

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

**SBI<sub>id</sub>C-MPH: SISTEMA BASEADO EM INTERNET  
DAS COISAS PARA O MONITORAMENTO DE  
PACIENTES COM HIPERTENSÃO**

**RÔNITTI JUNER DA SILVA RODRIGUES**

**ORIENTADOR: PROF. DR. WANDERLEY LOPES DE SOUZA**

São Carlos - SP  
Fevereiro /2022

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

**SBIIdC-MPH: SISTEMA BASEADO EM INTERNET  
DAS COISAS PARA O MONITORAMENTO DE  
PACIENTES COM HIPERTENSÃO**

**RÔNITTI JUNER DA SILVA RODRIGUES**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Sistemas Distribuídos, Arquiteturas e Redes de Computadores.

Orientador: Prof. Dr. Wanderley Lopes de Souza

São Carlos - SP  
Fevereiro/2022



# UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

## Folha de Aprovação

---

Defesa de Dissertação de Mestrado do candidato Ronitti Juner da Silva Rodrigues, realizada em 09/02/2022.

### Comissão Julgadora:

Prof. Dr. Wanderley Lopes de Souza (UFSCar)

Prof. Dr. Ricardo Rodrigues Ciferri (UFSCar)

Prof. Dr. Clever Ricardo Guareis de Farias (USP)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

*Dedico esse trabalho aos meus pais, Irineu e Marizete, aos meus irmãos Réulliner e Rallini, e a minha amada esposa Larissa. Vocês são muito importantes em minha vida!*

# AGRADECIMENTOS

A Deus, por ter me dado saúde e força necessária para superar as dificuldades do percurso e assim, obter essa conquista.

Aos meus pais, Irineu e Marizete, que são verdadeiros exemplos para mim, e com muito apoio e carinho, não mediram esforços para me ajudar sempre que precisei. Espero que, com meu esforço, consiga um dia retribuir ao menos uma parte do que já fizeram por mim.

Aos meus irmãos, pela amizade, alegria e constante apoio recebido durante o tempo que me dediquei a realizar este trabalho.

A minha amada esposa, Larissa, pelo incentivo, amor e companheirismo. Seu apoio em todos os momentos, sobretudo em minhas crises de ansiedade, foi fundamental para a conclusão deste trabalho. Agradeço imensamente por me ouvir, me aconselhar e por acreditar em mim, principalmente quando eu já não acreditava mais. Você é muito importante para mim.

Ao meu orientador, professor Dr. Wanderley Lopes de Souza, pela oportunidade concedida, por toda ajuda, compreensão e pelos ensinamentos repassados durante o desenvolvimento deste trabalho. Serão de grande valia para a minha vida.

Ao Programa de Pós-Graduação em Ciência da Computação (PPG-CC) da Universidade Federal de São Carlos (UFSCar), por fornecer os recursos humanos e a infraestrutura necessários ao desenvolvimento deste trabalho.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), através do Instituto Nacional de Ciência e Tecnologia (INCT) Medicina Assistida por Computação Científica (MACC), por fornecer os recursos financeiros necessários ao desenvolvimento deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Enfim, a todos que contribuíram direta ou indiretamente para a realização deste trabalho.

*"Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer"*

*Alan Turing.*

# RESUMO

As Doenças Crônicas Não Transmissíveis (DCNT) são um dos maiores desafios na Saúde, principalmente devido à sua magnitude e ao seu alto custo social. Os quatro grupos principais de DCNTs, as doenças cardiovasculares, o Câncer, o Diabetes e as doenças respiratórias crônicas, causam cerca de 70% das mortes no mundo, levando a óbito em torno de 41 milhões de pessoas a cada ano. As doenças cardiovasculares são a principal causa de morte prematura, sendo responsáveis por cerca de 17,9 milhões de pessoas anualmente, das quais 9,4 milhões são devidas à Hipertensão Arterial. Diversos fatores de risco contribuem para o desenvolvimento de DCNTs e o agravamento da saúde das pessoas. Alguns desses fatores, tais como o sedentarismo e a alimentação não saudável, podem ser alterados, pois estão relacionados ao estilo de vida. Para prevenir e controlar DCNTs, é necessário promover a mudança de hábitos e proporcionar um cuidado de saúde integral ao paciente. As tecnologias da Computação Ubíqua, Computação Pervasiva e, mais recentemente, Internet das Coisas (IdC), possibilitam o desenvolvimento de sistemas para o monitoramento à distância e em tempo real de pacientes acometidos com certas DCNTs, tais como a Hipertensão Arterial, o Diabetes, a Asma e a Obesidade. Neste sentido, foi desenvolvido o Sistema Baseado em Internet das Coisas para o Monitoramento de Pacientes com Hipertensão (SBIdC-MPH), que permite capturar sinais vitais do paciente, tais como pressão sanguínea, batimentos cardíacos e temperatura corpórea, via uma rede de sensores incorporadas a um dispositivo vestível semelhante a um relógio de pulso. Esses sinais são transmitidos ao dispositivo móvel do paciente para pré-processamento, sendo que os dados clínicos gerados são enviados a uma nuvem computacional. Esses dados por sua vez são processados na nuvem visando a sua apresentação, através de tabelas e gráficos, ao profissional de saúde responsável pelo paciente. O SBIdC-MPH permite também a geração de sinais de alertas, os quais são enviados ao paciente, ao profissional de saúde, ou até mesmo a pessoas designadas pelo paciente, sempre que situações críticas são detectadas.

**Palavras-chave:** Cuidado de Saúde Pervasivo, Doenças Crônicas Não Transmissíveis, Hipertensão Arterial, Sistemas de Monitoramento para o Cuidado de Saúde, Computação Ubíqua, Internet das Coisas, Redes de Sensores do Corpo Humano.

# ABSTRACT

Noncommunicable Diseases (NCDs) are one of the greatest challenges in Health, mainly due to their magnitude and their high social cost. The four main groups of NCDs, cardiovascular diseases, Cancer, Diabetes and chronic respiratory diseases, cause about 70% of deaths in the world, leading to the death of around 41 million people each year. Cardiovascular diseases are the leading cause of premature death, accounting for about 17.9 million people annually, of which 9.4 million are due to Hypertension. To prevent and control NCDs, it is necessary to promote a change in habits and provide comprehensive health care for the patient. The technologies of Ubiquitous Computing, Pervasive Computing and, more recently, the Internet of Things (IoT) enable the development of systems for remote and real-time monitoring of patients affected with certain NCDs, such as Hypertension, Diabetes, Asthma and Obesity. In this sense, the System Based on Internet of Things for Monitoring Patients with Hypertension (SBIdC-MPH) was developed, which allows for capturing the patient's vital signs, such as blood pressure, heart rate, and body temperature, via a network of built-in sensors to a wearable device similar to a wristwatch. These signals are transmitted to the patient's mobile device for processing, and the clinical data generated are sent to a computational cloud. These data, in turn, are treated in the cloud with a view to presenting them, through tables and graphics, to the health professional responsible for the patient. The SBIdC-MPH also allows the generation of alert signals, which are sent to the patient, the health professional, or even to people designated by the patient, whenever critical situations are detected.

**Keywords:** Pervasive Healthcare, Non-Communicable Diseases, Hypertension, Healthcare Monitoring Systems, Ubiquitous Computing, Internet of Things, Wireless Body Sensor Networks.

# LISTA DE FIGURAS

Figura 2.1 Monitoramento Remoto de Pacientes (adaptado de (WANG et al., 2013)) .....	23
Figura 3.1 Dimensões Computacionais (adaptado de (LYYTINEN; YOO, 2002)).....	31
Figura 3.2 Comunicação Cliente-API com GraphQL (adaptado de (BUNA, 2021)) ..	37
Figura 3.3 Modelo de Referência IdC (adaptado de (CISCO et al, 2014)) .....	40
Figura 3.4 Arquitetura de rede Wi-Fi 802.11 (adaptado de (KUROSE; ROSS, 2021)) .....	44
Figura 3.5 Arquitetura do BLE (AFANEH, 2018). .....	48
Figura 3.6 Modelo hierárquico de Perfil, Serviço e Característica .....	48
Figura 3.7 Cenário de uso de RSCH (adaptado de (ALI; SADEGHI; LIU, 2020)) .....	52
Figura 5.1 Ciclo TDD para a funcionalidade <i>Physiological Data Collection</i> .....	64
Figura 5.2 Visão geral do SBIdC-MPH.....	65
Figura 5.3 Arquitetura de hardware da Plataforma de Sensores.....	66
Figura 5.4 Esquema do hardware da Plataforma de Sensores .....	68
Figura 5.5 Protótipo da Plataforma de Sensores.....	68
Figura 5.6 Diagrama do filtro digital para a detecção do movimento do paciente .....	69
Figura 5.7 Diagrama de sequência do processo de registo e autenticação da aplicação no dispositivo móvel.....	71
Figura 5.8 Diagrama de Caso de Uso da Mobile App .....	72
Figura 5.9 Diagrama de Sequência para a funcionalidade Patient Data Analyze .....	74
Figura 5.10 Diagrama de classes para a funcionalidade Patient Data Analyze .....	75
Figura 5.11 Trecho do código fonte do método <code>executeAnalyze</code> .....	76
Figura 5.12 Interface do Usuário da Mobile App .....	77
Figura 5.13 Diagrama de Caso de Uso da Web App .....	78
Figura 5.14 Captura da tela de Visualização de Dados de paciente .....	79
Figura 5.15 Trechos do código do Schema para a API Server .....	80
Figura 5.16 Trecho do código de uma requisição GraphQL e resposta à mesma ....	81
Figura 6.1 Visão geral dos cenários de simulação .....	84
Figura 6.2 Ambiente de simulação.....	85

Figura 6.3 Tempo de Resposta Médio para o envio de dados clínicos .....	86
Figura 6.4 Tempo de Resposta Médio para a consulta de dados pessoais e clínicos .....	86

# LISTA DE TABELAS

Tabela 2.1 Classificação da Pressão Arterial.....	27
Tabela 3.1 Características IEEE 802.11 (adaptado de (KUROSE; ROSS, 2021))....	43
Tabela 4.1 Comparações entre o SBIdC-MPH e trabalhos correlatos .....	60
Tabela 6.1 Resultados da simulação relativos ao envio de dados clínicos .....	87
Tabela 6.2 Resultados da simulação relativos à recepção de dados pessoais e clínicos .....	87

# LISTA DE ABREVIATURAS E SIGLAS

- AHA** – *American Heart Association*
- AP** – *Access Point*
- API** – *Application Programming Interfaces*
- AVC** – *Acidente Vascular Cerebral*
- BC** – *Batimentos Cardíacos*
- BI** – *Business Intelligence*
- BLE** – *Bluetooth Low Energy*
- CoAP** – *Constrained Application Protocol*
- DCNT** – *Doenças Crônicas Não Transmissíveis*
- DTLS** – *Datagram Transport Layer Security*
- ECG** – *EletroCardioGrama*
- ERP** – *Enterprise Resource Planning*
- FFD** – *Full-Function Device*
- GAF** – *Grupos de Atividades Físicas*
- GAP** – *Generic Access Profile*
- GATT** – *Generic Attribute Profile*
- HA** – *Hipertensão Arterial*
- HTTP** – *Hypertext Transfer Protocol*
- HTTPS** – *Hyper Text Transfer Protocol Secure*
- IdC** – *Internet das Coisas*
- IIR** – *Infinite Impulse Response*
- IP** – *Internet Protocol*
- JSON** – *JavaScript Object Notation*
- JWT** – *JSON Web Token*
- M2M** – *Machine-to-Machine*
- MQTT** – *Message Queue Telemetry Transport*
- OP** – *Oxímetro de Pulso*
- PA** – *Pressão Arterial*
- PPG** – *PhotoPlethysmoGraphy*
- QoS** – *Quality of Service*

**REST** – *REpresentational State Transfer*

**RFD** – *Reduced-Function Device*

**RFID** – *Radio-Frequency IDentification*

**RNA** – Redes Neurais Artificiais

**RSCH** – Rede de Sensores do Corpo Humano

**RSSF** – Rede de Sensores Sem Fio

**SAF** – Sessões de Atividade Física

**SBIdC-MPH** – Sistema Baseado em Internet das Coisas para o Monitoramento de Pacientes com Hipertensão

**SIG** – *Special Interest Group*

**SMS** – *Short Message Service*

**SUS** – Sistema Único de Saúde

**TC** – Temperatura Cutânea

**TCP** – *Transmission Control Protocol*

**TDD** – *Test-Driven Development*

**TIC** – Tecnologias de Informação e Comunicação

**UI** – *User Interface*

**UML** – *Unified Modeling Language*

**URI** – *Uniform Resource Identifier*

**USF** – Unidades de Saúde da Família

**UUID** – *Universally Unique Identifier*

**WLAN** – *Wireless Local Area Networks*

**WPAN** – *Wireless Personal Area Networks*

**ZC** – *ZigBee Coordinator*

**ZED** – *ZigBee End Device*

**ZG** – *ZigBee Gateway*

**ZR** – *ZigBee Router*

**ZTC** – *ZigBee Trust Center*

# SUMÁRIO

<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>16</b>
1.1 Contexto .....	16
1.2 Motivação e Objetivos .....	17
1.3 Organização da Dissertação .....	18
<b>CAPÍTULO 2 - CUIDADO DE SAÚDE PERVASIVO.....</b>	<b>20</b>
2.1 Considerações Iniciais.....	20
2.2 Desafios na Saúde .....	21
2.3 Monitoramento Remoto de Pacientes .....	22
2.4 Doenças Crônicas Não Transmissíveis.....	24
2.5 Hipertensão Arterial.....	26
2.6 Considerações Finais .....	28
<b>CAPÍTULO 3 - INTERNET DAS COISAS.....</b>	<b>29</b>
3.1 Considerações Iniciais.....	29
3.2 Computações Ubíqua e Pervasiva e Internet das Coisas .....	30
3.2.1 Dispositivos .....	32
3.2.2 Tecnologias e Protocolos .....	34
3.2.2.1 HTTP .....	35
3.2.2.2 REST .....	35
3.2.2.3 GraphQL.....	36
3.2.2.4 MQTT .....	38
3.2.2.5 CoAP .....	39
3.2.3 Infraestrutura e Serviços .....	39
3.3 Redes de Comunicação sem Fio .....	42
3.3.1 Wi-Fi.....	43
3.3.2 Redes de Celular (3G/4G/5G) .....	44
3.3.3 ZigBee .....	45
3.3.4 Bluetooth Low Energy (BLE) .....	46
3.4 Redes de Sensores Sem Fio.....	49
3.5 Redes de Sensores do Corpo Humano.....	50

3.6 Internet das Coisas e o Cuidado de Saúde Pervasivo .....	52
3.7 Considerações Finais .....	53
<b>CAPÍTULO 4 - TRABALHOS CORRELATOS .....</b>	<b>54</b>
4.1 Considerações Iniciais.....	54
4.2 Sistemas de Monitoramento para o Cuidado de Saúde .....	55
4.3 Análise Comparativa .....	56
4.4 Considerações Finais .....	61
<b>CAPÍTULO 5 - SISTEMA BASEADO EM INTERNET DAS COISAS PARA O MONITORAMENTO DE PACIENTES COM HIPERTENSÃO .....</b>	<b>62</b>
5.1 Considerações Iniciais.....	62
5.2 Metodologia de Desenvolvimento .....	63
5.3 Visão Geral do Sistema.....	64
5.4 Camada de Sensores.....	66
5.4.1 Plataforma de Sensores .....	67
5.4.2 Pressão Arterial e Batimentos Cardíacos.....	69
5.4.3 Segurança dos Dados .....	70
5.5 Camada Fog.....	71
5.5.1 Aplicação Móvel .....	72
5.6 Camada da Nuvem .....	77
5.6.1 Aplicação Web .....	77
5.6.2 Servidor API .....	80
5.7 Considerações Finais .....	82
<b>CAPÍTULO 6 - EXPERIMENTO E ANÁLISE DOS RESULTADOS .....</b>	<b>83</b>
6.1 Considerações Iniciais.....	83
6.2 Descrição do Experimento .....	84
6.3 Resultados Obtidos .....	85
6.4 Considerações Finais .....	88
<b>CAPÍTULO 7 - CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b>	<b>89</b>
7.1 Limitações .....	90
7.2 Trabalhos Futuros .....	91
<b>REFERÊNCIAS.....</b>	<b>92</b>

<b>APÊNDICE A .....</b>	<b>99</b>
<b>APÊNDICE B .....</b>	<b>104</b>
<b>APÊNDICE C .....</b>	<b>105</b>

# Capítulo 1

## INTRODUÇÃO

---

*Este capítulo realiza uma introdução ao tema proposto neste projeto de mestrado, apresenta as motivações encontradas e os objetivos desejados. O capítulo 1 introdutório está estruturado da seguinte maneira: a seção 1.1 trata do contexto do trabalho; a seção 1.2 discorre sobre a motivação e os objetivos; e a seção 1.3 aborda a organização desta dissertação.*

### 1.1 Contexto

As Doenças Crônicas Não Transmissíveis (DCNT) são consideradas atualmente as principais causas de mortes e invalidez em todo o mundo, se tornando um dos maiores desafios de saúde e provocando preocupação mundial devido a sua magnitude e os altos custos sociais (WHO et al., 2018). Todo ano, morrem em todo o mundo cerca de 41 milhões de pessoas em decorrência de algum tipo de DCNT, correspondendo a mais de 70% do total de mortes mundiais, sendo  $\frac{3}{4}$  dessas mortes em países de baixa e média rendas (WHO, 2020; MALTA et al., 2016). As doenças cardiovasculares são as principais responsáveis pelas mortes, contabilizando 17,9 milhões de pessoas anualmente, seguido pelo câncer (9,3 milhões), doenças respiratórias crônicas (4,1 milhões) e por último o diabetes (1,5 milhões) (WHO et al., 2018).

A Hipertensão Arterial é uma das doenças cardiovasculares mais prevalentes, afetando 22% da população mundial adulta com mais de 18 anos, sendo a causa de morte de aproximadamente 9,4 milhões de pessoas todo ano (WHO et al., 2013). Além disso, essa doença é um forte fator de risco para outras doenças adquiridas

durante a vida, tais como doença coronária, hipertrofia ventricular esquerda, arritmias cardíacas, incluindo fibrilação atrial, acidente vascular cerebral e insuficiência renal (WHO et al., 2016; KJELDSEN, 2018). Caracterizada por níveis elevados e sustentados da pressão arterial sistólica maior ou igual a 140 mmHg e diastólica maior ou igual 90 mmHg, a Hipertensão arterial é considerada uma assassina silenciosa com sintomas não visíveis durante muitos anos, sendo normalmente descoberta quando surge uma complicação ou um órgão vital é lesionado (WHO et al., 2013).

Dessa forma, é muito importante o diagnóstico e o acompanhamento do paciente por um médico, a fim de manter a pressão arterial dentro dos níveis considerados normais, sendo que estes valores em adultos são 120 mmHg para a pressão sistólica e 80 mmHg para a pressão diastólica (WHO et al., 2013; KJELDSEN, 2018). No entanto, a tarefa do médico nem sempre é simples, pois para o diagnóstico e tratamento mais assertivos, é necessário levar em consideração os fatores de risco correlacionando-os com os dados de progresso da doença. Além disso, as aferições convencionais da pressão arterial são normalmente realizadas somente no momento da consulta e o armazenamento desses dados não ficam estruturados de modo a serem consultados quando necessário (SANDI; NUGRAHA; SUPANGKAT, 2013).

## 1.2 Motivação e Objetivos

Para um tratamento mais efetivo de pacientes com Hipertensão, é necessário um acompanhamento periódico das condições de saúde desses pacientes. A Hipertensão é uma das DCNTs que possibilita o monitoramento contínuo de pacientes, permitindo assim tanto aos profissionais de saúde quanto aos pacientes participar ativamente no controle da doença. Para tal, parâmetros fisiológicos, tais como frequência cardíaca, pressão arterial sistólica e diastólica, devem ser coletados periodicamente, via equipamentos ou sensores, armazenados, tratados e colocados à disposição para serem analisados.

A Computação Ubíqua e os recentes avanços das tecnologias da Internet das Coisas (IdC) podem contribuir para o desenvolvimento de sistemas de Telemedicina

com o objetivo de fornecer cuidados de saúde à distância, uma vez que é possível empregar diferentes tecnologias tais como dispositivos móveis, redes sem fio e sensores para o desenvolvimento de aplicações móveis e ambientes inteligentes. Desta forma, é possível aumentar a qualidade dos serviços de saúde e manter os custos reduzidos, permitir uma menor interação direta entre paciente e equipe médica, e fornecer o acesso aos serviços de saúde de modo onipresente (BRESNICK, 2015). Além disso, o cruzamento de dados providos de diferentes sensores pode contribuir para relatórios mais completos e, como consequência, diagnósticos médicos mais precisos e tratamentos mais efetivos.

Neste contexto, o projeto Sistema Baseado em Internet das Coisas para o Monitoramento de Pacientes com Hipertensão (SBIdC-MPH) teve por objetivos principais possibilitar o monitoramento à distância, contínuo e automático de pacientes com Hipertensão, através do emprego de tecnologias da IdC, Computação Ubíqua e Computação em Nuvem.

Para tal, o SBIdC-MPH foi projetado visando a capturar os sinais vitais do paciente, tais como pressão sanguínea, batimentos cardíacos, temperatura corpórea e movimento, via uma rede de sensores incorporadas a um dispositivo vestível semelhante a um relógio de pulso. Esses sinais são então transmitidos ao dispositivo móvel do paciente para pré-processamento, sendo que os dados clínicos gerados são enviados à uma nuvem computacional. Esses dados são por sua vez tratados na nuvem para apresentação através de tabelas e gráficos, os quais podem ser acessados pelos profissionais de saúde responsáveis pelo paciente. Além disso, esse sistema foi também projetado para permitir a geração de sinais de alertas, os quais são enviados ao paciente, ao profissional de saúde, ou até mesmo a pessoas designadas pelo paciente, sempre que situações críticas são detectadas.

### **1.3 Organização da Dissertação**

A sequência desta dissertação está assim organizada:

(a) Capítulo 2, introduz o conceito Cuidado de Saúde Pervasivo e discute como as tecnologias atuais podem contribuir para possibilitar a entrega de serviços

de cuidado à saúde a qualquer hora, em qualquer lugar e utilizando qualquer tipo de dispositivo;

(b) Capítulo 3, fornece uma visão geral do estado da arte das Computações Ubíqua e Pervasiva e da IdC e do seu emprego no Cuidado de Saúde Pervasivo;

(c) Capítulo 4, discute os trabalhos relacionados ao SBIdC-MPH, que foram considerados os mais relevantes na revisão bibliográfica realizada;

(d) Capítulo 5, discorre sobre o SBIdC-MPH, abordando o seu projeto e implementação, assim como a metodologia de desenvolvimento empregada;

(e) Capítulo 6, trata da avaliação do uso das tecnologias REST e GraphQL para a comunicação entre as aplicações do SBIdC-MPH; e

(f) Capítulo 7, apresenta as considerações finais e aponta para trabalhos futuros.

# Capítulo 2

## CUIDADO DE SAÚDE PERVASIVO

---

*Este capítulo introduz conceitos e tecnologias relacionados ao Cuidado de Saúde Pervasivo, focando no monitoramento remoto de pacientes. São discutidas e apresentadas algumas DCNTs passíveis de serem monitoradas à distância, em particular a Hipertensão Arterial.*

### 2.1 Considerações Iniciais

Atualmente a maioria dos países enfrentam o mesmo conjunto de desafios em relação ao Cuidado de Saúde: altos custos necessários ao cuidado de um número cada vez maior de idosos; um rápido aumento de doenças crônicas relacionadas ao “moderno estilo de vida”; uma demanda crescente por novos tratamentos médicos e tecnologias; e uma diminuição no número de médicos, enfermeiros e cuidadores de saúde (VARNESHEY, 2009; ARREDONDO; AVILES, 2015).

Em breve, o modelo vigente de Cuidado de Saúde não poderá atender a esses desafios, já que está concentrado em torno de pessoas altamente especializadas, localizadas em grandes hospitais e focando em casos agudos de tratamento. O Cuidado de Saúde precisa ser distribuído, a fim de produzir respostas mais rápidas e auxiliar o paciente a gerenciar a sua própria saúde. Um modelo de “Cuidado de Saúde Distribuído”, que permeie o cotidiano dos cidadãos, é muito mais apropriado para enfrentar os desafios citados.

De acordo com (BARDRAM; MIHAILIDIS; WAN, 2007), o objetivo do Cuidado de Saúde Pervasivo é habilitar esse modelo distribuído para o gerenciamento da

saúde e bem-estar, via o uso de Tecnologias de Informação e Comunicação (TICs), a fim de tornar o Cuidado de Saúde disponível em qualquer lugar, a qualquer momento e para qualquer um.

O Cuidado de Saúde Pervasivo foca explicitamente no uso de novas tecnologias, ferramentas e serviços que possibilitem ao paciente desempenhar um papel mais ativo no gerenciamento e tratamento de doenças, colocando-o no centro do processo de Cuidado de Saúde. Isso inclui suporte à realização de autogerenciamento, automonitoramento, autocuidado, esforços preventivos, cooperação entre paciente e instituições de saúde, cooperação entre lar e hospital, consulta e monitoramento remotos.

No decorrer deste capítulo será fornecida uma visão geral do Cuidado de Saúde Pervasivo da seguinte forma: a seção 2.2 apresenta alguns desafios na Saúde e como o emprego das TICs podem auxiliar no seu enfrentamento; a seção 2.3 trata do monitoramento remoto de pacientes, um dos principais instrumentos para alcançar o Cuidado de Saúde Pervasivo; a seção 2.4 discorre sobre as DCNTs, sobretudo as passíveis de monitoramento remoto; a seção 2.5 foca na DCNT Hipertensão Arterial, cujo os pacientes podem ser monitorados remotamente; e a seção 2.6 tece algumas considerações finais.

## **2.2 Desafios na Saúde**

De acordo com (VARSHNEY, 2009), apesar da introdução das TICs na Saúde terem proporcionado vários avanços, visando a tornar os seus processos mais eficientes e, conseqüentemente, melhorando a qualidade e o acesso aos seus serviços, vários problemas carecem de soluções, tais como: o número significativo de erros médicos que resultam em mortes e sequelas nos pacientes; o alto estresse dos profissionais de saúde, causado sobretudo pela sobrecarga de atendimento; e a cobertura parcial dos serviços de saúde, principalmente em áreas rurais e carentes. Esses problemas, combinados ao custo elevado e crescente dos serviços de saúde e ao aumento do número de idosos, geram vários desafios para formuladores de políticas públicas, prestadores de serviços, hospitais, seguradoras e pacientes.

O emprego de TICs na Saúde, particularmente das tecnologias da Computação Ubíqua e IdC, auxiliam na implantação de um modelo de Cuidado de Saúde Pervasivo, contribuindo assim para a redução de custos na saúde, a ampliação das áreas de cobertura dos serviços, a redução de erros médicos e o aumento da produtividade dos profissionais de saúde. Tais benefícios são alcançados, na medida em que as informações de saúde são disponibilizadas a qualquer hora e em qualquer lugar, via o uso de dispositivos sofisticados e de redes de comunicação amplamente implantadas (VARSHNEY, 2005).

De acordo com (BOYE, 2008), o Cuidado de Saúde Pervasivo deve ser suportado por uma estrutura computacional invisível, onipresente e interoperável, visando incrementar a saúde, a qualidade de vida e o bem-estar de cada indivíduo, seja este saudável ou não. Tais cuidados incluem: a prevenção; o monitoramento de curto prazo (e.g., o paciente em seu domicílio) e longo prazo (e.g., acompanhamento de idosos); a detecção e gerenciamento de incidentes; a intervenção em caso de emergência; e o tratamento. Dentre os principais instrumentos do Cuidado de Saúde Pervasivo destaca-se o monitoramento remoto de pacientes.

### **2.3 Monitoramento Remoto de Pacientes**

Há alguns anos, pacientes que necessitavam de tratamentos prolongados geralmente precisavam permanecer em ambientes confinados, tais como hospitais e casas de saúde. Hoje, graças a diversos avanços tecnológicos, envolvendo por exemplo sensores, redes sem fio, dispositivos móveis, Computação em Nuvem e Aprendizado de Máquina, diversos sistemas computacionais de suporte ao Cuidado de Saúde foram criados, dentre os quais os de monitoramento remoto de pacientes. Tais sistemas usualmente realizam um monitoramento contínuo de sinais vitais de pacientes, possibilitando a estes receberem os cuidados de profissionais de saúde e serviços de provedores de saúde, independentemente de suas localizações. Dessa forma, indivíduos acometidos de doenças crônicas e prolongadas, sobretudo os idosos, e que necessitam de cuidados contínuos, podem viver no conforto de suas casas e com maior qualidade de vida (BALAS, 2020).

De acordo com (VARSHNEY, 2009), o monitoramento remoto de pacientes visa automatizar processos no Cuidado de Saúde, a fim de possibilitar um atendimento mais seguro, bem-sucedido e humanizado aos pacientes, em qualquer hora e em qualquer lugar, e de forma mais sustentável para os serviços de saúde.

A Figura 2.1 ilustra um cenário típico de monitoramento remoto de pacientes, no qual sensores heterogêneos são dispostos no corpo do paciente, de forma a constituir uma Rede de Sensores do Corpo Humano (RSCH), a fim de coletar dados fisiológicos do paciente. Esses dados são transmitidos via tecnologias de comunicação sem fio (e.g., Bluetooth, ZigBee, Wi-Fi) a um dispositivo móvel, que atua como um controlador local, podendo realizar um pré-processamento nesses dados, e os encaminha via Internet a uma nuvem computacional. Tais dados são processados e armazenados na nuvem, ficando disponíveis para consultas posteriores. A partir do histórico dos dados coletados, análises podem ser realizadas de modo automático e, em casos de emergência, alertas podem ser enviados aos profissionais de saúde, aos provedores de serviços de saúde, ou até mesmo à ambulâncias, solicitando atendimento imediato.

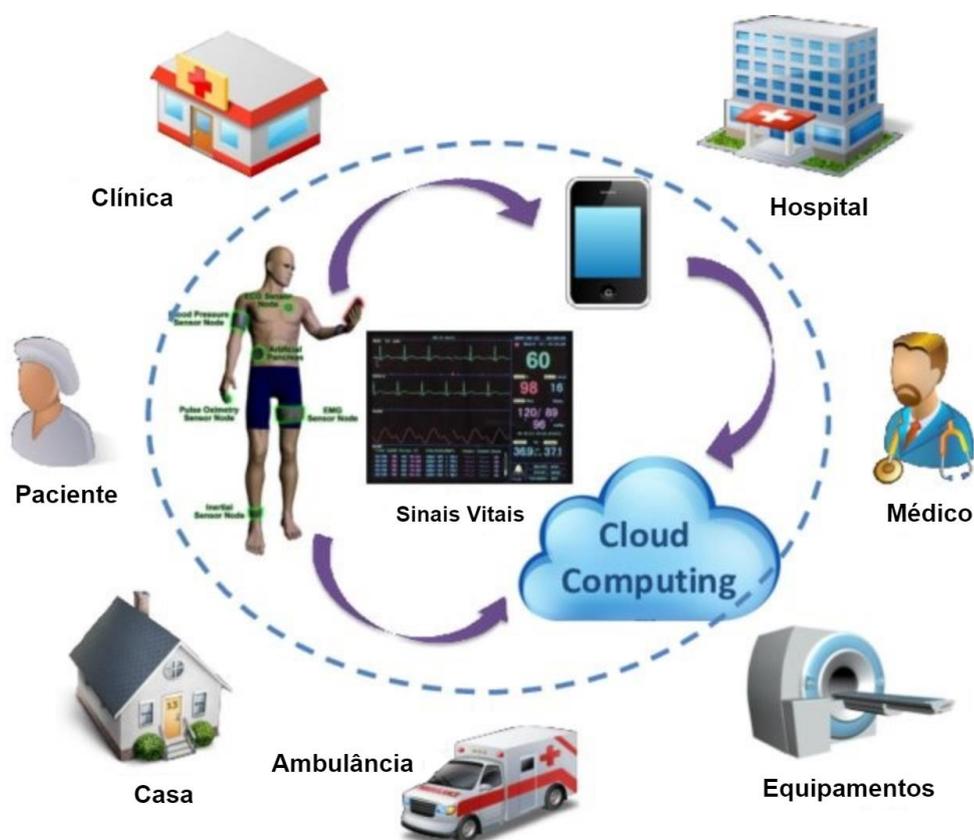


Figura 2.1 Monitoramento Remoto de Pacientes (adaptado de (WANG et al., 2013))

De acordo com (HUZOOREE; KUMAR KHEDO; JOONAS, 2019), sistemas de monitoramento na Saúde podem ser classificados, em função de seu uso, complexidade e necessidade de intervenção de profissionais de saúde, em quatro principais tipos:

(a) Auto Monitoramento, quando o sistema não envolve um cuidador externo e o paciente é o responsável pelo seu bem-estar e, conseqüentemente, depende de autocuidado, autodisciplina e autoconfiança do paciente. Geralmente são empregadas tecnologias visando à integração de sensores invasivos e não invasivos, dispositivos móveis e aplicações que possibilitam ao paciente coletar e monitorar seus próprios dados de saúde;

(b) Monitoramento Assistido, quando o sistema possibilita ao paciente requisitar a assistência de um profissional de saúde via mensagens, ligações, e-mails e alertas de emergência. Geralmente são empregadas tecnologias visando à integração de RSCHs, dispositivos móveis e um servidor remoto;

(c) Monitoramento Supervisionado, quando o sistema possibilita o envio e armazenamento periódicos na nuvem de dados fisiológicos do paciente, a fim de que possam ser consultados por profissionais de saúde, familiares e indivíduos devidamente autorizados. Geralmente, os profissionais de saúde podem intervir em casos de emergência e até mesmo configurar parâmetros nas RSCHs; e

(d) Monitoramento Contínuo, quando o sistema, além das funcionalidades descritas nas categorias anteriores, emprega uma abordagem de compartilhamento de dados em tempo real entre paciente e profissionais de saúde, utilizando técnicas inteligentes para correlacionar dados oriundos de diferentes sensores, visando a oferecer assistência ou ações preventivas aos pacientes.

## 2.4 Doenças Crônicas Não Transmissíveis

Segundo a OMS, as DCNTs têm origem não infecciosa, tendem a ser de longa duração e são resultado de uma combinação de fatores genéticos, fisiológicos, ambientais e de estilo de vida (WHO, 2020). As principais DCNTs, tais como doenças cardiovasculares, diabetes, doenças respiratórias crônicas e o câncer, resultam em conseqüências de longo prazo para a saúde das pessoas e

normalmente criam a necessidade de tratamentos e cuidados contínuos, provocando incapacidades e limitações em atividades de trabalho e lazer (WHO, 2020; WHO, 2021).

DCNTs surgem geralmente devido a hábitos de vida não saudáveis e ambientes físicos e sociais adversos. Conseqüentemente, fatores de risco, tais como dietas pobres, sedentarismo, consumo de tabaco, uso excessivo do álcool e estresse, devem ser monitorados e controlados. O desenvolvimento de DCNTs ocorre lentamente em alguns pacientes e rapidamente em outros, sendo que os sintomas são normalmente percebidos nos estágios já avançados dessas doenças.

A OMS estima que é possível prevenir até 80% dos casos de surgimento prematuro de doenças cardiovasculares, diabetes e de Acidente Vascular Cerebral (AVC) (WHO, 2020). Segundo (KRISTOFFERSSON; LINDÉN, 2020), esse objetivo pode ser alcançado a partir de duas ações: a redução dos fatores de risco relacionados ao estilo de vida; e agir assim que surgir os primeiros sinais, tais como pressão arterial elevada, altas taxas de glicose no sangue e colesterol “ruim”.

A falta de aderência ao tratamento é um problema comum aos pacientes com DCNTs, pois frequentemente não conseguem seguir as ordens terapêuticas, sejam estas o cumprimento de prescrição medicamentosa, realização de fisioterapia, ou mudanças de estilo de vida. Ainda que os altos custos do tratamento não sejam um problema, questões como o medo, o esquecimento e a falta de informações afetam diretamente a aderência do paciente. De acordo com (MCKENNEY, 1998), uma forma de lidar com essas questões é desenvolver um programa de aderência contendo os seguintes passos básicos:

(a) **Habilitação**, envolve a preparação do paciente a fim de que este siga adequadamente a prescrição médica, fornecendo informações de como a medicação deve ser tomada e criando uma relação de cuidado com o paciente;

(b) **Reforço**, envolve o fornecimento de feedbacks ao paciente sobre o seu estado de saúde ou sobre objetivos que foram cumpridos pelo mesmo;

(c) **Incentivo**, envolve esforços para lembrar o paciente de tomar as doses do medicamento no tempo correto; e

(d) **Resolução**, envolve a identificação de problemas específicos e a proposição de soluções para casos onde a habilitação, o reforço e o incentivo não foram suficientes para promover a aderência ao tratamento.

O emprego de tecnologias da Computação Ubíqua e IdC pode auxiliar em cada um desses passos. O paciente pode receber, em seu dispositivo móvel ou vestível, informações de profissionais de saúde relativas às medicações e/ou às adequações de estilo de vida a serem tomadas. Permite ao paciente receber rapidamente feedbacks de profissionais de saúde e/ou gerados automaticamente, a partir da análise dos dados relativos ao seu estado de saúde. Possibilita a criação de alertas para lembrar ao paciente o horário da medicação e/ou de realizar exercícios físicos. O monitoramento contínuo permite o fornecimento de uma grande quantidade e diversidade de informações, permitindo identificar possíveis causas que levam o paciente a não cumprir adequadamente o tratamento prescrito (AL-SIDDIQ, 2016).

Sensores vestíveis também desempenham um papel muito importante no monitoramento de DCNTs, na medida em que a coleta contínua e análise de vários tipos de dados fisiológicos, relacionados a essas doenças, possibilita aos profissionais de saúde obterem diagnósticos precocemente e rapidamente realizarem ações de prevenção e tratamento, que contribuem para redução da possibilidade de agravamento dessas doenças e de morte.

## 2.5 Hipertensão Arterial

A Hipertensão Arterial (HA) é uma DCNT caracterizada por níveis elevados e persistentes de pressão sanguínea nas artérias, com valores iguais ou maiores que 140/90 mmHg, e que fazem com que o coração exerça um esforço maior do que o normal para bombear sangue para todo o corpo. Os sintomas normalmente aparecem em situações em que a pressão arterial já atingiu valores muito altos e, nesse caso, podem ocorrer dores no peito, dor de cabeça, zumbido no ouvido, sangramento nasal, tonturas e visão embaçada (BRASIL, 2021).

Frequentemente assintomática, a HA tende a evoluir e provocar alterações estruturais e/ou funcionais em órgãos, tais como coração, cérebro, rins e vasos sanguíneos. Assim, torna-se um dos principais fatores de risco para a ocorrência de complicações, tais como AVC, infarto do miocárdio, aneurisma e insuficiência renal e cardíaca (BARROSO et al., 2020). A *American Heart Association (AHA)* classifica a

pressão arterial em cinco categorias, conforme apresentado na Tabela 2.1 (WHELTON et al., 2018).

**Tabela 2.1 Classificação da Pressão Arterial**

<b>Classificação</b>	<b>Sistólica mmHg</b>		<b>Diastólica mmHg</b>
Normal	Menor que 120	E	Menor que 80
Elevada	120-129	E	Menor que 80
Hipertensão Estágio 1	130-139	OU	80-89
Hipertensão Estágio 2	140 ou Maior	OU	90 ou Maior
Crise de Hipertensão	Maior que 180	E/OU	Maior que 120

Com base nessa classificação da pressão arterial, a AHA recomenda aos pacientes:

(a) Normal, continuem a manter os hábitos de vida saudáveis, tais como a prática regular de exercícios e a manutenção de uma dieta balanceada;

(b) Elevada, adotem medidas recomendadas, tais como adotar hábitos saudáveis de vida, pois estão correndo o risco de desenvolverem HA;

(c) Hipertensão Estágio 1, procurem um médico para acompanhamento, pois além de prescrever mudanças de hábitos, este poderá considerar o uso de medicações;

(d) Hipertensão Estágio 2, procurem o seu médico, pois provavelmente será necessário o uso de medicações para controlar a HA; e

(e) Crise de Hipertensão, se as leituras da pressão arterial atingirem esse nível e não se normalizarem em até cinco minutos, é necessário buscar atendimento médico imediato.

Mudanças no estilo de vida são fundamentais para o tratamento e prevenção da HA, pois mesmo doses progressivas de medicamentos podem não conseguir manter os níveis de pressão nos limites adequados. As principais mudanças a serem adotadas pelos pacientes para o tratamento da HA incluem a alimentação saudável, o controle do consumo de sal, a prática regular de atividades físicas, a manutenção de peso adequado, o abandono do tabagismo e o consumo não excessivo de álcool (BARROSO et al., 2020).

## **2.6 Considerações Finais**

Este capítulo apresentou uma síntese de um levantamento bibliográfico relativo ao Cuidado de Saúde Pervasivo, aos principais desafios atuais na Saúde e aos benefícios do monitoramento remoto de pacientes. Tratou também das DCNTs, sobretudo das passíveis de monitoramento remoto, apresentando os principais problemas enfrentados para o tratamento de pacientes com essas doenças e mostrando como o uso de tecnologias da Computação Ubíqua e IdC podem contribuir para superar tais problemas. Em particular, caracterizou a HA e discorreu sobre algumas recomendações para cada estágio dessa DCNT.

# Capítulo 3

## INTERNET DAS COISAS

---

*Este capítulo fornece uma visão geral das Computações Ubíqua e Pervasiva e da IdC, explorando os conceitos envolvidos, suas tecnologias e as aplicações relacionadas ao Cuidado de Saúde Pervasivo.*

### 3.1 Considerações Iniciais

As Computações Ubíqua e Pervasiva e os crescentes avanços da IdC têm transformado o cotidiano das pessoas, modificando a forma de interação e comunicação e permitindo a integração entre diferentes dispositivos de modo a otimizar as atividades nas mais diversas áreas. Cada vez mais acessíveis e com suporte a diversos sensores inteligentes, os dispositivos da IdC permitem a coleta de uma grande quantidade de sinais e dados a qualquer hora e em qualquer lugar e os torna disponíveis para análise futura, fornecendo subsídios para tomadas de decisão estratégicas, aumentando assim a produtividade e reduzindo custos.

No decorrer deste capítulo é fornecida uma visão geral das Computações Ubíqua e Pervasiva e da IdC da seguinte forma: a seção 3.2 introduz os conceitos envolvendo as Computações Ubíqua e Pervasiva e a IdC, e apresenta as suas principais tecnologias, softwares, protocolos e infraestruturas de serviços; a seção 3.3 trata das tecnologias de comunicação sem fio mais relevantes para os dispositivos da IdC e para a construção de Redes de Sensores do Corpo Humano (RSCH); a seção 3.4 conceitualiza as Redes de Sensores Sem Fio; a seção 3.5 aborda as Redes de Sensores do Corpo Humano; a seção 3.6 descreve uma

abordagem empregando IdC visando ao Cuidado de Saúde Pervasivo; e a seção 3.5 tece algumas considerações finais.

## 3.2 Computações Ubíqua e Pervasiva e Internet das Coisas

O termo Computação Ubíqua foi cunhado por Mark Weiser no início da década de 90 em seu trabalho intitulado *“The Computer for the 21 st Century”* (WEISER, 1991). Na sua visão, a Computação Ubíqua provocaria uma terceira “onda” revolucionária na história da Computação, as duas primeiras tendo sido provocadas pelos mainframes e computadores pessoais respectivamente, que causaria uma drástica alteração na forma como as pessoas usualmente interagem com os computadores e na forma como estes influenciavam até então em suas vidas.

Essa terceira onda seria resultado do uso crescente de objetos do cotidiano, tais como objetos pessoais (e.g., relógios, óculos, cartões), objetos do lar e escritórios (e.g., aquecedores, lâmpadas, fechaduras) e eletrodomésticos (e.g., fogões, geladeiras, máquinas de lavar), dotados de capacidades de processamento, armazenamento e comunicação, que permeariam as rotinas diárias de seus usuários.

Apesar de muitas vezes as Computações Pervasiva, Móvel e Ubíqua serem consideradas sinônimos, há autores que as distinguem conceitualmente (ADELSTEIN, 2005):

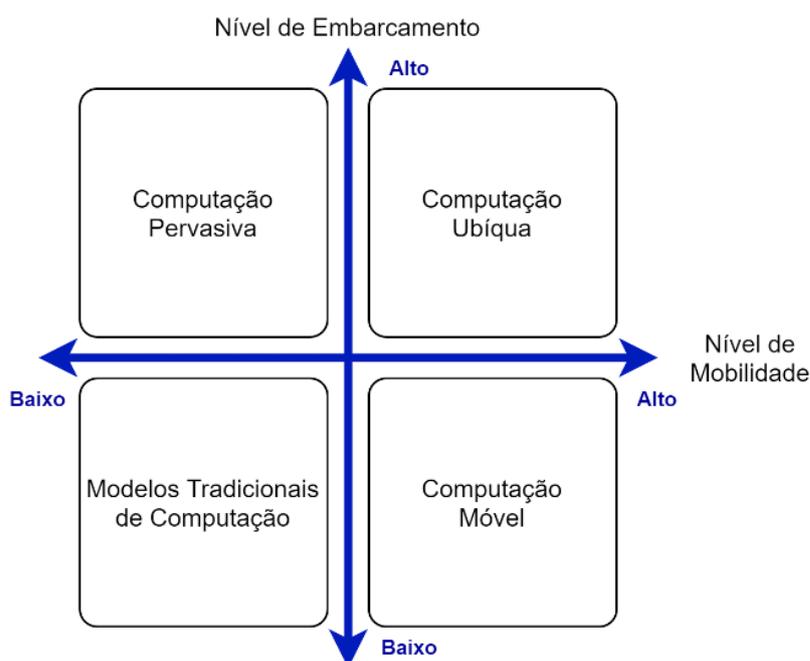
(a) Computação Pervasiva trata da integração de dispositivos computacionais ao ambiente, no qual estão inseridos, de forma imperceptível ao usuário. Esses dispositivos são capazes de obter informações relativas a esse ambiente e utilizá-las para dinamicamente gerar modelos computacionais, sendo também capazes de detectarem e interagirem com outros dispositivos computacionais e fornecerem serviços de modo “inteligente” aos seus usuários.

(b) Computação Móvel trata da capacidade de dispositivos computacionais móveis, tais como smartphones e tablets, de proverem acesso a informações e serviços em qualquer lugar e a qualquer hora. Apesar desses dispositivos normalmente possuírem capacidades de processamento, memória e energia inferiores aos desktops, estes são mais leves, portáteis e dispõem de interfaces de

comunicação com redes sem fio, possibilitando assim a conectividade com diversos serviços e facilitando a mobilidade do usuário.

(c) Computação Ubíqua engloba as duas computações anteriores, ou seja, trata da mobilidade e da integração de dispositivos computacionais ao ambiente e a outros dispositivos, permeando o cotidiano de seus usuários e agregando outras tecnologias para criar sistemas distribuídos, inteligentes e altamente conectados.

Em (LYYTINEN; YOO, 2002) são definidas 4 dimensões para a Computação em função do nível de embarcamento e mobilidade de seus dispositivos, conforme ilustrado na Figura 3.1. Os modelos tradicionais de Computação, tal como o empregado no desktop, apresentam baixo nível de embarcamento e mobilidade, enquanto que no sentido oposto a Computação Ubíqua apresenta alto nível de embarcamento e mobilidade. Essas características da Computação Ubíqua possibilitam aos seus dispositivos interagir com o ambiente e oferecer serviços aos seus usuários onipresentemente e de acordo com o contexto.



**Figura 3.1 Dimensões Computacionais (adaptado de (LYYTINEN; YOO, 2002))**

De acordo com (PEIRIS, 2017), a Computação Ubíqua foi enriquecida pela adoção em larga escala dos avanços tecnológicos em Computação nas últimas duas décadas. Tais avanços permitiram o desenvolvimento de dispositivos cada vez menores e com maior capacidade de processamento, armazenamento e comunicação, os quais puderam ser embarcados em diversos objetos do cotidiano e

serem interconectados via Internet mais efetivamente e com menor custo, originando o paradigma IdC.

O termo IdC foi cunhado por Kevin Ashton em 1999, para descrever uma rede de objetos do mundo físico, denominados “coisas”, que estão conectados à Internet via sensores onipresentes (ASHTON, 2009). Na IdC, dispositivos computacionais podem ser incorporados a outras interfaces, quer sejam máquinas ou humanos, podem ser conectados via redes com ou sem fio à Internet, e podem captar dados contextuais do ambiente onde estão inseridos, a fim de fornecerem novos serviços e modelos de negócios digitais (CHAUDHURI, 2019).

O propósito principal da IdC é fornecer uma ampla infraestrutura tecnológica, que facilite a troca de informação entre objetos de uma forma segura e confiável (WEBER, 2010). Essa estrutura envolve o emprego de uma grande variedade de dispositivos, softwares, protocolos e serviços.

### **3.2.1 Dispositivos**

Os dispositivos IdC podem ser embarcados em praticamente qualquer objeto do nosso cotidiano, sendo que de acordo com (ALAM; SHAKIL; KHAN, 2020) estes possuem componentes básicos tais como:

(a) Sensores, que são componentes eletrônicos capazes de produzir sinais elétricos, ópticos ou digitais a partir de algum evento ou condição física do ambiente no qual o dispositivo IdC está inserido. Os sinais ou dados mensurados por estes sensores estão diretamente relacionados à função desempenhada pelo dispositivo. Por exemplo, na função monitoramento da saúde de pacientes o dispositivo pode conter sensores para coletar sinais vitais dos pacientes, tais como batimentos cardíacos, pressão arterial e temperatura;

(b) Atuadores, que são componentes mecânicos capazes de executar alguma ação no mundo físico, decorrente do processamento e análise dos dados obtidos pelos sensores. Por exemplo, uma bomba de insulina pode atuar injetando automaticamente pequenas doses de insulina sob a pele do paciente, sendo que essas doses são calculadas com base na análise dos dados de níveis de açúcar no sangue coletado via sensores;

(c) Memórias, que permitem o armazenamento nos dispositivos IdC dos dados coletados via sensores, a fim de que esses dados possam estar disponíveis para posterior análise;

(d) Unidades de Processamento, que possibilitam aos dispositivos IdC processar e analisar os dados adquiridos via sensores. Além desta função básica, podem também desempenhar outras funções específicas, como por exemplo o controle de sistemas de alimentação e comunicação. Em grande parte dos dispositivos IdC, um microcontrolador é empregado para desempenhar o papel da unidade de processamento, devido ao seu tamanho reduzido, baixo consumo de energia, flexibilidade, simplicidade de programação e baixo custo;

(e) Unidades de Comunicação, que permitem a comunicação entre os dispositivos IdC e a Internet visando ao compartilhamento de informação. As formas de comunicação podem ser tanto via cabos quanto via redes sem fio, sendo que a segunda forma é preferível, devido ao seu menor custo e maior facilidade de implantação em diversos tipos de infraestrutura;

(f) Fontes de Energia, que fornecem a energia necessária ao funcionamento dos dispositivos IdC. Existem diversos tipos de fonte empregadas atualmente, sendo que as mais comuns são: bateria, energia solar, energia eólica e alimentação AC/DC. A escolha do tipo de fonte a ser empregada varia de acordo com os cenários de uso do dispositivo IdC, podendo até mais de um tipo ser utilizado.

As configurações dos componentes e as tecnologias empregadas na construção de um dispositivo IdC variam de acordo com a aplicação, seus requisitos e o ambiente no qual este será inserido. A prototipação física de dispositivos IdC tem sido facilitada pelo surgimento de diferentes plataformas, sendo que Arduino, ESP32 e Raspberry Pi estão dentre as mais populares atualmente. Tais plataformas usualmente dispõem da maioria dos componentes básicos IdC necessários ao projeto diferentes tipos de aplicações.

Arduino é uma plataforma *open-source* de hardware e software lançada em 2005, tendo como principais objetivos reduzir custos de prototipação e simplificar a criação de dispositivos e/ou objetos para ambientes interativos. Essa plataforma dispõe de um microcontrolador, memória e portas de entrada e saída, e pode ser facilmente integrada a módulos de sensores e/ou atuadores para a construção de dispositivos IdC. O software embarcado é na linguagem C ou C++, sendo que o

Ambiente de Desenvolvimento Integrado (IDE), desenvolvido para essa plataforma, pode ser facilmente carregado via um cabo USB (ARDUÍNO, 2020).

ESP 32 é um microcontrolador de baixo custo e consumo de energia, projetado para o desenvolvimento de dispositivos vestíveis e aplicações IdC. Este dispõe de redes WiFi 2.4GHz e Bluetooth, um processador de 32bits com dois núcleos, o qual permite o uso de softwares embarcados com capacidades de multitarefas. Esses softwares podem ser desenvolvidos na linguagem LUA ou ainda C e C++ utilizando a IDE da plataforma Arduino (ESPRESSIF, 2021).

Raspberry Pi é um pequeno dispositivo, o qual dispõe de maiores recursos computacionais quando comparado ao Arduino e ao ESP32, sendo capaz de executar sistemas operacionais, tais como Debian ou Fedora, a partir de um cartão SD. Este possui várias portas de entrada e saída, para a comunicação com sensores e/ou atuadores, e uma interface Ethernet para a comunicação em rede. As suas aplicações podem desenvolvidas nas linguagens Python, C, C++, Java e Ruby, e esta pode usar outros circuitos ou módulos de outros fabricantes para obter dados ou executar alguma ação no ambiente no qual está inserida (RASPBERRY PI).

Além das plataformas descritas, há no mercado uma grande diversidade de sensores, que podem ser facilmente integrados aos dispositivos de IdC. Como cada sensor é indicado para aplicações específicas, é necessário um estudo prévio de suas características e de sua compatibilidade à plataforma na qual será integrado.

### 3.2.2 Tecnologias e Protocolos

A conectividade em ambientes IdC é considerada a sua espinha dorsal e afeta diretamente o projeto de seus dispositivos. Atualmente, há uma grande variedade de protocolos que definem como dispositivos IdC devem comunicar-se diretamente, ou via um dispositivo intermediário denominado *gateway*, com serviços em uma nuvem computacional. *Hypertext Transfer Protocol (HTTP)*, *REpresentational State Transfer (REST)*, *GraphQL*, *Message Queue Telemetry Transport (MQTT)* e *Constrained Application Protocol (CoAP)* estão dentre os principais tecnologias e protocolos empregados em tais dispositivos ou em *Application Programming Interfaces (API)* que fazem parte do ambiente.

### 3.2.2.1 HTTP

Apesar de diversos projetos de IdC usarem o HTTP, este foi originalmente concebido para a Web como um protocolo do tipo *request-response*, operando sobre a pilha de protocolos *Transmission Control Protocol (TCP)/Internet Protocol (IP)* (TANENBAUM; FEAMSTER; WETHERALL, 2021). Tornou-se um dos protocolos mais usados na Internet, pois permite aos navegadores realizarem requisições a um servidor Web e a este responder com o conteúdo que lhe foi requisitado.

Apesar de bastante confiável, a comunicação via HTTP exige uma conexão e desconexão a cada requisição do cliente e a consequente troca de uma série de informações inseridas no cabeçalho das mensagens, o que acaba gerando uma sobrecarga e um consumo exagerado de recursos da rede e dos dispositivos IdC (YOKOTANI; SASAKI, 2016). Essas características fazem com que o HTTP não seja o mais adequado aos dispositivos IdC, uma vez que estes possuem poucos recursos computacionais e de energia. Além disso, o HTTP foi concebido para a comunicação entre dois dispositivos, sendo que em ambientes IdC majoritariamente é necessária a troca simultânea de mensagens entre diversos dispositivos (NIKOLOV, 2020).

### 3.2.2.2 REST

Trata-se de uma arquitetura uniforme, com interfaces escaláveis, independentes de tecnologia e que engloba um conjunto de restrições para a construção de APIs (WILDE; PAUTASSO, 2011). REST utiliza uma semântica direta de acesso ao serviço, o que permite interagir com recursos disponíveis em um servidor, onde cada recurso possui um identificador, o tipo de dados e suporta um conjunto de ações (FIELDING; TAYLOR, 2002).

REST é usualmente empregado em conjunto com o HTTP, sendo que neste caso é utilizado o *Uniform Resource Identifier (URI)*, o tipo de dados é definido no cabeçalho *Content-Type* da requisição e as ações são compostas por métodos HTTP, tais como POST, GET, UPDATE e DELETE, os quais são usados respectivamente para criar, consultar, alterar ou excluir um recurso no servidor (MASSÉ, 2012). O formato de dados mais empregado na construção de APIs REST é o *JavaScript Object Notation (JSON)*, uma vez que este é leve, de fácil leitura para humanos e suporta diferentes tipos de dados.

A forma simples de acesso a recursos do REST possibilita a integração com diferentes aplicações, uma maior confiabilidade e escalabilidade, e facilita o desenvolvimento de aplicações para diferentes plataformas. Entretanto, há também alguns problemas: dificuldades para o versionamento de API; necessidade de realizar requisições a múltiplos *endpoints* para recuperar diferentes recursos; e uma vez construídas APIs para atender diferentes aplicações (e.g. Mobile, Web, Desktop), a consulta a um determinado recurso pode trazer informações que não são necessárias a uma determinada aplicação e que aumentam o tráfego na rede.

O uso do REST tem facilitado a implantação de sistemas IdC, na medida em que possibilita a criação de APIs em servidores na nuvem ou na comunicação entre os próprios dispositivos IDC. Este último caso, deve-se ao fato de um sensor e suas informações poderem ser facilmente representados como um recurso na Web, o que permite a comunicação entre dispositivos e sistemas via o HTTP. Por outro lado, o uso conjunto com o HTTP, em dispositivos com recursos computacionais e de energia limitados, não é o ideal, sendo que outros protocolos mais restritivos devem ser empregados.

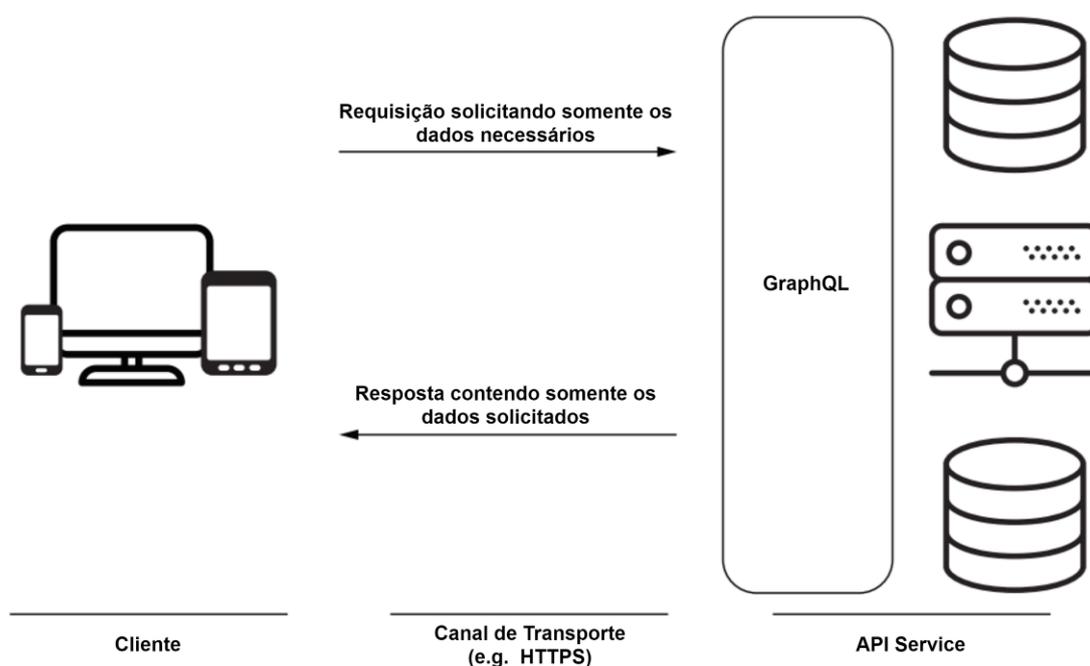
### 3.2.2.3 GraphQL

Trata-se de uma linguagem de consulta/manipulação de dados, desenvolvida em 2012 pelo Facebook para ser uma alternativa ao REST, e que foi disponibilizada publicamente em 2015 (BRITO; VALENTE, 2020). GraphQL busca representar dados do mundo real em uma estrutura do tipo grafo, já que praticamente qualquer modelo de dados é um grafo com vários objetos e relações entre os mesmos. Essa linguagem foi projetada para ser declarativa, intuitiva, flexível e eficiente, a fim de possibilitar a uma aplicação requisitar vários recursos em uma única requisição à API e receber somente os dados necessários à mesma (BUNA, 2021).

GraphQL é considerada mais eficiente que o REST, uma vez que consegue mitigar as seguintes inflexibilidades e complexidades associadas ao REST: requisições HTTP complexas, quando há necessidade de múltiplas requisições a diferentes *endpoints* para a consulta de diversos recursos; *overfetching* de dados, quando há o retorno de mais dados que o requerido pela aplicação; *underfetching* de dados, quando um *endpoint* não fornece todos os dados requeridos pela aplicação; e versionamento, quando há necessidade de criar uma nova versão de API sempre que novas funcionalidades são requeridas, o que torna onerosa a geração e

manutenção da documentação para esse grande número de APIs (MUKHIYA et al., 2019; JEON et al. 2019).

Visando a superar essas limitações do REST, GraphQL possibilita a construção de APIs em um alto nível de abstração, a partir de *Schemas* fortemente tipificados. Esta fornece um único *endpoint*, através do qual as aplicações clientes podem obter, em uma única requisição, todos e somente os dados solicitados. Isso permite manter uma única versão de API em constante evolução, adicionando-se novos campos ou descontinuando campos antigos, sem afetar as aplicações clientes já existentes. A Figura 3.2 ilustra um cenário típico de comunicação entre uma aplicação Cliente e uma API para consulta de dados.



**Figura 3.2 Comunicação Cliente-API com GraphQL (adaptado de (BUNA, 2021))**

Conforme ilustrado na Figura 3.2, Cliente envia à API uma requisição *Hyper Text Transfer Protocol Secure (HTTPS)*, contendo uma descrição no formato JSON de todos os recursos e dados desejados. API processa essa requisição, busca as informações no banco de dados ou em serviços de terceiros, monta e envia ao Cliente uma resposta também no formato JSON, contendo somente os recursos e dados requeridos.

GraphQL pode ser usado por dispositivos IdC na consulta e/ou envio de dados ao comunicarem-se diretamente com uma API GraphQL, já que a execução do lado servidor demanda mais recursos computacionais do que os usualmente

disponíveis nesses dispositivos. Atualmente, o seu uso tem sido mais indicado para aplicações na nuvem comunicando-se com aplicações de sistemas IdC.

### 3.2.2.4 MQTT

Trata-se de um protocolo para comunicação *Machine-to-Machine (M2M)* leve e altamente escalável, projetado para atender às restrições computacionais dos dispositivos IdC em ambientes com alta latência e baixa confiabilidade, cuja largura de banda de rede é usualmente um recurso limitado (HILLAR, 2017).

MQTT utiliza uma arquitetura cliente/servidor e emprega, como mecanismo básico de funcionamento, o conceito *Publisher/Subscriber* no qual há dois papéis principais: *Broker* e *Client*. O primeiro papel corresponde a um servidor que intermedia a troca de mensagens entre os *Client*, recebendo os dados publicados, tratando-os e os encaminhando aos *Clients* que se inscreveram para recebê-los. O segundo papel corresponde a um cliente, geralmente um dispositivo IdC, que atua postando os dados coletados pelos sensores (*Publisher*) ou recebendo dados provenientes de outros dispositivos (*Subscriber*) (YANG, 2021).

Configurada no Broker MQTT, a segurança é garantida via a autenticação de usuário e senha dos clientes no momento da conexão, sendo que toda troca de mensagens emprega criptografia do tipo TLS/SSL de ponta a ponta (NIKOLOV, 2020). MQTT também implementa mecanismos de *Quality of Service (QoS)* em três modos (HILLAR, 2017; OASIS, 2019):

(a) QoS 0, a mensagem é transferida uma única vez, sem qualquer feedback de recebimento pelo cliente ou armazenamento para posterior retransmissão em caso de falha. A transferência das mensagens é mais rápida, no entanto menos segura, pois a mesma pode ser pedida em caso de desconexão do cliente;

(b) QoS 1, garante a entrega da mensagem pelo menos uma vez, sendo que após o envio da mensagem é aguardado um tempo limite para o recebimento de um feedback e, em caso negativo, a mensagem é retransmitida. Portanto, pode ocorrer que a mensagem seja entregue e processada várias vezes pelo dispositivo IdC; e

(c) QoS 2, garante a entrega da mensagem uma única vez, sendo que a mensagem é armazenada, tanto no emissor quanto no receptor, até que esta seja efetivamente processada. A comunicação é mais lenta, comparada às dos modos

anteriores, mas a entrega é garantida pois é necessário um feedback tanto do emissor quanto do receptor.

Devido às vantagens do MQTT, quanto aos baixos consumos de memória, de processamento para envio de mensagens e de banda de rede, esse protocolo tem sido amplamente difundido para uso em dispositivos IdC, sendo que atualmente possui diversas implementações, tanto do Broker para diferentes sistemas operacionais, quanto do Cliente para diferentes plataformas de prototipagem.

### 3.2.2.5 CoAP

Trata-se de um protocolo para comunicação M2M entre nós e redes restritas, tal qual a IdC, que utiliza uma arquitetura semelhante à do REST, mas que opera sobre o protocolo UDP (BORMANN; CASTELLANI; SHELBY, 2012). De acordo com (IQBAL et al., 2020), CoAP resolve os problemas de comunicação do HTTP e da arquitetura REST para uso em dispositivos restritos, na medida em que possui as seguintes características:

- (a) Uso de URI;
- (b) Suporte à *Content-Type*;
- (c) Baixa complexidade para análise e processamento das mensagens;
- (d) Troca de mensagens de modo assíncrono;
- (e) uso de UDP para suporte ao envio de mensagens *unicast* e *broadcast*; e
- (f) Segurança garantida pelo *Datagram Transport Layer Security* (DTLS).

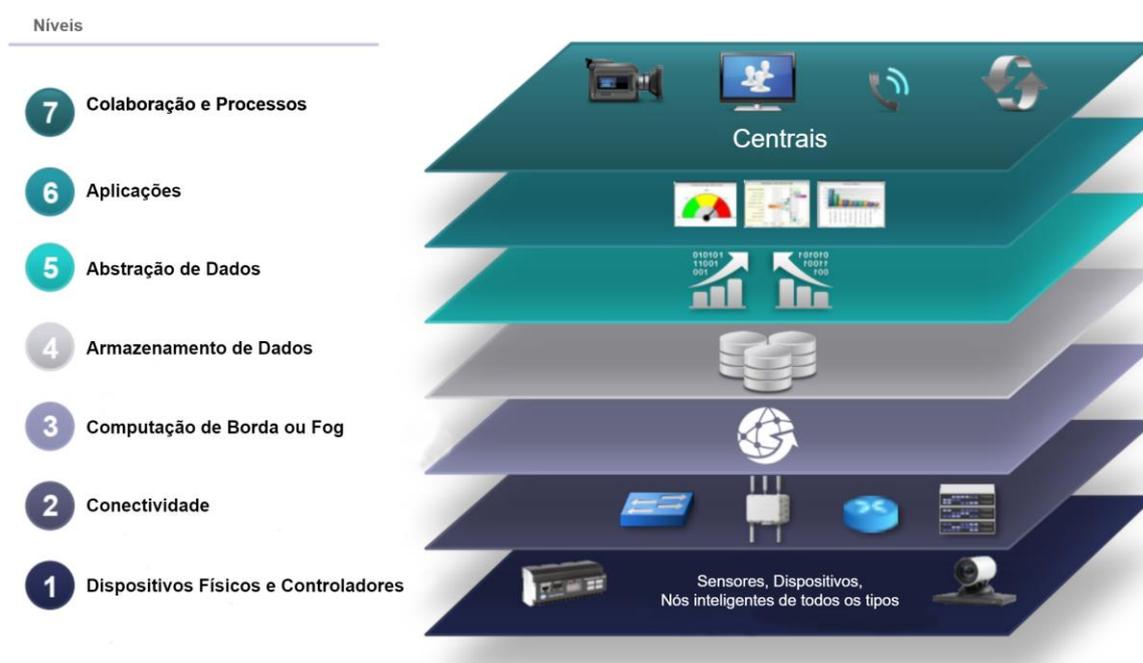
A comunicação ocorre de acordo com a arquitetura Cliente/Servidor, sendo que Cliente envia uma requisição ao Servidor para execução de uma ação ou para obter um recurso, e Servidor responde com a confirmação de execução da ação ou com o recurso solicitado. Entretanto, no CoAP normalmente ambos os dispositivos envolvidos na comunicação atuam tanto como Cliente quanto como Servidor (SHELBY; HARTKE; BORMANN, 2014).

### 3.2.3 Infraestrutura e Serviços

A IdC está desempenhando um papel fundamental em diversas áreas, tais como Saúde, Agricultura e Transporte, na medida em que possibilita a conexão de diversos tipos de dispositivos e a oferta eficiente de serviços de qualidade mundialmente via a Internet (MEHTA; SAHNI; KHANNA, 2018). Para suportar este

aumento de conectividade e a enorme quantidade de dados trafegados pela Internet, é necessário o uso de infraestruturas robustas que permitam transmitir e armazenar dados eficientemente.

A fim de prover uma melhor visualização dessas infraestruturas, o *Internet of Things World Forum* propôs um modelo de referência multinível para IdC, visando a simplificar e padronizar os conceitos e terminologias empregados, permitindo assim identificar com precisão os elementos e funções dos sistemas e aplicações IdC (CISCO et al, 2014). Esse modelo é baseado no fluxo de informação e tem sete níveis, conforme ilustrado na Figura 3.3.



**Figura 3.3 Modelo de Referência IdC (adaptado de (CISCO et al, 2014))**

Na Figura 3.3, as informações fluem do nível superior ao inferior, no modo de controle, e no sentido inverso, no modo de monitoramento. De acordo com (CISCO et al, 2014) e (CHAUDHURI, 2019), as principais características desses níveis são:

(a) Dispositivos Físicos e Controladores (Nível 1), os dispositivos físicos e controladores são as “coisas”, ou seja, os objetos capazes de enviar e receber informações. Os dispositivos podem ser consultados ou controlados via Internet e podem realizar conversões analógica/digital, a fim e transformar sinais capturados por um sensor em dados, ou ainda exercer alguma ação no ambiente no qual está inserido. Em razão dos recursos limitados, principalmente de armazenamento, os dados são encaminhados para algum equipamento no nível 2;

(b) Conectividade (Nível 2), mantém a comunicação e a conectividade entre os dispositivos do Nível 1 e a rede, entre as redes do Nível 2 e entre o Nível 2 e Nível 3. A transmissão de informação deve ser confiável, oportuna e usar preferencialmente as redes já existentes e compatíveis com IP, porém, equipamentos tais como um gateway, podem ser empregados para conectar dispositivos que ainda não são compatíveis com IP. A garantia da segurança da informação e a tradução da informação entre diferentes protocolos também devem ocorrer nesse nível;

(c) Computação de Borda ou Fog (Nível 3), as funções nesse nível são definidas pela necessidade de converter os dados provenientes da rede em informações adequadas a um posterior armazenamento e/ou processamento no nível superior. Nesse nível normalmente é realizado um processamento e uma análise de um alto volume de dados, o mais cedo e mais próximo possível da borda da rede, a fim de minimizar a latência de captura de dados, o impacto do tráfego de dados e a necessidade de processamento em níveis superiores;

(d) Armazenamento de Dados (Nível 4), os dados baseados em eventos e oriundos das camadas inferiores são armazenados e ficam disponíveis para posterior consulta por alguma aplicação não baseada em tempo real. Nesse nível normalmente ocorrem a filtragem por amostra e a comparação e agregação de dados com dados anteriormente armazenados, os quais podem ter como origem outras fontes não IdC;

(e) Abstração de Dados (Nível 5), os dados oriundos de diferentes dispositivos, que podem estar em diferentes regiões geográficas, são agregados e estruturados em modelos que simplificam o acesso por outras aplicações. Nesse nível normalmente ocorrem a combinação de dados de múltiplas fontes, a filtragem, a seleção, a normalização e a indexação, a fim de fornecer um rápido acesso às aplicações;

(f) Aplicações (Nível 6), mantém todos os tipos de aplicações, tais como *Enterprise Resource Planning (ERP)*, *Business Intelligence (BI)*, e os sistemas de gerenciamento e controle responsáveis pela interpretação da informação; e

(g) Colaboração e Processos (Nível 7), trata de pessoas e processos necessários à utilidade das aplicações IdC, e que agregam valor ao produzirem alguma ação a partir das informações obtidas das aplicações.

A partir da infraestrutura de IdC, uma série de serviços podem ser criados e disponibilizados para atender às demandas de diferentes áreas. De acordo com (XIAOJIANG; JIANLI; MINGDONG, 2020), esses serviços são classificados em função de suas características técnicas em quatro tipos:

(a) Serviços de Identificação, estão relacionados ao emprego de tecnologias, tal como a *Radio-Frequency IDentification (RFID)*, que é usada para obter informações relativas a um determinado objeto a partir de seu identificador único;

(b) Serviços de Agregação de Informação, estão relacionados aos processos de coleta, processamento e transmissão dos dados adquiridos de diferentes sensores para uma aplicação, onde são sumarizados a fim de facilitar a sua leitura;

(c) Serviços de Colaboração e Inteligência, utilizam os serviços de agregação da informação para a tomada de decisão, para em seguida realizar alguma ação a partir de um determinado cenário; e

(d) Serviços de Ubiquidade, visam oferecer Serviços de Colaboração e Inteligência quando e onde necessários.

### 3.3 Redes de Comunicação sem Fio

As redes de comunicação sem fio são essenciais na IdC, sobretudo para a comunicação entre dispositivos IdC e aplicações, já que cobrem áreas geográficas e regiões de difícil acesso às redes com fio, e já que possibilitam a redução dos custos com infraestrutura. De acordo com (POSLAND, 2009), o emprego de redes sem fio traz os seguintes benefícios quanto comparados ao das redes com fio:

(a) Ubiquidade, permite ao dispositivo IdC ter acesso à rede sem fio, desde que este tenha permissão, em qualquer lugar ao alcance do transmissor sem fio;

(b) Mobilidade, permite ao dispositivo IdC ter acesso à rede sem fio, ainda que este esteja em movimento;

(c) Adequabilidade, permite a instalação da rede sem fio em áreas onde as redes com fio seriam inconvenientes, com custo elevado, ou até mesmo inviável; e

(d) Adaptabilidade, permite aumentar ou reduzir a cobertura da rede sem fio, ou a quantidade de dispositivos IdC conectados à mesma, muito mais facilmente que as redes com fio.

Há atualmente uma variedade de tecnologias de comunicação sem fio, tais como Wi-Fi, Bluetooth Low Energy (BLE), Redes de Celular (3G/4G/5G) e ZigBee, que se diferenciam pela frequência, modulação e largura de banda empregadas e pelo consumo de energia (TANENBAUM; FEAMSTER; WETHERALL, 2021). Normalmente, sistemas IdC não se limitam ao uso de uma única tecnologia de comunicação sem fio, podendo utilizar mais de uma dessas tecnologias de forma alternada ou mesmo simultânea.

### 3.3.1 Wi-Fi

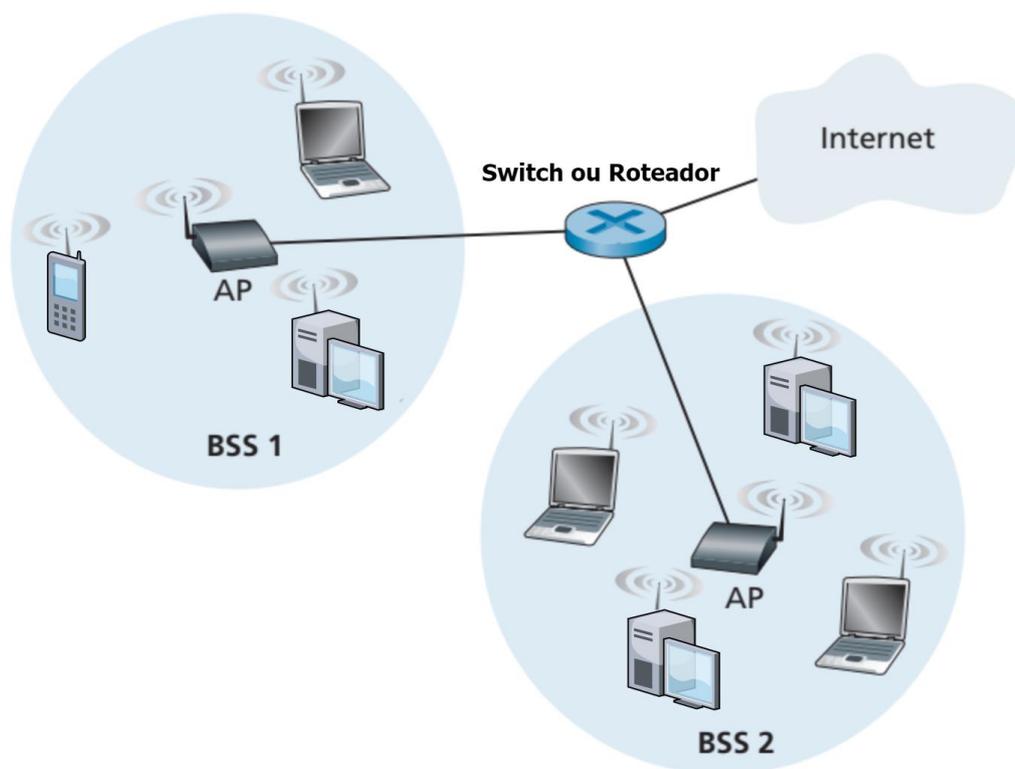
Wi-Fi é o termo utilizado para referenciar as redes de comunicação sem fio que operam de acordo com o padrão IEEE 802.11, o qual foi definido visando ao desenvolvimento de *Wireless Local Area Networks (WLAN)*, e que hoje estão presentes em diversos locais, tais como residências, hotéis, restaurantes, bares e aeroportos (OHRTMAN; ROEDER, 2003). A popularização do Wi-Fi, como meio de comunicação sem fio para conectar dispositivos móveis à Internet, gerou uma crescente demanda por conexões cada vez mais rápidas, o que impulsionou o surgimento de várias versões desse padrão ao longo dos anos, tais como 802.11b, 802.11g, 802.11n, 802.11ac e 802.11ax. Algumas das principais diferenças dessas versões na Camada Física são apresentadas em (KUROSE; ROSS, 2021) e resumidas na Tabela 3.1.

**Tabela 3.1 Características IEEE 802.11 (adaptado de (KUROSE; ROSS, 2021))**

Padrão	Faixa de Frequência	Taxa de Transferência	Alcance Médio
802.11b	2.4 GHz	Até 11 Mbps	30 metros
802.11g	2.4 GHz	Até 54 Mbps	30 metros
802.11n	2.4 GHz	Até 600 Mbps	70 metros
802.11ac	5 GHz	Até 3.47 Gbps	70 metros
802.11ax	2.4 GHz e 5 GHz	Até 14 Gbps	70 metros

Na Figura 3.4 é ilustrada uma arquitetura típica de rede Wi-Fi, constituída de dispositivos computacionais, tais como Desktops, Notebooks e, dotados de placas de rede sem fio padrão IEEE 802.11 e conectados a um *Access Point (AP)*, o qual

está conectado a um dispositivo de interconexão, tal como Switch ou Roteador, os quais por sua vez estão conectados à Internet (POSLAD, 2009).



**Figura 3.4 Arquitetura de rede Wi-Fi 802.11 (adaptado de (KUROSE; ROSS, 2021))**

O Wi-Fi foi desenvolvido como uma alternativa sem fio ao padrão Ethernet e sem grandes preocupações quanto ao consumo de energia. Conseqüentemente, para dispositivos IdC que dependem de fontes de energias, tais como baterias, não é o meio de comunicação sem fio mais adequado (SANTOS et al., 2016). Entretanto, o alto alcance de suas conexões e as altas taxas de transferências propiciadas pelo Wi-Fi, tornam essa tecnologia atraente aos dispositivos IdC que demandam por tais requisitos e são independentes ou pouco dependentes de fontes de energia.

### 3.3.2 Redes de Celular (3G/4G/5G)

As gerações de Redes de Celular 3G, 4G e 5G surgiram a partir da evolução das redes de Primeira Geração (1G), lançada em 1980 com tecnologia totalmente analógica, e de Segunda Geração (2G), lançada em 1991 já utilizando tecnologia de transmissão digital. Com a 3G ocorreu a popularização da Internet móvel, já que esta possibilitou taxas de transmissão de até 21Mbps. Com a 4G e 5G ocorreram

avanços significativos quanto à estabilidade, à capacidade de tráfego e velocidade, podendo em teoria está última, ultrapassar 1Gbps (KUROSE; ROSS, 2021).

Em princípio, tais redes possibilitam aos dispositivos IdC permanecerem conectados à Internet em qualquer lugar que seja coberto pelas mesmas. Entretanto, mesmo as redes 4G ainda não são adequadas para lidar com o grande número de dispositivos IdC conectados e com o conseqüente elevado consumo de energia. Esses problemas deverão ser sanados na medida em que as redes 5G forem sendo implantadas e ampliadas, já que estas deverão suportar essa grande quantidade dispositivos IdC conectados, com mais flexibilidade, baixa latência e baixo consumo de energia.

### 3.3.3 ZigBee

ZigBee é um conjunto de protocolos definido pela ZigBee Alliance com base no padrão IEEE 802.15.4. Opera nas bandas de rádio 868 MHz na Europa, 915 MHz nos Estados Unidos e 2.4 GHz para transmissão sem restrições em todo o mundo, e visa ao desenvolvimento de *Wireless Personal Area Networks (WPAN)* (POSLAD, 2009). De acordo (ELAHI; GSCHWENDER, 2010), as principais características das redes ZigBee são:

- (a) Baixo consumo de energia, permitindo aos dispositivos operem por meses ou até anos sem a necessidade da troca da bateria;
- (b) Baixo custo;
- (c) Baixa taxa de transmissão, alcançando velocidade máxima de 250 Kbps;
- (d) Facilidade de implementação;
- (e) Suportam até 65000 dispositivos conectados;
- (f) Rede em malha com autoconfiguração e auto recuperação; e
- (g) Emprega pacotes menores comparados aos das redes Wi-Fi e Bluetooth.

Há dois modos de operação para dispositivos em uma rede ZigBee: *Full-Function Device (FFD)* e *Reduced-Function Device (RFD)*. O primeiro é para dispositivos mais complexos, capazes de realizarem todas as tarefas definidas no padrão 802.15.4. O segundo é para dispositivos mais simples, capazes de realizarem uma quantidade limitada de tarefas desse padrão.

As formas de operação estão diretamente relacionadas ao papel que cada dispositivo desempenha na rede, que de acordo com (ELAHI; GSCHWENDER, 2010), pode ser de cinco tipos:

(a) *ZigBee Coordinator (ZC)*, é um dispositivo FDD responsável pelo gerenciamento de toda a rede, podendo ainda realizar a ponte com outras redes;

(b) *ZigBee Router (ZR)*, é um dispositivo FDD usado em topologias *Tree* e *Mesh* para expandir a cobertura da rede e que tem, como principal função, a escolha da melhor rota na rede para o destino dos dados;

(c) *ZigBee End Device (ZED)*, é um dispositivo RFD que dispõe apenas de funcionalidades para a comunicação com dispositivos ZC e ZR, consumindo assim pouca energia;

(d) *ZigBee Trust Center (ZTC)*, é o dispositivo responsável por gerenciar a segurança da rede, distribuindo chaves de segurança e realizando a autenticação de outros dispositivos; e

(e) *ZigBee Gateway (ZG)*, é um dispositivo usado para conectar uma rede ZigBee a outras redes, realizando a conversão entre protocolos.

Essas características, aliadas à flexibilidade de uso de diferentes topologias de rede, têm tornado as redes ZigBee uma alternativa bastante atraente para sistemas IdC, sobretudo no que se refere à aquisição automática de dados.

### 3.3.4 Bluetooth Low Energy (BLE)

Bluetooth é uma rede de comunicação sem fio do tipo WPAN baseada no padrão IEEE 802.15.1, que opera na frequência 2.4 GHz. Atualmente há duas versões do Bluetooth: *Bluetooth Classic*, que permite a transferência contínua de dados a taxas de até 3 Mbps, mas que possui um elevado consumo de bateria; e o *Bluetooth Low Energy (BLE)*, para uso em dispositivos com restrições de energia e que não exigem uma alta taxa de transferência de dados e uma alta frequência, tais como os sensores IdC (HEYDON, 2012).

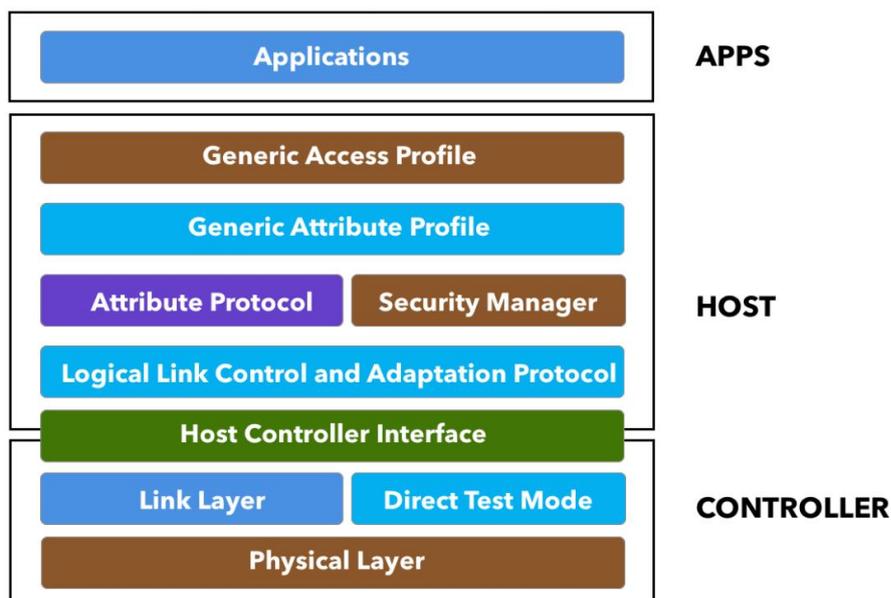
Já que há grandes diferenças entre essas versões, quanto à especificação técnica, implementação e tipo de aplicação adequada, estas são incompatíveis e não podem comunicar-se entre si (AFANEH, 2018). Isso levou a maioria dos fabricantes de dispositivos que usam Bluetooth, tais como smartphones, a implementarem ambas as versões em seus dispositivos, a fim de que estes possam

se comunicar com outros dispositivos que eventualmente usem somente uma das versões.

BLE foi introduzido na *Bluetooth Specification Version 4*, voltado para aplicações principalmente na Saúde, Entretenimento e Segurança, e fornecendo aos desenvolvedores uma enorme flexibilidade para a concepção de dispositivos semelhantes aos da IdC. O suporte às topologias Ponto a Ponto, *Broadcast* e *Mesh*, permite a criação de redes confiáveis e em larga escala. De acordo com (AFANEH, 2018), algumas das características mais importantes das redes BLE são:

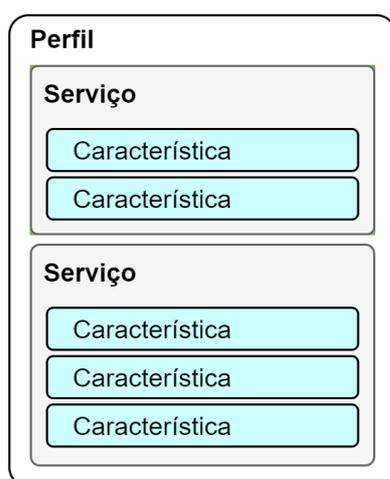
- (a) Taxa máxima de transferência de 2Mbps;
- (b) Alcance médio da rede varia em torno de 10 a 30 metros, sendo influenciado pelo ambiente e configurações do dispositivo;
- (c) Consumo de energia varia de acordo com a implementação, parâmetros de conexão e chipset do dispositivo;
- (d) Segurança é um item opcional, podendo usar vários modos pré-existentes ou ainda ser implementada pelo desenvolvedor;
- (e) Projetado para aplicações com baixas taxas de transferência, sendo que altas taxas comprometem significativamente um dos objetivos principais, que é o baixo consumo de energia; e
- (f) Novas versões do BLE são compatíveis com as anteriores, porém algumas funcionalidades podem ficar limitadas no dispositivo com o uso de versão antiga.

A Figura 3.5 ilustra a arquitetura do BLE, constituída de múltiplas camadas organizadas nos blocos *APPS*, *Host* e *Controller*. No primeiro bloco encontra-se a camada *Applications*, onde deve ser criada uma interface para a pilha de protocolos BLE, sendo que o próprio desenvolvedor do dispositivo é responsável por essa criação, quando da definição via código das funcionalidades desse dispositivo. No segundo bloco encontram-se as camadas superiores do BLE, onde os protocolos de rede e transporte garantem a comunicação entre dispositivos no mesmo nível e de forma interoperável. No terceiro bloco encontram-se as camadas inferiores do BLE, onde a maioria é implementada em hardware e são responsáveis pela comunicação sem fio (AFANEH, 2018; ZEPHYR, 2022).



**Figura 3.5 Arquitetura do BLE (AFANEH, 2018).**

As camadas *Generic Access Profile (GAP)* e *Generic Attribute Profile (GATT)* asseguram a interoperabilidade entre os dispositivos de diferentes fabricantes. GAP fornece um framework que define como os dispositivos devem interagir entre si para realizar tarefas, tais como transmissão de dados, descoberta de novos dispositivos, estabelecimento de conexões, gerenciamento de conexões e negociação de níveis de segurança. GATT estabelece modelos e hierarquias de dados a serem trocados pelos dispositivos usando os conceitos Perfil, Serviço e Característica, conforme ilustrado na Figura 3.6.



**Figura 3.6 Modelo hierárquico de Perfil, Serviço e Característica**

Perfil é a especificação genérica para recepção e envio de uma pequena quantidade de dados, que define o comportamento dos dispositivos quanto aos

serviços, características e conexões. Serviços são usados para segmentar os dados em entidades lógicas, que contém partes mais específicas, denominadas características. Característica é sempre a parte de um serviço que representa um pedaço de informação, como por exemplo o batimento cardíaco lido pelo dispositivo IdC. Cada característica pode ainda possuir dois atributos: Propriedade, que informa o tipo de acesso a ser realizado, tal como leitura, escrita, ou escrita sem resposta; e Descritor, que contém informações adicionais sobre a característica.

*Bluetooth Special Interest Group (SIG)* possui uma lista de especificações de Perfis, Serviços e Características pré-definidas para dispositivos BLE com aplicações na Saúde e Esportes, que podem ser utilizadas livremente e asseguram a interoperabilidade entre esses dispositivos. Cada Serviço deve possuir um identificador único, que o distingue dos demais, chamado *Universally Unique Identifier (UUID)*, cujo tamanho deve ser 16 bits para os pré-definidos pela Bluetooth SIG, ou de 128 bits para os definidos pelos desenvolvedores.

BLE tem sido amplamente empregado no desenvolvimento de sistemas IdC, sobretudo em razão dos baixos custos de implantação e baixo consumo de energia, principalmente quando comparado ao Wi-Fi e 3G. O emprego de BLE em praticamente todos os *smartphones* disponíveis no mercado, outorga uma imensa vantagem a essa tecnologia em relação a outras, tais como ZigBee, já que possibilita a criação de redes de sensores utilizando o próprio aparelho do usuário.

### 3.4 Redes de Sensores Sem Fio

Rede de Sensores Sem Fio (RSSF) é uma rede com mecanismos de auto-organização, constituída por um conjunto de “sensores inteligentes” ou nodos distribuídos espacialmente, que operam cooperativamente a fim de coletar e processar dados de objetos monitorados na região de abrangência da rede (CHAI et al., 2020). Nodos são dispositivos eletrônicos contendo memória, microcontrolador, transceptor sem fio, bateria e um ou mais sensores (e.g., temperatura, aceleração, batimentos cardíacos, pressão), dispostos estrategicamente para medir parâmetros de um ambiente, objeto, ou do próprio corpo humano.

RSSF é uma rede ad-hoc, na qual os nodos comunicam-se diretamente, ou usam um nodo intermediário que atua como um roteador e encaminha os pacotes de dados para o nodo de destino, permitindo assim a criação de uma infraestrutura descentralizada (IQBAL et al., 2020). Esse tipo de rede deve ser construído de forma autônoma em ambientes altamente dinâmicos, escaláveis e distribuídos, onde o emprego de tecnologias de comunicação com fio dificulta extremamente a sua implementação. Assim sendo, um requisito fundamental das RSSFs é o emprego de tecnologias de comunicação sem fio que minimizem o consumo de energia, tais como ZigBee e BLE.

RSSFs estão cada vez mais tornando-se essenciais na IdC, já que possuem vários domínios de aplicação em diversos setores da atividade humana, tais como industrial, residencial, ambiental, alimentício e saúde. Na Medicina, RSSFs têm sido amplamente usadas no monitoramento da saúde de pacientes, gerando inúmeros benefícios, dentre os quais a redução de custos e uma melhor qualidade de vida (ALHARBE; ATKINS; CHAMPION, 2015).

### **3.5 Redes de Sensores do Corpo Humano**

Rede de Sensores do Corpo Humano (RSCH) é um tipo específico de RSSF, usada para monitorar sinais vitais de humanos. Constituída de nodos colocados ou implantados no corpo do humano, que não atrapalham suas atividades diárias, é muito útil para monitorar e até detectar doenças crônicas, tais como, Hipertensão, Diabetes e Asma (SANGWAN; BHATTACHARYA, 2015). Todos os nodos que compõe uma RSCH devem ser altamente confiáveis, possuir um baixíssimo consumo de energia, serem resistentes a interferências e serem capazes de operar em um raio de até 5 metros (TAVERA et al., 2021).

RSCH é uma rede autônoma, capaz de corrigir falhas, tomar decisões e realizar ações solicitadas via uma aplicação, sem a necessidade de intervenção do usuário. Apesar dessa autonomia, esse tipo de rede deve oferecer mecanismos de controle aos seus usuários que, por exemplo, no domínio da Saúde pode ser um paciente ou um médico. De acordo com (SANGWAN; BHATTACHARYA, 2015), em

relação à decisão a ser tomada quanto aos dados coletados, há três tipos de classificação para uma RSCH:

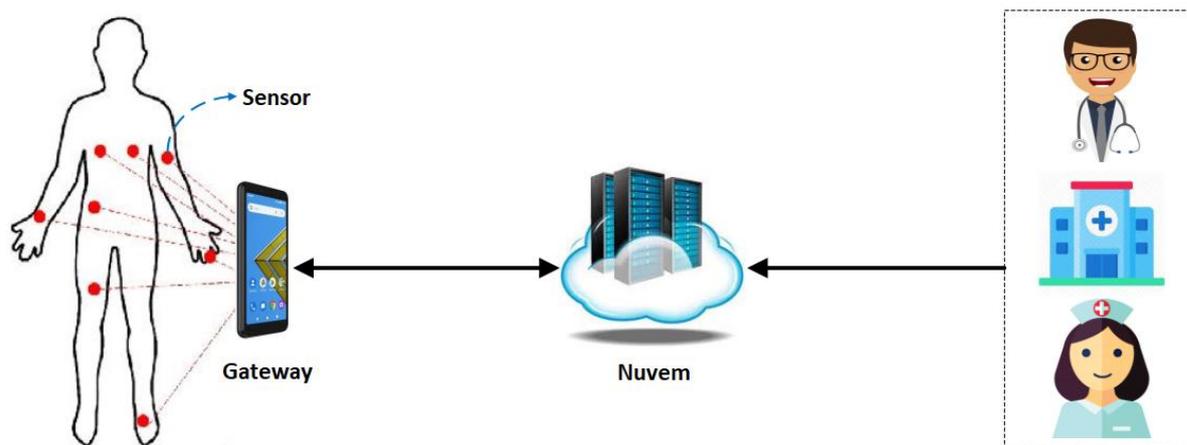
(a) RSCH Gerenciada, é uma rede onde os dados coletados por um ou mais sensores são enviados para um terceiro (e.g., um médico, um enfermeiro ou um centro médico), o qual realizará a análise desses dados, o diagnóstico e a tomada de decisão, por exemplo prescrevendo uma receita ao paciente;

(b) RSCH Autônoma, é uma rede capaz de autonomamente realizar a análise dos dados e, via um atuador, realizar alguma ação no corpo humano sem esperar por uma decisão de terceiros. Esse tipo de rede é indicado para aplicações que lidam com emergências, onde as decisões devem ser tomadas em tempo real; e

(c) RSCH Inteligente, é uma rede que combina as operações dos tipos anteriores, tomando decisões de forma autônoma em situações simples e encaminhando os dados a terceiros em situações complexas. Neste último caso, a rede aguarda um tempo pré-definido por um retorno e, caso este não ocorra, realiza a tomada de decisão.

Cada nodo da RSCH pode conter um único sensor, capaz de captar um único tipo de sinal fisiológico, ou conter múltiplos sensores, capazes de captar diferentes tipos desses sinais. Temperatura Cutânea (TC), Pressão Arterial (PA), Batimentos Cardíacos (BC), EletroCardioGramma (ECG) e Oxímetro de Pulso (OP), são alguns exemplos de sensores para captar sinais fisiológicos. Os tipos e número de sensores em cada nodo é determinado em função dos requisitos da aplicação.

Para um monitoramento contínuo e não invasivo, a usabilidade é de extrema relevância. Nesse caso, sensores podem ser acoplados em peças de roupas e acessórios (e.g., relógios, pulseiras, anéis, brincos e óculos), denominados *Wearable Devices*, tornando-os discretos, confortáveis e quase imperceptíveis aos seus usuários. A Figura 3.7 ilustra um cenário típico de uso de RSCH no domínio da Saúde.



**Figura 3.7 Cenário de uso de RSCH (adaptado de (ALI; SADEGHI; LIU, 2020))**

Na Figura 3.7 há diversos sensores captando e enviando dados fisiológicos a um gateway, que nesse caso é um smartphone portado pelo paciente. Esse dispositivo intermediário possibilita o gerenciamento da rede, pode realizar um pré-processamento nos dados fisiológicos recebidos, sendo que em seguida encaminha tais dados à nuvem, onde são armazenados e tratados para visualização por parte de profissionais de saúde ou provedores de serviço autorizados.

### 3.6 Internet das Coisas e o Cuidado de Saúde Pervasivo

A Computação Ubíqua, e mais recentemente a IdC, vem atuando há pelo menos duas décadas em diversos domínios, transformando significativamente alguns destes, como no caso da Saúde onde surgiu o conceito de Cuidado de Saúde Pervasivo. Com base nos dados fornecidos por diversos sensores IdC, profissionais de saúde podem monitorar seus pacientes a distância, a qualquer hora e em qualquer lugar, podendo realizar diagnósticos e prescrever tratamentos mais rapidamente e eficientemente, também a distância (CHAI et al., 2020).

A IdC reduz a necessidade de pacientes, sobretudo com doenças crônicas e idosos, deslocarem-se de seus lares e/ou locais de trabalho, na medida em que possibilita o acompanhamento remoto e em tempo real desses pacientes por profissionais de saúde. Possibilita também a pacientes, que frequentemente esquecem de tomar suas medicações, receberem alertas relativos às mesmas em momentos apropriados, podendo inclusive ocorrer o registro das datas e horários em

que essas medicações foram tomadas. Esse suporte tecnológico propiciado pela IdC, fornece bases mais sólidas aos profissionais de saúde para a tomada decisão em relação a seus diagnósticos e em relação aos melhores planos de tratamento terapêutico para seus pacientes (BALAS, 2020).

Sistemas IdC têm sido empregados em vários setores da Saúde, sobretudo visando a: reduzir o tempo de espera em salas de emergência; acompanhar idosos; monitorar pacientes com doenças crônicas; rastrear informações para assegurar que não ocorram erros na entrega e administração de medicação em pacientes; gerenciar a alimentação de pacientes; acompanhar pacientes com desordens neurológicas; e acompanhar pacientes exercendo atividades físicas (BHUIYAN et al., 2021; QADRI et al., 2020).

### **3.7 Considerações Finais**

Este capítulo sintetizou os resultados de um levantamento bibliográfico relativo às Computações Ubíqua e Pervasiva e à IdC, apresentando alguns de seus principais conceitos e tecnologias, assim como os benefícios que estes têm proporcionado sobretudo para a Saúde. Foram apresentados também os principais paradigmas, protocolos, infraestruturas e serviços envolvendo essas áreas da Ciência da Computação, particularmente as redes de comunicação sem fio, que são essenciais para o desenvolvimento de sistemas IdC.

# Capítulo 4

## TRABALHOS CORRELATOS

---

*Este capítulo apresenta os principais trabalhos correlacionados ao apresentado nesta dissertação, obtidos a partir de uma revisão bibliográfica da literatura relativa a trabalhos que visam ao emprego das tecnologias da Computação Ubíqua, Computação Pervasiva e IdC para o monitoramento de pacientes com Hipertensão.*

### 4.1 Considerações Iniciais

O impacto causado pela Computação Ubíqua, Computação Pervasiva e IdC na Saúde vem crescendo progressivamente nas últimas duas décadas, na medida em que os sistemas de saúde vêm utilizando cada vez mais tecnologias oriundas dessas áreas da Computação, sobretudo porque estas vêm assegurando uma melhor qualidade e maior eficiência no Cuidado da Saúde. Em particular, essas tecnologias têm sido empregadas em sistemas de monitoramento de pacientes e, uma vez que o SBIdC-MPH se enquadra nessa categoria, foi realizado um levantamento bibliográfico de tais sistemas, focando nos que realizam o monitoramento de pacientes com DCNTs e mais especificamente naqueles que monitoram pacientes com Hipertensão.

No decorrer deste capítulo os principais trabalhos correlatos ao SBIdC-MPH são apresentados e discutidos da seguinte forma: a seção 4.2 descreve a metodologia de pesquisa empregada na revisão bibliográfica e os principais trabalhos correlatos encontrados na literatura recente; a seção 4.3 realiza uma análise comparativa entre esses trabalhos e o SBIdC-MPH; e a seção 4.4 tece algumas considerações finais.

## 4.2 Sistemas de Monitoramento para o Cuidado de Saúde

A literatura apresenta alguns trabalhos que empregam tecnologias da Computação Ubíqua, Computação Pervasiva, Computação em Nuvem e Internet das Coisas (IdC) para o monitoramento de pacientes com Hipertensão. Uma busca foi realizada nas bases de dados IEEE Xplore e Science Direct por trabalhos publicados a partir de 2015, utilizando os seguintes termos de pesquisa: “Hypertension”, “Internet of Things”, “Pervasive”, “Ubiquitous”, “System”, “Platform”, “Environment” e “Monitoring”. A seguir os trabalhos correlatos mais relevantes são relacionados e comparados ao SBIdC-MPH.

Em (BRESNICK, 2015), foi apresentado um framework baseado em IdC e Fog Computing capaz de monitorar continuamente a pressão arterial e outros parâmetros de saúde tendo como objetivo identificar o estágio de Hipertensão e prever o risco de ataques hipertensivos utilizando Redes Neurais Artificiais (RNA). Esse framework é capaz de gerar continuamente alertas sobre flutuações de pressão arterial, os quais são enviados para uma aplicação móvel no smartphone dos pacientes. Os resultados das análises dos dados são encaminhados para um subsistema localizado na nuvem, onde são armazenados e ficam disponíveis para serem consultados futuramente por profissionais de saúde previamente autorizados pelos pacientes.

Em (KIRTANA; LOKESWARI, 2017), foi proposto um sistema para monitoramento remoto de pacientes com pré-Hipertensão, o qual utiliza tecnologias da IdC para coletar dados de Variabilidade de Frequência Cardíaca. Os dados coletados são transmitidos via o protocolo MQTT para um servidor, onde ficam armazenados e podem ser acessados pelos médicos através de uma aplicação Web. A aplicação no servidor analisa os dados recebidos e quando estes atingem uma faixa de risco pré-definida, alertas são enviados via *Short Message Service* (SMS) para os cuidadores e médicos responsáveis.

Em (KIM et al, 2018), foi proposto um sistema baseado em Multi-Agentes para o gerenciamento e diagnóstico de pacientes com hipertensão, que monitora continuamente e fornece recomendações diárias para o controle da pressão arterial e gestão de dietas e exercícios. Esse sistema utiliza dados de saúde do paciente, dados do ambiente e uma base de conhecimento com recomendações médicas para

identificar situações de risco à saúde do paciente e imediatamente iniciar uma ação de acordo com regras pré-definidas.

Em (PUTRA; WIBISONO, 2017), foi apresentado um sistema M2M para o monitoramento de pacientes com Hipertensão, sendo capaz de fornecer recomendações médicas com o objetivo de manter a pressão arterial dentro dos limites normais. Os dados de pressão arterial são lidos por um esfigmomanômetro ligado a um microcontrolador e em seguida enviados para um servidor, onde são analisados para fornecer recomendações médicas sobre medicações e número de doses. As recomendações são encaminhadas para os médicos responsáveis para aprovação e, uma vez aprovadas, são enviadas e recebidas pelo paciente via uma aplicação móvel. Uma caixa de remédios conectada a um microcontrolador é utilizada para monitorar se os pacientes estão ingerindo corretamente os medicamentos.

Em (GHOSHACHANDRA et al., 2017), foi proposto uma aplicação para monitorar pacientes idosos com Hipertensão, através da qual é possível obter dados coletados por um dispositivo de leitura de pressão arterial, conectado ao smartphone do paciente, e registrar manualmente dados sobre dieta, remédios e exercícios. Todos os registros de dados são enviados para a nuvem, onde são armazenados e em seguida analisados via o método de Regressão Logística Multiclasse. Os dados analisados podem ser compartilhados com médicos, assistentes e familiares do paciente.

Em (PATIL; CHAWARE, 2021), foi apresentado um sistema baseado em IdC para o monitoramento dos batimentos cardíacos, como sendo uma alternativa de baixo custo para a coleta de dados em tempo real. Esse sistema emprega um sensor baseado em fotopletismografia e um sensor piezoelétrico para obter dados de batimento cardíaco e frequência ressonante, os quais são enviados via rede sem fio para a nuvem, onde são armazenados para futura análise.

### **4.3 Análise Comparativa**

Todos esses trabalhos têm por objetivo principal o monitoramento de pacientes com Hipertensão. Para tal, empregam diferentes tipos de sensores e

obrigam os pacientes a realizem as aferições manualmente, ou então não consideram o estado vigente do paciente para realizar as leituras dos dados clínicos. A fim de obter leituras mais precisas sobre a pressão arterial do paciente, a Plataforma de Sensores do SBIdC-MPH analisa continuamente o movimento do paciente e realiza a leitura dos dados clínicos somente quando é detectado que o mesmo está em repouso. Essa plataforma foi projetada e implementada de modo a não interferir nas atividades diárias do paciente e também contém um sistema de segurança que garante a confidencialidade dos dados clínicos coletados do mesmo, na medida que estes só podem ser lidos via o dispositivo móvel desse paciente. Além disso, o hardware dessa plataforma foi projetado buscando outorgar flexibilidade à mesma, a fim de que outros tipos de sensores pudessem ser incorporados a essa plataforma, permitindo assim a captação de outros tipos de dados clínicos.

Com relação a arquitetura do SBIdC-MPH, esta foi projetada para permitir o uso de outros sensores que possam fornecer dados adicionais para o monitoramento de pacientes com Hipertensão Arterial, ou mesmo permitir o monitoramento de pacientes com outros tipos de DCNT, tais como o Diabetes, Asma e Obesidade. Essa arquitetura permite realizar um pré-processamento dos dados clínicos no dispositivo móvel do paciente, sendo que esse dispositivo se encontra na camada denominada Fog dessa arquitetura, possibilitando assim identificar níveis de risco e imediatamente alertar o próprio paciente e/ou os contatos de emergência designados pelo mesmo. Além disso, essa arquitetura permite a geração de alertas mesmo que o dispositivo móvel não esteja temporariamente conectado à Internet, sendo que o pré-processamento realizado nesse dispositivo permite também reduzir a carga para processamento e análise a serem realizados na nuvem computacional. Para a comunicação entre as diversas aplicações do SBIdC-MPH foi empregado o protocolo GraphQL, a fim de possibilitar uma maior escalabilidade dessas aplicações e do próprio sistema, sendo este também um importante diferencial em relação aos trabalhos correlatos mencionados.

Com relação ao trabalho (RODRIGUES; MENDES; SOUZA, 2020), intitulado “MyHealth: A System for Monitoring Non-Communicable Diseases”, este relata um projeto inicial, desenvolvido durante uma disciplina de pós-graduação, no qual basicamente é proposto um framework, constituído de aplicações genéricas para dispositivo móvel e para nuvem, visando ao desenvolvimento de sistemas para o

monitoramento de DCNTs. Conforme já afirmado, o MyHealth foi um trabalho inicial, cujas principais limitações e problemas técnicos são os seguintes:

(a) Não foi realizado nenhum desenvolvimento na Camada de Sensores, sendo que esse sistema parte do princípio que, nessa camada, há um conjunto de dispositivos tradicionais para a medida de dados clínicos que devem ser manipulados pelo próprio paciente para a coleta e registro de tais dados em seu dispositivo móvel. Portanto, esse procedimento é totalmente manual e requer a interferência do paciente;

(b) A aplicação na Camada Fog, que deve ser executada no dispositivo móvel do paciente é genérica, sendo que foram efetivamente desenvolvidos somente os componentes comuns a todas as DCNTs, restando adaptar e desenvolver componentes específicos à DCNT que está sendo alvo de monitoramento. Portanto, essa aplicação não é executável na forma em que se encontra; e

(c) As aplicações na Camada Nuvem, que devem ser executadas na nuvem computacional, também são genéricas, sendo que também foram efetivamente desenvolvidos somente os componentes comuns a todas as DCNTs, restando adaptar e desenvolver componentes específicos à DCNT que está sendo alvo de monitoramento. Portanto, essas aplicações também não são executáveis na forma em que se encontram.

Já o SBIdC-MPH, foi fruto deste projeto de mestrado totalmente desenvolvido, ou seja, concepção, especificação, implementação e prototipagem. O framework do MyHealth serviu apenas como ponto de partida para o desenvolvimento do SBIdC-MPH, cujos principais avanços técnicos alcançados foram os seguintes:

(a) Desenvolvimento completo de uma Plataforma de Sensores para a Camada de Sensores, que possibilitou a coleta e a transmissão automáticas, assim como o registro automático no dispositivo móvel do paciente, dos dados clínicos relevantes à DCNT Hipertensão. Portanto, esse procedimento foi totalmente automatizado, não requerendo qualquer interferência do paciente;

(b) Desenvolvimento completo da aplicação Mobile App para a Camada Fog, através do reuso e adaptação de componentes do framework do MyHealth e do desenvolvimento de componentes específicos para a DCNT Hipertensão. Portanto, essa aplicação é executável e passível de ser utilizada no contexto do SBIdC-MPH;

e

(c) Desenvolvimento completo das aplicações API Server e Web App para a Camada Nuvem, através do reuso e adaptação de componentes do framework do MyHealth e do desenvolvimento de componentes específicos para a DCNT Hipertensão. Portanto, essas aplicações são executáveis e passíveis de serem utilizadas no contexto do SBIdC-MPH.

Cabe salientar ainda que o protótipo como um todo do SBIdC-MPH é passível de ser transformado em um produto, o que não é absolutamente o caso do MyHealth. A Tabela 4.1 apresenta uma análise comparativa entre o SBIdC-MPH e os principais trabalhos correlatos encontrados na literatura, incluindo o MyHealth, de acordo com 09 parâmetros qualitativos considerados os mais relevantes.

**Tabela 4.1 Comparações entre o SBIdC-MPH e trabalhos correlatos**

	(SOOD; MAHAJAN, 2018)	(KIRTANA; LOKESWARI, 2017)	(KIM et al, 2018)	(PUTRA; WIBISONO, 2017)	(GHOSHACHA NDRA et al., 2017)	(PATIL; CHAWARE, 2021)	MyHealth	SBIdC-MPH
Possui uma plataforma de sensores	Não	Sim	Sim	Sim	Não	Sim	Não	Sim
Dispositivos vestíveis com baixo impacto nas atividades diárias	Não	Não	Não	Não	Não	Não	Não	Sim
Leva em consideração o movimento do usuário para a coleta de dados	Não	Não	Não	Não	Não	Não	Não	Sim
O sistema permite a fácil adaptação para a integração com outros sensores	Não	Não	Não	Não	Não	Não	Não	Sim
Onde é realizado o processamento e análise dos dados	Dispositivos Fog	Cloud	Smartphone	Cloud	Cloud	Cloud	Cloud	Dispositivos Fog e Cloud
Permite o cadastro e gerenciamento de médicos e pacientes	Não	Não	Não	Não	Não	Não	Não	Sim
Apresenta uma interface para análise dos dados pelo médico	Não	Sim	Não	Sim	Sim	Não	Sim	Sim
O paciente pode receber recomendações médicas a respeito do tratamento	Não	Não	Não	Sim	Não	Não	Não	Sim
Gera alertas sobre o nível de risco à saúde do paciente	Sim	Sim	Sim	Sim	Não	Sim	Não	Sim

## 4.4 Considerações Finais

Este capítulo apresentou os principais trabalhos correlatos ao SBIdC-MPH encontrados na revisão bibliográfica realizada, os quais empregam ou propõem o uso de tecnologias da Computação Ubíqua, Computação Pervasiva e IdC para o monitoramento de pacientes com a DCNT Hipertensão.

Embora o SBIdC-MPH incorpore algumas das funcionalidades dos sistemas descritos nesses trabalhos correlatos, este oferece várias funcionalidades adicionais, sobretudo devido à Plataforma de Sensores incorporada a um dispositivo vestível, a qual foi projetada para ter um baixo impacto nas atividades diárias do paciente. Outras funcionalidades adicionais são, o pré-processamento na Camada Fog e a possibilidade de gerenciamento de profissionais de saúde e pacientes.

# Capítulo 5

## SISTEMA BASEADO EM INTERNET DAS COISAS PARA O MONITORAMENTO DE PACIENTES COM HIPERTENSÃO

---

*Este capítulo descreve o SBIdC-MPH. Ao longo das próximas seções serão apresentados detalhes da arquitetura, aplicações e funcionalidades do mesmo para o monitoramento de pacientes com hipertensão.*

### 5.1 Considerações Iniciais

O SBIdC-MPH foi desenvolvido visando a melhoria dos serviços de saúde e da qualidade de vida de pacientes, mais especificamente, visando a suportar o monitoramento remoto de pacientes com Hipertensão Arterial. Para tal, tecnologias da IdC, das Computações Ubíqua, Pervasiva e Nuvem foram empregadas no desenvolvimento de uma estrutura para a coleta, processamento e análise de dados fisiológicos de pacientes, e para disponibilizar esses dados a profissionais de saúde, os quais podem acompanhar à distância e fornecer feedbacks imediatos a esses pacientes. Apesar de direcionado ao monitoramento de pacientes com Hipertensão Arterial, o SBIdC-MPH poder ser adaptado e/ou estendido para permitir o monitoramento de pacientes com outras DCNTs que envolvam a coleta de dados fisiológicos do corpo humano.

Este capítulo central desta dissertação de mestrado, apresenta o SBIdC-MPH da seguinte forma: a seção 5.2 discorre sobre a metodologia empregada para o desenvolvimento das aplicações; a seção 5.3 fornece uma visão geral do sistema e de seus componentes; a seção 5.4 apresenta a Camada de Sensores, que é responsável pela coleta de dados fisiológicos; a seção 5.5 apresenta a Camada Fog, que é responsável pelo pré-processamento e envio de dados à nuvem computacional; a seção 5.6 apresenta a Camada da Nuvem, que é responsável pelo processamento de dados e pela visualização dos mesmos pelos profissionais de saúde; a seção 5.7 tece algumas considerações finais.

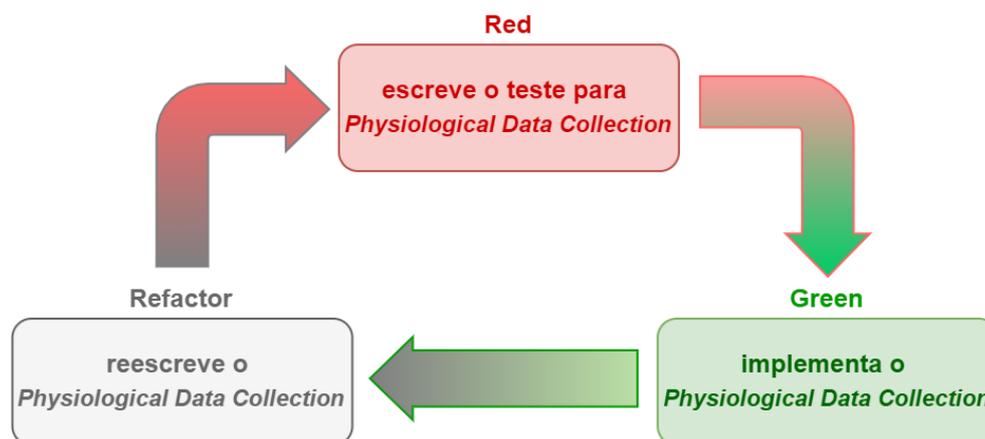
## 5.2 Metodologia de Desenvolvimento

As aplicações API Server, Web App, e Mobile App foram desenvolvidas de acordo com *Test-Driven Development (TDD)*, visando uma maior produtividade no desenvolvimento, qualidade do software e extensibilidade dessas aplicações. O TDD é uma metodologia ágil para de desenvolvimento de software, baseado em ciclos curtos de desenvolvimento, sendo que testes automatizados são descritos antes da efetiva implementação das funcionalidades em código. Esses testes especificam e validam exatamente o que o código deve fazer (BECK, 2002; XU; LI, 2009).

Tal metodologia propõe que o processo de desenvolvimento seja em pequenos ciclos, divididos em três etapas: *Red*, *Green* e *Refactor*. Na primeira etapa, é realizada a escrita de um teste automatizado para uma pequena funcionalidade, sendo que o teste deve obrigatoriamente falhar. Na segunda etapa, é desenvolvido apenas uma pequena parte do código, correspondente à funcionalidade, e que irá permitir passar no teste. Na última etapa, a funcionalidade é reescrita, visando a remoção de possíveis códigos duplicados e para otimização. Após essa etapa, um novo ciclo é iniciado para o desenvolvimento de outra funcionalidade (GLEASON; SPROVIERO; GONDA, 2019).

Segundo (BECK, 2002), o desenvolvimento utilizando esses ciclos de desenvolvimento possui algumas vantagens: é possível obter um feedback mais rápido a respeito da funcionalidade desenvolvida e outras já existentes na aplicação; o código final é mais limpo e simples, pois contém somente o que é necessário para

a requerida funcionalidade; promove a segurança, pois cada etapa é testada e qualquer possível problema é facilmente detectado; reduz o tempo de desenvolvimento; e a divisão do código em partes mais específicas propicia um menor acoplamento. A Figura 5.1 ilustra o ciclo TDD para a funcionalidade *Physiological Data Collection*, da aplicação Mobile App, que permite o envio de dados à nuvem.

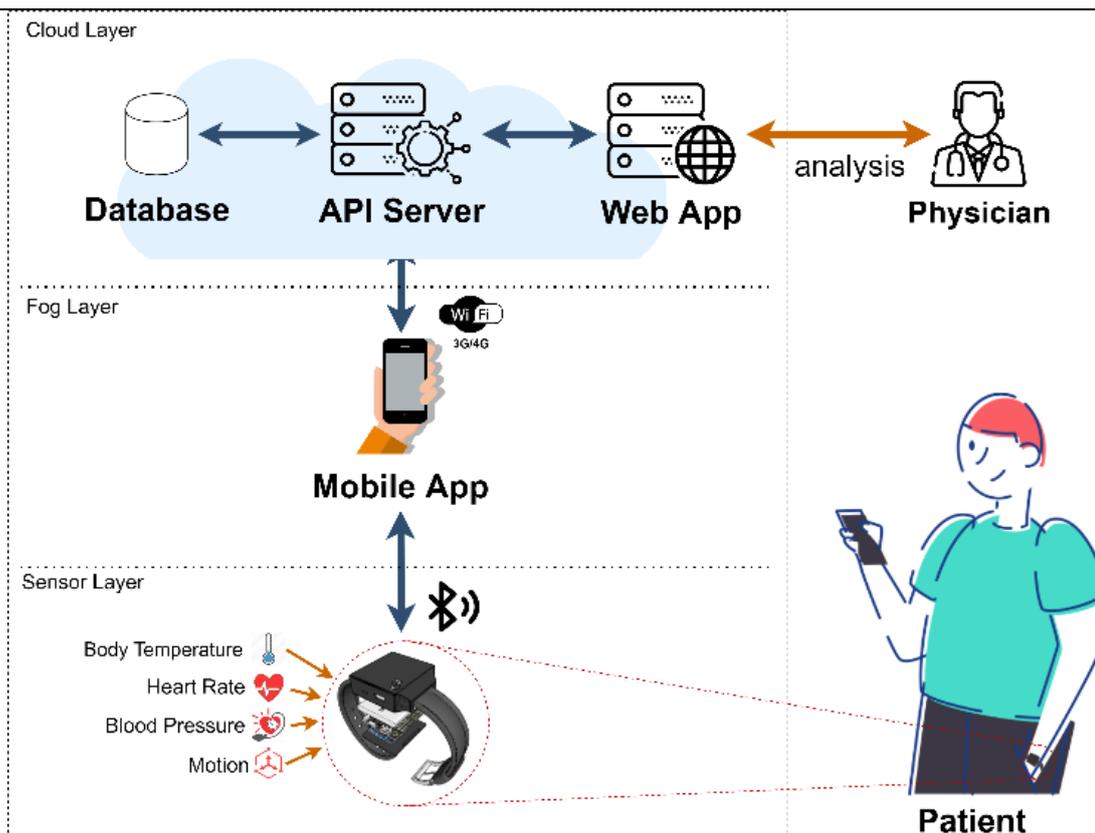


**Figura 5.1** Ciclo TDD para a funcionalidade *Physiological Data Collection*

Conforme ilustrado na Figura 5.1, primeiro um teste é escrito, o qual tem por objetivo validar o correto envio dos dados fisiológicos do paciente à nuvem. Esse teste deve receber como retorno uma confirmação de envio dos dados. Entretanto, como a funcionalidade *Physiological Data Collection* ainda não foi implementada, o teste falha. Como consequência, é implementado um código mínimo para a funcionalidade *Physiological Data Collection*, a qual realiza o envio dos dados via Internet e, dessa forma, possibilita ao teste passar. Finalmente, o código é reescrito visando a eliminar duplicidades, aumentar a confiabilidade e manutenibilidade, sempre assegurando que o mesmo continue passando no teste.

### 5.3 Visão Geral do Sistema

A Figura 5.2 apresenta uma visão geral do SBIdC-MPH, sendo que seus componentes foram estruturados, de acordo com as suas localizações e funcionalidades, em três camadas: Sensor Layer, Fog Layer e Cloud Layer.



**Figura 5.2 Visão geral do SBIc-MPH**

A Sensor Layer engloba dispositivos vestíveis atrelados ao corpo do paciente, os quais possuem diferentes sensores responsáveis por capturar sinais vitais do paciente e os disponibilizar no formato digital. No protótipo desenvolvido, essa camada contém uma Plataforma de Sensores, que coleta esses sinais vitais e transmite automaticamente dados clínicos relativos a esses sinais à Fog Layer via o padrão BLE.

A Fog Layer contém a aplicação Mobile App, que processa os dados recebidos e os encaminha à Cloud Layer via redes WiFi ou 3G/4G.

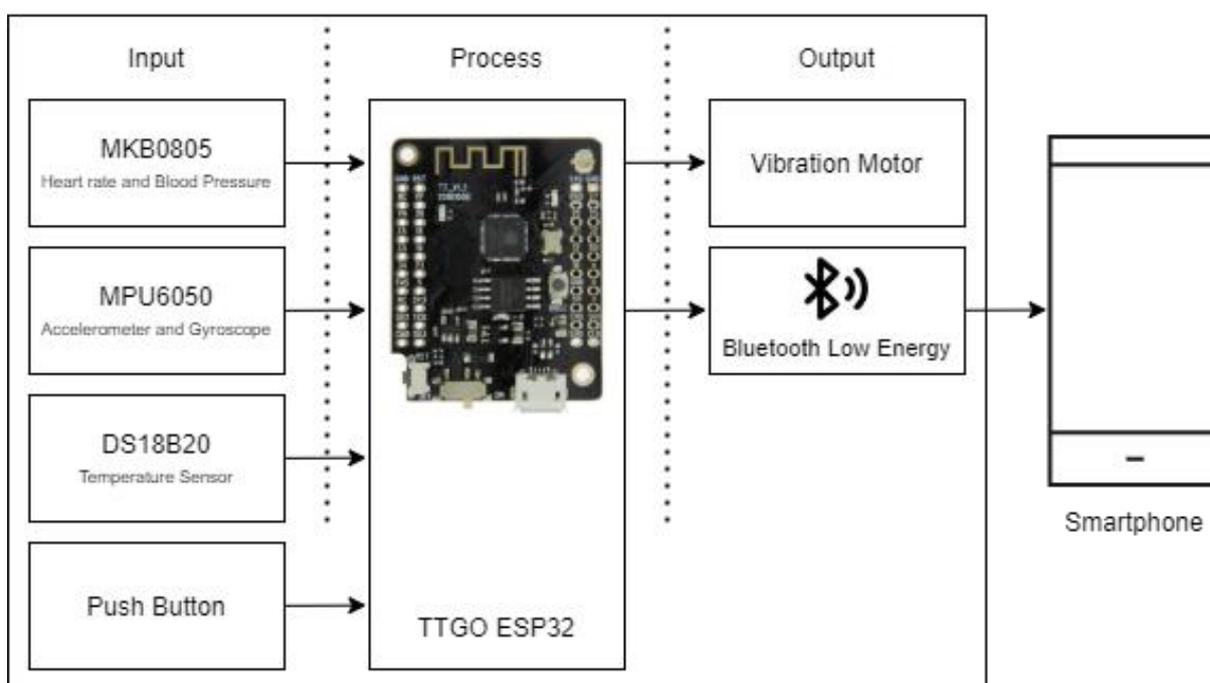
A Cloud Layer contém as aplicações API Server e Web App: a primeira fornece uma API para que as demais aplicações possam consultar e armazenar dados do sistema; e a segunda fornece uma *User Interface (UI)* para que os profissionais de saúde possam visualizar e analisar os dados clínicos do paciente e para o gerenciamento do sistema.

Atualmente, a Plataforma de Sensores coleta, analisa e armazena os seguintes dados clínicos do paciente: pressão arterial, batimentos cardíacos e temperatura corporal. Entretanto, a arquitetura modular desse sistema permite capturar outros sinais vitais, bastando para tal a troca ou adição de novos sensores

específicos para os tipos de sinais e correspondentes dados clínicos a serem monitorados.

## 5.4 Camada de Sensores

A Plataforma de Sensores consiste de uma RSCH, a ser colocada no pulso do paciente como um relógio, que capta sinais vitais desse paciente, tais como batimentos cardíacos, pressão arterial e temperatura corpórea. Os dados clínicos, obtidos a partir desses sinais, são transmitidos via BLE para uma aplicação que executa no dispositivo móvel (e.g., smartphone, tablet) do paciente. A Figura 5.3 ilustra os principais módulos da arquitetura do hardware da plataforma de sensores desenvolvida.



**Figura 5.3** Arquitetura de hardware da Plataforma de Sensores

O módulo MKB0805 é um sensor de batimentos cardíacos e pressão arterial, que emprega o método *PhotoPlethysmoGraphy (PPG)* para detectar alterações no volume sanguíneo no leito microvascular do tecido. Esse método, não invasivo e de baixo custo, consiste em aplicar uma fonte de luz na superfície da pele e medir, por meio de um fotodetector, as variações de intensidade de luz causadas pela absorção e reflexão dessa luz pelo tecido da pele (CASTANEDA et al., 2018). Uma

vez que essas variações dependem da quantidade de sangue presente no caminho óptico, esse sensor faz a leitura dos sinais e os cálculos para estimar as pressões sistólica e diastólica, e batimentos cardíacos, dispondo esses dados para leitura pelo módulo principal.

O módulo MPU6050 é um sensor que mede a aceleração de um objeto de acordo com três eixos de coordenadas (X, Y, Z), sendo utilizado para detectar os movimentos do paciente necessários à habilitação do módulo MKB0805 para a leitura de dados clínicos. O módulo DS18B20 é um sensor digital de temperatura, que mede temperaturas entre  $-55^{\circ}\text{C}$  e  $150^{\circ}\text{C}$  com uma precisão de  $\pm 0,5^{\circ}\text{C}$ , o qual é utilizado para medir a temperatura corporal do paciente.

Os sinais e dados obtidos pelos sensores são processados pelo módulo TTGO T7 V1.3 MINI 32, uma placa de hardware para o desenvolvimento de projetos de IdC. Essa placa é equipada com um o microcontrolador ESP32 da Espressif Systems de baixo custo, o qual consome pouca energia e dispõe de conectividades Wi-Fi 802.11b e Bluetooth v4.2. A plataforma dispõe ainda do *Vibration Motor*, que é um pequeno motor capaz de gerar vibrações para alertar o paciente quando situações críticas são detectadas.

#### **5.4.1 Plataforma de Sensores**

A Figura 5.4 ilustra o esquema do circuito de hardware da plataforma de sensores, cujo os componentes principais são: TTGO T7 V1.3 MINI 32, MKB0805, MPU6050, DS18B20, Vibration Motor, Botão, Led e Bateria. Três tipos de barramentos são empregados para a comunicação entre o módulo TTGO T7 V1.3 MINI 32 e os módulos sensores: o Serial, para a comunicação com o módulo MKB0805; o I2C, para a comunicação com o módulo MPU6050; e o OneWire, para a comunicação com o módulo DS18B20.

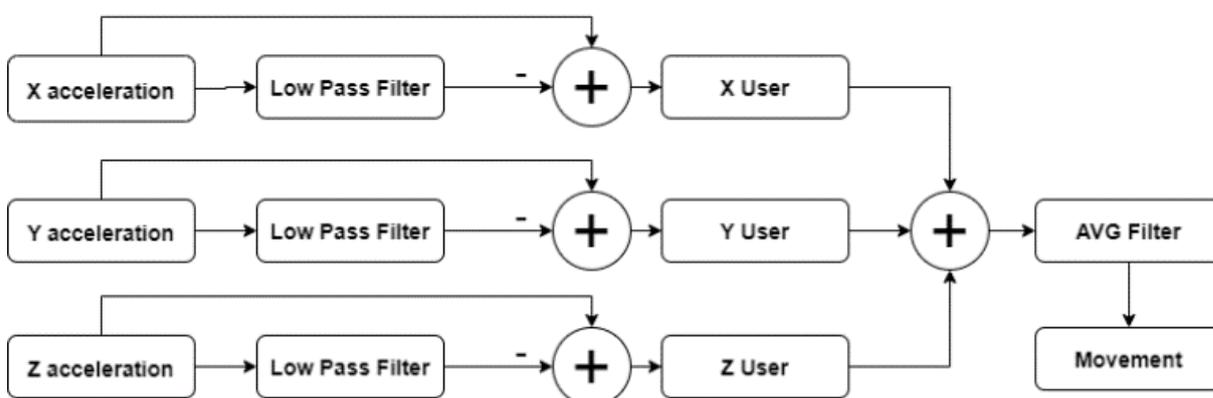


processadores disponíveis no microcontrolador ESP32 e um melhor gerenciamento dos vários periféricos disponíveis.

O software embarcado na plataforma executa essencialmente uma Tarefa Principal, uma Tarefa de Aquisição de Dados e uma Tarefa BLE. A Tarefa Principal é executada quando a plataforma é acionada, sendo responsável por: configurar os periféricos; inicializar os sensores; criar os serviços e discriminar as características de cada sensor; registrar os serviços criados junto ao GATT Server da BLE; e criar as demais tarefas alocando-as dinamicamente. A Tarefa de Aquisição de Dados é responsável por adquirir os dados dos módulos sensores, processá-los e os armazenar em memória não volátil, até os mesmos serem enviados via BLE para a aplicação no smartphone do paciente. A Tarefa BLE é responsável por manter a comunicação BLE com o smartphone do paciente.

#### 5.4.2 Pressão Arterial e Batimentos Cardíacos

Uma vez que a medição das pressões arteriais sistólica e diastólica têm sua precisão afetada pelo movimento do usuário, a plataforma proposta monitora esse movimento e efetua a leitura de tais dados, somente quando o usuário se encontra em repouso. Para realizar esse monitoramento, os sinais correspondentes a aceleração em cada um dos três eixos de coordenadas (X, Y, Z) são obtidos do módulo MPU6050 a uma taxa de amostragem de 60Hz e são filtrados, conforme ilustrado na Figura 5.6.



**Figura 5.6 Diagrama do filtro digital para a detecção do movimento do paciente**

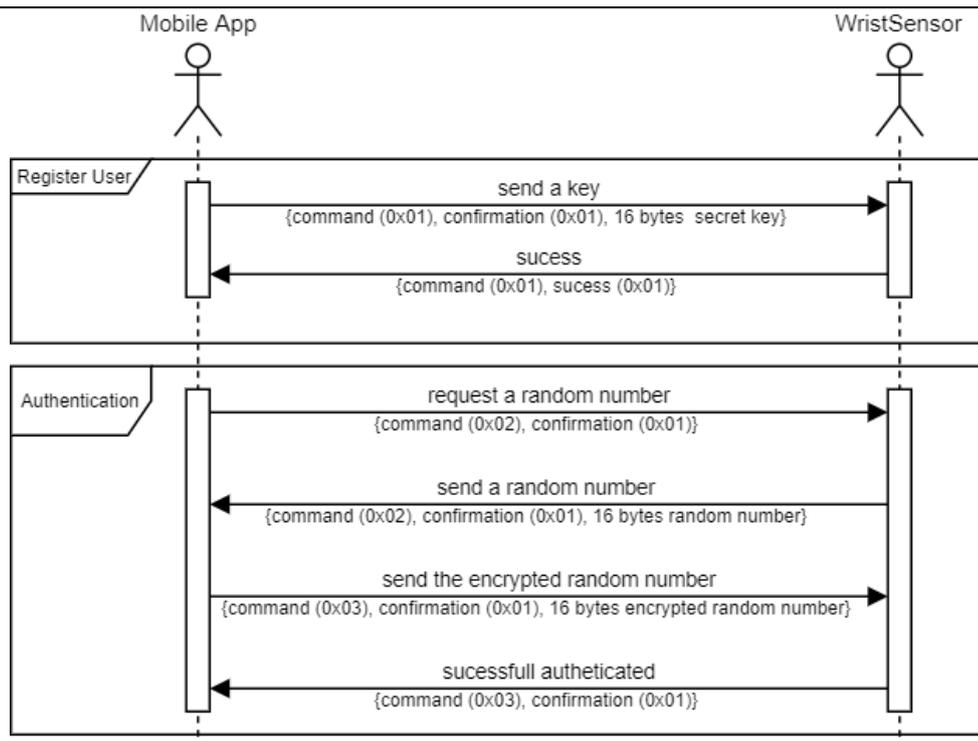
Tendo em vista que além da aceleração do usuário propriamente dita, outra componente presente em cada sinal obtido é a força da gravidade, o processo de filtragem ilustrado na Figura 5.6 busca isolar as acelerações do usuário e obter um

único sinal que representa somente o movimento do usuário. Neste sentido, inicialmente a cada sinal é aplicado um filtro passa baixas do tipo *Infinite Impulse Response (IIR)*, o qual atenua todas as frequências maiores que 0.2 Hz, tendo como resultado o sinal referente à força da gravidade que incide sobre cada eixo de coordenada. Em seguida, a aceleração do usuário em cada eixo é obtida subtraindo a força da gravidade da aceleração total do eixo correspondente. Finalmente, um único sinal é obtido por meio da soma vetorial dessas acelerações do usuário, sendo que ao resultado obtido é aplicado um filtro de média móvel para atenuar os ruídos, o qual tem como saída o sinal que representa o movimento do usuário.

Os dados, tais como os das pressões sistólica e diastólica e dos batimentos cardíacos, são obtidos pela plataforma em intervalos de 30 minutos, quando detectado o repouso do paciente (HERMIDA et al., 2007; STOJANOVIC, 2018). Para tal, o movimento do usuário é analisado com base em um valor limite predefinido e, caso esse valor não seja ultrapassado em um intervalo de 02 minutos, o sensor MKB0805 é habilitado e o valor médio, de um total de 20 amostras de cada dado, é enviado ao dispositivo móvel do paciente. Caso seja detectado movimento do paciente durante o intervalo de coleta das amostras, estas são descartadas e novas amostras são coletadas tão logo seja detectado o repouso do mesmo. Se o processo de coleta de dados não for concluído em um tempo predefinido de até 5 minutos, o paciente é notificado via uma vibração na plataforma a fim de que permaneça em repouso, sendo novamente notificado na conclusão desse processo.

### 5.4.3 Segurança dos Dados

A segurança dos dados na plataforma proposta foi implementada em duas etapas. Na primeira, de acordo com os métodos e protocolos definidos na camada de segurança da *Bluetooth Core Specification*, a plataforma e o smartphone executam o processo de pareamento e estabelecem uma conexão encriptada. Na segunda etapa, ocorre o processo de registro e autenticação da aplicação no smartphone, ilustrado na Figura 5.7, o qual é necessário para a leitura dos dados coletados pela plataforma.



**Figura 5.7 Diagrama de sequência do processo de registo e autenticação da aplicação no dispositivo móvel**

O registo da aplicação é efetuado somente a primeira vez em que ocorre o pareamento entre o smartphone e a plataforma. Para tal, a aplicação no smartphone envia um comando de registo e uma chave de 16 bytes para a plataforma, o qual armazena essa chave e retorna uma mensagem de sucesso. Finalizado o registo, a aplicação no smartphone inicia a autenticação requisitando uma sequência de 16 bytes gerado aleatoriamente na plataforma, o qual deve ser encriptada utilizando a chave registrada, sendo que o valor obtido deve ser enviado para a plataforma. Na plataforma, a sequência criptografada pela aplicação no smartphone é comparada com a mesma sequência gerada e também criptografada utilizando a mesma chave. Caso a autenticidade seja confirmada, a aplicação é autorizada e pode ler os dados provenientes da plataforma.

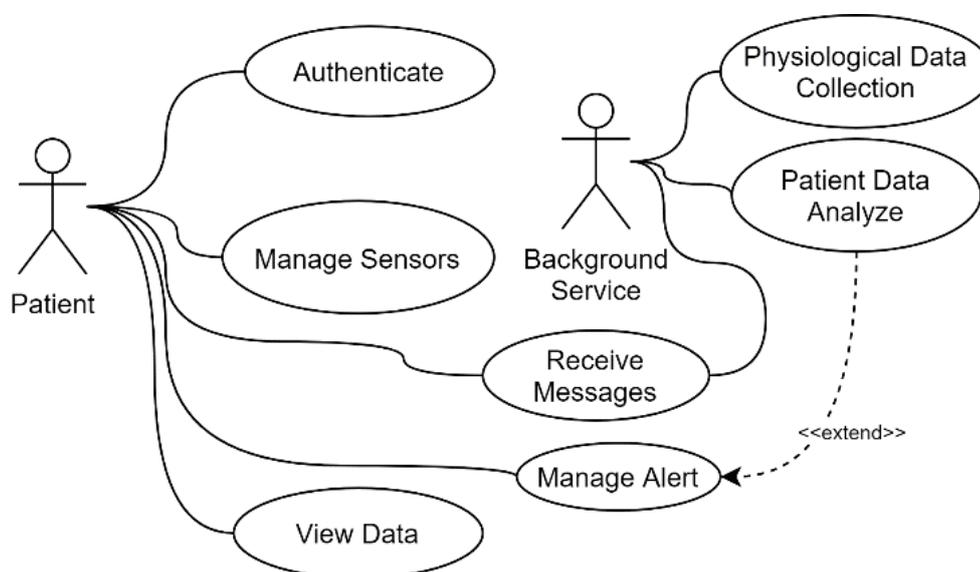
## 5.5 Camada Fog

A Camada Fog permite atribuir aos dispositivos mais próximos da RSCH parte da responsabilidade de processamento e análise dos sinais e dados e, ao promover

o processamento distribuído, reduz o investimento em recursos computacionais na nuvem, permite a transferência de uma quantidade menor de dados para a nuvem, contribui para a geração de feedbacks mais rápidos para o paciente, e evita a sobrecarga de rede. No SBIdC-MPH esta camada é composta pelo smartphone do paciente e a aplicação Mobile App.

### 5.5.1 Aplicação Móvel

A Mobile App é uma aplicação para dispositivos Android, instalada em um smartphone ou tablet do paciente, que atua principalmente como um gateway, recebendo dados da Plataforma de Sensores e os encaminhando à nuvem computacional. O design dessa aplicação envolveu a identificação, a análise e a descrição dos requisitos, sendo que foram empregados os diagramas de Caso de Uso, de Sequência e de Classe da *Unified Modeling Language (UML)*. A Figura 5.8 apresenta o diagrama de Caso de Uso da Mobile App, focando nos atores, interações e principais funcionalidades dessa aplicação.



**Figura 5.8 Diagrama de Caso de Uso da Mobile App**

Conforme ilustrado na Figura 5.8, dois atores interagem com a Mobile App: Patient, e Background Service. O primeiro ator representa o paciente com hipertensão, cujos dados clínicos relevantes a essa DCNT são monitorados, sendo que este é responsável por lidar com as seguintes funcionalidades:

(a) Authenticate, o paciente fornece o nome de usuário e senha previamente registrados no Web App e, uma vez autorizado, este pode gerenciar os sensores, visualizar seus dados clínicos, mensagens e alertas de risco;

(b) Manage Sensors, o paciente efetua o registro de novos sensores e pode verificar o status de conexão e o nível de bateria dos sensores; e

(c) View Data, o paciente consulta os registros de dados das últimas 24 horas, que ficam armazenados e disponibilizados em um banco de dados no smartphone do paciente.

O segundo ator representa os componentes de serviço, executados em segundo plano no dispositivo Android, responsáveis pelo provimento de forma automática das seguintes funcionalidades:

(a) Physiological Data Collection, uma vez efetuado pelo Patient o registro dos sensores, é iniciado o processo de conexão e de autenticação da Mobile App com a Plataforma de Sensores. Após realizada a conexão, os dados clínicos são coletados, tratados e enviados à nuvem, onde são armazenados e disponibilizados para análise pelos profissionais de saúde;

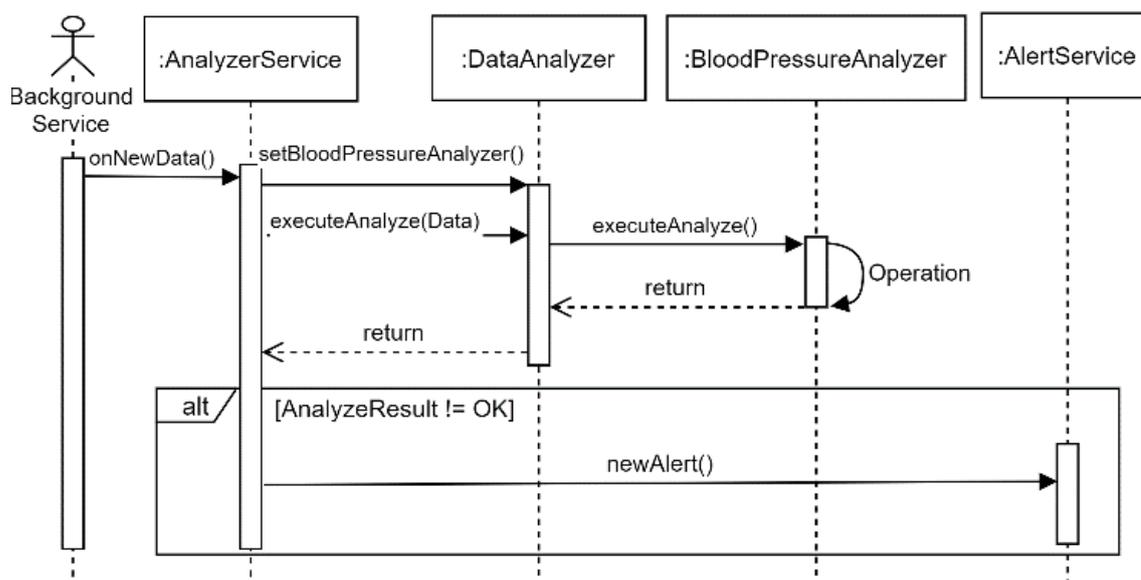
(b) Patient Data Analyze, os dados recebidos são analisados e classificados de acordo com quatro níveis de risco à saúde do paciente: NORISK, quando não há risco; LOW, quando o risco é pequeno, sendo possível o controle da hipertensão via mudanças no estilo de vida do paciente, tais como a prática de exercícios físicos e hábitos alimentares mais saudáveis; MODERATE, quando o risco exige, além da mudança de estilo de vida, o uso de medicamentos para o controle da hipertensão; e HIGH, quando há uma crise e o paciente necessita da intervenção imediata de um profissional de saúde;

(c) Manage Alert, alertas são gerados de acordo com o nível de risco e enviados à nuvem para análise por um profissional de saúde, sendo que o paciente recebe automaticamente notificações relativas a esses alertas no seu smartphone e na Plataforma de Sensores. Em situações de risco MODERATE ou HIGH, o profissional de saúde e o familiar, cadastrado como contato de emergência, recebem imediatamente uma mensagem SMS com informações sobre o estado de saúde do paciente; e

(d) Receive Messages, as mensagens registradas pelo profissional de saúde no Web App, tais como recomendações de mudanças no estilo de vida, ajustes no tratamento envolvendo alterações nas dosagens dos medicamentos ou a prescrição

de novos medicamentos, são obtidas e armazenadas no smartphone do paciente, sendo que este recebe notificações relativas às mesmas.

Os casos de uso foram refinados via diagramas de Sequência, a fim de possibilitar a visualização das interações envolvidas na execução das funcionalidades da Mobile App. A Figura 5.9 apresenta um exemplo desse diagrama para a funcionalidade Patient Data Analyze, focando na análise dos dados relativos à Pressão Arterial.

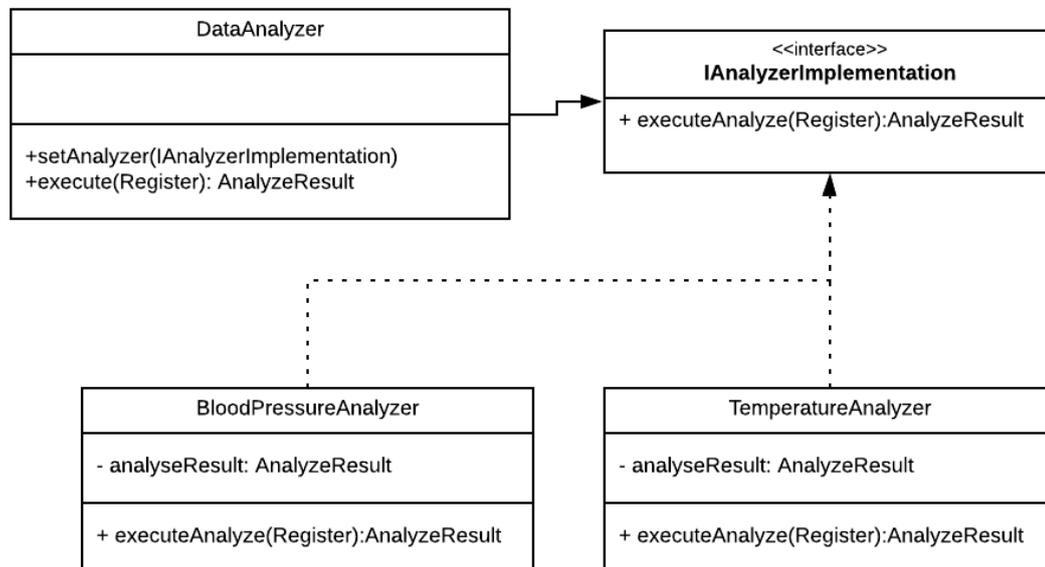


**Figura 5.9 Diagrama de Sequência para a funcionalidade Patient Data Analyze**

Conforme ilustrado na Figura 5.9, as seguintes interações ocorrem na execução da funcionalidade Patient Data Analyze: inicialmente, ao receber os dados da Plataforma de Sensores, o BackgroundService emite a mensagem OnNewData para o AnalyzerService; em seguida, o AnalyzerService configura a instância do DataAnalyzer para tratar dados de Pressão Arterial, sendo os dados recebidos repassados para o DataAnalyzer, que por vez, realiza a análise por meio das rotinas definidas no BloodPressureAnalyzer; finalmente, o resultado dessa análise é retornado, sendo que se for detectado algum tipo de risco ao paciente, um alerta é gerado pelo AnalyzerService e tratado pela funcionalidade ManageAlert.

Dando continuidade ao design da Mobile App, as suas funcionalidades foram refinadas via diagramas de Classes. Por exemplo, Patient Data Analyze foi implementada empregando-se o padrão de projeto Strategy, que permite a definição de uma família de algoritmos e a seleção, em tempo de execução, de um destes para uso. O emprego desse padrão possibilita a extensão do Mobile APP via a

adição de novos analisadores de dados, sem que seja necessário mudar os já existentes. A Figura 5.10 apresenta um exemplo de Diagrama de Classes para a funcionalidade Patient Data Analyze, onde o Strategy é empregado.



**Figura 5.10 Diagrama de classes para a funcionalidade Patient Data Analyze**

Conforme ilustrado na Figura 5.10, há quatro classes principais classes da funcionalidade Patient Data Analyze relacionadas ao Strategy. As classes **BloodPressureAnalyzer** e **TemperatureAnalyzer**, denominadas estratégias no padrão Strategy, implementam a análise dos dados clínicos do paciente, sendo a primeira referente à pressão arterial e a segunda à temperatura corporal. A classe **DataAnalyzer** contém uma referência para uma das estratégias de análise, a qual é configurada em tempo de execução pelos clientes dessa classe. As estratégias implementam a interface **IAnalyzerImplementation**, a qual expõe o método único `executeAnalyze`, para a execução do algoritmo selecionado. A Figura 5.11 apresenta um trecho do código fonte do método `executeAnalyze` da classe **BloodPressureAnalyzer**, no qual é realizada a análise dos dados de pressão arterial.

```

public AnalyzeResult executeAnalyze(Register register) {
    if (register.getType() == RegisterType.BLOOD_PRESSURE) {
        Data dia =
            register.getDataByType(DataType.DIASTOLIC);
        Data sys =
            register.getDataByType(DataType.SYSTOLIC);
        if (sys.getValue() < 120 && dia.getValue() < 80) {
            analyzeResult.setStatus(RiskStatus.NORISK);
        }
        else if ((sys.getValue() >= 120 && sys.getValue() <= 139) ||
            (dia.getValue() >= 80 && dia.getValue() <= 89)) {
            analyzeResult.setStatus(RiskStatus.LOW);
        }
        else if ((sys.getValue() >= 140 && sys.getValue() < 180) ||
            (dia.getValue() >= 90 && dia.getValue() < 120)) {
            analyzeResult.setStatus(RiskStatus.MODERATE);
        } else {
            analyzeResult.setStatus(RiskStatus.HIGH);
        }
    }
    return analyzeResult;
}

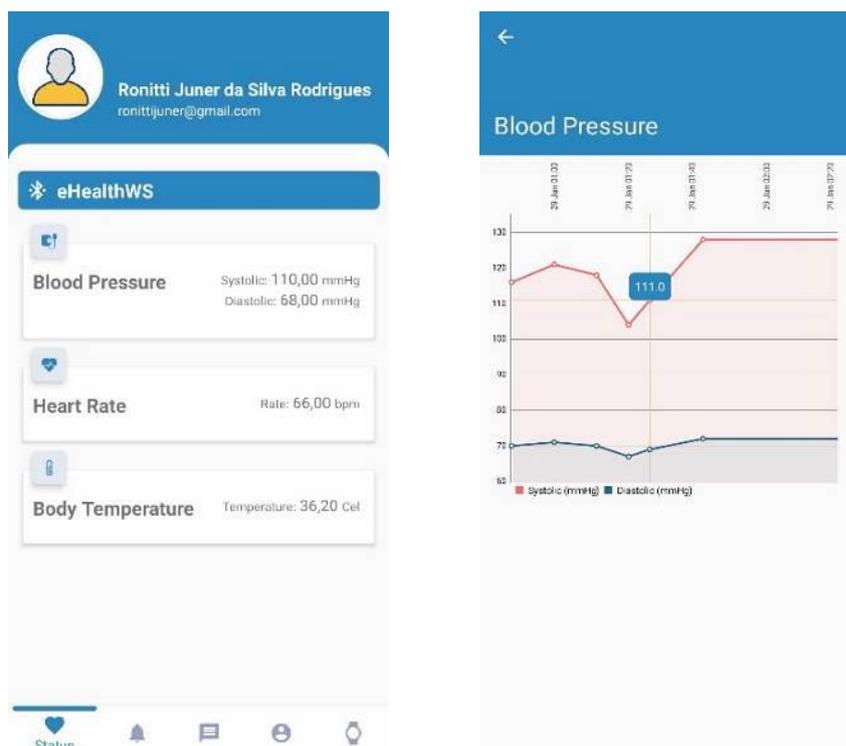
```

**Figura 5.11 Trecho do código fonte do método executeAnalyze**

Conforme ilustrado na Figura 5.11, inicialmente é verificado se o registro de dados é do tipo BLOOD\_PRESSURE e, em caso positivo, os valores das pressões DIASTOLIC e SYSTOLIC são capturados, para finalmente o RiskStatus ser classificado de acordo com as recomendações da AHA, apresentadas na seção 2.5.

A partir das funcionalidades da Mobile APP descritas via diagramas UML, essa aplicação foi implementada na linguagem Java, usando a API de desenvolvimento para aplicações Android na versão 21, que a torna compatível com cerca de 94,1% dos dispositivos baseados em Android. A fim de garantir a segurança dos dados armazenados no smartphone ou tablet do usuário, foi utilizado a biblioteca SQLite com a extensão SQLChipher, que fornece ao banco de dados uma criptografia AES de 256 bits. Foi empregada a biblioteca Apollo GraphQL na versão 2.4, que permite gerar os modelos em Java para as consultas GraphQL (JEON et al. 2019).

A Figura 5.12 apresenta duas capturas de tela da Interface do Usuário, que são mostradas no smartphone do usuário: (a) a tela interativa de Coleta de Dados, que permite ao usuário visualizar os sensores e os dados atualizados coletados por cada sensor, e também clicar sobre cada um destes para ter acesso à tela de visualização do histórico desses dados; (b) a tela de Histórico de Dados, que permite visualizar por meio de um gráfico o histórico dos dados (na figura Blood Pressure), os quais estão registrados no smartphone do paciente.



(a) Coleta de Dados

(b) Histórico dos Dados

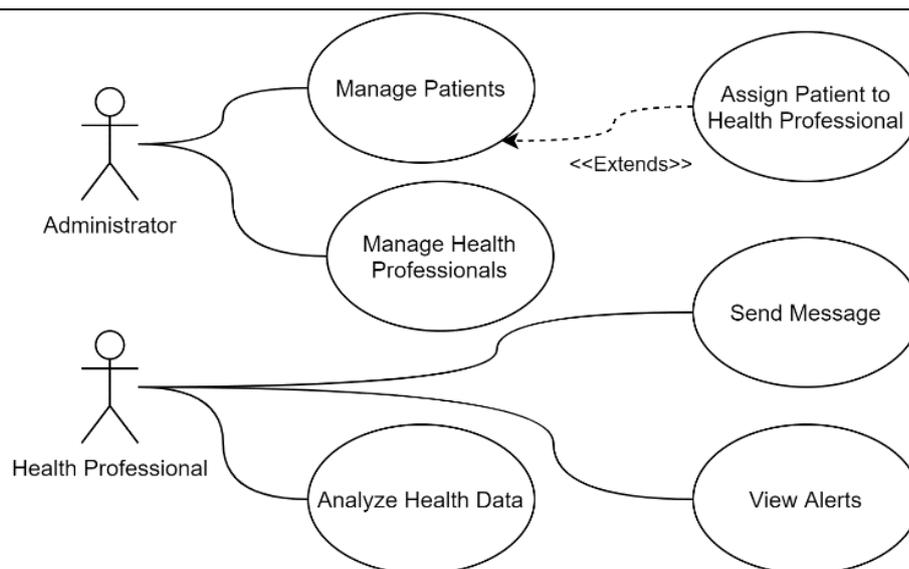
**Figura 5.12 Interface do Usuário da Mobile App**

## 5.6 Camada da Nuvem

As aplicações API Server e Web App foram implantadas na Cloud Layer, a fim de possibilitar o uso de recursos sob demanda da nuvem, já que estas desempenham tarefas que exigem um maior poder computacional, e a fim de possibilitar o seu acesso via Internet, tanto por parte do Mobile App quanto dos profissionais de saúde.

### 5.6.1 Aplicação Web

A aplicação Web App é acessada via um navegador Web pelos profissionais de saúde, para a visualização e análise dos dados clínicos de seus pacientes, e também por um administrador, para o gerenciamento desses profissionais e desses pacientes. A Figura 5.13 apresenta o diagrama de Caso de Uso da Web App, focando nos atores, interações e principais funcionalidades dessa aplicação.



**Figura 5.13 Diagrama de Caso de Uso da Web App**

Conforme ilustrado na Figura 5.13, dois atores interagem com a Web App: Administrator e Health Professional. O primeiro ator representa um usuário que desempenha o papel de administrador da aplicação, sendo que este é responsável por lidar com as seguintes funcionalidades:

(a) Manage Patients, realiza o cadastro e a manutenção dos dados dos pacientes, tais como seus dados pessoais e os tipos de doenças a serem monitorados;

(b) Assign Patient to Health Professional, atribui um profissional de saúde ao paciente, o qual ficará responsável pelo acompanhamento dos dados clínicos desse paciente; e

(c) Manage Health Professionals, realiza o cadastro e manutenção dos dados dos profissionais de saúde.

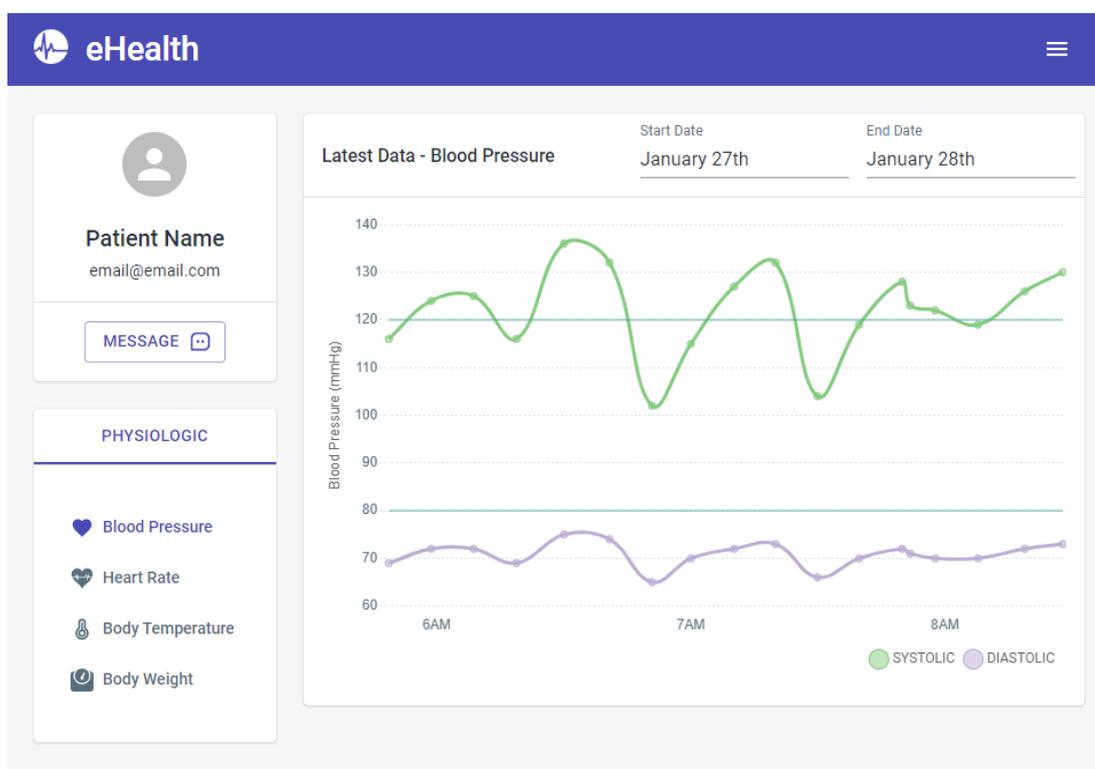
O segundo ator representa o profissional de saúde, sendo que este é responsável por lidar com as seguintes funcionalidades:

(a) Analyze Health Data, acessa os dados clínicos dos pacientes do profissional de saúde, o qual pode então analisar tais dados e tomar decisões relativas aos tratamentos desses pacientes;

(b) Send Message, envia mensagens ao smartphone do paciente via a Mobile App, com recomendações relativas ao tratamento ou ao estilo de vida desse paciente. Tais mensagens podem ser categorizadas pelo profissional de saúde em 3 níveis de prioridade, visando a determinar a ordem na qual serão exibidas ao paciente; e

(c) View Alerts, visualiza os alertas gerados na Mobile App automaticamente e que foram encaminhados ao API Server, auxiliando o profissional de saúde na análise dos dados clínicos de seus pacientes.

A Figura 5.14 apresenta uma captura da tela de Visualização de Dados do paciente, que possibilita ao profissional de saúde aplicar filtros, tais como o tipo de dados clínicos e o período desejados para a análise. A fim de auxiliá-lo nessa análise, para cada tipo de dados clínicos apresentado, são plotadas no gráfico linhas de referência relativas a valores normais pré-estabelecidos para esse tipo de dados.



**Figura 5.14 Captura da tela de Visualização de Dados de paciente**

A Web App foi implementada na linguagem Javascript e foram utilizadas as bibliotecas React e Material UI, uma vez que estas possibilitam uma rápida construção da Interface de Usuário. A integração com o API Server foi realizada via a biblioteca Apollo Client, uma vez que esta permite executar as consultas a uma API GraphQL, realizar o gerenciamento de cache de dados e atualizar automaticamente os dados da Interface de Usuário.

## 5.6.2 Servidor API

A aplicação API Server opera como um servidor, fornecendo uma API segura para que as aplicações Mobile App e Web App possam armazenar e recuperar as informações do sistema. Essa API emprega a linguagem de consulta e manipulação de dados GraphQL, visando a redução de sobrecarga de rede, respostas mais rápidas, a comunicação com diferentes aplicações, a possibilidade de evoluir o sistema mantendo a compatibilidade com as aplicações já existentes, e a simplicidade no processo de desenvolvimento da API.

A API fornece um único *endpoint*, através do qual as demais aplicações do SBIdC-MPH armazenam ou recuperam dados do sistema. Este *endpoint* foi definido por meio do modelo de *Schemas* do GraphQL, o qual contém todas as funcionalidades disponíveis no sistema. Um trecho do código, que especifica o *Schema* da API Server e descreve os recursos *Patient*, *Register* e *Data*, é ilustrado na Figura 5.15.

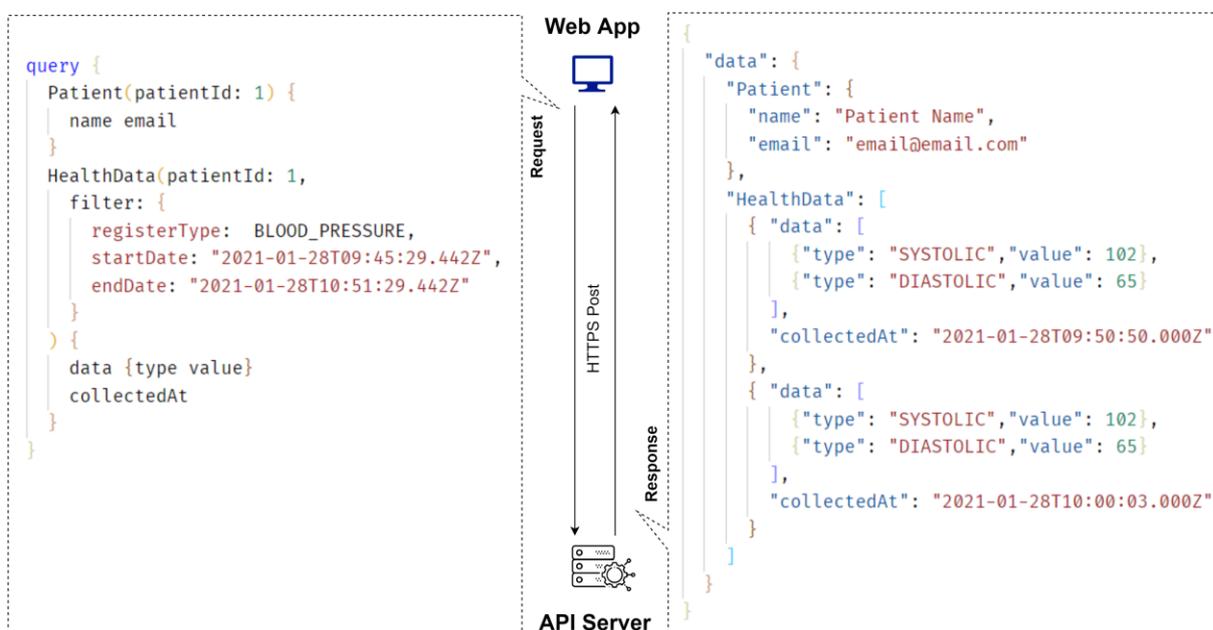
```
type Patient {
  id: Int
  name: String
  birth_date: Date
  gender: EnumGender
  susNumber: String
  phoneNumber: String
  email: String
  cpf: String
  addresses: [Address]!
}

type Data {
  type: DataType
  value: Float
}

type Register {
  type: RegisterType!
  data: [Data]!
  createdAt: Date
  collectedAt: Date
}
```

Figura 5.15 Trechos do código do Schema para a API Server

A Figura 5.16 ilustra um trecho de código de uma requisição GraphQL, realizada pela aplicação Web App ao API Server para obter os dados a serem mostrados na tela de Visualização de Dados do paciente, e uma resposta típica à mesma. Nessa requisição, um argumento `patientId` é fornecido com valor "1", para recuperar tanto o recurso `Patient`, requisitando o nome e email, quanto o recurso `HealthData`, requisitando os dados clínicos e a data da coleta. Em `HealthData` é fornecido o argumento adicional `filter` visando obter dados do tipo `BLOOD_PRESSURE` no intervalo entre `startDate` e `endDate`. Ao receber essa requisição, o API Server realiza o processamento da consulta e retorna somente os dados requisitados no formato JSON.



**Figura 5.16 Trecho do código de uma requisição GraphQL e resposta à mesma**

A API Server foi implementada na linguagem JavaScript e foi utilizado o Node.js como ambiente de execução server-side, uma vez que essa tecnologia tem apresentado melhor performance, quando comparada às tecnologias tradicionais para servidores (CHITRA; SATAPATHY, 2020; LEI; MA; TAN, 2014). Foi empregada a biblioteca open-source Apollo Server na versão 2.16, a qual permite criar a API GraphQL compatível com qualquer cliente GraphQL. Essa API provê um mecanismo de autenticação e autorização que utiliza JSON Web Token (JWT), mantendo assim os dados durante uma comunicação segura via o protocolo HTTPS.

## **5.7 Considerações Finais**

Este capítulo apresentou o SBIdC-MPH. Foram descritas informações detalhadas sobre as aplicações desse sistema, assim como as funcionalidades disponibilizadas para o monitoramento de pacientes com Hipertensão Arterial e metodologia empregada no processo de desenvolvimento. Também foi apresentado a arquitetura de hardware da plataforma de sensores desenvolvida para a coleta de dados fisiológicos e a metodologia para a coleta de dados de pressão arterial. A plataforma de sensores utiliza um sistema de segurança de duas etapas a fim de garantir que os dados do paciente estão sendo obtidos apenas por seu dispositivo móvel. O SBIdC-MPH utiliza a tecnologia GraphQL a fim de permitir a maior escalabilidade do sistema.

# Capítulo 6

## EXPERIMENTO E ANÁLISE DOS RESULTADOS

---

---

*Este capítulo discorre sobre um experimento realizado visando à escolha da melhor tecnologia a ser empregada para a comunicação entre as aplicações do SBIdC-MPH.*

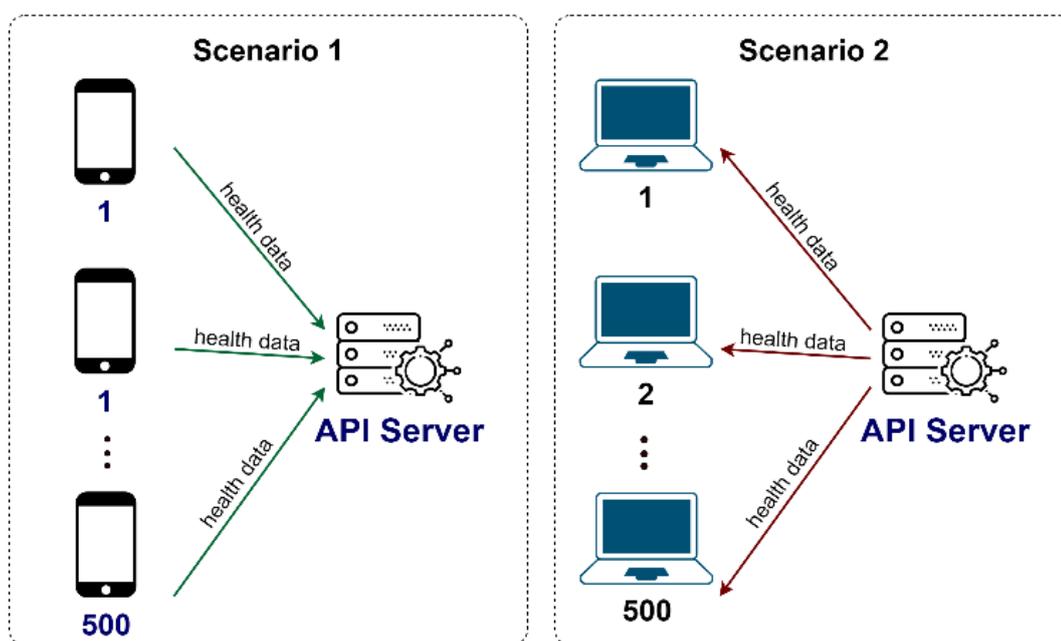
### 6.1 Considerações Iniciais

Tendo em vista o grande número de usuários que podem utilizar o sistema e os consequentes gargalos que podem ocorrer tanto na comunicação entre as aplicações Mobile App e API Server, quanto na comunicação entre as aplicações Web App e API Server, durante o projeto do sistema foram considerados o uso das tecnologias REST e GraphQL para a comunicação entre as aplicações, sendo que um experimento foi realizado, similar ao reportado em (BRITO; VALENTE, 2020), visando a fornecer subsídios para a escolha da tecnologia a ser utilizada.

O experimento foi realizado buscando avaliar o Tempo de Resposta e Vazão Média do sistema, quando essas tecnologias são utilizadas na implementação da API responsável pela comunicação com as aplicações Web App e Mobile App. Como o sistema foi desenvolvido em camadas de modo a separar as regras de negócio da camada de API, foi possível realizar a avaliação das tecnologias e minimizar o impacto provocado por alterações na implementação.

## 6.2 Descrição do Experimento

Para a condução do experimento foi empregado o Apache JMeter (HALILI, 2008), uma ferramenta para realizar testes de performance de funcionalidades e de carga em aplicações desenvolvidas para Web, sendo que os testes foram configurados em dois cenários conforme ilustrado na Figura 6.1.



**Figura 6.1** Visão geral dos cenários de simulação

Os cenários ilustrados na Figura 6.1 simulam de 1 até 500 usuários realizando requisições concorrentemente ao API Server, sendo que: o Cenário 1 simula o envio de dados clínicos de pacientes via a aplicação Mobile App, tais como os das pressões sistólica e diastólica e dos batimentos cardíacos, assim como a data e a hora da coleta desses dados; e o Cenário 2 simula as consultas realizadas pelos Profissionais de Saúde via a aplicação Web App para obter os dados pessoais e clínicos dos pacientes, considerando-se 20 registros de saúde por paciente. É importante ressaltar que no cenário 2, devido às diferenças entre o GraphQL e o REST já apresentadas anteriormente, para obter os dados clínicos e pessoais de um paciente utilizando REST são necessárias 2 requisições, uma para os dados pessoais e outra para os dados clínicos, enquanto que, utilizando GraphQL apenas uma requisição é necessária. As simulações foram executadas durante o intervalo de 1 hora para ambas a tecnologias REST e GraphQL.

Para minimizar as interferências, que poderiam ser causadas por instabilidades da Internet, as simulações foram realizadas em ambiente controlado composto de dois computadores, um atuando como Servidor e o outro como Cliente, interligados por uma rede local IEEE 802.3 de 100Mbps, conforme ilustrado na Figura 6.2.



**Figura 6.2 Ambiente de simulação**

O Servidor possui um processador Intel Core I5 3337U 1.8GHz, uma memória RAM de 8GB e um SSD com 32GB, sendo que a aplicação API Server e o banco de dados PostgreSQL foram empacotados em containers Docker e executam no Sistema Operacional Ubuntu Server 18.04 LTS. O Cliente possui um processador Intel Core I7 3337U 1.8GHz, uma memória RAM de 8GB e um SSD de 250GB, sendo que o sistema operacional é o Windows 10 Single Language.

### 6.3 Resultados Obtidos

As Figura 6.3 e Figura 6.4 apresentam os gráficos resultantes das simulações para os cenários 1 e 2 respectivamente, sendo que estes mostram a relação entre o tempo de resposta médio do API Server e o número de usuários que realizam requisições por meio das aplicações Mobile App e Web App respectivamente. A partir dos resultados apresentados, é possível perceber que em ambos os cenários o tempo de resposta médio foi menor quando utilizado o GraphQL, e que esse comportamento tende a se manter na medida em que aumenta o número dos usuários.

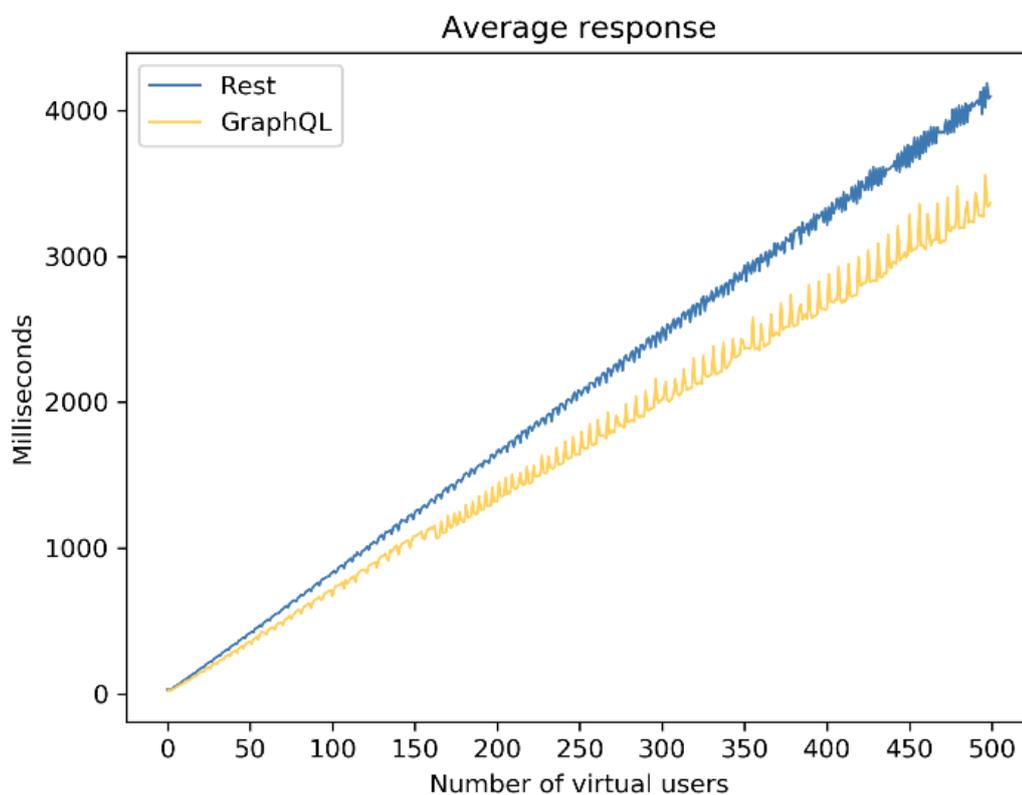


Figura 6.3 Tempo de Resposta Médio para o envio de dados clínicos

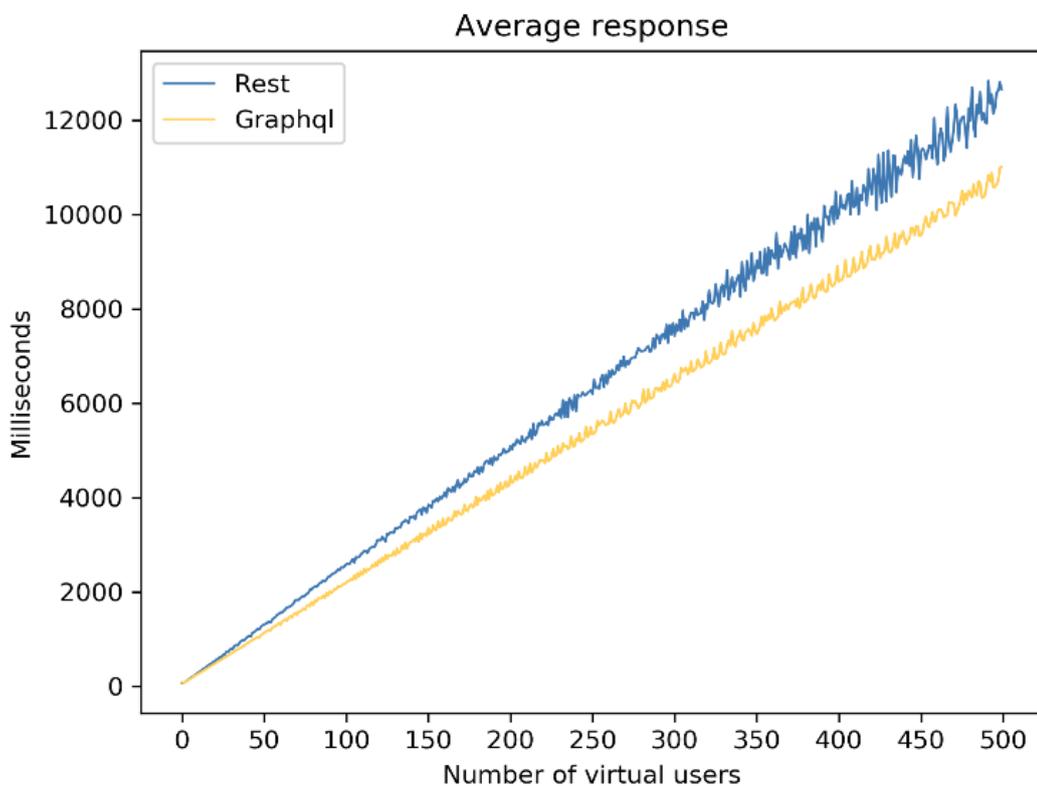


Figura 6.4 Tempo de Resposta Médio para a consulta de dados pessoais e clínicos

As Tabela 6.1 e Tabela 6.2 apresentam os resultados das simulações para os cenários 1 e 2 respectivamente, sendo que estas contêm os tempos de resposta mínimo e máximo, e a taxa média em Kilobytes por segundo dos dados enviados e recebidos. Observa-se que, em ambos os cenários, esses tempos mantiveram-se menores nas simulações realizadas com o GraphQL, enquanto que as taxas médias de dados enviados e recebidos apresentaram diferenças entre os cenários 1 e 2. No cenário 1, a taxa média manteve-se menor nas simulações realizadas com o REST, uma vez que o GraphQL requer um cabeçalho maior para cada requisição. No Cenário 2, a taxa média de dados enviados para o API Server, tanto usando o GraphQL quanto o REST, foram semelhantes, enquanto que a taxa média de dados recebidos do API Server foi menor usando o GraphQL, uma vez que este permite definir as informações desejadas numa única resposta para cada requisição.

**Tabela 6.1 Resultados da simulação relativos ao envio de dados clínicos**

End Point	Tempo de Resposta (ms)		Rede (KB/s)	
	Min	Max	Recebido	Enviado
REST	16	4332	43.35	70.33
GraphQL	13	3901	52.68	92.37

**Tabela 6.2 Resultados da simulação relativos à recepção de dados pessoais e clínicos**

End Point	Tempo de Resposta (ms)		Rede (KB/s)	
	Min	Max	Recebido	Enviado
REST Dados Clínicos	28	9863	192,04	18,71
REST Dados Pessoais	6	5278	30,13	14,83
REST Total	34	15141	222,17	33,54
GraphQL	26	11210	173.48	33.55

A partir dos resultados obtidos no experimento, é possível concluir que o GraphQL apresentou um menor tempo de resposta tanto para o envio dos dados de saúde dos pacientes, por meio da Mobile App, quanto para consulta destes dados pelos profissionais de saúde através da Web App. Observa-se também que apesar

de a taxa média para envio das requisições ao API Server ser semelhante ao REST, a taxa média para o recebimento dos dados é menor quando utilizado o GraphQL, visto que é possível recuperar todas e somente as informações necessárias em uma única requisição. Em suma, o menor tempo de resposta em conjunto com a redução da taxa média de dados proporcionado pelo emprego do GraphQL, pode reduzir os gargalos da rede, tornando esta tecnologia a mais indicada para uso no sistema.

## 6.4 Considerações Finais

A partir dos resultados obtidos nesse experimento, é possível concluir que o uso do GraphQL acarreta em um menor tempo de resposta que o uso do REST, tanto para o envio de dados clínicos de pacientes via a Mobile App, quanto para a consulta de dados pessoais e clínicos de pacientes via a Web App por parte de profissionais de saúde. Observa-se também que, apesar das taxas médias para o envio de requisições ao API Server serem semelhantes em ambos os usos, a taxa média para a recepção de dados é menor quando utilizado o GraphQL do que quando utilizado o REST, visto que com o primeiro é possível recuperar em uma única requisição todas e somente as informações desejadas. Concluindo, o menor tempo de resposta, conjuntamente com a redução da taxa média de dados, que o uso do GraphQL proporciona, podendo assim reduzir os gargalos do sistema, levou ao emprego dessa tecnologia no SBIdC-MPH.

# Capítulo 7

## CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

---

---

Esta dissertação de mestrado apresentou o Sistema Baseado em Internet das Coisas para o Monitoramento de Pacientes com Hipertensão (SBIdC-MPH), um sistema que emprega tecnologias da IdC e das Computações Ubíqua, Pervasiva e Nuvem para o monitoramento remoto de pacientes com Hipertensão Arterial.

O SBIdC-MPH visa a fornecer serviços de saúde de qualidade e de modo onipresente, contribuindo para diagnósticos mais precisos e tratamentos mais efetivos. Esse sistema outorga ao paciente, autonomia para o gerenciamento da própria saúde, na medida em que possibilita ao mesmo acompanhar a evolução de seus dados clínicos via o seu dispositivo móvel.

Para tal, o SBIdC-MPH emprega uma Plataforma de Sensores para a coleta de dados clínicos do paciente, tais como pressão arterial, batimentos cardíacos e temperatura, que são pré-processados no dispositivo móvel do paciente e posteriormente encaminhados à nuvem, onde são armazenados, tratados e ficam disponíveis para a análise pelos profissionais de saúde. Esse sistema é ainda capaz de gerar alertas, baseados em limites pré-definidos, que são enviados ao paciente, ao profissional de saúde, ou pessoas designadas pelo paciente, sempre que detectadas situações de risco à saúde do paciente.

Um experimento foi realizado para avaliar o tempo de resposta e a vazão de dados, quando empregadas as tecnologias GraphQL e REST para a comunicação entre as aplicações do SBIdC-MPH. Uma vez que este experimento mostrou que o emprego de GraphQL produziu um menor tempo de resposta e uma redução da taxa

média de dados, quando comparado ao emprego do REST, o primeiro foi escolhido para a comunicação entre as aplicações do SBIdC-MPH.

## 7.1 Limitações

A primeira limitação está relacionada aos testes e à avaliação de usabilidade da Plataforma de Sensores, que não foram possíveis de serem realizados em decorrência do cenário de pandemia durante o processo de desenvolvimento do trabalho. Os testes seriam realizados para verificar a acurácia dos dados coletados pela plataforma de sensores, comparando os valores medidos pelos sensores com os valores medidos por profissionais de saúde do Hospital Universitário (HU) da UFSCar, utilizando equipamentos já certificados e devidamente calibrados. Já a avaliação seria realizada por profissionais de saúde e pacientes do HU/UFSCar, visando a avaliar a facilidade de uso e a utilidade percebida do SBIdC-MPH por esses usuários.

A segunda limitação está ligada a ausência de uma aplicação mobile para uso pelos profissionais de saúde, a qual seria um grande facilitador para o acompanhamento dos pacientes, fornecendo acesso rápido às informações e permitindo a visualização dos dados coletados pelos sensores em tempo real. Tal aplicação também facilitaria a comunicação mais efetiva dos profissionais de saúde para com os pacientes, sobretudo para o fornecimento de informações relativas a tratamentos médico.

A terceira limitação está ligada à atual implementação da análise dos dados dos pacientes, uma vez que a análise dos dados de saúde condicionada a valores limites pode gerar um maior número de falsos positivos que outras abordagens, sobretudo aquelas que empregam Inteligência Artificial.

## 7.2 Trabalhos Futuros

Durante a fase de projeto, a avaliação de usabilidade do sistema foi planejada, onde os profissionais de saúde e estudantes do curso de Medicina da UFSCar poderiam utilizar o SBIdC-MPH para o acompanhamento de pacientes no Hospital Universitário da UFSCar. Infelizmente, a pandemia causada pelo COVID-19 postergou essa avaliação e, conseqüentemente, nosso próximo trabalho é realizá-la assim que retornarem as atividades presenciais na UFSCar.

Para a continuidade desse trabalho, está previsto a extensão da plataforma de sensores, para permitir a coleta de outros tipos de dados clínicos, e a adaptação e/ou extensão das aplicações do SBIdC-MPH para monitorar pacientes com outros tipos de DCNTs, tais como Diabetes, Asma e Obesidade. Também está previsto a integração de mais um componente à arquitetura do SBIdC-MPH, para permitir a coleta de dados ambientais relativos por exemplo à residência do paciente, os quais podem fornecer maiores subsídios para um diagnóstico mais assertivo.

Além de recomendar o monitoramento de pacientes com DCNTs, a Organização Mundial da Saúde também recomenda a prática de atividades físicas para a prevenção de tais doenças. No Brasil, o Sistema Único de Saúde (SUS) mantém um programa ligado as Unidades de Saúde da Família (USF), onde Grupos de Atividades Físicas (GAF), formados por moradores próximos as áreas das USF, realizam regularmente Sessões de Atividade Física (SAF) monitoradas pelos profissionais de saúde.

Em (QI et al., 2018), são relatados os resultados de uma revisão sistemática sobre o uso de tecnologias de IdC no reconhecimento e monitoramento de atividades físicas. Uma vez que os cenários de monitoramento de atividades físicas são bastante similares aos cenários de monitoramento de pacientes com DCNTs, o SBIdC-MPH também poderá ser adaptado para o monitoramento de tais atividades. Para tal, a plataforma de sensores precisa ser reconfigurada, por exemplo para permitir a captura da pressão arterial enquanto os participantes das GAFs estão praticando as atividades físicas, e também expandida para inclusão de outros sensores e desenvolvimento de algoritmos que permitam capturar dados relevantes para o monitoramento de tais atividades. Também é necessário adaptar/estender o SBIdC-MPH para manipular esses novos tipos de dados.

# REFERÊNCIAS

---

ADELSTEIN, Frank et al. **Fundamentals of mobile and pervasive computing**. New York: McGraw-Hill, 2005.

AFANEH, Mohammad. **Intro to Bluetooth low energy**. Novel Bits, LLC. 2018.

AHA. **Understanding Blood Pressure Readings**. American Heart Association, 2021. Disponível em: <<https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings>> Acesso em: 20 mar. 2021.

ALHARBE, Nawaf; ATKINS, Anthony S.; CHAMPION, Justin. Use of cloud computing with wireless sensor networks in an Internet of Things environment for a smart hospital network. In: **Proceedings of the Seventh International Conference on eHealth, Telemedicine, and Social Medicine, Lisbon, Portugal**. 2015. p. 22-27.

ALI, Mohammad; SADEGHI, Mohammad-Reza.; LIU, Ximeng. Lightweight Fine-Grained Access Control for Wireless Body Area Networks. **Sensors**, v. 20, n. 4, p. 1088, 17 fev. 2020.

AL-SIDDIQ, Waqaas. **Why IoT might prove to be the missing link in chronic disease management**. 2016. Disponível em: <<https://us.hitleaders.news/why-iot-might-prove-to-be-the-missing-link-in-chronic-disease-management/>>. Acesso em: 8 jan. 2022.

ARDUÍNO. **Getting Started with Arduino products**. 2020. Disponível em: <<https://www.arduino.cc/en/Guide>>. Acesso em: 01 jan. 2022.

ARREDONDO, A.; AVILES, R. Costs and Epidemiological Changes of Chronic Diseases: Implications and Challenges for Health Systems. **PLOS ONE**, v. 10, n. 3, p. e0118611, 17 mar. 2015.

ASHTON, Kevin et al. That 'internet of things' thing. **RFID journal**, v. 22, n. 7, p. 97-114, 2009.

BALAS, Valentina. E. **Healthcare paradigms in the internet of things ecosystem**. MA: Elsevier, 2020.

BARDRAM, Jakob E.; MIHAILIDIS, Alex.; WAN, Dadong. **Pervasive Computing in Healthcare**. CRC Press, 2007.

BARROSO, Weimar K. S. et al. Diretrizes Brasileiras de Hipertensão Arterial–2020. **Arquivos Brasileiros de Cardiologia**, v. 116, p. 516-658, 2021.

BECK, Kent. **Test-driven development: by example**. Boston: Addison-Wesley, 2002.

---

BHUIYAN, Mohammad N. et al. Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. **IEEE Internet of Things Journal**, v. 8, n. 13, p. 10474–10498, 1 jul. 2021.

BORMANN, Carsten; CASTELLANI, Angelo P.; SHELBY, Zach. Coap: An application protocol for billions of tiny internet nodes. **IEEE Internet Computing**, v. 16, n. 2, p. 62-67, 2012.

BOYE, Niels. Pervasive Healthcare: Problems and Potentials. In: KUSHNIRUK, A. W.; BORYCKI, E. M. (Eds.). Human, Social, and Organizational Aspects of Health Information Systems. **IGI Global**, 2008. p. 84–101.

BRASIL. Ministério da Saúde. **Hipertensão** (pressão alta). 2021. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/h/hipertensao-pressao-alta>> Acesso em: 15 jan. 2022.

BRESNICK, Jennifer. Healthcare Internet of Things Driving Global Market Growth. **Health Itanalytics**, 2015. Disponível em: <<https://healthitanalytics.com/news/healthcare-internet-of-things-driving-global-market-growth>>. Acesso em: 20 out. 2021.

BRITO, Gleison; VALENTE, Marco Tulio. Rest vs graphql: A controlled experiment. In: 2020 **IEEE International Conference on Software Architecture (ICSA)**. IEEE, 2020. p. 81-91.

BUNA, Samer. GraphQL in action. **Shelter Island**, NY: Manning Publications Co, 2021.

CASTANEDA, Denisse et al. A review on wearable photoplethysmography sensors and their potential future applications in health care. **International journal of biosensors & bioelectronics**, v. 4, n. 4, p. 195, 2018.

CHAI, Senchun. et al. **Wireless sensor networks**. Singapore: Springer, 2020.

CHAUDHURI, Abhik. Internet of things, for things, and by things. **Boca Raton**, FL: CRC Press/Taylor & Francis Group, 2019.

CHITRA, Lakshmi P.; SATAPATHY, Ravikanth. Performance comparison and evaluation of Node. js and traditional web server (IIS). In: 2017 **International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)**. IEEE, 2017. p. 1-4.

CISCO et al. **The Internet of Things Reference Model**. 2014. Disponível em: <[http://cdn.iotwf.com/resources/71/IoT\\_Reference\\_Model\\_White\\_Paper\\_June\\_4\\_2014.pdf](http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf)> Acesso em: 04 jan. 2022.

ELAHI, Ata.; GSCHWENDER, Adam. ZigBee wireless sensor and control network. **Upper Saddle River**, NJ: Prentice Hall, 2010.

ESPRESSIF. **ESP32 Technical Reference Manual**. 2021. Disponível em: <<https://www.espressif.com/en/products/socs/esp32>>. Acesso em: 30 dez. 2021.

---

FIELDING, Roy. T.; TAYLOR, Richard. N. Principled Design of the Modern Web Architecture. **ACM Transactions on Internet Technology (TOIT)**, v. 2, n. 2, p. 115-150, 2002.

GHOSHACHANDRA, Phukkhapan et al. oHealth: A self-care android application for senior citizens with hypertension. In: **2017 6th ICT International Student Project Conference (ICT-ISPC)**. IEEE, 2017. p. 1-5.

HALILI, Emily H. **Apache JMeter**. Birmingham: Packt Publishing, 2008.

HERMIDA, Ramón C. et al. Influence of measurement duration and frequency on ambulatory blood pressure monitoring. **Revista Española de Cardiología (English Edition)**, v. 60, n. 2, p. 131-138, 2007.

HEYDON, Robin. **Bluetooth low energy: the developer's handbook**. Upper Saddle River, NJ: Prentice Hall, 2012.

HILLAR, Gaston C. **MQTT Essentials-A lightweight IoT protocol**. Packt Publishing Ltd, 2017.

HUZOOREE, Geshwaree; KUMAR KHEDO, Kavi; JOONAS, Noorjehan. Pervasive mobile healthcare systems for chronic disease monitoring. **Health Informatics Journal**, v. 25, n. 2, p. 267–291, jun. 2019.

IQBAL, Muhammad Azhar et al. **Enabling the Internet of Things: Fundamentals, Design and Applications**. John Wiley & Sons, 2020.

JANDL JUNIOR, Peter. Computação, ubiquidade e transparência. **Revista de Ubiquidade**, v. 1, n. 1, p. 79-94, 2011.

JEON, Dong-cheol et al. Design of Hybrid Application Based on GraphQL for Efficient Query for PHR. In: **2019 International Conference on Information and Communication Technology Convergence (ICTC)**. IEEE, 2019. p. 381-383.

KHAN, Samiya; SHAKIL, Kashish Ara; ALAM, Mansaf. **Internet of Things (IoT): Concepts and Applications**. Springer, 2020.

KIM, Eungyeong et al. A hypertension management system with emergency monitoring. In: **2018 International Conference on Information Security and Assurance**. IEEE, 2018.

KIRTANA, R. N.; LOKESWARI, Y. V. An IoT based remote HRV monitoring system for hypertensive patients. In: **2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)**. IEEE, 2017. p. 1-6.

KJELDSEN, Sverre E. Hypertension and cardiovascular risk: general aspects. **Pharmacological research**, v. 129, p. 95-99, 2018.

KRISTOFFERSSON, Annica; LINDÉN, Maria. Wearable Sensors for Monitoring and Preventing Noncommunicable Diseases: A Systematic Review. **Information**, v. 11, n. 11, p. 521, 6 nov. 2020.

---

KUROSE, James. F.; ROSS, Keith. W. **Computer networking: a top-down approach**. 8. ed. Harlow: Pearson, 2021.

LEI, Kai; MA, Yining; TAN, Zhi. Performance comparison and evaluation of web development technologies in php, python, and node. js. In: **2014 IEEE 17th international conference on computational science and engineering**. IEEE, 2014. p. 661-668.

LOUREIRO, Antônio A. F. et al. Computação ubíqua ciente de contexto: Desafios e tendências. **27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, p. 99-149, 2009.

LYYTINEN, Kalle.; YOO, Youngjin. Issues and Challenges in Ubiquitous Computing. **Communications of the ACM**, v. 45, n. 12, p. 62–65, dez. 2002.

MALTA, Deborah C. et al. Progress with the strategic action plan for tackling chronic non-communicable diseases in Brazil, 2011-2015. **Epidemiologia e Serviços de Saúde**, v. 25, p. 373-390, 2016.

MASSÉ, Mark. **REST API design rulebook: designing consistent RESTful Web Service Interfaces**. Beijing Köln: O'Reilly, 2012.

MCKENNEY, James. M. Patient Education and Compliance: How to Make It Cost-Effective. **Value in Health**, v. 1, n. 4, p. 212–215, nov. 1998.

MEHTA, Rishika.; SAHNI, Jyoti.; KHANNA, Kavita. Internet of Things: Vision, Applications and Challenges. **Procedia Computer Science**, v. 132, p. 1263–1269, 2018.

MUKHIYA, Suresh. K. et al. A GraphQL approach to Healthcare Information Exchange with HL7 FHIR. **Procedia Computer Science**, v. 160, p. 338–345, 2019.

NIKOLOV, Neven. Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems. In: **2020 XXIX International Scientific Conference Electronics (ET)**. IEEE, 2020.

OASIS. MQTT Version 5.0. 2019. Disponível em: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf> Acesso em: 22 fev. 2022.

OHRTMAN, Frank.; ROEDER, Konrad. **Wi-Fi handbook: building 802.11b wireless networks**. New York: McGraw-Hill, 2003.

PATIL, Charushila; CHAWARE, Anita. Heart (Pulse Rate) Monitoring using Pulse Rate Sensor, Piezo Electric Sensor and NodeMCU. In: **2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)**. IEEE, 2021. p. 337-340.

PEIRIS, Pradeep. **How IoT Strengthens Ubiquitous Computing**. DZone, 2017. Disponível em: <https://dzone.com/articles/how-iot-strengthens-ubiquitous-computing>. Acesso em: 28 nov. 2021.

---

POSLAD, Stefan. **Ubiquitous computing: smart devices, environments and interactions**. Chichester, U.K: Wiley, 2009.

PUTRA, Rizky D.; WIBISONO, Gunawan. System Design and Implementation of Machine-to-Machine (M2M) for Hypertension Patients. In: **2017 IEEE 85th Vehicular Technology Conference (VTC Spring)**. IEEE, 2017. p. 1-5.

QADRI, Yazdan. A. et al. The Future of Healthcare Internet of Things: A Survey of Emerging Technologies. **IEEE Communications Surveys & Tutorials**, v. 22, n. 2, p. 1121–1167, 2020.

QI, Jun et al. Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review. **Journal of biomedical informatics**, v. 87, p. 138-153, 2018.

RASPBERRY PI. **Getting Started**. Disponível em: <<https://www.raspberrypi.com/documentation/computers/getting-started.html>>. Acesso em: 30 dez. 2021.

RODRIGUES, Ronitti J. S.; MENDES, Ivo P.; SOUZA, Wanderley L. MyHealth: a system for monitoring non-communicable diseases. In: **17th International Conference on Information Technology–New Generations (ITNG 2020)**. Springer, Cham, 2020. p. 439-444.

SANDI, Gitarja; NUGRAHA, I. Gusti Bagus Baskara; SUPANGKAT, Suhono Harso. Mobile health monitoring and consultation to support hypertension treatment. In: **International conference on ICT for smart society**. IEEE, 2013. p. 1-5.

SANGWAN, Aarti.; BHATTACHARYA, Partha P. Wireless Body Sensor Networks: A Review. **International Journal of Hybrid Information Technology**, v. 8, n. 9, p. 105–120, 30 set. 2015.

SANTOS, Bruno. P. et al. Internet das Coisas: da Teoria à Prática. **Minicursos Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. 2016.

SHELBY, Z.; HARTKE, K.; BORMANN, C. **The Constrained Application Protocol (CoAP)**. [s.l.] RFC Editor, jun. 2014. Disponível em: <<https://www.rfc-editor.org/info/rfc7252>>. Acesso em: 2 jan. 2022.

SOOD, Sandeep K.; MAHAJAN, Isha. IoT-fog-based healthcare framework to identify and control hypertension attack. **IEEE Internet of Things Journal**, v. 6, n. 2, p. 1920-1927, 2018.

STOJANOVIC, M. Is 30 Minutes Interval Enough for Analyzing Ambulatory Blood Pressure Monitoring?. **Journal of Hypertension**, v. 36, p. e71, 2018.

TANENBAUM, Andrew. S.; FEAMSTER, Nickolas.; WETHERALL, David. **Computer networks**. 6ª ed. Harlow, United Kingdom: Pearson, 2021.

TAVERA, Carlos. A. et al. Wearable Wireless Body Area Networks for Medical Applications. **Computational and Mathematical Methods in Medicine**, v. 2021, p. 1–9, 24 abr. 2021.

---

VARSHNEY, Upkar. **Pervasive Healthcare Computing**. Boston, MA: Springer US, 2009.

VARSHNEY, Upkar. Pervasive Healthcare: Applications, Challenges And Wireless Solutions. **Communications of the Association for Information Systems**, v. 16, 2005.

WEBER, Rolf. H.; WEBER, Romana. **Internet of Things**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

WEISER, Mark. The Computer for the 21 st Century. **Scientific American**, v. 265, n. 3, p. 94-105, 1991.

WHELTON, Paul. K. et al. 2017  
ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults: A Report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. **Hypertension**, v. 71, n. 6, jun. 2018.

WHO et al. **A global brief on hypertension: silent killer, global public health crisis: World Health Day 2013**. World Health Organization, 2013.

WHO et al. **Global NCD target: reduce high blood pressure**. World Health Organization, 2016. Disponível em: < <https://www.who.int/beat-ncds/take-action/policy-brief-reduce-high-blood-pressure.pdf> > Acesso em: 20 out. 2021.

WHO et al. **Noncommunicable diseases country profiles 2018**. World Health Organization. 2018. Disponível em: < <https://apps.who.int/iris/handle/10665/274512>> Acesso em: 20 out. 2021.

WHO. **Noncommunicable diseases progress monitor**. Geneva. World Health Organization. 2020. Disponível em: <<https://www.who.int/publications/i/item/9789240000490>> Acesso em: 22 out. 2021.

WHO. **Noncommunicable Diseases**. World Health Organization, 13 de abr. 2021. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases>>. Acesso em: 20 out. 2021

WILDE, Erik.; PAUTASSO, Cesare. **REST: From Research to Practice**. New York, NY: Springer New York, 2011.

XIAOJIANG, Xing; JIANLI, Wang; MINGDONG, Li. Services and key technologies of the Internet of Things. **Zte Communications**, v. 8, n. 2, p. 26-29, 2020.

XU, Shaochun; LI, Tong. Evaluation of test-driven development: An academic case study. In: **Software Engineering Research, Management and Applications 2009**. Springer, Berlin, Heidelberg, 2009. p. 229-238.

YANG, Kunpeng et al. Design of Remote Control Inverter Based on MQTT Communication Protocol. In: **2021 IEEE International Conference on Mechatronics and Automation (ICMA)**. IEEE, 2021. p. 1374-1378.

---

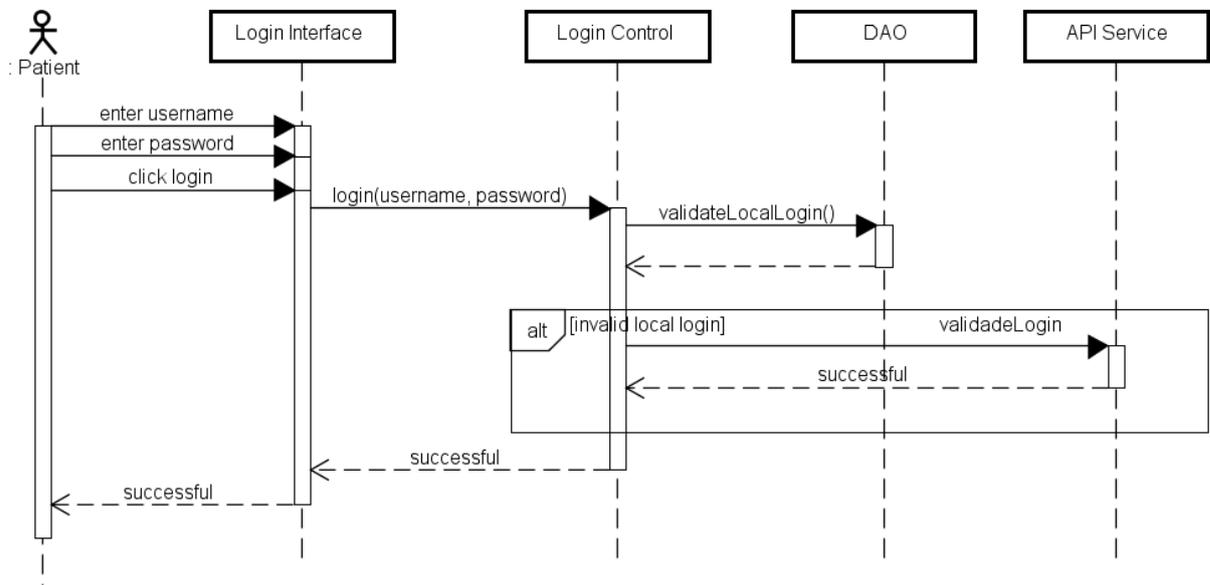
YOKOTANI, Tetsuya; SASAKI, Yuya. Comparison with HTTP and MQTT on required network resources for IoT. In: **2016 international conference on control, electronics, renewable energy and communications (ICCEREC)**. IEEE, 2016.

ZEPHYR. **Bluetooth Stack Architecture**. 2022. Disponível em: <<https://docs.zephyrproject.org/latest/guides/bluetooth/bluetooth-arch.html>> Acesso em: 13 jan. 2022.

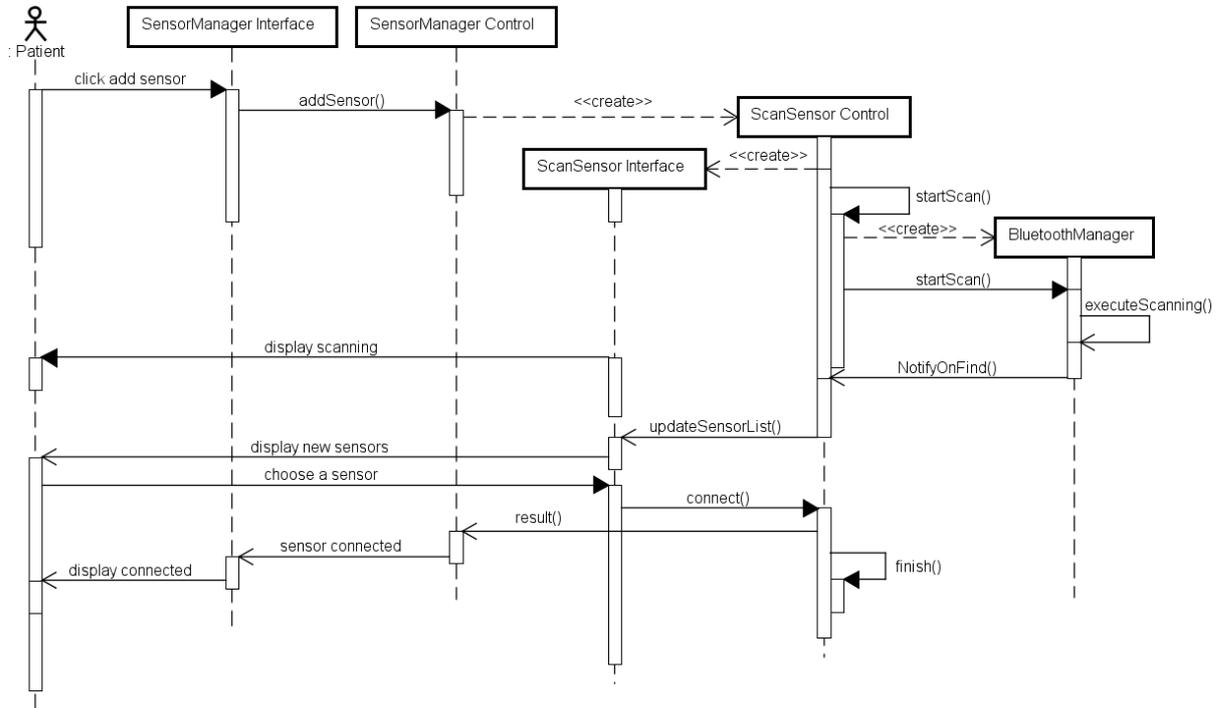
# Apêndice A

## DIAGRAMAS DE SEQUÊNCIA PARA MOBILE APP

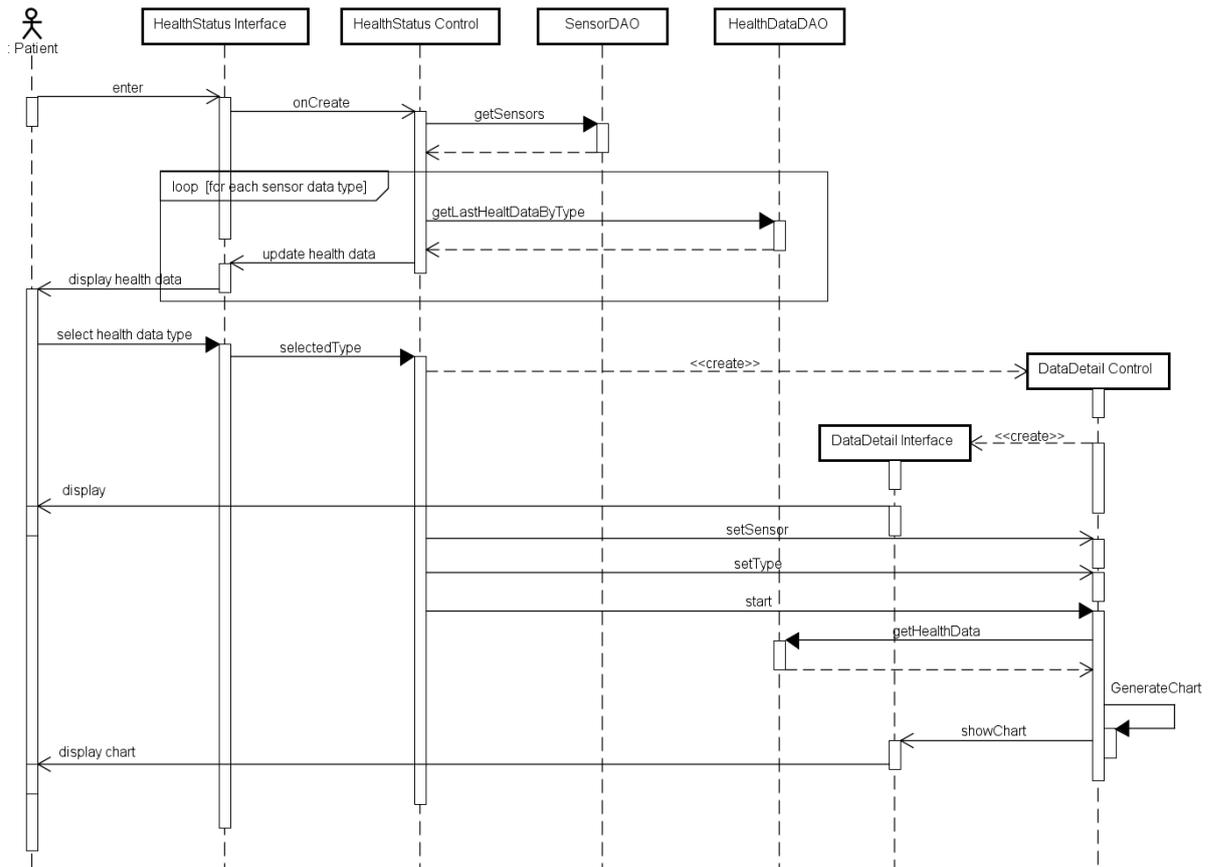
### 1. Funcionalidade Authenticate



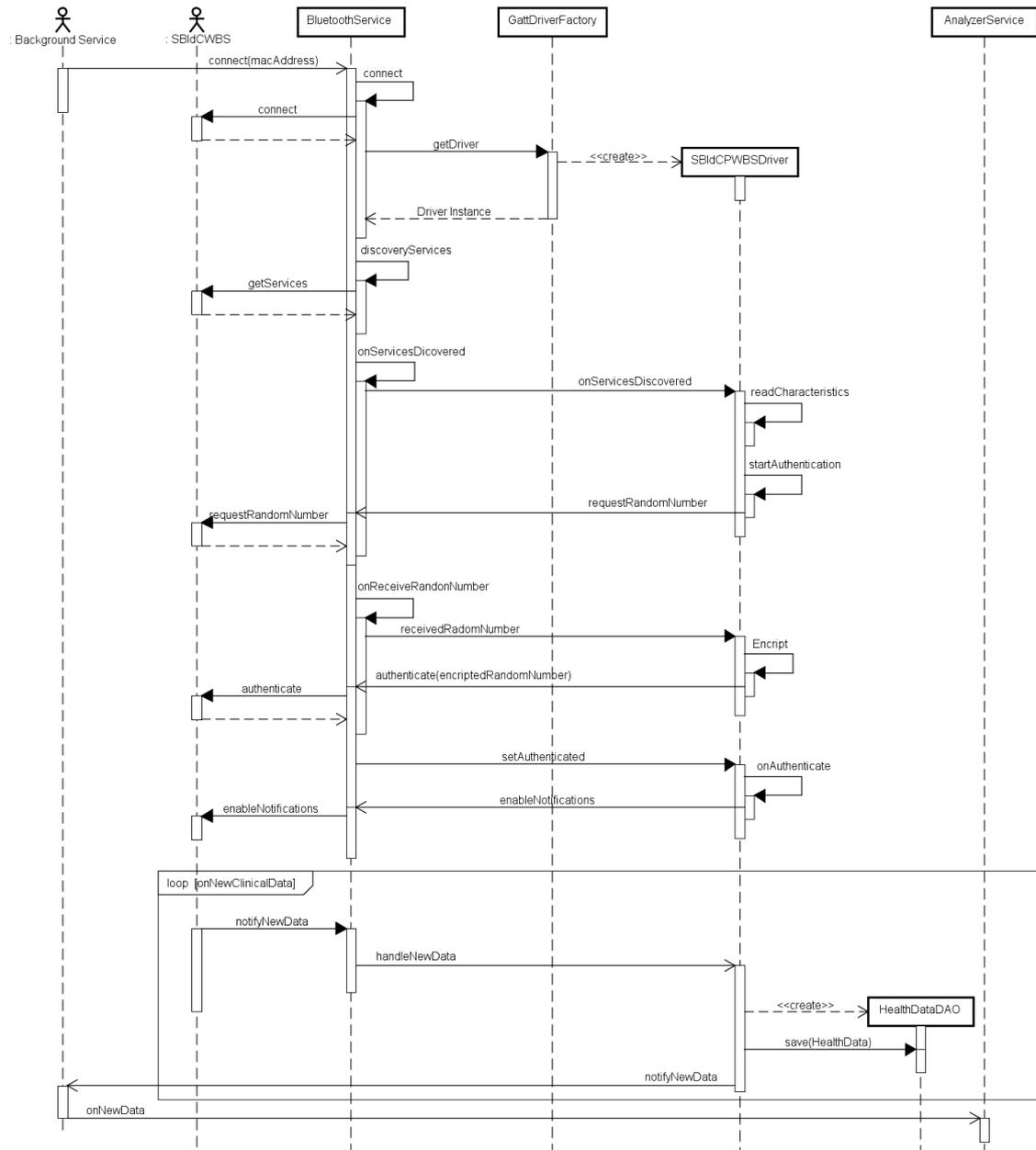
## 2. Funcionalidade Manage Sensors



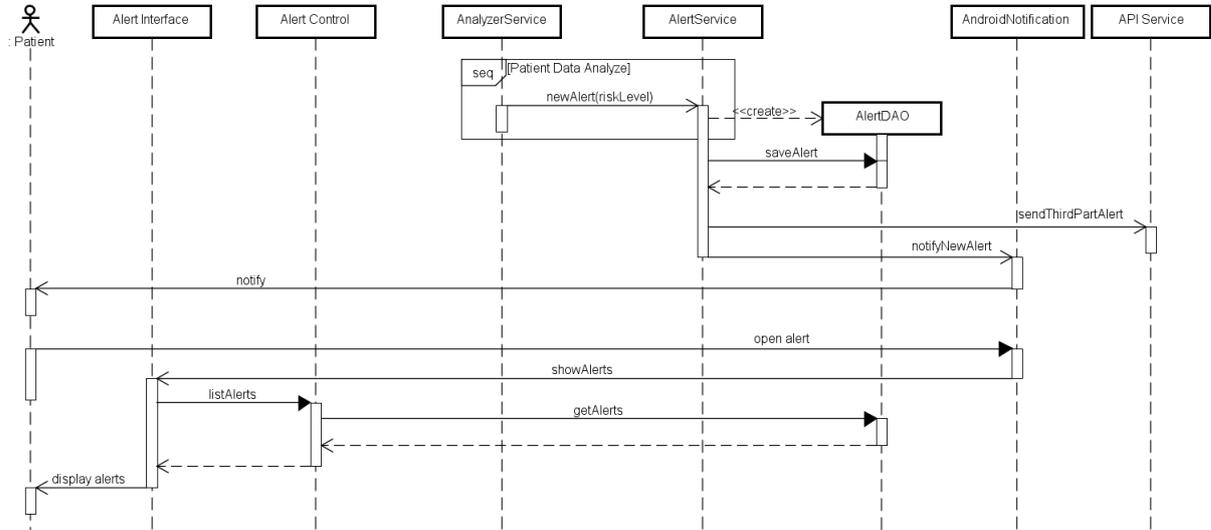
### 3. Funcionalidade View Data



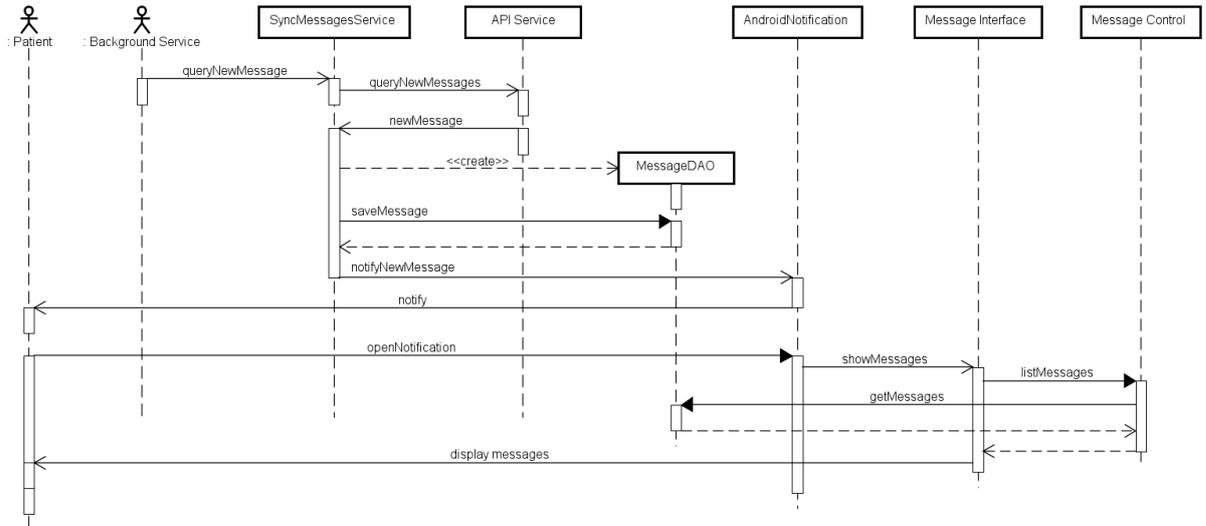
## 4. Funcionalidade Physiological Data Collection



## 5. Funcionalidade Manage Alerts

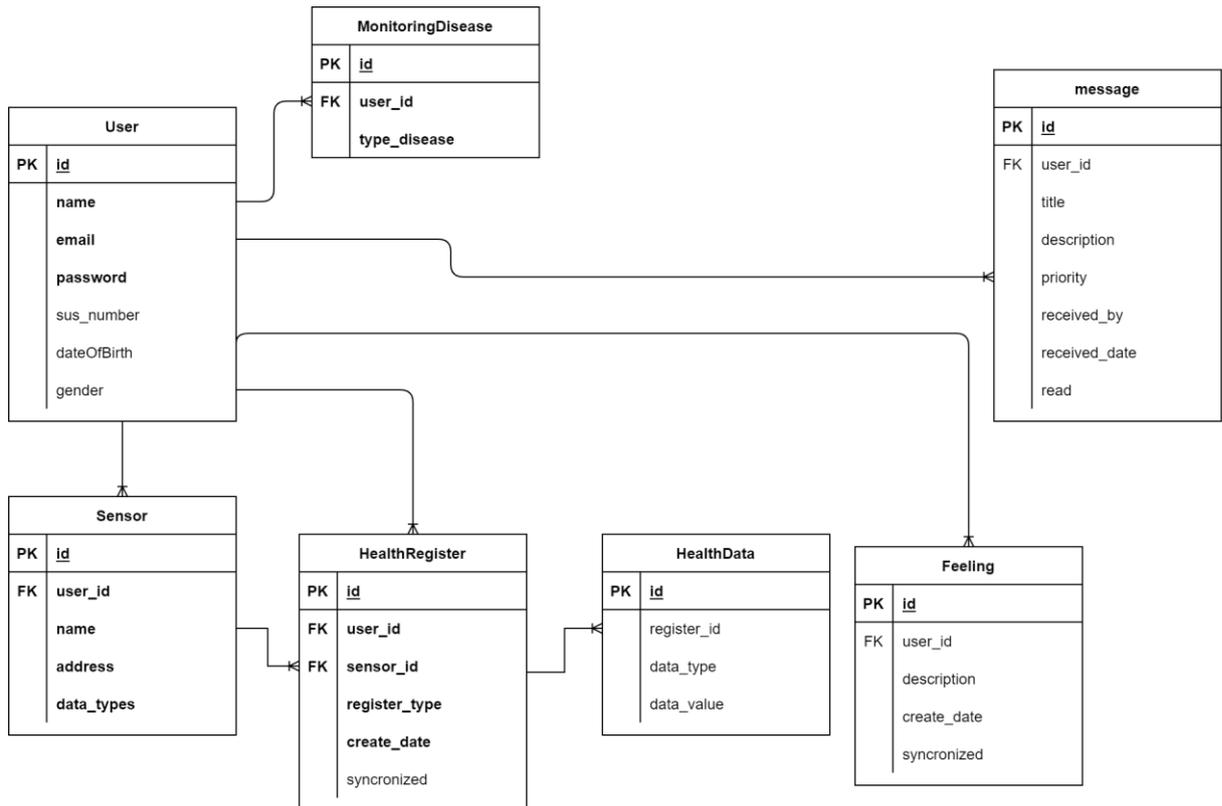


## 6. Funcionalidade Receive Messages



# APÊNDICE B

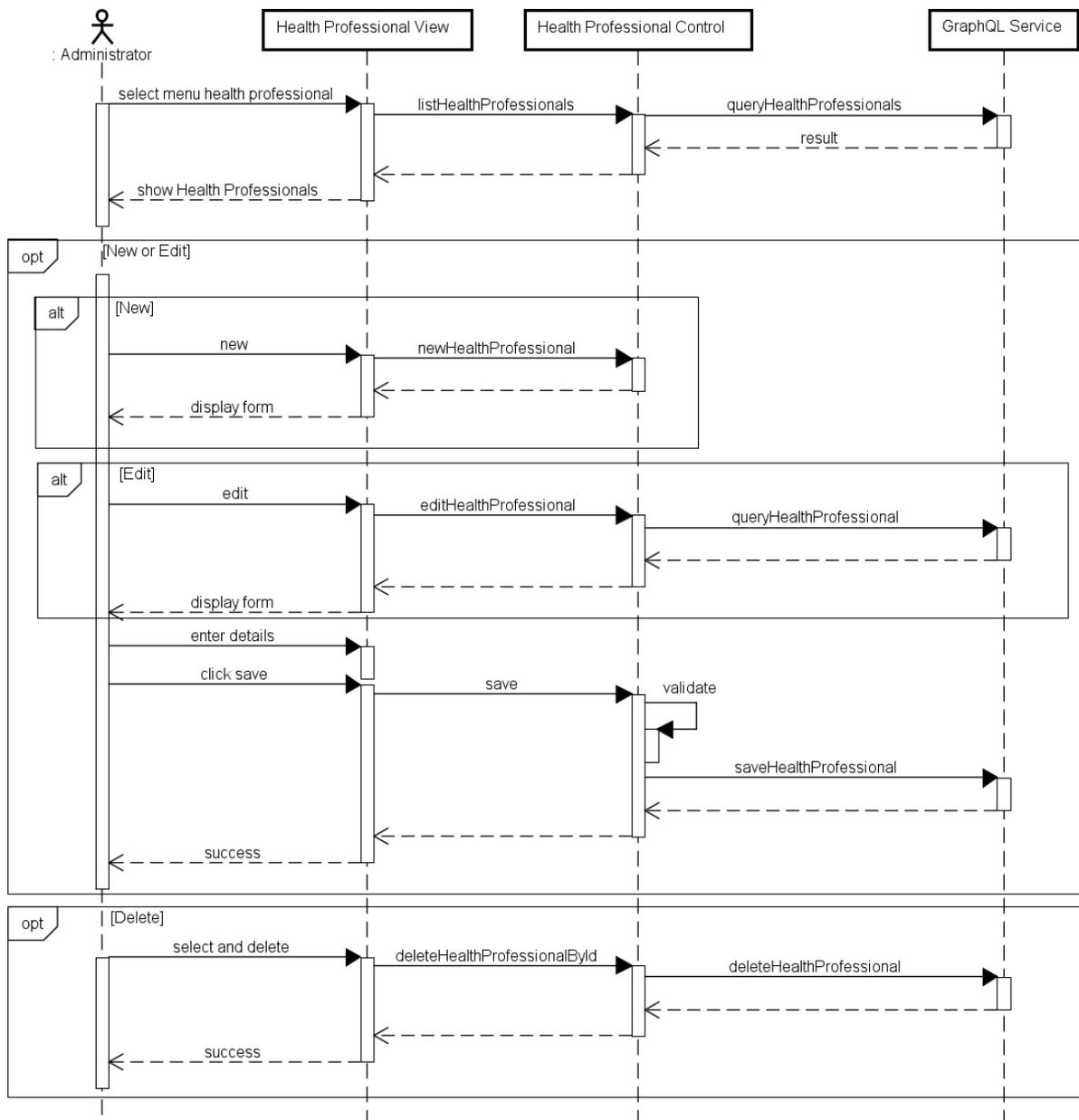
## DIAGRAMA ENTIDADE RELACIONAMENTO DA MOBILE APP



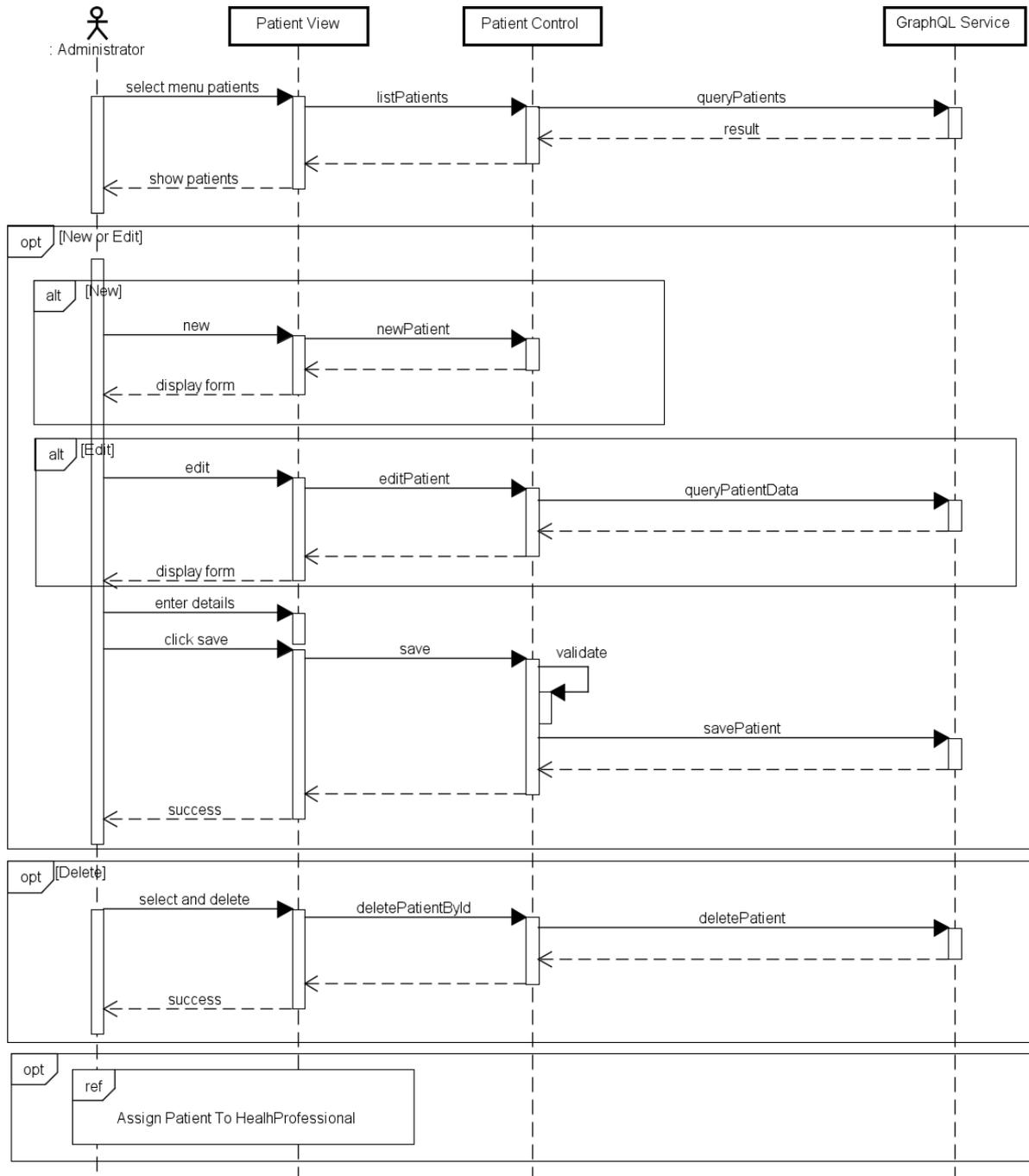
# Apêndice C

## DIAGRAMAS DE SEQUÊNCIA DA WEB APP

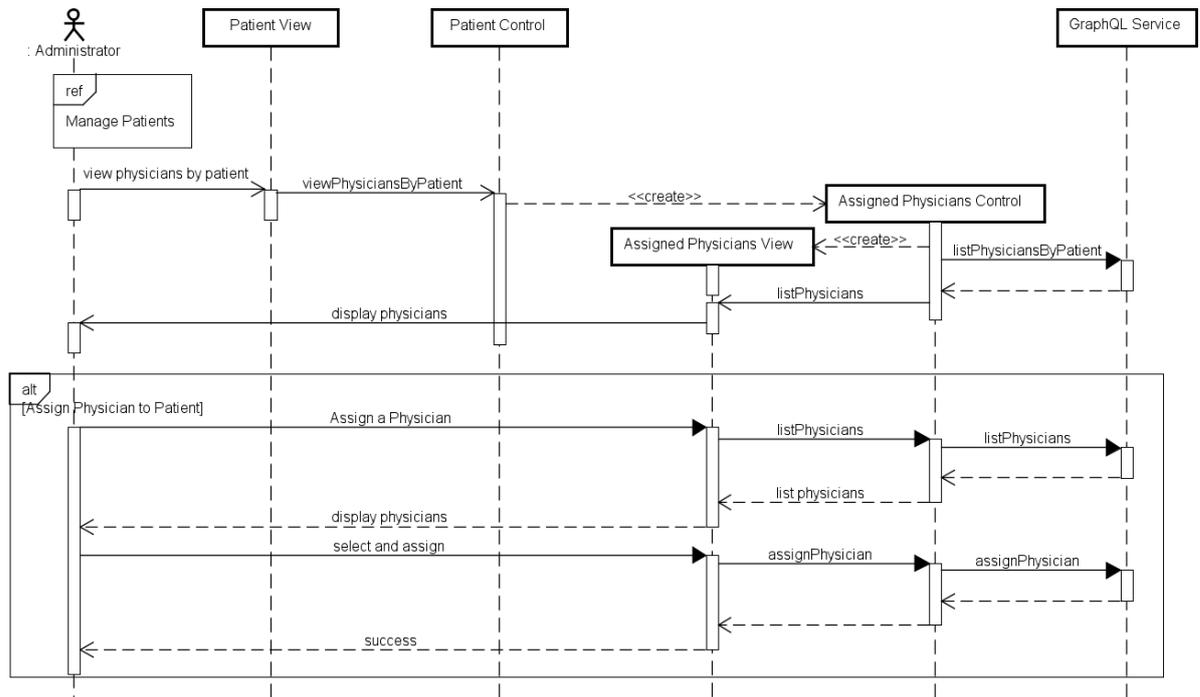
### 1. Funcionalidade Manage Health Professionals



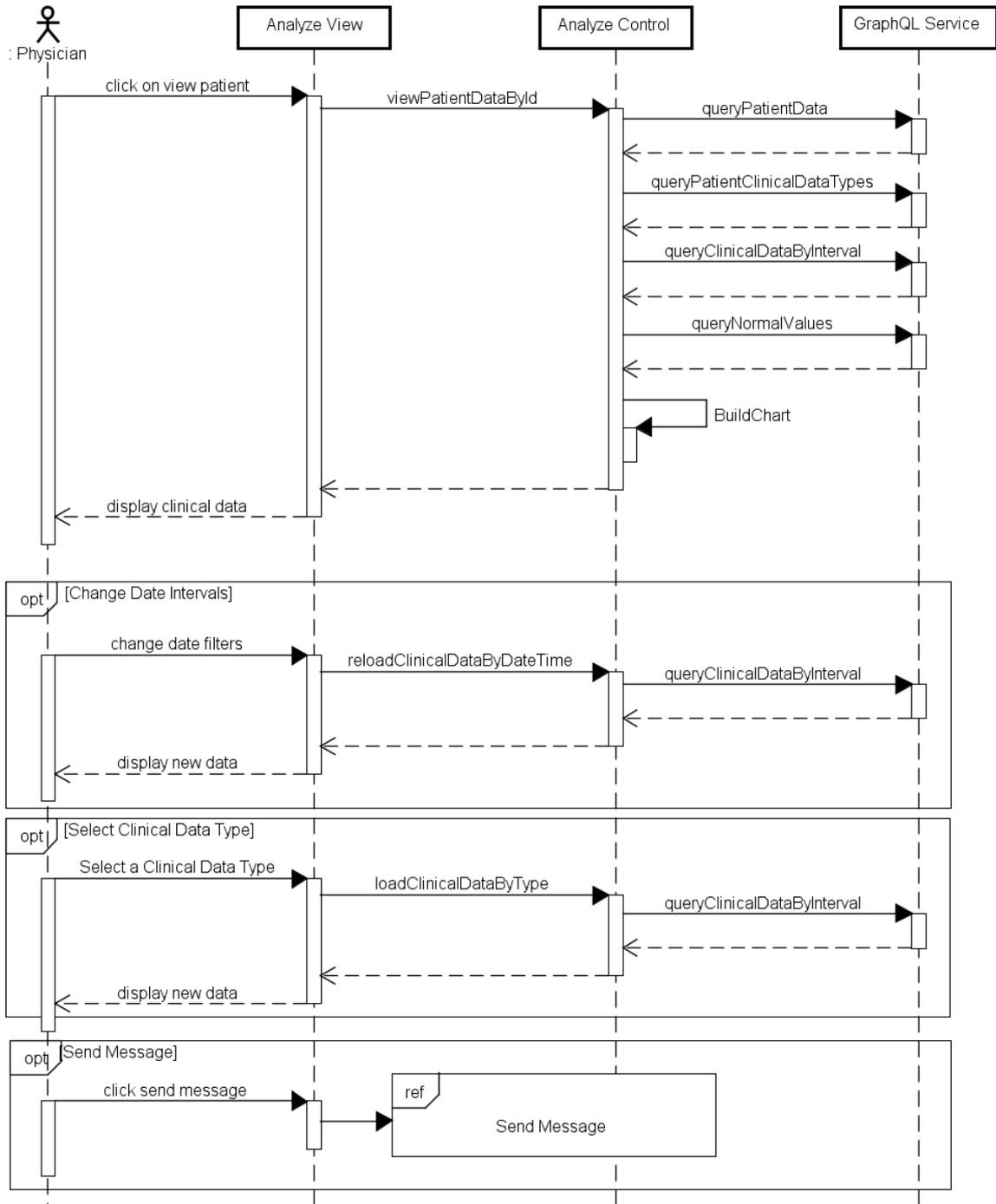
## 2. Funcionalidade Manage Patients



### 3. Funcionalidade Assign Patient to Health Professional



## 4. Funcionalidade Analyze Health Data



## 5. Funcionalidade Send Message

