do *Network Usage* para o cenário 2 (*Cloud*) no Modelo 1 (*display after storage*)

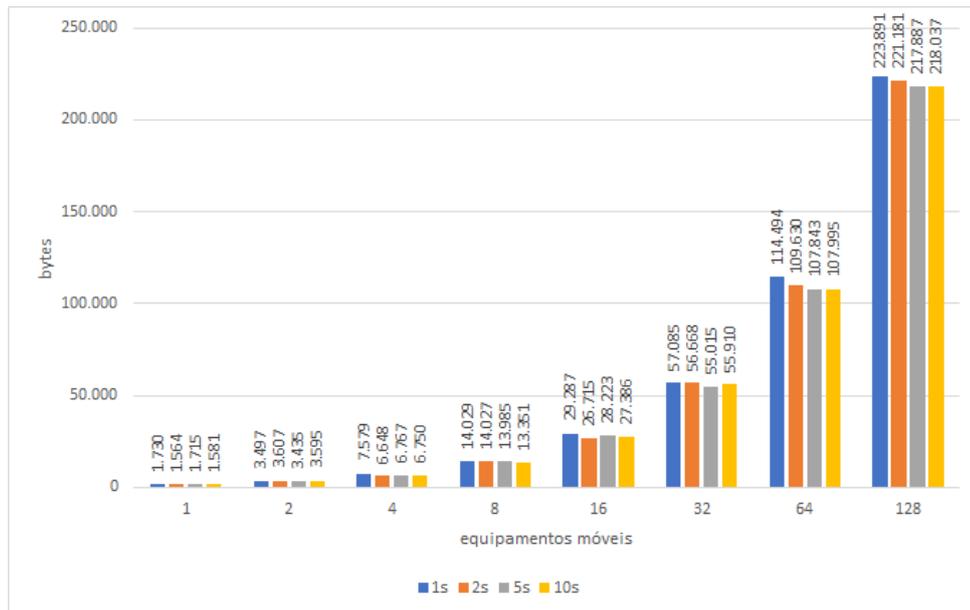


Fonte: Autor.

O simulador, no cenário 3, retornou 217.887 bytes e 218.037 bytes ao utilizar 128 (cento e vinte e oito) equipamentos transmitindo a cada 5 (cinco) e 10 (dez) segundos respectivamente, conforme ilustra a [Figura 25](#).

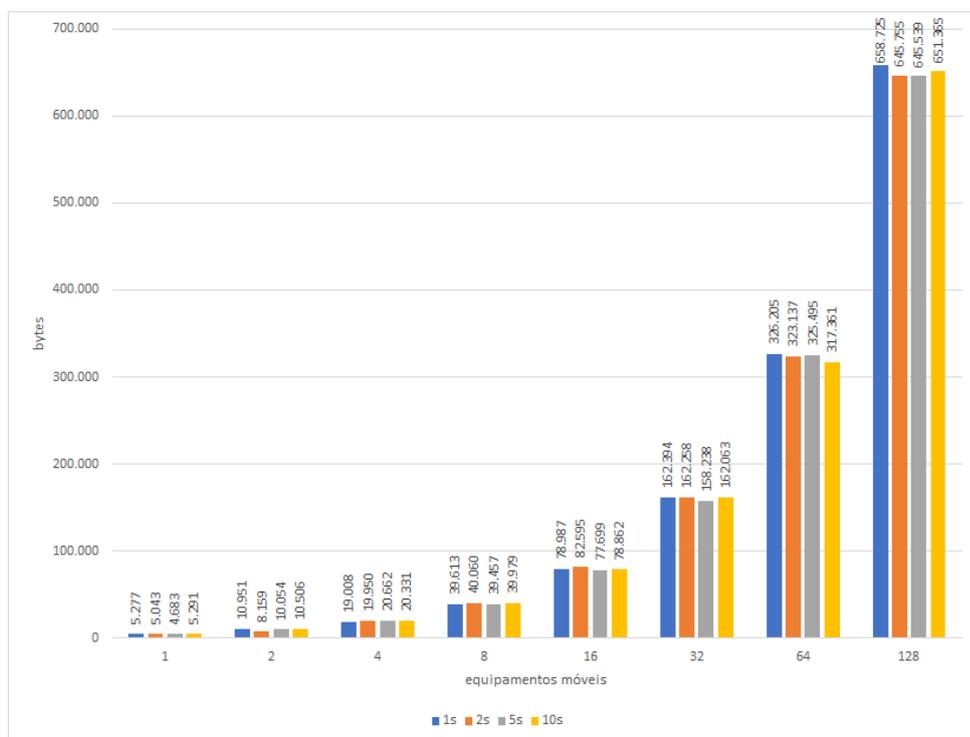
A [Figura 26](#), [Figura 27](#) e [Figura 28](#) mostram a média do uso da rede para os três cenários do Modelo 2. A [Figura 26](#) mostra o uso da rede do aplicativo para no cenário 1. Transmitindo a cada 5 (cinco) segundos e 128 (cento e vinte e oito) equipamentos, o simulador retornou 645.539 bytes e 651.365 bytes ao transmitir a cada 10 (dez) segundos.

Figura 25 – Média do *Network Usage* para o cenário 3 (Fog) no Modelo 1 (*display after storage*)



Fonte: Autor.

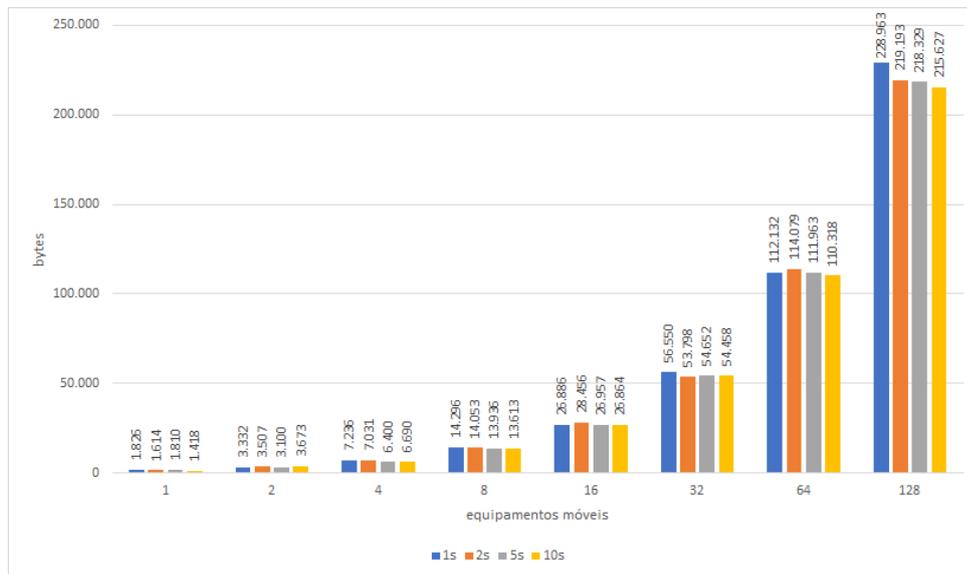
Figura 26 – Média do *Network Usage* para o cenário 1 (Edge) no Modelo 2 (*display before storage*)



Fonte: Autor.

No cenário 2, transmitindo a cada 5 (cinco) segundos e 128 (cento e vinte e oito) equipamentos, o simulador retornou 218.329 bytes e 215.627 bytes ao transmitir a cada 10 (dez) segundos, uma redução de aproximadamente 1,23%, conforme ilustra a [Figura 27](#).

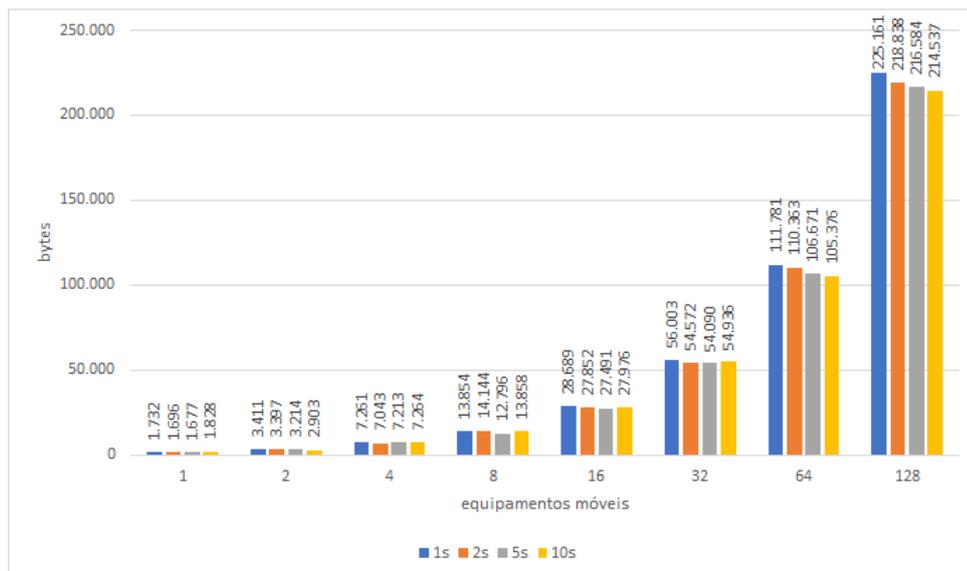
Figura 27 – Média do *Network Usage* para no cenário 2 (*Cloud*) no Modelo 2 (*display before storage*)



Fonte: Autor.

A Figura 28 mostra o uso da rede do aplicativo proposto para o cenário 3. Transmitindo a cada 5 (cinco) segundos e 128 (cento e vinte e oito) equipamentos, o simulador retornou 216.584 bytes e 214.537 bytes ao transmitir a cada 10 (dez) segundos, uma redução aproximada de 0,94%.

Figura 28 – Média do *Network Usage* para o cenário 3 (*Fog*) no Modelo 2 (*display before storage*)

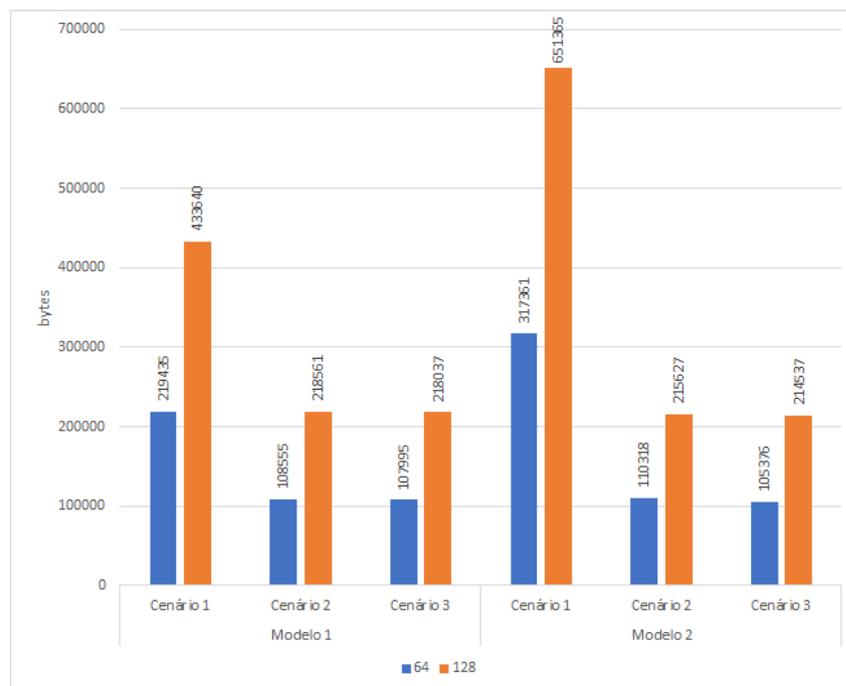


Fonte: Autor.

A Figura 29 mostra o comparativo dos cenários 1, 2 e 3 comparando o Modelo 1 com o Modelo 2 utilizando 64 (sessenta e quatro) e 128 (cento e vinte e oito) equipamentos móveis, transmitindo a cada 10 (dez) segundos. Pode-se observar que no Modelo 1, o cenário 3 (*Fog Computing*) reduziu aproximadamente 0,51% em relação ao cenário 2 (*Cloud Computing*) com

64 equipamentos e uma redução de aproximadamente 0,23% com 128 equipamentos. Já no Modelo 2, reduziu aproximadamente 4,47% com a utilização de 64 equipamentos, e uma redução aproximada de 0,5% com 128 equipamentos. Ao compararmos o cenário 2 do Modelo 1 com o cenário 2 do Modelo 2, obtivemos um aumento aproximado de 1,62% com 64 equipamentos e uma redução de aproximadamente 1,34% com 128 equipamentos. No cenário 3, ao utilizarmos o Modelo 2 com 64 equipamentos, obtivemos uma redução aproximada de 2,42% e uma redução aproximada de 1,61% com 128 equipamentos. O cenário 1 (*Edge Computing*) apresentou pouco mais que o dobro do *network usage* ao utilizar 64 equipamentos e praticamente o dobro ao utilizar 128 equipamentos no Modelo 1. Já no Modelo 2, o cenário 1 apresentou mais que o dobro de *network usage* em relação aos outros cenários.

Figura 29 – Comparativo do *Network Usage* no Modelo 1 (*display after storage*) e Modelo 2 (*display before storage*)



Fonte: Autor.

A Tabela 9 mostra os resultados do *network usage*, em bytes, nos cenários de *Edge*, *Cloud* e *Fog Computing* nos Modelos 1 (*Display after Storage*) e Modelo 2 (*Display before Storage*).

4.4.3 Execution Time

O *execution time* é o tempo que o simulador demora executando o *script*. Conforme esperado, ao aumentarmos o número de equipamentos, aumenta também o tempo de execução do simulador. Ao compararmos o tempo de execução do Modelo 1 com o Modelo 2, no cenário 2 (*Cloud Computing*) houve uma redução de aproximadamente 14,3% e no cenário 3 (*Fog Computing*), uma redução de aproximadamente 3,16%.

Tabela 9 – Tabela geral do *network usage* nos cenários de *Edge*, *Cloud* e *Fog* nos Modelos 1 (*display after storage*) e Modelo 2 (*display before storage*)

Cen.	Mod.	Ret.	Equipamentos móveis (sensores)							
			1	2	4	8	16	32	64	128
Edge	Mod. 1	1s	2.047	6.832	13.368	25.566	53.470	108.548	218.089	432.421
	Mod. 2		5.277	10.951	19.008	39.613	78.987	162.394	326.205	658.725
	Mod. 1	2s	3.582	7.077	13.021	26.548	51.980	110.594	222.778	440.096
	Mod. 2		5.043	8.159	19.950	40.060	82.595	162.258	323.137	645.755
	Mod. 1	5s	2.877	7.247	14.845	25.819	55.698	106.416	218.053	428.190
	Mod. 2		4.683	10.054	20.662	39.457	77.699	158.238	325.495	645.539
	Mod. 1	10s	3.613	6.840	14.040	26.332	54.853	105.042	219.435	433.640
	Mod. 2		5.291	10.506	20.331	39.979	78.862	162.063	317.361	651.365
Cloud	Mod. 1	1s	1.828	3.911	6.854	13.912	27.378	56.428	112.537	226.048
	Mod. 2		1.826	3.332	7.236	14.296	26.886	56.550	112.132	228.963
	Mod. 1	2s	1.596	3.422	6.667	14.314	28.910	55.896	112.766	223.123
	Mod. 2		1.614	3.507	7.031	14.053	28.456	53.798	114.079	219.193
	Mod. 1	5s	1.765	3.285	6.995	13.451	26.610	53.930	107.738	218.312
	Mod. 2		1.810	3.100	6.400	13.936	26.957	54.652	111.963	218.329
	Mod. 1	10s	1.750	2.998	7.281	13.626	27.816	53.704	108.555	218.561
	Mod. 2		1.418	3.673	6.690	13.613	26.864	54.458	110.318	215.627
Fog	Mod. 1	1s	1.730	3.497	7.579	14.029	29.287	57.085	114.494	223.891
	Mod. 2		1.732	3.411	7.261	13.854	28.689	56.003	111.781	225.161
	Mod. 1	2s	1.564	3.607	6.648	14.027	26.715	56.668	109.630	221.181
	Mod. 2		1.696	3.397	7.043	14.144	27.852	54.572	110.363	218.838
	Mod. 1	5s	1.715	3.435	6.767	13.985	28.223	55.015	107.843	217.887
	Mod. 2		1.677	3.214	7.213	12.796	27.491	54.090	106.671	216.584
	Mod. 1	10s	1.581	3.595	6.750	13.351	27.386	55.910	107.995	218.037
	Mod. 2		1.828	2.903	7.264	13.858	27.976	54.936	105.376	214.537

Fonte: Autor.

A [Tabela 10](#) mostra os resultados do execution time, em milissegundos, nos cenários de *Edge*, *Cloud* e *Fog Computing* nos Modelos 1 (Display after Storage) e 2 (Display before Storage).

4.4.4 MIPS 4000 vs MIPS 22400

Para avaliarmos o impacto do poder de processamento em MIPS (Milhões de Instruções Por Segundo) na *Fog Computing*, foi alterado o parâmetro MIPS conforme [Tabela 11](#).

A [Figura 30](#) mostra a média do comparativo do *application loop delay* com MIPS 4000 e MIPS 22400 utilizando-se 128 (cento e vinte e oito) equipamentos móveis. No Modelo 1, a [Figura 30a](#) mostra que a maior redução foi de aproximadamente 3,16% ao transmitir a cada 1s (um segundo). Já no Modelo 2, a [Figura 30b](#) mostra que a maior redução foi de aproximadamente 5,63% obtida ao transmitir a cada 2s.

A [Figura 31](#) mostra a média do *network usage* comparando o MIPS 4000 e MIPS 22400

Tabela 10 – Tabela geral do *execution time* nos cenários de *Edge*, *Cloud* e *Fog* nos Modelos 1 (*display after storage*) e Modelo 2 (*display before storage*)

Cenário	Modelo	Retransmissão	Equipamentos móveis (sensores)							
			1	2	4	8	16	32	64	128
Edge	Modelo 1	1s	252	342	478	659	989	1254	2556	4082
	Modelo 2		324	378	726	857	1168	1654	2636	4131
	Modelo 1	2s	212	280	349	510	841	1713	3352	6281
	Modelo 2		224	276	397	674	1294	2098	4269	346018
	Modelo 1	5s	243	267	355	463	913	1824	4079	73695
	Modelo 2		203	260	410	549	1000	2099	6992	126223
	Modelo 1	10s	161	216	291	402	807	1643	3346	16665
	Modelo 2		328	278	318	596	852	2243	4598	17880
Cloud	Modelo 1	1s	259	323	428	706	1043	1899	3342	5746
	Modelo 2		289	314	467	766	1366	1842	2886	4965
	Modelo 1	2s	223	358	449	577	874	1406	2673	4412
	Modelo 2		215	312	393	552	838	1605	2259	3781
	Modelo 1	5s	204	268	376	573	967	1623	3092	7862
	Modelo 2		174	216	302	444	701	1473	2652	6700
	Modelo 1	10s	159	201	255	384	643	1361	2431	5628
	Modelo 2		150	201	283	396	613	1243	2863	6652
Fog	Modelo 1	1s	251	348	446	618	1073	1757	3120	5310
	Modelo 2		280	328	442	716	1377	1702	2663	4749
	Modelo 1	2s	201	262	333	516	776	1245	2659	4196
	Modelo 2		231	306	381	544	824	1450	2166	4063
	Modelo 1	5s	192	269	313	447	757	1418	2728	7167
	Modelo 2		173	221	308	443	747	1761	2646	7553
	Modelo 1	10s	165	207	289	388	642	1241	2458	5603
	Modelo 2		154	191	248	367	630	1396	2932	6083

Fonte: Autor.

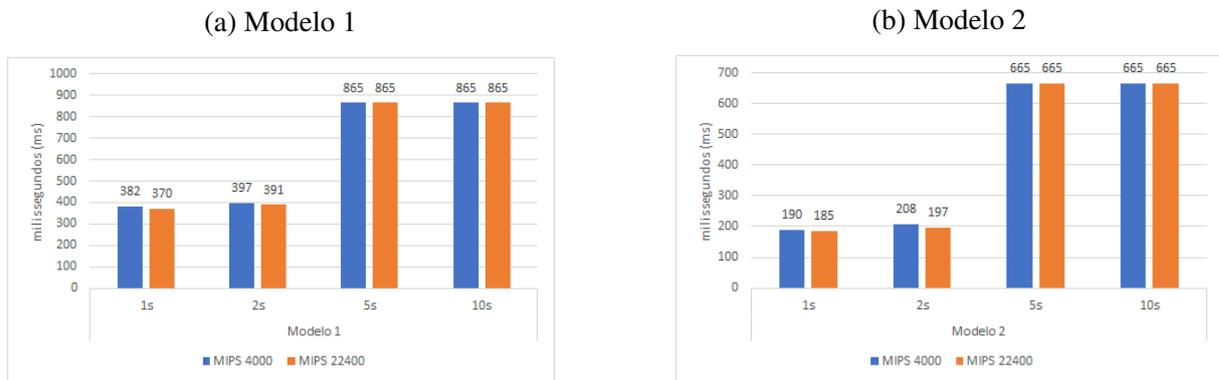
Tabela 11 – Configuração dos Dispositivos *Fog* com alteração do MIPS

Atributo	Antes	Depois
MIPS	4000	22400
RAM (MB)	10240	10240
Uplink (MB)	10000	10000
Downlink (MB)	10000	10000
Hierarquia	2	2

Fonte: Autor.

utilizando-se 128 (cento e vinte e oito) equipamentos móveis. No Modelo 1, a [Figura 31a](#) mostra que houve aumento em todos os casos, com exceção da transmissão a cada 5s, que reduziu aproximadamente 0,59%. No Modelo 2, a [Figura 31b](#) mostra que reduziu para transmissão de 5s

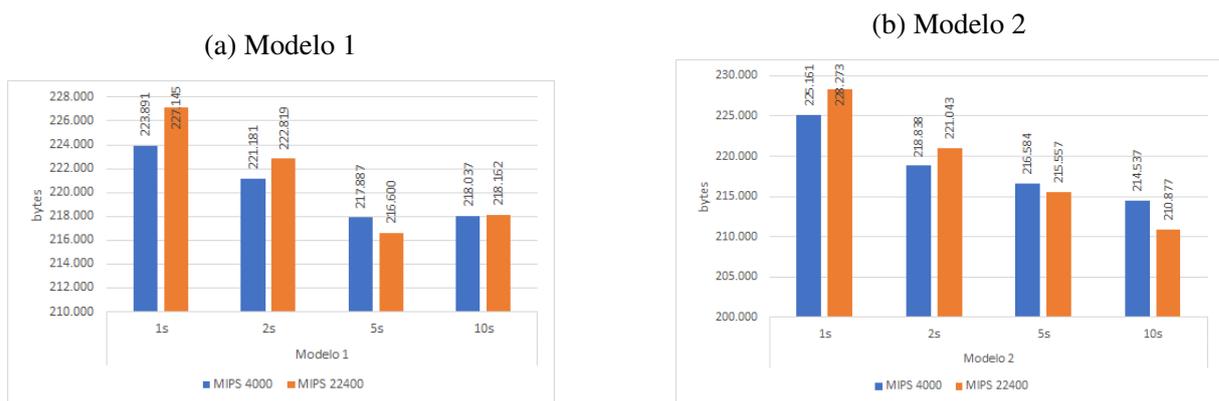
Figura 30 – Média do *Application Loop Delay* com alteração no MIPS



Fonte: Autor.

e 10s, sendo a maior redução de aproximadamente 1,71% obtida ao transmitir a cada 10s.

Figura 31 – Média do *Network Usage* com alteração no MIPS



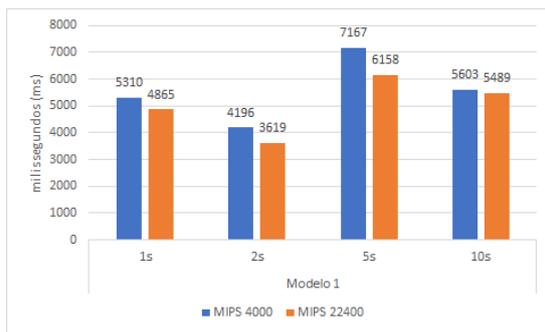
Fonte: Autor.

A Figura 32 mostra a média do execution time comparando o MIPS 4000 e MIPS 22400 utilizando-se 128 (cento e vinte e oito) equipamentos móveis. Pode-se observar na Figura 32a que houve redução em todos os casos. A maior redução ocorreu ao transmitir a cada cinco segundos, essa redução foi de aproximadamente 14,07%. No Modelo 2, a Figura 32b mostra que houve um aumento de aproximadamente 1,91% para transmissão de 1s, e redução para transmissão de 2s, 5s e 10s. A maior redução de aproximadamente 14,05% foi obtida ao transmitir a cada 5s.

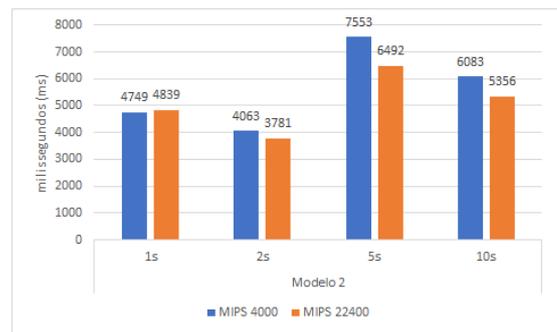
Cabe ressaltar que, apesar do incremento no tempo de execução da simulação ao aumentar o número de nós, a viabilidade da execução da simulação não foi afetada, pois em poucos segundos já se obtém o resultado desejado.

Figura 32 – Média do *Execution Time* com alteração no MIPS

(a) Modelo 1



(b) Modelo 2



Fonte: Autor.

5

Conclusão

Este trabalho realizou avaliação de uma proposta para localização interna sugerindo seu uso na localização de equipamentos hospitalares propondo dois modelos de aplicação (Modelo 1 e Modelo 2) e três cenários diferentes (*Edge*, *Cloud* e *Fog Computing*) para avaliar a localização interna de equipamentos hospitalares. No Modelo 1, o cenário 2 (*Cloud Computing*) obteve melhor resultado para *Application Loop Delay*, o que significa que em menor tempo já será possível saber a localização do equipamento. Porém, no Modelo 2, o menor tempo apresentado foi o cenário 3 (*Fog Computing*), sendo também inferior aos melhores tempos do Modelo 2.

Foi possível notar que à medida que aumenta o número de equipamentos/sensores na rede, aumenta também o tempo de resposta da aplicação, assim como o tráfego de dados na rede e também o tempo de execução do simulador tanto no Modelo 1 como no Modelo 2. Este trabalho também observou que alterar a capacidade de processamento (MIPS) dos equipamentos localizados na *Fog Computing* não houve ganhos significativos para o *Application Loop Delay* tendo em vista o alto custo de equipamentos com maior poder computacional. O *execution time*, como esperado, foi o principal beneficiado do aumento no MIPS. Com equipamentos mais robustos, com maior capacidade de executar milhões de instruções por segundos, houve uma redução em todos os cenários, com exceção da transmissão a cada 1s no Modelo 2.

Na falta de recursos financeiros para contratação de servidores robustos, o cenário 1 (*Edge Computing*) é recomendado quando se tem poucos equipamentos na rede (até 16 equipamentos), sugere-se o uso para aplicações em ambientes da casa (*Smart Home*). O cenário 2 (*Cloud Computing*), é recomendado utilizar quando o *delay* de conexão com servidores na nuvem for irrelevante. O cenário 3 (*Fog Computing*), é recomendado utilizar quando o tempo de resposta da aplicação é de suma importância, sendo exibido logo após o processamento dos dados. Além de que, o cenário 3 permite que a aplicação continue em funcionamento em momentos de queda da internet, podendo sincronizar com os servidores na nuvem após a estabilização da internet.

5.1 Contribuições alcançadas

Este trabalho realizou testes e demonstrou o processo de simulação para a localização interna de equipamentos. Foi detalhado a ferramenta e cargas utilizadas bem como resultados alcançados em cada etapa, tornando possível a replicação do experimento. Dessa forma, os seguintes itens também foram alcançados:

- desenvolvimento do mapeamento sistemático
- identificado que a *Fog Computing* obteve melhor desempenho no tempo de resposta da latência média utilizando o Modelo 2 (*Display before Storage*)
- demonstrado redução do tráfego de rede após aumento do MIPS com transmissão de 5s ou 10 segundos
- demonstrado que o aumento do MIPS não compensa nos demais casos
- identificado que o uso da *Edge Computing*, com retransmissão de dados a cada 2s, 5s ou 10 segundos tende a tornar ruim o desempenho de um sistema a partir de 16 equipamentos

5.2 Dificuldades e Limitações

No decorrer deste trabalho, algumas dificuldades e limitações foram encontradas. Devido à pandemia da Covid 19, além do acesso restrito ao ambiente hospitalar, as empresas que entramos em contato tiveram dificuldades em fornecer material para os testes em ambiente real. Com isso, foi necessário desenvolver o trabalho exclusivamente em ambiente de simulação e encontrar as ferramentas adequadas para tal. Com o uso do iFogSim, houve dificuldades em entender o funcionamento e as unidades de medidas utilizadas pelo simulador, o que acarretou em atrasos no cronograma dessa dissertação.

Com o experimento por meio de simulação foi possível avaliar o *Application Loop Delay*, *Network Usage* e o *Execution Time*, sem levar em consideração se o posicionamento dos equipamentos podem ser lidos por mais de um *gateway* gerando dados ambíguos. Outra limitação do trabalho foi que os pontos escolhidos arbitrariamente no Google Maps podem não representar uma ala/sala real.

5.3 Trabalhos Futuros

Como trabalho futuro sugere-se a experimentação em ambiente hospitalar controlado, assim como avaliar o posicionamento real dos *gateways*, coletando a geolocalização das salas no próprio ambiente proposto (uma vez que o simulador utiliza as coordenadas para realizar a simulação), analisar seu funcionamento e verificar possíveis interferência entre eles. É interessante

também avaliar o impacto de diferentes configurações de rede, a exemplo da latência entre *Fog* e *Proxy*, *Proxy* e *Cloud*.

5.4 Publicações Relacionadas à Dissertação

Nesta seção são apresentadas as contribuições relacionadas a esta dissertação.

5.4.1 Artigo Aprovado

IoT Applications in Universities: a systematic mapping, ICIOT-2021 6th International Conference Held as Part of the Services Conference Federation, SCF 2021, Qualis A4.

5.4.2 Artigo Submetido

Evaluation of fog computing to support indoor tracking systems: A simulation study for hospital equipment management, Qualis A1.

Referências

- ADARSH, T. K.; VIJAYAKUMAR, K. P. A classical method for health monitoring on IoT: From architecture, security & application. *International Journal of Innovative Technology and Exploring Engineering*, Blue Eyes Intelligence Engineering and Sciences Publication, v. 8, n. 11 Special Issue, p. 1043–1046, sep 2019. ISSN 22783075. Citado 3 vezes nas páginas 25, 32 e 35.
- AMADEO, M. et al. Fog Computing in IoT Smart Environments via Named Data Networking: A Study on Service Orchestration Mechanisms. *Future Internet*, MDPI AG, v. 11, n. 11, p. 222, oct 2019. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/11/11/222>>. Citado na página 20.
- Amazon Web Services. *Aws iot: Guia do desenvolvedor*. 2017. Disponível em: <<http://docs.aws.amazon.com/pt-br/iot/latest/developerguide/iot-dg>> Citado na página 39.
- BONAFINI, F. et al. Evaluating indoor and outdoor localization services for LoRaWAN in Smart City applications. *2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings*, p. 300–305, 2019. Citado na página 20.
- CHANG, Y. C.; LAI, Y. H. Campus Edge Computing Network Based on IoT Street Lighting Nodes. *IEEE Systems Journal*, Institute of Electrical and Electronics Engineers Inc., v. 14, n. 1, p. 164–171, mar 2020. ISSN 19379234. Citado na página 20.
- CRUZ, M. A. D. et al. *A Reference Model for Internet of Things Middleware*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. 871–883 p. Citado na página 19.
- DIN, S.; PAUL, A. Smart health monitoring and management system: Toward autonomous wearable sensing for Internet of Things using big data analytics. *Future Generation Computer Systems*, v. 91, p. 611–619, 2019. Disponível em: <<https://doi.org/10.1016/j.future.2017.12.059>>. Citado na página 25.
- FALBO, R. D. A. *Mapeamento Sistemático*. n. 2010, 2013. Citado 2 vezes nas páginas 17 e 27.
- FAMA, F. B. G.; SANTOS, D. F. S.; PERKUSICH, A. Edge computing: Um estudo sobre ferramentas de simulação e suas características. *Sociedad Brasileira de Telecomunicacoes*, 2018. Disponível em: <<http://www.smartsantander.eu/>>. Citado na página 23.
- FONSECA, J. J. S. da. *Apostila de metodologia da pesquisa científica*. [S.l.]: João José Saraiva da Fonseca, 2002. Citado na página 18.
- GERHARDT, T. E. et al. *Métodos de pesquisa*. [s.n.], 2009. ISBN 9788538600718. Disponível em: <<http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf>>. Citado na página 17.
- GIL, A. C. et al. *Como elaborar projetos de pesquisa*. [S.l.]: Atlas São Paulo, 2002. v. 4. Citado na página 17.
- GRECO, L. et al. Trends in IoT based solutions for health care: Moving AI to the edge. *Pattern Recognition Letters*, Elsevier B.V., v. 135, p. 346–353, jul 2020. ISSN 01678655. Citado na página 24.

- GUPTA, H. et al. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software - Practice and Experience*, v. 47, n. 9, p. 1275–1296, 2017. ISSN 1097024X. Citado 2 vezes nas páginas 22 e 23.
- HAGGAG, M.; OUF, A. *PERFORMANCE EVALUATION OF VIDEO PROCESSING AT THE EDGE*. 2020. Citado na página 14.
- HOU, J. L.; YEH, K. H. Novel Authentication Schemes for IoT Based Healthcare Systems. *International Journal of Distributed Sensor Networks*, v. 2015, n. ii, 2015. ISSN 15501477. Citado 2 vezes nas páginas 34 e 35.
- ISLAM, S. M. et al. The internet of things for health care: A comprehensive survey. *IEEE Access*, IEEE, v. 3, p. 678–708, 2015. ISSN 21693536. Citado na página 24.
- KANAN, R.; ELHASSAN, O. A combined batteryless radio and wifi indoor positioning for hospital nursing. *Journal of Communications Software and Systems*, University of Split, v. 12, n. 1, p. 34–44, 2016. ISSN 18466079. Citado 2 vezes nas páginas 32 e 35.
- KIM, T. Y.; LIM, J. B. An edge cloud-based body data sensing architecture for artificial intelligence computation. *International Journal of Distributed Sensor Networks*, SAGE Publications Ltd, v. 15, n. 4, apr 2019. ISSN 15501477. Disponível em: <<https://us.sagepub.com/en-us/nam/>>. Citado 3 vezes nas páginas 22, 33 e 35.
- KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007. Citado na página 27.
- LACHTAR, A.; VAL, T.; KACHOURI, A. Elderly monitoring system in a smart city environment using LoRa and MQTT. *IET Wireless Sensor Systems*, Institution of Engineering and Technology (IET), v. 10, n. 2, p. 70–77, 2020. ISSN 2043-6386. Disponível em: <<http://10.0.4.25/iet-wss.2019.0121>><<https://dx.doi.org/10.1049/iet-wss.2019.0121>>. Citado 2 vezes nas páginas 33 e 35.
- LAWAL, K. N.; OLANIYI, T. K.; GIBSON, R. M. Fog Computing Infrastructure Simulation Toolset Review for Energy Estimation, Planning and Scalability. *International Journal of Sustainable Energy Development*, v. 9, n. 1, p. 421–426, 2021. Citado na página 23.
- LENG, J.; LIN, Z.; WANG, P. Poster abstract: An implementation of an internet of things system for smart hospitals. *Proceedings - 5th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2020*, p. 254–255, 2020. Citado 2 vezes nas páginas 13 e 16.
- LOPES, S. P. *Fog computing e a IoT - Engenho & Engenhocas*. 2018. Disponível em: <<http://engenhoe Engenhocas.blogspot.com/2018/01/fog-computing-e-iot.html>>. Citado na página 23.
- LÓPEZ, Y. Álvarez et al. RFID technology for management and tracking: E-health applications. *Sensors (Switzerland)*, MDPI AG, v. 18, n. 8, aug 2018. ISSN 14248220. Citado 3 vezes nas páginas 25, 32 e 35.
- LU, S. et al. A study on service-oriented smart medical systems combined with key algorithms in the IoT environment. *China Communications*, China Institute of Communications, v. 16, n. 9, p. 235–249, 2019. ISSN 16735447. Citado na página 16.

MAHMUD, R. et al. IFogSim2: An Extended iFogSim Simulator for Mobility, Clustering, and Microservice Management in Edge and Fog Computing Environments. *arXiv preprint arXiv:2109.05636*, 2021. Disponível em: <<http://arxiv.org/abs/2109.05636>>. Citado na página 14.

MOHANTY, S. P.; KOUIGIANOS, E.; GUTURU, P. SBPG: Secure Better Portable Graphics for Trustworthy Media Communications in the IoT. *IEEE Access*, v. 6, p. 5939–5953, 2018. ISSN 2169-3536 VO - 6. Citado na página 13.

NIC.BR. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos estabelecimentos de saúde brasileiros - TIC Saúde 2019*. 2019. Disponível em: <<https://www.nic.br/publicacao/pesquisa-sobre-o-uso-das-tecnologias-de-informacao-e-comunicacao-nos-estabelecimentos-de-saude-brasil>>. Citado na página 15.

NIC.BR. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos estabelecimentos de saúde brasileiros - TIC Saúde 2021*. 2021. Disponível em: <<https://www.nic.br/publicacao/pesquisa-sobre-o-uso-das-tecnologias-de-informacao-e-comunicacao-nos-estabelecimentos-de-saude-brasil>>. Citado na página 15.

OLIVEIRA, A. L. *Por que implantar sistemas ciber físicos na sua empresa*. 2020. Disponível em: <<https://certi.org.br/blog/sistemas-ciber-fisicos/>>. Citado na página 14.

OUEIDA, S. et al. An edge computing based smart healthcare framework for resource management. *Sensors (Switzerland)*, MDPI AG, v. 18, n. 12, dec 2018. ISSN 14248220. Citado na página 25.

PASCHOAL, M. L. H.; CASTILHO, V. Consumo de materiais em centro cirúrgico após implementação de sistema de gestão informatizado. *Revista brasileira de enfermagem*, v. 63, n. 6, p. 887–893, 2010. ISSN 00347167. Citado na página 15.

PASHA, M.; SHAH, S. M. W. Framework for E-Health Systems in IoT-Based Environments. *Wireless Communications and Mobile Computing*, Hindawi Limited, v. 2018, 2018. ISSN 15308677. Citado na página 15.

POLENTA, A. et al. An internet of things approach to contact tracing-the bubblebox system. *Information (Switzerland)*, MDPI AG, v. 11, n. 11, jul 2020. ISSN 20782489. Citado 2 vezes nas páginas 33 e 35.

RATHERT, C. et al. *Seven years after Meaningful Use: Physicians' and nurses' experiences with electronic health records - PubMed*. 2019. <<https://pubmed-ncbi-nlm-nih.ez20.periodicos.capes.gov.br/28614166/>>. (Accessed on 11/10/2020). Citado na página 16.

RATHORE, M. M. et al. Hadoop-based intelligent care system (hics): Analytical approach for big data in iot. *ACM Transactions on Internet Technology*, Association for Computing Machinery, v. 18, p. 1–24, 11 2017. ISSN 15576051. Disponível em: <<https://dl.acm.org/doi/10.1145/3108936>>. Citado 2 vezes nas páginas 20 e 21.

RUBÍ, J. N.; GONDIM, P. R. Iomt platform for pervasive healthcare data aggregation, processing, and sharing based on onem2m and openehr. *Sensors (Switzerland)*, MDPI AG, v. 19, 10 2019. ISSN 14248220. Citado na página 30.

SAHA, R. et al. Privacy ensured e-Healthcare for fog-enhanced IoT based applications. *IEEE Access*, v. 7, p. 44536–44543, 2019. ISSN 21693536. Citado 2 vezes nas páginas 33 e 35.

SANCHEZ-GALLEGOS, D. D. et al. On the Continuous Processing of Health Data in Edge-Fog-Cloud Computing by Using Micro/Nanoservice Composition. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 8, p. 120255–120281, 2020. ISSN 21693536. Citado na página 24.

SCHENFELD, M. C. et al. Fog e edge computing: uma arquitetura híbrida em um ambiente de internet das coisas. Pontifícia Universidade Católica do Rio Grande do Sul, 2017. Citado na página 34.

SHI-LIN, H. et al. Development of physiological parameters gateway for medical iot ward. *Open Automation and Control Systems Journal*, Bentham Science Publishers B.V., v. 7, p. 1667–1673, 10 2015. ISSN 18744443. Citado 3 vezes nas páginas 30, 32 e 35.

SHUKLA, S. et al. A 3-tier architecture for network latency reduction in healthcare internet-of-things using fog computing and machine learning. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, Part F147956, n. February, p. 522–528, 2019. ISSN 21531633. Citado na página 22.

SOGIMIG. *Informatização na saúde reduz custos operacionais e aumenta eficiência operacional - SOGIMIG - Nós por Elas*. 2014. <<http://www.sogimig.org.br/informatizacao-na-saude-reduz-custos-operacionais-e-aumenta-eficiencia-operacional/>>. (Accessed on 11/04/2020). Citado na página 15.

SONG, Y. et al. Prospect and application of Internet of Things technology for prevention of SARIs. *Clinical eHealth*, Elsevier BV, v. 3, p. 1–4, jan 2020. ISSN 25889141. Citado na página 26.

SUBAHI, A. F. Edge-Based IoT Medical Record System: Requirements, Recommendations and Conceptual Design. *IEEE Access*, v. 7, p. 94150–94159, 2019. ISSN 2169-3536 VO - 7. Citado 4 vezes nas páginas 21, 30, 33 e 35.

TEJONIDHI, M. R.; SUNITHA, M. R.; SWATHI, H. Y. A Survey on IOT-Middleware and Image Processing. *1st IEEE International Conference on Advances in Information Technology, ICAIT 2019 - Proceedings*, p. 210–216, 2019. Citado 2 vezes nas páginas 19 e 20.

THINGSPEAK. *ThingSpeak for IoT Projects*. 2021. <<https://thingspeak.com/>>. (Accessed on 01/15/2021). Citado na página 39.

TIAN, S. et al. Smart healthcare: making medical care more intelligent. *Global Health Journal*, Elsevier BV, v. 3, n. 3, p. 62–65, sep 2019. ISSN 24146447. Citado na página 24.

TRIGO, J. D. et al. Patient tracking in a multi-building, tunnel-connected hospital complex. *IEEE Sensors Journal*, v. 20, n. 23, p. 14453–14464, 2020. Citado na página 34.

TRIPATHI, G.; AHAD, M. A.; PAIVA, S. Sms: A secure healthcare model for smart cities. *Electronics (Switzerland)*, MDPI AG, v. 9, n. 7, p. 1–18, jul 2020. ISSN 20799292. Citado na página 20.

TSENG, T. W.; WU, C. T.; LAI, F. Threat analysis for wearable health devices and environment monitoring internet of things integration system. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 144983–144994, 2019. ISSN 21693536. Citado na página 31.

UNGUREAN, L.; BREZULIANU, A. An internet of things framework for remote monitoring of the healthcare parameters. *Advances in Electrical and Computer Engineering*, v. 17, n. 2, p. 11–16, 2017. ISSN 18447600. Citado na página 30.

VENČKAUSKAS, A. et al. Model-driven approach for body area network application development. *Sensors (Switzerland)*, MDPI AG, v. 16, 5 2016. ISSN 14248220. Citado na página 30.

VIJAYALAKSHMI, A.; ABISHEK, B. E.; STEPHEN, C. A. IoT based sensor fusion algorithm for online smart health monitoring. *International Journal of Recent Technology and Engineering*, Blue Eyes Intelligence Engineering and Sciences Publication, v. 8, n. 2, p. 6040–6046, jul 2019. ISSN 22773878. Citado 2 vezes nas páginas 13 e 39.

VILLEGAS-CH, W. et al. Application of a Big Data Framework for Data Monitoring on a Smart Campus. *Sustainability*, MDPI AG, v. 11, n. 20, p. 5552, oct 2019. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/11/20/5552>>. Citado na página 19.