



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
MESTRADO ACADÊMICO EM COMPUTAÇÃO

LUCAS FERREIRA MENDES

**UMA ARQUITETURA BASEADA EM IOT PARA SISTEMAS DE POSICIONAMENTO
INTERNO COMO SUPORTE A APLICAÇÕES DE AMBIENTE DE VIDA ASSISTIDA**

QUIXADÁ

2023

LUCAS FERREIRA MENDES

UMA ARQUITETURA BASEADA EM IOT PARA SISTEMAS DE POSICIONAMENTO
INTERNO COMO SUPORTE A APLICAÇÕES DE AMBIENTE DE VIDA ASSISTIDA

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

Orientador: Prof. Dr. Paulo Armando Cavalcante Aguiar

Coorientadora: Profa. Dra. Carla Ilane Moreira Bezerra

QUIXADÁ

2023

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

M491a Mendes, Lucas Ferreira.
Uma arquitetura baseada em IoT para sistemas de posicionamento interno como suporte a aplicações de ambiente de vida assistida / Lucas Ferreira Mendes. – 2023.
113 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Quixadá, Programa de Pós-Graduação em Computação, Quixadá, 2023.

Orientação: Prof. Dr. Paulo Armando Cavalcante Aguiar.
Coorientação: Profa. Dra. Carla Ilane Moreira Bezerra.

1. Arquitetura de Software. 2. Fog Computing. 3. Internet das Coisas. 4. Sistemas de Posicionamento Interno. 5. Vida Assistida pelo Ambiente. I. Título.

CDD 005

LUCAS FERREIRA MENDES

UMA ARQUITETURA BASEADA EM IOT PARA SISTEMAS DE POSICIONAMENTO
INTERNO COMO SUPORTE A APLICAÇÕES DE AMBIENTE DE VIDA ASSISTIDA

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

Aprovada em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. Paulo Armando Cavalcante
Aguilar (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Carla Ilane Moreira
Bezerra (Coorientadora)
Universidade Federal do Ceará (UFC)

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

Prof. Dr. Rodrigo Varejão Andreão
Instituto Federal de Educação, Ciência e Tecnologia
do Espírito Santo (IFES)

Dedico este trabalho à minha família, em especial minha mãe Francisca e minha irmã Adrileia, por se fazerem presentes e serem o apoio fundamental para que essa trajetória fosse possível, sempre com muito amor, carinho e compreensão.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida e por sempre me guiar pelos melhores caminhos, sem ele nada seria possível.

Agradeço imensamente à minha família, em especial aos meus pais Luís Ferreira Lima e Francisca Mendes Lima, por estarem sempre presentes em minha vida, me incentivando, me apoiando, sem nunca medirem esforços para que eu tivesse acesso à educação, mesmo com todas as dificuldades que se fizeram presentes.

À minha irmã, Adrileia Ferreira Mendes, por sempre estar comigo e ser, muitas vezes, uma base de apoio para não desistir desta caminhada. O meu muito obrigado por todas as renúncias que você fez para me acompanhar e me apoiar durante todo meu percurso acadêmico.

Agradeço aos meus amigos, por tornar esta trajetória mais leve, por todos os momentos alegres que compartilhamos, por todo o incentivo e motivação que sempre me deram. Agradeço, em especial, ao meu amigo Sidney José Rogrigues Lima, que esteve sempre presente e contribuiu diretamente com este trabalho.

Agradeço também a todos que compõem o IFCE - *Campus* Tauá, por todo o suporte e incentivo durante este percurso formativo. Não somente enquanto servidor da instituição, mas como aluno do curso de Tecnologia em Telemática, onde sempre recebi todas as oportunidades de desenvolvimento acadêmico, pessoal e profissional.

Agradeço fortemente ao meu orientador, professor Paulo Armando Cavalcante Aguiar, por ter acreditado na minha capacidade e por dividir sua experiência e seus ensinamentos extremamente valiosos. Agradeço também de forma muito especial à minha coorientadora, professora Carla Ilane Moreira Bezerra, por aceitar contribuir com esta jornada e por me fazer acreditar mais em mim mesmo e em minha capacidade.

Por fim, gostaria de agradecer a mim mesmo pela resiliência e pela força de vontade durante esta jornada cheia de percalços e momentos difíceis.

“The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge. ”

(Daniel Joseph Boorstin)

RESUMO

Aplicações de Internet das Coisas (IoT) sensíveis ao contexto fornecem um ambiente ubíquo e inteligente com serviços para auxiliar as pessoas em suas atividades de vida diária. Nesse sentido, os Sistemas de Posicionamento Interno (IPS) desempenham um papel importante ao fornecer informações de localização interna, sendo informações essenciais para tais aplicações. Através dessas informações, as aplicações podem fornecer dados e serviços personalizados de acordo com as necessidades dos usuários. Assim, várias são as áreas de aplicações de sistemas IPS em uma infraestrutura de IoT, como o suporte a pessoas idosas ou com deficiências em suas residências ou ambientes de saúde. Esse é um tópico de pesquisa que tem ganhado bastante relevância, chamado de Ambiente de Vida Assistida (AAL). Contudo, percebe-se a ausência de um padrão ou estilo arquitetural bem definido, bem como de um entendimento claro sobre quais requisitos não funcionais (RNF) são mais relevantes para aplicações nesse domínio. Este trabalho propõe uma arquitetura de software para aplicações IPS baseadas em IoT para dar suporte a sistemas AAL. Realizou-se uma revisão sistemática da literatura (RSL) para identificar soluções e RNFs a serem incorporados à arquitetura de tais aplicações. A partir dos resultados da RSL, foi projetada e modelada uma arquitetura de software para aplicações IPS baseadas em IoT, usando elementos de *Fog Computing*, para suportar sistemas AAL. A proposta foi avaliada usando o ambiente de simulação iFogSim com relação à latência, uso da rede, consumo de energia e custos computacionais e operacionais em nuvem. A abordagem proposta foi comparada a uma implantação baseada somente em nuvem. Os resultados experimentais mostram que a arquitetura proposta reduz significativamente a latência, o consumo energético e os custos operacionais e computacionais em nuvem e é adequada para cenários de resposta em tempo real.

Palavras-chave: Arquitetura de Software; Fog Computing; Internet das Coisas; Sistemas de Posicionamento Interno; Vida Assistida pelo Ambiente.

ABSTRACT

Context-aware Internet of Things (IoT) applications provide a ubiquitous and intelligent environment with services to assist people in their daily life activities. Thus, Internal Positioning Systems (IPS) play an important role in providing indoor location information, which is essential for such applications. This information allows applications to provide personalized data and services according to users' needs. Thus, there are several application areas for IPS systems in an IoT infrastructure, such as supporting elderly or disabled people in their homes or healthcare environments. This is a research topic that has been gaining a lot of relevance, called Ambient Assisted Living (AAL). However, there is a lack of a well-defined architectural pattern or style, as well as a clear understanding of which non-functional requirements (NFR) are most relevant to applications in this domain. This work proposes a software architecture for IoT-based IPS applications to support AAL systems. A systematic literature review (SLR) was carried out to identify solutions and NFRs to be incorporated into the architecture of such applications. Based on the SLR results, a software architecture was designed and modeled for IoT-based IPS applications using Fog Computing elements to support AAL systems. The proposal was evaluated using the iFogSim simulation environment with regard to latency, network usage, energy consumption, computing costs, and operational costs in the cloud. The proposed approach was compared to a cloud-only deployment. The experimental results show that the proposed architecture significantly reduces latency, energy consumption and operational and computational costs in the cloud and is suitable for real-time response scenarios.

Keywords: Software Architecture; Fog Computing; Internet of Things; Internal Positioning Systems; Ambient Assisted Living.

LISTA DE FIGURAS

Figura 1 – Metodologia utilizada	20
Figura 2 – Arquitetura básica de um sistema IPS baseado em servidor	24
Figura 3 – Técnica de Trilateração de Sinais	26
Figura 4 – Exemplo de aplicação do <i>fingerprinting</i>	28
Figura 5 – Arquiteturas básicas para IoT	30
Figura 6 – Arquitetura básica de paradigmas emergentes em IoT baseada em Cloud	32
Figura 7 – Arquitetura de Sistemas IoT definida em Pawar <i>et al.</i> (2021)	33
Figura 8 – Objetivos de soluções de AAL	35
Figura 9 – Características de uma solução de AAL	36
Figura 10 – Trabalhos selecionados por ano de publicação	47
Figura 11 – Palavras-chave utilizadas nos trabalhos	48
Figura 12 – Visão geral da arquitetura proposta por Mainetti <i>et al.</i> (2016)	56
Figura 13 – Características da arquitetura proposta por Mainetti <i>et al.</i> (2016)	56
Figura 14 – Requisitos não funcionais considerados	58
Figura 15 – Arquitetura proposta	62
Figura 16 – Modelo de visão 4+1 de arquitetura de software	65
Figura 17 – Principal diagrama de sequência da arquitetura proposta	66
Figura 18 – Diagrama de classes geral da arquitetura proposta	67
Figura 19 – Diagrama de componentes da camada de percepção	68
Figura 20 – Diagrama de componentes da camada fog	69
Figura 21 – Diagrama de componentes da camada de nuvem	70
Figura 22 – Diagrama de componentes da camada de aplicação	71
Figura 23 – Diagrama de atividades para detecção de eventos críticos	72
Figura 24 – Diagrama de implantação da arquitetura proposta	73
Figura 25 – Diagrama geral de casos de uso da arquitetura proposta	74
Figura 26 – Ambiente de aplicação do exeperimento	78
Figura 27 – Visão geral dos componentes do experimento de localização interna	80
Figura 28 – Diagrama de sequência da aplicação do experimento de localização interna	81
Figura 29 – Sistema AAL em um cenário de <i>Smart City</i>	84
Figura 30 – Topologia genérica definida no iFogSim para a estratégia baseada em Fog	85
Figura 31 – Modelo de aplicação utilizado na simulação	86

Figura 32 – Latência observada variando o número de áreas de cobertura	88
Figura 33 – Uso de rede observado variando o número de áreas de cobertura	89
Figura 34 – Consumo energético observado variando o número de áreas de cobertura . .	90
Figura 35 – Custo de execução em nuvem observado variando o número de áreas de cobertura	91
Figura 36 – Uso de CPU em nuvem observado variando o número de áreas de cobertura	92
Figura 37 – Evolução do uso de CPU em nuvem na configuração com 3 <i>mist nodes</i> . . .	92
Figura 38 – Evolução do uso de CPU em nuvem na configuração com 6 <i>mist nodes</i> . . .	93

LISTA DE TABELAS

Tabela 1 – Comparação entre trabalhos relacionados	42
Tabela 2 – Critérios de seleção	45
Tabela 3 – Processo de seleção dos trabalhos	46
Tabela 4 – Trabalhos selecionados na RSL	46
Tabela 5 – Principais características das soluções propostas	49
Tabela 6 – Tecnologias, técnicas e métodos para localização interna	50
Tabela 7 – Tecnologias de comunicação e computação em nuvem	51
Tabela 8 – Requisitos não funcionais abordados nos trabalhos selecionados	52
Tabela 9 – Decisões de Projeto	59
Tabela 10 – Descrição resumida dos casos de uso definidos	74
Tabela 11 – FRP utilizado pelos dispositivos <i>scanners</i>	82
Tabela 12 – FRP utilizado pelo dispositivo coordenador para comunicação com os <i>scanners</i>	82
Tabela 13 – FRP utilizado pelo dispositivo coordenador para comunicação com o <i>home gateway</i>	82
Tabela 14 – Comparação de desempenho na determinação do cômodo correto	83
Tabela 15 – Valores dos parâmetros dos dispositivos para a simulação	86
Tabela 16 – Variações no cenário de avaliação	87
Tabela 17 – Artigos publicados em conferências	95
Tabela 18 – Formulário de Extração de Dados da RSL	105
Tabela 19 – Parâmetros de configuração da simulação no iFogSim	106
Tabela 20 – Resultados obtidos em todos os cenários simulados	111

LISTA DE ABREVIATURAS E SIGLAS

IPS	Sistemas de Posicionamento Interno / <i>Internal Positioning System</i>
IoT	Internet das Coisas / <i>Internet of Things</i>
AAL	Ambiente de Vida Assistida / <i>Ambient Assisted Living</i>
ADL	Atividades de Vida Diária / <i>Activities Daily Living</i>
RSSF	Redes de Sensores sem Fio
RNF	Requisitos não Funcionais
RSL	Revisão Sistemática da Literatura
GPS	Sistema de Posicionamento Global / <i>Global Positioning System</i>
GNSS	Sistemas Globais de Navegação por Satélite / <i>Global Navigation Satellite System</i>
RF	radiofrequência
RFID	Identificação por Radiofrequência / <i>Radio Frequency Identification</i>
NFC	<i>Near Field Communication</i>
BLE	<i>Bluetooth Low Energy</i>
RSSI	Indicador de Intensidade do Sinal Recebido / <i>Received Signal Strength Indication</i>
PR	ponto de referência
IMU	Unidades de Medição Inercial / <i>Inertial Measurement Units</i>
PDR	<i>Pedestrian Dead-Reckoning</i>
FK	Filtro Kalman
FP	Filtro de Partículas
ZUPT	<i>Zero-velocity UPdaTes</i>
TIC	Tecnologias da Informação e Comunicação
IoHT	<i>Internet of Health Things</i>
BAN	Rede de Área Corporal / <i>Body Area Network</i>
MCI	<i>Micro Computing Instances</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
GPRS	<i>General Packet Radio Service</i>
DDF	Fluxo de Dados Distribuído / <i>Distributed Data Flow</i>
DAG	Grafo Acíclico Dirigido / <i>Directed Acyclic Graph</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Motivação e Contextualização do Problema	16
1.2	Objetivos e Questões de Pesquisa	18
1.3	Metodologia Adotada	20
1.4	Estrutura e Organização do Trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Sistemas de Posicionamento Interno	22
2.2	Principais Tecnologias e Técnicas para Localização Interna	23
2.2.1	<i>Localização Baseada em RF</i>	24
2.2.2	<i>Localização Baseada em Medições Inerciais</i>	27
2.2.3	<i>Outras Tecnologias para Localização Interna</i>	29
2.3	Arquitetura de Software para Aplicações IoT	29
2.3.1	<i>Novos Paradigmas de Computação em Nuvem</i>	31
2.4	Ambiente de Vida Assistida	33
2.4.1	<i>Sistemas IoT para AAL</i>	35
2.4.2	<i>Localização Interna para AAL</i>	37
3	TRABALHOS RELACIONADOS	39
3.1	Arquiteturas de software para aplicações AAL baseadas em IoT	39
3.2	Propostas em AAL com uso de informação contextual de localização interna	40
3.3	Comparação dos trabalhos relacionados	42
4	REVISÃO SISTEMÁTICA E ELICITAÇÃO DE REQUISITOS	44
4.1	Revisão Sistemática	44
4.1.1	<i>Discussão sobre a questão de pesquisa RQ1</i>	47
4.1.2	<i>Discussão sobre a questão de pesquisa RQ2</i>	50
4.1.2.1	<i>Acurácia</i>	51
4.1.2.2	<i>Tempo de Resposta</i>	53
4.1.2.3	<i>Eficiência Energética</i>	53
4.1.2.4	<i>Integração com Localização Externa</i>	54
4.1.2.5	<i>Escalabilidade</i>	54

4.1.2.6	<i>Privacidade e Segurança</i>	54
4.1.3	<i>Discussão sobre a Questão de Pesquisa RQ3</i>	55
4.2	Elicitação de Requisitos	57
5	ARQUITETURA PROPOSTA	59
5.1	Camada de Percepção	61
5.2	Camada Fog	63
5.3	Camada de Nuvem	63
5.4	Camada de Aplicação	64
5.5	Modelos Abstratos da Arquitetura Proposta	65
5.5.1	<i>Visão Lógica</i>	65
5.5.2	<i>Visão de Implementação</i>	67
5.5.2.1	<i>Componentes da Camada de Percepção</i>	67
5.5.2.2	<i>Componentes da Camada Fog</i>	68
5.5.2.3	<i>Componentes da Camada de Nuvem</i>	69
5.5.2.4	<i>Componentes da Camada de Aplicação</i>	71
5.5.3	<i>Visão de Processo</i>	72
5.5.4	<i>Visão de Implantação</i>	72
5.5.5	<i>Visão de Cenários</i>	73
6	AVALIAÇÃO DE DESEMPENHO DA ARQUITETURA PROPOSTA .	77
6.1	Experimento de Localização Interna	77
6.2	Projeto de Experimento da Simulação com iFogSim	83
6.3	Resultados da Simulação com iFogSim	87
7	CONSIDERAÇÕES FINAIS	94
7.1	Publicações	94
7.2	Principais Contribuições	94
7.3	Limitações do Trabalho	95
7.4	Trabalhos Futuros	96
	REFERÊNCIAS	97
	APÊNDICES	105
	APÊNDICE A –FORMULÁRIO DE EXTRAÇÃO DE DADOS DA RSL	105
	APÊNDICE B –SETUP DA SIMULAÇÃO NO IFOGSIM	106

APÊNDICE C –RESULTADOS COMPLETOS PARA TODOS OS CE- NÁRIOS SIMULADOS	111
--	------------

1 INTRODUÇÃO

Neste Capítulo são apresentadas a contextualização e a motivação desta pesquisa que tem o objetivo de propor uma arquitetura de software para Sistemas de Posicionamento Interno / *Internal Positioning System* (IPS) baseados em uma infraestrutura de Internet das Coisas / *Internet of Things* (IoT) para prover suporte a aplicações de Ambiente de Vida Assistida / *Ambient Assisted Living* (AAL).

Na Seção 1.1 são apresentadas e discutidas a motivação e a contextualização do problema. Na Seção 1.2 é apresentado o objetivo geral desta dissertação, bem como os objetivos específicos e questões de pesquisa. Na Seção 1.3 é descrita a metodologia adotada neste trabalho a fim de responder às questões de pesquisa e alcançar os objetivos definidos. Por fim, a estrutura e organização desta dissertação é apresentada na Seção 1.4.

1.1 Motivação e Contextualização do Problema

Aplicações sensíveis ao contexto estão se tornando comuns com a implantação de serviços baseados em IoT de forma mais próxima aos usuários finais, como casas inteligentes (*smart homes*), cidades inteligentes (*smart cities*) e até mesmo serviços de cuidado e assistência à saúde (*healthcare*). Nesse sentido, informações contextuais como a localização de objetos e pessoas em ambientes internos têm sido cada vez mais importantes para tais aplicações (BASIRI *et al.*, 2017).

Nesse contexto, os sistemas IPS estão recebendo bastante atenção da comunidade acadêmica e científica, sendo um tema amplamente pesquisado e aplicado (MENDOZA-SILVA *et al.*, 2019). Aplicações como rastreamento e localização de pessoas, reconhecimento e rastreamento de objetos, navegação de robôs autônomos, rastreamento de animais e suporte a pessoas idosas ou com deficiências podem se beneficiar com o avanço e aprimoramento de tais sistemas (MAINETTI *et al.*, 2014).

Quando se trata de pessoas idosas e/ou com alguma incapacidade física ou mental, soluções IPS podem ser úteis no sentido de fornecer serviços de cuidado e monitoramento de saúde ou de auxílio em suas Atividades de Vida Diária / *Activities Daily Living* (ADL) baseados em sua localização. Dessa forma, pode-se ter um ambiente inteligente que auxilia tais pessoas a conseguirem melhor qualidade de vida e autonomia, tornando-se um ambiente de vida assistida. Além disso, com a capacidade de reconhecimento de localização e contexto, os

serviços fornecidos nesses ambientes são mais personalizados e proativos (KLEINBERGER *et al.*, 2007).

Uma das possibilidades de implementar uma solução IPS está na utilização de Redes de Sensores sem Fio (RSSF), visto que a localização de nós sensores, assim como as aplicações de rastreamento e localização tem sido um importante estudo na área (MOREIRA *et al.*, 2017). Com isso, a infraestrutura provida por soluções IoT se mostra bem aderente às necessidades e requisitos de sistemas IPS, visto que, em grande parte, tais soluções IoT utilizam RSSFs com a finalidade de prover um ambiente inteligente, ubíquo e com consciência de contexto (MACAGNANO *et al.*, 2014).

Contudo, ainda há vários desafios em soluções IoT, dado sua natureza extremamente heterogênea, o que implica na dificuldade de desenvolver arquiteturas de software para tais aplicações (KHAN *et al.*, 2012; GIRI *et al.*, 2017; LIN *et al.*, 2017; TAIVALSAARI; MIKKONEN, 2018; KRISHNAN; VASUDEVAN, 2019; NIKOUI *et al.*, 2021). Basicamente temos duas principais arquiteturas básicas definidas e bastante discutidas: arquitetura de três camadas e arquitetura de cinco camadas (KRISHNAN; VASUDEVAN, 2019). Partindo dessas proposições base, as arquiteturas de software para aplicações IoT se moldam a depender da natureza e finalidade das aplicações.

Assim, definir uma arquitetura de software para um sistema baseado em IoT não é uma tarefa trivial e várias decisões de projeto devem ser levadas em consideração para sua formulação. Algumas dessas decisões incluem o domínio de aplicação, as tecnologias empregadas, os protocolos de comunicação, os modelos de representação de dados e a topologia física do sistema (TAIVALSAARI; MIKKONEN, 2018; KRISHNAN; VASUDEVAN, 2019; GIRI *et al.*, 2017; KHANNA; KAUR, 2020).

Essas decisões de projeto envolvem fatores como: custos, segurança, eficiência energética, latência de comunicação, bem como requisitos do domínio de aplicação (TAIVALSAARI; MIKKONEN, 2018; KHANNA; KAUR, 2020). Com isso pode-se elencar várias categorias de arquiteturas, em uma visão a nível de aplicação, que vão desde proposições baseadas somente em comunicação entre dispositivos finais a aplicações baseadas em sistemas operacionais de tempo real e processamento em servidores (TAIVALSAARI; MIKKONEN, 2018).

A ausência de um padrão arquitetural bem definido, e um entendimento claro sobre os Requisitos não Funcionais (RNF) mais importantes a serem levados em consideração no projeto de um sistema no domínio de AAL, implicam na dificuldade de tornar interoperáveis

diversos serviços e sistemas em AAL. Por consequência, existem diversas soluções propostas pela academia e indústria para esse contexto, tratando desde monitoramento de ADL até aplicações de assistência à mobilidade. Porém, tais soluções muitas vezes não se comunicam entre si e não possuem capacidade de agregar um conjunto de dados e serviços aos usuários. Pelo contrário, tem-se vários sistemas especializados em um determinado contexto de ambiente de vida assistida que não se apresentam de fato escaláveis nem generalizáveis (GIRI *et al.*, 2017; AHMADI *et al.*, 2019).

Além disso, questões como segurança e privacidade são desafios presentes em qualquer área de aplicação de soluções IoT e mais fortemente em soluções voltadas ao monitoramento e assistência à saúde (AHMADI *et al.*, 2019). Essa problemática fica ainda mais evidente quando relacionamos essas soluções à informações contextuais de localização de usuários dada a natureza sensível de tais informações. Nesse caso, aplicações de IPS baseadas em IoT para ambientes de vida assistida necessitam abordar essa questão de projeto desde sua concepção até a fase de implantação. Na verdade, muitas aplicações voltadas a navegação e posicionamento internos não levam em consideração aspectos relativos à privacidade e segurança de seus usuários e informações (BASIRI *et al.*, 2017).

Nesse contexto, este trabalho propõe uma arquitetura de software para sistemas IPS, baseada em IoT, para suporte a aplicações no domínio de AAL. A arquitetura proposta foi projetada baseada em evidências, por meio da realização de uma Revisão Sistemática da Literatura (RSL), e foi validada por meio de simulações computacionais, utilizando a ferramenta iFogSim, em termos de latência, uso de rede, consumo energético, e custos operacionais e computacionais em nuvem.

1.2 Objetivos e Questões de Pesquisa

Com base nas discussões supracitadas, percebe-se a necessidade de uma padronização relativa à arquitetura de software utilizada por aplicações IPS baseadas em IoT para ambientes de vida assistida. Nesse contexto, este trabalho tem o objetivo geral de propor uma arquitetura de software IoT que explore informações contextuais de localização e navegação interna em ambientes de vida assistida e ofereça um padrão de gerenciamento de dados e serviços para pessoas dependentes de cuidados especiais.

A fim de alcançar o objetivo geral, este trabalho traz como objetivos específicos:

- a) selecionar, baseado em evidências, tecnologias, técnicas e métodos de localização

- interna que utilizem uma infraestrutura IoT;
- b) identificar os RNFs mais importantes em um contexto de aplicações IPS, baseadas em IoT, para suporte a ambientes de vida assistida;
 - c) propor uma arquitetura de software IoT para aplicações IPS em um contexto de AAL, com decisões de projeto baseadas em evidências; e
 - d) avaliar a arquitetura proposta por meio de simulações computacionais, visando analisar os requisitos de desempenho, eficiência energética, escalabilidade e custos operacionais e computacionais.

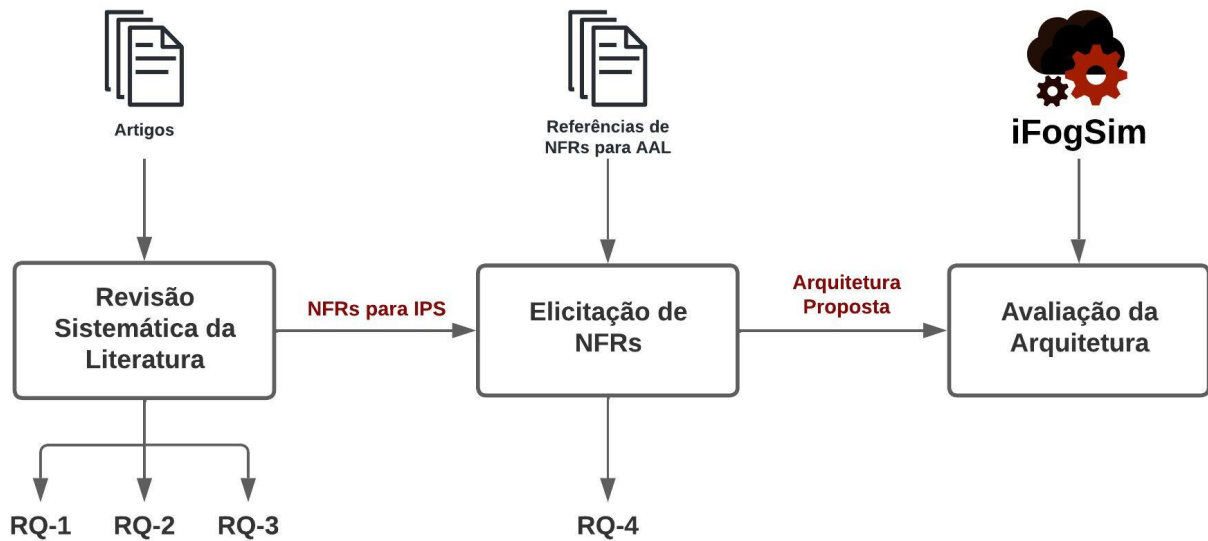
Nesse contexto, foi realizada uma consolidação de estilos arquiteturais e propostas existentes na literatura, bem como um processo de análise e reuso de RNFs para tais sistemas. Para essa finalidade, foram definidas as seguintes questões de pesquisa (RQs):

- a) **RQ1:** Quais são as soluções implementadas em arquiteturas de software para sistemas IPS baseados em IoT? - A **RQ1** visa investigar as principais características arquiteturais, tecnologias, estilos e padrões usados na definição da arquitetura de um sistema IPS baseado em IoT.
- b) **RQ2:** Quais são os requisitos não funcionais incorporados às arquiteturas de software definidas para sistemas IPS baseados em IoT? - A **RQ2** visa entender como os principais RNFs dos sistemas IPS baseados em IoT são atendidos nas arquiteturas propostas. Assim, é possível avaliar essas arquiteturas em relação à sua completude.
- c) **RQ3:** Existem arquiteturas propostas para sistemas IPS baseados em IoT para AAL? - A **RQ3** visa investigar se há alguma arquitetura proposta especificamente para sistemas IPS baseados em IoT para AAL. Assim, espera-se que seja possível analisar quais decisões importantes de projeto são consideradas e como os RNFs são atendidos no domínio específico de AAL.
- d) **RQ4:** Quais são os principais requisitos não funcionais para sistemas AAL baseados em IoT? - A **RQ4** tem como objetivo entender quais são os principais RNFs levados em consideração nos projetos de sistemas IoT para AAL. Esses RNFs são baseados em evidências, usando resultados de pesquisas recentes e relevantes.

1.3 Metodologia Adotada

De forma geral, a metodologia proposta para o presente trabalho, buscando responder as questões de pesquisa supracitadas e alcançar o objetivo do estudo, constitui-se essencialmente de três etapas, como pode-se observar na Figura 1. De forma geral, na primeira etapa foi realizada uma RSL sobre definições e implementações de arquiteturas de software para sistemas IPS no contexto de uma infraestrutura IoT a fim de responder as questões de pesquisa RQ1, RQ2 e RQ3.

Figura 1 – Metodologia utilizada



Fonte: Elaborado pelo Autor.

Na segunda etapa, buscando responder a questão de pesquisa RQ4, foi realizada uma elicitação de requisitos não funcionais para aplicações IPS de auxílio a ambientes de vida assistida com base nos resultados apresentados em Ahmadi *et al.* (2019), Islam *et al.* (2015) e Barbosa *et al.* (2020). Por fim, na terceira etapa foi realizada a definição de uma arquitetura de software por meio dos resultados das etapas anteriores e realizada uma avaliação experimental usando simulações com o iFogSim¹ (GUPTA *et al.*, 2017).

1.4 Estrutura e Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma. O Capítulo 2 traz a fundamentação teórica sobre os principais conceitos abordados neste trabalho, a saber: (i) sistemas de posicionamento interno, (ii) tecnologias e técnicas para localização interna, (iii) arquitetura de

¹ Conjunto de ferramentas de simulação de sistemas de *Fog computing*, disponível em: <https://github.com/Cloudslab/iFogSim>

software para aplicações IoT, e (iv) ambiente de vida assistida. Os trabalhos relacionados a esta proposta são discutidos no Capítulo 3, agrupados em: (i) propostas de arquiteturas de software para aplicações AAL baseadas em IoT, e (ii) propostas em AAL com uso de informações contextuais de localização interna. No Capítulo 4 são apresentados os resultados encontrados nas etapas da RSL e elicitação de requisitos realizadas. O Capítulo 5 apresenta a arquitetura proposta, partindo das decisões de projeto, e o detalhamento da proposta por meio de modelos abstratos da arquitetura. Os resultados da avaliação de desempenho da arquitetura proposta, bem como a definição do projeto de experimento baseado em simulações, são apresentados no Capítulo 6. Por fim, no Capítulo 7 são apresentadas as considerações finais, além das principais contribuições e limitações do trabalho, bem como as perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos mais importantes relacionados ao contexto em que este trabalho está inserido. São discutidos conceitos relativos a: sistemas de posicionamento interno; tecnologias, técnicas e métodos para localização interna; arquiteturas de software para aplicações IoT; e ambientes de vida assistida.

2.1 Sistemas de Posicionamento Interno

Sistemas de posicionamento estão presentes em nossa rotina diária, direta ou indiretamente. Por exemplo, é difícil imaginar-se viajando a um local desconhecido sem o auxílio de um sistema desse tipo, como o Sistema de Posicionamento Global / *Global Positioning System* (GPS). Muitas aplicações móveis, por exemplo, nos fornecem informações e serviços personalizados de acordo com nossa localização, às vezes até mesmo de forma transparente. Contudo, a grande maioria dos sistemas de posicionamento mais amplamente utilizados são voltados para ambientes externos e não funcionam adequadamente em ambientes internos (PIRAS; CINA, 2010).

Dessa forma, podemos classificar inicialmente os sistemas de posicionamento em dois grupos: global e local. O primeiro é composto por sistemas bem consolidados e conhecidos como Sistemas Globais de Navegação por Satélite / *Global Navigation Satellite System* (GNSS), onde o GPS é classificado e apresenta-se como o sistema global mais utilizado no mundo. Já sistemas de posicionamento local, chamados de Sistemas de Posicionamento Interno, são voltados para ambientes internos, como casas, *shoppings*, universidades e prédios em geral. Tais sistemas tornam-se necessários para aplicações específicas, onde os sistemas GNSS não conseguem entregar o nível de acurácia exigida por tais aplicações.

Sistemas IPS são projetados e construídos com a finalidade de prover um nível de acurácia mais adequado aos ambientes internos e às necessidades específicas de seus usuários e aplicações, visto que o sistema GPS comumente apresenta erros de localização acima de 10 m (MERRY; BETTINGER, 2019). Além disso, dada a complexidade de um ambiente interno composto por paredes, obstáculos e características das estruturas prediais, a acurácia do sistema GPS decai nesses ambientes por diversos fatores, como ausência de linha de visada e efeito de multipercurso (NASCIMENTO *et al.*, 2018).

A complexidade relativa a ambientes internos, mencionada acima, é também um grande desafio para sistemas IPS. Dado isso, as pesquisas e iniciativas dentro desse tópico nos

últimos anos, se baseiam em analisar e aprimorar tecnologias, técnicas e métodos para alcançar maiores níveis de acurácia. Dessa forma, há uma ampla variedade de implementações possíveis para IPS, bem como há diversas classificações para tais sistemas, não havendo uma taxonomia amplamente aceita nem um consenso geral entre os pesquisadores da área. A classificação mais comum de sistemas de posicionamento interno refere-se às tecnologias utilizadas para estimar a localização interna (MENDOZA-SILVA *et al.*, 2019).

Vale destacar que para este trabalho foram adotados conceitos distintos para sistemas de posicionamento interno e localização interna. Esse último está relacionado ao processo de estimação da localização de um alvo (indivíduo ou objeto) em um ambiente interno, utilizado pelo sistema de posicionamento interno. Logo, desse conceito, pode-se assumir que uma ou mais técnicas de localização interna podem ser utilizadas em um mesmo sistema IPS, de forma alternada ou até mesmo de forma conjunta.

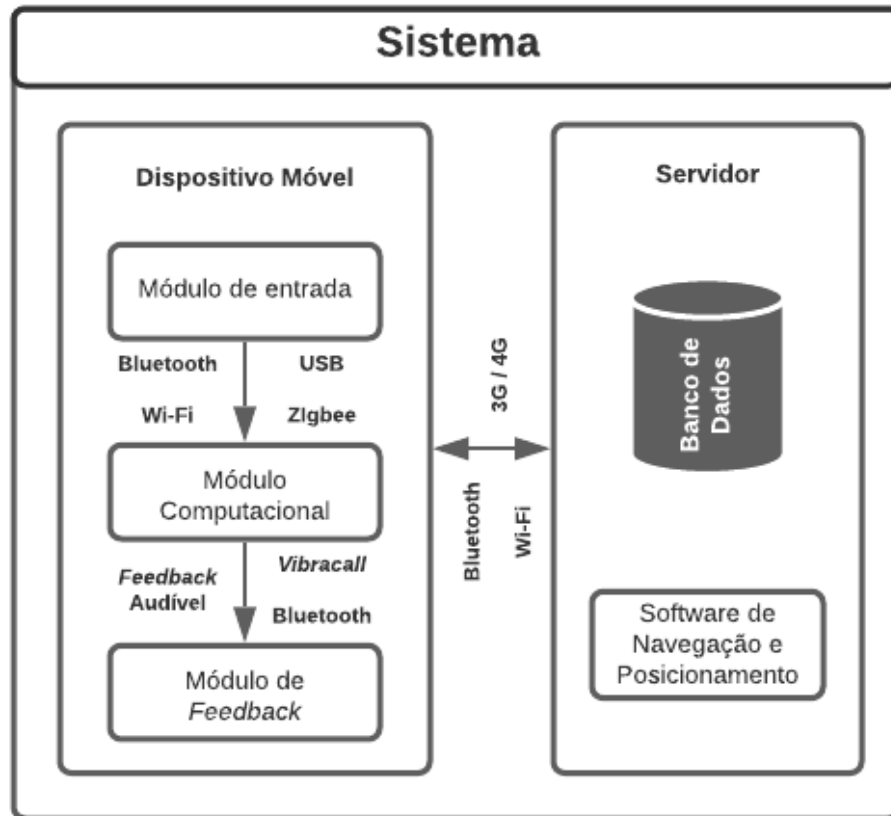
Dessa forma, para este trabalho foi utilizada uma classificação baseada no tipo de tecnologia para implementar um serviço de localização interna. Mais especificamente, utilizou-se uma convergência entre as classificações dadas em Gu *et al.* (2009) e Mendoza-Silva *et al.* (2019). Assim, para este trabalho a classificação considerada para sistemas IPS é a seguinte: baseados em visão computacional, baseados em radiofrequência (RF), baseados em ultrassom e sons audíveis, baseados em medições inerciais, baseados em infravermelho e baseados em luz visível.

Além disso, em relação à arquitetura computacional básica, sistemas IPS podem ser classificados como baseados em servidor ou autônomos (PASCACIO *et al.*, 2021). Essa classificação é interessante para este trabalho pois traz uma visão de divisão do sistema em componentes com responsabilidades distintas. No primeiro caso, existe um servidor remoto responsável por receber e tratar os dados brutos vindos dos dispositivos de sensoriamento e então realizar o processo de determinação da localização. No caso de um sistema autônomo no dispositivo final, todo o processo de adquirir dados dos sensores e realizar a estimação da localização fica a cargo do próprio dispositivo. A Figura 2 representa a arquitetura básica utilizada em IPS baseado em servidor.

2.2 Principais Tecnologias e Técnicas para Localização Interna

Como visto na seção 2.1, existe uma grande variedade de tecnologias utilizadas para implementar serviços de posicionamento interno. Detalhes das tecnologias de localização interna baseadas em RF que são abordadas neste trabalho, são apresentados na subseção 2.2.1. Outras

Figura 2 – Arquitetura básica de um sistema IPS baseado em servidor



Fonte: adaptado de Plikynas *et al.* (2020).

tecnologias são abordadas de forma breve nas subseções 2.2.2 e 2.2.3.

2.2.1 Localização Baseada em RF

Abordagens de localização baseadas em radiofrequência são amplamente utilizadas em sistemas IPS visto sua boa relação entre área de cobertura e custo de hardware empregado (LIU; JAIN, 2014). Como exemplos de tecnologias sem fio muito aplicadas a sistemas de posicionamento interno, pode-se citar: Wi-Fi, Zigbee, Identificação por Radiofrequência / *Radio Frequency Identification* (RFID), *Near Field Communication* (NFC) e *Bluetooth Low Energy* (BLE). Isso deve-se ao fato de que tais tecnologias possuem ondas eletromagnéticas capazes de penetrar materiais como paredes e corpos humanos, além de apresentar, em geral, melhor acurácia que outras tecnologias não baseadas em computação visual, como localização baseada em sensores infravermelhos e ultrassônicos (OBEIDAT *et al.*, 2021).

Redes sem fio baseadas em Wi-Fi e BLE são bastante utilizadas e discutidas na literatura. Isso deve-se às características que essas tecnologias apresentam, favoráveis a sistemas de localização interna em RSSFs. O termo Wi-Fi é usado para referir-se a redes que operam com as especificações IEEE 802.11 em faixas de frequência de 2,4 GHz ou 5 GHz. No caso do Wi-Fi,

um sistema IPS pode se beneficiar do baixo custo de implantação devido à infraestrutura de rede já existente em diversos ambientes internos, utilizando essa tecnologia (LASHKARI et al., 2010). Trilateração de sinais e *fingerprinting* são as técnicas mais comumente empregadas para realizar localização interna com as tecnologias Wi-Fi e BLE, dada a facilidade e praticidade em implementá-las. Contudo, a complexidade do ambiente diminui o desempenho de tais técnicas devido a diversos fatores, como efeito multipercurso e obstáculos que bloqueiam o sinal, que afetam o sinal Wi-Fi propagado (KHALAJMEHRABADI et al., 2017).

Em relação ao BLE, a característica principal que o torna uma potencial tecnologia de rede sem fio a ser utilizada em sistemas IPS é o baixo consumo de energia. O BLE tem foco na alta eficiência energética, sendo especificado a partir da versão 4.0 da tecnologia *Bluetooth* (ZAFARI et al., 2016). Algumas pesquisas mostram que a tecnologia BLE alcança maiores níveis de acurácia comparada ao Wi-Fi e se apresenta como a tecnologia mais adequada atualmente para uma ampla variedade de aplicações IPS. Isso geralmente ocorre devido a possibilidade de uma maior densidade de pontos de referência dispostos no ambiente, baixo custo dos sensores BLE e baixo consumo de energia (MENDOZA-SILVA et al., 2019).

Outras tecnologias como RFID, NFC e Zigbee são menos comuns porém também são utilizadas para aplicações IPS apresentando performances, requisitos e desafios distintos. Em comum, pode-se citar os desafios relativos à infraestrutura necessária e à pouca disponibilidade de dispositivos móveis ou vestíveis que possuam serviços de comunicação utilizando tais tecnologias (MENDOZA-SILVA et al., 2019).

Entre as principais técnicas de localização baseadas em RF, as mais utilizadas tomam como informação mais importante o Indicador de Intensidade do Sinal Recebido / *Received Signal Strength Indication* (RSSI) de um nó emissor. Seja para estimar a distância entre dois nós da rede ou para construir um mapa baseado nos padrões de potências recebidas de pontos de referência no ambiente. Assim, duas técnicas amplamente utilizadas são a trilateração de sinais e a técnica de *fingerprinting* (LIU; JAIN, 2014).

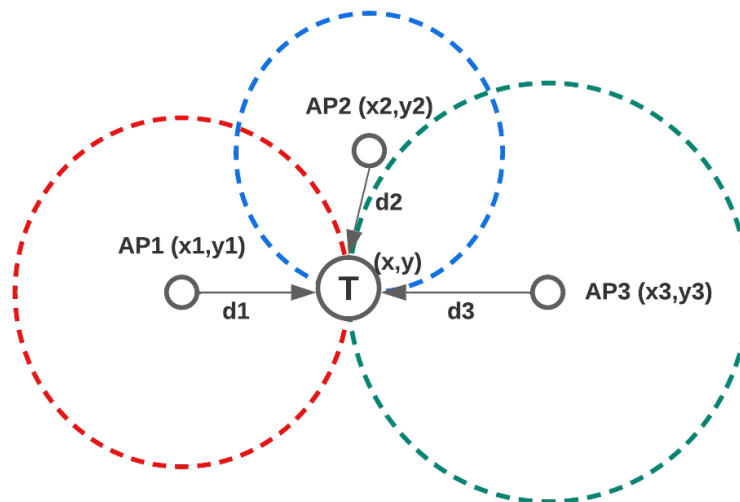
O RSSI é uma variável bastante utilizada como parâmetro para calcular a distância entre dois sensores sem fio. A partir desse indicador, que é facilmente aferido e disponibilizado pelos dispositivos atuais, é possível medir a distância entre dois nós através do princípio de atenuação do sinal e sua relação com a distância entre emissor e receptor de RF (RAPPAPORT, 2009). A estimação de distância baseada no indicador RSSI, utilizando o modelo de propagação log-distância é dada por:

$$RSSI = P_r(d) = P_r(d_0) - 10 \cdot \beta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma. \quad (2.1)$$

Na Equação 2.1, $P_r(d)$ é a intensidade do sinal recebido em uma distância d , $P_r(d_0)$ é a intensidade do sinal recebido em uma distância d_0 , geralmente 1 metro, β representa o coeficiente de perda de percurso, que pode ter seu valor variando de 2 a 6 dependendo do ambiente, e X_σ é uma variável aleatória com distribuição normal (gaussiana) de média zero e desvio padrão σ que representa o efeito aleatório de sombreamento.

Como traz Moreira *et al.* (2017), para que a técnica de lateração possa ser utilizada é necessário que haja pelo menos três pontos de referência. Dessa forma temos uma trilateração (Figura 3), onde as distâncias entre as posições dos pontos de referência e a posição do nó desconhecido (T) podem ser representadas como os raios dos círculos com centro na posição de cada ponto de referência (PR).

Figura 3 – Técnica de Trilateração de Sinais



Fonte: adaptado de Moreira *et al.* (2017).

As coordenadas de cada PR são previamente conhecidas e mapeadas no sistema. Assumindo as coordenadas dos PRs como $AP_1(x_1, y_1)$, $AP_2(x_2, y_2)$ e $AP_3(x_3, y_3)$, enquanto as coordenadas do dispositivo de posição desconhecida representadas por $T(x, y)$, e utilizando o Teorema de Pitágoras, pode-se descrever as equações que relacionam as distâncias e as coordenadas das posições dos dispositivos. Como as distâncias podem ser calculadas pelo valor do RSSI recebido de cada PR, as coordenadas do dispositivo de posição desconhecida podem ser calculadas pela Equação 2.2.

$$\begin{aligned}
 d_1^2 &= (x_1 - x)^2 + (y_1 - y)^2 \\
 d_2^2 &= (x_2 - x)^2 + (y_2 - y)^2 \\
 d_3^2 &= (x_3 - x)^2 + (y_3 - y)^2
 \end{aligned}
 \tag{2.2}$$

Por sua vez, a técnica de *fingerprinting* vem recebendo muita visibilidade e sendo aplicada em sistemas de posicionamento interno. O método consiste em um mapeamento, previamente construído, da força do sinal recebido de cada PR, em cada região dentro do ambiente, sem que seja necessário o conhecimento da localização dos PRs da rede (PU; YOU, 2018).

Sistemas de localização que utilizam *fingerprinting* geralmente apresentam duas fases operacionais: mapeamento (treino) e localização online (teste). Na primeira fase o ambiente é dividido em diversos setores, então o mapeamento de cada uma dessas regiões é realizado com base na medida do RSSI de pontos de referência conhecidos. A etapa de localização online, ou testes, consiste em determinar a localização de um dispositivo alvo, dentro do ambiente mapeado, comparando as medidas RSSI coletadas por esse dispositivo e os dados que foram mapeados previamente.

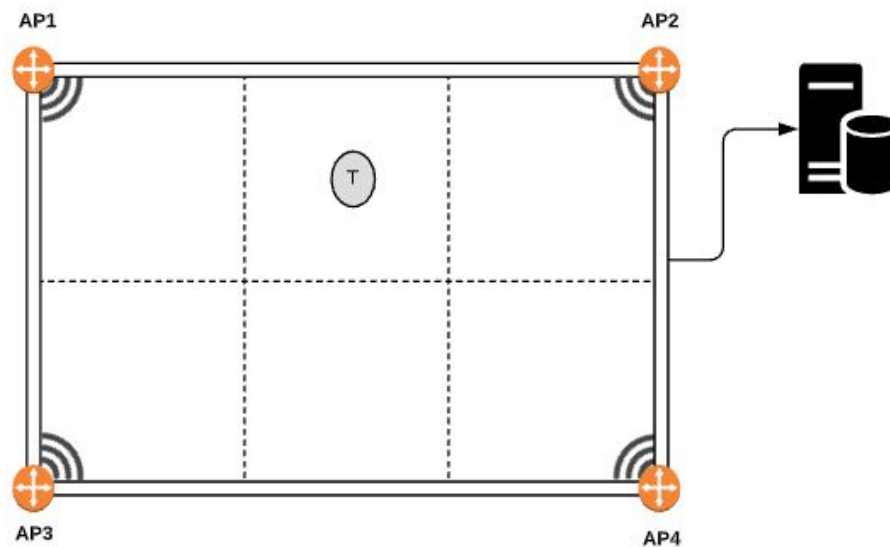
A Figura 4 representa o processo de localização interna com a técnica de *fingerprinting*. Os PRs são identificados pelos pontos AP_1 , AP_2 , AP_3 e AP_4 , já o dispositivo alvo, que se deseja determinar a localização, é identificado pelo ponto T. Na fase de mapeamento, algum dispositivo é utilizado para coletar as medições RSSI recebidas de cada PR em cada região e armazená-las em uma base de dados. Na fase de localização, o dispositivo alvo realiza um escaneamento procurando o sinal dos pontos de referência e realiza a coleta das medidas de RSSI. Com base nesses dados, o dispositivo se comunica com um servidor de contexto que realiza a comparação dos dados atuais com os dados armazenados e retorna sua posição estimada.

Com o intuito de melhorar a precisão de localização, muitos estudos adotam técnicas de Aprendizagem de Máquina, principalmente algoritmos classificadores como o K-NN para determinar com base nos dados de treino, previamente mapeados, a região em que o dispositivo alvo está localizado (MENDOZA-SILVA *et al.*, 2019).

2.2.2 Localização Baseada em Medições Inerciais

Os *smartphones* possuem uma grande variedade de sensores, entre eles os sensores de medição inercial. Unidades de Medição Inercial / *Inertial Measurement Units* (IMU) são

Figura 4 – Exemplo de aplicação do *fingerprinting*



Fonte: Elaborado pelo Autor.

compostas geralmente por acelerômetro, giroscópio e bússola. Utilizando essas unidades, pode-se calcular velocidade de deslocamento e orientação, por exemplo. O acelerômetro retorna a aceleração total nos três eixos (a_x , a_y e a_z), resultante da combinação da aceleração da gravidade (g_x , g_y e g_z) e da força exercida pelo utilizador (SILVA *et al.*, 2021). Já o giroscópio e a bússola permitem medir a orientação do dispositivo em relação ao polo magnético da Terra (DAVIDSON; PICHÉ, 2017).

A principal técnica utilizada para navegação baseada em IMUs é a *Pedestrian Dead-Reckoning* (PDR). Tipicamente essa técnica consiste em três etapas: detecção de passos, estimativa de tamanho do passo e estimativa de direção. Isso é feito por meio de integrações das medições de aceleração e orientação fornecidas pela IMU. Contudo, de forma geral essa técnica é bastante suscetível à perda de acurácia por causa da acumulação de erros que aumentam rapidamente devido às integrações sucessivas e pela própria característica instável dos sensores (SILVA *et al.*, 2021; MENDOZA-SILVA *et al.*, 2019).

Dessa maneira, existem várias propostas para melhorar a acurácia de localização em propostas baseadas em PDR. Dentre elas, pode-se citar a utilização do Filtro Kalman (FK), Filtro de Partículas (FP), *Zero-velocity UPdaTes* (ZUPT), entre outras (DAVIDSON; PICHÉ, 2017). Conseguindo aliar redução de erros e um baixo custo de implantação, além de não haver necessidade de infraestrutura no local, essa técnica é bem aceita e bastante empregada em sistemas de navegação interna (MENDOZA-SILVA *et al.*, 2019).

2.2.3 Outras Tecnologias para Localização Interna

De forma breve, pode-se citar, como alternativas às tecnologias apresentadas anteriormente, técnicas de localização baseadas em visão computacional, infravermelho, ultrassom e sons audíveis. Técnicas baseadas em infravermelho e ultrassom fornecem uma localização por proximidade e não são capazes, isoladamente, de retornar a posição precisa de um alvo. Técnicas baseadas em sons audíveis podem ser utilizadas para localização, inclusive baseada em *fingerprinting*, porém não são adequados para ambientes de maior escala, pois estão suscetíveis a diversos ruídos variáveis no ambiente (NAVYA *et al.*, 2020; MENDOZA-SILVA *et al.*, 2019).

Tratando-se de técnicas baseadas em visão computacional, vários estudos demonstram altos níveis de acurácia em diversos cenários. Contudo, uma desvantagem dessa abordagem é o alto custo computacional exigido para realizar o processamento de imagens capturadas. Alguns estudos indicam que futuramente as exigências de recursos por tais técnicas sejam facilmente supridas e elas passem a ser mais amplamente utilizadas (MENDOZA-SILVA *et al.*, 2019; CADENA *et al.*, 2016).

2.3 Arquitetura de Software para Aplicações IoT

A internet das coisas se tornou um importante tópico de pesquisa nas últimas duas décadas, recebendo bastante atenção tanto da comunidade acadêmica e científica, como da indústria (KRISHNAN; VASUDEVAN, 2019; TAIVALSAARI; MIKKONEN, 2018). A principal motivação é permitir que diversos objetos (coisas), inicialmente comuns, possam interagir com o ambiente onde estão inseridos e tenham conectividade, permitindo novos serviços e tecnologias colaborativas. Tais serviços podem ser utilizados para nos auxiliar em nossas tarefas diárias, em nossas tomadas de decisões ou até mesmo em nosso trabalho (WASHIZAKI *et al.*, 2020).

A principal infraestrutura utilizada para conectar e prover serviços de comunicação a esses objetos é a internet. Dessa maneira, a quantidade de coisas conectadas e aplicações IoT desenvolvidas cresce de forma exponencial, trazendo junto vários desafios de projeto, implementação e manutenção de tais soluções (ALY *et al.*, 2019). Com tantas possibilidades de aplicações, tecnologias e dispositivos, soluções IoT se apresentam extremamente heterogêneas e muitas vezes complexas. Dessa maneira, não há um consenso sobre padrões, principalmente em relação à arquitetura de software (GIRI *et al.*, 2017; NIKOUI *et al.*, 2021; GUPTA; QUAMARA, 2020).

Há dois tipos básicos de arquitetura (Figura 5) definidos por pesquisadores para aplicações IoT: arquitetura de três camadas e arquitetura de cinco camadas (LIN *et al.*, 2017; KHAN *et al.*, 2012; KRISHNAN; VASUDEVAN, 2019). A arquitetura de software que constitui a base da maioria dos estilos arquiteturais em IoT é definida em três camadas. A camada inicial, chamada de camada de percepção (ou camada física), é responsável por obter e reunir informações sensoriais do mundo real. A camada de rede, camada intermediária, é responsável por conectar as outras duas camadas adjacentes usando protocolos de comunicação e segurança para troca de dados entre as demais camadas. A análise, tratamento e visualização das informações reunidas das várias fontes são realizados na camada de aplicação (LIN *et al.*, 2017; KRISHNAN; VASUDEVAN, 2019).

Figura 5 – Arquiteturas básicas para IoT



Fonte: Elaborado pelo Autor.

Por sua vez, a arquitetura de cinco camadas define ainda a camada de *middleware*, logo acima da camada de rede, e a camada de negócios localizada logo acima da camada de aplicação. A camada de *middleware* é responsável por reunir, analisar e processar uma enorme quantidade de dados obtidos pelos objetos inteligentes, tratando das heterogeneidades presentes nas camadas física e de rede e intermediando a comunicação com a camada de aplicação. Já a camada de negócio trata dos modelos de negócio específicos do domínio da aplicação e gerencia as aplicações do sistema IoT bem como seus serviços e privacidade dos usuários (KRISHNAN; VASUDEVAN, 2019; KHAN *et al.*, 2012).

Uma outra maneira de classificar opções de estilos e padrões arquiteturais em sistemas IoT é apresentado em Taivalsaari e Mikkonen (2018), onde adota-se uma visão de mais

alto nível levando em consideração a complexidade de cada aplicação. Dessa maneira, elenca-se sete categorias de arquiteturas de software, partindo de um contexto onde não há a necessidade da presença de um sistema operacional até arquiteturas totalmente baseadas na utilização de containers de sistemas operacionais.

2.3.1 *Novos Paradigmas de Computação em Nuvem*

Novos paradigmas IoT surgem com a sua interligação à computação em nuvem. Serviços de computação em nuvem oferecem um modelo de compartilhamento online de recursos computacionais sob demanda, possibilitando suporte para que as aplicações tornem-se escaláveis, interoperáveis e seguras. Isso possibilitou o surgimento de novos paradigmas na internet das coisas, auxiliando a mitigar desafios presentes em tal domínio, como a falta de padrões bem definidos, segurança e confiabilidade (AL-KHAFAJIY *et al.*, 2019; MUNISWAMAIAH *et al.*, 2021).

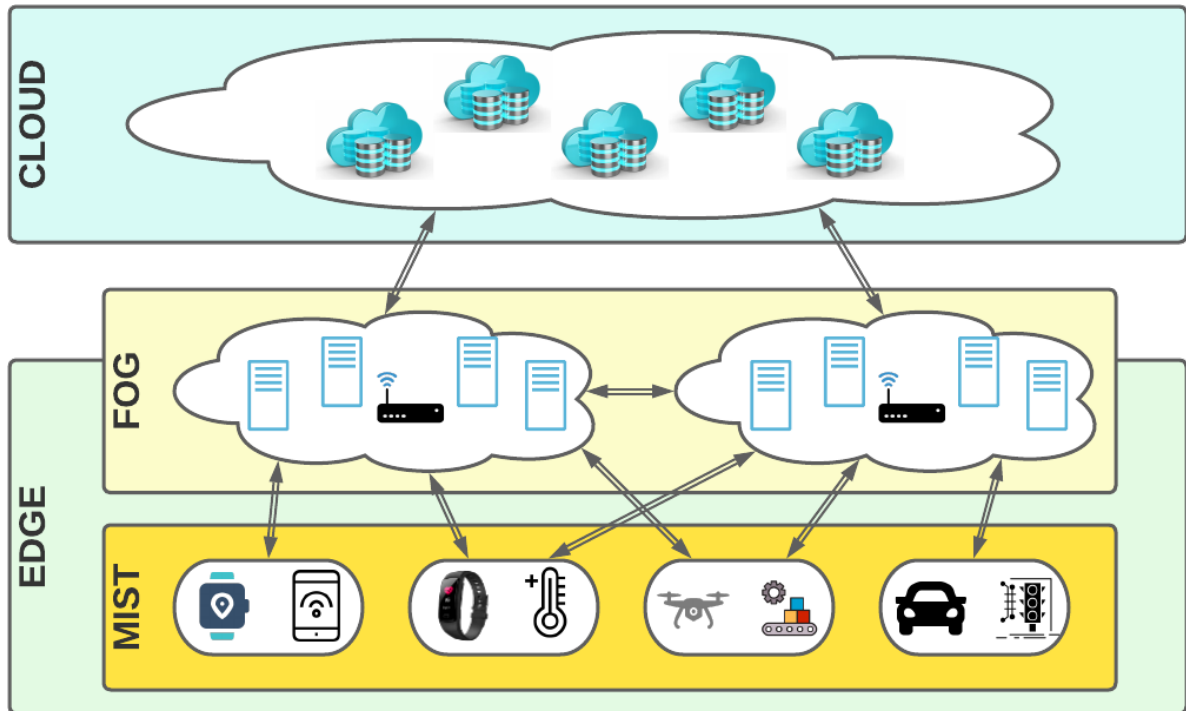
Assim, diversas abordagens em relação à integração de infraestrutura IoT com serviço em nuvem vêm sendo discutidas. A tendência é trazer recursos computacionais, antes disponíveis somente na camada de nuvem, para a borda da rede. Com isso, espera-se uma diminuição no tempo de resposta das aplicações, maior segurança e interoperabilidade (MUNISWAMAIAH *et al.*, 2021). Entre os principais paradigmas emergentes em IoT baseada em nuvem, pode-se citar: *Fog computing*, *Edge computing* e *Mist computing*.

Uma integração desses paradigmas pode ser utilizada a fim de alocar recursos e serviços de processamento para a camada de computação adequada em tempo real, distribuindo as tarefas entre todos os dispositivos disponíveis na rede. Com isso, pode-se alcançar um melhor aproveitamento dos recursos computacionais ao longo da comunicação fim-a-fim, como apresentado na Figura 6, e obter vantagens do paradigma de *fog computing*, como diminuição na latência e no consumo de rede, bem como consciência de contexto e melhorias relacionadas à segurança e privacidade.

Em uma abordagem *top-down*, a camada de *Fog Computing* oferece serviços de rede, armazenamento e processamento, localizados entre os dispositivos finais e a camada de nuvem tradicional. O objetivo é trazer processamento, armazenamento e serviços de rede para a borda da infraestrutura, possibilitando também uma distribuição geográfica de tais serviços (BONOMI *et al.*, 2012; ESCOBAR *et al.*, 2022).

O termo *Edge Computing* é geralmente empregado para se referir ao processamento

Figura 6 – Arquitetura básica de paradigmas emergentes em IoT baseada em Cloud



Fonte: adaptado de Escobar *et al.* (2022).

que ocorre entre os dispositivos finais dos usuários e a nuvem, oferecendo serviços e recursos na borda da rede. Por isso, o paradigma de *fog computing* pode ser incluído em um arranjo de *edge computing*, ou, ao contrário, os dispositivos *edge* podem se localizar na fronteira de um arranjo de *fog computing*. Ainda, é possível que esses dois termos sejam considerados paradigmas independentes (ESCOBAR *et al.*, 2022; BONOMI *et al.*, 2012; ROSENDO *et al.*, 2022).

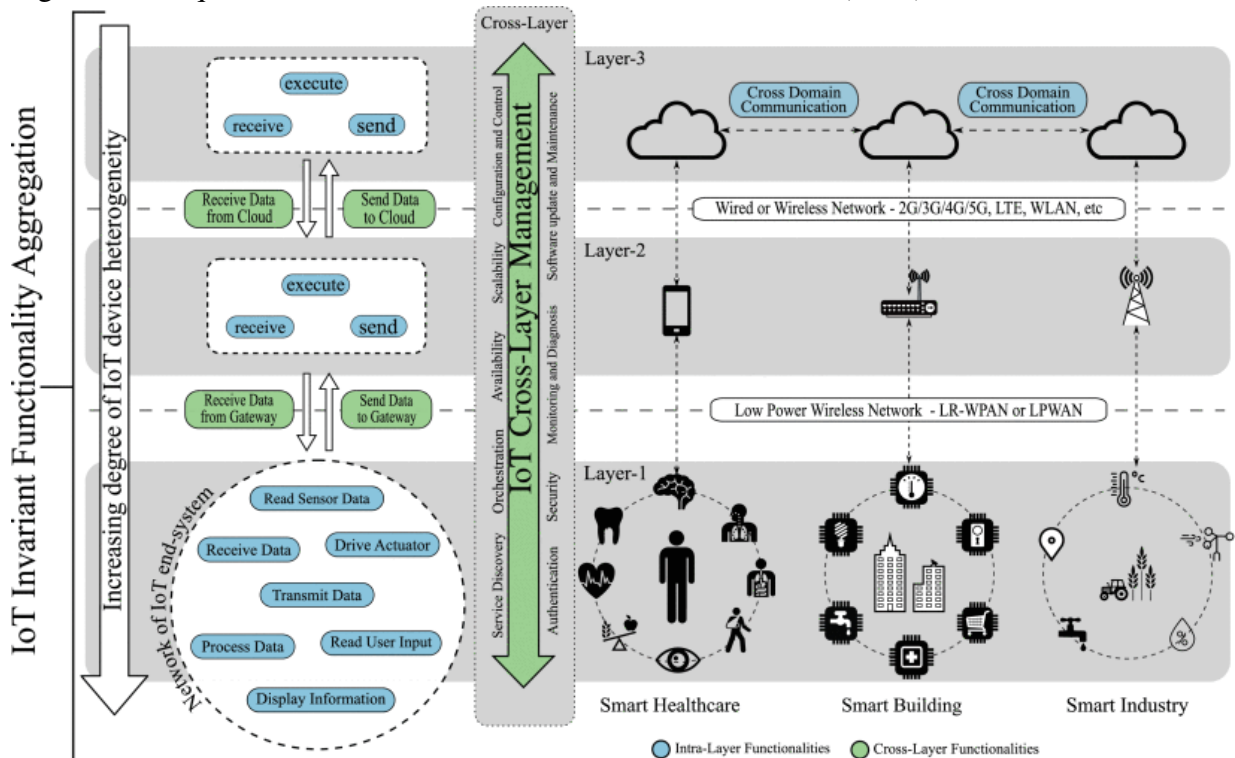
Por fim, a camada de *Mist Computing* atua diretamente com os dispositivos finais, algumas vezes podendo ser incorporada a eles, provendo algum processamento em tempo real para respostas instantâneas (AL-KHAFAJIY *et al.*, 2019; SIKARWAR *et al.*, 2020). Todo esse arranjo e as possibilidades de integração de tais paradigmas e suas tecnologias, contribuem com o desenvolvimento e aprimoramento do conceito de computação contínua (ROSENDO *et al.*, 2022).

Nesse contexto, levando em consideração a grande diversidade de aplicações IoT com características totalmente distintas e específicas, torna-se um grande desafio a definição de um padrão arquitetural ou mesmo de programação que atenda as especificidades de cada domínio de aplicação (PAWAR *et al.*, 2021). Assim, a definição de uma arquitetura de software apropriada a um determinado domínio de uma solução IoT não é uma tarefa trivial e depende de várias decisões de projeto e diversos fatores como custo, escalabilidade, eficiência energética,

latência de comunicação, entre outros (TAIVALSAARI; MIKKONEN, 2018).

Em (PAWAR *et al.*, 2021), os autores realizam um estudo sistemático a fim de entender as características básicas e invariantes relativas à arquitetura e programação de aplicações IoT. Com base nisso foi proposto um padrão arquitetural base, generalista, para qualquer aplicação IoT, levando em consideração as características invariantes de tais aplicações. Como resultado, foi apresentado um padrão de arquitetura de software baseado em quatro camadas (Figura 7): camada física, camada de borda, camada de nuvem (*cloud*) e camada de gerenciamento. A arquitetura define três camadas principais, como apresentado na Seção 2.3, adicionando uma camada *cross-layer* com funções de gerenciamento das demais camadas.

Figura 7 – Arquitetura de Sistemas IoT definida em Pawar *et al.* (2021)



Fonte: Pawar *et al.* (2021).

2.4 Ambiente de Vida Assistida

O termo Ambiente de Vida Assistida / *Ambient Assisted Living* (AAL) é derivado do conceito de ambientes inteligentes, surgindo inicialmente a partir de esforços em propor tecnologias, serviços e soluções de assistência a idosos no que diz respeito à sua condição de saúde e bem estar. Outro importante aspecto relativo ao conceito de AAL é que esse conjunto de tecnologias e soluções propostas são, de forma geral, voltadas a prover assistência ao usuário em

sua própria residência (RASHIDI; MIHAILIDIS, 2013).

A necessidade de prover assistência a idosos em seu ambiente domiciliar, o que motivou o surgimento do conceito de AAL, se justifica pelo envelhecimento da população a nível mundial, devido aos avanços recentes na medicina. Um estudo do Departamento de Assuntos Econômicos e Sociais das Nações Unidas, projeta que em 2050 mais de 20 % da população mundial terá 60 anos ou mais (MCNICOLL, 2002).

Logo, é esperado que esse aumento no envelhecimento da população mundial resulte em enormes desafios para a sociedade e para o sistema de saúde, como: aumento de doenças e condições crônicas, aumento nos custos de cuidados à saúde, falta de cuidadores, dependência e impactos na economia. Assim, o objetivo principal de aplicações de AAL é aumentar a qualidade de vida, segurança e bem estar pessoal, beneficiando não somente o indivíduo, mas também a economia e a sociedade (RASHIDI; MIHAILIDIS, 2013; MCNICOLL, 2002; DOHR *et al.*, 2010).

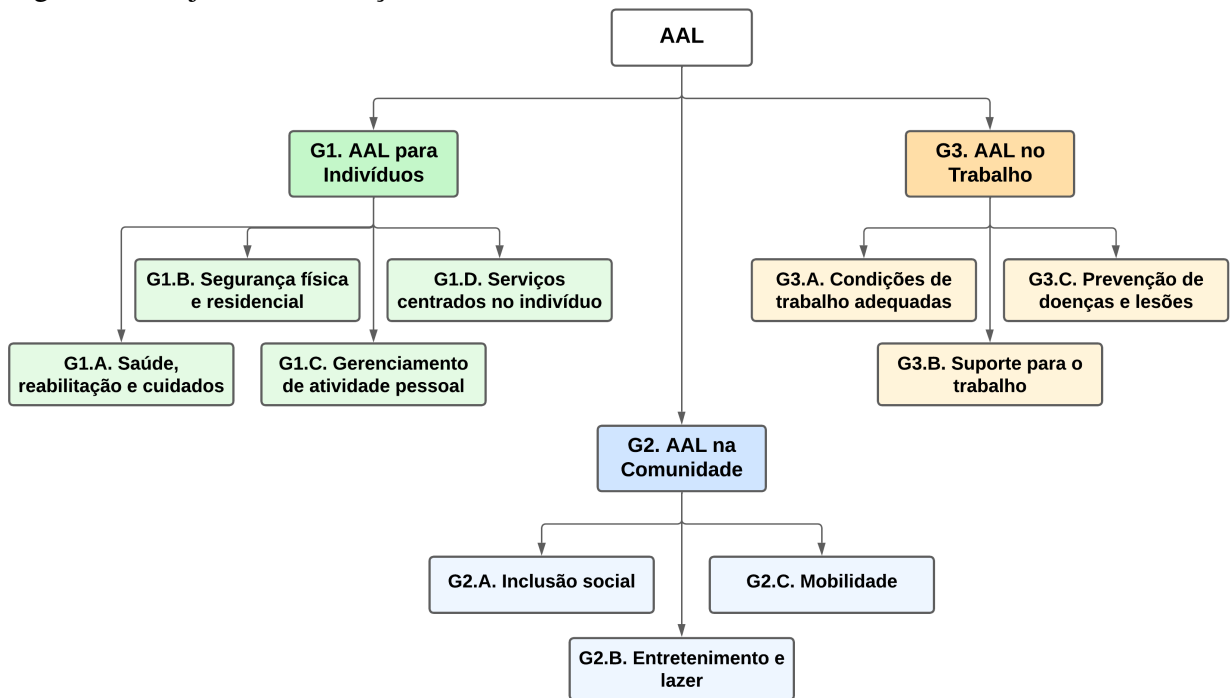
Embora os esforços em propor ambientes de vida assistida tenham sido inicialmente direcionados a pessoas idosas, de forma mais ampla, soluções de AAL são projetadas para melhorar a autonomia, conforto, segurança, saúde e bem estar de todos os indivíduos em todos os estágios de sua vida (BROEK *et al.*, 2010). Logo, além de idosos, pessoas com alguma deficiência ou condição crônica podem se beneficiar de tais soluções. Também vale ressaltar que além do ambiente primário do indivíduo, sua residência, um sistema de AAL pode auxiliá-lo em outros tipos de ambiente, como hospitais, casas de apoio, áreas de lazer e até mesmo em seu ambiente de trabalho (BROEK *et al.*, 2010; GARCÉS *et al.*, 2020). A Figura 8 apresenta a classificação de sistemas AAL quanto ao seu objetivo.

Resumidamente, sistemas AAL podem ser classificados, de acordo com sua finalidade, em três principais objetivos: (G1) AAL para indivíduos, geralmente em sua própria residência; (G2) AAL na comunidade; (G3) AAL como apoio ao trabalho. De forma mais detalhada, cada objetivo pode ter subclassificações, como AAL voltado à cuidados em saúde e reabilitação de indivíduos (G1.A).

Em relação às tecnologias de suporte à construção de sistemas AAL, podemos citar a utilização de ferramentas e tecnologias associadas a *smart homes*, ambientes inteligentes, rede de sensores sem fio (RSSF), rede de sensores vestíveis, robôs assistivos e tecnologias voltadas a *eHealth* (MONEKOSSO *et al.*, 2015; GARCÉS *et al.*, 2020; RASHIDI; MIHAILIDIS, 2013).

Esse conjunto de tecnologias, de forma integrada, podem ser utilizadas para construir

Figura 8 – Objetivos de soluções de AAL



Fonte: adaptado de Broek *et al.* (2010).

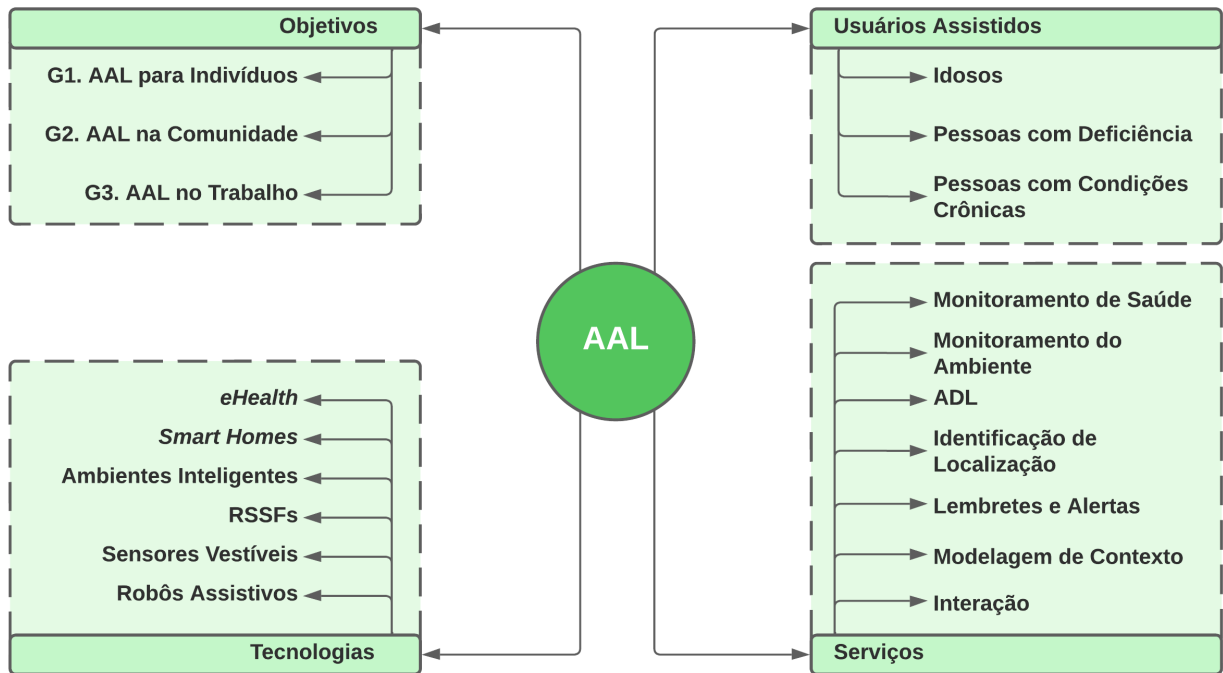
soluções de AAL a fim de fornecer um amplo conjunto de serviços, como: monitoramento remoto de estado de saúde, monitoramento do ambiente, reconhecimento de atividades de vida diária / *Activities of Daily Living* (ADL), análise de comportamento e bem estar, modelagem de contexto, identificação de localização, lembretes e alertas, suporte a interação social ou com tecnologias, entre outros (CALVARESI *et al.*, 2017; RODRIGUES *et al.*, 2018; DUARTE *et al.*, 2018). A Figura 9 apresenta de forma resumida as características relativas a uma solução de AAL, discutidas nesta seção, quanto aos seus objetivos, público alvo, tecnologias e serviços oferecidos.

2.4.1 Sistemas IoT para AAL

Como apresentado na Seção 2.4, um dos principais objetivos de um sistema AAL é possibilitar o monitoramento de saúde e bem estar do indivíduo, utilizando-se de tecnologias e serviços na área de *eHealth*. Dentro desse contexto de assistência médica mediada ou suportada por Tecnologias da Informação e Comunicação (TIC), surge um novo paradigma conhecido como *Healthcare 4.0*, tendo como pilares as seguintes tecnologias: IoT, *Cloud/Fog Computing*, e *Big Data* (ACETO *et al.*, 2020).

Nesse contexto, ambientes inteligentes podem ser potencializados com a utilização da internet das coisas e tecnologias associados a ela, proporcionando-lhes serviços ubíquos e

Figura 9 – Características de uma solução de AAL



Fonte: Elaborado pelo Autor.

pervasivos. Dessa forma, sistemas de assistência à vida, que buscam dar suporte a pessoas com necessidades específicas, como idosos e pessoas com deficiência, podem se beneficiar com os serviços oferecidos por um ambiente inteligente. Nesse caso, o ambiente torna-se um ambiente de vida assistida (KLEINBERGER *et al.*, 2007; DOHR *et al.*, 2010).

A existência de ambientes de vida assistida não está condicionada à aplicação de tecnologias baseadas em IoT, porém o suporte e serviços dados aos usuários podem ser potencializados pelo emprego de ambientes inteligentes que utilizem essa infraestrutura. Além disso, com a utilização de tecnologias de sensoriamento, monitoramento e comunicação, é possível fornecer cuidados de saúde personalizados e de forma mais rápida e precisa. Essa aplicação de IoT dentro das áreas de *eHealth* e AAL é conhecida como *Internet of Health Things* (IoHT) e é vista como um paradigma que pode gerar um grande impacto econômico no mundo, causado por redução de custos, monitoramento preventivo de saúde e melhorias na qualidade de vida de pessoas com doenças crônicas (RODRIGUES *et al.*, 2018).

Contudo, alguns desafios ainda se fazem presentes na tentativa de projetar e implementar um sistema IoT para AAL. Muitas vezes, esses desafios são advindos das próprias características da área de IoT de forma geral. Por exemplo, é difícil estabelecer uma arquitetura padrão para alguma solução baseada em IoT, o que implica em problemas de interoperabilidade e integração de tecnologias, características importantes em uma aplicação de *eHealth* (ISLAM

et al., 2015). Além disso, questões como restrições de energia, escalabilidade, segurança, privacidade, desempenho de comunicação e uso de rede são desafios inerentes a aplicações IoT (ACETO *et al.*, 2020).

O paradigma de *Fog Computing*, já discutido na Seção 2.3.1, surge então como uma tecnologia que pode permitir a mitigação desses desafios (PENG *et al.*, 2022). Tal abordagem traz vantagens, comparada a uma infraestrutura de integração *Cloud-IoT* simples, relativas a latência de comunicação, uso de rede, privacidade e segurança, consciência de contexto e disponibilidade de conexão em ambientes hostis/remotos (PULIAFITO *et al.*, 2019).

Através de uma camada *fog* é possível se beneficiar de processamento de dados locais, adaptabilidade a diferentes contextos de uso, armazenamento local, políticas de segurança aplicadas localmente, interoperabilidade, suporte a mobilidade, eficiência energética e redução na latência de comunicação e uso da rede (RAHMANI *et al.*, 2018; PULIAFITO *et al.*, 2019). Isso é possível devido a natureza distribuída do paradigma de *fog computing*, possibilitando o fornecimento de serviços e recursos nos mais diversos níveis da rede, tornando contínuo o fluxo de comunicação e processamento ao longo do caminho *Thing-to-Cloud*.

O cenário de transformação e integração digital de serviços nas mais diversas áreas, dentro de um conceito de cidades conectadas e inteligentes, gera a necessidade de repensar a forma de prestação de cuidados e assistência à saúde (MUHAMMED *et al.*, 2018). Nesse sentido, o paradigma de *fog computing* tem o potencial de possibilitar uma maior escalabilidade para aplicações no domínio de *Healthcare 4.0*, bem como permitir uma melhor integração com diversos outros serviços em um contexto de *smart cities* (MUHAMMED *et al.*, 2018; ARFAT *et al.*, 2017; KUMARI *et al.*, 2018).

2.4.2 Localização Interna para AAL

Uma solução de AAL utiliza tecnologias para prover um ambiente inteligente com consciência de contexto e de forma a se adaptar às necessidades do usuário assistido. Logo, se mostra muito importante o uso de informações contextuais, além de dados relativos ao monitoramento de saúde, bem estar e segurança do indivíduo (DUARTE *et al.*, 2018). Como informações contextuais, pode-se citar dados relativos ao monitoramento do ambiente, como temperatura e umidade, instante de tempo em que determinada medição foi realizada e dados contextuais referentes à localização do indivíduo, seja em ambiente interno ou externo (RASHIDI; MIHAILIDIS, 2013; BELMONTE-FERNÁNDEZ *et al.*, 2016).

No que se refere especificamente à utilização de informações contextuais de localização, sistemas de AAL podem se beneficiar de tal abordagem para prover serviços voltados a reconhecimento de atividades de vida diária (ADL) (BAE, 2014; FORTINO *et al.*, 2021), detecção de quedas (NOORUDDIN *et al.*, 2022), assistência quanto às necessidades do usuário (COSTA *et al.*, 2014) e monitoramento de localização de pessoas com mal de Alzheimer (SURENDRAN; ROHINIA, 2019), por exemplo. Localizar indivíduos e objetos dentro de um ambiente inteligente torna uma solução de AAL capaz de oferecer serviços de forma personalizada e proativa, pois permite identificar contextos situacionais dentro desse ambiente.

A identificação de residentes é outro tópico relacionado à localização interna, sendo, também, um serviço muito útil em um ambiente de vida assistida que conta com múltiplos usuários (RASHIDI; MIHAILIDIS, 2013). Assim, diversos serviços baseados em localização interna podem ser projetados em uma aplicação de ambiente de vida assistida. De forma geral, a acurácia necessária a um sistema de posicionamento interno (IPS) dentro de uma solução de AAL é variável, pois depende dos requisitos, público-alvo e finalidade da aplicação. No entanto, para muitas soluções, uma acurácia de localização a nível de cômodo se mostra adequada (MONEKOSSO *et al.*, 2015).

3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados e descritos de forma breve trabalhos correlatos, que se assemelham em algum aspecto ao presente trabalho. Os trabalhos relacionados foram agrupados em dois tópicos: (i) arquiteturas de software para aplicações AAL baseadas em IoT; e (ii) aplicações AAL com uso de informação contextual de localização interna.

3.1 Arquiteturas de software para aplicações AAL baseadas em IoT

Em Barbosa *et al.* (2020), os autores apresentam a definição de uma arquitetura de software para projetos e tecnologias IoT para a área de saúde. Além da definição da arquitetura, foram implementados os componentes de software na forma de recursos *open-source* (código aberto) e aplicados a projetos reais. Ao final, foi feita uma avaliação a partir de requisitos de qualidade, por parte de diferentes desenvolvedores que utilizaram e testaram os componentes em seus projetos, destacando a facilidade de integração e a utilização de padrões usando código aberto, facilitando assim o reuso.

No trabalho de Gulati e Kaur (2022) é proposto um sistema AAL baseado em IoT Social para idosos, chamado *FriendCare-AAL*. A arquitetura do sistema propõe uma abordagem esquemática para estabelecer parcerias entre dispositivos na rede e para possibilitar transferência de responsabilidade entre os dispositivos. O sistema provê assistência dentro do ambiente domiciliar do idoso e, ao identificar alguma situação de emergência, gera alertas automáticos para entidades interessadas (por exemplo, cuidadores ou membros da família). O sistema é avaliado experimentalmente por meio de simulações e apresenta uma acurácia de 83,3% na determinação de situações de emergência.

Hajvali *et al.* (2022) propõem uma arquitetura de software baseada em IoT para sistemas de *healthcare*. A proposta apresentada pelos autores se baseia na utilização de um modelo de serviço *cloud/fog*, com três camadas: camada de dispositivos (aplicações e sensores de *e-Health*), camada fog e camada de nuvem. A arquitetura é modelada a partir de uma abordagem formal, utilizando sistema de transformação de grafos, que também é utilizada para realizar verificações formais relativas aos requisitos de alcance, segurança e vivacidade. Além disso, é realizada uma avaliação arquitetural com uso do método de análise de *tradeoff* de arquitetura, demonstrando atendimento aos requisitos de confiabilidade, disponibilidade e desempenho.

Um sistema de monitoramento de pacientes, baseado em *fog computing*, para ambi-

ente de vida assistida, chamado *FAAL*, é apresentado em Vora *et al.* (2017). O sistema utiliza registros de dados de movimento dos pacientes, coletados através de nós sensores por meio de uma Rede de Área Corporal / *Body Area Network* (BAN) e transmitidos para um servidor de borda. A solução utiliza os algoritmos K-means e C4.5 para detectar anomalias com base em dados de movimento de pacientes com doenças neurológicas crônicas. O sistema é avaliado com relação à latência, à sobrecarga de dados, ao consumo de energia e à escalabilidade, demonstrando um desempenho superior comparado a um ambiente baseado somente em nuvem.

Mahmud *et al.* (2018) propõem uma arquitetura de referência para prover interoperabilidade a soluções de *healthcare* por meio de uma integração *cloud-fog*. O arranjo dos nós e serviços *fog* foi projetado de forma a replicar uma estrutura clássica de nuvem, com um conjunto de nós computacionais em forma de *cluster* oferecendo serviços da camada *fog* por meio de *Micro Computing Instances* (MCI). Um MCI é uma instância virtual de recursos computacionais como núcleos de processamento, memória e armazenamento (similar a uma máquina virtual em nuvem). A avaliação de desempenho da arquitetura proposta foi realizada por meio de simulações e mostrou que a abordagem baseada em *fog* apresenta uma redução significativa nos requisitos de latência, consumo de energia e custo operacional, comparada a uma abordagem de nuvem clássica.

3.2 Propostas em AAL com uso de informação contextual de localização interna

Surendran e Rohinia (2019) apresentam a proposição de um sistema que utiliza a tecnologia BLE para detecção de saída de um paciente com Alzheimer de sua residência (ou qualquer local interno definido como seguro). Tal solução possibilita o monitoramento de movimento no ambiente interno através da estimação de distância entre o *beacon* BLE, implementado em um dispositivo vestível, e os receptores *bluetooth* dispostos no ambiente. Ao detectar que o paciente deixou a zona segura, um sistema de alerta envia mensagens para aplicações instaladas nos dispositivos dos cuidadores. Além disso, tais aplicações permitem o monitoramento em tempo real da localização do paciente no ambiente interno.

No trabalho de Almeida *et al.* (2019) os autores apresentam a definição de um sistema AAL baseado em IoT para monitoramento de idosos a fim de detectar previamente condições de deficiência cognitiva leve e debilidade. O sistema consiste em uma arquitetura de três camadas: camada de sistema de captura de dados pessoais, camada de repositório compartilhado e camada de modelo de análise de risco e painéis de monitoramento. A solução ainda define um subsistema

de localização interna/externa que utiliza localização baseada em *beacons* BLE para ambientes internos, e localização baseada em GPS para ambientes externos.

Em Yacchirema *et al.* (2017), um protótipo de sistema IoT para AAL, chamado *AAL-IoTSys*, é apresentado. O sistema traz o uso de um *Gateway* IoT como elemento chave para possibilitar interoperabilidade entre dispositivos e protocolos de comunicação heterogêneos. O objetivo é monitorar parâmetros de ambiente, de saúde e de localização em residências de pessoas idosas para detectar eventos críticos e enviar alertas aos cuidadores responsáveis. O serviço de localização proposto é baseado somente no uso de um sensor GPS, fornecendo uma informação de localização a nível externo.

Um sistema de monitoramento de atividades internas para detecção de debilidade em idosos é proposto em Tegou *et al.* (2019). Os autores citam o baixo custo e a facilidade de implantação como foco da solução proposta. O sistema utiliza um procedimento de localização baseado na técnica de *fingerprinting* BLE. A avaliação da solução proposta foi realizada em um ambiente real de uso, apresentando uma acurácia de localização a nível de cômodo acima de 93% e uma acurácia de 83% na classificação do status de debilidade da pessoa monitorada.

Quanto à definição de arquiteturas e/ou sistemas IoT para aplicações IPS, especificamente voltadas a prover suporte a ambientes de vida assistida, pode-se citar os trabalhos de Belmonte-Fernández *et al.* (2016) e Mainetti *et al.* (2016).

Em Belmonte-Fernández *et al.* (2016), os autores propõem um sistema de posicionamento interno baseado em dispositivos vestíveis para AAL. A solução utiliza a técnica de *fingerprinting* Wi-Fi para localização interna, onde os dados coletados por um *smartwatch* são submetidos a diferentes algoritmos de classificação utilizados em conjunto para determinar a localização final do usuário. Um *broker* IoT, utilizando o protocolo *Message Queuing Telemetry Transport* (MQTT), é responsável pela comunicação entre o *smartwatch* e o *backend* do sistema. Foram realizados experimentos em cenários reais de uso, sendo obtida uma taxa de acerto médio de 71,07% na determinação da localização do usuário a nível de cômodo.

Já o trabalho de Mainetti *et al.* (2016), apresenta a proposição de um sistema IPS para AAL voltado a pessoas idosas. A arquitetura proposta consiste em quatro camadas e utiliza BLE para realizar a determinação da localização interna, enquanto utiliza GPS para a determinação da localização externa do usuário. A avaliação é feita por meio da descrição de um caso de uso como uma prova de conceito, porém não há discussão aprofundada sobre os requisitos e critérios que o sistema atende.

3.3 Comparação dos trabalhos relacionados

A Tabela 1 apresenta uma comparação geral entre os estudos mencionados neste Capítulo e este trabalho. Foram definidos os seguintes critérios para uma visão comparativa entre os trabalhos: (i) Tipo de Proposta, que informa se o trabalho propõe uma arquitetura de referência (AR)/modelo de referência (MR), uma arquitetura concreta (AC), ou um sistema final (SF); (ii) Baseado em IoT, que informa se a proposta é definida com base em uma infraestrutura IoT; (iii) Localização Interna, que informa se a proposta fornece serviço de localização em ambientes internos; (iv) Objetivo AAL, que informa qual a finalidade da proposta de acordo com a classificação apresentada na Seção 2.4; (v) Baseado em Evidências, que informa se houve uma revisão mais rigorosa de trabalhos anteriores, como uma RSL, para chegar à proposta apresentada; e, (vi) Elicitação de RNF, que informa se foi realizado algum processo de elicitação de requisitos não funcionais para a definição da arquitetura ou sistema proposto.

Tabela 1 – Comparação entre trabalhos relacionados

Trabalho	Tipo de Proposta	Baseado em IoT	Localização Interna	Objetivo AAL	Baseado em Evidências	Elicitação de RNF
Belmonte-Fernández <i>et al.</i> (2016)	SF	Sim	Sim	G1	Não	Não
Mainetti <i>et al.</i> (2016)	AC/SF	Sim	Sim	G1.A	Não	Não
Vora <i>et al.</i> (2017)	AC/SF	Sim	Não	G1.A	Não	Não
Yacchirema <i>et al.</i> (2017)	AC/SF	Sim	Sim	G1	Não	Sim
Mahmud <i>et al.</i> (2018)	AR/AC	Sim	Não	G1.A	Parcialmente	Não
Almeida <i>et al.</i> (2019)	AC/SF	Sim	Sim	G1 e G2	Parcialmente	Não
Surendran e Rohinia (2019)	AC/SF	Não	Sim	G1.C	Não	Não
Tegou <i>et al.</i> (2019)	SF	Não	Sim	G1.A e G1.C	Não	Não
Barbosa <i>et al.</i> (2020)	AC	Sim	Não	G1.A e G1.C	Parcialmente	Não
Gulati e Kaur (2022)	SF	Sim	Não	G1.A e G1.C	Não	Sim
Hajvali <i>et al.</i> (2022)	AC	Sim	Não	G1.A e G1.C	Não	Sim
Este Trabalho	AC	Sim	Sim	G1	Sim	Sim

Fonte: Elaborado pelo Autor.

Pela Tabela 1 é possível perceber que somente três trabalhos relatam a consideração de requisitos não funcionais na definição ou avaliação da arquitetura/solução proposta. Porém, vale ressaltar que tais requisitos são abordados de forma superficial, não baseada em evidências, e limitam-se a requisitos específicos como em Yacchirema *et al.* (2017), que aborda somente o requisito de interoperabilidade.

Além disso, a definição dos componentes, tecnologias e elementos arquiteturais das soluções propostas não é baseada em evidências da academia ou indústria. Nenhum trabalho apresenta um estudo de revisão sistemática, ou processo semelhante, para fundamentar as

decisões de projeto. Alguns trabalhos baseiam-se em arquiteturas de referências já definidas (BARBOSA *et al.*, 2020; ALMEIDA *et al.*, 2019) ou realizam estudo da literatura (MAHMUD *et al.*, 2018), porém sem um método de seleção bem definido e sem deixar claro como as evidências encontradas fundamentam as decisões de projeto tomadas para a proposição da solução apresentada.

Outro aspecto importante a se levar em consideração é que poucos trabalhos apresentam de forma detalhada a especificação dos componentes, tecnologias e elementos arquiteturais definidos para a solução proposta. Dessa forma, este trabalho se propõe a ir além, definindo, baseado em evidências, uma arquitetura de software para aplicações IPS baseadas em IoT para suporte a ambientes de vida assistida.

Os principais RNFs para aplicações IPS e para aplicações no domínio de AAL são elicitados, por meio de um estudo da literatura, e levados em consideração nas decisões de projeto e na avaliação da arquitetura proposta. Além disso, um conjunto de modelos abstratos e especificações de cenários de uso da arquitetura são fornecidos com o objetivo de contribuir com o projeto de sistemas, serviços e aplicações IoT que se utilizem de informações contextuais de localização interna para suporte a ambientes de vida assistida.

4 REVISÃO SISTEMÁTICA E ELICITAÇÃO DE REQUISITOS

Neste Capítulo são apresentadas as etapas referentes à RSL realizada e à elicitação de RNF para aplicações IPS como suporte a soluções em ambientes de vida assistida. Inicialmente, fez-se necessário a realização de um estudo sobre os padrões de arquitetura de software para aplicações IoT conscientes de localização. Além disso, foi necessário compreender os principais desafios, critérios e requisitos de sistemas IPS para ambientes AAL. Ao final desse processo, chegou-se a uma proposta de arquitetura para o domínio em questão, apresentada e descrita no Capítulo 5.

4.1 Revisão Sistemática

Optou-se pela condução de uma revisão sistemática justamente para selecionar de forma mais rigorosa e baseada em evidências, os trabalhos relevantes que apresentem a implementação e/ou definição de arquiteturas de softwares para sistemas IPS no contexto de IoT. Os processos metodológicos utilizados para a RSL deste trabalho se resumem em três etapas: planejamento, condução e resultados (KITCHENHAM; CHARTERS, 2007).

Inicialmente, a fim de realizar o planejamento da RSL, foi utilizada uma adaptação do processo PICO (*Population, Intervention, Comparison, Outcome*) (KITCHENHAM; CHARTERS, 2007). O processo PICO do presente trabalho foi definido como: (P) Arquiteturas de software para sistemas IoT; (I) Implementações ou padrões arquiteturais em sistemas IPS baseados em IoT; (C) Características comuns nas arquiteturas encontradas; (O) Síntese das principais características arquiteturais em sistemas IPS baseados em IoT e arquiteturas propostas em aplicações para AAL.

Definiu-se então a *string* de busca, montando os termos de acordo com palavras-chave dentro do contexto que se pretendeu pesquisar, bem como observando as questões de pesquisa. A *string* de busca completa ficou assim:

(iot OR "internet of things") AND ("software architecture" OR "architectur pattern" OR "iot architecture") AND ("indoor location" OR "indoor localization" OR "indoor navigation" OR "indoor positioning" OR ips)*

Vale ressaltar que testes iniciais foram realizados com diversas combinações de termos relacionados ao escopo da RSL, incluindo outros que não foram considerados na *string* base (por exemplo, "architecture style" e "indoor tracking"). Assim, foi possível verificar que

a *string* apresentada acima retornou resultados mais aderentes ao escopo da RSL. Além disso, para obter resultados mais amplos sobre sistemas IPS em uma infraestrutura de IoT, optou-se por não utilizar termos relacionados ao conceito de AAL na *string* de busca. Isso permitiu capturar elementos e características compartilhados por tais sistemas e que aderiam ao paradigma de IoT, bem como investigar a existência de tais arquiteturas no domínio de AAL.

As seguintes bases de busca foram escolhidas por sua relevância científica e aderência à área de ciência da computação: *ACM Digital Library*, *IEEE Xplore* e *Scopus*. A ferramenta Parsifal¹ foi usada para o gerenciamento da revisão. Considerando as questões de pesquisa, foram definidos os critérios de inclusão e de exclusão (Tabela 2). Tais critérios foram usados para filtrar artigos especialmente relevantes e aderentes ao objetivo da RSL. A escolha de um período de 10 anos para esta seleção de artigos é justificada pela atualidade do assunto, bem como pela rápida evolução das tecnologias de IoT.

Tabela 2 – Critérios de seleção

Critérios de Inclusão	Critérios de Exclusão
CI-1: Trabalhos escritos em língua inglesa	CE-1: Trabalhos duplicados
CI-2: Trabalhos primários	CE-2: Resumo simples
CI-3: Trabalhos que apresentem a definição de uma arquitetura de software ou padrão de arquitetura de software para aplicações IPS baseadas em IoT	CE-3: Trabalhos que possuam apenas o <i>abstract</i> disponível
CI-4: Trabalhos que apresentem RNF para aplicações IPS baseadas em IoT	CE-4: Trabalhos com mais de 10 anos desde sua publicação
CI-5: Trabalhos que apresentem algum tipo de validação da arquitetura proposta	CE-5: Trabalhos que não satisfaçam às questões de pesquisa

Fonte: Elaborado pelo Autor.

A seleção dos trabalhos por meio da RSL se deu em quatro etapas, a saber: (a) pesquisa nas bases selecionadas aplicando a *string* de busca definida, resultando em 383 trabalhos retornados; (b) exclusão de artigos duplicados e com mais de dez anos desde sua publicação, resultando em dois trabalhos duplicados e sete trabalhos com data de publicação anterior a 2012; (c) primeira aplicação dos critérios de seleção por meio da leitura do título, resumo e palavras-chave, resultando em 30 artigos pré-selecionados; (d) segunda aplicação dos critérios de seleção a partir da leitura dos trabalhos por completo. Após esse procedimento, foram selecionados 13 trabalhos aderentes ao escopo desta RSL e que permitiram extrair aspectos e características importantes para a proposta da arquitetura apresentada neste trabalho. A Tabela 3 apresenta o

¹ Disponível em: <https://parsif.al/>

processo de seleção dos trabalhos com a quantidade de estudos selecionados em cada etapa. Já a Tabela 4 apresenta cada trabalho selecionados na condução desta RSL.

Tabela 3 – Processo de seleção dos trabalhos

Base	1ª Etapa	2ª Etapa	3ª Etapa	4ª Etapa
ACM Digital Library	309	301	5	1
IEEE Xplore	68	67	23	11
Scopus	6	6	2	1
Total	383	374	30	13

Fonte: Elaborado pelo Autor.

Tabela 4 – Trabalhos selecionados na RSL

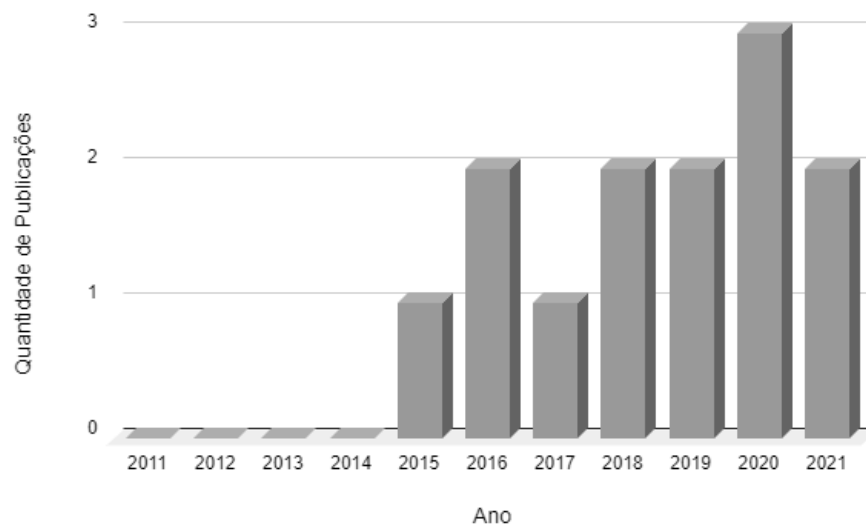
ID	Autor(es)	Título	Aplicação
A1	(ALLETTO <i>et al.</i> , 2016)	An Indoor Location-Aware System for an IoT-Based Smart Museum	Entrega de conteúdos personalizados em um museu inteligente
A2	(ALLOULAH; TUOMINEN, 2021)	IMULet: A Cloudlet for Inertial Tracking	Rastreamento inercial
A3	(BATTISTONI <i>et al.</i> , 2019)	Experimenting with a Fog-computing Architecture for Indoor Navigation	Localização interna
A4	(DU <i>et al.</i> , 2018)	An AP-centred Smart Probabilistic Fingerprint System for Indoor Positioning	Localização interna
A5	(BOUDANI <i>et al.</i> , 2021)	Positioning as Service for 5G IoT Networks	Serviço de posicionamento para redes IoT 5G
A6	(LI <i>et al.</i> , 2020)	A Self-Adaptive Bluetooth Indoor Localization System using LSTM-based Distance Estimator	Sistema de localização interna auto-adaptativo
A7	(MAINETTI <i>et al.</i> , 2015)	A Location-aware Architecture for Heterogeneous Building Automation Systems	Automação residencial
A8	(MAINETTI <i>et al.</i> , 2016)	An IoT-aware AAL System for Elderly People	AAL
A9	(MPEIS <i>et al.</i> , 2020)	The Anyplace 4.0 IoT Localization Architecture	Localização interna
A10	(NAVYA <i>et al.</i> , 2020)	A ROOF Computing Architecture based Indoor Positioning System for IoT Applications	Localização interna
A11	(NGUYEN <i>et al.</i> , 2019)	A novel architecture using iBeacons for localization and tracking of people within health-care environment	Ambiente de Assistência à Saúde
A12	(SIKERIDIS <i>et al.</i> , 2018)	Unsupervised Crowd-Assisted Learning Enabling Location-Aware Facilities	Instalações com consciência de localização
A13	(TERÁN <i>et al.</i> , 2017)	IoT-based System for Indoor Location using Bluetooth Low Energy	Localização interna

Fonte: Elaborado pelo Autor.

A distribuição dos trabalhos por ano de publicação pode ser observada no gráfico da Figura 10. Percebe-se claramente a atualidade do tópico de pesquisa relacionado ao uso de

infraestrutura IoT para aplicações de posicionamento interno. De modo geral, existem várias iniciativas e pesquisas relacionadas a sistemas IPS desde a última década. Porém, quando se trata especificamente de trabalhos que propõem e exploram uma infraestrutura de internet das coisas para prover serviços de localização, observa-se uma produção ainda tímida no meio acadêmico e científico, como é apresentado nos trabalhos de Mendoza-Silva *et al.* (2019), Lashkari *et al.* (2010) e Zafari *et al.* (2016).

Figura 10 – Trabalhos selecionados por ano de publicação



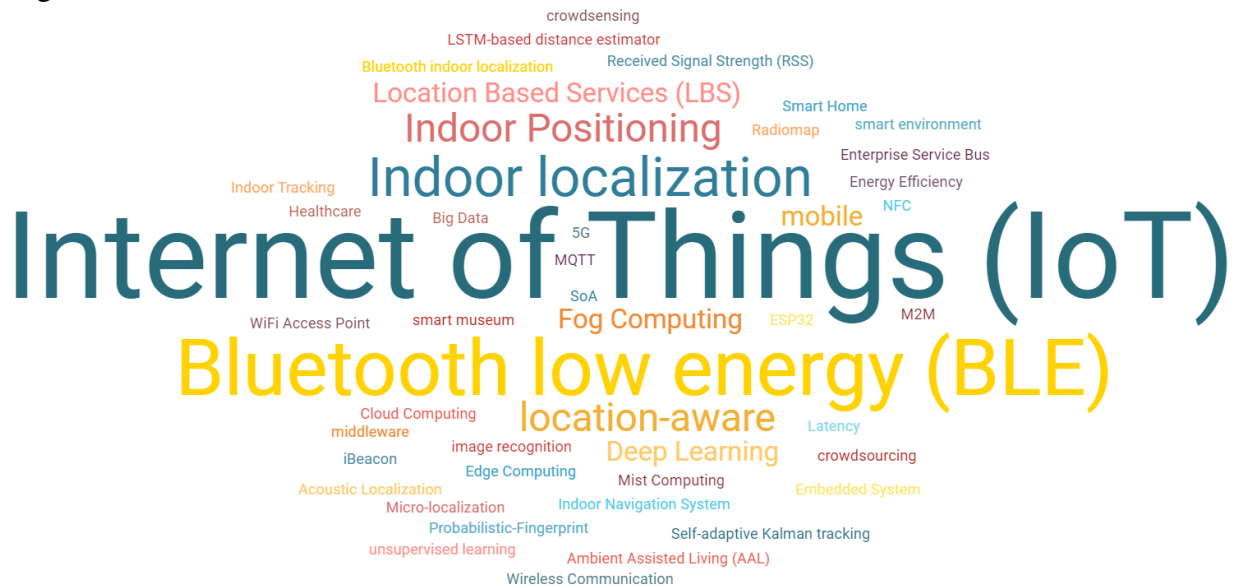
Fonte: Elaborado pelo Autor.

A Figura 11 apresenta uma nuvem de palavras com as palavras-chave encontradas nos artigos selecionados. É possível observar que os termos mais frequentes foram "*internet of things*", "*bluetooth low energy*" e "*indoor localization*". Tal informação sugere que boa parte das soluções propostas tem o BLE como tecnologia principal aplicada no processo de localização interna, o que pode ser constatado na Subseção 4.1.1.

Após a seleção dos trabalhos, por meio da leitura completa e aplicação dos critérios de seleção, foi realizado o processo de extração dos dados, seguindo o formulário de extração apresentado no Apêndice A. Com isso, a fim de responder às questões de pesquisa RQ1 e RQ2, foi realizada uma análise sobre as principais características arquiteturais, escolhas de projeto e RNF atendidos em cada proposta. Para responder a questão de pesquisa RQ3, foram identificados os trabalhos que apresentam uma solução arquitetural proposta no domínio de AAL a fim de endentificar decisões de projeto e características arquiteturais importantes nesse domínio.

4.1.1 Discussão sobre a questão de pesquisa RQ1

Figura 11 – Palavras-chave utilizadas nos trabalhos



Fonte: Elaborado pelo Autor.

RQ1: *Quais são as soluções implementadas em arquiteturas de software para sistemas IPS baseados em IoT?*

Na Tabela 5 é apresentada uma síntese das principais características das soluções propostas nos trabalhos selecionados. Como pode-se observar, o BLE é a tecnologia mais utilizada para prover informações de localização entre as soluções propostas (A1, A3, A6, A7, A8, A11, A12 e A13). Isso corrobora com uma revisão sistemática recente apresentada em Mendoza-Silva *et al.* (2019), que aponta a tecnologia BLE como uma das tecnologias mais adequadas para sistemas IPS em geral, por apresentar acurácia satisfatória em diversos contextos e por oferecer baixo custo energético e de implantação. Entre as técnicas mais utilizadas para determinar a localização ou região (zona) de um alvo, pode-se observar uma predominância das técnicas de *fingerprinting* (A4, A5, A9, A12 e A13) e proximidade por meio de medições RSSI (A1, A3, A7, A8).

O uso de técnicas de localização interna baseadas em aprendizagem de máquina também se mostrou expressivo. Os trabalhos A2, A5 e A6 utilizam técnicas de *Deep Learning*, como CNN e LSTM, para a determinação da localização do usuário. Em A4, A9, A10, A11 e A13, os autores utilizam algoritmos de aprendizado supervisionado, como o WK-NN, SVM, K-NN e Naive Bayes, em conjunto com técnicas de localização, como *fingerprinting* e localização acústica. Já em A12 os autores utilizam uma técnica de localização totalmente baseada em aprendizado não supervisionado, através do algoritmo de maximização de expectativa. Em A13, os autores propõem um outro cenário de avaliação, utilizando uma estratégia não supervisionada,

Tabela 5 – Principais características das soluções propostas

ID	Características Arquiteturais	Tecnologias/Métodos	Comunicação	Acurácia
A1	Arquitetura composta por 6 componentes: infraestrutura de localização BLE, dispositivo vestível, centro de processamento, <i>middleware</i> , ambiente IoT heterogêneo e nuvem	BLE - Técnica de Proximidade (RSSI)	Wi-Fi	100% (Proximidade)
A2	Utiliza uma arquitetura cloudlet, ou nuvem de borda, distribuindo o processamento dos dados provenientes da IMU entre dispositivo final e nuvem de borda	IMU - CNN (Convolutional Neural Network) e LSTM (Long Short Term Memory)	NR	< 1,00 m
A3	Utiliza uma arquitetura baseada em <i>Fog Computing</i> , com possibilidade de distribuição geográfica	BLE - Técnica de Proximidade (RSSI)	Wi-Fi e MQTT	NR
A4	Utiliza uma arquitetura baseada em dois módulos: infraestrutura Wi-Fi e servidor centralizado em nuvem com RESTful API, que realiza o processamento dos dados	Wi-Fi - Fingerprinting (WKN)	Wi-Fi	≈ 2,00 m
A5	Arquitetura baseada em 5 camadas: borda, acesso, núcleo, controle e aplicação	Wi-Fi - Fingerprinting e Deep Learning	5G	1,60 m
A6	Arquitetura distribuída baseada em 5 camadas: hardware, camada de dados, camada de negócio, camada de apresentação e interface de usuário	BLE - Trilateração, LSTM e Filtro Kalman	MQTT e gRPC	≈ 0,90 m - 1,50 m
A7	Arquitetura baseada em 3 componentes: infraestrutura de dispositivos IoT, <i>home gateway</i> (processamento e <i>middleware</i>) e dispositivo móvel	BLE - Técnica de Proximidade (RSSI)	KNX e CoAP	NR
A8	Arquitetura baseada em 4 camadas: dispositivos finais, <i>middleware</i> de sensoriamento, <i>middleware</i> de integração e camada de negócios	BLE/GPS - Técnica de Proximidade (RSSI)	Wi-Fi e GPRS	NR
A9	Arquitetura baseada em 3 componentes: clientes IoT, servidor e persistência	Wi-Fi - Fingerprinting	NR	NR
A10	Arquitetura composta de 3 camadas: borda, ROOF e aplicação	Localização Acústica - Aprendizado supervisionado (SVM, KNN e Naive Bayes)	Wi-Fi e MQTT	> 71% (Zona)
A11	Arquitetura composta de 3 componentes: infraestrutura BLE, dispositivo móvel com app e servidor	BLE - Least Square Estimation	NR	0,12 m
A12	Arquitetura composta de 3 camadas: camada de sensoriamento, camada de borda e camada de nuvem centralizada	BLE - Fingerprinting, Aprendizado não supervisionado e crowdsensing	Wi-Fi e MQTT	≈ 3,00 m - 5,00 m
A13	Arquitetura cliente servidor com dois componentes: sistema de aquisição de dados e servidor central	BLE - Fingerprinting (KNN e k-Means)	Wi-Fi	70,2% (Zona)

Fonte: Elaborado pelo Autor.

com o algoritmo de clusterização k-Means. A Tabela 6 apresenta a distribuição do uso das tecnologias, técnicas e métodos de aprendizagem de máquina para localização interna utilizados em cada um dos trabalhos.

Em relação à comunicação entre os módulos ou camadas das arquiteturas propostas, há uma predominância do uso do protocolo MQTT (A3, A6, A10 e A12) e de conectividade sem fio com a tecnologia Wi-Fi (A1, A3, A4, A8, A10, A12 e A13). O MQTT funciona como um protocolo leve para troca de mensagens entre dispositivos produtores e consumidores, através de um componente intermediário chamado Broker MQTT. Tal protocolo possibilita a comunicação entre sensores, dispositivos móveis e diversos outros dispositivos e componentes do sistema,

Tabela 6 – Tecnologias, técnicas e métodos para localização interna

		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
Tecnologias	BLE	X		X			X	X	X			X	X	X
	Wi-Fi				X	X				X				
	GPS								X					
	Localização Acústica										X			
	IMU		X											
Técnicas	Proximidade	X		X				X	X					
	Fingerprinting				X	X				X			X	X
	Trilateração						X							
Abordagens de ML	Deep Learning		X			X	X							
	Aprendizado Supervisionado				X					X	X	X		X
	Aprendizado não Supervisionado												X	X

Fonte: Elaborado pelo Autor.

mantendo baixo consumo energético e menor ocupação da taxa de transferência de dados da rede. Esses requisitos são desejáveis em sistemas IPS baseados em IoT.

A organização predominantemente adotada nas soluções apresentadas é de uma arquitetura baseada em camadas, corroborando com o que foi discutido na Seção 2.3. Arquiteturas baseadas em 3 camadas são apresentadas em A3, A7, A9 e A10, onde de forma geral há uma camada de sensoriamento, uma camada de rede e uma camada de aplicação. Outras arquiteturas se baseiam em 4 camadas (A8) ou 5 camadas (A5 e A6), havendo a adição de componentes de *middleware* para lidar com dados provenientes de sensores heterogêneos (A8) ou para realizar comunicação entre módulos heterogêneos e/ou remotos (A6).

Outro aspecto arquitetural importante é relativo ao emprego de computação em nuvem e tecnologias relacionadas. Nesse contexto, os trabalhos A1, A4 e A12 abordam o uso de uma camada de nuvem centralizada responsável por receber os dados provenientes dos sensores e da camada de borda, a fim de realizar o processamento e determinação da localização, bem como entregar outros serviços às camadas inferiores e a usuários remotos. Já os trabalhos A2, A3 e A10, empregam conceitos modernos relativos à *cloudlet*, *fog computing* e *roof computing*, na tentativa de oferecer serviços de localização em tempo real, com baixa latência, por meio do processamento mais próximo ao usuário final. A Tabela 7 apresenta a distribuição do uso das tecnologias de comunicação e abordagens de computação em nuvem empregadas em cada trabalho.

4.1.2 Discussão sobre a questão de pesquisa RQ2

Tabela 7 – Tecnologias de comunicação e computação em nuvem

		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
Tecnologias de Comunicação	Wi-Fi	X		X	X				X		X		X	X
	MQTT			X			X				X		X	
	5G					X								
	gRPC						X							
	GPRS								X					
	KNX							X						
	CoAP							X						
Computação em Nuvem	Nuvem Centralizada	X			X								X	
	Fog Computing			X										
	Roof Computing										X			
	Cloudlet		X											

Fonte: Elaborado pelo Autor.

RQ2: *Quais são os requisitos não funcionais incorporados às arquiteturas de software definidas para sistemas IPS baseados em IoT?*

Uma convergência entre os requisitos de projeto de sistemas IPS apresentados em (ZAFARI *et al.*, 2019), (ZAFARI *et al.*, 2016), (LIU; JAIN, 2014) e (BASIRI *et al.*, 2017) foi utilizada para analisar cada um dos trabalhos selecionados. Com isso, foram considerados os seguintes requisitos: acurácia, eficiência energética, custo computacional, custo de implantação, complexidade, cobertura, tempo de resposta, integração com localização externa, privacidade, segurança e escalabilidade.

A seguir é feita uma breve discussão sobre como são contemplados os requisitos de sistemas IPS nos trabalhos selecionados. Somente os requisitos de acurácia, tempo de resposta, eficiência energética, integração com localização externa, privacidade, segurança e escalabilidade serão discutidos. Em relação ao custo de implantação, todas as soluções propostas abordam tal critério utilizando ou propondo tecnologias abertas e dispositivos de baixo custo. Nenhum trabalho aborda explicitamente os requisitos de complexidade, custo computacional e cobertura. A Tabela 8 mostra quais requisitos são atendidos/abordados em cada um dos trabalhos analisados.

4.1.2.1 Acurácia

O requisito de acurácia em um sistema IPS está relacionado ao nível de proximidade da localização estimada em relação à localização real do objeto/indivíduo naquele ambiente, sendo o requisito mais importante para fins de avaliação e validação de soluções de localização interna.

Tabela 8 – Requisitos não funcionais abordados nos trabalhos selecionados

		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
Requisitos não Funcionais	Acurácia	X	X		X	X	X				X	X	X	X
	Tempo de Resposta		X	X	X						X			
	Eficiência Energética	X		X	X		X	X	X			X	X	X
	Integração com Localização Externa								X					
	Escalabilidade			X							X			
	Privacidade									X				
	Segurança			X										

Fonte: Elaborado pelo Autor.

Inicialmente, vale ressaltar que os trabalhos apresentam divergências em relação à forma de validação de suas propostas de sistemas IPS, bem como em relação às métricas de avaliação utilizadas. Ainda nesse sentido, as dimensões dos ambientes de testes dos sistemas propostos são extremamente variáveis. Alguns trabalhos trazem uma validação funcional da arquitetura por meio de análise de requisitos e/ou prova de conceito, não havendo assim um experimento prático que possibilite verificar a acurácia de localização (A3, A7, A8 e A9). Visto que o foco de tais trabalhos, assim como este, é a proposição de uma arquitetura baseada em IoT para um sistema IPS, esse requisito está bastante dependente das tecnologias e técnicas de localização que o sistema poderá utilizar.

No trabalho A1 (ALLETTO *et al.*, 2016), os autores utilizam uma técnica simples de proximidade e apresentam uma acurácia de 100% na determinação da sala correta onde o usuário está localizado dentro do ambiente. Dessa forma, a solução não contempla situações onde é necessário uma determinação mais precisa da localização, como rastreamento e navegação interna em tempo real. Os experimentos realizados em A10 (NAVYA *et al.*, 2020) e A13 (TERÁN *et al.*, 2017), utilizam a técnica de *fingerprinting* baseada em sinais acústicos e BLE respectivamente, apresentando acurácia de 71% e 70,2% na determinação da região alvo, nessa ordem.

Os demais trabalhos apresentam resultados de acurácia em valores absolutos, utilizando metros (m) como unidade de medida. Pela Tabela 4 pode-se perceber erros de localização bastante variáveis, que vão desde 0,12 m até erros acima de 5 m. Isso pode ser explicado, como mencionado anteriormente, pela divergência nas dimensões dos ambientes de testes e nas métricas utilizadas para representar o valor de acurácia. Por exemplo, o trabalho A11 (NGUYEN *et al.*, 2019) apresentou uma acurácia de 0,12 m, o que pode ser entendido como uma acurácia ideal para sistemas críticos de navegação interna em tempo real. Porém, a área do ambiente de teste foi de somente 20 m². Em contrapartida, o trabalho A12 (SIKERIDIS *et al.*, 2018)

apresentou uma acurácia de aproximadamente 3 m para um cenário estático e 5 m para um cenário de movimento. Contudo, o ambiente de teste tinha uma área de $5.100 m^2$ consistindo de 3 andares de um prédio.

4.1.2.2 *Tempo de Resposta*

O tempo de resposta de um sistema IPS está relacionado ao tempo total necessário para o sistema, a partir do momento que é solicitado, retornar a posição estimada de um alvo à quem realizou a solicitação. Em cenários de localização em tempo real, como em um AAL, esse critério se torna extremamente relevante. Além disso, esse é um critério fortemente influenciado pela arquitetura em uma infraestrutura IoT.

Em A3 (BATTISTONI *et al.*, 2019), os autores realizam uma avaliação da arquitetura por meio de um método baseado em cenários. Logo, não há apresentação de nenhum resultado experimental em relação ao tempo de resposta do sistema. Porém, ao analisarem o critério de baixa latência em sua avaliação, eles afirmam que a arquitetura proposta satisfaz aplicações com requisitos de baixo tempo de resposta. Isso é devido à utilização de *fog computing*, tornando o processamento dos dados mais próximo aos usuários finais. Nesse sentido, os trabalhos A10 (NAVYA *et al.*, 2020) e A2 (ALLOULAH; TUOMINEN, 2021) também abordam o uso de computação próxima aos dispositivos finais, utilizando conceitos de *roof computing* e *cloudlet*.

O trabalho A4 (DU *et al.*, 2018) é o único trabalho que apresenta resultados experimentais para o tempo de resposta. No experimento é alcançada uma latência um pouco superior a 1.500 ms, mesmo utilizando uma estratégia de servidor em nuvem centralizada. Os autores ainda comentam que o tempo de resposta seria inferior caso o servidor fosse localizado na borda da rede.

4.1.2.3 *Eficiência Energética*

O requisito de eficiência energética faz-se bastante relevante uma vez que a infraestrutura de localização é composta por sensores e dispositivos de pequeno porte que devem se comportar de forma pervasiva no ambiente. Vários trabalhos abordam esse requisito por meio da utilização da tecnologia BLE, que oferece um custo energético baixo e atrativo a tais aplicações (A1, A3, A6, A7, A8, A11, A12, A13).

Em A4 (DU *et al.*, 2018), os autores conseguem uma redução de 90% no consumo energético do dispositivo móvel do usuário por meio da aplicação da técnica de *fingerprinting*

centrada nos pontos de referência (PR). Isso retira a responsabilidade do dispositivo móvel de escanear pacotes de anúncio dos *beacons* e repassa aos PRs.

4.1.2.4 *Integração com Localização Externa*

Sistemas de posicionamento interno, muitas vezes necessitam trabalhar em conjunto com mecanismos de posicionamento externo para oferecer uma solução ubíqua ao usuário. Assim, ao sair de um ambiente interno assistido por meio de informações de localização, o sistema deve, automaticamente, alternar para uma estratégia de localização externa. Nesse contexto, somente o estudo apresentado em A8 (MAINETTI *et al.*, 2016) aborda esse critério, onde ao ser detectado que o usuário deixou o ambiente interno, o sistema passa a colher informações de um receptor GPS e realizar comunicação por meio de um módulo de *General Packet Radio Service* (GPRS).

4.1.2.5 *Escalabilidade*

O sistema tem natureza escalável quando consegue atender demandas futuras, possivelmente crescentes, sem grandes aumentos em custos de implantação e operação que inviabilizem tal expansão. Nesse contexto, os trabalhos apresentados em A3 (BATTISTONI *et al.*, 2019) e A10 (NAVYA *et al.*, 2020), apresentam forte aderência a esse critério pelo emprego de *fog computing* e *roof computing*. Dessa forma, podem oferecer escalabilidade vertical adicionando nós em camadas superiores e escalabilidade horizontal adicionando nós em uma mesma camada, seja localmente ou remotamente em uma estratégia de *fog computing* distribuído.

4.1.2.6 *Privacidade e Segurança*

Privacidade e segurança são dois critérios interligados e bastante relevantes no contexto de sistemas IPS dada a sensibilidade das informações geradas e processadas. Em alguns contextos a vulnerabilidade à exposição da localização e demais dados pessoais de usuários do sistema pode ter um impacto extremamente negativo. Contudo, muitas aplicações voltadas a posicionamento interno não levam em consideração aspectos relativos à privacidade e segurança de seus usuários e informações (BASIRI *et al.*, 2017). Isso pode ser percebido nos trabalhos selecionados na RSL.

Somente dois trabalhos abordam aspectos relativos à privacidade e segurança. Em A3 (BATTISTONI *et al.*, 2019) é proposto uma estratégia de segurança a nível eletrônico,

garantindo *boot* seguro e criptografia da memória *flash*. Isso previne modificação ou acesso indevidos ao código presente nos nós de coleta das informações. Porém, nenhuma medida de segurança é aplicada às tarefas de comunicação de dados, nem mesmo alguma medida de garantia de privacidade é discutida.

Já em A9 (MPEIS *et al.*, 2020), o critério de privacidade é abordado por meio de uma biblioteca android fornecida aos desenvolvedores de aplicações de posicionamento interno que utilizem o *framework* proposto no trabalho. Essa biblioteca possibilita o armazenamento e gerenciamento privado das informações de localização provenientes do servidor. Contudo, nenhuma medida de segurança contra ataques, interceptações ou invasões, por exemplo, é discutida.

4.1.3 Discussão sobre a Questão de Pesquisa RQ3

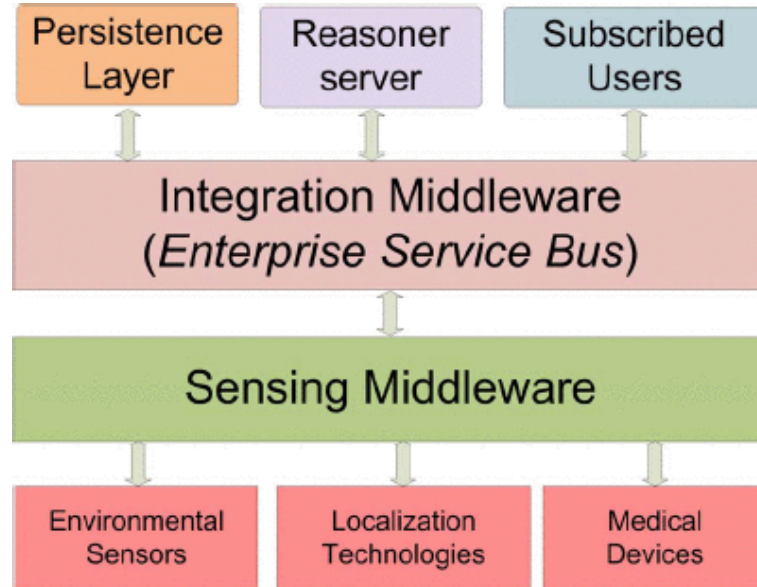
RQ3: Existem arquiteturas propostas para sistemas IPS baseados em IoT para AAL?

Em relação à questão de pesquisa RQ3, que investiga se há algum trabalho que propõe uma arquitetura de sistema IPS baseado em IoT para ambientes de vida assistida, somente um trabalho foi identificado para essa finalidade. O estudo A8 (MAINETTI *et al.*, 2016) traz a proposta de um sistema IoT com consciência de contexto para suporte a AAL voltado a pessoas idosas. A arquitetura constitui-se de quatro camadas, como pode ser observado na Figura 12: infraestrutura IoT, *middleware* de sensoriamento, *middleware* de integração, camada de negócios e aplicações.

Na camada de infraestrutura são projetados três componentes: sensores, tecnologias de localização e dispositivos médicos. A ideia é fornecer a capacidade de prover localização do usuário e dados sobre sua saúde e eventos críticos, como por exemplo detecção de queda. Na camada superior, um *middleware* de sensoriamento realiza a coleta dos dados dos sensores e envia à camada de *middleware* de integração. Tal camada, por sua vez, interliga a camada de *middleware* de sensoriamento com o nível mais alto, onde estão presentes os componentes de notificação (em caso de algum evento crítico), persistência e um servidor com as regras de negócio definidas.

Além disso, a proposta traz uma integração com mecanismo de posicionamento externo, por meio de um receptor GPS, permitindo um monitoramento contínuo da localização do indivíduo. Quanto à sua classificação, em relação ao objetivo AAL, a solução proposta configura-se como um sistema voltado à saúde, reabilitação e cuidados (G1.A). A Figura 13

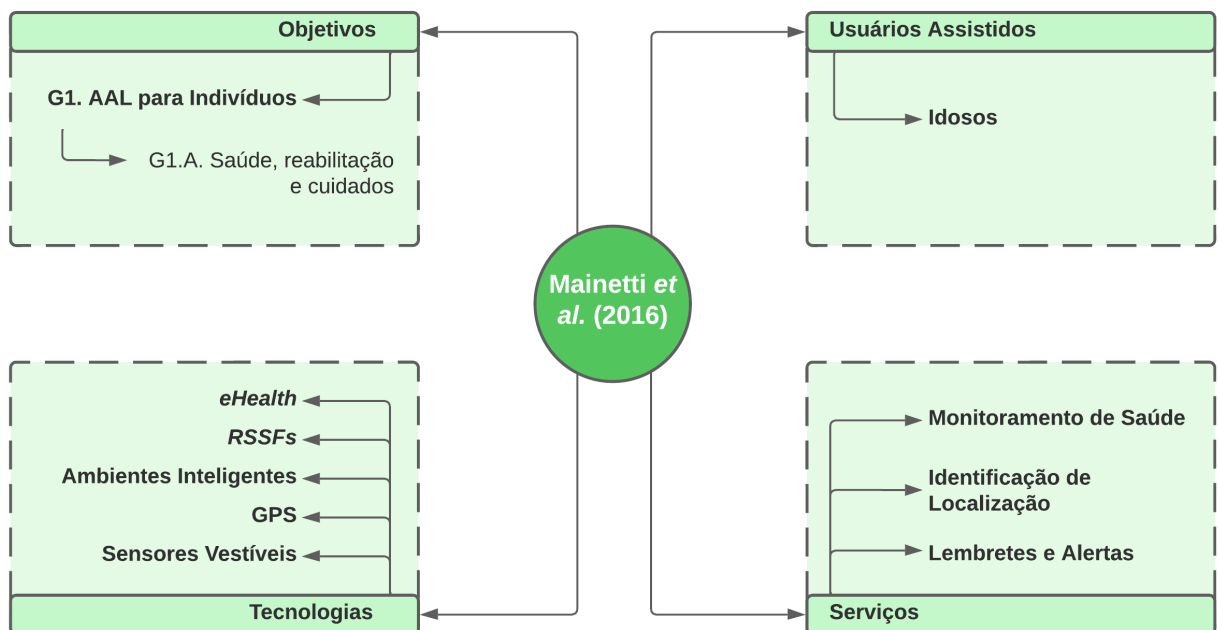
Figura 12 – Visão geral da arquitetura proposta por Mainetti *et al.* (2016)



Fonte: Mainetti *et al.* (2016)

sintetiza as principais características encontradas na proposta de Mainetti *et al.* (2016).

Figura 13 – Características da arquitetura proposta por Mainetti *et al.* (2016)



Fonte: Elaborado pelo Autor.

Por fim, vale ressaltar que nenhum outro requisito de sistemas IPS é abordado explicitamente. A validação feita se baseia em uma prova de conceito sem qualquer análise baseada em algum método de validação mais criterioso. Não há discussão em relação ao atendimento aos requisitos básicos de um sistema IPS para ambientes de vida assistida. Seguindo

nesse mesmo entendimento, o artigo também não apresenta claramente nenhum levantamento de requisitos para sistemas AAL.

4.2 Elicitação de Requisitos

RQ4: *Quais são os principais requisitos não funcionais para sistemas AAL baseados em IoT?*

Com a síntese de características e padrões para arquiteturas de aplicações de IPS baseadas em IoT, realizada pela RSL descrita na Subseção 4.1, partiu-se para um processo de elicitação de requisitos não funcionais para sistemas de apoio a AAL. Isso possibilita propor uma arquitetura que possa oferecer um bom suporte ao desenvolvimento e operação de aplicações de IPS voltadas a AAL se utilizando de uma infraestrutura IoT.

A fim de responder a questão de pesquisa RQ4 e realizar uma elicitação de requisitos não funcionais necessários a uma aplicação de AAL com informações de localização interna em uma infraestrutura IoT, os estudos apresentados em Ahmadi *et al.* (2019), Islam *et al.* (2015) e Barbosa *et al.* (2020) foram analisados. O intuito foi realizar um reuso de requisitos já consolidados em estudos anteriores.

O trabalho de Islam *et al.* (2015) traz uma análise abrangente e relevante sobre vários aspectos relacionados ao uso de uma infraestrutura de IoT em sistemas e serviços de saúde. Os autores discutem possibilidades de arquiteturas/plataformas, aplicativos, tecnologias e requisitos de segurança para sistemas de saúde baseados em IoT. Em termos de RNF, o trabalho se concentra nos desafios e requisitos de segurança e privacidade, como: confidencialidade, integridade, autenticação/autorização, disponibilidade e tolerância a falhas.

Já no trabalho de Ahmadi *et al.* (2019), os autores realizaram uma RSL para determinar as principais áreas de aplicação da IoT na área da saúde, bem como arquiteturas, componentes e tecnologias-chave. Em relação aos RNF, o estudo destaca os requisitos de segurança e interoperabilidade como os principais desafios a serem considerados em um projeto de sistema de saúde que usa uma infraestrutura IoT. Com relação à segurança, os autores enfatizam a necessidade de garantir a privacidade e a confidencialidade dos dados que estão sendo transmitidos e de proteger o sistema contra vulnerabilidades que podem se manifestar em três categorias: vulnerabilidade de hardware, vulnerabilidade de rede e vulnerabilidade de aplicativo.

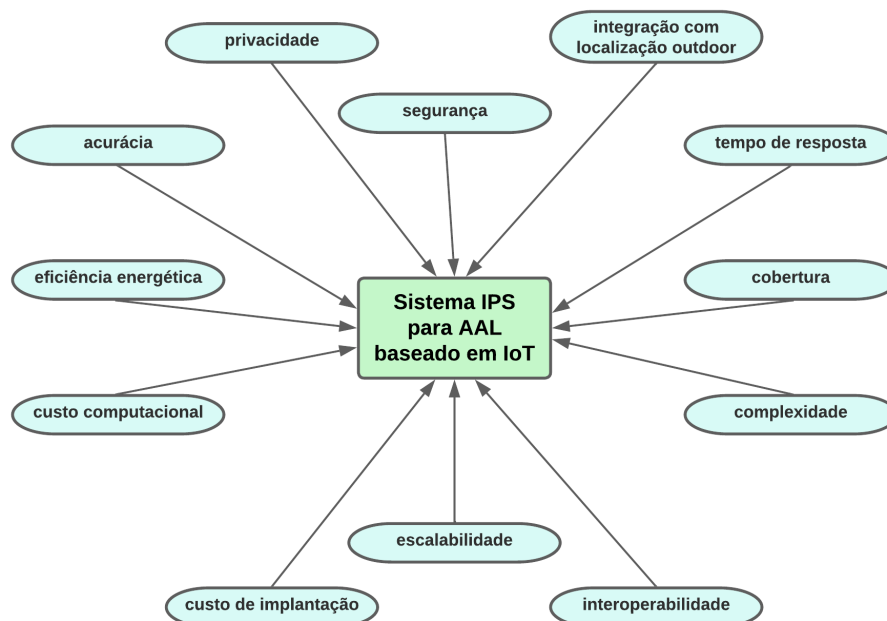
Ainda de acordo com Ahmadi *et al.* (2019), a heterogeneidade dos dados, os protocolos de comunicação e as várias tecnologias que podem ser empregadas tornam a interoperabili-

dade dos sistemas de IoT um grande desafio no domínio da saúde, dificultando o uso de soluções integradas.

Calvillo-Arbizu *et al.* (2021) apresentam uma RSL que busca identificar os principais requisitos relacionados aos sistemas de IoT no domínio da saúde. Como resultado, os autores destacam os requisitos de segurança, privacidade e questões de interação humano-computador. Em relação à segurança e à privacidade, há a necessidade de autenticação/autorização, confidencialidade, privacidade dos dados, confiabilidade, integridade e disponibilidade. Os autores também destacam a necessidade de considerar questões relacionadas à interação humano-computador, como acessibilidade, usabilidade e interação com o usuário.

Considerando que o presente trabalho apresenta uma proposta inicial, as questões relativas à interação humano-computador não foram abordadas neste primeiro momento. Portanto, os requisitos de segurança, privacidade e interoperabilidade foram destacados nos trabalhos analisados e considerados na definição da arquitetura proposta. A partir da convergência desses requisitos com os RNF descritos na Subseção 4.1.2, a arquitetura proposta foi projetada com base nos seguintes requisitos (Figura 14): acurácia de localização, eficiência energética, custo computacional, custo de implementação, complexidade, cobertura, tempo de resposta, integração com localização externa, privacidade, segurança, escalabilidade e interoperabilidade.

Figura 14 – Requisitos não funcionais considerados



Fonte: Elaborado pelo Autor.

5 ARQUITETURA PROPOSTA

A arquitetura proposta neste trabalho foi resultado de decisões de projeto baseadas nas evidências encontradas na RSL realizada (Capítulo 4). A Tabela 9 apresenta e discute as principais decisões de projeto, bem como o mapeamento entre elas e os artigos revisados na RSL.

Tabela 9 - Decisões de Projeto

DP-01: Paradigmas Fog e Mist Computing	
RNFs Alvos	tempo de resposta, eficiência energética, escalabilidade, custo computacional, complexidade, cobertura
Artigos Considerados	A2, A3 e A10
Camadas e Componentes	Camada de Percepção → Módulo Mist Computing; Camada Fog
Discussões Críticas	Ao empregar recursos de processamento de dados mais próximos dos usuários finais, o tempo de resposta do sistema pode ser significativamente reduzido. Além disso, ao adotar essa abordagem, o sistema se torna escalável por natureza, já que a escalabilidade horizontal pode ser obtida com a adição de mais nós de computação na mesma camada e a escalabilidade vertical com a adição de mais níveis na camada fog. Por fim, com relação à cobertura do sistema, usando uma estratégia de fog distribuída, é possível cobrir uma área geográfica maior e interconectar vários pontos de atendimento no contexto de uma <i>smart city</i> , por exemplo.
DP-02: Localização Baseada em BLE	
RNFs Alvos	eficiência energética, acurácia, custo de implantação
Artigos Considerados	A1, A3, A6, A7, A8, A11, A12 e A13
Camadas e Componentes	Camada de Percepção → Sensores e Atuadores
Discussões Críticas	O uso da tecnologia BLE foi considerado devido à característica inerente de baixo consumo de energia que ela oferece. Além disso, essa tecnologia é bastante comum em vários tipos de dispositivos de computação de baixo custo, como microcontroladores e plataformas de desenvolvimento de sistemas embarcados. Por fim, há evidências na literatura que apontam para um nível mais alto de acurácia de localização quando se usa BLE em vez de Wi-Fi.

Continua na próxima página

Tabela 9 - *Continuação da tabela*

DP-03: Localização Centrada na Camada Fog	
RNFs Alvos	eficiência energética, complexidade
Artigos Considerados	A4
Camadas e Componentes	Camada Fog → Módulo de Pré-processamento; Camada Fog → Módulo de Processamento
Discussões Críticas	Diferentemente de um esquema clássico de localização interna baseado em RSSI e/ou <i>fingerprinting</i> , a responsabilidade pela execução do processamento de localização agora é da camada fog e não mais do dispositivo do usuário. Dessa forma, é possível reduzir significativamente a complexidade computacional das tarefas executadas no dispositivo final. Como resultado, o dispositivo final do usuário pode ter um consumo de energia muito baixo, o que é fundamental para prolongar a vida útil da bateria no dispositivo do usuário.
DP-04: Localização Externa via GPS	
RNFs Alvos	integração com localização externa e cobertura
Artigos Considerados	A8
Camadas e Componentes	Camada de Aplicação
Discussões Críticas	O aplicativo do usuário final deve ser projetado para mudar para uma estratégia de localização externa quando o usuário deixar o ambiente interno. Assim, por meio de uma localização baseada em GPS, é possível manter o monitoramento contínuo da localização do usuário.
DP-05: Nuvem Baseada em Microsserviços e APIs RESTful	
RNFs Alvos	escalabilidade, interoperabilidade, segurança, privacidade
Artigos Considerados	A5 e A9
Camadas e Componentes	Camada de Nuvem

Continua na próxima página

Tabela 9 - *Continuação da tabela*

Discussões Críticas	O uso de uma camada de nuvem baseada em microsserviços ajuda a obter maior escalabilidade do servidor, ajustando os recursos e serviços necessários de acordo com a variação da demanda. Além disso, o uso de APIs RESTful permite uma melhor interoperabilidade com outros sistemas e aplicativos e pode ser usado para oferecer serviços de autenticação e autorização, proporcionando um nível mais alto de segurança e privacidade ao acessar dados e serviços.
DP-06: Broker Local	
RNFs Alvos	tempo de resposta, interoperabilidade
Artigos Considerados	A3, A10 e A12
Camadas e Componentes	Camada Fog → Broker
Discussões Críticas	Um broker de mensagens local é usado para realizar uma comunicação rápida e leve entre os dispositivos finais (atuadores e sensores) e os dispositivos de processamento da camada fog. Isso contribui potencialmente para a baixa latência e o baixo uso da rede. Além disso, por ser um padrão de comunicação amplamente difundido, com protocolos bem estabelecidos, a comunicação entre dispositivos heterogêneos é facilitada.

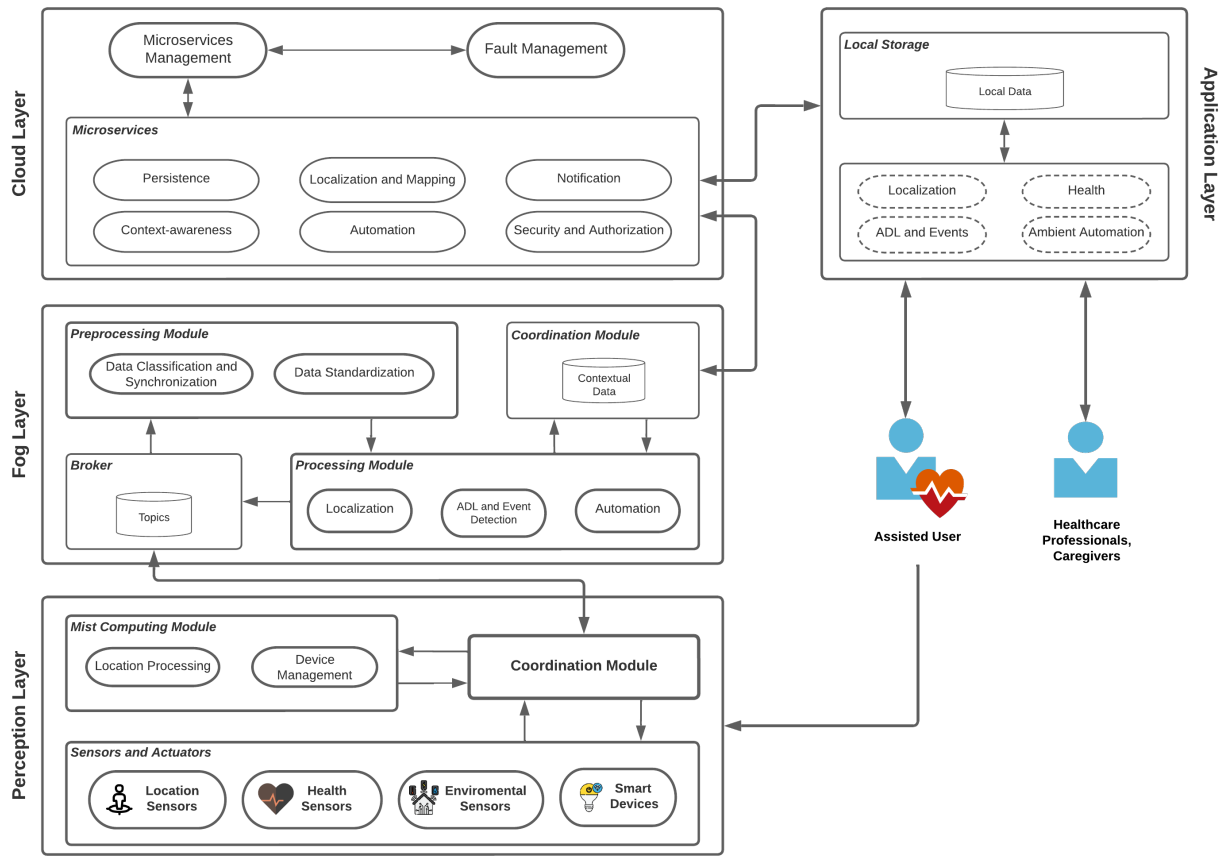
Fonte: Elaborado pelo Autor.

A partir das decisões de projeto supracitadas chegou-se a definição de uma arquitetura de software para sistemas IPS baseados em IoT para suporte a AAL consistindo de quatro camadas principais: camada de percepção, camada fog, camada de nuvem, e camada de aplicação, como pode ser observado na Figura 15. Cada camada é descrita a seguir.

5.1 Camada de Percepção

Essa camada contém uma ampla variedade de sensores (por exemplo, sensores de saúde, sensores de ambiente e sensores de localização) e atuadores, proporcionando um ambiente de vida assistida, inteligente e personalizado. Os usuários assistidos usam um dispositivo vestível com BLE e Wi-Fi, unidades de medição inercial (IMU) e sensores para parâmetros de saúde (por exemplo, frequência cardíaca). Esse dispositivo, por sua vez, envia periodicamente pacotes de anúncios BLE para *scanners* presentes no ambiente. Além disso, o dispositivo vestível transmite dados dos sensores inerciais e parâmetros de saúde para o módulo de comunicação, que, por sua

Figura 15 – Arquitetura proposta



Fonte: Elaborado pelo Autor.

vez, os encaminha para a camada fog.

A infraestrutura de localização BLE (módulo de *mist computing*) é composta por vários *scanners (mist nodes)* posicionados ao longo dos espaços do ambiente assistido, responsáveis por escanear os pacotes de anúncio dos dispositivos finais e extrair a informação referente à intensidade do sinal recebido. Nesse momento, esses dispositivos encaminham uma mensagem ao módulo de coordenação (*coordination module*) informando o identificador do dispositivo escaneado, seu identificador, a potência do sinal recebido e uma marca de tempo (*timestamp*). Essa abordagem se baseou na proposta apresentada em Du *et al.* (2018), centrando nos pontos de referência (*scanners* BLE) a responsabilidade sobre a coleta de informações para localização.

Outra responsabilidade dos *mist nodes* é a interação com dispositivos inteligentes, presentes no ambiente (ar-condicionado, aparelhos de TV e iluminação, por exemplo) e o monitoramento de sensores de parâmetros ambientais, como sensor de temperatura, sensor de incêndio e sensor de gás.

5.2 Camada Fog

A camada fog consiste em dispositivos com mais recursos computacionais do que a camada inferior. Como resultado, esses dispositivos, chamados de *fog nodes*, podem executar tarefas de processamento e armazenamento, trazendo elementos da computação em nuvem para a borda da rede. Conforme discutido na Subseção 2.3.1, o objetivo é distribuir serviços mais próximos dos usuários finais, reduzindo o tempo de resposta, que é um requisito desejável no domínio AAL.

Essa camada é responsável por processar os dados recebidos usando algoritmos para localização, detecção de ADL e detecção de eventos críticos, por exemplo. O resultado desse processamento pode ser encaminhado para o *mist node* mais próximo do ambiente onde o usuário assistido está para ativar algum serviço ou dispositivo, se for o caso. Essas informações também devem ser enviadas ao servidor de nuvem por meio de uma API RESTful.

Em caso de detecção de qualquer evento crítico, o dispositivo fog deve encaminhar uma notificação para a camada de nuvem juntamente com a localização do usuário. Além disso, a camada fog pode ser organizada de forma hierárquica e distribuída. Dessa forma, os recursos computacionais podem ser garantidos mais perto dos dispositivos finais, reduzindo a necessidade de comunicação com a camada de nuvem.

5.3 Camada de Nuvem

O servidor em nuvem é responsável por armazenar, concentrar e processar as informações provenientes dos ambientes assistidos para disponibilizá-las convenientemente às aplicações. A fim de facilitar a comunicação entre o *back-end* e a infraestrutura do ambiente assistido, bem como garantir interoperabilidade e segurança no acesso aos serviços e dados do servidor, uma API RESTful é utilizada para realizar a integração com o ambiente assistido. De forma análoga, outra API é disponibilizada para as aplicações desenvolvidas que consomem os dados e serviços da plataforma e interagem com o ambiente assistido.

Objetivando garantir escalabilidade e modularidade ao servidor, o sistema é organizado em uma arquitetura de microsserviços, composta de seis serviços principais. O serviço de persistência garante o armazenamento seguro dos dados provenientes do ambiente assistido, possibilitando recuperá-los posteriormente a fim de obter um histórico ou realizar alguma análise dos dados. Os processos de mapeamento do ambiente, configuração do posicionamento da

infraestrutura BLE e configuração dos parâmetros utilizados no algoritmo de localização, são realizados pelo serviço de mapeamento.

O serviço de notificações envia alertas aos usuários da plataforma, sejam usuários assistidos, cuidadores ou profissionais, referentes a eventos relevantes como, por exemplo, a detecção de uma queda. Por sua vez, o serviço de automação é responsável por gerenciar os dispositivos inteligentes bem como as regras de negócio para interagir com tais dispositivos, por exemplo, para definir quando ligar uma luz ou um ar-condicionado. O serviço de reconhecimento de contexto executa regras sobre os dados coletados a fim de identificar e produzir dados contextuais utilizados nos módulos de processamento da camada fog para, por exemplo, identificar/detectar possíveis situações de risco. Por fim, o serviço de autorização é projetado para garantir um acesso seguro aos serviços e informações da plataforma. Ele garante a identidade dos usuários (ou entidades), permitindo ou negando o acesso por meio das APIs disponibilizadas pelo servidor.

5.4 Camada de Aplicação

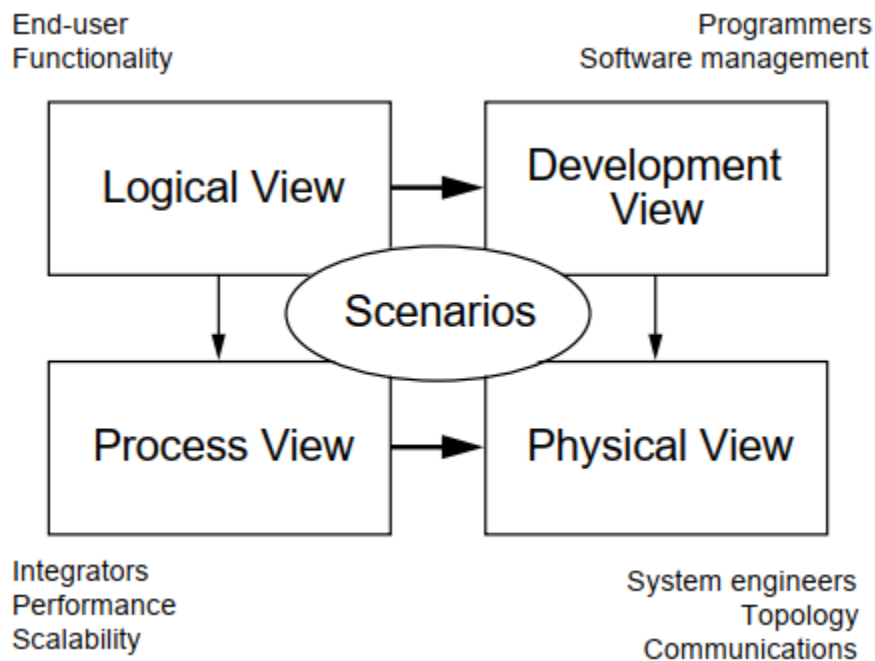
A camada de aplicação compreende a integração de aplicações finais baseadas nas necessidades de apresentação de dados e regras de negócio. Tal domínio pode ser utilizado para prover a possibilidade de um cuidador pessoal, familiar ou responsável profissional, monitorar remotamente o estado dos usuários assistidos no ambiente. Um usuário, autorizado para tal, também pode interagir com o ambiente de forma remota a fim de controlar ou verificar o estado dos dispositivos inteligentes. Além disso, o próprio usuário assistido poderá utilizar uma aplicação em seu *smartphone* para controlar e/ou cadastrar regras de ativação dos dispositivos inteligentes. Essa aplicação também pode passar a funcionar como serviço de localização via GPS quando o usuário deixar o ambiente interno.

Outra possibilidade é a integração com serviços de saúde e/ou emergência a fim de oferecer um serviço de notificação e alerta para eventos críticos, como quedas graves e anomalias severas nos parâmetros de saúde reportados. Profissionais e ambientes prestadores de serviços de saúde poderão também utilizar registros históricos do paciente a fim de auxiliar em um diagnóstico mais preciso. Para que isso seja possível, é necessário a utilização de APIs de integração que utilizem padrões de representação de dados de saúde e registros médicos como os padrões HL7 (*Health Level Seven*) (BENSON, 2012).

5.5 Modelos Abstratos da Arquitetura Proposta

O modelo de visão 4+1 de arquitetura de software, proposto e apresentado inicialmente por Kruchten (1995), foi utilizado neste trabalho para representar as especificações da arquitetura proposta. Esse modelo auxilia na organização e especificação de uma arquitetura de software por meio de quatro visões arquiteturais múltiplas e concorrentes e uma visão de cenários (ou casos de uso) como ilustração de funcionalidades do sistema (Figura 16).

Figura 16 – Modelo de visão 4+1 de arquitetura de software



Fonte: Kruchten (1995)

5.5.1 Visão Lógica

A visão lógica envolve características funcionais que o sistema deve apresentar e prover em termos de serviços aos usuários finais. Geralmente utiliza-se diagramas de classe e/ou de sequência para representar essa visão.

Nesse contexto, a Figura 17 apresenta uma visão lógica da arquitetura por meio de um diagrama de sequência que mostra o principal fluxo de interação entre os componentes. Tal fluxo inclui a leitura dos sensores, o processamento da localização e dos dados dos demais sensores, o acionamento de um dispositivo inteligente, o envio dos dados para o armazenamento em nuvem e o envio de notificações de eventos críticos. Já a Figura 18 apresenta uma visão lógica da arquitetura por meio de um diagrama de classes a nível de domínio, trazendo os conceitos e

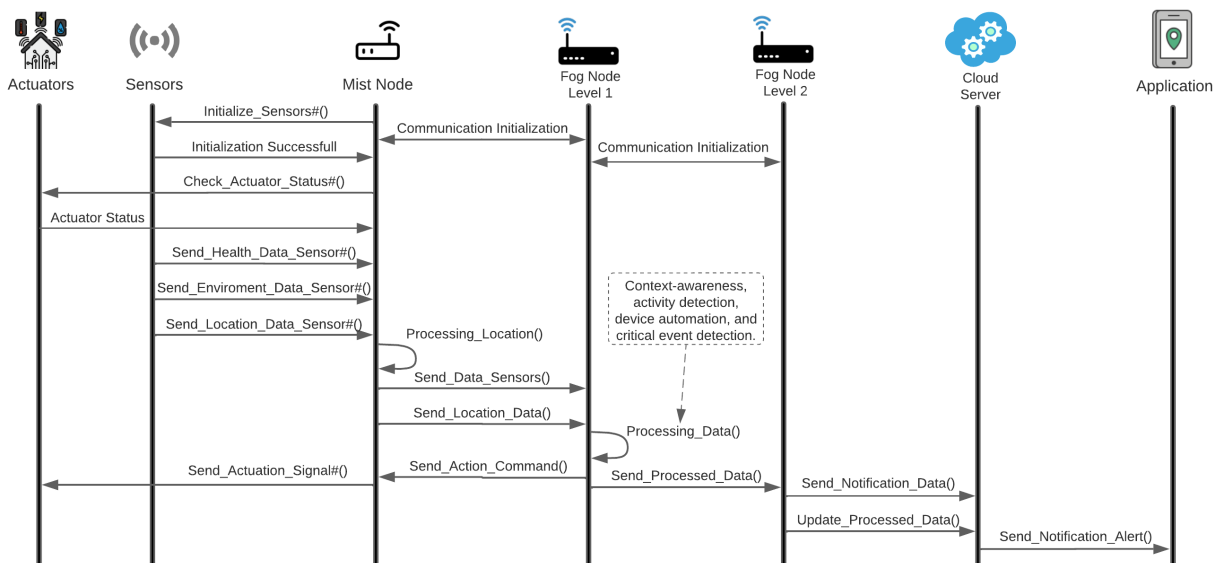
relacionamentos importantes no domínio do sistema.

Cada *mist node* inicializa os sensores e checa os atuadores sob sua responsabilidade, além de estabelecer a comunicação com a camada fog. Após esse processo de preparação dos dispositivos e estabelecimento da comunicação, os dados coletados pelos sensores de saúde, de ambiente e de localização, são coletados pelos *mist nodes*. Na sequência, o *mist node* prepara os dados de localização, realizando um processo de sincronização e agrupamento, e os encaminha junto com os dados dos demais sensores ao dispositivo fog de nível 1.

Por sua vez, o dispositivo fog de nível 1 realiza o processamento dos dados, com base em algoritmos e regras predefinidas (por exemplo, para detecção de eventos críticos, determinação da localização e automação de algum dispositivo inteligente no ambiente). O resultado desse processamento é então encaminhado ao dispositivo fog de nível 2 que tem o papel de manter atualizado os dados contextuais de todos os ambientes sob sua responsabilidade, bem como enviar à camada de nuvem os dados coletados no ambiente assistido.

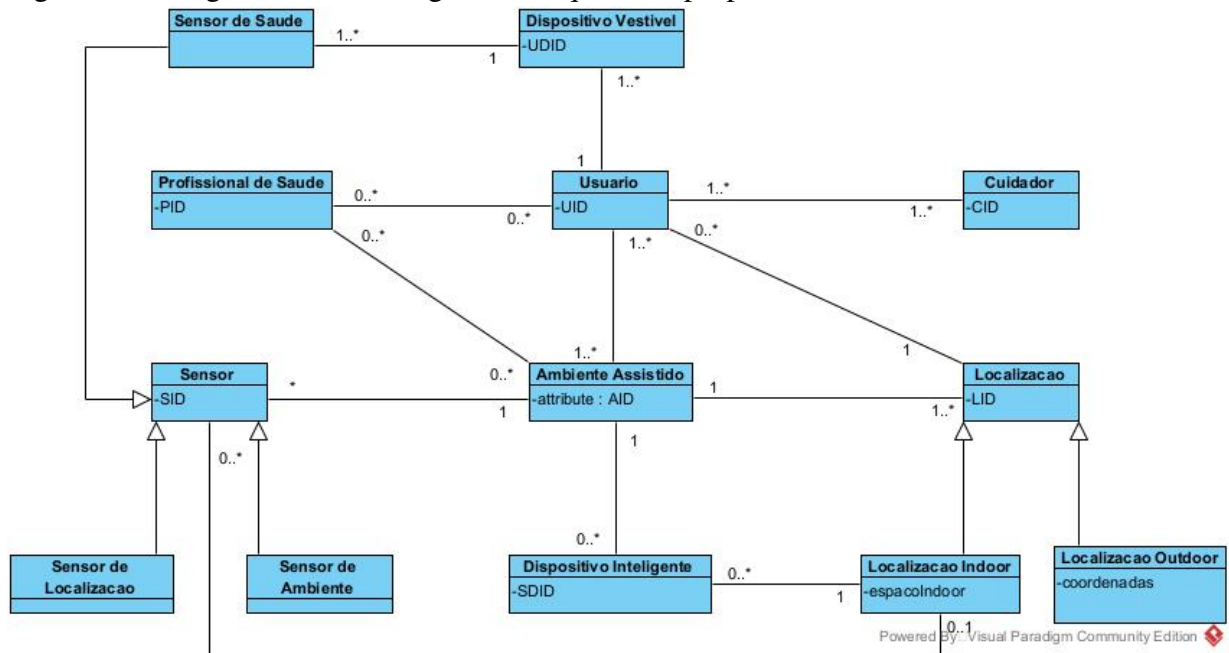
Em caso de detecção de algum evento crítico, o dispositivo fog de nível 2 envia uma requisição com dados da notificação de alerta para a camada de nuvem que, por sua vez, encaminha à aplicação do(s) responsável(is) pelo usuário assistido. Além disso, com base no processamento realizado no dispositivo fog de nível 1, em caso de necessidade de acionamento de algum dispositivo inteligente no ambiente, um sinal de controle é enviado ao *mist node* correspondente.

Figura 17 – Principal diagrama de sequência da arquitetura proposta



Fonte: Elaborado pelo Autor.

Figura 18 – Diagrama de classes geral da arquitetura proposta



Fonte: Elaborado pelo Autor.

5.5.2 Visão de Implementação

A visão de implementação é utilizada para representar o sistema do ponto de visto do desenvolvedor. O foco é na organização dos módulos e componentes de software do sistema, sendo geralmente representada por um diagrama de componentes ou de pacotes.

Para a arquitetura proposta neste trabalho, os componentes de software foram organizados de forma a manter correspondência com a estrutura definida em camadas e apresentada na Figura 15. Assim, a Figura 19 apresenta o diagrama de componentes para a camada de percepção, a Figura 20 apresenta o diagrama de componentes para a camada fog, a Figura 21 apresenta o diagrama de componentes para a camada de nuvem e a Figura 22 apresenta o diagrama de componentes para a camada de aplicação.

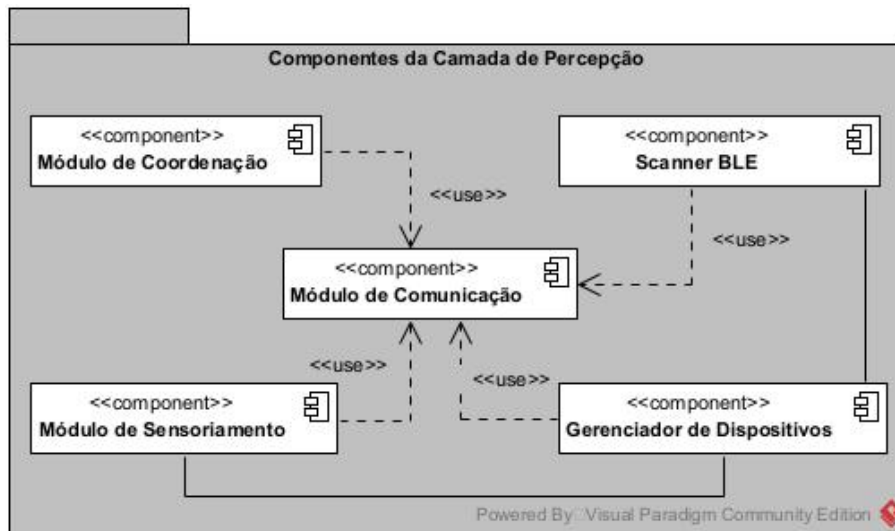
5.5.2.1 Componentes da Camada de Percepção

Na camada de percepção, os seguintes componentes estão presentes, como apresentado na Figura 19:

- a) **Módulo de Comunicação:** responsável por possibilitar a troca de dados entre os dispositivos da camada de percepção. Esse módulo também é responsável por definir o formato de representação utilizados para os dados nesta camada;
- b) **Módulo de Sensoriamento:** responsável por inicializar os sensores conectados

- ao *mist node* e coletar medições periodicamente;
- Gerenciador de Dispositivos:** esse módulo tem o papel de verificar o estado dos dispositivos inteligentes presentes no ambiente e enviar sinais de acionamento quando necessário ativar/desativar algum desses dispositivos;
 - Scanner BLE:** esse módulo é responsável por implementar a lógica necessária de escaneamento do dispositivo final do usuário, utilizado para fins de determinação de sua localização interna;
 - Módulo de Coordenação:** responsável por sincronizar todos os dispositivos da camada de percepção, bem como coordenar a troca de dados entre esta camada e a camada fog.

Figura 19 – Diagrama de componentes da camada de percepção



Fonte: Elaborado pelo Autor.

5.5.2.2 Componentes da Camada Fog

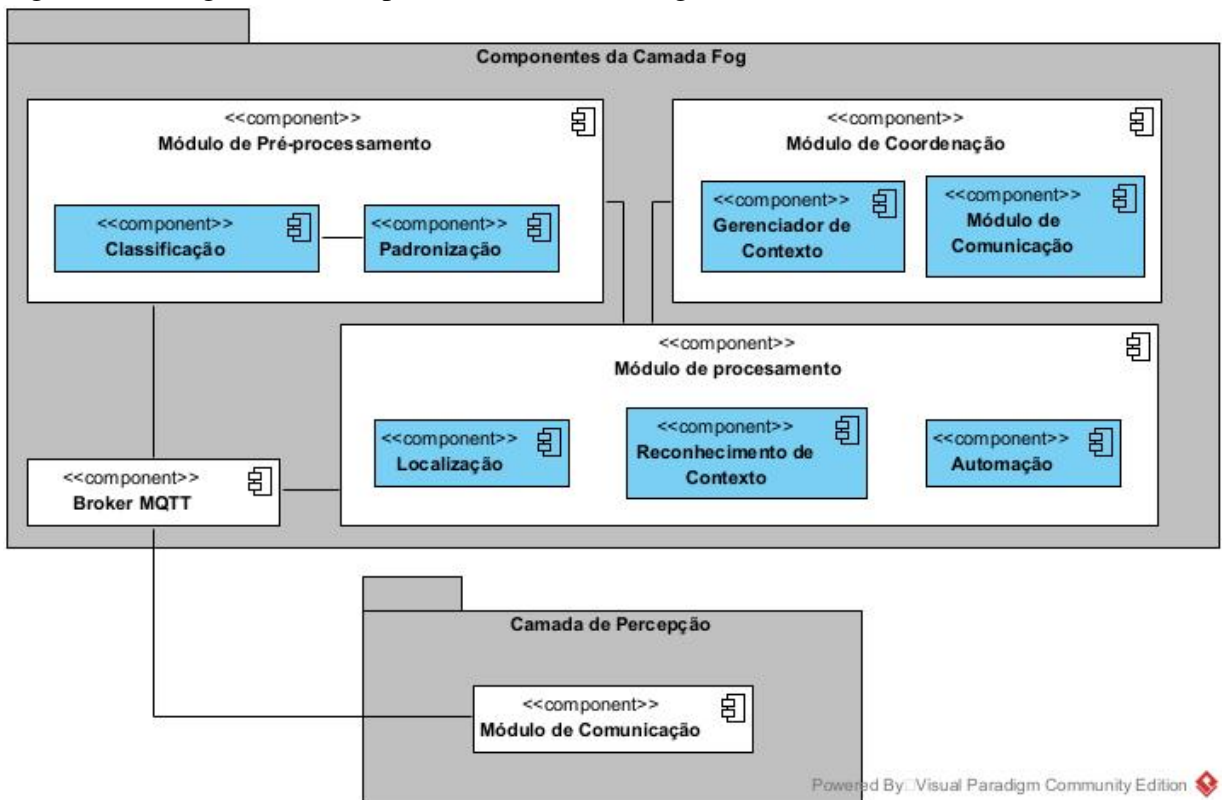
Os componentes da camada fog, apresentados na Figura 20, incluem:

- Broker MQTT:** responsável por intermediar a comunicação entre esta camada e a camada de percepção, através de tópicos definidos para publicação e consumo dos dados que precisam ser trafegados entre essas duas camadas;
- Módulo de Pré-processamento:** responsável por realizar os processos de classificação e padronização dos dados coletados na camada de percepção, utilizando formatos de representação de dados especificados para esta camada;
- Módulo de Processamento:** esse módulo desempenha o papel mais importante

do sistema, onde os dados coletados, classificados e padronizados, são processados a fim de executar serviços como localização, reconhecimento de contexto, automação de dispositivos e detecção de eventos críticos, por exemplo;

- d) **Módulo de Coordenação:** responsável por sincronizar todos os dispositivos da camada fog, bem como coordenar a troca de dados entre esta camada e a camada de nuvem, mantendo atualizados os dados contextuais referentes aos ambientes assistidos sob a responsabilidade de um dispositivo fog de nível 2.

Figura 20 – Diagrama de componentes da camada fog



Fonte: Elaborado pelo Autor.

5.5.2.3 Componentes da Camada de Nuvem

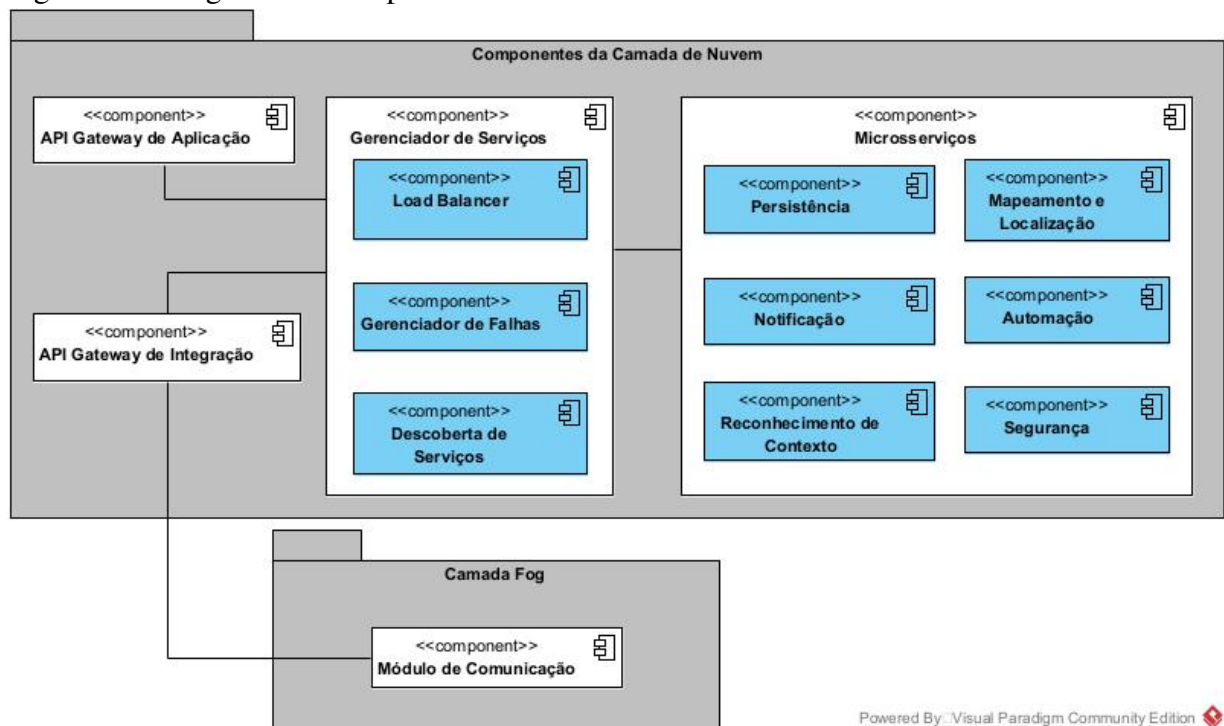
Os componentes da camada de nuvem, apresentados na Figura 21, incluem:

- a) **API de Integração:** esse componente define uma API para integração de diversos ambientes assistidos com a plataforma em nuvem. Essa API deve expor os serviços e dados de forma a garantir métodos de autorização e autenticação dos agentes solicitantes;
- b) **Gerenciador de Serviços:** responsável por manter o gerenciamento dos mi-

crossserviços que compõem o sistema, que inclui: realizar balanceamento de carga entre múltiplas instâncias de um microsserviço; possibilitar o registro e descoberta de novos microsserviços de forma dinâmica e automática; e implementar a estratégia de gerenciamento e recuperação de falhas a fim de manter a confiabilidade do serviço;

- c) **Microsserviços:** conjunto de componentes que especificam e implementam os microsserviços da plataforma em nuvem, como: persistência, mapeamento e localização, notificação, automação, reconhecimento de contexto e segurança;
- d) **API de Aplicação:** esse componente define uma API para integração de diversas aplicações com a plataforma em nuvem, possibilitando diversos tipos de usuários acessarem os serviços oferecidos pelo sistema. Essa API deve expor os serviços e dados de forma a garantir métodos de autorização e autenticação dos agentes solicitantes, bem como utilizar padrões de representação de dados médicos a fim de possibilitar a integração da plataforma com sistemas de saúde existentes.

Figura 21 – Diagrama de componentes da camada de nuvem

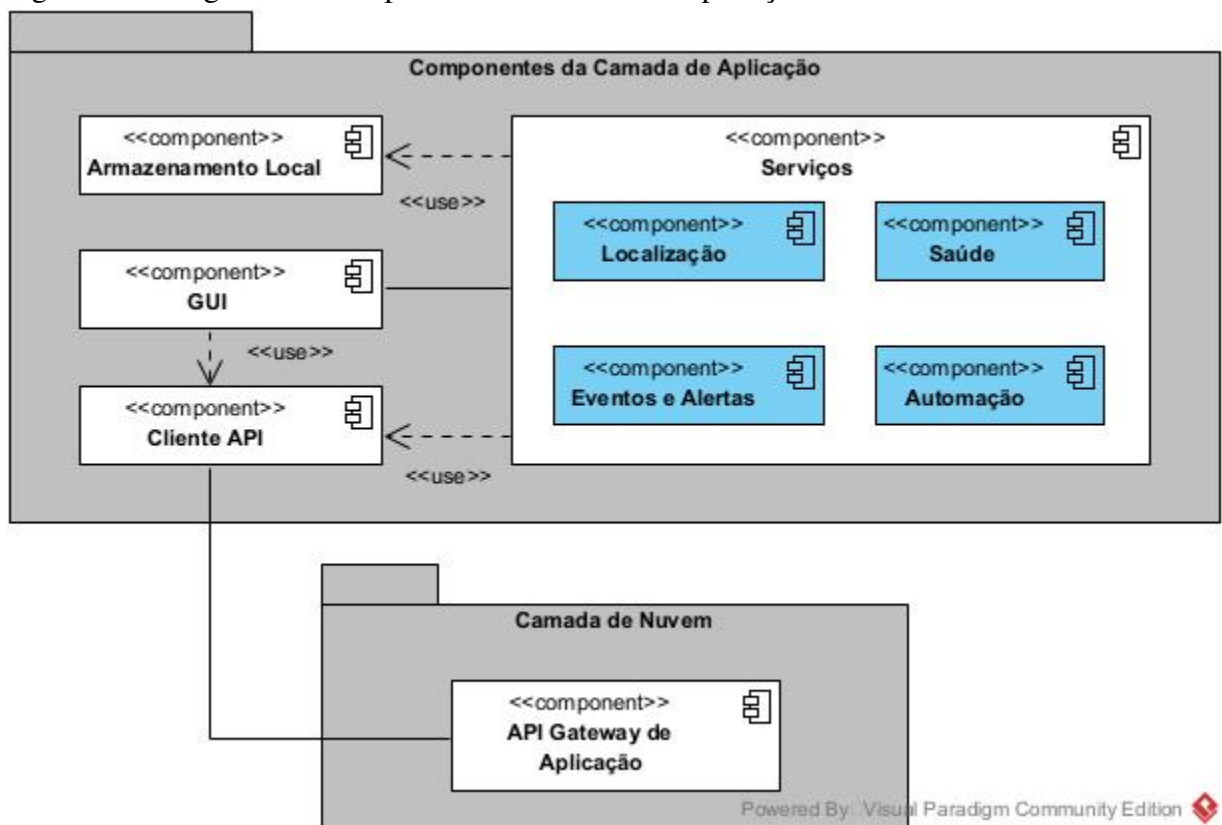


5.5.2.4 Componentes da Camada de Aplicação

Na camada de aplicação, os seguintes componentes estão presentes, como apresentado na Figura 22:

- API Cliente:** esse componente implementa a comunicação com a plataforma em nuvem por meio de requisições à API de aplicação, disponibilizada pela camada de nuvem;
- GUI:** responsável pela apresentação e gerenciamento de eventos dos componentes da interface gráfica da aplicação;
- Armazenamento Local:** responsável por manter dados da aplicação de forma permanente no armazenamento local do dispositivo onde a aplicação será executada;
- Serviços:** conjunto de componentes que implementam as regras de negócio para a disponibilização de serviços ao usuário, como: monitoramento de localização; monitoramento de saúde; recebimento de eventos e alertas; e controle e automação dos dispositivos inteligentes presentes no ambiente assistido.

Figura 22 – Diagrama de componentes da camada de aplicação



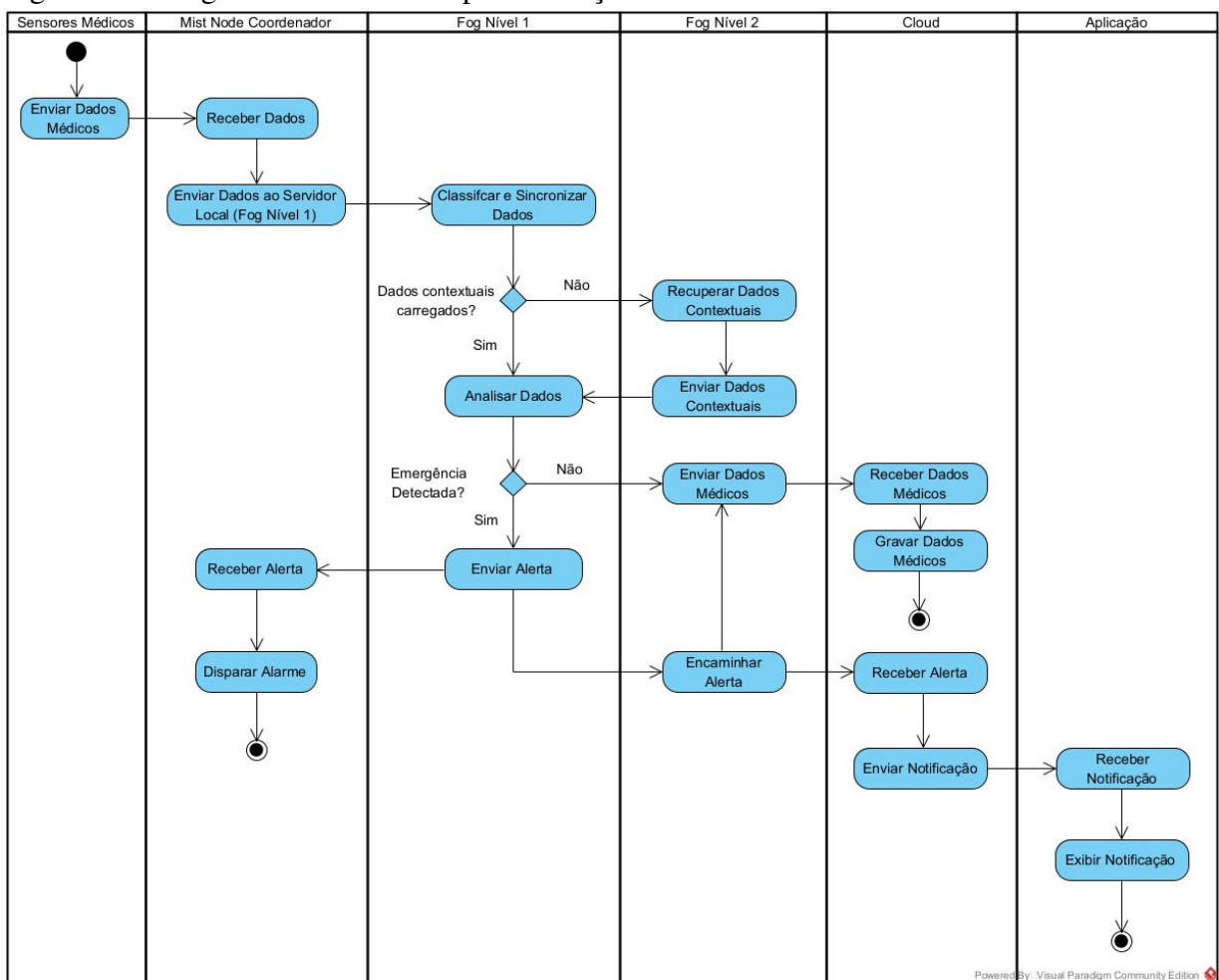
Fonte: Elaborado pelo Autor.

5.5.3 Visão de Processo

O foco da visão de processo é representar o comportamento do sistema, permitindo visualizar seus aspectos dinâmicos e como os processos se comunicam. De forma geral, o diagrama de atividades é utilizado nesta visão arquitetural.

A Figura 23 apresenta o diagrama de atividades para o processo de detecção de eventos críticos relacionados aos parâmetros de saúde do usuário, bem como o envio de notificações de alerta aos responsáveis (cuidadores e profissionais de saúde).

Figura 23 – Diagrama de atividades para detecção de eventos críticos



Fonte: Elaborado pelo Autor.

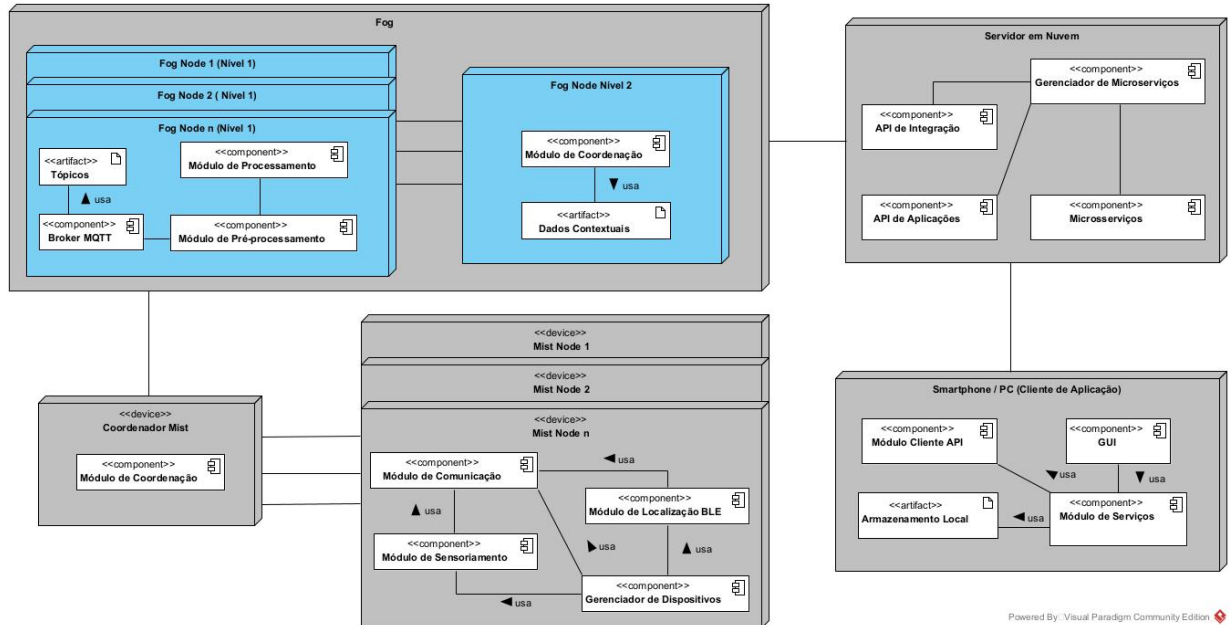
5.5.4 Visão de Implantação

A visão de implantação, ou visão física, da arquitetura, define a topologia física dos componentes de software, bem como a comunicação entre eles. O foco é mapear os componentes de software para os nós de hardware necessários, levando em consideração os requisitos não

funcionais do sistema. O diagrama de implantação é comumente utilizado para representar esta visão.

A Figura 24 mostra a visão de implantação da arquitetura proposta por meio de um diagrama de implantação, onde é possível identificar os nós de hardware necessários, bem como o mapeamento desses nós com os componentes de software do sistema.

Figura 24 – Diagrama de implantação da arquitetura proposta

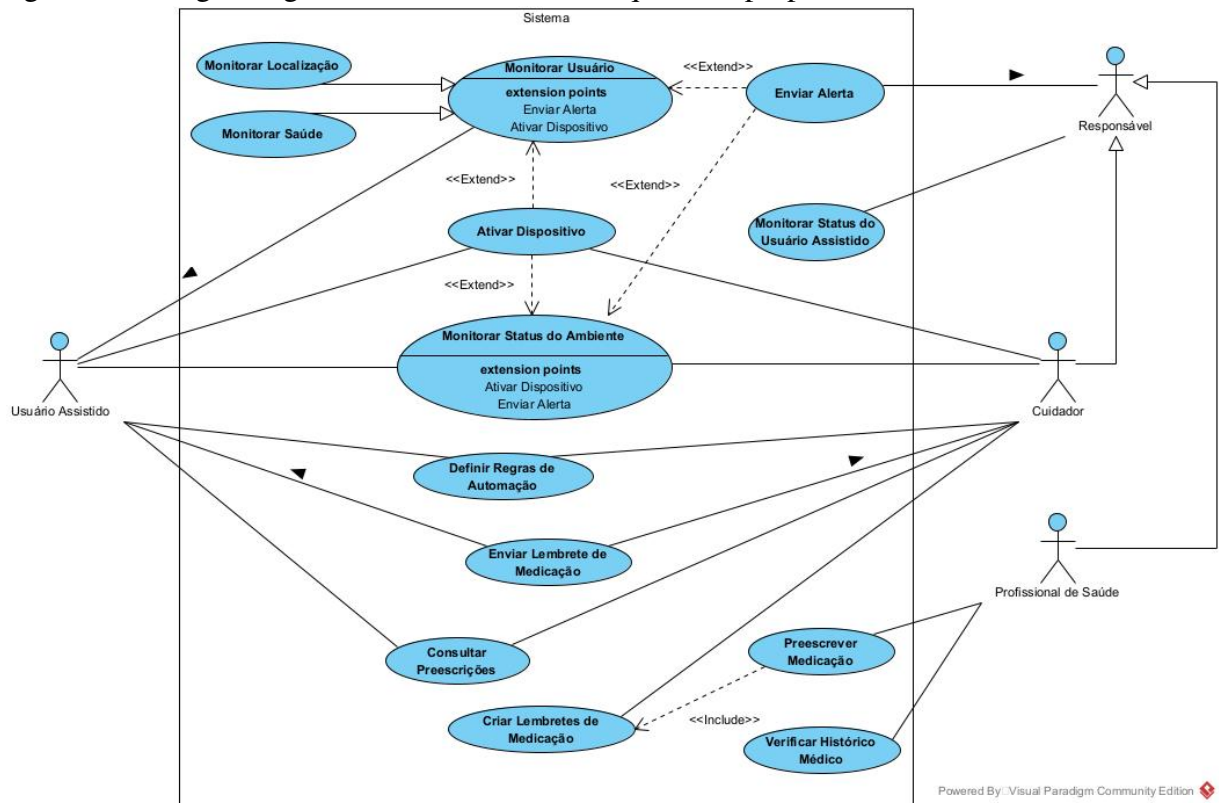


Fonte: Elaborado pelo Autor.

5.5.5 Visão de Cenários

A visão de cenários, ou casos de uso, se configura como o quinto elemento de visão arquitetural. Esta visão é utilizada em conjunto com os elementos das demais visões para descrever sequências de interações entre objetos, componentes e processos em um conjunto de cenários importantes. Com isso, este artefato é útil tanto para descobrir elementos arquiteturais durante o projeto da arquitetura do sistema, quanto para ilustrar e validar a arquitetura após definida. Nesse sentido, a Figura 25 apresenta os casos de uso gerais da arquitetura proposta, trazendo as principais funcionalidades que devem ser oferecidas pelo sistema. Já a Tabela 10 apresenta uma breve descrição dos casos de uso definidos.

Figura 25 – Diagrama geral de casos de uso da arquitetura proposta



Fonte: Elaborado pelo Autor.

Tabela 10 – Descrição resumida dos casos de uso definidos

Código	Nome	Descrição Resumida
UC.01	Monitorar Usuário	O sistema monitora de forma contínua e em tempo real, os dados de saúde e localização do usuário assistido. O usuário deve utilizar um dispositivo vestível (como uma <i>smartband</i> , por exemplo) com conexão BLE e Wi-Fi, para que os dados coletados possam ser encaminhados para o módulo de comunicação e entregues à camada <i>fog</i> . Com base na análise desses dados, alertas podem ser enviados aos responsáveis pelo acompanhamento do usuário assistido.
UC.02	Monitorar Status do Ambiente	O sistema monitora de forma contínua e em tempo real, os dados relativos ao ambiente assistido, por meio dos sensores implantados no local (como sensor de temperatura, sensor de luminosidade, sensor de vazamento de gás, sensor de incêndio, por exemplo). Com base nos dados coletados, a camada <i>fog</i> pode enviar alertas aos responsáveis e/ou comandos aos dispositivos inteligentes presentes no ambiente.

Continua na próxima página

Tabela 10 – *Continuação da tabela*

Código	Nome	Descrição Resumida
UC.03	Enviar Alerta	Ao ser detectado qualquer evento adverso, como um parâmetro de saúde anômalo, um evento de detecção de queda, ou mesmo algum evento crítico relativo ao ambiente (como detecção de incêndio, por exemplo), uma notificação de alerta deve ser enviada aos responsáveis pelo acompanhamento do usuário assistido. Os dados de alerta devem conter todas as informações sobre o evento detectado, bem como a informação de localização (interna ou externa) do usuário. Além disso, o sistema pode manter um histórico de eventos críticos detectados.
UC.04	Ativar Dispositivo	A infraestrutura da camada de percepção deve ser projetada de modo a permitir o controle de dispositivos inteligentes (como lâmpadas e ares-condicionados, por exemplo). O acionamento desses dispositivos segue regras predefinidas pelos usuários assistidos ou cuidadores responsáveis pelo ambiente, com base no monitoramento do ambiente e do status de saúde e localização do usuário assistido. Além disso, o usuário assistido e o cuidador podem utilizar uma aplicação em seus dispositivos móveis, por exemplo, para acionar diretamente o dispositivo inteligente.
UC.05	Verificar Status do Usuário	O status do usuário assistido pode ser verificado pelo cuidador ou pelo profissional de saúde responsável pelo seu acompanhamento. Dados referentes aos parâmetros de saúde e localização coletados pela camada de percepção devem ser exibidos em tempo real, podendo o sistema permitir a exibição de uma série histórica a partir dos registros mantidos no nele.
UC.06	Definir Regras de Automação	Os usuários assistidos e seus respectivos cuidadores podem definir regras de automação para o ambiente onde o sistema está implantado. Dessa forma, os dispositivos inteligentes presentes no ambiente podem ser controlados pelo sistema de acordo com os dados coletados e com as regras previamente definidas.
UC.07	Enviar Lembretes de Medicação	Lembretes de medicação são enviados aos dispositivos móveis dos usuários assistidos e/ou cuidadores com base nos horários de medicação previamente cadastrados no sistema. A solução pode ainda oferecer suporte à integração com sistemas de assistência pessoal virtual para gerenciar esses lembretes.

Continua na próxima página

Tabela 10 – *Continuação da tabela*

Código	Nome	Descrição Resumida
UC.08	Preescrever Medicação	O profissional de saúde responsável pelo acompanhamento do usuário assistido pode, após realização de uma consulta, inserir no sistema sua prescrição de medicamentos para o paciente. Esse caso de uso inclui o caso de uso UC.11, onde podem ser definidos os lembretes de medicação com base nos intervalos preestabelecidos na prescrição médica.
UC.09	Consultar Prescrições	Tanto o usuário assistido quanto o seu cuidador podem consultar as prescrições médicas definidas pelo profissional de saúde através do sistema.
UC.10	Verificar Histórico Médico	O profissional de saúde responsável pelo acompanhamento do usuário assistido tem a possibilidade de consultar o histórico de registros médicos do usuário. Tal registro deve conter informações sobre dados de parâmetros de saúde coletados pelos sensores, eventos críticos e anômalos detectados pelo sistema, bem como sobre as prescrições definidas anteriormente.
UC.11	Criar Lembretes de Medicação	O cuidador responsável pelo acompanhamento do usuário assistido pode criar lembretes de medicação através do sistema, para que seja possível gerar alertas nos horários definidos, bem como manter um controle sobre o cumprimento das prescrições médicas. Além disso, esse caso de uso também é executado no momento da definição de uma prescrição médica realizada pelo profissional de saúde responsável pelo acompanhamento do usuário assistido.

Fonte: Elaborado pelo Autor.

6 AVALIAÇÃO DE DESEMPENHO DA ARQUITETURA PROPOSTA

A arquitetura proposta foi avaliada por meio de simulações com o simulador iFogSim, de acordo com as políticas indicadas em Gupta *et al.* (2017). O iFogSim é um kit de ferramentas de simulação para soluções de IoT que usam o paradigma de *fog computing*. Ele permite a modelagem de aplicativos e ambientes fog, bem como a análise do desempenho da solução em termos de utilização de recursos (GUPTA *et al.*, 2017). Além disso, a fim de avaliar a viabilidade da arquitetura em termos de descrição dos componentes e RNFs para sistemas IPS, um experimento de localização interna em ambiente real de operação foi conduzido, sendo apresentado na Seção 6.1. A configuração experimental da simulação, bem como os resultados obtidos e as discussões sobre eles, são descritos nas Seções 6.2 e 6.3, respectivamente.

6.1 Experimento de Localização Interna

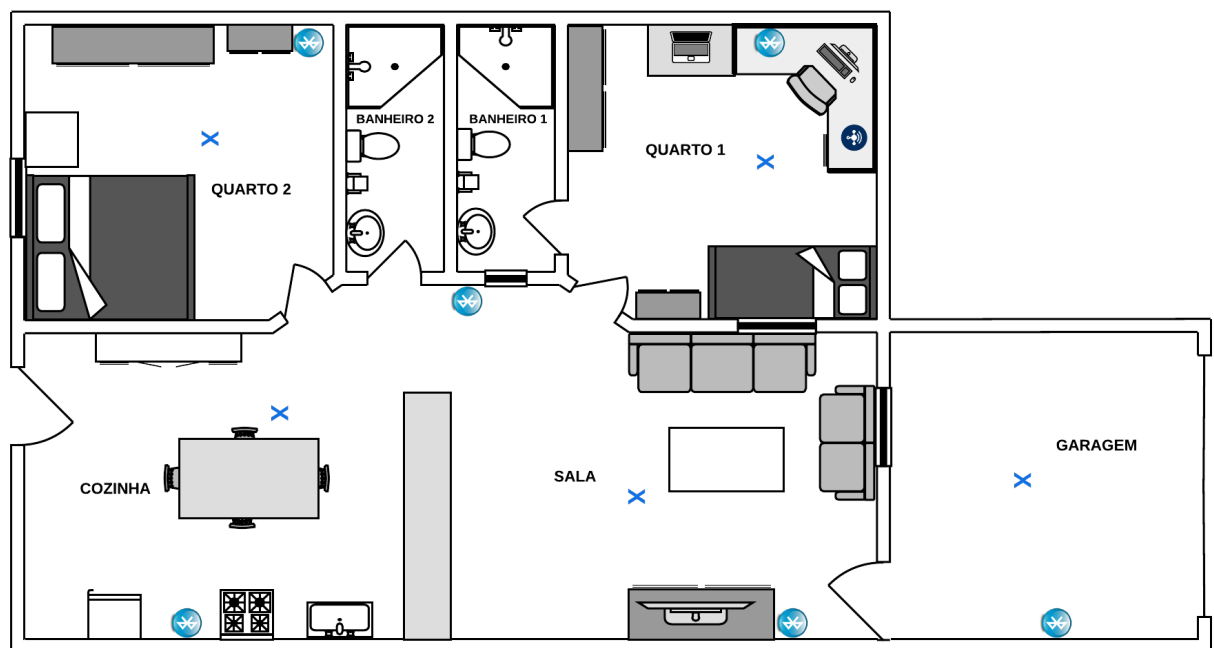
A fim de realizar uma validação inicial da decisão de projeto DP-03 (ver Tabela 9), localização centrada na camada fog, um experimento foi conduzido em um ambiente correspondente a uma pequena residência com dimensão total de $57 m^2$. A estratégia utilizada baseia-se no trabalho de Du *et al.* (2018), que transfere a responsabilidade pela determinação da localização para os pontos de referência (PR) presentes no ambiente. Dessa forma, o dispositivo final do usuário não tem mais a responsabilidade de escanear, continuamente, sinais dos pontos de referência, nem de realizar o processamento necessário para a determinação de sua própria localização.

Vale ressaltar que essa decisão de projeto foi pensada a fim de diminuir o consumo energético do dispositivo final do usuário, prologando seu tempo de bateria e possibilitando sua utilização para outras atividades como, por exemplo, o monitoramento contínuo de parâmetros de saúde. Contudo, é importante notar que isso adiciona uma complexidade extra à rede e às camadas superiores. Com a descentralização da atividade de processamento de localização interna, há a necessidade de manter uma sincronia precisa entre todos os dispositivos da rede de escaneamento (pontos de referência). Embora haja essa condição, entende-se que para o domínio do problema e objetivos apresentados neste trabalho, essa decisão de projeto se mostra mais adequada.

Foram utilizados seis PRs de forma distribuída no ambiente interno, realizando o papel de *scanners* BLE. Um *smartwatch* com conectividade BLE foi utilizado como dispositivo

vestível portado pelo usuário a ser localizado. Nesse caso, o *smartwatch* é programado como um *beacon* BLE a fim de enviar, periodicamente, pacotes de anúncios que serão escaneados pelos pontos de referência distribuídos na residência. A Figura 26 apresenta a planta baixa do ambiente utilizado na aplicação do experimento. São sete cômodos ao total, sendo dois quartos, dois banheiros, uma cozinha, uma sala de estar e uma garagem. Os ícones em azul com o símbolo da tecnologia *bluetooth* representam a localização dos *scanners* BLE distribuídos no ambiente, todos posicionados a uma altura de 1,5 m do piso.

Figura 26 – Ambiente de aplicação do experimento



Fonte: Elaborado pelo Autor.

A técnica de localização baseada em *fingerprinting* BLE foi utilizada na aplicação deste experimento. Os pontos marcados com um “X” em azul na Figura 26 representam os pontos de coleta nas fases de treino (*offline*) e teste (*online*) do *fingerprinting*. Na fase *offline* foram realizadas 100 (cem) coletas em cada um dos locais definidos, com um intervalo de tempo t , inicialmente definido em cinco segundos, entre cada uma. Na fase *online* foram realizadas 48 (quarenta e oito) coletas, com o mesmo intervalo t entre cada uma.

Cada um dos seis PRs, no intervalo de tempo definido, realizou uma quantidade k de leituras do valor RSSI do dispositivo *beacon* e calculou uma média desses valores. Essa abordagem foi utilizada na tentativa de minimizar possíveis erros inseridos por leituras anômalas. Ainda, foram descartadas quaisquer leituras RSSI com valor zero a fim de manter uma maior coerência no valor médio das leituras, visto que verificou-se empiricamente uma considerável

ocorrência de leituras com valor 0 para o RSSI do dispositivo ¹. Assim, da quantidade k de leituras, uma quantidade m de leituras diferentes de zero são consideradas. Dessa forma, o valor RSSI considerado em cada PR, dentro do intervalo de tempo t definido, é dado por:

$$\frac{\sum_{i=1}^k RSSI_i}{m}, \quad \forall RSSI_i \neq 0 \quad (6.1)$$

Após esse processo, cada PR (dispositivo *scanner* BLE) encaminha o valor RSSI médio, coletado em um determinado intervalo de leituras, para o módulo coordenador, que se encarrega de sincronizar os pacotes recebidos de cada ponto de referência e criar o vetor de medições $RSSI_v = [RSSI_1, \dots, RSSI_{l-1}, RSSI_l]$, onde l representa a quantidade de pontos de referência presentes no ambiente.

Para a implementação do cenário de experimento descrito nesta Seção, foram utilizadas sete placas de prototipação IoT ESP32, com conectividade Wi-Fi e BLE, sendo seis *scanners* e um dispositivo coordenador. A Figura 27 apresenta uma visão geral dos componentes do sistema e de sua comunicação.

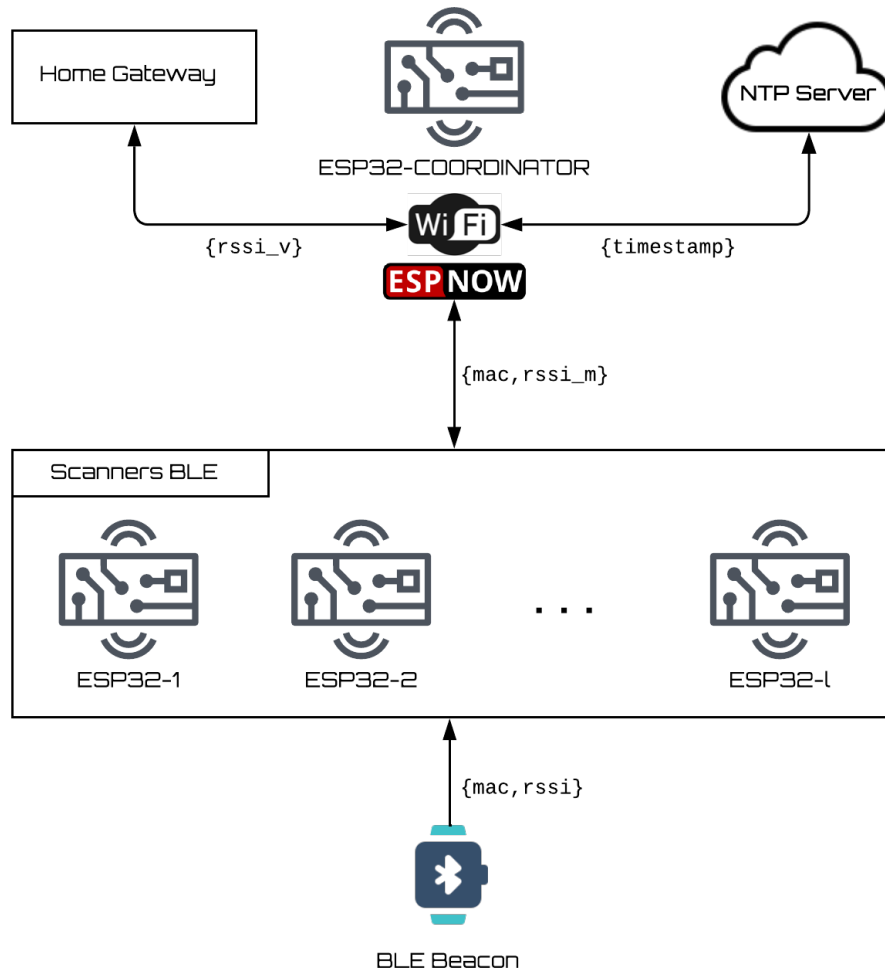
Os pacotes de anúncio BLE contêm dados relativos ao endereço físico do dispositivo vestível utilizado (endereço *mac*) e o valor de RSSI. Por sua vez, os *scanners* BLE leem tais informações e encaminham o valor RSSI médio ($rssi_m$), dentro do intervalo de coleta t , junto ao seu endereço *mac* para o dispositivo coordenador. Essa comunicação se dá por meio do protocolo ESP-NOW². O dispositivo coordenador, por sua vez, recebe as coletas individuais de cada *scanner* e cria o vetor de medições RSSI ($rssi_v$), encaminhando-o para um servidor local (*home gateway*) responsável por executar o processamento dos dados, seja na fase *offline* ou na fase *online* do processo de localização. Além disso, o dispositivo coordenador se conecta a um servidor NTP, a fim sincronizar os relógios de todos os dispositivos *scanners*.

A Figura 28 apresenta o diagrama de sequência com o fluxo de comunicações e interações entre os componentes para realizar as coletas de leituras RSSI, bem como a sincronização dos dispositivos. O dispositivo *scanner*, ao iniciar sua operação, envia uma requisição de registro ao dispositivo coordenador com o seu endereço *mac*. O dispositivo coordenador, por sua vez, verifica se o dispositivo *scanner* solicitante está autorizado a ingressar na rede, via requisição

¹ Isso pode ser explicado pela ocorrência de variações repentinas no ambiente (como obstáculos móveis, por exemplo) ou no posicionamento do dispositivo alvo em relação aos PRs.

² O ESP-NOW é um protocolo de comunicação sem fio definido pela Espressif, que permite o controle de dispositivos inteligentes de forma rápida, direta e com baixo consumo de energia.

Figura 27 – Visão geral dos componentes do experimento de localização interna



Fonte: Elaborado pelo Autor.

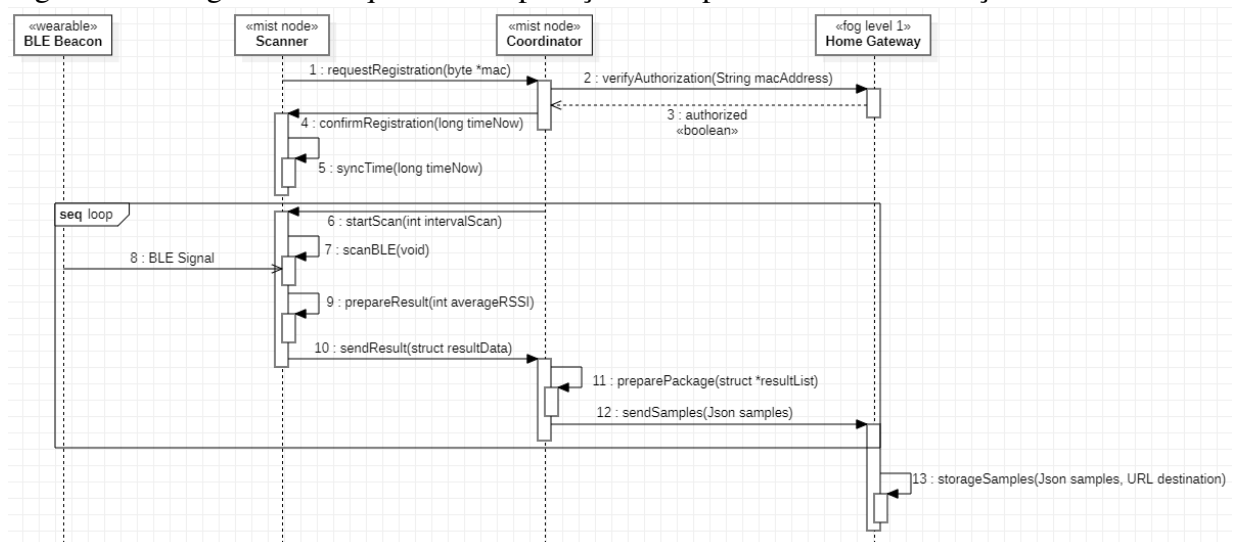
ao servidor local (*home gateway*), que armazena uma lista de controle de acesso baseado nos endereços *mac* dos dispositivos autorizados.

Ao ser confirmado sua autorização de ingresso na rede de *scanners* BLE do ambiente, o dispositivo solicitante recebe o instante de tempo atual (sincronizado via NTP), para garantir a sincronia dos relógios entre todos os *scanners*. A partir do momento em que todos os *scanners* estão em operação, o dispositivo coordenador envia um sinal de controle para início do processo de leituras RSSI, especificando o intervalo de tempo da janela de escaneamento. Assim, os dispositivos *scanners* realizam uma quantidade k de leituras RSSI, com um segundo de espera entre cada leitura, dentro do intervalo t definido. Logo, para um intervalo de escaneamento de cinco segundos, serão realizadas cinco leituras RSSI do dispositivo a ser localizado.

Ao final da janela de escaneamento, cada *scanner* calcula a média de valores RSSI lidos, seguindo a Equação 6.1, como explicado acima, e envia os dados lidos ao módulo coordenador. Por sua vez, o módulo coordenador recebe e sincroniza todos os pacotes de dados

dos dispositivos *scanners* e prepara o vetor de medições RSSI, enviando-o ao *home gateway*. Por sua vez, o *home gateway* realiza o processo de treinamento da técnica *fingerprinting* ou, na fase *online*, realiza o processo de determinação da localização a nível de cômodo a partir do modelo treinado na fase *offline*.

Figura 28 – Diagrama de sequência da aplicação do experimento de localização interna



Fonte: Elaborado pelo Autor.

A fim de abstrair detalhes de baixo nível entre as camadas e possibilitar uma representação comum para os dados capturados e que desempenham papéis importantes no processo de localização, foi definido um formato de representação padrão (FRP) para os dados produzidos em cada camada do sistema. A Tabela 11 apresenta o FRP utilizado para o escaneamento BLE, a Tabela 12 apresenta o FRP utilizado pelo dispositivo coordenador para comunicação com os dispositivos *scanners*, enquanto a Tabela 13 apresenta o FRP utilizado para transmitir dados ao *home gateway*.

Para a avaliação da estratégia de localização aplicada no experimento, foram utilizados três modelos de aprendizagem de máquina baseados nos algoritmos KNN (*K - Nearest Neighbors*), SVM (*Support Vector Machine*) e RF (*Random Forest*). O treinamento de cada modelo se deu por meio dos dados coletados na fase *offline* do *fingerprinting*, como explicado acima. Para a fase *online*, foram avaliados dois cenários principais: usuário parado dentro do cômodo e usuário em movimento dentro do cômodo. Os resultados foram então analisados e comparados levando em consideração as métricas de acurácia e F1-score.

A Tabela 14 apresenta os resultados de acurácia e F1-score para cada modelo, em cada cenário aplicado. Com base nos resultados obtidos, pode-se perceber que uma acurácia de

Tabela 11 – FRP utilizado pelos dispositivos *scanners*

Nome do Campo	Descrição
<i>anchor_mac</i>	Endereço <i>MAC</i> do dispositivo <i>scanner</i>
<i>anchor_description</i>	Descrição do dispositivo <i>scanner</i>
<i>message_type</i>	Informação sobre tipo da mensagem: requisição de registro na rede, envio de coleta RSSI para treinamento, envio de coleta RSSI para localização <i>online</i> (<i>registration_request=0, data_loc_offline=1, data_loc_online=2</i>)
<i>scanner_data</i>	Dados relativos ao escaneamento do dispositivo final a ser localizado
- <i>device_mac</i>	- Endereço <i>MAC</i> do dispositivo escaneado
- <i>device_name</i>	- Nome vinculado ao dispositivo escaneado
- <i>rss_i_m</i>	- Média de valores RSSI lidos na janela de escaneamento
- <i>scan_window</i>	- Janela de escaneamento definida
- <i>number_samples</i>	- Quantidade de amostras válidas (<i>rss_i ≠ 0</i>)
- <i>start_time</i>	- Instante de tempo de início do escaneamento
- <i>stop_time</i>	- Instante de tempo de término do escaneamento

Fonte: Elaborado pelo Autor.

Tabela 12 – FRP utilizado pelo dispositivo coordenador para comunicação com os *scanners*

Nome do Campo	Descrição
<i>coordinator_mac</i>	Endereço <i>MAC</i> do dispositivo coordenador
<i>message_type</i>	Informação sobre tipo da mensagem: resposta à requisição de registro na rede, envio de sinal de início de janela de escaneamento (<i>registration_reply=0, start_scan=1</i>)
<i>coordinator_data</i>	Dados enviados ao dispositivo <i>scanner</i>
- <i>time_now</i>	- Instante de tempo atual, obtido via NTP
- <i>scan_window</i>	- Janela de escaneamento a ser utilizada
- <i>anchor_description</i>	- Descrição atribuída ao <i>scanner</i> , com base em seu endereço <i>MAC</i>

Fonte: Elaborado pelo Autor.

Tabela 13 – FRP utilizado pelo dispositivo coordenador para comunicação com o *home gateway*

Nome do Campo	Descrição
<i>coordinator_mac</i>	Endereço <i>MAC</i> do dispositivo coordenador
<i>start_time</i>	Instante de tempo de início da janela de escaneamento
<i>stop_time</i>	Instante de tempo de término da janela de escaneamento
<i>place</i>	Local (cômodo) onde as coletas foram realizadas (para treinamento)
<i>samples[]</i>	Vetor de coletas RSSI realizadas pelos <i>scanners BLE</i>
- <i>anchor_mac</i>	- Endereço <i>MAC</i> do dispositivo <i>scanner</i>
- <i>device_mac</i>	- Endereço <i>MAC</i> do dispositivo escaneado
- <i>rss_i_m</i>	- RSSI médio do dispositivo escaneado

Fonte: Elaborado pelo Autor.

100%, na determinação do cômodo em que o usuário está, foi alcançada utilizando os modelos baseados em KNN e SVM para o cenário onde o usuário não está em movimento pelo cômodo. Já no cenário onde o usuário está em movimento, o modelo baseado no algoritmo *Random Forest*

obteve uma acurácia significativamente maior comparado aos modelos baseados em KNN e SVM.

Ainda, observa-se que mesmo não sendo o melhor modelo para o cenário sem movimento, o *Random Forest* possibilitou uma acurácia muito próxima aos valores de acurácia obtidas pelos modelos baseados em KNN e SVM, de 99%. Em relação à métrica F1-score, é possível perceber que, em todos os casos, ela acompanhou ou mesmo superou o valor de acurácia. Isso torna a métrica de acurácia confiável para analisar o desempenho dos modelos.

Tabela 14 – Comparação de desempenho na determinação do cômodo correto

Cenário	Acurácia			F1-score		
	KNN	SVM	RF	KNN	SVM	RF
Usuário parado no cômodo	1,00	1,00	0,99	1,00	1,00	0,99
Usuário em movimento no cômodo	0,65	0,66	0,87	0,67	0,67	0,90

Fonte: Elaborado pelo Autor.

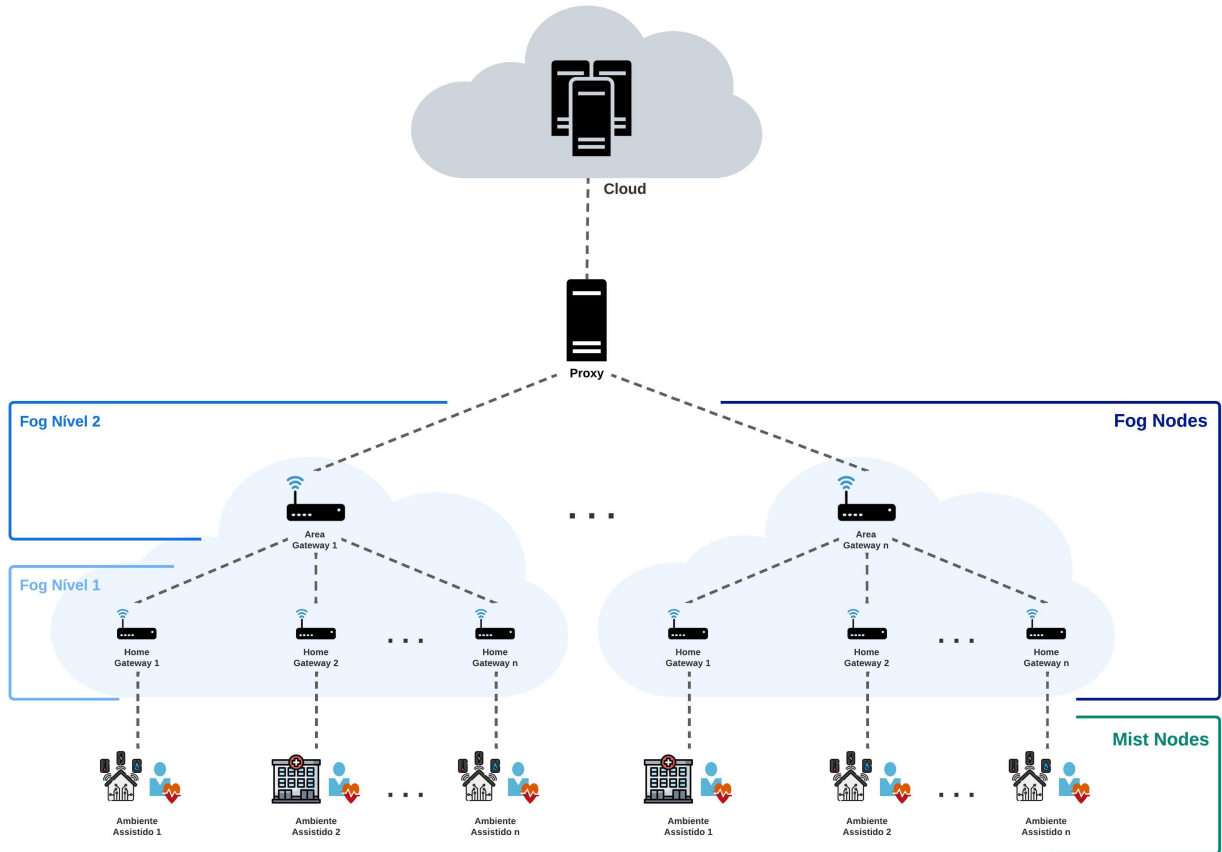
Por fim, vale ressaltar que o experimento descrito nesta Seção teve a finalidade de implementar, de forma aderente à arquitetura proposta e apresentada no Capítulo 5, um serviço de localização interna em um cenário real de operação. Assim, não foram aplicadas nenhuma técnica de correção e minimização de erros, ou fusão de sensores, que pudesse melhorar a qualidade dos dados coletados. O objetivo foi demonstrar a viabilidade da arquitetura proposta em termos de descrição dos componentes necessários para a implementação de um sistema IPS, satisfazendo a decisão de projeto DP-03 (ver Tabela 9). Além disso, o conceito de formatos de representação padrão (FRP) foi introduzido e auxiliou a obter dados reais utilizados na simulação realizada para avaliar a arquitetura proposta por meio da ferramenta iFogSim.

6.2 Projeto de Experimento da Simulação com iFogSim

Para a simulação, definiu-se um cenário envolvendo um sistema AAL para *smart cities*, no qual vários ambientes assistidos podem ser gerenciados (por exemplo, residências, hospitais, casas de repouso), como representado pela Figura 29. Cada ambiente contém uma coleção de sensores e atuadores que são vinculados a *mist nodes*, que são então vinculados a um *gateway* de ambiente (*home gateway* - fog nível 1). Um *gateway* de área (*area gateway* - fog nível 2) gerencia um conjunto de ambientes geograficamente próximos. Por sua vez, os *gateways* de área se comunicam com o servidor de nuvem para atualizar os dados sobre os ambientes

assistidos, fornecer notificações de alerta e permitir a interação remota com o ambiente por meio de uma aplicação.

Figura 29 – Sistema AAL em um cenário de *Smart City*



Fonte: Elaborado pelo Autor.

As variáveis definidas no experimento incluem o número de *mist nodes* por ambiente, o número de ambientes por área e o número total de áreas que o sistema cobre. A quantidade de *mist nodes* no ambiente define quantos sensores e atuadores estão vinculados ao sistema, o que, por sua vez, influencia na quantidade de dados gerados. É possível analisar a capacidade de gerenciamento do *gateway* de área e seu custo de processamento associado alterando o número de ambientes por área. Por fim, ao variar o número de áreas, podemos analisar o fornecimento de serviços de forma hierárquica e geograficamente distribuída.

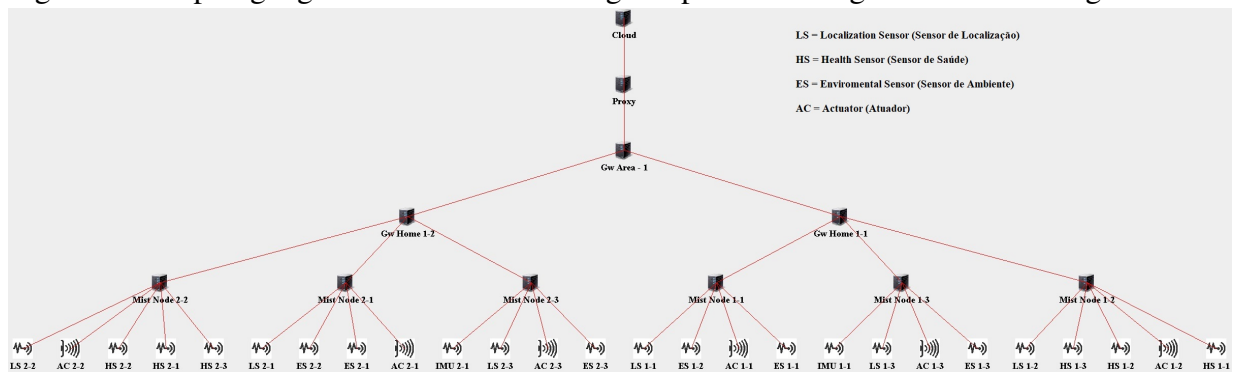
O cenário de simulação inicial consiste em uma área de gerenciamento responsável por quatro ambientes assistidos, que, por sua vez, contém três *mist nodes*. Cada *mist node* controla um total de três sensores e um atuador. Em seguida, o número de ambientes por área e o número total de áreas cobertas variaram. As mesmas configurações foram aplicadas em duas variações com relação ao número de *mist nodes*: três *mist nodes* por ambiente e seis *mist nodes* por ambiente. Essa configuração permitiu avaliar o desempenho do sistema em relação a

diferentes quantidades de dados produzidos pelo ambiente assistido.

Todas essas configurações de cenários de simulação também foram aplicadas em uma abordagem em que os serviços do sistema são fornecidos em uma infraestrutura somente em nuvem. O objetivo é avaliar e comparar o desempenho do sistema em uma abordagem baseada em *fog computing* (arquitetura proposta) e uma abordagem somente em nuvem. A avaliação foi feita em termos de latência, uso da rede, consumo de energia e custo de execução na nuvem. Essas métricas foram definidas para analisar inicialmente os requisitos de desempenho (por meio de resultados de latência e uso da rede), eficiência energética (por meio de resultados de consumo de energia), custo computacional (por meio de resultados de custo de execução na nuvem) e a escalabilidade (por meio de resultados observados ao variar o número de ambientes e dispositivos conectados).

Inicialmente, a topologia do sistema foi estabelecida no iFogSim, configurando os dispositivos de rede, personalizando-os para o cenário apresentado anteriormente, como pode ser observado na Figura 30. A Tabela 15 apresenta os parâmetros configurados para o servidor de nuvem, para o proxy, para os *fog nodes* de nível 1 (*home gateway*) e nível 2 (*area gateway*) e para os *mist nodes*, como indicado em Gupta *et al.* (2017). Um dispositivo proxy foi usado na simulação para conectar os *gateways* de área ao servidor de nuvem, também de acordo com Gupta *et al.* (2017).

Figura 30 – Topologia genérica definida no iFogSim para a estratégia baseada em Fog



Em seguida, foi necessário definir um modelo de aplicação para ser utilizado na simulação. O modelo de aplicação foi projetado utilizando um modelo de Fluxo de Dados Distribuído / *Distributed Data Flow* (DDF), por meio de um Grafo Acíclico Dirigido / *Directed Acyclic Graph* (DAG). Os vértices do DAG representam os módulos que executam o processamento de dados de entrada, e as bordas denotam a dependência de dados entre os módulos (tuplas)

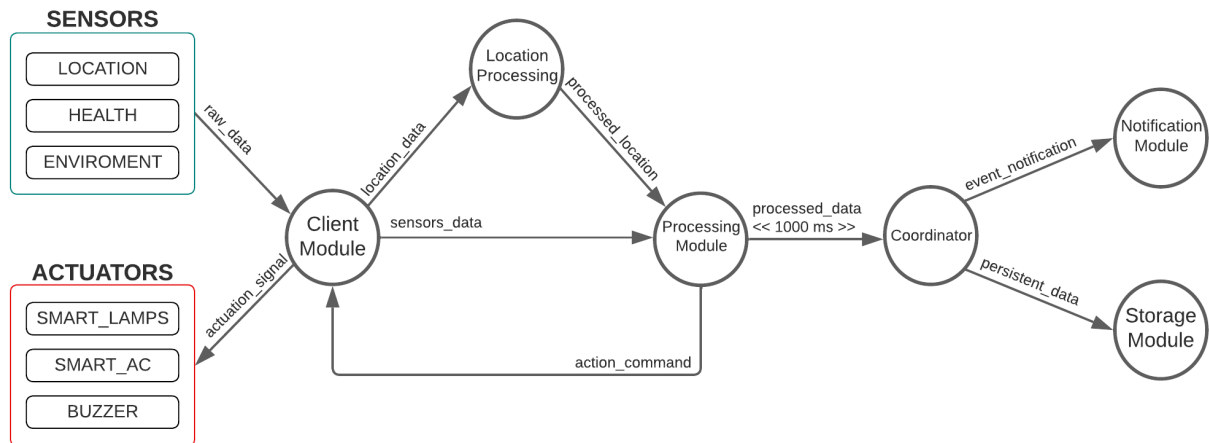
Tabela 15 – Valores dos parâmetros dos dispositivos para a simulação

Parameter	Cloud	Proxy	Area Gateway	Home Gateway	Mist Node
CPU length (MIPS)	89600	11200	11200	2800	700
RAM (MB)	40000	8000	8000	4000	250
Uplink bandwidth (MB)	100	10000	10000	10000	1000
Downlink bandwidth (MB)	10000	10000	10000	10000	1000
Device Level	0	1	2	3	4
Cost/MIPS (\$)	$1 \cdot 10^{-7}$	0.0	0.0	0.0	0.0
Busy power (Watt)	32*103	4*107.339	4*107.339	107.339	87.53
Idle power (Watt)	32*83.25	4*83.43	4*83.43	83.43	82.44

Fonte: Elaborado pelo Autor.

(GUPTA *et al.*, 2017; MAHMUD *et al.*, 2022). O modelo de aplicação definido para a simulação consiste em seis módulos, como pode ser observado na Figura 31: módulo cliente, módulo de processamento de localização, módulo de processamento, coordenador, módulo de notificação e módulo de persistência. O Apêndice B apresenta um maior detalhamento dos parâmetros configurados para a aplicação simulada, utilizando a API Java disponibilizada pelo iFofSim.

Figura 31 – Modelo de aplicação utilizado na simulação



Fonte: Elaborado pelo Autor.

Resumidamente, o módulo cliente é responsável por coletar dados dos sensores e enviar sinais de comando para os atuadores. O módulo de processamento de localização é responsável por agregar e sincronizar os dados de localização, RSSI recebido em cada *mist node*, por exemplo, coletados na camada de percepção e enviá-los à camada fog. O módulo de processamento recebe os dados dos sensores de ambiente e de saúde, bem como os dados de localização do usuário. Em seguida, executa regras de negócios para, por exemplo, detectar atividades e eventos críticos e acionar algum dispositivo inteligente com base na localização do

usuário.

O módulo coordenador recebe periodicamente os dados do módulo de processamento e os encaminha para o módulo de persistência e para o módulo de notificação, quando aplicável. O módulo de notificação recebe uma informação do módulo coordenador com dados referentes ao evento detectado, à localização do usuário e aos seus dados de saúde. Por fim, o módulo de persistência recebe os dados do módulo coordenador e os armazena na nuvem.

Na estratégia baseada em fog, o mapeamento adotado para a alocação dos módulos foi o seguinte: módulos cliente e processamento de localização alocados nos *mist nodes*, módulo de processamento alocado no *gateway* de ambiente, módulo coordenador alocado no *gateway* de área e módulos de notificação e persistência alocados na nuvem. Na configuração baseada somente em nuvem, todos os módulos são alocados na camada de nuvem, exceto o módulo do cliente, que permanece alocado nos *mist nodes*.

6.3 Resultados da Simulação com iFogSim

Como supracitado, a avaliação experimental da arquitetura proposta foi feita por meio de simulações, utilizando a ferramenta iFogSim, em um cenário de *smart city*. Foram aplicadas diversas configurações no cenário de avaliação, variando o número de áreas cobertas pelo sistema, o número de ambientes por área e o número de *mist nodes* em cada ambiente, como mostra a Tabela 16. Os resultados obtidos foram analisados em termos de latência de comunicação, uso de rede, consumo energético, custo de operação e uso de CPU em nuvem.

Com isso, a quantidade total de sensores utilizados nas simulações variou de 160 sensores (2 áreas, 8 ambientes e 3 *mist nodes*) a 20.480 sensores (16 áreas, 64 ambientes e 6 *mist nodes*). Optou-se por apresentar na sequência os resultados obtidos nas configurações do cenário de avaliação com 64 ambientes por área, variando o número de áreas e com três e seis *mist nodes*. Os resultados completos para todos os cenários de avaliação estão disponíveis no Apêndice C.

Tabela 16 – Variações no cenário de avaliação

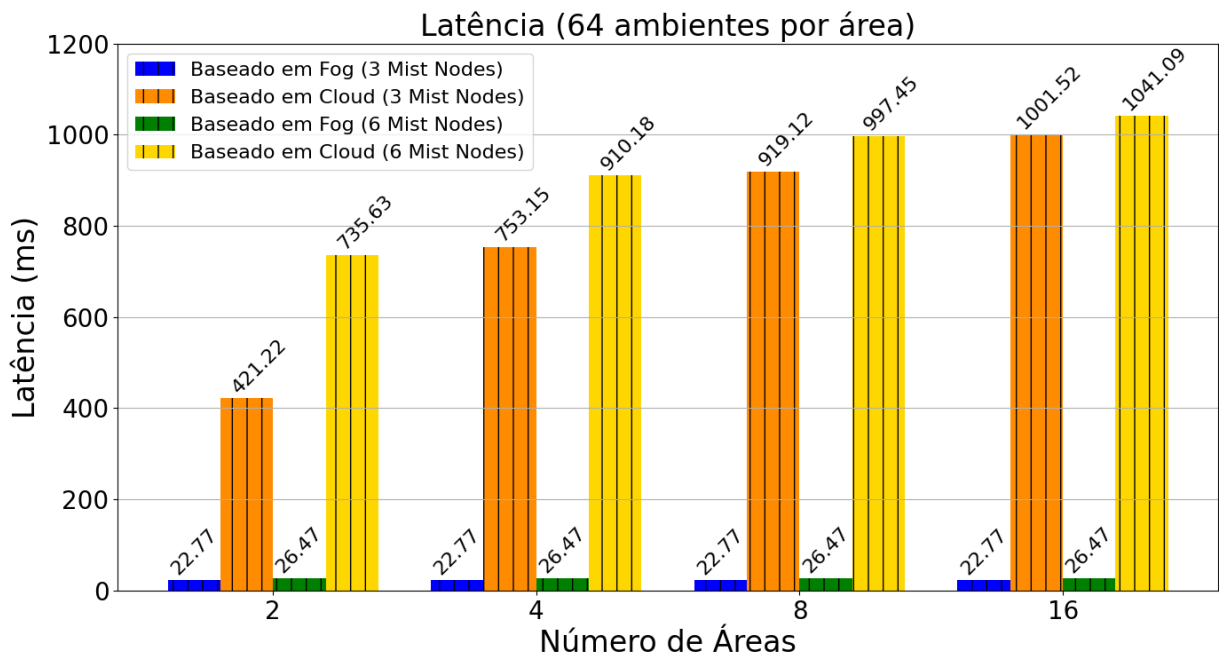
<i>Mist Nodes</i>	Áreas	Ambientes	Total de Sensores
3 <i>mist nodes</i>	2 - 4 - 8 - 16	8 - 16 - 32 - 64	160 a 10.240
6 <i>mist nodes</i>	2 - 4 - 8 - 16	8 - 16 - 32 - 64	320 a 20.480

Fonte: Elaborado pelo Autor.

A latência é um requisito fundamental em sistemas que precisam operar em tempo

real, como um sistema AAL baseado em localização interna. Portanto, a arquitetura deve ser projetada de forma a garantir que o tempo de resposta do sistema seja reduzido o máximo possível. A Figura 32 mostra os valores de latência obtidos para os cenários com 64 ambientes por área, variando o número áreas de cobertura.

Figura 32 – Latência observada variando o número de áreas de cobertura



Fonte: Elaborado pelo Autor.

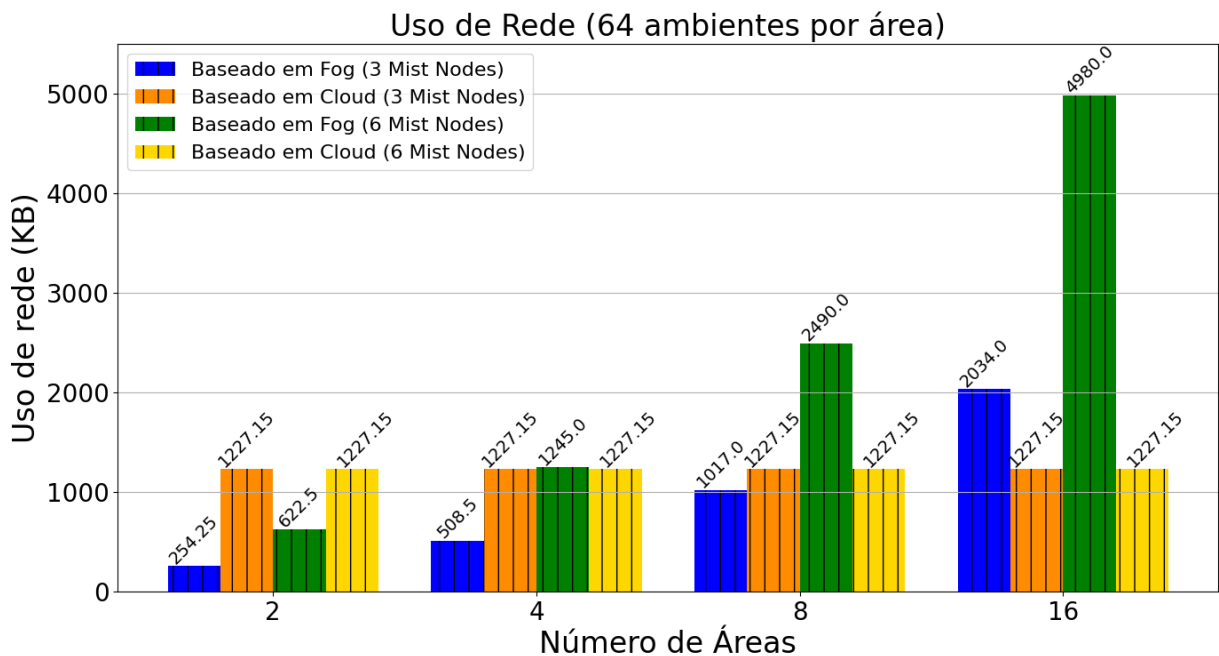
Com base nos resultados, pode-se observar que o cenário baseado em fog reduz significativamente a latência em comparação com o cenário somente em nuvem, tornando viável sua utilização em aplicações de tempo real. Para o mesmo número de sensores e atuadores, os valores de latência da estratégia baseada em fog permanecem constantes. Além disso, mesmo depois de dobrar o número de sensores e atuadores no ambiente, o aumento no valor de latência obtido na estratégia baseada em fog foi muito baixo, inferior a 4 ms. Isso se deve ao fato de que todo o processamento dos dados provenientes dos sensores no ambiente é feito pelo próprio *gateway* local (fog nível 1), e só é necessário encaminhar periodicamente os dados processados para o módulo coordenador, alocado no *gateway* de área (fog nível 2).

Outro fator importante é o uso de rede pelo sistema. A Figura 33 mostra os valores de uso de rede obtidos com as estratégias baseadas em fog e somente nuvem. Aqui pode-se notar que a estratégia baseada em fog apresenta menores taxas de uso de rede em comparação à estratégia somente em nuvem quando observamos as variações do cenário de avaliação com três *mist nodes* e até oito áreas de cobertura. No cenário com 16 áreas de cobertura e três *mist nodes*

já é possível observar um consumo de rede aproximadamente 66% maior na estratégia baseada em fog em comparação com um arranjo somente em nuvem.

Já nas configurações com seis *mist nodes* por ambiente, é possível notar que somente no cenário com duas áreas de cobertura, a estratégia baseada em fog apresenta um menor consumo de rede. Ao observar o cenário com 16 áreas de cobertura nota-se um consumo de rede, na estratégia fog, aproximadamente quatro vezes superior. Além disso, é possível notar que os valores de consumo de rede para todos os cenários em um arranjo somente nuvem se mantiveram constantes (1.227,15 KB).

Figura 33 – Uso de rede observado variando o número de áreas de cobertura

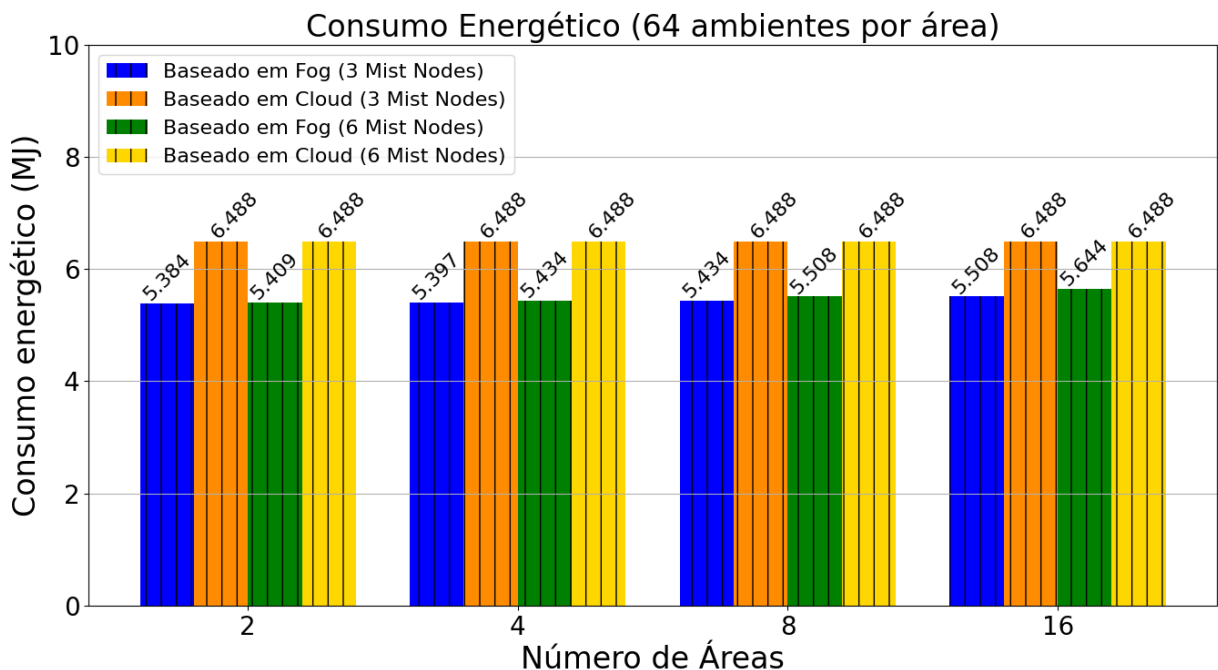


Fonte: Elaborado pelo Autor.

Analisando esses dados e levando em consideração o crescimento significativo do consumo de rede na estratégia fog e o consumo constante na estratégia somente nuvem, pode-se inferir que nessa última houve uma utilização de toda a capacidade de processamento dos servidores já no cenário com duas áreas de cobertura e 64 ambientes por área. Isso explica o nível constante de consumo de rede da estratégia somente nuvem, pois atingindo a capacidade máxima de processamento nos servidores, naturalmente o envio de novos dados sofre uma espera até que possam ser recebidos e processados pelos servidores em nuvem. Corroborando com isso, a Figura 32, apresentada anteriormente, mostra uma latência crescente nos cenários baseados somente em nuvem, o que pode indicar que os servidores de fato atingiram sua capacidade máxima de processamento, mantendo um uso de rede constante e uma latência crescente.

A simulação também levou em consideração a energia consumida pelos servidores em nuvem, buscando investigar o impacto da estratégia de posicionamento baseada em fog sobre esse fator. Inicialmente, esperava-se uma redução no consumo de energia devido a menor carga de processamento em comparação com uma estratégia de posicionamento somente na nuvem. Assim, a Figura 34 apresenta os resultados referentes ao consumo de energia dos servidores em nuvem, onde se pode observar que houve, de fato, uma redução significativa em todos os cenários simulados baseados em fog em comparação com os cenários baseados em nuvem. Percebe-se ainda que, como observado em relação ao consumo de rede, os valores de consumo energético dos servidores em nuvem mantiveram-se constantes em todos os cenários, o que mais uma vez sugere que a capacidade máxima de processamento dos servidores foi atingida na estratégia baseada somente em nuvem.

Figura 34 – Consumo energético observado variando o número de áreas de cobertura

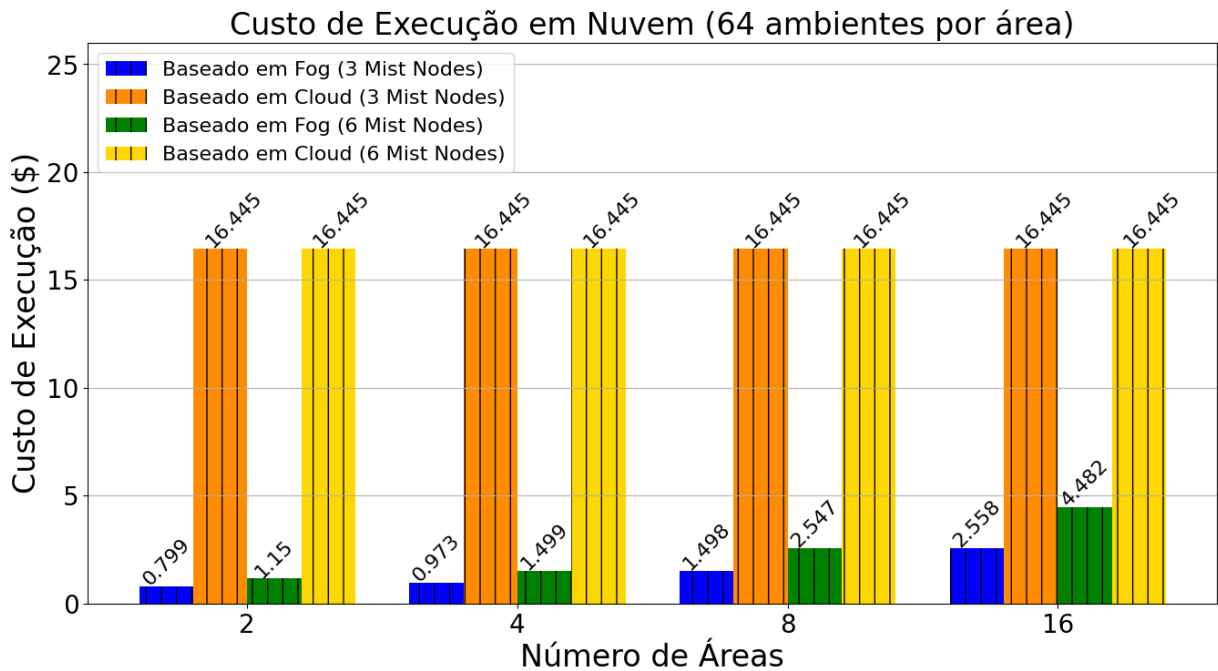


Fonte: Elaborado pelo Autor.

Também é desejável avaliar o impacto da arquitetura proposta no custo total de execução em nuvem. O parâmetro usado para realizar essa medição é o custo por MI (milhão de instruções) executado na nuvem, conforme definido na Tabela 15, em que o valor foi definido sem usar nenhuma moeda específica como base. A ideia é ter uma comparação na ordem de magnitude do nível de redução do custo de execução obtido pela abordagem baseada em fog. A Figura 35 apresenta o custo total de execução de tarefas em nuvem nas duas abordagens, fog e somente nuvem. Observa-se que usando a abordagem baseada em fog, houve redução substancial

nos custos de processamento em nuvem, chegando a cerca de 95 % nos cenários mais extremos.

Figura 35 – Custo de execução em nuvem observado variando o número de áreas de cobertura

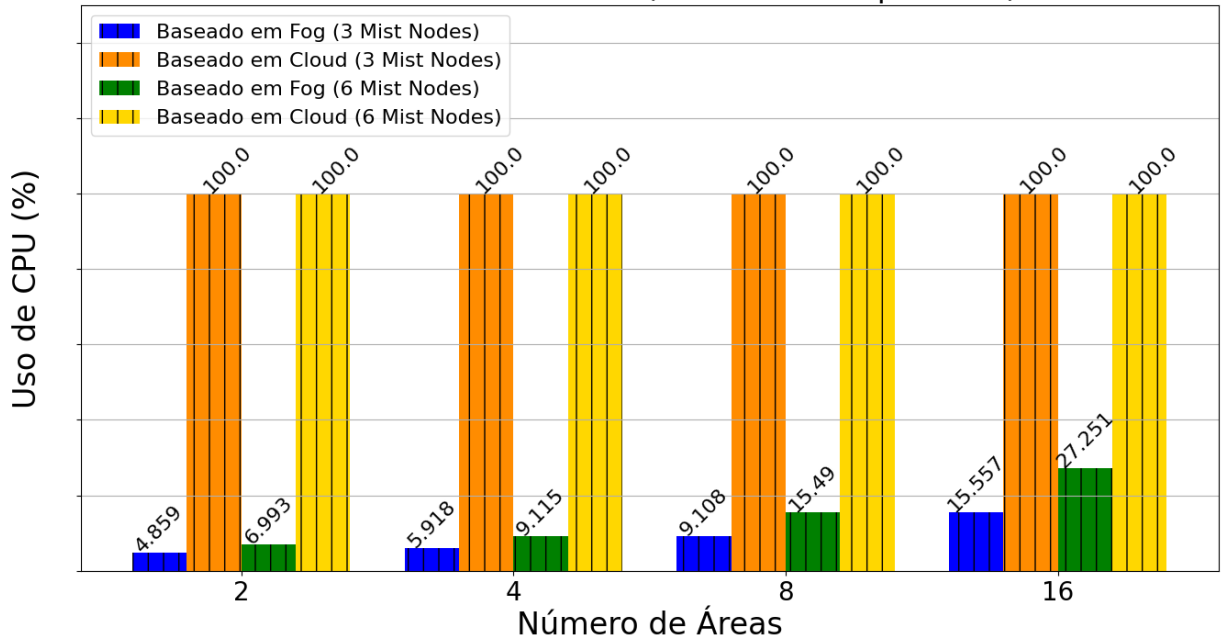


Fonte: Elaborado pelo Autor.

O custo de execução em nuvem está relacionado diretamente com a carga de processamento nos servidores em nuvem, por levar em consideração a quantidade de MI executadas. Sendo assim, tomando como custo máximo de execução os valores apresentados nos cenários baseados somente em nuvem (\$ 16,445) e levando em consideração o alcance de capacidade máxima de processamento dos servidores, como discutido anteriormente, a Figura 36 apresenta a porcentagem de uso de CPU em cada cenário. Pode-se perceber que há uma utilização muito menor dos serviços em nuvem na abordagem baseada em fog, ajudando assim a alcançar reduções substanciais em custo de execução em nuvem, latência e consumo energético, como apresentado anteriormente. No último cenário avaliado, 16 áreas de cobertura com 64 ambientes e seis *mist nodes* por ambiente, totalizando 20.480 sensores, houve uma utilização de somente 27,25 % do poder de processamento dos servidores em nuvem.

Por fim, as Figuras 37 e 38 apresentam a evolução da carga de processamento nos servidores em nuvem para os cenários baseados somente em nuvem com três e com seis *mist nodes*, respectivamente. Pode-se observar que o sistema atinge o uso máximo de CPU nos servidores em nuvem no cenário com duas áreas e 64 ambientes por área, na configuração com três *mist nodes* por ambiente. Já na configuração com seis *mist nodes* por ambiente, a carga máxima de processamento é atingida no cenário com duas áreas e 32 ambientes por área. Logo,

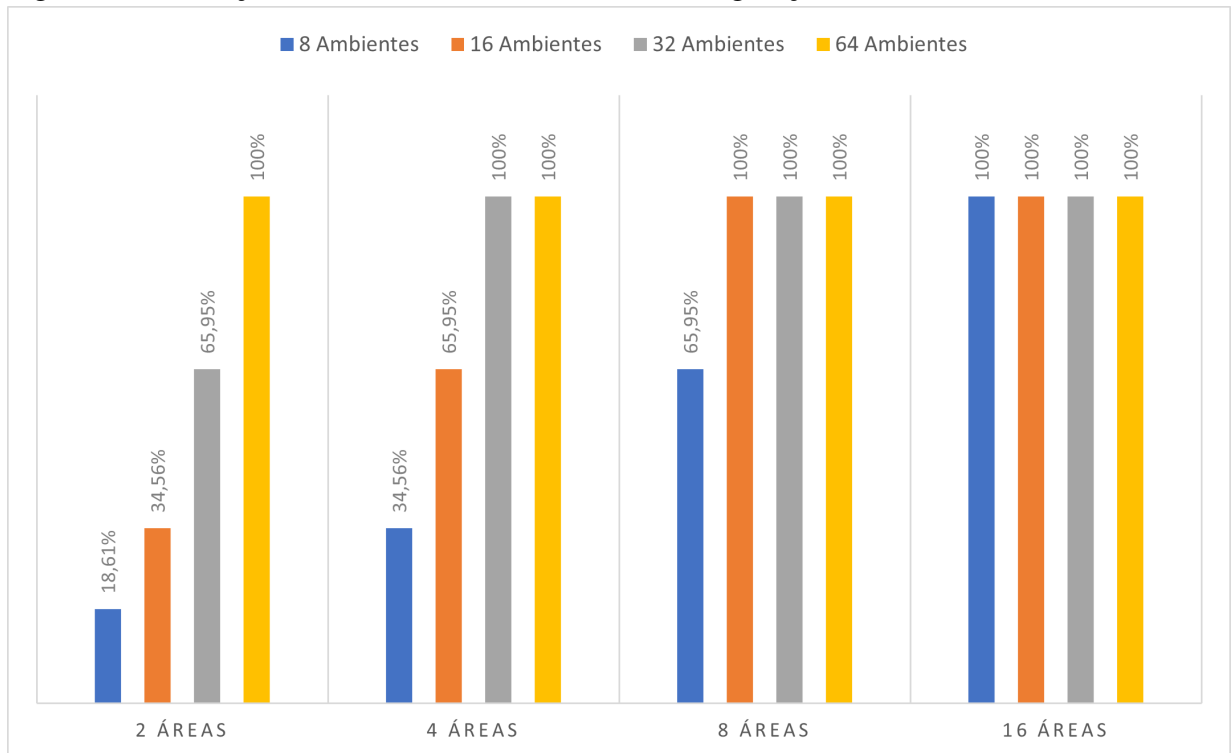
Figura 36 – Uso de CPU em nuvem observado variando o número de áreas de cobertura
 Uso de CPU em Nuvem (64 ambientes por área)



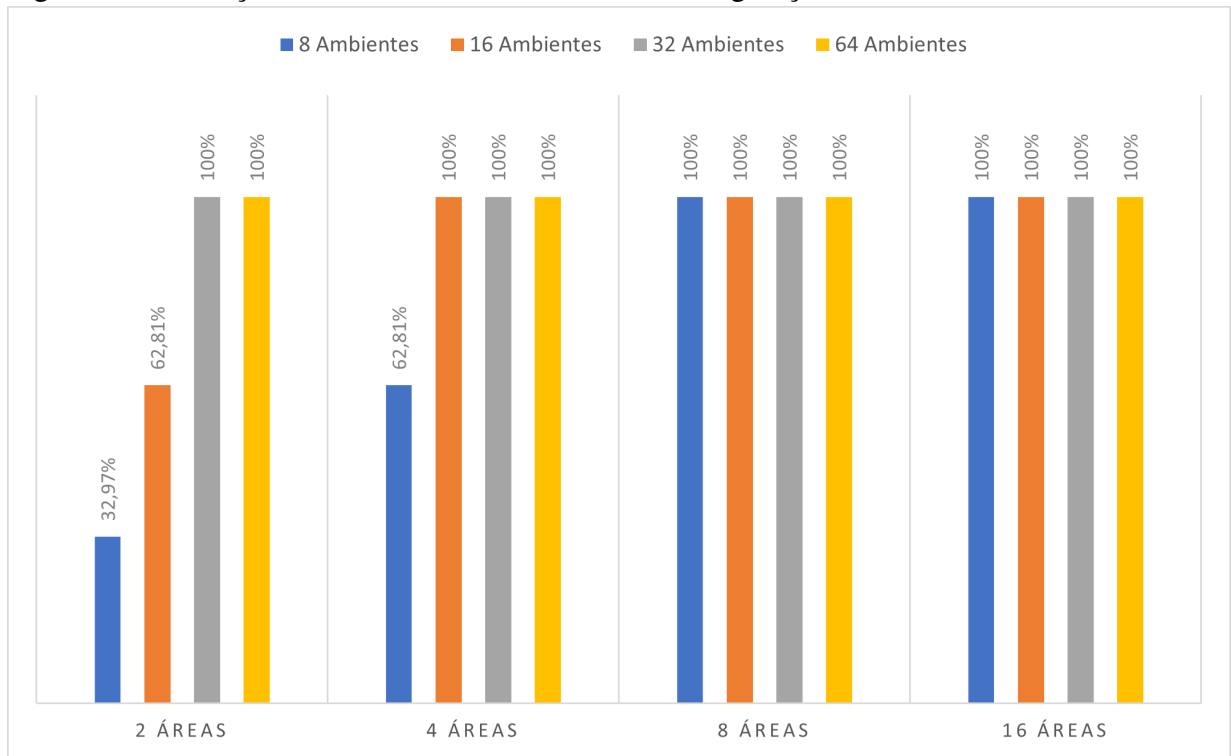
Fonte: Elaborado pelo Autor.

é perceptível que a estratégia baseada em *fog computing* permite uma maior escalabilidade do sistema, quanto à utilização dos serviços em nuvem.

Figura 37 – Evolução do uso de CPU em nuvem na configuração com 3 *mist nodes*



Fonte: Elaborado pelo Autor.

Figura 38 – Evolução do uso de CPU em nuvem na configuração com 6 *mist nodes*

Fonte: Elaborado pelo Autor.

7 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma arquitetura de software para sistemas IPS baseados em IoT para dar suporte a ambientes de vida assistida. Uma revisão sistemática apoiou a definição da arquitetura para identificar as principais soluções e os principais requisitos não funcionais abordados nas arquiteturas propostas para sistemas IPS. Além disso, foi feito um processo de análise e reuso de RNFs consolidados para aplicações de AAL baseados em IoT. Para garantir que a arquitetura proposta possa interagir com serviços em nuvem e atender, principalmente, aos requisitos de desempenho, eficiência energética, redução de custos operacionais e escalabilidade, foram utilizados elementos de *fog computing* em seu projeto.

Nesse sentido, a RSL mostrou que a maioria dos trabalhos não valida as propostas com base em RNF bem definidos e ignora requisitos essenciais para aplicações IPS e AAL, como segurança, privacidade e escalabilidade. A avaliação da arquitetura foi feita utilizando o kit de ferramentas de simulação iFogSim em termos de uso da rede, consumo de energia, latência e custo operacional na nuvem.

A arquitetura proposta, comparada a uma abordagem somente em nuvem, reduziu significativamente a latência, o consumo energético e os custos operacionais do processamento em nuvem, possibilitando maior escalabilidade e melhor desempenho. Assim, os resultados apresentados neste trabalho destacam a viabilidade da proposta para aplicações com restrições de tempo real, como sistemas de AAL baseados em informações de localização interna.

7.1 Publicações

Foram publicados três artigos relacionados a esta pesquisa que auxiliaram o processo de desenvolvimento deste trabalho, bem como apresentaram e validaram resultados intermediários. A Tabela 17 lista os artigos e as conferências onde foram publicados.

7.2 Principais Contribuições

Como principais contribuições deste trabalho pode-se destacar:

- a) revisão sistemática da literatura sobre tecnologias, técnicas, métodos e requisitos não funcionais utilizados no projeto de arquiteturas de software para sistemas IPS baseados em IoT (Capítulo 4);
- b) concepção de uma arquitetura de software para sistemas IPS baseados em IoT

Tabela 17 – Artigos publicados em conferências

Artigo	Veículo de Publicação	Ano
Tecnologias Assistivas de Orientação e Mobilidade para PcDV: Uma Revisão Sistemática da Literatura Brasileira	Congresso Brasileiro de Pesquisa e Desenvolvimento em Tecnologia Assistiva (CBTA)	2021
Análise da Aplicação da Técnica de Trilateração em Redes de Sensores Wi-Fi Para Localização em Ambientes Indoor	XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)	2021
Software Architecture for IoT-based Indoor Positioning Systems for Ambient Assisted Living	IEEE 20th International Conference on Software Architecture (ICSA)	2023

Fonte: Elaborado pelo Autor.

como suporte a ambientes de vida assistida (Capítulo 5);

- c) proposta de abordagem baseada em *Fog Computing*, o que permite uma melhor distribuição geográfica dos serviços, diminuindo latência, consumo energético e custos de execução em nuvem (Capítulo 6);
- d) suporte a serviços baseados em informações contextuais de localização interna e integração com localização externa para um monitoramento contínuo e pervasivo (Capítulo 5 e Seção 6.1).

7.3 Limitações do Trabalho

Com relação às ameaças à validade deste trabalho, pode-se destacar como ameaças a escolha das bases de estudo usadas na RSL e o método de validação usado. Foram utilizadas três bases de pesquisa de estudos para aplicar a string de busca da SLR. Várias outras bases poderiam ser utilizadas e poderiam retornar um número maior de estudos relevantes e aderentes às perguntas da pesquisa. No entanto, acredita-se que as bases de dados utilizadas, devido à sua relevância e aderência bem estabelecidas à área de Ciência da Computação, podem retornar os estudos mais relevantes dentro do escopo da RSL realizada.

O método de avaliação utilizado consistiu em um conjunto de simulação da operação do sistema com base na arquitetura proposta. Não houve avaliação experimental da arquitetura proposta. No entanto, acreditamos que o uso de um kit de ferramentas de simulação otimizado para o contexto de sistemas de IoT baseados em nuvem, como o iFogSim, permite uma avaliação inicial confiável. Isso torna viável avaliar o desempenho do sistema em termos de suas características técnicas e operacionais, o que permite que a arquitetura evolua com mais precisão.

7.4 Trabalhos Futuros

Em trabalhos futuros, pretende-se:

- a) realizar mais simulações para analisar os custos de funcionamento e os requisitos de consumo de energia também nos dispositivos de borda da rede.
- b) desenvolver um protótipo experimental para testar e validar a arquitetura proposta em um cenário real de operação.
- c) realizar experimentos extensivos para verificar o comportamento do sistema em relação a diferentes tipos de técnicas e métodos de localização interna, bem como em cenários de ameaça à segurança e à privacidade de dados e serviços.

REFERÊNCIAS

- ACETO, G.; PERSICO, V.; PESCAPÉ, A. Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0. **Journal of Industrial Information Integration**, Elsevier, v. 18, p. 100129, 2020.
- AHMADI, H.; ARJI, G.; SHAHMORADI, L.; SAFDARI, R.; NILASHI, M.; ALIZADEH, M. The application of internet of things in healthcare: a systematic literature review and classification. **Universal Access in the Information Society**, v. 18, n. 4, p. 837–869, 2019. ISSN 1615-5297. Disponível em: <https://doi.org/10.1007/s10209-018-0618-4>.
- AL-KHAFAJIY, M.; GHAREEB, S.; AL-JUMEILY, R.; ALMURSHEDI, R.; HUSSIEN, A.; BAKER, T.; JARARWEH, Y. A holistic study on emerging iot networking paradigms. In: **2019 12th International Conference on Developments in eSystems Engineering (DeSE)**. [S. l.: s. n.], 2019. p. 943–949.
- ALLETTO, S.; CUCCHIARA, R.; FIORE, G. D.; MAINETTI, L.; MIGHALI, V.; PATRONO, L.; SERRA, G. An indoor location-aware system for an iot-based smart museum. **IEEE Internet of Things Journal**, v. 3, n. 2, p. 244–253, 2016.
- ALLOULAH, M.; TUOMINEN, L. Imulet: A cloudlet for inertial tracking. In: **Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications**. New York, NY, USA: Association for Computing Machinery, 2021. (HotMobile '21), p. 50–56. ISBN 9781450383233. Disponível em: <https://doi.org/10.1145/3446382.3448364>.
- ALMEIDA, A.; MULERO, R.; RAMETTA, P.; UROŠEVIĆ, V.; ANDRIĆ, M.; PATRONO, L. A critical analysis of an iot—aware aal system for elderly monitoring. **Future Generation Computer Systems**, Elsevier, v. 97, p. 598–619, 2019.
- ALY, M.; KHOMH, F.; HAOUES, M.; QUINTERO, A.; YACOUT, S. Enforcing security in internet of things frameworks: A systematic literature review. **Internet of Things**, v. 6, p. 100050, 2019. ISSN 2542-6605. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2542660518300805>.
- ARFAT, Y.; AQIB, M.; MEHMOOD, R.; ALBESHRI, A.; KATIB, I.; ALBOGAMI, N.; ALZAHIRANI, A. Enabling smarter societies through mobile big data fogs and clouds. **Procedia Computer Science**, v. 109, p. 1128–1133, 2017. ISSN 1877-0509. 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050917311213>.
- BAE, I.-H. An ontology-based approach to adl recognition in smart homes. **Future Generation Computer Systems**, v. 33, p. 32–41, 2014. ISSN 0167-739X. Special Section on Applications of Intelligent Data and Knowledge Processing Technologies; Guest Editor: Dominik Ślęzak. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X13000642>.
- BARBOSA, P.; FIGUEIREDO, A.; SOUTO, S.; GAETA, E.; ARAUJO, E.; TEIXEIRA, T. An open source software architecture and ready-to-use components for health iot. In: **2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)**. [S. l.: s. n.], 2020. p. 374–379.

- BASIRI, A.; LOHAN, E. S.; MOORE, T.; WINSTANLEY, A.; PELTOLA, P.; HILL, C.; AMIRIAN, P.; Figueiredo e Silva, P. Indoor location based services challenges, requirements and usability of current solutions. **Computer Science Review**, v. 24, p. 1–12, 2017. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013716301782>.
- BATTISTONI, P.; SEBILLO, M.; VITIELLO, G. Experimenting with a fog-computing architecture for indoor navigation. In: **2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)**. [S. l.: s. n.], 2019. p. 161–165.
- BELMONTE-FERNÁNDEZ, Ó.; PUERTAS-CABEDO, A.; TORRES-SOSPEDRA, J.; MONTOLIU-COLÁS, R.; TRILLES-OLIVER, S. An indoor positioning system based on wearables for ambient-assisted living. **Sensors**, MDPI, v. 17, n. 1, p. 36, 2016.
- BENSON, T. **Principles of health interoperability HL7 and SNOMED**. [S. l.]: Springer Science & Business Media, 2012.
- BONOMI, F.; MILITO, R.; ZHU, J.; ADDEPALLI, S. Fog computing and its role in the internet of things. In: **Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing**. New York, NY, USA: Association for Computing Machinery, 2012. (MCC '12), p. 13–16. ISBN 9781450315197. Disponível em: <https://doi.org/10.1145/2342509.2342513>.
- BOUDANI, B. E.; KANARIS, L.; KOKKINIS, A.; CHRYSOULAS, C.; DAGIUKLAS, T.; STAVROU, S. Positioning as service for 5g iot networks. In: **2021 Telecoms Conference (ConfTELE)**. [S. l.: s. n.], 2021. p. 1–6.
- BROEK, G. van den; CAVALLO, F.; WEHRMANN, C. **AALIANCE ambient assisted living roadmap**. [S. l.]: IOS press, 2010. v. 6.
- CADENA, C.; CARLONE, L.; CARRILLO, H.; LATIF, Y.; SCARAMUZZA, D.; NEIRA, J.; REID, I.; LEONARD, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. **IEEE Transactions on Robotics**, v. 32, n. 6, p. 1309–1332, 2016.
- CALVARESI, D.; CESARINI, D.; SERNANI, P.; MARINONI, M.; DRAGONI, A. F.; STURM, A. Exploring the ambient assisted living domain: a systematic review. **Journal of Ambient Intelligence and Humanized Computing**, Springer, v. 8, p. 239–257, 2017.
- CALVILLO-ARBIZU, J.; ROMÁN-MARTÍNEZ, I.; REINA-TOSINA, J. Internet of things in health: Requirements, issues, and gaps. **Computer Methods and Programs in Biomedicine**, v. 208, p. 106231, Sep 2021. ISSN 0169-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169260721003059>.
- COSTA, N.; DOMINGUES, P.; FDEZ-RIVEROLA, F.; PEREIRA, A. A mobile virtual butler to bridge the gap between users and ambient assisted living: A smart home case study. **Sensors**, MDPI, v. 14, n. 8, p. 14302–14329, 2014.
- DAVIDSON, P.; PICHÉ, R. A survey of selected indoor positioning methods for smartphones. **IEEE Communications Surveys Tutorials**, v. 19, n. 2, p. 1347–1370, 2017.
- DOHR, A.; MODRE-OPSRIAN, R.; DROBICS, M.; HAYN, D.; SCHREIER, G. The internet of things for ambient assisted living. In: **2010 Seventh International Conference on Information Technology: New Generations**. [S. l.: s. n.], 2010. p. 804–809.

- DU, X.; YANG, K.; BISIO, I.; LAVAGETTO, F.; SCIARRONE, A. An ap-centred smart probabilistic fingerprint system for indoor positioning. In: **2018 IEEE International Conference on Communications (ICC)**. [S. l.: s. n.], 2018. p. 1–6.
- DUARTE, P. A.; BARRETO, F. M.; AGUILAR, P. A.; BOUDY, J.; ANDRADE, R. M.; VIANA, W. Aal platforms challenges in iot era: a tertiary study. In: IEEE. **2018 13th Annual Conference on System of Systems Engineering (SoSE)**. [S. l.], 2018. p. 106–113.
- ESCOBAR, J. J. L.; REDONDO, R. P. D.; GIL-CASTIÑEIRA, F. In-depth analysis and open challenges of mist computing. **Journal of Cloud Computing**, v. 11, n. 1, p. 81, Nov 2022. ISSN 2192-113X. Disponível em: <https://doi.org/10.1186/s13677-022-00354-x>.
- FORTINO, G.; GUZZO, A.; IANNI, M.; LEOTTA, F.; MECCELLA, M. Predicting activities of daily living via temporal point processes: Approaches and experimental results. **Computers & Electrical Engineering**, Elsevier, v. 96, p. 107567, 2021.
- GARCÉS, L.; OQUENDO, F.; NAKAGAWA, E. Y. Assessment of reference architectures and reference models for ambient assisted living systems: Results of a systematic literature review. **International Journal of E-Health and Medical Communications (IJEHMC)**, IGI Global, v. 11, n. 1, p. 17–36, 2020.
- GIRI, A.; DUTTA, S.; NEOGY, S.; DAHAL, K.; PERVEZ, Z. Internet of things (iot): A survey on architecture, enabling technologies, applications and challenges. In: **Proceedings of the 1st International Conference on Internet of Things and Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2017. (IML '17). ISBN 9781450352437. Disponível em: <https://doi.org/10.1145/3109761.3109768>.
- GU, Y.; LO, A.; NIEMEGERERS, I. A survey of indoor positioning systems for wireless personal networks. **IEEE Communications Surveys Tutorials**, v. 11, n. 1, p. 13–32, 2009.
- GULATI, N.; KAUR, P. D. Friendcare-aal: a robust social iot based alert generation system for ambient assisted living. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–28, 2022.
- GUPTA, B.; QUAMARA, M. An overview of internet of things (iot): Architectural aspects, challenges, and protocols. **Concurrency and computation**, Wiley Subscription Services, Inc, Hoboken, v. 32, n. 21, p. n/a, 2020. ISSN 1532-0626.
- GUPTA, H.; DASTJERDI, A. V.; GHOSH, S. K.; BUYYA, R. 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. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2509>.
- HAJVALI, M.; ADABI, S.; REZAEI, A.; HOSSEINZADEH, M. Software architecture for iot-based health-care systems with cloud/fog service model. **Cluster Computing**, Springer, v. 25, n. 1, p. 91–118, 2022.
- ISLAM, S. M. R.; KWAK, D.; KABIR, M. H.; HOSSAIN, M.; KWAK, K.-S. The internet of things for health care: A comprehensive survey. **IEEE Access**, v. 3, p. 678–708, 2015.
- KHALAJMEHRABADI, A.; GATSIS, N.; AKOPIAN, D. Modern wlan fingerprinting indoor positioning methods and deployment challenges. **IEEE Communications Surveys Tutorials**, v. 19, n. 3, p. 1974–2002, 2017.

KHAN, R.; KHAN, S. U.; ZAHEER, R.; KHAN, S. Future internet: The internet of things architecture, possible applications and key challenges. In: **2012 10th International Conference on Frontiers of Information Technology**. [S. l.: s. n.], 2012. p. 257–260.

KHANNA, A.; KAUR, S. Internet of things (iot), applications and challenges: A comprehensive review. **Wireless Personal Communications**, v. 114, n. 2, p. 1687–1762, Sep 2020. ISSN 1572-834X. Disponível em: <https://doi.org/10.1007/s11277-020-07446-4>.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Keele, Staffs ST5 5BG, UK, 2007.

KLEINBERGER, T.; BECKER, E. R.; HOLZINGER ANDREAS AMD MÜLLER, P. Ambient intelligence in assisted living: Enable elderly people to handle future interfaces. In: **Universal Access in Human-Computer Interaction (UAHCI 2007)**. [S. l.: s. n.], 2007. v. 4555.

KRISHNAN, J.; VASUDEVAN, S. K. An extensive survey on iot smart gateways, software architecture, related protocols and challenges. In: **2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)**. [S. l.: s. n.], 2019. p. 1–6.

KRUCHTEN, P. The 4+1 view model of architecture. **IEEE Software**, v. 12, n. 6, p. 42–50, 1995.

KUMARI, A.; TANWAR, S.; TYAGI, S.; KUMAR, N. Fog computing for healthcare 4.0 environment: Opportunities and challenges. **Computers & Electrical Engineering**, v. 72, p. 1–13, 2018. ISSN 0045-7906. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045790618303860>.

LASHKARI, A. H.; PARHIZKAR, B.; NGAN, M. N. A. Wifi-based indoor positioning system. In: **2010 Second International Conference on Computer and Network Technology**. [S. l.: s. n.], 2010. p. 76–78.

LI, Z.; CAO, J.; LIU, X.; ZHANG, J.; HU, H.; YAO, D. A self-adaptive bluetooth indoor localization system using lstm-based distance estimator. In: **2020 29th International Conference on Computer Communications and Networks (ICCCN)**. [S. l.: s. n.], 2020. p. 1–9.

LIN, J.; YU, W.; ZHANG, N.; YANG, X.; ZHANG, H.; ZHAO, W. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. **IEEE Internet of Things Journal**, v. 4, n. 5, p. 1125–1142, 2017.

LIU, J.; JAIN, R. Survey of wireless based indoor localization technologies. **Computer Science & Engineering**, 2014. Disponível em: <https://www.cse.wustl.edu/~jain/cse574-14/ftp/indoor.pdf>.

MACAGNANO, D.; DESTINO, G.; ABREU, G. Indoor positioning: A key enabling technology for iot applications. In: **2014 IEEE World Forum on Internet of Things (WF-IoT)**. [S. l.: s. n.], 2014. p. 117–118.

MAHMUD, R.; KOCH, F. L.; BUYYA, R. Cloud-fog interoperability in iot-enabled healthcare solutions. In: **Proceedings of the 19th international conference on distributed computing and networking**. [S. l.: s. n.], 2018. p. 1–10.

- MAHMUD, R.; PALLEWATTA, S.; GOUDARZI, M.; BUYYA, R. ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments. **Journal of Systems and Software**, v. 190, p. 111351, 2022. ISSN 0164-1212. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121222000863>.
- MAINETTI, L.; MIGHALI, V.; PATRONO, L. A location-aware architecture for heterogeneous building automation systems. In: **2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. [S. l.: s. n.], 2015. p. 1065–1070.
- MAINETTI, L.; PATRONO, L.; SECCO, A.; SERGI, I. An iot-aware aal system for elderly people. In: **2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)**. [S. l.: s. n.], 2016. p. 1–6.
- MAINETTI, L.; PATRONO, L.; SERGI, I. A survey on indoor positioning systems. In: **2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)**. [S. l.: s. n.], 2014. p. 111–120.
- MCNICOLL, G. World population ageing 1950-2050. **Population and Development Review**, v. 28, p. 814+, Dec 2002. ISSN 00987921. 4. Disponível em: <https://link.gale.com/apps/doc/A96904156/AONE?u=egn&sid=googleScholar&xid=e48b6717>.
- MENDOZA-SILVA, G. M.; TORRES-SOSPEDRA, J.; HUERTA, J. A meta-review of indoor positioning systems. **Sensors**, v. 19, n. 20, 2019. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/19/20/4507>.
- MERRY, K.; BETTINGER, P. Smartphone gps accuracy study in an urban environment. **PLOS ONE**, Public Library of Science, v. 14, n. 7, p. 1–19, 07 2019. Disponível em: <https://doi.org/10.1371/journal.pone.0219890>.
- MONEKOSSO, D.; FLOREZ-REVUELTA, F.; REMAGNINO, P. Ambient assisted living [guest editors' introduction]. **IEEE Intelligent Systems**, IEEE, v. 30, n. 4, p. 2–6, 2015.
- MOREIRA, F. M.; FARIAS, M. S.; CARVALHO, P. V. R. Posicionamento em ambientes internos com dispositivos wi-fi de baixo custo. In: **International Nuclear Atlantic Conference (INAC)**. Belo Horizonte: [S. n.], 2017.
- MPEIS, P.; ROUSSEL, T.; KUMAR, M.; COSTA, C.; LAOUDIASENIS, C.; CAPOT-RAY, D.; ZEINALIPOUR-YAZTI, D. The anyplace 4.0 iot localization architecture. In: **2020 21st IEEE International Conference on Mobile Data Management (MDM)**. [S. l.: s. n.], 2020. p. 218–225.
- MUHAMMED, T.; MEHMOOD, R.; ALBESHRI, A.; KATIB, I. Ubehealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. **IEEE Access**, v. 6, p. 32258–32285, 2018.
- MUNISWAMAIAH, M.; AGERWALA, T.; TAPPERT, C. C. Fog computing and the internet of things (iot): A review. In: **2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)**. [S. l.: s. n.], 2021. p. 10–12.

- NASCIMENTO, P. P.; KIMURA, B.; GUIDONI, D.; VILLAS, L. A. Uma solução cooperative positioning integrada com dead reckoning para suporte a gps sem linhas de visada utilizando a tecnologia de radio 802.11p. In: **Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. Porto Alegre, RS, Brasil: SBC, 2018. p. 1145–1158. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/2484>.
- NAVYA, C.; SALVI, S.; JACOB, N. D.; KUMAR, S. A roof computing architecture based indoor positioning system for iot applications. In: **2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)**. [S. l.: s. n.], 2020. p. 19–24.
- NGUYEN, Q. H.; JOHNSON, P.; NGUYEN, T. T.; RANGLES, M. A novel architecture using ibeacons for localization and tracking of people within healthcare environment. In: **2019 Global IoT Summit (GIoTS)**. [S. l.: s. n.], 2019. p. 1–6.
- NIKOUI, T. S.; RAHMANI, A. M.; BALADOR, A.; JAVADI, H. H. S. Internet of things architecture challenges: A systematic review. **International Journal of Communication Systems**, v. 34, n. 4, p. e4678, 2021. E4678 IJCS-19-1067.R1. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4678>.
- NOORUDDIN, S.; ISLAM, M. M.; SHARNA, F. A.; ALHETARI, H.; KABIR, M. N. Sensor-based fall detection systems: a review. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–17, 2022.
- OBEIDAT, H.; SHUAIEB, W.; OBEIDAT, O.; ABD-ALHAMEED, R. A review of indoor localization techniques and wireless technologies. **Wireless Personal Communications**, v. 119, n. 1, 2021. ISSN 1572-834X. Disponível em: <https://doi.org/10.1007/s11277-021-08209-5>.
- PASCACIO, P.; CASTELEYN, S.; TORRES-SOSPEDRA, J.; LOHAN, E. S.; NURMI, J. Collaborative indoor positioning systems: A systematic review. **Sensors**, v. 21, n. 3, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/3/1002>.
- PAWAR, N.; BOURGEOU, T.; CHAOUCHI, H. Study of iot architecture and application invariant functionalities. In: **2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. [S. l.: s. n.], 2021. p. 667–671.
- PENG, D.; SUN, L.; ZHOU, R.; WANG, Y. Study qos-aware fog computing for disease diagnosis and prognosis. **Mobile Networks and Applications**, Springer, p. 1–8, 2022.
- PIRAS, M.; CINA, A. Indoor positioning using low cost gps receivers: Tests and statistical analyses. In: **2010 International Conference on Indoor Positioning and Indoor Navigation**. [S. l.: s. n.], 2010. p. 1–7.
- PLIKYNAS, D.; ŽVIRONAS, A.; BUDRIONIS, A.; GUDAUSKIS, M. Indoor navigation systems for visually impaired persons: Mapping the features of existing technologies to user needs. **Sensors**, v. 20, n. 3, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/3/636>.
- PU, Y.-C.; YOU, P.-C. Indoor positioning system based on ble location fingerprinting with classification approach. **Applied Mathematical Modelling**, v. 62, p. 654–663, 2018. ISSN 0307-904X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0307904X18302841>.

PULIAFITO, C.; MINGOZZI, E.; LONGO, F.; PULIAFITO, A.; RANA, O. Fog computing for the internet of things: A survey. **ACM Transactions on Internet Technology (TOIT)**, ACM New York, NY, USA, v. 19, n. 2, p. 1–41, 2019.

RAHMANI, A. M.; GIA, T. N.; NEGASH, B.; ANZANPOUR, A.; AZIMI, I.; JIANG, M.; LILJEBERG, P. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. **Future Generation Computer Systems**, Elsevier, v. 78, p. 641–658, 2018.

RAPPAPORT, T. S. **Comunicações Sem Fio: Princípios e Práticas**. [S. l.: s. n.], 2009.

RASHIDI, P.; MIHAILIDIS, A. A survey on ambient-assisted living tools for older adults. **IEEE Journal of Biomedical and Health Informatics**, v. 17, n. 3, p. 579–590, 2013.

RODRIGUES, J. J.; SEGUNDO, D. B. D. R.; JUNQUEIRA, H. A.; SABINO, M. H.; PRINCE, R. M.; AL-MUHTADI, J.; ALBUQUERQUE, V. H. C. D. Enabling technologies for the internet of health things. **Ieee Access**, IEEE, v. 6, p. 13129–13141, 2018.

ROSENDO, D.; COSTAN, A.; VALDURIEZ, P.; ANTONIU, G. Distributed intelligence on the edge-to-cloud continuum: A systematic literature review. **Journal of Parallel and Distributed Computing**, v. 166, p. 71–94, 2022. ISSN 0743-7315. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0743731522000843>.

SIKARWAR, R.; YADAV, P.; DUBEY, A. A survey on iot enabled cloud platforms. In: **2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)**. [S. l.: s. n.], 2020. p. 120–124.

SIKERIDIS, D.; RIMAL, B. P.; PAPAPANAGIOTOU, I.; DEVETSIKIOTIS, M. Unsupervised crowd-assisted learning enabling location-aware facilities. **IEEE Internet of Things Journal**, v. 5, n. 6, p. 4699–4713, 2018.

SILVA, R. R. e; GONÇALVES, G.; MEDEIROS, V. Avaliação de unidade de medição inercial utilizando técnicas de reconstrução de trajetória de pedestres. In: **Anais Estendidos do XI Simpósio Brasileiro de Engenharia de Sistemas Computacionais**. Porto Alegre, RS, Brasil: SBC, 2021. p. 24–31. ISSN 2763-9002. Disponível em: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/18490.

SURENDRAN, D.; ROHINIA, M. Ble bluetooth beacon based solution to monitor egress of alzheimer's disease sufferers from indoors. **Procedia Computer Science**, Elsevier, v. 165, p. 591–597, 2019.

TAIVALSAARI, A.; MIKKONEN, T. A taxonomy of iot client architectures. **IEEE Software**, v. 35, n. 3, p. 83–88, 2018.

TEGOU, T.; KALAMARAS, I.; TSIPOURAS, M.; GIANNAKEAS, N.; VOTIS, K.; TZOVARAS, D. A low-cost indoor activity monitoring system for detecting frailty in older adults. **Sensors**, MDPI, v. 19, n. 3, p. 452, 2019.

TERÁN, M.; ARANDA, J.; CARRILLO, H.; MENDEZ, D.; PARRA, C. Iot-based system for indoor location using bluetooth low energy. In: **2017 IEEE Colombian Conference on Communications and Computing (COLCOM)**. [S. l.: s. n.], 2017. p. 1–6.

VORA, J.; TANWAR, S.; TYAGI, S.; KUMAR, N.; RODRIGUES, J. J. Faal: Fog computing-based patient monitoring system for ambient assisted living. In: IEEE. **2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom)**. [S. l.], 2017. p. 1–6.

WASHIZAKI, H.; OGATA, S.; HAZEYAMA, A.; OKUBO, T.; FERNANDEZ, E. B.; YOSHIOKA, N. Landscape of architecture and design patterns for iot systems. **IEEE Internet of Things Journal**, v. 7, n. 10, p. 10091–10101, 2020.

YACCHIREMA, D. C.; PALAU, C. E.; ESTEVE, M. Enable iot interoperability in ambient assisted living: Active and healthy aging scenarios. In: IEEE. **2017 14th IEEE annual consumer communications & networking conference (CCNC)**. [S. l.], 2017. p. 53–58.

ZAFARI, F.; GKELIAS, A.; LEUNG, K. K. A survey of indoor localization systems and technologies. **IEEE Communications Surveys & Tutorials**, v. 21, n. 3, p. 2568–2599, 2019.

ZAFARI, F.; PAPAPANAGIOTOU, I.; CHRISTIDIS, K. Microlocation for internet-of-things-equipped smart buildings. **IEEE Internet of Things Journal**, v. 3, n. 1, p. 96–112, 2016.

APÊNDICE A – FORMULÁRIO DE EXTRAÇÃO DE DADOS DA RSL

Tabela 18 – Formulário de Extração de Dados da RSL

Paper ID					
Title					
Abstract					
Keywords					
Publication Year					
Publication Type	Article	Conference Paper	Other		
Type of solution presented	Algorithm	Concrete Architecture	Concrete System	Reference Architecture	Other
Does the paper answer the research questions?	No	Partially	Yes		
Objectives					
Does the work provide any validation of the architecture or standard?	No	Partially	Yes		
Setup configuration for testing and validation of the proposed solution					
Technologies, protocols, and patterns used for communication in the solution presented					
Localization accuracy obtained by the proposed solution					
What are the NFRs incorporated in the proposed solution?					
Is it an IoT-based IPS architecture for AAL?	No	Partially	Yes		
Which AAL goals are considered in the proposed solution?	G1.	G2.	G3.		
				A	
				B	
				C	
				D	

Fonte: Elaborado pelo Autor.

APÊNDICE B – *SETUP* DA SIMULAÇÃO NO IFOGSIM

A tabela 19 apresenta os parâmetros de configuração da aplicação nas simulações realizadas no iFogSim, onde é possível observar os valores de carga de processamento, em MIPS, e de uso de rede, em bytes, bem como os módulos de origem e destino de cada tupla definida. Os trechos de código Java apresentados nos Códigos-fonte 1, 2, 3 e 4 apresentam o *setup* de simulação definido utilizando a API Java disponibilizada pelo iFogSim. Os Códigos-fonte 1 e 2 apresentam a estratégia de mapeamento dos módulos da aplicação nos cenários baseados em fog e em nuvem, respectivamente. Já os trechos de código apresentados nos Códigos-fonte 3 e 4 demonstram a criação do modelo de aplicação, baseado no DAG apresentado no Capítulo 6 e utilizando os parâmetros da Tabela 19, para os cenários baseados em fog e em nuvem, respectivamente.

Tabela 19 – Parâmetros de configuração da simulação no iFogSim

Tipo de Tupla	Processamento (MIPS)	Uso de Rede (B)	Módulo Origem	Módulo Destino
<i>raw_data</i>	100	300	<i>sensors</i>	<i>client</i>
<i>location_data</i>	250	150	<i>client</i>	<i>location_processing</i>
<i>sensors_data</i>	500	250	<i>client</i>	<i>processing_module</i>
<i>processed_location</i>	500	400	<i>location_processing</i>	<i>processing_module</i>
<i>processed_data</i>	300	800	<i>processing_module</i>	<i>context_coordinator</i>
<i>event_notification</i>	100	200	<i>context_coordinator</i>	<i>notification</i>
<i>persistent_data</i>	150	1000	<i>context_coordinator</i>	<i>storage</i>
<i>action_command</i>	50	50	<i>processing_module</i>	<i>client</i>
<i>actuation_signal</i>	50	50	<i>client</i>	<i>actuators</i>

Fonte: Elaborado pelo Autor.

Código-fonte 1 – Mapeamento dos módulos da aplicação no cenário baseado em *fog computing*

```

1 String appId = "iot4amar";
2 FogBroker broker = new FogBroker("broker");
3 Application application = createApplication(appId, broker.getId());
4 application.setUserId(broker.getId());
5 createFogDevices(broker.getId(), appId);
6 Controller controller = null;
7 ModuleMapping moduleMapping = ModuleMapping.createModuleMapping();
8
9 for(FogDevice device: fogDevices) {
10     if(device.getName().startsWith("m")) {
11         moduleMapping.addModuleToDevice("client", device.getName());
12     }
13 }
14

```

```

15 moduleMapping.addModuleToDevice("storage", "cloud");
16 moduleMapping.addModuleToDevice("notification", "cloud");
17
18 for(FogDevice device: fogDevices) {
19     if(device.getName().startsWith("h")) {
20         moduleMapping.addModuleToDevice("processing_module", device.getName());
21     }
22     if(device.getName().startsWith("a")) {
23         moduleMapping.addModuleToDevice("context_coordinator", device.getName());
24     }
25 }
26
27 controller = new Controller("master-controller", fogDevices, sensors, actuators);
28 controller.submitApplication(application, new ModulePlacementEdgewards(fogDevices, sensors,
    actuators, application, moduleMapping));

```

Código-fonte 2 – Mapeamento dos módulos da aplicação no cenário baseado em nuvem

```

1 String appId = "iot4amar";
2
3 FogBroker broker = new FogBroker("broker");
4
5 Application application = createApplication(appId, broker.getId());
6 application.setUserId(broker.getId());
7
8 createFogDevices(broker.getId(), appId);
9
10 Controller controller = null;
11
12 ModuleMapping moduleMapping = ModuleMapping.createModuleMapping();
13
14 moduleMapping.addModuleToDevice("client", "cloud");
15 moduleMapping.addModuleToDevice("storage", "cloud");
16 moduleMapping.addModuleToDevice("notification", "cloud");
17 moduleMapping.addModuleToDevice("location_processing", "cloud");
18 moduleMapping.addModuleToDevice("processing_module", "cloud");
19 moduleMapping.addModuleToDevice("context_coordinator", "cloud");
20
21 controller = new Controller("master-controller", fogDevices, sensors, actuators);
22
23 controller.submitApplication(application, new ModulePlacementMapping(fogDevices,
    application, moduleMapping));

```

Código-fonte 3 – Criação do modelo de aplicação seguindo o modelo DAG na estratégia baseada em fog

```

1 private static Application createApplication(String appId, int userId){
2     Application application = Application.createApplication(appId, userId);
3
4     application.addAppModule("client", 10);
5     application.addAppModule("location_processing", 10);
6     application.addAppModule("processing_module", 10);
7     application.addAppModule("context_coordinator", 10);
8     application.addAppModule("storage", 10);
9     application.addAppModule("notification", 10);
10
11    application.addAppEdge("raw_data", "client", 100, 300, "raw_data", Tuple.UP, AppEdge.
        SENSOR);
12    application.addAppEdge("client", "location_processing", 250, 150, "location_data",
        Tuple.UP, AppEdge.MODULE);
13    application.addAppEdge("client", "processing_module", 500, 250, "sensors_data", Tuple.
        UP, AppEdge.MODULE);
14    application.addAppEdge("location_processing", "processing_module", 500, 400, "
        processed_location", Tuple.UP, AppEdge.MODULE);
15
16    application.addAppEdge("processing_module", "context_coordinator", 1000, 300, 800, "
        processed_data", Tuple.UP, AppEdge.MODULE);
17
18    application.addAppEdge("context_coordinator", "notification", 100, 200, "
        event_notification", Tuple.UP, AppEdge.MODULE);
19    application.addAppEdge("context_coordinator", "storage", 150, 1000, "persistent_data",
        Tuple.UP, AppEdge.MODULE);
20    application.addAppEdge("processing_module", "client", 50, 50, "action_command", Tuple.
        DOWN, AppEdge.MODULE);
21    application.addAppEdge("client", "ACTUATORS", 50, 50, "actuation_signal", Tuple.DOWN,
        AppEdge.ACTUATOR);
22
23    application.addTupleMapping("client", "raw_data", "location_data", new
        FractionalSelectivity(1.0));
24    application.addTupleMapping("client", "raw_data", "sensors_data", new
        FractionalSelectivity(1.0));
25    application.addTupleMapping("location_processing", "location_data", "processed_location
        ", new FractionalSelectivity(1.0));
26    application.addTupleMapping("context_coordinator", "processed_data", "
        event_notification", new FractionalSelectivity(1.0));
27    application.addTupleMapping("context_coordinator", "processed_data", "persistent_data",
        new FractionalSelectivity(1.0));
28
29    application.addTupleMapping("processing_module", "sensors_data", "action_command", new
        FractionalSelectivity(1.0));
30    application.addTupleMapping("processing_module", "processed_location", "action_command"
        , new FractionalSelectivity(1.0));
31
32    application.addTupleMapping("client", "action_command", "actuation_signal", new
        FractionalSelectivity(1.0));

```

```

33     application.addTupleMapping("storage", "storage_request", "stored_data", new
        FractionalSelectivity(1.0));
34
35     final AppLoop loop1 = new AppLoop(new ArrayList<String>(){add("raw_data");add("client"
        );add("location_processing");add("processing_module");add("client");add("ACTUATORS"
        );}));
36     List<AppLoop> loops = new ArrayList<AppLoop>(){add(loop1);});
37     application.setLoops(loops);
38
39     return application;
40 }

```

Código-fonte 4 – Criação do modelo de aplicação seguindo o modelo DAG na estratégia baseada em nuvem

```

1 private static Application createApplication(String appId, int userId){
2     Application application = Application.createApplication(appId, userId);
3
4     application.addAppModule("client", 10);
5     application.addAppModule("location_processing", 10);
6     application.addAppModule("processing_module", 10);
7     application.addAppModule("context_coordinator", 10);
8     application.addAppModule("storage", 10);
9     application.addAppModule("notification", 10);
10
11     application.addAppEdge("raw_data", "client", 100, 300, "raw_data", Tuple.UP, AppEdge.SENSOR);
12     application.addAppEdge("client", "location_processing", 250, 150, "location_data",
        Tuple.UP, AppEdge.MODULE);
13     application.addAppEdge("client", "processing_module", 500, 250, "sensors_data", Tuple.
        UP, AppEdge.MODULE);
14     application.addAppEdge("location_processing", "processing_module", 500, 400, "
        processed_location", Tuple.UP, AppEdge.MODULE);
15
16     application.addAppEdge("processing_module", "context_coordinator", 1000, 300, 800, "
        processed_data", Tuple.UP, AppEdge.MODULE);
17
18     application.addAppEdge("context_coordinator", "notification", 100, 200, "
        event_notification", Tuple.UP, AppEdge.MODULE);
19     application.addAppEdge("context_coordinator", "storage", 150, 1000, "persistent_data",
        Tuple.UP, AppEdge.MODULE);
20     application.addAppEdge("processing_module", "client", 50, 50, "action_command", Tuple.
        DOWN, AppEdge.MODULE);
21     application.addAppEdge("client", "ACTUATORS", 50, 50, "actuation_signal", Tuple.DOWN,
        AppEdge.ACTUATOR);
22
23     application.addTupleMapping("client", "raw_data", "location_data", new

```

```
24     FractionalSelectivity(1.0));
    application.addTupleMapping("client", "raw_data", "sensors_data", new
25         FractionalSelectivity(1.0));
    application.addTupleMapping("location_processing", "location_data", "processed_location
26         ", new FractionalSelectivity(1.0));
    application.addTupleMapping("context_coordinator", "processed_data", "
27         event_notification", new FractionalSelectivity(1.0));
    application.addTupleMapping("context_coordinator", "processed_data", "persistent_data",
28         new FractionalSelectivity(1.0));
29
    application.addTupleMapping("processing_module", "sensors_data", "action_command", new
30         FractionalSelectivity(1.0));
    application.addTupleMapping("processing_module", "processed_location", "action_command"
31         , new FractionalSelectivity(1.0));
32
    application.addTupleMapping("client", "action_command", "actuation_signal", new
33         FractionalSelectivity(1.0));
    application.addTupleMapping("storage", "storage_request", "stored_data", new
34         FractionalSelectivity(1.0));
35
    final AppLoop loop1 = new AppLoop(new ArrayList<String>(){add("raw_data");add("client"
36         );add("location_processing");add("processing_module");add("client");add("ACTUATORS"
37         );});
    List<AppLoop> loops = new ArrayList<AppLoop>(){add(loop1);};
38
    application.setLoops(loops);
39
    return application;
40 }
```

APÊNDICE C – RESULTADOS COMPLETOS PARA TODOS OS CENÁRIOS SIMULADOS

A tabela 20 apresenta os resultados obtidos para todos os cenários simulados no experimento, em termos de: latência, expressa em milissegundos (ms); custo energético em nuvem, expresso em Megajoule (MJ); custo de execução em nuvem, expresso em unidade monetária genérica (\$); e uso de rede, expresso em kilobyte (KB). A configuração do cenário simulado é especificado na coluna *configuração*, onde o primeiro número representa a quantidade de áreas, o segundo número representa a quantidade de residências por área e o terceiro número indica a quantidade de *mist nodes* por residência. A coluna *tipo* indica a estratégia utilizada no cenário de avaliação (baseada em fog - *fog-based* ou baseada em nuvem - *cloud-based*).

Tabela 20 – Resultados obtidos em todos os cenários simulados

Tipo	Configuração	Latência	Custo Energético	Custo de Execução	Uso de Rede
<i>fog-based</i>	2-8-3	22,77 ms	5,36 MJ	0,49 \$	31,78 KB
	2-8-6	26,47 ms	5,37 MJ	0,54 \$	77,81 KB
	2-16-3	22,77 ms	5,37 MJ	0,54 \$	63,56 KB
	2-16-6	26,47 ms	5,37 MJ	0,62 \$	155,63 KB
	2-32-3	22,77 ms	5,37 MJ	0,62 \$	127,13 KB
	2-32-6	26,47 ms	5,38 MJ	0,80 \$	311,25 KB
	2-64-3	22,77 ms	5,38 MJ	0,80 \$	254,25 KB
	2-64-6	26,47 ms	5,41 MJ	1,15 \$	622,50 KB
<i>cloud-based</i>	2-8-3	222,80 ms	5,54 MJ	3,06 \$	197,66 KB
	2-8-6	224,96 ms	5,71 MJ	5,42 \$	375,55 KB
	2-16-3	225,20 ms	5,73 MJ	5,68 \$	395,31 KB
	2-16-6	229,52 ms	6,06 MJ	10,33 \$	749,68 KB
	2-32-3	230,00 ms	6,09 MJ	10,84 \$	789,06 KB
	2-32-6	386,26 ms	6,49 MJ	16,45 \$	1.227,15 KB
	2-64-3	421,22 ms	6,49 MJ	16,45 \$	1.227,15 KB
	2-64-6	735,63 ms	6,49 MJ	16,45 \$	1.227,15 KB
<i>fog-based</i>	4-8-3	22,77 ms	5,36 MJ	0,51 \$	63,56 KB
	4-8-6	26,47 ms	5,37 MJ	0,58 \$	155,63 KB
	4-16-3	22,77 ms	5,37 MJ	0,58 \$	127,13 KB
	4-16-6	26,47 ms	5,38 MJ	0,71 \$	311,25 KB
	4-32-3	22,77 ms	5,38 MJ	0,71 \$	254,25 KB
	4-32-6	26,47 ms	5,40 MJ	0,97 \$	622,50 KB

Continua na próxima página

Tabela 20 – Continuação da tabela

Tipo	Configuração	Latência	Custo Energético	Custo de Execução	Uso de Rede
	4-64-3	22,77 ms	5,40 MJ	0,97 \$	508,50 KB
	4-64-6	26,47 ms	5,43 MJ	1,50 \$	1.245,00 KB
<i>cloud-based</i>	4-8-3	225,20 ms	5,73 MJ	5,68 \$	395,31 KB
	4-8-6	229,52 ms	6,06 MJ	10,33 \$	749,68 KB
	4-16-3	230,00 ms	6,09 MJ	10,84 \$	789,06 KB
	4-16-6	386,26 ms	6,49 MJ	16,45 \$	1.227,15 KB
	4-32-3	421,22 ms	6,49 MJ	16,45 \$	1.227,15 KB
	4-32-6	735,63 ms	6,49 MJ	16,45 \$	1.227,15 KB
	4-64-3	753,15 ms	6,49 MJ	16,45 \$	1.227,15 KB
	4-64-6	910,18 ms	6,49 MJ	16,45 \$	1.227,15 KB
<i>fog-based</i>	8-8-3	22,77 ms	5,37 MJ	0,58 \$	127,13 KB
	8-8-6	26,47 ms	5,38 MJ	0,71 \$	311,25 KB
	8-16-3	22,77 ms	5,38 MJ	0,71 \$	254,25 KB
	8-16-6	26,47 ms	5,40 MJ	0,97 \$	622,50 KB
	8-32-3	22,77 ms	5,40 MJ	0,97 \$	508,50 KB
	8-32-6	26,47 ms	5,43 MJ	1,50 \$	1.245,00 KB
	8-64-3	22,77 ms	5,43 MJ	1,50 \$	1.017,00 KB
	8-64-6	26,47 ms	5,51 MJ	2,55 \$	2.490,00 KB
<i>cloud-based</i>	8-8-3	230,00 ms	6,09 MJ	10,84 \$	789,06 KB
	8-8-6	386,26 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-16-3	421,22 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-16-6	735,63 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-32-3	753,15 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-32-6	910,18 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-64-3	919,12 ms	6,49 MJ	16,45 \$	1.227,15 KB
	8-64-6	997,45 ms	6,49 MJ	16,45 \$	1.227,15 KB
<i>fog-based</i>	16-8-3	22,77 ms	5,39 MJ	0,86 \$	254,25 KB
	16-8-6	26,47 ms	5,41 MJ	1,12 \$	622,50 KB
	16-16-3	22,77 ms	5,41 MJ	1,12 \$	508,50 KB
	16-16-6	26,47 ms	5,44 MJ	1,60 \$	1.245,00 KB
	16-32-3	22,77 ms	5,44 MJ	1,60 \$	1.017,00 KB
	16-32-6	26,47 ms	5,51 MJ	2,56 \$	2.490,00 KB
	16-64-3	22,77 ms	5,51 MJ	2,56 \$	2.034,00 KB
	16-64-6	26,47 ms	5,64 MJ	4,48 \$	4.980,00 KB
<i>cloud-based</i>	16-8-3	421,22 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-8-6	735,63 ms	6,49 MJ	16,45 \$	1.227,15 KB

Continua na próxima página

Tabela 20 – *Continuação da tabela*

Tipo	Configuração	Latência	Custo Energético	Custo de Execução	Uso de Rede
	16-16-3	753,15 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-16-6	910,18 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-32-3	919,12 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-32-6	997,45 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-64-3	1.001,52 ms	6,49 MJ	16,45 \$	1.227,15 KB
	16-64-6	1.041,09 ms	6,49 MJ	16,45 \$	1.227,15 KB

Fonte: Elaborado pelo Autor.