



Caroline Stéphanie Cabral Silva

Desenvolvimento de Sistema Computacional para Processamento e Transmissão de Dados Eletrofisiológicos Utilizando *IoT*

Macaíba

2017



INSTITUTO INTERNACIONAL
DE NEUROCIÊNCIAS
Edmond e Lily Safra

INSTITUTO INTERNACIONAL
DE NEUROCIÊNCIAS
EDMOND E LILY SAFRA

Programa de Pós-Graduação em Neuroengenharia

Caroline Stéphanie Cabral Silva

**Desenvolvimento de Sistema Computacional para
Processamento e Transmissão de Dados Eletrofisiológicos
Utilizando *IoT***

Dissertação apresentada ao Programa de Pós-Graduação em Neuroengenharia do Instituto Internacional de Neurociências - Edmond e Lily Safra como parte dos requisitos exigidos para a obtenção do título de Mestre em Neuroengenharia.

Orientador: Prof. Dr. Renan Cipriano Moioli

Co-orientadora Prof. Dr. Cicilia Raquel Maia Leite

Macaíba

2017

Caroline Stéphanie Cabral Silva

Desenvolvimento de Sistema Computacional para Processamento e Transmissão de Dados Eletrofisiológicos Utilizando *IoT*/ Caroline Stéphanie Cabral Silva.

– Macaíba, 2017-

96 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Renan Cipriano Moioli

Dissertação (Mestrado) – INSTITUTO INTERNACIONAL DE NEUROCIÊNCIAS EDMOND E LILY SAFRA

Programa de Pós-Graduação em Neuroengenharia, 2017.

1. *IoT*. 2. *Android*. I. Dr. Renan Cipriano Moioli. II. Instituto Internacional de Neurociências Edmond e Lily Safra. III. Programa de Pós-Graduação em Neuroengenharia. IV. Desenvolvimento de Sistema computacional para Processamento e Transmissão de Dados Eletrofisiológicos Utilizando *IoHT*

CDU 02:141:005.7

Caroline Stéphanie Cabral Silva

Desenvolvimento de Sistema Computacional para Processamento e Transmissão de Dados Eletrofisiológicos Utilizando *IoT*

Dissertação apresentada ao Programa de Pós-Graduação em Neuroengenharia do Instituto Internacional de Neurociências - Edmond e Lily Safra como parte dos requisitos exigidos para a obtenção do título de Mestre em Neuroengenharia.

Trabalho aprovado. Macaíba, 31 de Março de 2017:

Prof. Dr. Renan Cipriano Moioli
Orientador

Prof. Dr. Augusto José Venâncio Neto
Convidado 1

Prof. Dr. Edgard Morya
Convidado 2

Macaíba

2017

Resumo

O trabalho descreve o desenvolvimento de um sistema computacional para processamento e transmissão de dados eletrofisiológicos utilizando internet das coisas, para monitoramento remoto de pacientes, promovendo o aumento da qualidade de vida e os cuidados com a saúde. O sistema propõe um meio alternativo para acompanhamento de tratamentos devido ao alto índice de mortes e condições limitantes derivadas de desordens neurológicas. Foram utilizadas a abordagem de programação orientada a objetos e a linguagem de modelagem unificada para o planejamento, a linguagem de programação *Java* e o ambiente de desenvolvimento integrado *Android Studio* para a plataforma *Android* em dispositivos móveis, serviços do *Firebase* como o "Authentication" para cadastro do usuário, o "Storage" para armazenamento dos dados e o "Realtime Database" para interação de mensagens entre usuários relacionados. Foram implementadas a simulação de aquisição de um conjunto de dados artificiais (gerados pelo sistema), a visualização do registro dos dados em gráfico em tempo real e a transmissão dos dados para os usuários.

Palavras-chaves: *Internet das Coisas; Android; cuidados com a saúde (homecare); saúde móvel (mhealth); sinais eletrofisiológicos.*

Abstract

This work describes a development of a computational system for the processing and transmission of electrophysiological data using internet of things (IoT), for remote monitoring (homecare) of patients, promoting the increase of quality of life and healthcare. The system proposes an alternative way for monitoring treatments due to the high death rate and limiting conditions derived from neurological disorders. It were used an object-oriented programming (OOP) approach and unified modeling language (UML) for planning, Java programming language, and the integrated development environment (IDE) Android Studio for the Android platform on mobile devices, Firebase services such as "Authentication" for user login, "Storage" for data storage and "Realtime Database" for message interaction between related users. A simulation of the acquisition of a set of artificial data (data generated by the system), the visualization of the data recorded in real time plot and the transmission of the data to the users were developed.

Keywords: *IoT; Android; homecare; mHealth; electrophysiological signals.*

Sumário

1	Introdução	1
1.1	Objetivo Geral	9
1.2	Objetivos Específicos	9
1.3	Estrutura do Trabalho	9
2	Embasamento Teórico	11
2.1	Sinais Eletrofisiológicos e Desordens Neurológicas	11
2.2	Interface Cérebro-Máquina (ICM)	13
2.3	<i>Homecare, e-Care, e-Saúde, Distributed Healthcare, Telemedicina e Saúde Móvel (mHealth)</i>	15
2.4	Computação Pervasiva Ubíqua	21
2.5	Internet das Coisas	21
2.5.1	Dispositivo para Dispositivo (<i>Device-to-Device</i>)	22
2.5.2	Dispositivo para Nuvem (<i>Device-to-Cloud</i>)	22
2.5.3	Dispositivo para <i>Gateway</i> (<i>Device-to-Gateway</i>)	23
2.6	<i>Data Warehouse, Business Intelligence e Big Data</i>	24
2.7	Programação Orientada a Objetos e Linguagem de Modelagem Unificada	26
3	Tecnologias	31
3.1	Java	31
3.2	Android	32
3.3	Android Studio	34
3.4	Firebase	36
4	Desenvolvimento do <i>Spike Scan</i>	39
4.1	Análise (<i>Object-Oriented Analysis</i>)	40
4.1.1	Descrição Geral do Sistema	40
4.1.1.1	Funcionalidades do Sistema	40
4.1.1.2	Recursos para os Usuários	41
4.1.1.3	Aplicações do Sistema	41
4.1.2	Definição de Requisitos	42
4.1.3	Modelo Conceitual de Objetos	43
4.1.4	Tabelas de Casos de Uso	45
4.1.4.1	Monitorar Sinal Online	45
4.1.4.2	Gravar Sinal Online	46
4.1.4.3	Seleção de Sinal Parcial	46

4.1.4.4	Processamento de Sinal <i>offline</i>	47
4.1.4.5	Pré-Processamento de Sinal <i>online</i>	47
4.1.4.6	Análise de Sinal <i>offline</i>	48
4.1.5	Diagrama de Casos de Uso	49
4.1.6	Diagrama de Classes	50
4.2	Design (<i>Object-Oriented Design</i>)	52
4.2.1	Ferramentas Específicas	52
4.2.2	Arquitetura do Sinal	53
4.2.3	<i>Templates</i>	56
4.2.3.1	<i>Permissions</i> e <i>Login</i>	56
4.2.3.2	<i>Home</i> e <i>Personal Report</i>	57
4.2.3.3	<i>Menu</i> e <i>Devices</i>	58
4.2.3.4	Sinal e Análise	58
4.2.3.5	Emergência	59
4.2.3.6	Mensagens e Pesquisador	60
4.2.4	Fluxo de Telas	61
4.3	Implementação (<i>Object-Oriented Programming</i>)	63
4.3.1	Conexão do Android Studio com o Firebase	63
4.3.2	Autenticação do Usuário	66
4.3.3	Seleção de Pacientes	69
4.3.4	Cadastro do Paciente	70
4.3.5	Cadastro do Profissional de Saúde	71
4.3.6	Implementação do Menu Principal	72
4.3.7	Seleção de Registros	73
4.3.8	Interação entre Usuários	74
4.3.9	Seleção de Canais	75
4.3.10	Visualização do Sinal em Tempo Real	76
4.3.11	Gravação dos Sinais	77
4.3.12	Envio da Gravação por Email	78
4.3.13	Visualização do arquivo enviado	79
4.4	Validação do Sinal	81
Conclusão	83
Referências	85
ANEXO A Manuscrito	89

“Learning never exhausts the mind.” Leonardo da Vinci

Listas de ilustrações

Figura 1.1 – Cálculo da quantidade de mortes atribuídas a desordens neurológicas.	1
Figura 1.2 – Tempo em anos de desenvolvimento clínico para um fármaco iniciar no mercado.	3
Figura 1.3 – Custo do desenvolvimento clínico (em milhões)	3
Figura 1.4 – Porcentagem de aprovação clínica dos testes dos fármacos para o SNC.	3
Figura 1.5 – Visualização do aumento do tráfego de dados em <i>bits</i> por segundo em décadas.	5
Figura 1.6 – Visualização do tráfego de dados em <i>bits</i> por segundo diariamente em algumas regiões do Brasil.	6
Figura 2.1 – Diferentes métodos de gravação extracelular ((a) formas de onda relativas a sinais de <i>LFP</i> (invasivo), ECoG (semi-invasivo) e EEG (não-invasivo); (b) EEG (vermelho), <i>LFPs</i> (azul) e potenciais de ação neuronais em diferentes áreas do cérebro - coordenada Cz, área motora suplementar (SM), cortex entorrinal (EC), hipocampo (HC) e amidala (Am); (c) gravação simultânea de MEG (preto) e EEG (vermelho) (d) <i>LFPs</i> e potenciais de ação em profundidades diferentes do córtex motor de um gato.	12
Figura 2.2 – Fases de um potencial de ação: membrana em repouso (equilíbrio dos íons de sódio e potássio entre a membrana), despolarização (após estímulo, o limiar de excitação é ultrapassado), repolarização (período de retorno lentamente da membrana para o limiar) e período refratário (reestabelecimento do equilíbrio dos íons de sódio e potássio).	13
Figura 2.3 – Exemplo de Interface Cérebro-Máquina envolvendo um primata não-humano em uma tarefa com imaginação motora.	14
Figura 2.4 – Impacto na implantação de sistemas	17
Figura 2.5 – Percentual sobre o total de médicos com acesso a computador no estabelecimento de saúde	18
Figura 2.6 – Percentual sobre o total de estabelecimentos que utilizaram internet nos últimos 12 meses	19
Figura 2.7 – Proporção de estabelecimentos de saúde que possuem departamento ou área de tecnologia de informação (2015)	20
Figura 2.8 – Padrão de comunicação entre dispositivos <i>Device-to-Device</i> estabelecendo uma comunicação utilizando tecnologia <i>bluetooth</i>	22

Figura 2.9 – Padrão de comunicação entre dispositivos <i>Device-to-Device</i> estabelecendo uma comunicação entre sensores de diferentes fabricantes.	22
Figura 2.10–Padrão de comunicação entre dispositivo e a nuvem <i>Device-to-Cloud</i>	23
Figura 2.11–Padrão de comunicação entre dispositivo e o <i>Gateway</i> (<i>Device-to-Gateway</i>). .	23
Figura 2.12–Padrão de comunicação entre dispositivo e o <i>Gateway</i> (<i>Device-to-Gateway</i>). .	24
Figura 2.13–Sistema de Suporte à Decisão	25
Figura 2.14–BigData Environment	27
Figura 3.1 – Linguagens de Programação mais Utilizadas	31
Figura 3.2 – Estrutura <i>Android</i> com as camadas: <i>Linux Kernel</i> , <i>ART</i> , <i>Libraries</i> , <i>Application Framework</i> e <i>Applications</i>	33
Figura 3.3 – Compilação do <i>Android</i>	34
Figura 3.4 – Diferentes Telas e Densidades <i>Android</i>	35
Figura 3.5 – Recursos do <i>Firebase</i>	37
Figura 3.6 – Modelo request-response	37
Figura 3.7 – Modelo utilizando <i>Firebase</i>	38
Figura 3.8 – Envio de informação para o <i>Firebase</i>	38
Figura 3.9 – Atualização das informações automáticas pelo <i>Firebase</i>	38
Figura 4.1 – Fases gerais do sistema: <i>Object-Oriented Analysis</i> , <i>Object-Oriented Design</i> , <i>Object-Oriented Programming</i> e suas etapas respectivas.	39
Figura 4.2 – Modelo conceitual de objetos <i>Spike Scan</i>	44
Figura 4.3 – Modelo conceitual de objetos relacionados <i>Spike Scan</i>	45
Figura 4.4 – Diagrama de casos de uso.	49
Figura 4.5 – Diagrama de classes do sistema <i>Spike Scan</i> : Em cor lilás estão as classes referentes aos usuários do sistema; em cor laranja as observações relacionadas as classes ligadas; em cor amarela as classes relacionadas aos dados; em cor cinza as classes que possivelmente utilizarão os dados para gerar relatórios; e em verde a classe com os atributos e métodos comuns aos usuários.	51
Figura 4.6 – Versões da <i>API</i> mais utilizadas pelos usuários. <i>Ice Cream Sandwich</i> com 97.4%.	53
Figura 4.7 – Fases da transmissão do sinal eletrofisiológico no sistema: aquisição (dado de um dispositivo externo ou de um arquivo), pré-processamento (monitoramento <i>online</i> e seleção de atributos, processamento (análise do dado) e resposta (<i>feedback</i>) aos usuários).	55
Figura 4.8 – Tela Permissions e Login	56
Figura 4.9 – Tela <i>Home</i> e <i>Personal Report</i>	57
Figura 4.10–Tela Menu e Devices.	58

Figura 4.11–Tela Sinal e Análise	59
Figura 4.12–Tela Solicitação de Emergência	60
Figura 4.13–Tela Mensagens	61
Figura 4.14–Fluxo da relação entre as telas do sistema <i>Spike Scan</i>	62
Figura 4.15–Criar Projeto Firebase	63
Figura 4.16–Selecionar Android	64
Figura 4.17–Inserir Pacote e SHA-1	64
Figura 4.18–Localização do SHA-1	65
Figura 4.19–Salvar Arquivo JSON	65
Figura 4.20–Adicionar Dependência	66
Figura 4.21–Adicionar Plugin	66
Figura 4.22–Adicionar Plugin	66
Figura 4.23–Telas de autenticação do usuário: <i>login</i> e verificação de autorização prévia dos dados de acesso (usuário e senha).	67
Figura 4.24–Telas de autenticação do usuário: inserção de <i>email</i> ou telefone e preenchimento do email como exemplo.	68
Figura 4.25–Telas autenticação do usuário:	69
Figura 4.26–Lista de pacientes vinculados ao usuário profissional da saúde	70
Figura 4.27–Cadastro de Paciente	71
Figura 4.28–Edição de cadastro de profissional da saúde	72
Figura 4.29–Menu Principal: <i>Home, Devices, Signal, Analysis, Messages, Emergency, MyAccount, Settings</i>	73
Figura 4.30–Lista de Gravações Arquivadas	74
Figura 4.31–Exemplo de troca de mensagens entre usuários conectados no sistema vinculados ao acompanhamento remoto de paciente.	75
Figura 4.32–Seleção dos canais desejados pelo usuário referentes ao sistema internacional 10-20.	76
Figura 4.33–Visualização dos sinais artificiais nos canais selecionados.	77
Figura 4.34–Gravação dos Sinais por Visualização de Canais	78
Figura 4.35–Envio da Gravação dos dados por Email	79
Figura 4.36–Visualização do arquivo com extensão ".csv" com os dados artificiais gravados e enviados pelo usuário.	80
Figura 4.37–Validação do sinal com os dados artificiais gravados e enviados pelo usuário. .	82

Lista de tabelas

Tabela 2.1 – Aplicativos e suas características: S.O, desenvolvedor, nome, licença e dispositivos suportados.	15
Tabela 4.1 – Monitorar sinal online	46
Tabela 4.2 – Gravar sinal online	46
Tabela 4.3 – Seleção de Sinal Parcial	47
Tabela 4.4 – Processamento de sinal offline	47
Tabela 4.5 – Pré-Processamento de Sinal online	48
Tabela 4.6 – Análise de Sinal offline	48

1 Introdução

As desordens neurológicas representam uma parcela notável nos custos em sistemas hospitalares, baixa qualidade de vida e dependência de indivíduos. Estima-se que 1/3 da população sofra com alguma condição crônica, e que 15% dessas condições gere 70% dos custos no sistema de saúde (WHO, 2016). Esses pacientes precisam enfrentar a doença por um longo período de tempo, podendo perdurar anos de tratamento. Assim, a carga causada pela doença é intensa e gera dificuldades ao lidar com as limitações, além dos alto índice de mortalidade.

A Organização Mundial da Saúde (OMS), do inglês *World Health Organization (WHO)*, conduziu um estudo global que mostrou que 8 dos 10 distúrbios nas 3 classes mais elevadas de deficiência são ocasionados por desordens neurológicas (WHO, 2016). Este estudo também estima os anos de vida saudável perdidos por incapacidade (*Disability-Adjusted Life Year ou DALYs*), sendo considerados fatores de ponderação de acordo com a gravidade da incapacidade gerada. Essas e outras pesquisas retratam um crescente acúmulo de mortes causadas por desordens neurológicas, e apontam o aumento destas nos próximos anos (Figura 1.1).

Figura 1.1 – Cálculo da quantidade de mortes atribuídas a desordens neurológicas.

	Deaths	% total	per 100 000	Deaths	% total	per 100 000	Deaths	% total	per 100 000
WORLD									
Population	2005			2015			2030		
TOTAL DALYs	58 028 152			63 458 962			73 247 767		
Epilepsy	126 096	0.22	1.96	130 569	0.21	1.84	139 276	0.19	1.76
Alzheimer and other dementias	425 331	0.73	6.60	513 230	0.81	7.23	671 372	0.92	8.48
Parkinson's disease	105 012	0.18	1.63	127 293	0.20	1.79	165 418	0.23	2.09
Multiple sclerosis	16 275	0.03	0.25	16 669	0.03	0.23	17 012	0.02	0.21
Migraine	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
Cerebrovascular disease	5 745 748	9.90	89.19	6 466 232	10.19	91.03	7 787 656	10.63	98.36
Poliomyelitis	774	0.00	0.01	654	0.00	0.01	577	0.00	0.01
Tetanus	191 592	0.33	2.97	145 640	0.23	2.05	95 587	0.13	1.21
Meningitis	152 004	0.26	2.36	106 372	0.17	1.50	69 946	0.10	0.88
Japanese encephalitis	11 625	0.02	0.18	7 282	0.01	0.10	4 318	0.01	0.05
Total	6 774 457	11.67	105.16	7 513 942	11.84	105.78	8 951 162	12.22	113.06

Fonte: *World Health Organization* <http://www.who.int/mental_health/publications/neurological_disorders_ph_challenges/en/>

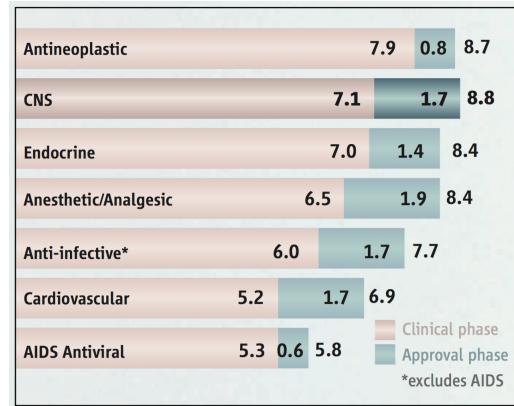
As doenças originadas por desordens neurológicas são complexas e possuem comportamentos intermitentes. A forma mais utilizada para analisar as reações e predições desses

distúrbios em seres humanos é através de experimentos com modelos animais, que permitem a investigação e testes de fenômenos biológicos, seus processos e possíveis riscos. Porém, segundo Emiliangelo Ratti, vice-presidente sênior e chefe do Centro de Empresas Farmacêuticas de Descoberta de Drogas na Neurociência, os testes clínicos são mais complicados em doenças advindas do Sistema Nervoso Central (SNC) (MILLER, 2010).

Além disso, a indústria farmacêutica tem dificuldades no desenvolvimento de novos fármacos para o tratamento de doenças neurológicas. Segundo Thomas Insel, diretor do Instituto Nacional de Saúde Mental nos Estados Unidos, a expectativa na evolução em tratamentos para doenças mentais é baixa (MILLER, 2010). O relatório da *Tufts Medical Center* demonstra que o investimento em desenvolvimento de fármacos para desordens decorrentes do SNC possuem um risco maior comparado a outros tipos de doenças (MILLER, 2010). Isso ocorre, dentre outros fatores, devido a demora para iniciar no mercado (Figura 1.2), o custo elevado de desenvolvimento (Figura 1.3), e o tão somente 8% de aprovação dos testes clínicos (Figura 1.4). Diante dos desafios econômicos e do tempo investido, outras alternativas são necessárias para melhorar esses números.

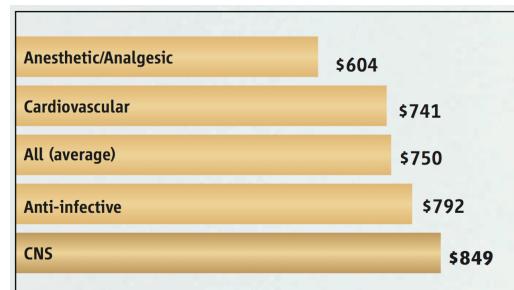
Uma das áreas de pesquisa da neurociência que apresenta um grande avanço é a de Interfaces Cérebro-Máquina (ICM). Seu conceito foi criado no final de 1990, a partir de estudos de circuitos neurais. As ICM tentam estabelecer uma conexão direta entre o cérebro e um dispositivo externo, como artefatos mecânicos, eletrônicos ou virtuais (NICOLELIS, 2016). A primeira demonstração experimental envolvendo ICM foi realizada em 1999, onde foi comprovado que um grupo de neurônios corticais em ratos é capaz de controlar um braço robótico em tempo real (CHAPIN *et al.*, 1999). Desde então, diversas pesquisas em ICM vêm contribuindo para o aumento da qualidade de vida de indivíduos com capacidades reduzidas. Com ICM, é possível manipular próteses (SHEDEED, 2013; OUYANG *et al.*, 2013; PRINCE *et al.*, 2016), se comunicar (CHAURASIYA *et al.*, 2016), controlar corpos virtuais por meio de avatares (HASHIMOTO *et al.*, 2010; LUU *et al.*, 2015; PAVONE *et al.*, 2016), e ainda realizar tratamentos mais efetivos com a combinação de diferentes métodos, como por exemplo a reabilitação neurológica de lesionados medulares (DONATI *et al.*, 2016) e a fisioterapia com ICM para recuperação motora de pacientes com AVC (CURADO *et al.*, 2015).

Figura 1.2 – Tempo em anos de desenvolvimento clínico para um fármaco iniciar no mercado.



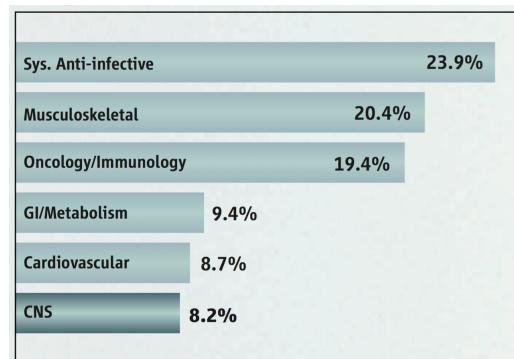
Fonte: Miller (2010)

Figura 1.3 – Custo do desenvolvimento clínico (em milhões)



Fonte: Miller (2010)

Figura 1.4 – Porcentagem de aprovação clínica dos testes dos fármacos para o SNC.



Fonte: Miller (2010)

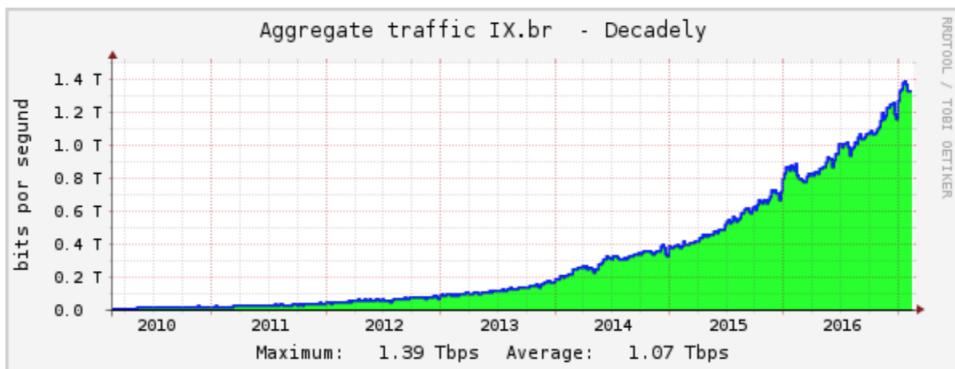
Alguns desafios limitam a abrangência das ICM nas práticas médicas atuais, como: dispositivos de alto custo e conhecimento técnico para sua manipulação; grande volume de dados para armazenar, transmitir, processar e analisar; alta variabilidade intra-indivíduo com consequente especificidade de tratamentos; receptividade de pacientes a novas práticas. Nesse contexto, discutimos a seguir dois conceitos recentes que podem contribuir com uma maior difusão de tecnologias e tratamentos médicos: *internet of things* e *distributed healthcare*.

Em ICM e outras abordagens biomédicas a transmissão contínua de sinais eletrofisiológicos resulta em uma quantidade elevada de dados, os quais requerem uma alta capacidade de armazenamento e um processamento ágil para gerar informações úteis de forma prática e segura. Além disso, há uma maior receptividade por parte dos utilizadores da tecnologia se esta for compacta, fácil de vestir e sem fios (BLABE *et al.*, 2015). Com o advento de sensores, dispositivos de processamento e comunicação miniaturizados e portáteis, praticamente qualquer dispositivo pode ser conectado à internet para fornecer ou receber dados. A conexão entre esses dispositivos é chamada de Internet das Coisas (*Internet of Things* ou *IoT*). A saúde é um dos setores que buscam a automação e informatização de processos visando a logística hospitalar, a acessibilidade, a eficácia e a eficiência aos pacientes. Nesse ambiente, existem dispositivos voltados para necessidades específicas da área, que são também chamados de objetos que se conectam em uma rede biomédica, que possui o intuito de melhorar a aquisição de diagnósticos, tratamento, reabilitação de pacientes e monitoramento, o qual é chamado de Internet das Coisas Médicas (*Internet of Medical Things* ou *IoMT*).

Essa conectividade existe há muito tempo, como o uso de um caixa eletrônico bancário conectado, que reconhece dados pessoais, previne ataques e fornece dados em tempo real a uma rede anti-fraudes, por exemplo. Entende-se por tempo-real sistemas reativos que operam de acordo com uma resposta de tempo restrita. Por sistemas reativos, entende-se como aqueles que funcionam continuamente, recebendo entradas e enviando saídas para componentes de *hardware* (BEN-ARI, 2006). Além disso, essa rede está cada vez mais dinâmica, se conectando com diversos outros objetos pervasivos, obtendo informações pela transmissão de dados a todo o tempo e com disponibilidade de acesso cada vez maior. Um exemplo disso foi dado em uma entrevista na *Cable News Network (CNN)*, onde o diretor executivo da *Local Motors*, *Jay Rogers*, informou que o *Olli*, ônibus autônomo elétrico, já começou a passear nas ruas de *Washington* nos Estados Unidos (MONKS, 2016). Além do transporte usual de pessoas, ele está apto a conversar com seus passageiros e transmitir informações sobre o itinerário sob comando de voz, através da tecnologia cognitiva da *Watson*, da *International Business Machines Corporation (IBM)*. Outro aspecto relevante é o fluxo de acesso à internet, que vem aumentando continuamente. No Brasil, esse acesso foi estimado em 102 milhões de indivíduos em 2015 (58% da população brasileira), um aumento de 12 milhões frente ao ano anterior,

de acordo com publicação da pesquisa pelo Comitê Gestor de Internet (Comitê Gestor da Internet no Brasil, 2015). Em especial, o Brasil possui o maior ponto de troca de tráfego do Hemisfério Sul e um dos maiores do mundo. Os centros de processamentos de dados em São Paulo, no Núcleo de Informação e Coordenação do Ponto BR (NIC.br), foram interligados através de um anel óptico subterrâneo de 20 quilômetros em 2016, aumentando a qualidade e segurança da infraestrutura da internet no país. Com isso, a capacidade e qualidade do serviço de troca de tráfego é ampliada e considerada uma das melhores *datacenters* em nível mundial (PTT.br, .IX.br). As Figuras 1.5 e 1.6 representam o tráfego de dados em *bits* por segundo em décadas e o tráfego de dados em *bits* por segundo diário em algumas regiões do Brasil, respectivamente.

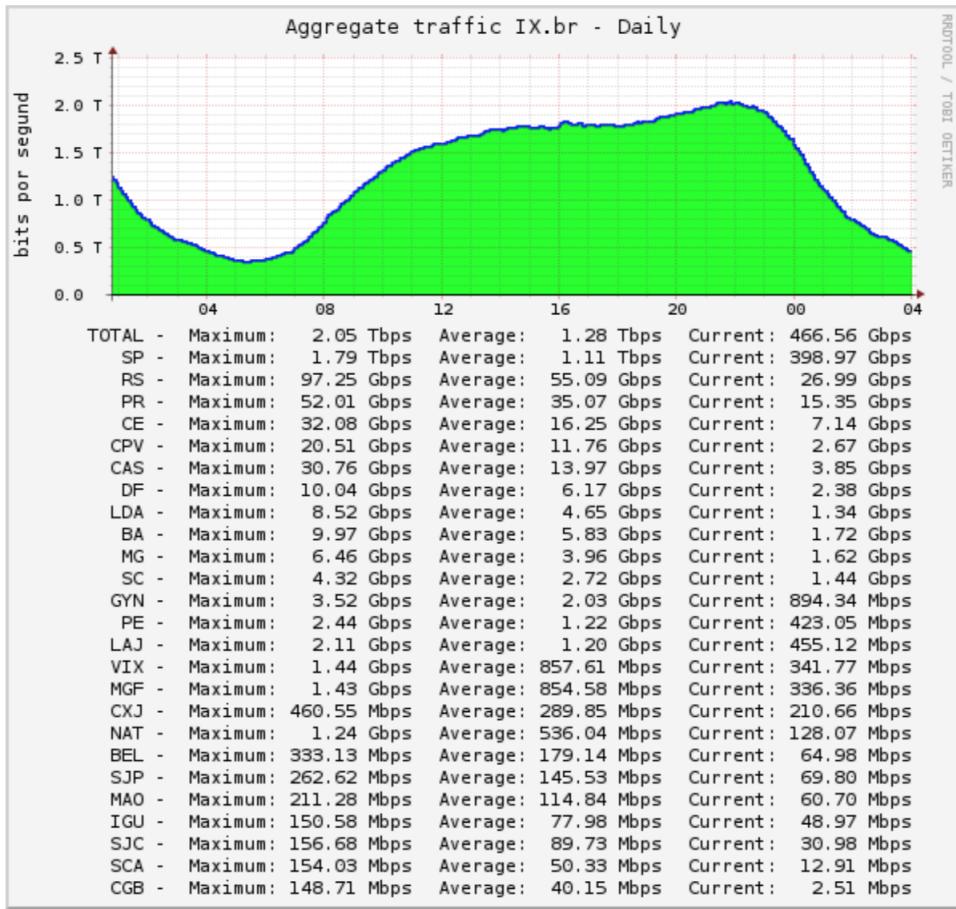
Figura 1.5 – Visualização do aumento do tráfego de dados em *bits* por segundo em décadas.



Fonte: <<http://ix.br>>

Também foi publicado pelo CGI o impacto do uso das Tecnologias de Informação e Comunicação (TIC) nos estabelecimentos de saúde, demonstrando grande interesse no aprimoramento desses sistemas para adaptá-lo à essa demanda. O sistema de proteção e recuperação da saúde do país, criado em 1990, o Sistema Único de Saúde (SUS), principiou conceitos para o progresso da qualidade no atendimento da saúde. Nesse setor, a propagação das TIC é significativa para o aumento do sistema, redução de custos operacionais, clínicos e administrativos. Soluções que envolvam automação dos processos de gerenciamento de informações de pacientes, registro clínicos e administrativos, assim como a substituição desses registros em forma de papel para o meio eletrônico são consideradas uma mudança necessária tanto para o aumento da qualidade do atendimento ao cidadão, como para o compartilhamento de informações para outros estabelecimentos ou pesquisas científicas (Comitê Gestor da Internet no Brasil, 2015). Assim, o Brasil destaca-se ante a seus vizinhos pela estrutura de suporte a IoT mas também pela necessidade e demanda de mais serviços que explorem a plataforma.

Figura 1.6 – Visualização do tráfego de dados em *bits* por segundo diariamente em algumas regiões do Brasil.



Fonte: <<http://ix.br>>

Para suprir o alto volume de informações captadas por sensores de diversos pacientes, algumas abordagens foram propostas nos últimos anos utilizando tecnologias mais sofisticadas. Em 2012, foi apresentado um *framework* para acesso aos serviços médicos de urgência (*Emergency Medical Systems* ou *EMS*), permitindo a comunicação remota entre pontos de emergência utilizando dados na nuvem e uma plataforma desenvolvida para dispositivos *Android* (KOUFI *et al.*, 2012). Em 2015, foi apresentado uma arquitetura distribuída para conectar serviços na nuvem existentes de diferentes provedores, e utilizou a saúde eletrônica *eHealth* para monitorar um sensor de pulso (RADU *et al.*, 2015). Ainda em 2015, com intuito de ter uma estrutura escalável e flexível, surgiu uma plataforma inteligente chamada *iHome*, baseada no *iMedPack* e utilizando *IoT* (Yang *et al.*, 2015). Na *Kyungpook National University*, Coreia, pesquisadores apresentaram um trabalho de desenvolvimento de um sistema para gerenciamento de emergências médicas baseada em *Hadoop* (*software open-source*) para

processamento de dados distribuídos e internet das coisas (*IoT*) (RATHORE *et al.*, 2015).

O segundo conceito relevante para este trabalho é o *distributed healthcare*. *Healthcare* tem o objetivo de manter ou renovar a saúde corporal e mental e *distributed healthcare* é o termo utilizado para a descentralização de serviços de saúde. Projetos de gerenciamento em *healthcare* tem grande potencial para gerar benefícios à sociedade e podem ser utilizados em diversas áreas específicas, como: instalação ou modificação de sistemas de tecnologia da informação, construção ou modificação de instalações, *healthcare delivery*, alteração de processos de gerenciamento em um ambiente de saúde, gerenciamento de equipamentos de saúde (raio-X, *fMRI*, ressonância magnética), atingindo a otimização de processos hospitalares, profissionais da saúde, ambulatórios e outros. É necessário considerar alguns pontos importantes em um desenvolvimento de sistemas para *healthcare*: as informações contidas e trafegadas podem colocar a vida de um ser humano em risco, trata-se de um ambiente fortemente regulamentado, é necessário dados detalhados com informações confiáveis e *backups* consolidados, distribuídos, acessíveis de forma rápida de todos os processos envolvidos, entre outros desafios.

Tipicamente, os sistemas de saúdes geram mais lucros quando os indivíduos estão com problemas de saúde, porém isso está mudando e um dos exemplos é o *Accountable Care Act* que visa o inverso, render mais com o controle da saúde das pessoas à longo prazo (MOLINARI, 2014). O projeto foi iniciado em 5 comunidades: *Spartanburg* (South Carolina), *Clatsop Country* (Oregon), *Lake Country* (California), *Muskegon* (Michigan) e *North Hartford* (Connecticut), como modelo para qualquer outro lugar no mundo. Esse modelo foca no acesso à saúde e consciência das pessoas de aspectos como nutrição, educação e comportamentos em geral que precisam ser mudados para obter uma melhoria da qualidade de vida. Com a evolução da medicina e da tecnologia foi possível a disponibilização de equipamentos específicos e meios de monitoração diretamente para as pessoas que precisam, é uma forma de manter a saúde dos indivíduos independente de onde estejam, com suporte no tempo necessário. Estudos recentes demonstram redução de custos de 15% a 30% comparados a tratamentos obtidos em hospitais em populações de pacientes parecidas e também projeta uma economia de 200 bilhões de dólares nos próximos 25 anos (AUERSWALD, 2015).

Por fim, das 12 tecnologias inovadoras para 2025 previstas pelo *McKinsey Global Institute* (MANYIKA *et al.*, 2013), 8 são relacionadas diretamente com a saúde: a internet das coisas (utilizada para conectar sensores acoplados em pessoas e objetos), o armazenamento de grandes quantidades de dados na nuvem (usufruindo da heterogeneidade de informações), robótica avançada (suporte à tomada de decisão, precisão de diagnósticos e mecanização dos laboratórios), a genômica (engenharia genética e diagnóstico utilizando sequenciamento rápido), saúde móvel (*mHealth*), impressão 3D (obtenção de próteses), inteligência artificial

(auxílio na tomada de decisão) e nanopartículas (utilizadas para novos medicamentos). No Brasil, um acordo recente entre a *Fleury Medicina e Saúde* e a *IBM* (a primeira parceria do tipo na América Latina) utilizará a tecnologia de computação cognitiva da *Watson Genomics* para a análise de dados de sequenciamento genético (FLEURY, 2016). Com o uso dessa ferramenta, é possível identificar a reação de cada mutação e predizer possíveis comportamentos associados a cada tipo de doença do respectivo organismo, permitindo assim tratamentos personalizados. Em resumo, a utilização de dados de múltiplos dispositivos, conectados em redes dinâmicas e remotas, apresenta-se como uma das principais fronteiras da medicina moderna e será fundamental para o acompanhamento personalizado dos pacientes, melhorando os diagnósticos e a eficiência dos tratamentos e investimentos em saúde.

O objetivo deste trabalho é o desenvolvimento de um sistema computacional, integrando diferentes conceitos, tecnologias e ferramentas que ofereçam um meio interpolado de processamento e transmissão de sinais com análise de dados e posterior suporte à tomada de decisão. O foco do desenvolvimento foi no contexto de interfaces cérebro-máquina, *IoT* e *distributed healthcare*, com a diretriz de construção de um sistema de fácil expansão a outros cenários. Como alternativa aos meios tradicionais, os pacientes poderão acessar os registros de seus próprios dados de forma digital, manter um controle de seu histórico e evolução da sua condição de forma descentralizada. Esses dados poderão orientar profissionais da área da saúde ou serem consultados por pesquisadores para posteriores pesquisas. A aquisição dos sinais de interesse pode ser realizada conectando-se ao sistema através de diversos dispositivos externos. As análises desses dados podem ser realizadas de forma remota pelos profissionais vinculados, ativando-se um registro clínico, facilitando a decisão do diagnóstico e a decisão de tratamentos mais específicos.

A contribuição desse trabalho atinge a área social e a científica, fornecendo um meio alternativo para comunicação, armazenamento e transmissão da informação, facilitando a continuidade de tratamentos por diferentes profissionais, aumentando a riqueza de detalhes de informação da condição do paciente, diminuindo o custo do sistema de saúde, viabilizando o tratamento remoto (*distributed healthcare*), diminuindo a exposição de pacientes aos riscos infectológicos existentes em ambiente hospitalar, oferecendo mais autonomia ao paciente, maior conforto, interesse do paciente em manter a patologia controlada, e possibilidade do uso dos dados para futuras pesquisas. Nessa pesquisa não foi encontrado sistemas específicos para interação de profissionais da saúde que acompanhem o tratamento remoto de pacientes com desordens neurológicas. Destaca-se que esse trabalho visa flexibilizar a interação entre os usuários envolvidos e favorecer a portabilidade utilizando *IoMT*.

1.1 Objetivo Geral

Desenvolver um sistema computacional que permita receber, processar e transmitir dados eletrofisiológicos para monitoramento remoto de pacientes por profissionais da saúde.

1.2 Objetivos Específicos

Os objetivos específicos envolvem as etapas de análise, planejamento e implementação do sistema. Sendo elas:

- Análise: descrição geral do sistema, definição de requisitos, modelo conceitual de objetos, tabelas de casos de uso, diagrama de casos de uso e diagrama de classes;
- Design: ferramentas específicas, arquitetura do sinal, *templates*, fluxo de telas;
- Implementação: conexão do *Android* com *Firebase*, autenticação do usuário, seleção do paciente, cadastro do paciente, cadastro do profissional da saúde, definição do menu principal, seleção de registros, interação entre usuários, visualização do sinal em tempo real, gravação dos dados, envio por *email* e visualização do arquivo gerado com os dados.

1.3 Estrutura do Trabalho

Esse trabalho está estruturado da seguinte forma: No Capítulo 2 são apresentados conceitos relevantes para o desenvolvimento do trabalho (sinais eletrofisiológicos, desordens neurológicas, interface cérebro-máquina, *distributed healthcare* e telemedicina, computação pervasiva ubíqua e *IoMT*), *data warehouse*, *business intelligence* e *big data*, programação orientada a objetos e *UML*). O Capítulo 3 cita as ferramentas e tecnologias relevantes utilizadas (*Java*, *Android*, *Android Studio* e *Firebase*) e o Capítulo 4 explica os processos realizados durante o desenvolvimento do projeto. Por fim, o Capítulo 5 apresenta as conclusões e perspectivas de trabalhos futuros.

2 Embasamento Teórico

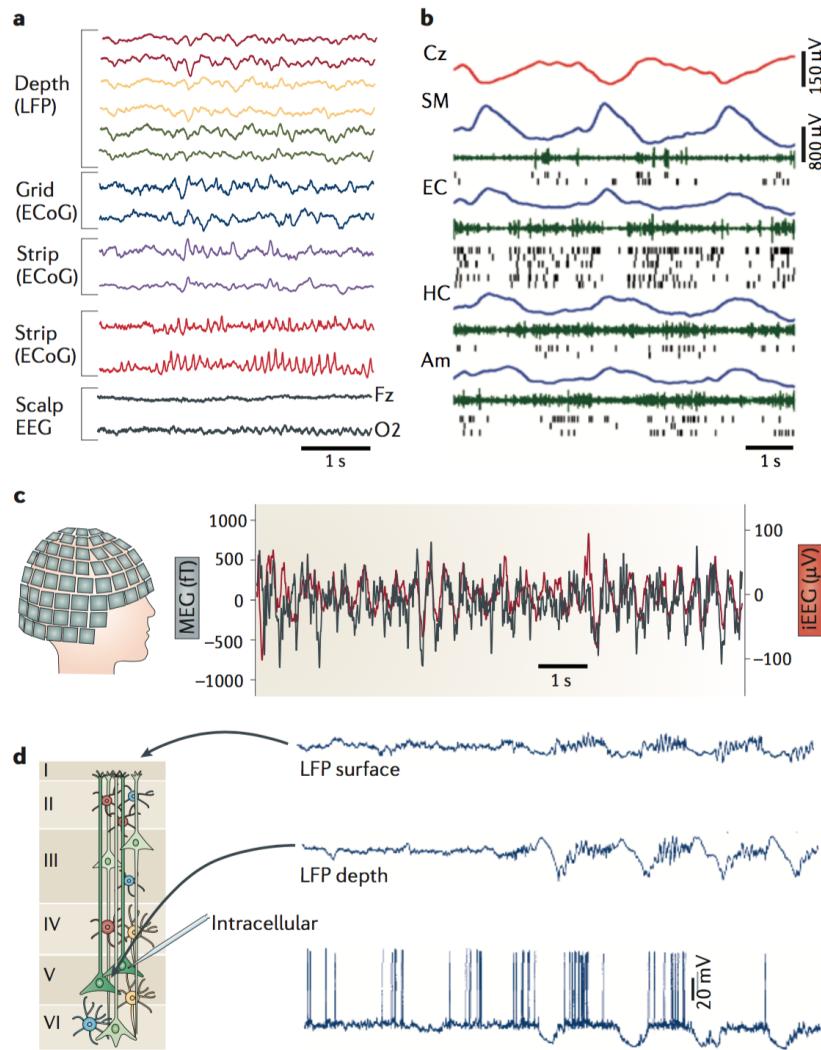
2.1 Sinais Eletrofisiológicos e Desordens Neurológicas

A engenharia elétrica, a biotecnologia e diversos campos da tecnologia moderna são algumas de muitas áreas onde os sinais eletrofisiológicos estão presentes. Eles apresentam-se em variadas formas, sendo possível o uso da biometria, o reconhecimento de voz, a utilização em jogos, em veículos autônomos, entre outros. Na medicina, o número de equipamentos eletrônicos capazes de adquirir e analisar sinais biomédicos com alto nível de detalhes é vasto. Sensores ou transdutores são usados para medir sinais como os da pressão sanguínea, o som da válvula do coração ou sinais de raio-x, (DRONGELEN, 2011) por exemplo, e transformá-los em formato digital para posteriormente serem analisados computacionalmente.

De modo semelhante, a atividade neuronal no cérebro gera correntes entre as membranas do tecido que podem ser medidas no meio extracelular (BUZSÁKI *et al.*, 2012), através dos sinais de eletroencefalografia (EEG), eletrocorticografia (ECOG), potencial de campo local (*local field potential* ou *LFP*) demonstrado (Figura 2.1 (a)), em áreas diferentes do cérebro (Cz, SM, EC, HC e Am) (Figura 2.1 (b)), captação de sinais de magnetoencefalografia (MEG) (Figura 2.1 (c)) e a comparação da gravação realizada em profundidades diferentes de LFP simultaneamente (Figura 2.1 (d)). As análises dessas medidas nos ajudam a entender mais sobre o comportamento dos neurônios e, consequentemente, as desordens neurológicas.

Os neurônios disparam em diferentes formas de onda, frequências, amplitude e duração de tempo. A (Figura 2.2) apresenta o potencial de ação e suas fases: despolarização, repolarização, hiperpolarização e repouso (KANDEL *et. al.*, 2003). Partindo dessa premissa, existem alguns tipos de pré-processamento para destacar os disparos neuronais e eliminar os ruídos captados, alguns deles são: intervalo entre disparos, considerando o tempo após o disparo que a membrana foi afetada até o início da excitação (período refratário), formato de onda dos disparos, sendo usualmente detectados visualmente, análise de componentes principais (ACP), que identifica nuvens (*clusters*) relativamente próximos que identificam cada neurônio, covariância da taxa média de disparos (Quian Quiroga; PANZERI, 2009; PANZERI *et al.*, 2010; SANTOS *et al.*, 2011; SANTOS *et al.*, 2013), coincidência temporal de *spikes* (HARRIS, 2005; Quian Quiroga; PANZERI, 2009), informação mútua (Quian Quiroga; PANZERI, 2009; PANZERI *et al.*, 2010; LONG; CARMENA, 2011). Essas e outras análises já encontram-se em *softwares open-source* como o *KlustaViewa* (ROSSANT *et al.*, 2015). Para a análise desses sinais, pode-se citar como exemplo a Transformada Rápida de

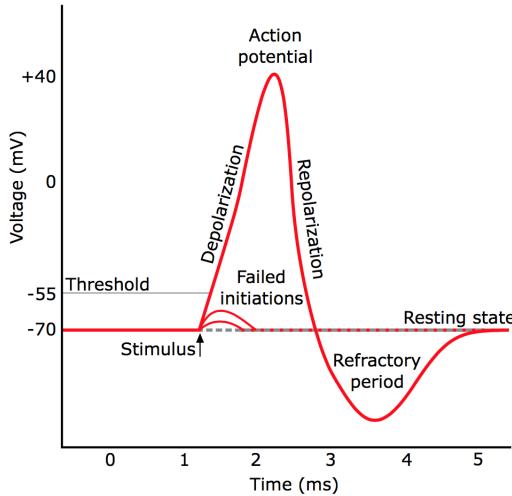
Figura 2.1 – Diferentes métodos de gravação extracelular ((a) formas de onda relativas a sinais de *LFP* (invasivo), ECoG (semi-invasivo) e EEG (não-invasivo); (b) EEG (vermelho), *LFPs* (azul) e potenciais de ação neuronais em diferentes áreas do cérebro - coordenada Cz, área motora suplementar (SM), cortex entorrinal (EC), hipocampo (HC) e amidala (Am); (c) gravação simultânea de MEG (preto) e EEG (vermelho) (d) *LFPs* e potenciais de ação em profundidades diferentes do córtex motor de um gato.



Fonte: Buzsáki (2012)

Fourier, do inglês *Fast Fourier Transform (FFT)*, que facilita a interpretação do dado no domínio da frequência, o espectrograma de potência, que representa uma série de janelas *FFT*, e a coerência espectral, que mede a covariância da potência espectral.

Figura 2.2 – Fases de um potencial de ação: membrana em repouso (equilíbrio dos íons de sódio e potássio entre a membrana), despolarização (após estímulo, o limiar de excitação é ultrapassado), repolarização (período de retorno lentamente da membrana para o limiar) e período refratário (reestabelecimento do equilíbrio dos íons de sódio e potássio).



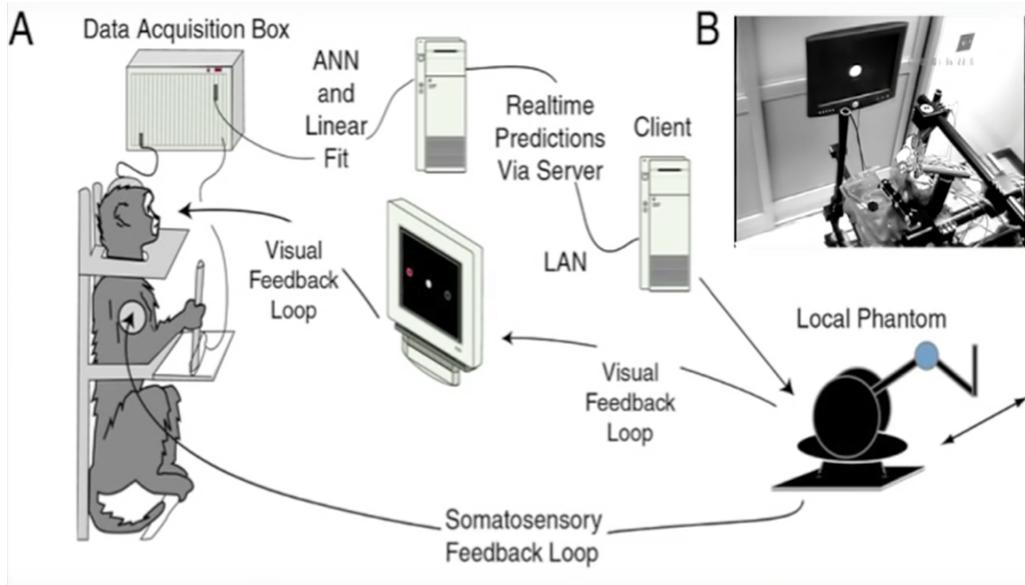
Fonte: <https://upload.wikimedia.org/wikipedia/commons/thumb/4/4a/Action_potential.svg/491px-Action_potential.svg.png>

As desordens neurológicas são derivadas da dessincronização e sincronização de sinais advindos do cérebro, medula espinal, nervos cranianos e periféricos, raízes nervosas, sistema nervoso autônomo, junções neuromusculares e músculos (BEAR *et al.*, 2008). Sendo classificadas em mais de 600 tipos de variações (epilepsia, Alzheimer, Parkinson, tumores cerebrais), algumas causadas por infecções bacterianas (tuberculose, meningite), virais (Zika), fúngicas (*Aspergillus*) e parasitárias (malária, Chagas) (WHO, 2016). Além do retrato do estado do sistema nervoso, os sinais eletrofisiológicos podem ser utilizados em tratamentos por *neuro-feedback* (técnica de terapia que apresenta ao usuário informações em tempo real sobre a sua atividade no cérebro através da medição por sensores elétricos ou de fluxo sanguíneo) ou em Interfaces Cérebro-Máquina (ICM), auxiliados por dispositivos portáteis e/ou vestíveis.

2.2 Interface Cérebro-Máquina (ICM)

As interfaces cérebro-máquina (ICM) realizam a integração entre o cérebro e dispositivos externos (NICOLELIS, 2016). Pesquisas com essa abordagem tem ampliado nos últimos anos, e um dos seus objetivos é restaurar funções motoras perdidas parcialmente ou em sua totalidade de pessoas que sofreram algum tipo de lesão severa. Experimentos com ICM foram

Figura 2.3 – Exemplo de Interface Cérebro-Máquina envolvendo um primata não-humano em uma tarefa com imaginação motora.



Fonte: <<https://www.youtube.com/watch?v=OvqHP7elwTM>>

realizados em roedores e primatas não-humanos em que a atividade cortical foi modulada para realizar tarefas específicas a partir da imaginação da ação motora. Os experimentos demonstraram que quando a imagem ou a informação tátil é rica o suficiente, o cérebro dos animais citados incorporam os dispositivos robóticos ou virtuais como parte do corpo (NICOLELIS, 2016). Ainda, dispositivos neuroprostéticos podem contribuir para a reabilitação motora e neurológica (CURADO *et al.*, 2015).

Para aplicações em humanos, as ICM não-invasivas, que não requerem procedimentos cirúrgicos para utilização, são as mais empregadas. A maioria dos dispositivos que integram as ICM não-invasivas são comerciais, com licença proprietária, em plataformas dependentes de computadores *desktop* e de pouco suporte a outros tipos de dispositivos encontrados no mercado. A (Tabela 2.1) resume algumas das principais aplicações.

Tabela 2.1 – Aplicativos e suas características: S.O, desenvolvedor, nome, licença e dispositivos suportados.

S.O	Desenvolvedor	Nome	Licença	Dispositivos Suportados
Android/Windows	PROATECH LLC	Bioera	Proprietária	Brainmaster/QDS
Android	Myndlift LTD	Techneuro	Proprietária	Muse
Android	Mensia	NeuroRT Suite	Proprietária	ANT: All devices/Biosemi
Java	Amy Palke	Brainathalon	Open source	Nexus
Windows/MacOS	Emotiv	Emotiv Xavier	Proprietária	EPOC/EPOC+/ Insight
Windows	Cyberévolution	Bioexplorer	Proprietária	Pocket/NeuroSky/QDS
Windows	Thought Technology	Biograph Infinity	Proprietária	Thought Technology
Windows	Mind Media	Biotrace	Proprietária	Nexus
Windows	Deymed	BrainBay	Proprietária	Deymed
Windows	Deymed	BrainFeedback Pro	Proprietária	Deymed
Windows	BrainMaster	BrainMaster	Proprietária	BrainMaster
Windows	BrainTrain	SmartMind	Proprietária	Pocket/Neurobics/Neurosky
Linux/Windows	INRIA	OpenVibe	AGPL-3	EPOC/OpenEEG/OpenBCI

Fonte: Autoria Própria

2.3 *Homecare, e-Care, e-Saúde, Distributed Healthcare, Telemedicina e Saúde Móvel (*mHealth*)*

O termo *Home Care*, também chamado de “cuidados em domicílio”, ou ainda “assistência domiciliar”, regulamentada pela Agência Nacional de Vigilância Sanitária (ANVISA) pela Resolução da Diretoria Colegiada no 11 em 26 de Janeiro de 2006, é a modalidade da prestação de serviços da área da saúde realizada na residência do paciente. Esse modelo visa à ações de promoção à saúde, prevenção, tratamento de doenças e reabilitação desenvolvidas em domicílio. Os serviços são utilizados principalmente por pacientes que possuem dificuldade de locomoção e por aqueles que necessitam de internação que possa ser praticada na residência (se enquadrando nas regras e padrões estabelecidos pela resolução RDC no 11). A estabilidade clínica domiciliar pode evitar infecções adquiridas em ambientes hospitalares, otimizar o tempo de recuperação do paciente, oferecer maior conforto e autonomia, diminuir o estresse do paciente e familiares, aumentar o interesse em manter o tratamento controlado, entre outros. Além disso, muitos lugares já enfrentam problemas referentes à insuficiência da quantidade de operadores da saúde. O número de especialistas na área não acompanha o crescimento da população mundial (OMS, 2014).

A palavra *e-Care* é referenciada ao atendimento à distância através do uso de tecnologias que permitem executar serviços de saúde. Essas tecnologias podem ser investidas para englobar um número maior de pacientes, facilitando o controle pelos profissionais de saúde, diminuindo o custo e aumentando a brevidade do atendimento. Seu uso colabora tanto para o quesito da escassez dos profissionais da saúde, como também viabiliza o atendimento de

pacientes que vivem em regiões distantes, por exemplo, em residências rurais (RHIHUB, 2014).

O termo *e-Saúde* (saúde eletrônica) foi criado para incluir tecnologias em saúde emergentes que utilizavam a internet e transações de comércio eletrônico no início dos anos 2000 (MITCHELL, 1999; EYSENBACH, 20001; DELLA, 2001) e pode ser considerada um meio “em que a informação em saúde seja direcionada à pessoa adequada no lugar e no tempo certos de forma eletrônica segura, otimizando a qualidade e eficiência da assistência sanitária, a investigação, a educação e o conhecimento”(OMS, 2014).

Alguns programas apresentaram resultados significativos, como o Sistema de Registro Eletrônico de Saúde (RES) implantado em 2008 pela *Kaiser Permanente Northern* nos Estados Unidos. O sistema envolveu 3,4 milhões de pacientes hospitalizados e ambulatoriais. Desenvolveram-se ferramentas baseadas na internet e dispositivos móveis e vídeos foram utilizados para interação com o paciente (PEARL, 2014). Foi mantida a comunicação anterior como os papéis, telefones e atendimento presencial para assegurar o acesso aos serviços pelos pacientes. A utilização desse sistema teve um crescimento de 4,1 milhões em 2008 para 10,5 milhões em 2013. Segundo relatos dos médicos, 90% dos entrevistados demonstraram satisfação, afirmando que conseguiram manter a qualidade do serviço utilizando as ferramentas *online* desenvolvidas.

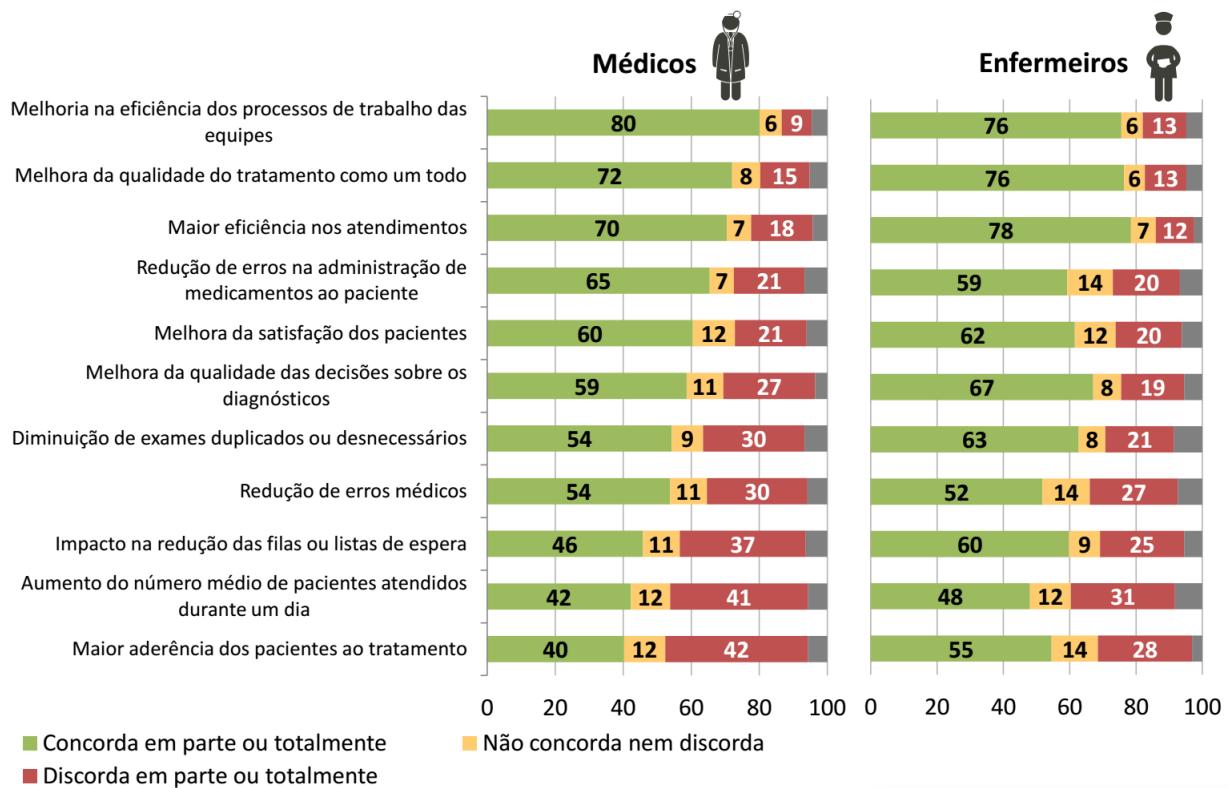
Telemedicina corresponde à utilização da tecnologia digital (computadores, *tablets*, celulares, internet, mídia sociais) nos cuidados à saúde em geral. É possível obter uma digitalização mais rica e segura de registros, retransmitir dados sem cabos, reduzir arquivos em papéis, ter um rápido acesso aos testes e exames realizados, monitorar pacientes por tempo integral, notificar pessoas próximas em casos específicos ou solicitar serviços de emergência. A disponibilidade dos registros com acesso a qualquer tempo e o compartilhamento de alta qualidade das informações de forma segura contribuem na definição de diagnósticos mais adequados, mais esclarecidos e permite alcançar tratamentos mais efetivos.

Uma pesquisa realizada pelo CGI registrou o impacto da implantação de sistemas em ambientes hospitalares, avaliando a eficiência nos processos (Figura 2.4), a proporção de médicos com acesso a computador no estabelecimento de saúde e que consultam de forma eletrônica os dados clínicos sobre os pacientes (Figura 2.5), a proporção de estabelecimentos de saúde com funcionalidades de suporte à decisão disponíveis no sistema eletrônico do estabelecimento (Figura 2.6), e a proporção de estabelecimentos de saúde que possuem departamento ou área de tecnologia da informação (Figura 2.7).

Assim, pode-se verificar a importância do impacto na implantação de sistemas na área da saúde, como as melhorias de eficiência de atendimentos, qualidade do tratamento, redução

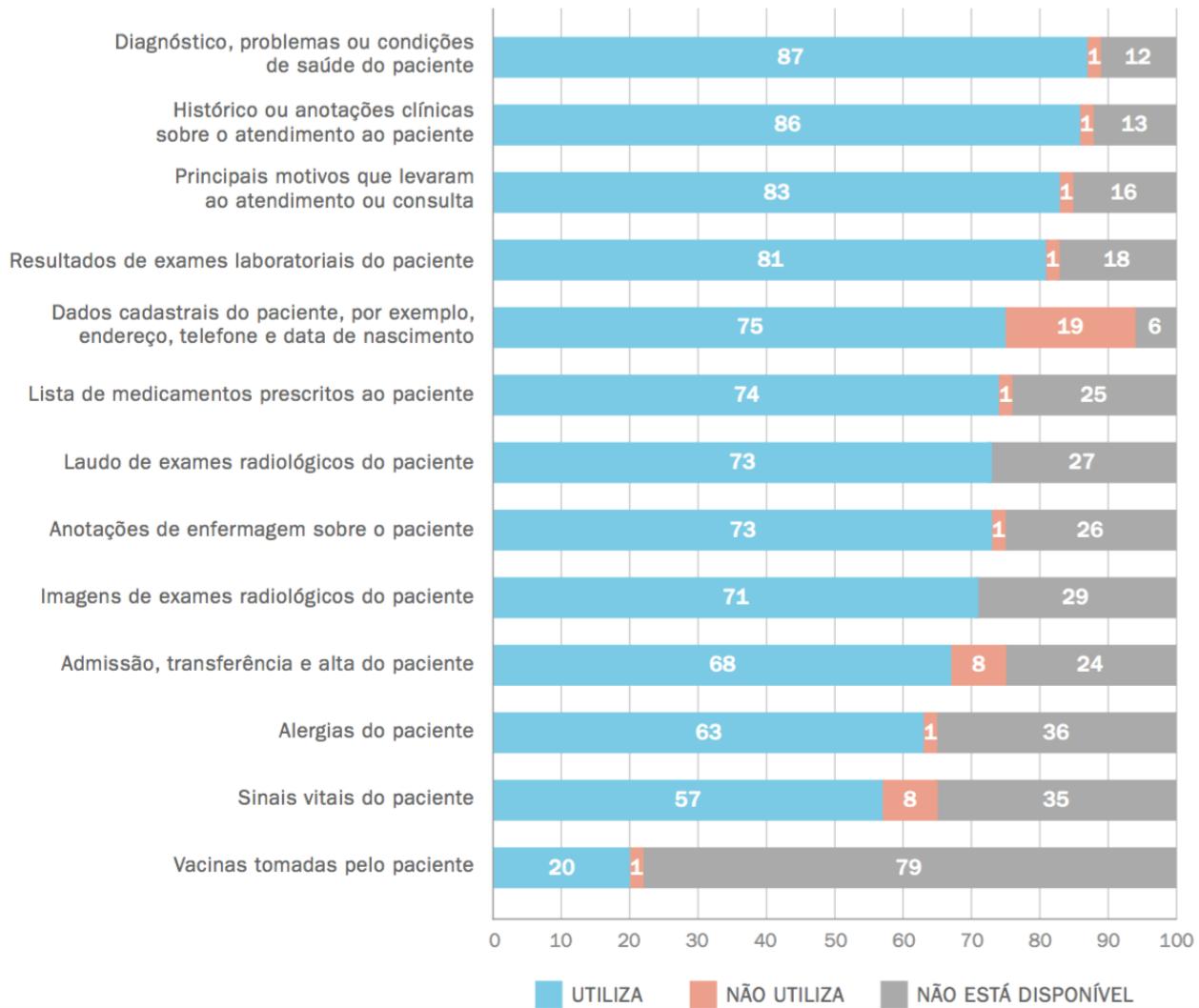
de erros, satisfação dos pacientes, o acesso de profissionais da saúde à tecnologia adequada para a manipulação do uso do sistema e o suporte necessário para manter a utilização. A porcentagem no geral é positiva em relação a aceitação dos usuários, a estrutura para recebimento de sistemas e é uma área que está precisando de mais opções para suprir a demanda.

Figura 2.4 – Impacto na implantação de sistemas



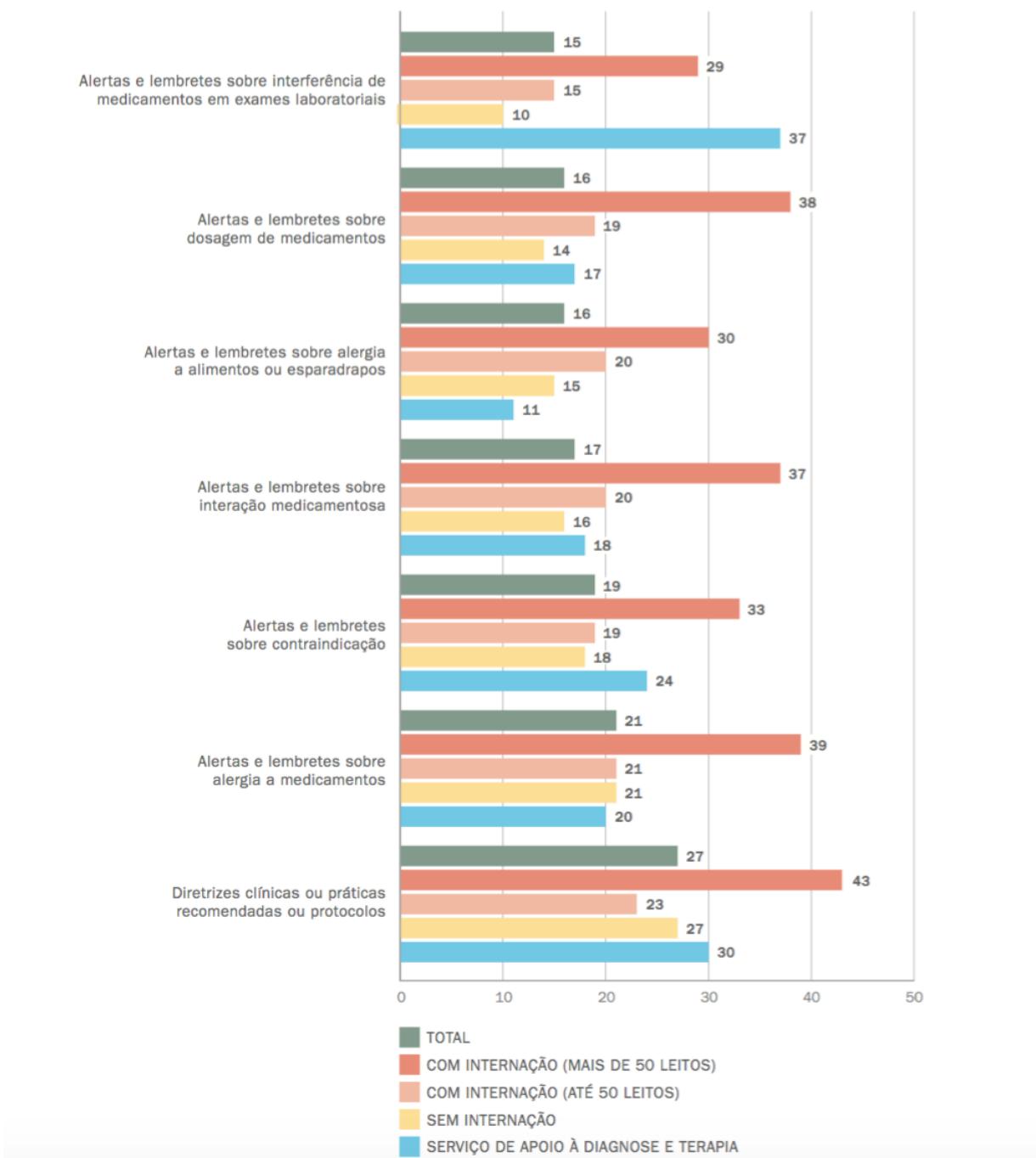
Fonte:CGI, 2015

Figura 2.5 – Percentual sobre o total de médicos com acesso a computador no estabelecimento de saúde



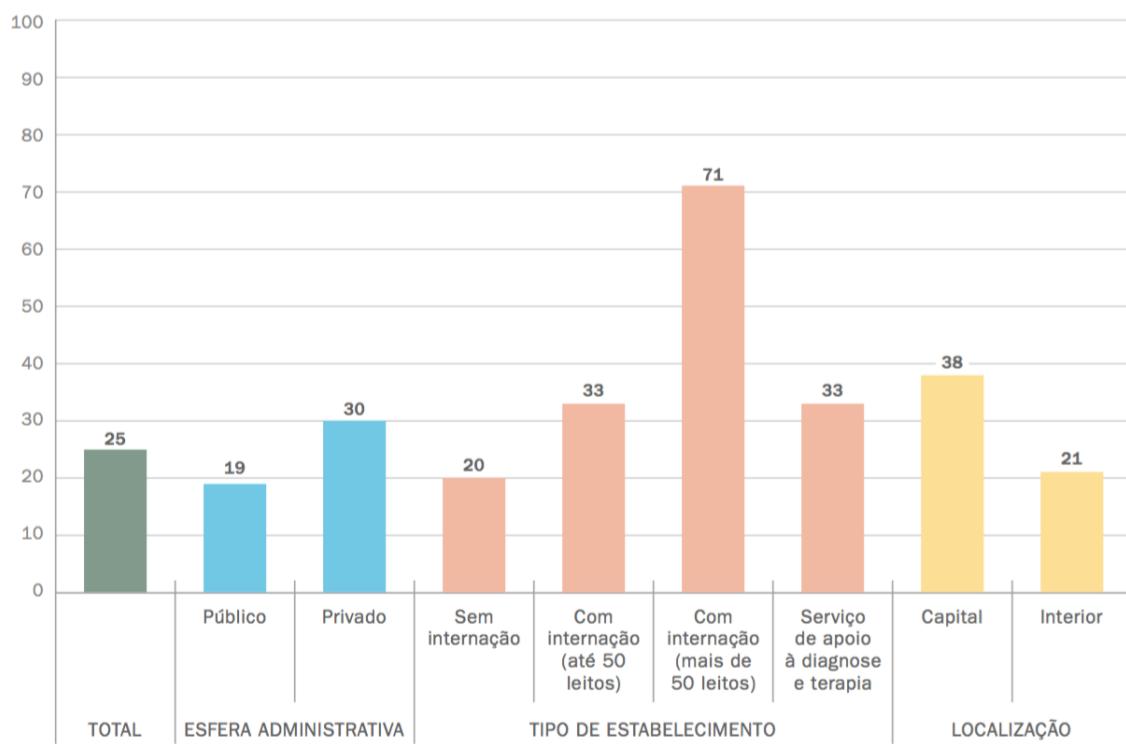
Fonte: CGI, 2015

Figura 2.6 – Percentual sobre o total de estabelecimentos que utilizaram internet nos últimos 12 meses



Fonte: CGI, 2015

Figura 2.7 – Proporção de estabelecimentos de saúde que possuem departamento ou área de tecnologia de informação (2015)



Fonte: CGI, 2015

Saúde móvel, mHealth ou mobile health é o termo utilizado para o uso de dispositivos móveis para suporte dos cuidados médicos e saúde pública, como acompanhamento de tratamentos, rastreio de epidemias e gestão de doenças crônicas (OMS, 2015). É utilizada para comunicação entre indivíduos e serviços de saúde, comunicação entre agências de saúde, prover acesso à informação aos profissionais da saúde, trazendo benefícios de baixo custo, disponibilidade do serviço, maior velocidade, entre outros. De acordo com a PwC, comunidade de pesquisa pela Unidade de Inteligência Econômica, a percepção geral é que a saúde móvel tenha um impacto significante em entrega de cuidados a saúde, trazendo mudanças fundamentais na forma que a indústria de saúde gerencia hoje as agências e profissionais, mudando a percepção entre médicos e pacientes.

2.4 Computação Pervasiva Ubíqua

A computação ubíqua (ubicomp ou *ubiquitous computing*) foi introduzida por Mark Weiser no final dos anos 80, como uma desfragmentação da computação em pequenos objetos que utilizam a tecnologia em nosso favor (WEISER, 2002). Weiser citou ainda que ela seria parte da 3a grande era computacional, o que foi se concretizando a partir dos anos 2000. A ubicomp tem o intuito de tornar a computação invisível no ambiente, incorporada no cotidiano das pessoas através de interfaces naturais. É o enriquecimento computacional nos dispositivos que já fazem parte de nossas vidas, podendo também ser aplicadas em ambientes inteiros e chamados de *smart spaces*. Existem hospitais que se empenham para tornar o ambiente inteligente utilizando-se objetos para detectar as necessidades dos pacientes e então alocar os recursos apropriados.

2.5 Internet das Coisas

A Internet das Coisas (*IoT*) é um paradigma tecnológico que envolve a presença perVASIVA em uma rede de comunicação entre coisas ou objetos variados (ATZORI, 2010). A RFC 7452 (*Architectural Considerations in Smart Object Networking*) submetida pelo *Internet Architecture Board (IAB)* e publicada pelo *Internet Engineering Task Force (IETF)*, que são comunidades dedicadas para evolução da internet, definem que o termo *IoT* é utilizado para designar a tendência que um grande número de dispositivos incorporados que empregam serviços de comunicação oferecidos por protocolos de internet, sendo esses considerados "objetos inteligentes" controlados por componentes posicionados estratégicamente nos ambientes, e não por seres humanos diretamente (RFC7452).

Apesar de ser um conceito relativamente novo, essa conectividade foi discutida em 1991 por Bill Joy, e o termo foi criado em 1999 por Kevin Ashton, que publicou seu artigo "A Coisa da Internet das Coisas" (ASHTON, 2009). Objetos como roupa, tênis, fogão, geladeira, carro, chave, lâmpada, podem mesclar o nosso dia a dia com o mundo digital. Eles podem ser gerenciados por *smartphones*, com intuito de automatizar as tarefas em favor das pessoas, permitindo maior organização, economia de energia, controle da saúde, entre outros.

Seguindo o tema "Tudo que pode ser conectado será conectado", engenheiros e pesquisadores se empenham em projetos que envolvem redes entre os objetos inteligentes. Para essa conexão, é necessário determinar por exemplo como a interação é feita com o mundo real, se é utilizado sensor e/ou atuador, se a bateria é intermitente ou contínua, como é conectado a rede, entre outros (RFC7452). Existem diversos padrões de comunicação entre os objetos inteligentes e é possível utilizar mais de um padrão em um mesmo sistema, sendo documen-

tados e reutilizados pelos desenvolvedores. Alguns deles são: Dispositivo para Dispositivo (*Device-to-Device*), Dispositivo para Nuvem (*Device-to-Cloud*) e Dispositivo para *Gateway* (*Device-to-Gateway*), descritos a seguir.

2.5.1 Dispositivo para Dispositivo (*Device-to-Device*)

Esse modelo de padrão de comunicação é quando dois dispositivos se conectam afim de prover o serviço, como exemplo um aplicativo no celular que envia os dados do tipo de café e quantidade de açúcar via *bluetooth* para uma cafeteira (Figura 2.8), ou a comunicação entre sensores de fabricantes diferentes (Figura 2.9).

Figura 2.8 – Padrão de comunicação entre dispositivos *Device-to-Device* estabelecendo uma comunicação utilizando tecnologia *bluetooth*.



Fonte: <<https://www.embarcados.com.br/modelos-de-comunicacao-para-iot/>>

Figura 2.9 – Padrão de comunicação entre dispositivos *Device-to-Device* estabelecendo uma comunicação entre sensores de diferentes fabricantes.



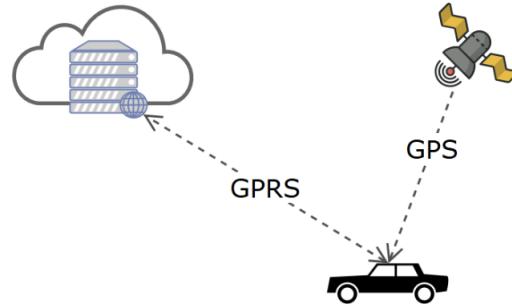
Fonte: <<https://www.embarcados.com.br/modelos-de-comunicacao-para-iot/>>

2.5.2 Dispositivo para Nuvem (*Device-to-Cloud*)

O padrão de comunicação do dispositivo para nuvem envolve a comunicação direta com a internet sem um equipamento intermediário. Por exemplo a transmissão da informação

de posicionamento de um carro sendo transmitida pela rede de dados de telefonia para um serviço na nuvem, através da placa de telemetria com GPS e um rádio GPRS (Figura 2.10).

Figura 2.10 – Padrão de comunicação entre dispositivo e a nuvem *Device-to-Cloud*.

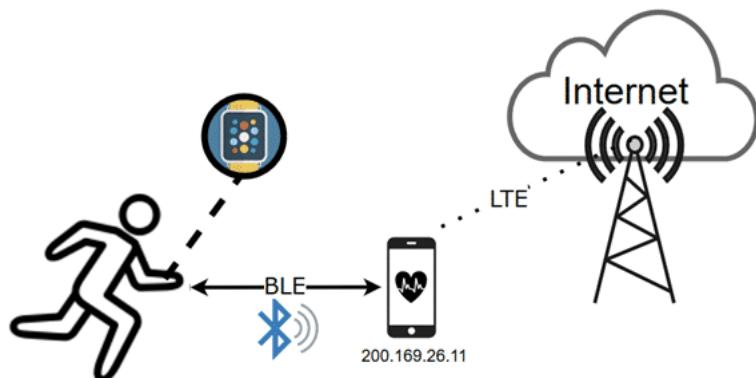


Fonte: <<https://www.embarcados.com.br/modelos-de-comunicacao-para-iot/>>

2.5.3 Dispositivo para *Gateway* (*Device-to-Gateway*)

O padrão de comunicação *Device-to-Gateway* é utilizado o *Gateway* como interoperar o acesso à *internet*. Como exemplo (Figura 2.11), um aplicativo que utiliza o *smartphone* como *gateway* para integrar mais de um tipo de tecnologia (*Long Term Evolution* e *Bluetooth Low Energy*), gerenciando coleta de dados de exercícios físicos por uma pulseira que não está conectada na rede, enviando a informação de pulsação, velocidade e outros dados que possam gerar informações de valor utilizando o *bluetooth* para o celular, que por sua vez conecta à *internet*.

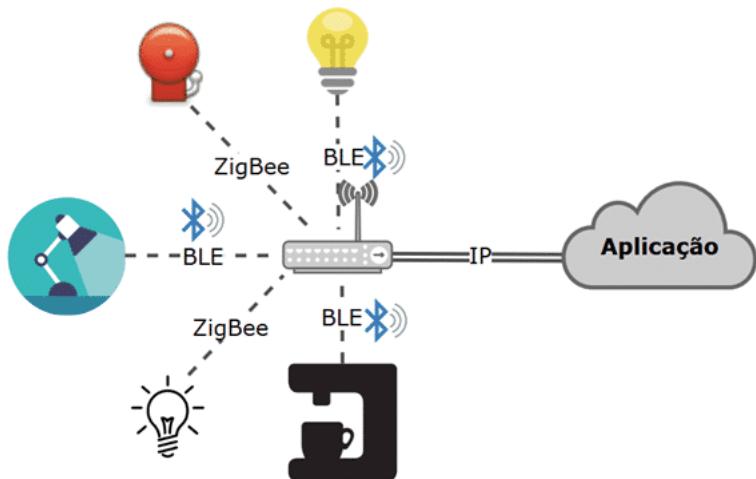
Figura 2.11 – Padrão de comunicação entre dispositivo e o *Gateway* (*Device-to-Gateway*).



Fonte: <<https://www.embarcados.com.br/modelos-de-comunicacao-para-iot/>>

Outro exemplo utilizando o mesmo tipo de padrão de comunicação é em ambientes inteligentes *smart places*, onde os objetos inteligentes como tomadas, lâmpadas e sensores por exemplo utilizam tecnologias como *bluetooth* e *ZigBee* para se comunicar com o *gateway* para conexão com a internet (Figura 2.12).

Figura 2.12 – Padrão de comunicação entre dispositivo e o *Gateway* (*Device-to-Gateway*.)



Fonte: <<https://www.embarcados.com.br/modelos-de-comunicacao-para-iot/>>

No Brasil, o mercado de dispositivos na rede gerou cerca de 2 bilhões de dólares (IDC, 2014), e seu aumento é facilitado pela evolução tecnológica e disponibilidade de dispositivos e de conexão à rede. O uso dessas tecnologias geram uma grande quantidade de dados, e para suprir esse fluxo, foi necessário uma estrutura cada vez mais robusta, descrita nos próximos itens.

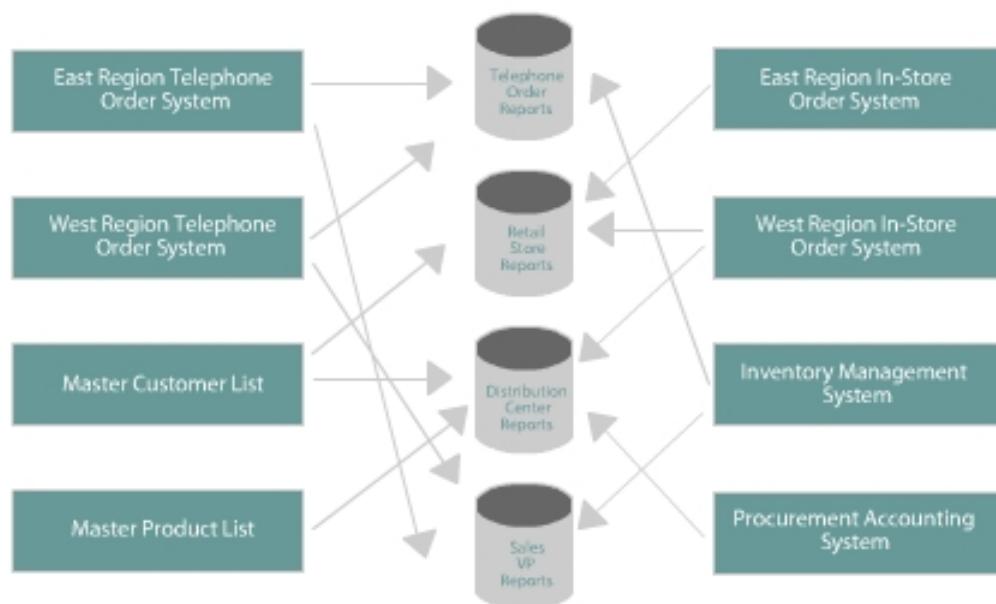
2.6 Data Warehouse, Business Intelligence e Big Data

Na década de 70, diferentes abordagens foram testadas com o intuito de manipular dados para extrair conhecimento. Foram desenvolvidos pacotes de *softwares*, os quais eram utilizados para estruturar dados e produzir um certo nível de análise e criação de relatórios específicos. Um dos modelos criados na época foi o Sistema de Informação de Gestão (SIG), um sistema computacional que auxilia o processo de tomada de decisão (Finlay, 1994). Outros modelos na época foram o Sistema de Suporte à Decisão (SSD) e o *Enterprise Information Systems* (EIS), com propostas similares. Na mesma época, softwares de dados estatísticos começaram a surgir e tinham os chamados "*need-by-need extract files*". Porém, a forma que

os dados foram organizados inicialmente gerava duplicação e conflitos de informação (Figura 2.13).

Devido a necessidade da organização de dados e de suprir transações conflitantes, o *Data Warehouse* surgiu com intuito de prover a estruturação de dados de diferentes aplicações e sistemas de forma eficiente bem como possibilitar ao usuário analisar e reportar resultados de um grande volume de dados. Através dessa estrutura é possível coletar dados específicos e fornecer suporte à tomada de decisão. O *Business Intelligence (BI)*, por sua vez, utiliza os dados gerados no *Data Warehouse* e separa-os de forma específica para gerar relatórios mais refinados, de diferentes pontos de vista, além de permitir o processo analítico online, do inglês *Online Analytical Processing (OLAP)*.

Figura 2.13 – Sistema de Suporte à Decisão



Fonte: Autoria Própria

O *Big Data* está relacionado ao grande fluxo de dados, o qual geralmente não é suportado por computadores pessoais. Não se trata de dados comuns que costumam ser armazenados, manipulados, processados, transmitidos e analisados com a utilização de métodos tradicionais, como alguns softwares estatísticos, banco de dados relacionais ou simples planilhas. Algumas características são o tamanho (volume), a velocidade e formato do dado (variedade) (Laney, 2001). O volume de dados depende da capacidade de armazenamento de um computador pessoal, o qual é relativo ao longo dos anos devido a evolução da tecnologia,

porém, de acordo com a conhecida Lei de *Moore*, na ciência da computação, a cada dois anos aproximadamente a capacidade física e a performance é duplicada. A capacidade de armazenagem de dados de programas também varia. Por exemplo, o *Microsoft Excel*, em 2003, tinha um limite de 65.536 linhas e 256 colunas; em 2016, está em 1.048.576 linhas e 16.384 colunas (*Microsoft, 2016*), e, mesmo passando de 1 milhão, o número ainda é insatisfatório para muitas aplicações. Em pesquisas científicas, são investidas semanas para analisar dados coletados por meses para possibilitar que resultados de trabalhos sejam compartilhados e possivelmente utilizados para a sociedade. Todo o processo pode demorar anos. Portanto, há um ênfase no desenvolvimento de métodos adaptados ao grande fluxo de dados.

Com a necessidade de se processar e analisar uma quantidade de dados maior, com a captação de tipos de dados variados e ainda mais veloz, o paradigma do *Big Data* surgiu com novas ferramentas e regras remodeladas (Figura 2.10). É estruturado para receber volume de dados de diversas fontes, dados estruturados (números, *strings*, datas), semi-estruturados (*emails*, *blogs*, *tweets*) e não estruturados (imagens, vídeos, áudios). Além disso, o dado pode ser transacionado em tempo real.

Para enfatizar o impacto do *Big Data* no desenvolvimento de métodos de processamento de dados, considere a coleção de dados *Iris Data*, que contém a largura e comprimento de pétalas e sépalas de três espécies da planta Iris e é um dos mais utilizados para testes de desempenho de algoritmos, foi coletado por Edgar Anderson e analisado por Ronald Fisher em 1936. No contexto atual, não se trata mais de estruturas simples como do *Iris Data*, pois hoje a cada segundo 2.435 chamadas são feitas no *skype*, 6 mil *tweets* são gerados, 736 *uploads* de fotos no *Instagram* acontecem, 41.426 GB de tráfego na internet são consumidos, 58.395 pesquisas no *Google* são feitas e 67.894 vídeos no *youtube* são visualizados, é necessário uma estrutura adequada para suprir essa necessidade.

2.7 Programação Orientada a Objetos e Linguagem de Modelagem Unificada

A Programação Orientada a Objetos (POO), do inglês (*Object-Oriented Programming* ou *OOP*) possibilita o desenvolvimento de *softwares* mais complexos, utilizando como auxílio da Linguagem de Modelagem Unificada, do inglês *Unified Modeling Language* (UML). A UML é a forma mais prática de diagramação para modelar sistemas baseados em POO, sendo possível planejar e obter a visualização do programa em partes menores, evitando desperdício de tempo e possibilitando o reaproveitamento de código. Arquitetar um sistema antes de codificá-lo é tão importante quanto elaborar uma planta de uma casa antes de

Figura 2.14 – BigData Environment



Fonte: Autoria Própria

construí-la. A probabilidade de erros no código na ausência de planejamento é alta, assim como levantar paredes sem consultar previamente sua altura, largura, sem saber o quanto e quais materiais são necessários para ergue-la pode gerar desperdício de tempo, dinheiro, material e outros problemas posteriores. Conceitos, técnicas e ideias de POO ajudam no processo de desenvolvimento de software. Elas são utilizadas nas etapas de *Object-Oriented Analysis* e *Object-Oriented Design*, onde a primeira foca no entendimento do problema e o que é necessário desenvolver, e a segunda orienta em como o projeto será desenvolvido.

Existem diversas abordagens e metodologias formais disponíveis para orientar um projeto, como: *Scrum*, *XP*, *Unified Process*, *SSADM*, *FDD*, *Kanban*, *CDM*, *DSDM*. Cada uma é escolhida dependendo do nível de complexidade do projeto, quantos *stakeholders* (participantes no projeto) estão envolvidos. O estilo de planejamento varia desde projetos simples, como um jogo de memória em um celular por exemplo, até mais elaborados e sofisticados, como o desenvolvimento de um sistema bancário ou de programas de lançamento de um foguete.

Uma das abordagens mais antigas é a em cascata, a qual segue um modelo linear, seu fluxo é dividido em etapas onde a anterior precisa estar completa para iniciar-se a etapa posterior, sendo que cada uma pode demorar meses para serem executadas. Em desenvolvimento de *software*, é muito comum a criação de algo novo, da elaboração de necessidades específicas

para resolução de problemas diferentes, e criação de suporte alternativo à processos peculiares que surgem com o tempo. Como muitos projetos mudam ao decorrer do desenvolvimento, com novas ideias e necessidades dos usuários, o modelo de processo precisa se adaptar ao estilo do projeto. Assim, a abordagem de metodologia Ágil é muito usada na criação de projetos dinâmicos, onde as atividades são definidas em tempo mais curto, oferecendo maior flexibilidade e possibilitando a execução de forma responsiva.

A POO é um paradigma de programação, um conjunto de convenções, métodos, ferramentas e técnicas utilizadas por diversas linguagens de programação e que começou a ser utilizado com mais popularidade nos anos 80. A POO surgiu da necessidade de desfragmentar um grande programa em "mini programas" que interagem entre si, de substituir a forma procedural por formas mais manuteníveis e modulares, facilitando o planejamento e gerenciamento no ciclo de vida de desenvolvimento de software. Existem outros paradigmas além dos já citados, como programação lógica, utilizada pela linguagem *Prolog*, e programação funcional, suportada por *Haskell*, por exemplo. Para o desenvolvimento de aplicativos é utilizado POO, a qual engloba linguagens como *PHP*, *Python*, *Perl*, *Ruby*, *Objective C*, *C++*, *C#* e *Java*, sendo a última a utilizada nesse projeto por ser a linguagem oficial utilizada no *Android Studio*.

Cada objeto da POO se refere a algo independente, com características e comportamentos próprios, sendo visuais ou não, como uma imagem ou um vetor, respectivamente. Para criar os objetos precisamos identificá-los, descrever o que é e sua funcionalidade. Para isso, criamos classes ou utilizamos classes prontas oferecidas pela POO em bibliotecas específicas ou genéricas, como as típicas de manipulação de dados em vetores e matrizes, ou uso de arquivos e dispositivos de entrada e saída por exemplo. Cada linguagem possui suas próprias classes pré-definidas, e como um dos principais benefícios de POO é o reaproveitamento de código, é necessário manter o padrão e convenções estabelecidas para o entendimento e utilização por todos os envolvidos. Para isso, é imprescindível seguir alguns conceitos, como: *Composition*, *Association*, *Aggregation*, *Constructors*, *Destructors*, *Cardinality*, *Singleton*, *Class-Responsability Collaboration*, *Chain Responsibility*, *Agility*, *Accuracy*, *Maintainability*. Porém os principais são: Abstração (*Abstraction*), Encapsulamento (*Encapsulation*), Herança (*Inheritance*) e Polimorfismo (*Polymorphism*).

A abstração significa manter características básicas, essenciais do objeto, sem se preocupar com detalhes. Por exemplo, em um objeto "carro", é essencial que tenha rodas, que tenha portas, um motor, mas ele não deixa de ser um carro se não tiver película no vidro ou se não for automático. Esse princípio é utilizado na abordagem POO para organizar e aproveitar o código o máximo possível. O encapsulamento é utilizado para diminuir a complexidade desnecessária, protegendo os dados e informações da exposição ampla e direcionando apenas

para manipulações devidas especificadas no programa, diminuindo assim possíveis erros. A herança é uma forma de criar uma classe nova, aproveitando características de uma classe já criada, e polimorfismo é o uso de estruturas para diferentes finalidades, sobrescrevendo o código, permitindo a elaboração de tratamentos específicos.

Como descrito anteriormente, cada metodologia possui suas peculiaridades e características inerentes, porém também apresentam alguns conceitos similares. Dentre as possíveis etapas que podem abranger um projeto de desenvolvimento de software, podemos agrupar genericamente em 5: definição de requisitos, descrição da aplicação, identificação de objetos, descrição da interação entre os objetos existentes no sistema, e elaboração do diagrama de classes.

Nesse capítulo citamos alguns conceitos e características da abordagem POO, sua importância, a comparação com outros paradigmas e cita as etapas do desenvolvimento do sistema. Cada um desses pontos são explorados no Capítulo 4. Na sequência (Capítulo 3), é descrito as tecnologias utilizadas para a execução do sistema.

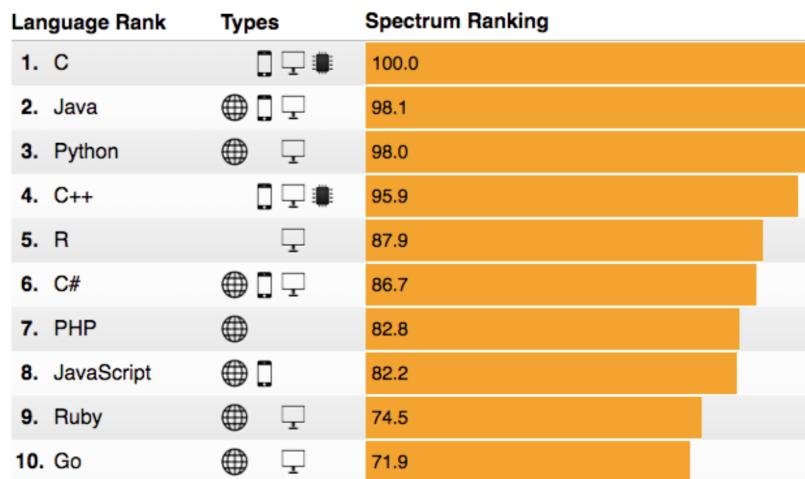
3 Tecnologias

Nesse capítulo temos a descrição sobre as tecnologias relevantes utilizadas para o desenvolvimento do sistema apresentado: Java, Android, Android Studio e Firebase.

3.1 Java

As linguagens mais populares desenvolvidas nas últimas três décadas são oriundo da Programação Orientadas a Objetos (POO), do inglês *Object-Oriented Programming* (OOP), sendo Java uma delas. A primeira versão do Java foi lançada em 1995 pela Sun Microsystems. Foi expandida para desenvolvimento de aplicativos *Web* e *desktop*. O Java possui mais de 4 mil classes pré-definidas, testadas e utilizadas, sendo considerada uma linguagem bem robusta. Java se mantém no ranking das 10 linguagens mais utilizadas nas últimas décadas, e em 2016 ocupou o segundo lugar de acordo com o IEEE (Figura 3.1). Java, no entanto, oferece suporte na *Web*, o que não é verdade para a linguagem C, classificada em primeiro lugar. Java é também a linguagem oficial utilizada em desenvolvimento de aplicativos na plataforma Android utilizando o *software* Android Studio, sendo portanto a utilizada nesse projeto.

Figura 3.1 – Linguagens de Programação mais Utilizadas



Fonte: <<http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>>

O Java suporta o processamento paralelo *concurrency*, possibilitando a execução de processos simultâneos, é orientado a objetos, sendo a estrutura baseada em classes *class-based*,

é independente de plataforma *platform-independent*, gerando *bytecode* que gera em qualquer máquina virtual java (JVM), incorpora o *garbage collector*, a qual gerencia automaticamente a memoria no ciclo de vida do objeto, possui a sintaxe derivada do C++.

3.2 Android

O *Android* é um pacote completo de software, que atua tanto como um sistema operacional para dispositivos móveis, como também um *framework* (*Android Studio*) no qual oferece um ambiente e ferramentas para a criação de aplicativos (*Apps*). O *Android* é baseado em *Linux* e teve sua primeira versão publicada em 2008. Nesse período, já foram incorporados diversos serviços incluindo mapas, mensagens instantâneas, câmera, serviços de busca, serviços de sincronização e tocadores de mídia.

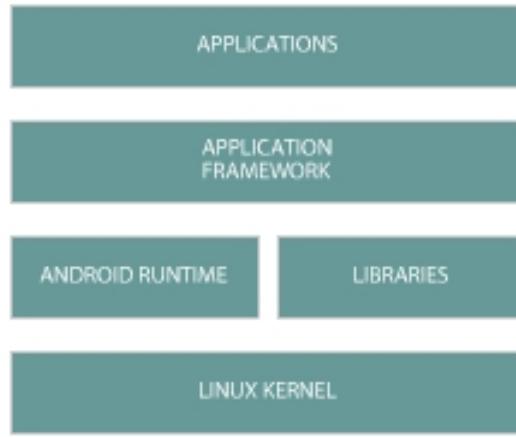
A partir da versão 1.5, em 2009, a qual recebeu o nome "*Cupcake*", cada nova versão começou a também receber nomes específicos de doces. Nesse período foram somados novas características, como: a opção de salvar anexos, a capacidade de salvar imagens nos contatos e o surgimento da animação da transição entre telas. Seguindo o padrão em que a primeira letra do nome é a próxima no alfabeto, as seguintes versões foram: *Donut* (2009), *Éclair* (2009), *Froyo* (2009), *Gingerbread* (2010), *Honeycomb* (2011), *Ice Cream Sandwich* (2011), *Jelly Bean* (2011-2013), *KitKat* (2013), *Lollipop* (2014), *Marshmallow* (2015) e *Nougat* (2016). A junção de todas elas resultaram nas demais funcionalidades no *Android*: a câmera foi integrada com a galeria de imagens, os gestos foram criados para dispositivos multi-touch, o uso do *bluetooth*, as mensagens SMS, a diversidade do tamanho e qualidade das telas, o *Wifi hotspot*, o USB *tethering*, o motor JavaScript V8 no Chrome, o *Garbage Collector* concorrente, a tecnologia NFC, a customização da tela inicial, a introdução da acessibilidade, o modelo de compilação evoluído para o *Android Runtime (ART)*, o suporte duplo de cartões SIM, o reconhecimento de digital, o *Google Now* e os muitos outros recursos disponíveis.

Todas essas tecnologias são estruturadas na arquitetura do *Android* (Figuras 3.2 e 3.3), que tem como base o *kernel Linux*, no qual foi adaptado, considerando as restrições e as particularidades exigidas de dispositivos móveis. Um nível acima da base, encontra-se o modelo de compilação *Android Runtime (ART)*, adotado atualmente, e as bibliotecas de apoio. Mais acima, o *framework* de desenvolvimento, e no topo o aplicativo propriamente dito. O resultado dessa combinação depende também do fabricante do aparelho (Asus, HTC, LG, Motorola), os quais recebem uma parcela de responsabilidade na otimização do resultado final na experiência do usuário. É na camada do "*Linux Kernel*" que se encontram os drivers (audio, *keypad*, *Binder (IPC)*, câmera, *Wi-Fi*, display), a memória *flash* e o gerenciamento de bateria. Na camada do "*Android Runtime*" armazena-se as bibliotecas principais

e as máquinas virtuais (*Dalvik* e *ART*). Ainda no mesmo nível, porém em outro *container*, estão as "*Libraries*", que são bibliotecas extensíveis, como as *FreeType*, o *OpenGL*, *SQLite*, *SSL*, *SGL*, *libc*, *libwebcore* e *media framework*, que são responsáveis por gerenciar bancos de dados, imagens e encriptação, por exemplo. A camada superior, "*Application Framework*", fornece módulos para gerenciar componentes de mais alto nível, como as telas, tecnicamente representada pelas *Activities*, o fluxo de dados entre os "*Apps*", os componentes que fazem parte da aplicação, notificações, localizações. Por fim, a camada do topo "*Applications*" representa tanto os aplicativos instalados na fábrica como aqueles instalados posteriormente pelo usuário.

Ao desenvolver um aplicativo em *Android*, os arquivos do código fonte possuem extensão .java e as bibliotecas ou classes reutilizadas são representadas pelo .class. Esses arquivos são compilados inicialmente pelo *Java C*, resultando em um compilado das classes em um único arquivo .class, que é chamado de *Java Bytecode*. Após essa etapa, existe um passo opcional de se utilizar a ferramenta *ProGuard* para minimizar mais ainda o arquivo, também em extensão .class, chamado *Minimized Bytecode*. A partir do arquivo .class compilado, seja um *bytecode* ou *minimized bytecode*, é que se gera o *DEX Bytecode*, com a extensão .dex, o qual permite a execução pelas máquinas virtuais e posterior tradução para código de máquina no dispositivo.

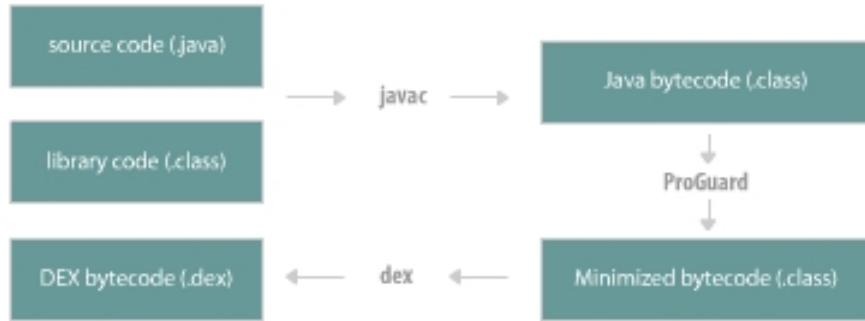
Figura 3.2 – Estrutura *Android* com as camadas: *Linux Kernel*, *ART*, *Libraries*, *Application Framework* e *Applications*.



Fonte: Autoria Própria

São pacotes básicos do *Java* suportados pelo *Android*: *java.xml*, *java.sql*, *java.util*, *java.lang*, *java.io*, *java.nio*, *java.text*, *java.net*, *java.math..*, responsáveis pelo gerenciamento de entrada e saída, tipos de dados, operações aritméticas, entre outros. São pacotes exclusivos

Figura 3.3 – Compilação do Android



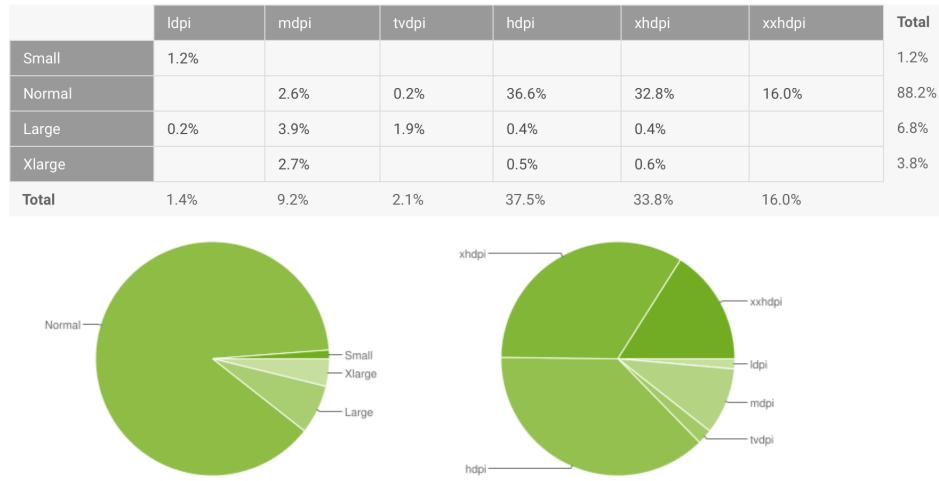
Fonte: Autoria Própria

do *android*: *android.app*, *android.database*, *android.net*, *android.widget*, *android.graphics* e *android.hardware*.

3.3 Android Studio

O *Android Studio* é um ambiente de desenvolvimento integrado, do inglês *Integrated Development Environment (IDE)* oficial do *Google*, baseado no produto da *IntelliJ IDEA* da *JetBrains*. Ele permite o desenvolvimento de aplicativos (*Apps*) que podem ser utilizados em dispositivos que possuem o sistema operacional *Android*. Para criar os aplicativos, essa plataforma exige o conhecimento das linguagens *Java* e *XML*.

O *Java* utilizado é uma versão adaptada para as funcionalidades oferecidas em dispositivos móveis, baseada nas versões do *Java 6* e *Java 7*. A linguagem *XML* é utilizada para os layouts e configurações. Para o desenvolvimento, além da IDE, é necessário o kit de desenvolvimento de software *Android (SDK - Software Development Kit)*, o qual já está embutido no *Android Studio*, o *Oracle JDK*, um kit de desenvolvimento *Java (JDK - Java Development Kit)* oferecido pela *Oracle* (java.oracle.com), e dispositivos móveis virtuais e físicos. O *SDK* é composto por compiladores, ferramentas de *debug* e outras que facilitam a construção, testes e distribuição de um App. É possível criar diversos dispositivos virtuais (*AVD - Android Virtual Devices*) para testes de telas e diferentes versões oferecidas pelo *Android*, porém é importante também ter aparelhos em mãos (celulares, *tablets*, dispositivos vestíveis) para garantir resultados mais precisos. Após a criação finalizada, esses aplicativos podem ser disponibilizados através de alguns canais de distribuição como a *Google Play Store* e a *Amazon Play Store*.

Figura 3.4 – Diferentes Telas e Densidades *Android*

Fonte:<<https://firebase.google.com/>>

A variedade de tamanhos de telas e a densidade de *pixels* caracterizam a fragmentação nos dispositivos *Android*, o que interfere diretamente no desenvolvimento de aplicativos (Figura 3.4). São diversos os fabricantes de aparelhos que suportam o sistema operacional *Android*, dentre muitos estão a LG, Lenovo, Huawei, Sony, HTC, Amazon e a Samsung, essa última dominando 37% do mercado (de acordo com o relatório anual da fragmentação do *Android* pela *OpenSignal*). O aumento da diversidade de modelos alcançou mais de 28% no intervalo de 1 ano (2014 para 2015) referentes a *tablets* e celulares (*OpenSignal*), e em 2016 foram acrescidos os *Chromebooks*. Assim, os desenvolvedores precisam se adaptar para criar aplicativos flexíveis que reconheçam cada dispositivo e suas propriedades específicas de hardware para usufruir da melhor forma possível a tecnologia de cada aparelho.

A *Application Program Interface (API) Fragments* foi desenvolvida a partir do *Android 4.0*, com intuito de reutilizar os elementos da *User Interface (UI)*, através da sua desfragmentação e manipulação em diversos *layouts*, se adaptando assim as diferentes configurações dos dispositivos. Utilizando arquivos *XML* e *Java*, é possível detectar em tempo de execução as configurações do aparelho. O uso de *Fragments API* permite ajuste dinâmico quando o aplicativo é carregado em diferentes tipos de dispositivos com diferentes tamanhos de telas. Ele adapta o conteúdo de forma que aproveita o máximo possível do espaço de tela. Sem o uso do *fragments*, a organização do *layout* é feita manualmente para cada tipo de dispositivo. É gerado mais de um tipo de layout *XML*, englobando para telas *wide* com quantidade de *independent pixels* específicas, e nele a estrutura utiliza o *framelayout* substituindo o *relative*, e o *fragment* encapsulando a classe *Fragment*, que por sua vez carrega o arquivo *layout XML*,

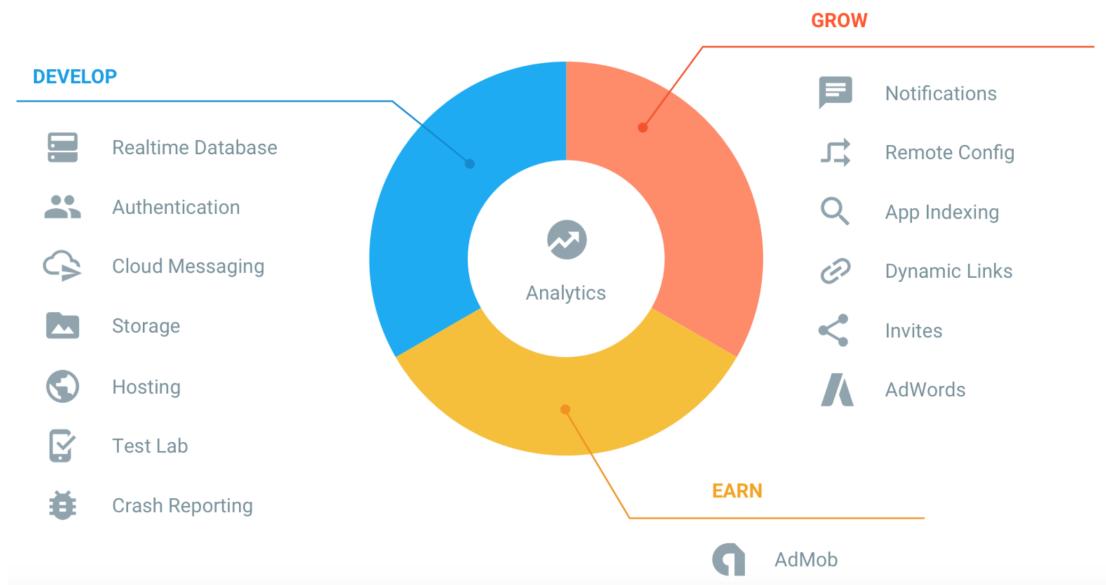
onde finalmente estão os componentes.

3.4 Firebase

Firebase é uma coleção de *API's* do *Google* utilizada para desenvolvimento de aplicativos *mobile* e *web*. A estrutura fornece serviços como: *Real-Time Database*, *Storage*, *Test Lab*, *Authentication*, *Hosting*, *Crash Reporting*, *Cloud Messaging*, *Remote Config*, *Notifications*, *App Indexing*, *Invites*, *AdWords*, *Dynamic Links* e *AdMob*, classificadas em 3 categorias: *Develop*, *Grow* e *Earn* e uma característica central: *Analytics*, que integra as *API's* das categorias anteriores (Figura 3.5). Esses recursos auxiliam o desenvolvedor a medir informações de uso geral do usuário, possibilitando a implementação personalizada ao usuário, utilizando sua localidade, tipo do dispositivo por exemplo ou qualquer outra informação gerada de dados desejados da aplicação, a implementar com mais praticidade a autenticação do usuário sem necessidade de novos cadastros através de contas pessoais pré existentes (*Google*, *Facebook*, *Twitter*, *Github*), compartilhamento de conteúdo fornecidos pelo usuário, interação com usuário por envio de notificações, possibilita o desenvolvimento de aplicativos responsivos mesmo *offline*, pois os dados são armazenados quando ocorre problemas de conexão, entre outros. O Firebase tem um amplo SDK de fácil uso, o site oficial (<https://firebase.google.com>) explica detalhadamente e auxilia na utilização de cada uma de suas características.

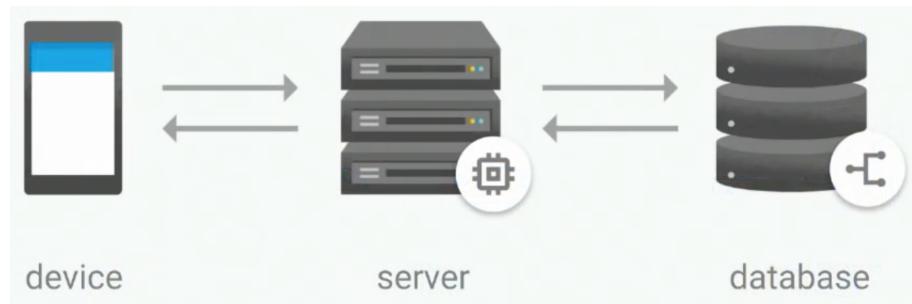
O *Firebase* é baseado na nuvem (*Cloud hosted*), tendo seu conteúdo armazenado no *Google Cloud*, que é o serviço da nuvem oferecido pelo *Google*. A maioria dos servidores trabalham com modelo *response-request* com típicas solicitações HTTP, no qual necessita a criação de uma *query*, disparar a solicitação *request* e aguardar uma resposta *response*, que além de ocupar o processamento e do espaço alocado, também leva mais tempo para aquisição do dado desejado (Figura 3.6). Com a utilização do *Firebase*, uma barreira é eliminada e o acesso aos dados torna-se mais rápido, e não há espera de resposta (Figura 3.7). Com o serviço *Firebase Realtime Database*, é possível que o sistema envie dados (Figura 3.8) e receba o retorno das informações (Figura 3.9) assim que forem alteradas e informe aos usuários de forma sincronizada (250 milissegundos em média).

Figura 3.5 – Recursos do Firebase



Fonte:<<https://firebase.google.com/>>

Figura 3.6 – Modelo request-response



Fonte:<<https://firebase.google.com/>>

Figura 3.7 – Modelo utilizando Firebase



Fonte:<<https://firebase.google.com/>>

Figura 3.8 – Envio de informação para o *Firebase*



Fonte:<<https://firebase.google.com/>>

Figura 3.9 – Atualização das informações automáticas pelo *Firebase*



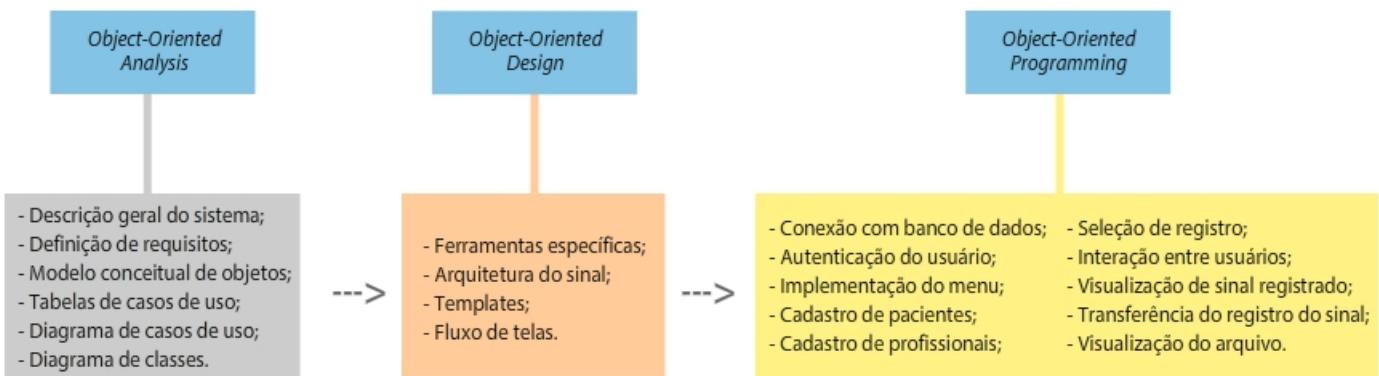
Fonte:<<https://firebase.google.com/>>

4 Desenvolvimento do *Spike Scan*

No Capítulo 2.7 foram apresentados alguns conceitos da programação orientada à objetos, que serão utilizadas ao longo desse capítulo. A sequência estruturada do desenvolvimento foi dividida em 3 fases gerais: Análise (*Object-Oriented Analysis*), referente ao entendimento do sistema, Design (*Object-Oriented Design*), que detalha o planejamento do sistema e a Programação (*Object-Oriented Programming*), que descreve a implementação de etapas do sistema.

Na primeira fase, foram escolhidas as etapas de descrição geral do sistema, definição de requisitos, modelo conceitual de objetos, a elaboração de tabelas de casos de uso, o diagrama de casos de uso e o diagrama de classes. A fase seguinte abrange as etapas de utilizadas, a arquitetura do sinal, a criação de *templates* e a definição do fluxo das telas do sistema. Na última fase, as etapas foram de implementação da conexão com o banco de dados, a autenticação dos usuários, a implementação do menu principal, a seleção de registros, a interação entre usuários, a solicitação de alertas, a visualização do sinal registrado, a transferência do registro do sinal, a consulta do envio do sinal e a visualização do arquivo com os dados.

Figura 4.1 – Fases gerais do sistema: *Object-Oriented Analysis*, *Object-Oriented Design*, *Object-Oriented Programming* e suas etapas respectivas.



Fonte: Autoria Própria.

4.1 Análise (*Object-Oriented Analysis*)

Nessa etapa são descritas as fases de análise do sistema: descrição geral do sistema, definição de requisitos, modelo conceitual de objetos, tabelas de casos de uso e diagrama de classes.

4.1.1 Descrição Geral do Sistema

O *Spike Scan* é um sistema modular de aquisição de dados eletrofisiológicos, com intuito de se integrar a diferentes dispositivos de captação de sinais oferecidos hoje no mercado, bem como armazenar, processar e transmitir as informações pelos possíveis usuários do sistema: pacientes, profissionais da saúde, pesquisadores e alertas (familiares e amigos vinculados).

4.1.1.1 Funcionalidades do Sistema

O sistema visa o acesso à dados de forma remota e prática, possibilitando a monitoração de pacientes, oferecendo as seguintes funcionalidades:

1. Registro de dados relevantes da condição do paciente:
 - Captação de sinais de dispositivos externos (dados originais);
 - Registro do usuário paciente (relato pessoal/auto-avaliação);
 - Registro do usuário profissional (registro clínico) e exames;
 - Armazenamento dos dados processados;
 - Análise de dados registrados pelo profissional ou pesquisador;
 - Dados previamente armazenados em arquivos digitais.
2. Plotagem do dado captado por dispositivo externo, possibilitando o monitoramento *online* e remoto;
3. Seleção de atributos para a filtragem do sinal desejado;
4. Transmissão das informações geradas pelo sistema aos usuários;
5. Seleção de gravação e armazenamento de dados parciais;
6. Comunicação remota entre os usuários;
7. Alertas à usuários externos registrados.

4.1.1.2 Recursos para os Usuários

O sistema apresentado possibilita a interação entre 4 tipos de usuários: *PatientUser* (*PU*), *HealthcareProfessionalUser* (*HU*), *ResearcherUser* (*RU*) e *AllertUser* (*AU*).

O usuário *PatientUser* poderá se cadastrar no sistema independente de vínculo com algum usuário *HealthcareProfessionalUser*, podendo registrar relatos pessoais/auto-avaliação de sintomas e observações detalhadas julgadas relevantes para registro no histórico da condição específica.

O usuário *HealthcareProfessionalUser* poderá adicionar usuários *PatientUser* em sua lista de acompanhamento individual de pacientes, participar do registro de informações no histórico da condição do paciente por registro clínico, podendo adicionar exames realizados na clínica, consultar, processar, analisar dados registrados, solicitar uso remoto de dispositivo para monitoramento *online* ou gravação para posterior consulta, enviar e receber mensagens instantâneas.

O usuário *ResearcherUser*, caso autorizado poderá ter acesso desde informações mais gerais, como a quantidade de usuários com uma determinada condição específica e ou dados e informações parciais geradas, processadas, analisadas pelos usuários *PatientUser* e/ou *HealthcareProfessionalUser* e disponibilizadas para fins de pesquisa de acordo com autorização em termos pré-estabelecidos, protegendo a identidade do usuário e demais características relevantes para pesquisa.

O usuário nomeado como *AllertUser* poderá receber mensagens instantâneas caso o sistema detecte alguma alteração considerada crítica no sinal monitorado online e classificados como emergência, ou da própria solicitação de emergência direta do usuário *PatientUser*, estando ciente assim da atual condição e localização de um usuário vinculado.

4.1.1.3 Aplicações do Sistema

O sistema desenvolvido possui o propósito de gerar os seguintes benefícios aos usuários:

- Tratamentos de pacientes de forma remota;
- Otimização do tempo em espera de pacientes em clínicas e hospitais;
- Minimização de deslocamento físico desnecessário para acesso à informações digitais;
- Registro de dados em situações cotidianas;
- Praticidade na comunicação entre paciente e profissional da saúde;

- Auxílio a pesquisas científicas, fornecendo dados reais de forma segura;
- Dinamicidade e sincronia no acesso aos dados gerados;
- Diminuição de duplicação de informação em estabelecimentos de saúde;
- Rapidez na transmissão de alertas aos usuários em situação de emergência pré-determinada.

4.1.2 Definição de Requisitos

Para a etapa de levantamento de requisitos foi realizado entrevistas informais com os possíveis usuários do sistema, pesquisadores, pacientes e profissionais da saúde, assim como a técnica de etnografia, que permitiu a visualização do fluxo de informações em atendimentos locais. Através desses métodos foram identificados os seguintes requisitos:

- O sistema deve reconhecer dispositivos externos de captação de sinal conforme implementação realizada do módulo específico;
- O sistema deve disponibilizar sinais capturados pelo dispositivo externo de captação de sinal;
- O sistema deve disponibilizar meio para *upload* de arquivos de sinais previamente gravados;
- O sistema deve gravar informações dos dados captados total ou parcialmente pelo dispositivo conforme seleção do usuário;
- O sistema deve plotar gráficos utilizando dados originais captados dos dispositivos externos ou dados pré-gravados conforme atributos pré-selecionados;
- O sistema deve transmitir o sinal recebido por um usuário para outro conforme permissões do *PatientUser*;
- O sistema deve processar dados de forma *online* ou *offline*;
- O sistema deve disponibilizar arquivos previamente gravados pelos usuários;
- O sistema deve sincronizar os dados manipulados por diferentes usuários;
- O sistema deve registrar análises clínicas elaboradas pelo usuário *HealthcareProfessionalUser*;
- O sistema deve registrar auto análise descrita pelo *PatientUser*;

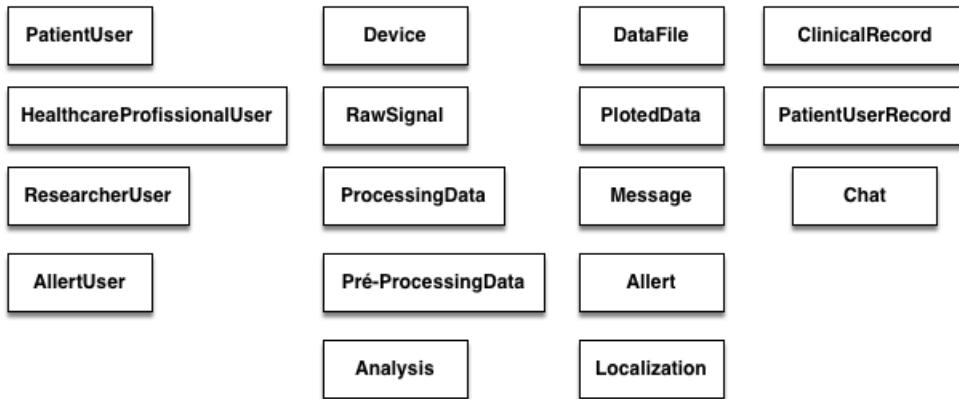
- O sistema deve gravar todas as informações no servidor remoto quando *online*;
- O sistema deve gravar todas as informações no dispositivo local quando *offline*;
- O sistema deve sincronizar dados pendentes caso tenha registro offline;
- O sistema deve permitir o envio de mensagens com no máximo 200 caracteres cada;
- O sistema deve funcionar mesmo sem conexão à internet e sincronizar quando conectar;
- O sistema deve produzir automaticamente uma análise comparativa mensal (usando a data de registro atual com a data desejada registrada anteriormente) relatadas por *PatientUser* e *HealthcareProfessionalUser*;
- O sistema deve listar os dispositivos externos de captação de sinal detectados;
- O sistema deve exibir um gráfico com resumo do monitoramento dos sintomas referentes ao tratamento;
- O sistema exibe a opção de adicionar registro de auto análise no dia atual do acesso;
- O sistema permite adicionar outros usuários para compartilhar informações relevantes para o acompanhamento do tratamento;
- O sistema oferece solicitação de emergência, enviando notificação com a localização atual registrada do *PatientUser* para os usuários vinculados: *AllertUsers* e *HealthcareProfessionalUser*;
- O sistema deve listar os *PatientUser* vinculados e se estão ativos ou não;
- O sistema exibe a opção de adicionar registro clínico no dia atual do acesso.

4.1.3 Modelo Conceitual de Objetos

Utilizando as informações dos diagramas de casos de uso do item anterior, foi possível criar o modelo conceitual de objetos. Os objetos representam usuários, entidades ou características que o programa necessita. Tipicamente, estão associados à objetos no mundo real como um celular, uma caneta, um abajur, um foguete. Cada objeto possui seus atributos como cor, tamanho, peso (caneta pode ser azul e espessura fina) e seus métodos (abajur pode estar ligado ou desligado). Esses atributos e métodos são independentes dos demais objetos existentes – se um objeto celular está ligado, não significa que outro objeto celular também esteja. Assim, cada objeto possui as 3 características citadas (identidade, atributos e comportamentos). É importante enfatizar que há uma tendência em identificar objetos como

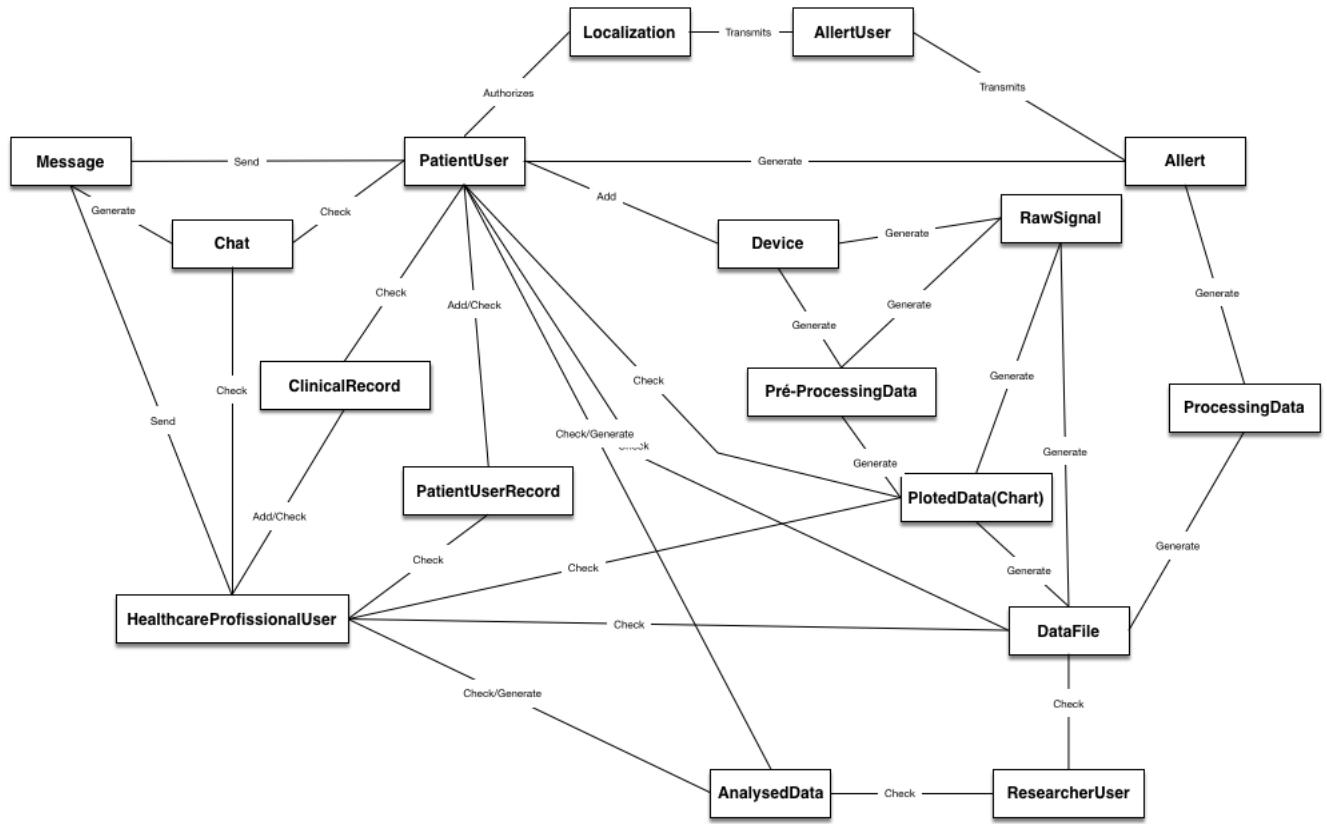
algo palpável, como os objetos citados anteriormente, mas em computação pode haver maior abstração (tempo, conta bancária, vetor). O modelo conceitual de objetos trata-se da identificação inicial dos possíveis objetos mais importantes do sistema de forma genérica, não se trata de uma lista exaustiva mas um mapeamento inicial para ajudar no desenvolvimento dos processos seguintes. Nesse caso foram detectados os objetos: *PatientUser*, *HealthcareProfessionalUser*, *ResearcherUser*, *AllertUser*, *Device*, *RawSignal*, *ProcessingData*, *Pre-processingData*, *Analysis*, *DataFile*, *PlotedData*, *Message*, *Allert*, *Localization*, *ClinicalRecord*, *PatientUserRecord*, *Chat*, conforme (Figura), e a relação entre eles está expressa na (Figura). Esse processo facilita a identificação das interações necessárias entre os objetos no sistema, reunindo os dados necessários para a etapa de identificação das classes.

Figura 4.2 – Modelo conceitual de objetos Spike Scan



Fonte: Autoria Própria.

Figura 4.3 – Modelo conceitual de objetos relacionados Spike Scan



Fonte: Autoria Própria.

4.1.4 Tabelas de Casos de Uso

A elaboração de casos de uso contribui para o esclarecimento do objetivo da aplicação, bem como na solução de problemas e no suporte a criação das classes. A seguir, apresenta-se alguns diagramas de casos de uso elaborados no desenvolvimento do sistema Spike Scan.

4.1.4.1 Monitorar Sinal Online

O caso de uso de monitoramento de sinal online descreve a visualização dos dados transmitidos de um dispositivo externo conectado no sistema, podendo ser visualizados pelo *PatientUser*, que por sua vez pode compartilhar a visualização para o *HealthcareProfessionalUser*.

Tabela 4.1 – Monitorar sinal online

Título	Monitorar sinal online
Autor	<i>PatientUse, HealthcareProfessionalUser</i>
Cenário	<p>1. Usuário logado conecta dispositivo no sistema;</p> <p>2. Sistema reconhece dispositivo;</p> <p>3. Usuário seleciona dispositivo no sistema;</p> <p>4. Dispositivo é reconhecido como ativo;</p> <p>5. Usuário solicita monitoramento online para o sistema;</p> <p>6. Sistema recebe sinal do dispositivo;</p> <p>7. Sistema exibe gráfico com dados do dispositivo.</p>
Extensões	Sistema não reconhece o dispositivo
	a) Usuário reconecta dispositivo;
	b) Usuário cancela solicitação.
Pré-condições	Dispositivo ativo.
Pós-condições	Gráfico na tela.

4.1.4.2 Gravar Sinal Online

O caso de uso gravar sinal online descreve o fluxo do armazenamento do sinal captado do dispositivo externo, que pode ser acessado posteriormente pelos usuários.

Tabela 4.2 – Gravar sinal online

Título	Gravar sinal online
Autor	<i>PatientUser</i>
Cenário	<p>1. Usuário logado conecta dispositivo no sistema;</p> <p>2. Sistema reconhece dispositivo;</p> <p>3. Usuário seleciona dispositivo no sistema;</p> <p>4. Dispositivo é reconhecido como ativo;</p> <p>5. Usuário solicita gravação do sinal para o sistema;</p> <p>6. Sistema recebe sinal do dispositivo;</p> <p>7. Sistema armazena dados transferidos pelo dispositivo.</p>
Extensões	Sistema não reconhece o dispositivo
	a) Usuário reconecta dispositivo;
	b) Usuário cancela solicitação.
Pré-condições	Dispositivo ativo.
Pós-condições	Arquivo com dados registrados do dispositivo.

4.1.4.3 Seleção de Sinal Parcial

O caso de uso seleção de sinal parcial pode ser utilizado pelo usuário *PatientUser* ou *HealthcarePatientUser* para selecionar apenas o sinal de interesse para análise.

Tabela 4.3 – Seleção de Sinal Parcial

Título	Seleção de sinal parcial
Ator	<i>PatientUser, HealthcareProfissionalUser</i>
Cenário	1. Usuário logado consulta sinal gravado; 2. Sistema exibe arquivo selecionado; 3. Usuário seleciona a parte específica do arquivo; 4. Sistema grava dado em arquivo separado.
Extensões	Sistema não possui arquivo gravado. a) Usuário cancela solicitação.
Pré-condições	Arquivo de gravação de sinal aberto.
Pós-condições	Criação de novo arquivo com dados parciais.

4.1.4.4 Processamento de Sinal *offline*

O processamento de sinal *offline* refere-se à aplicação da análise em arquivos previamente armazenados no sistema, podendo ser utilizando tanto pelo *PatientUser* como pelo *HealthcareProfissionalUser*.

Tabela 4.4 – Processamento de sinal offline

Título	Processamento de sinal offline
Autor	<i>PatientUser, HealthcareProfissionalUser</i>
Cenário	1. Usuário logado seleciona arquivo com sinal desejado; 2. Sistema exibe arquivo selecionado; 3. Usuário seleciona atributos de processamento; 4. Sistema gera novo sinal resultante da aplicação de atributos.
Extensões	Sistema não possui arquivo gravado. a) Usuário cancela solicitação.
Pré-condições	Arquivo de gravação de sinal aberto.
Pós-condições	Criação de novo arquivo com dados pré-processados.

4.1.4.5 Pré-Processamento de Sinal *online*

O caso de uso pré-processamento de sinal *online* descreve o fluxo de processo da captação do sinal de um dispositivo externo conectado ao sistema com aplicação de atributos selecionados pelo usuário *PatientUser*.

Tabela 4.5 – Pré-Processamento de Sinal online

Título	Pré-Processamento de sinal online
Ator	<i>PatientUser</i>
Cenário	1. Usuário logado conecta dispositivo no sistema; 2. Sistema reconhece dispositivo; 3. Usuário seleciona dispositivo no sistema; 4. Dispositivo é reconhecido como ativo; 5. Usuário solicita pré-processamento da aquisição do sinal; 6. Usuário seleciona atributos de pré-processamento; 7. Sistema recebe sinal do dispositivo; 8. Sistema pré-processa o sinal recebido pelo dispositivo; 9. Sistema gera novo sinal resultante da aplicação dos atributos. Sistema não detecta dispositivo.
Extensões	a) Usuário reconecta dispositivo; b) Usuário cancela solicitação.
Pré-condições	Dispositivo ativo; Seleção de pelo menos um atributo;
Pós-condições	Criação de novo arquivo com dados pré-processados.

4.1.4.6 Análise de Sinal *offline*

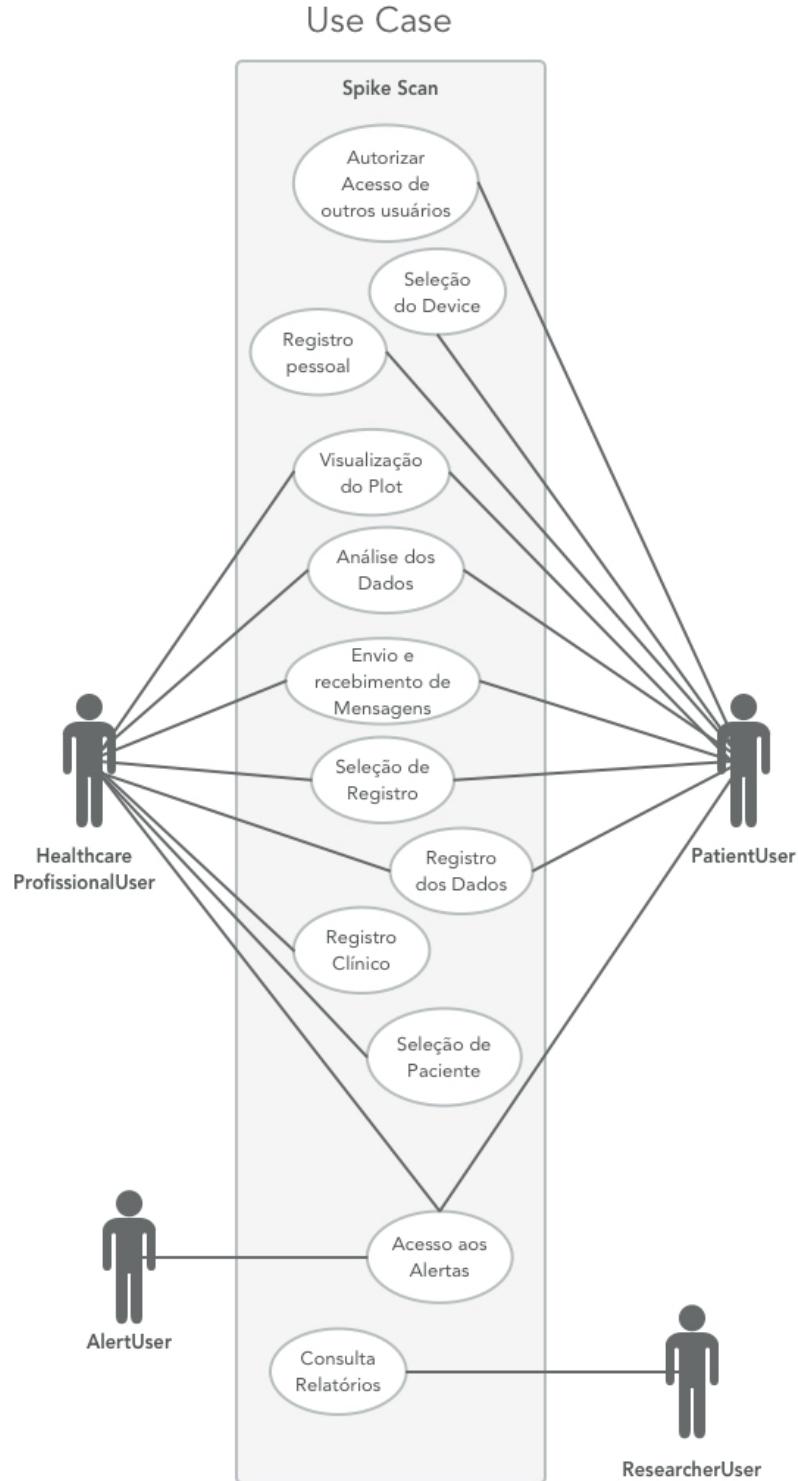
Análise de sinal offline descreve a aplicação de análises em dados previamente gravados pelo sistema, podendo ser manipulados pelos usuários *PatientUser* e *HealthcareProfessionalUser*.

Tabela 4.6 – Análise de Sinal offline

Título	Análise de sinal offline
Ator	<i>PatientUser, HealthcareProfessionalUser</i>
Cenário	1. Usuário logado seleciona arquivo com sinal desejado; 2. Sistema exibe arquivo selecionado; 3. Usuário seleciona opções de análise desejada dentre as disponíveis no sistema; 4. Sistema gera novo arquivo resultante da análise.
Extensões	Sistema não possui arquivo gravado. a) Usuário cancela solicitação.
Pré-condições	Arquivo de gravação de sinal aberto; Seleção do tipo de análise.
Pós-condições	Criação de novo arquivo com dados analisados.

4.1.5 Diagrama de Casos de Uso

Figura 4.4 – Diagrama de casos de uso.

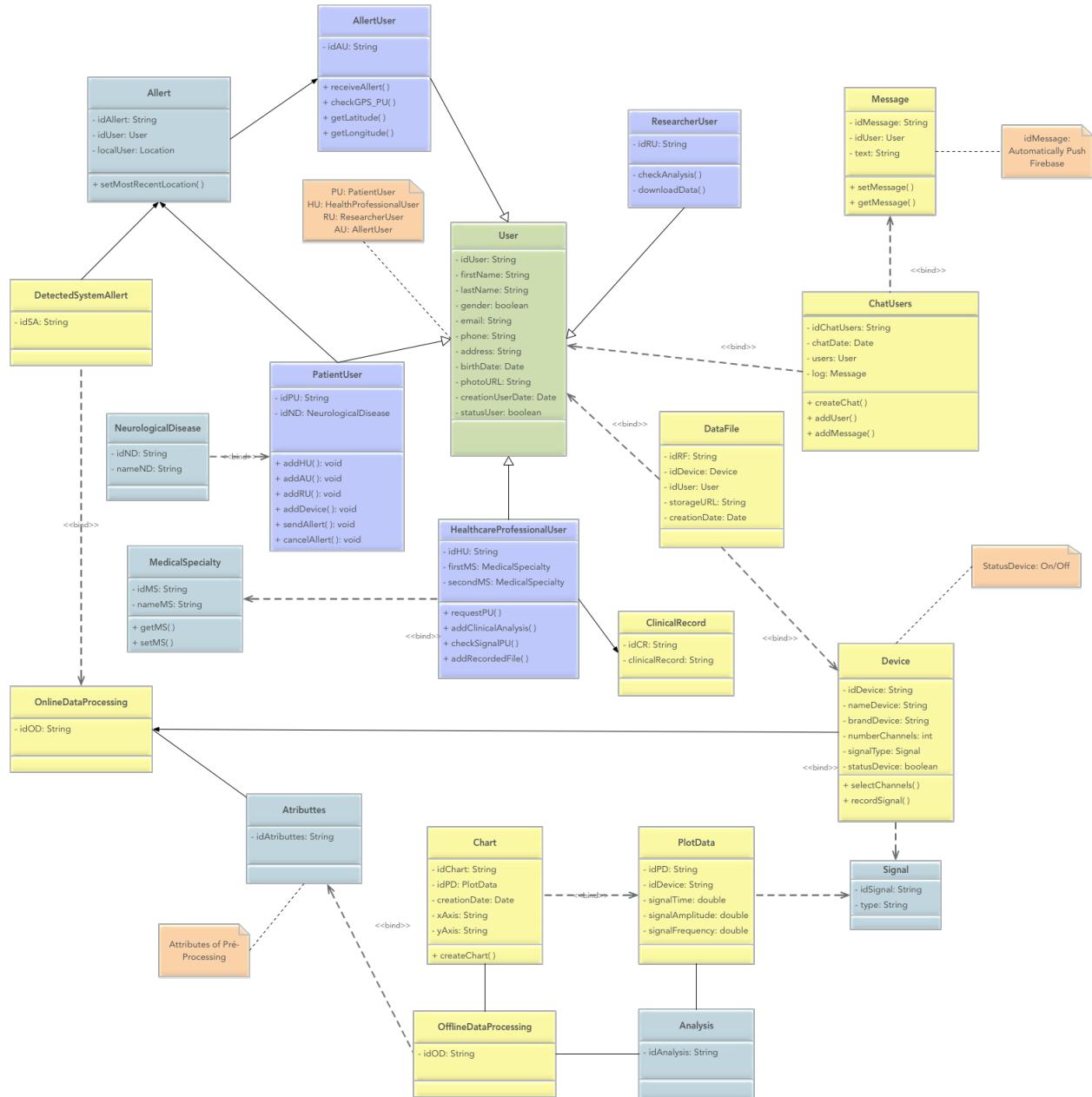


Fonte: Autoria Própria.

4.1.6 Diagrama de Classes

As classes especificam as propriedades do objeto a ser criado. É como uma planta, um *blueprint*, uma estrutura. Para criar-se um objeto, deve-se antes criar a classe. A estrutura da classe é feita para comportar os valores do objeto (identidade, atributos e métodos). A figura abaixo (*Figura 4.5*) contém a representação inicial do diagrama de classes para o desenvolvimento da aplicação *SpikeSan*, incluindo os conceitos comentados no capítulo 2: abstração, encapsulamento, herança e polimorfismo.

Figura 4.5 – Diagrama de classes do sistema *Spike Scan*: Em cor lilás estão as classes referentes aos usuários do sistema; em cor laranja as observações relacionadas as classes ligadas; em cor amarela as classes relacionadas aos dados; em cor cinza as classes que possivelmente utilizarão os dados para gerar relatórios; e em verde a classe com os atributos e métodos comuns aos usuários.



Fonte: Autoria Própria.

4.2 Design (*Object-Oriented Design*)

Nessa etapa são descritas as fases de *design* do sistema: ferramentas específicas, arquitetura do sinal, *templates* e fluxo de telas.

4.2.1 Ferramentas Específicas

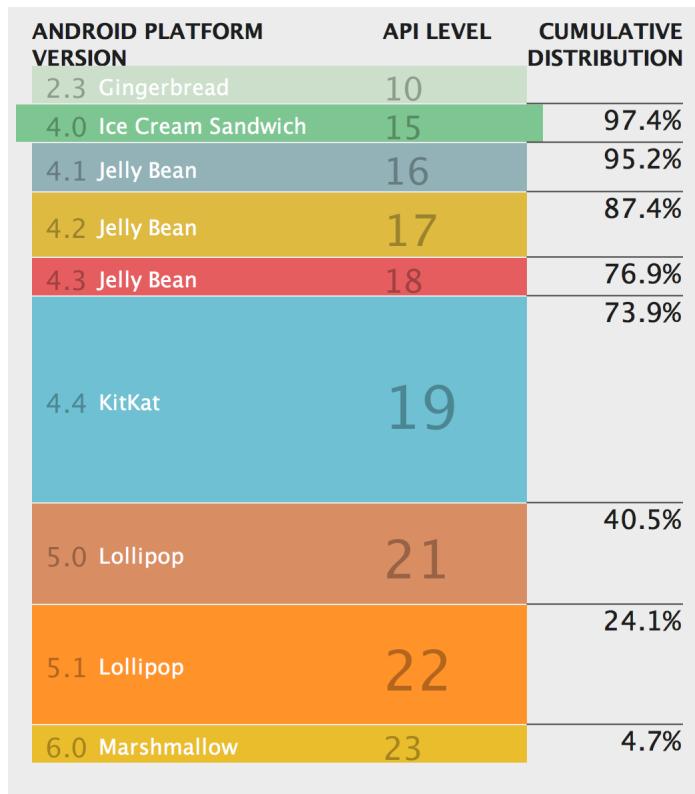
Como se trata do desenvolvimento de um sistema para monitoração remota, utilizamos o *Android* como exemplo para a implementação, por ser de fácil utilização e maior afinidade pelos usuários. Pelo fato do *Android* utilizar o *Java* como linguagem de programação. As ferramentas específicas necessárias para o desenvolvimento do sistema em questão, foram:

- *JDK* na versão *Java Standard Edition (SE) 8*;
- *Android Studio* versão 2.2.3;
- Versão 6.0, *API level 23, Marshmallow*;
- *Google APIs Intel x86 Atom System Image* (das duas *APIs*);
- *Sources for Android 23*;
- *Android SDK Build Tools*;
- *Android SDK Tools*;
- *Android SDK Platform-Tools*;
- *Documentation for Android SDK*;
- *Android Support Repository*;
- *Google Repository*;
- *Google USB Driver*;
- *Intel x86 Emulator Accelerator (HAXM)*, acelerador do emulador.

Ao instalar o *SDK*, é necessário escolher as versões de *target* e a de testes no emulador de versões anteriores desejadas. Caso o *target* seja a versão 23 (*Android 6.0 Marshmallow*), por exemplo, é preciso instalar o *SDK* da *API* 23. Caso necessite-se testar uma versão anterior, como a 19 (*Kit Kat*), é necessário instalar o *SDK* do *Android 4.4*, que é uma das versões anteriores presentes em um número significativo de dispositivos no mercado (neste caso, um terço dos dispositivos *Android* ativos (Maio/2016)).

Considerando que a adaptação de uma versão leva bastante tempo para solidificar nos aparelhos dos usuários finais, e que mais de 97% dos dispositivos *Android* possuem a versão 4.0 (*Ice Cream Sandwich*) instalada (Figura), a *API* mínima escolhida para o desenvolvimento desse projeto foi a de nível 15.

Figura 4.6 – Versões da *API* mais utilizadas pelos usuários. *Ice Cream Sandwich* com 97.4%.



Fonte:Android Studio

4.2.2 Arquitetura do Sinal

A estrutura para a transmissão do fluxo do sinal foi dividida em 4 fases: aquisição, pré-processamento, processamento e pós-processamento. Na primeira, temos o contato do paciente com o dispositivo de aquisição de dados, por exemplo um sinal de eletroencefalografia (EEG) utilizando um dispositivo externo, um banco de dados público validado, dados previamente gravados ou dados artificiais. Os sinais de EEG são pré-processados na etapa seguinte, onde o usuário tem a opção de monitorar *online* o dado transmitido pelo dispositivo portátil. Ainda nessa etapa, é possível também selecionar atributos de interesse referente ao sinal recebido, através de uma classificação pré-estabelecida que possibilite filtrar características importantes dependendo da necessidade do usuário e da aplicação escolhida. Na fase

de processamento, os sinais passam por uma análise mais refinada, de acordo com a necessidade da informação requerida pelo profissional da saúde. A última etapa é a de *feedback* das informações advindas dos dados processados pelo sistema aos usuários. O sinal de EEG original é captado e o formato de onda é exibido em tempo real para cada canal. Pode-se ainda visualizar informações mais detalhadas, como por exemplo a transformada rápida de Fourier (FFT) de cada canal, a intensidade de cada onda ($\text{delta}(\delta)$, $\text{theta}(\theta)$, $\text{alpha}(\alpha)$ e $\text{beta}(\beta)$), ou a posição do giroscópio nos dois eixos (x e y). Os dados exibidos podem ser gravados e armazenados (formato .edf ou .CSV) para posteriores análises ou visualizações no próprio sistema.

Figura 4.7 – Fases da transmissão do sinal eletrofisiológico no sistema: aquisição (dado de um dispositivo externo ou de um arquivo), pré-processamento (monitoramento online e seleção de atributos, processamento (análise do dado) e resposta (*feedback*) aos usuários).



Fonte: Autoria Própria.

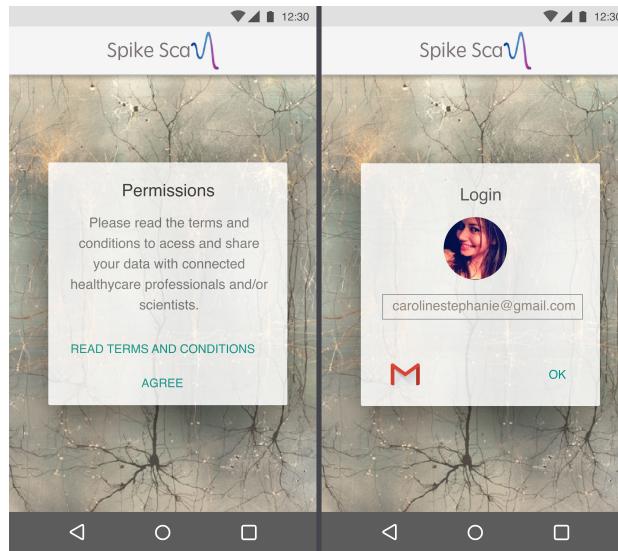
4.2.3 Templates

Nessa seção estão as últimas versões dos *templates* após o processo de *brainstorms*, entrevistas com possíveis usuários, etapas com post-its e rascunhos. Cada template simula o resultado final após a implementação completa do sistema.

4.2.3.1 Permissions e Login

A tela inicial do sistema consiste da informação indispensável sobre os termos e condições do sistema, o uso e compartilhamento dos dados em relação a todos os usuários envolvidos. O usuário *PatientUser* terá as opções: escolher entre não compartilhar as informações, portanto o uso dos dados pessoais fica sendo privado e de acesso único pelo próprio usuário; permitir o acesso à usuários *Healthcare User*, que seriam profissionais da saúde que acompanham o tratamento de sua condição específica; permitir o acesso dos dados pessoais à *Research User*, sendo a visibilidade de gráficos com estatísticas globais. O *Healthcare User* necessitará seguir os regulamentos tradicionais da relação entre profissional da saúde e pacientes vinculados, e o *Research User* poderá utilizar os dados estatísticos para fins de pesquisa científica.

Figura 4.8 – Tela Permissions e Login



Fonte: Autoria Própria.

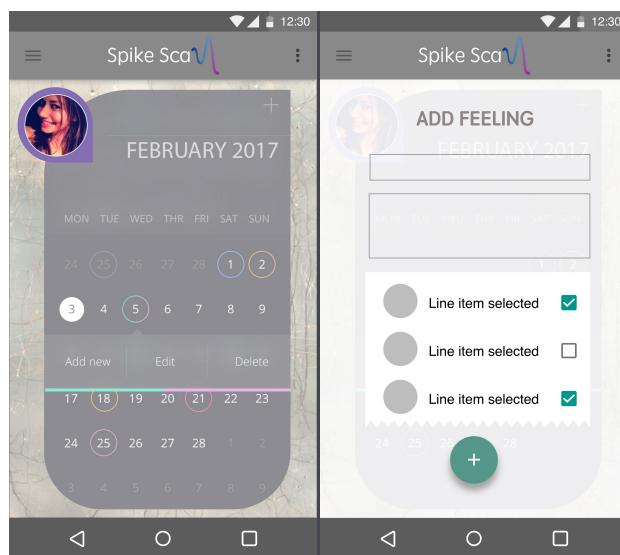
Após concordar com os termos e condições para uso do sistema, o usuário poderá se cadastrar e realizar o login no aplicativo. É possível logar no sistema, criar uma nova conta, ou utilizar um cadastro pré-existentes de outras contas como o login via autenticação do

Google utilizando o serviço *Authentication* do Firebase. O usuário terá que digitar sua conta pessoal do Gmail no campo especificado e confirmar clicando no botão *OK* na parte inferior direita da tela de login (Figura 4.8).

4.2.3.2 Home e Personal Report

Após a autenticação ser processada com sucesso, o *PatientUser* terá acesso a sua tela principal com o calendário atual e informações referentes a cada dia, sendo automaticamente detectado a data atual. As informações contidas podem ser por captação do sinal por dispositivos externos, relato pessoal do paciente, registro clínico de um profissional da saúde vinculado, exames, armazenamento de dados processados, análises de dados ou dados armazenados por arquivos externos. Ao clicar na data é possível verificar o que já está registrado, adicionar um novo registro, editar ou deletar. O relato pessoal do paciente pode ser adicionado por características pré-determinadas de acordo com sua condição específica e/ou descrição livre de fatos relevantes. O primeiro campo do relato é o título do registro, o segundo a descrição do registro, seguido pelos *checklists* de parâmetros pré-cadastrados e por fim o botão de adicionar o relato representado pelo sinal da adição na parte inferior central da tela (Figura 4.9). Ao clicar no botão de adicionar, o registro será adicionado no dia respectivo e poderá ser visualizado na tela Home, descrita anteriormente.

Figura 4.9 – Tela Home e Personal Report



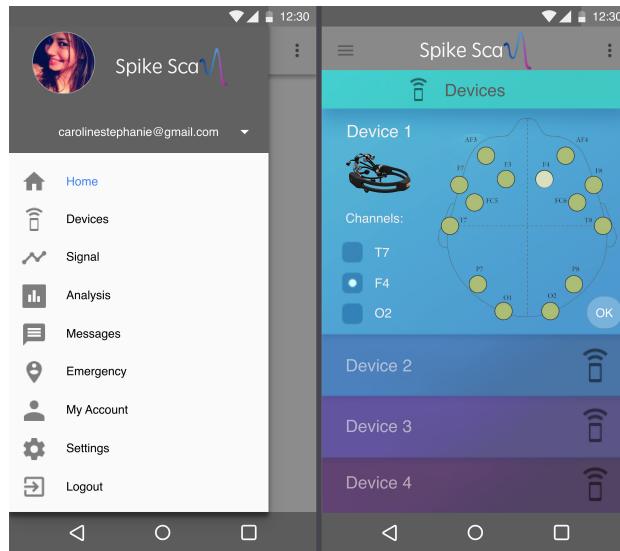
Fonte: Autoria Própria.

4.2.3.3 Menu e Devices

O menu principal é deslizante lateral, mais conhecido como *Navigation Drawer*. Possui uma barra superior informativa, *Navigation Header*, que contém a identificação do aplicativo e usuário. As opções no menu são: *Home*, *Devices*, *Signal*, *Analysis*, *Messages*, *Emergency*, *MyAccount*, *Settings* e *Logout*.

A tela de dispositivos terá uma lista dos possíveis *neuroheadsets* conectados e reconhecidos pelo sistema. Ao selecionar o item desejado, o usuário possui mais opções específicas referentes ao aparelho, como a imagem do dispositivo cadastrado, a opção de selecionar os canais disponíveis e a verificação da qualidade do sinal transmitido dos eletrodos reconhecidos. Ao clicar no botão *OK*, no canto inferior direito do campo reservado para o item selecionado, o sistema poderá captar os sinais utilizando o dispositivo escolhido (Figura 4.10).

Figura 4.10 – Tela Menu e Devices.



Fonte: Autoria Própria.

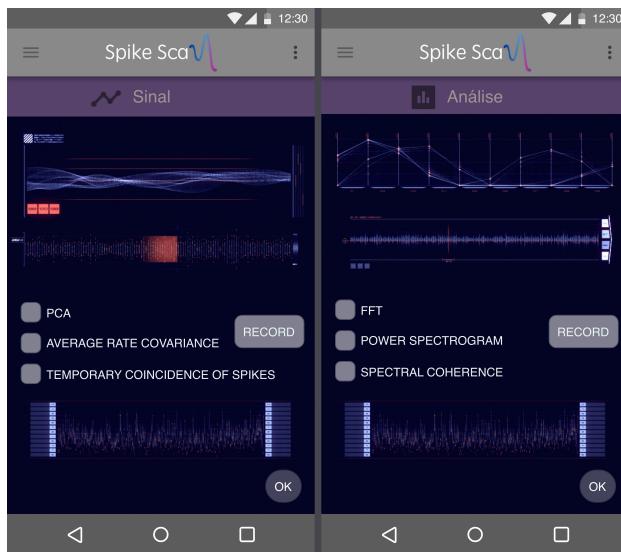
4.2.3.4 Sinal e Análise

Para simular o processamento do sinal foram utilizados dois grupos de dados (grupo A e B), o primeiro, que chamamos de "grupo A" é um arquivo com extensão *.csv* (*Comma-Separated Values*), tipicamente utilizado em armazenamento de dados e aceito pela maioria dos sistemas e o segundo, "grupo B" é um vetor de dados randômicos gerado pelo código. Na tela de Sinal, é possível visualizar o dado capturado pelo dispositivo selecionado *online*, com as opções de selecionar atributos de pré-processamento para filtrar os dados e captar informações relevantes de acordo com o interesse do usuário, minimizando ruídos e atributos

fora do interesse do usuário. Nessa tela será possível a seleção parcial de dados, visualização após o filtro e gravação do sinal desejado (Figura 4.11). Após a gravação, caso o usuário seja o paciente, ele pode solicitar por email a análise de um dado para o profissional da saúde vinculado.

A tela de análise foi desenvolvida para suprir as necessidades de tratamento dos dados já gravados *offline*, tanto os gravados pelo aplicativo utilizando um dispositivo externo, como por arquivos com dados apropriados carregados previamente de outros sistemas. É possível abrir o arquivo desejado, selecionar o conteúdo total ou parcial, escolher a análise de aplicação, visualizar o sinal após a análise e gravar do novo arquivo gerado. Após clicar no botão de confirmação *OK*, localizado na parte inferior direita da tela, o arquivo será armazenado no sistema para consultas posteriores (Figura 4.11).

Figura 4.11 – Tela Sinal e Análise



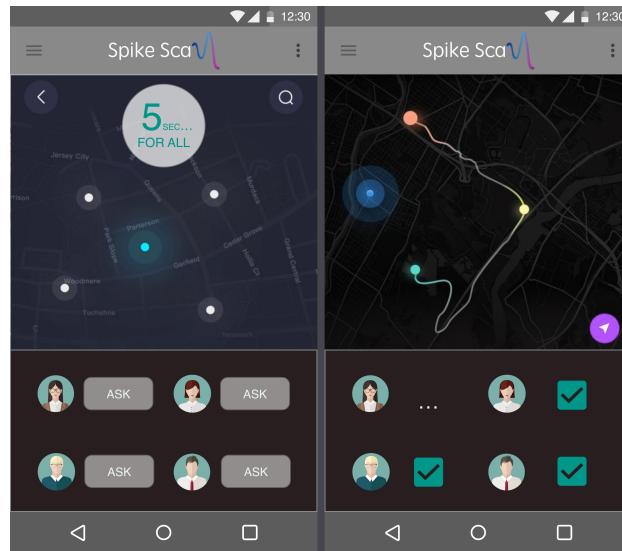
Fonte: Autoria Própria.

4.2.3.5 Emergência

A tela de emergência foi desenvolvida para envio de notificações caso os usuários *PatientUser* tenham alguma necessidade urgente. Caso o paciente selecione essa opção, aparecerá a localização dos usuários vinculados em relação à dele próprio. O sistema fornece um tempo curto para o paciente selecionar o envio do alerta à algum(uns) usuário(s) específico(s) através dos botões *ASK* em cada usuário detectado ou cancelar a solicitação. Caso a escolha não aconteça nesse período estipulado, o sistema enviará alerta para todos os usuários vinculados,

como familiares e amigos *AlertUsers* e profissionais da saúde *HealthcareUsers*, que por sua vez receberão a localização do paciente em relação à própria (Figura 4.12).

Figura 4.12 – Tela Solicitação de Emergência



Fonte: Autoria Própria.

4.2.3.6 Mensagens e Pesquisador

A tela de mensagens é o meio de comunicação remoto e formal em que os usuários vinculados poderão interagir. Foi utilizado o serviço *Realtime Database* do *Firebase* para a transmissão instantânea, sincronização e persistência dos dados no servidor.

Figura 4.13 – Tela Mensagens



Fonte: Autoria Própria.

O pesquisador *ResearchUser*, quando autorizado, terá acesso à informações estatísticas gerais em relação à desordens neurológicas desejadas de acordo com a localidade escolhida. As informações serão detectadas no *GrowDatabase* do sistema e exibidas na tela de acordo com os atributos informados pelo usuário (Figura 4.13).

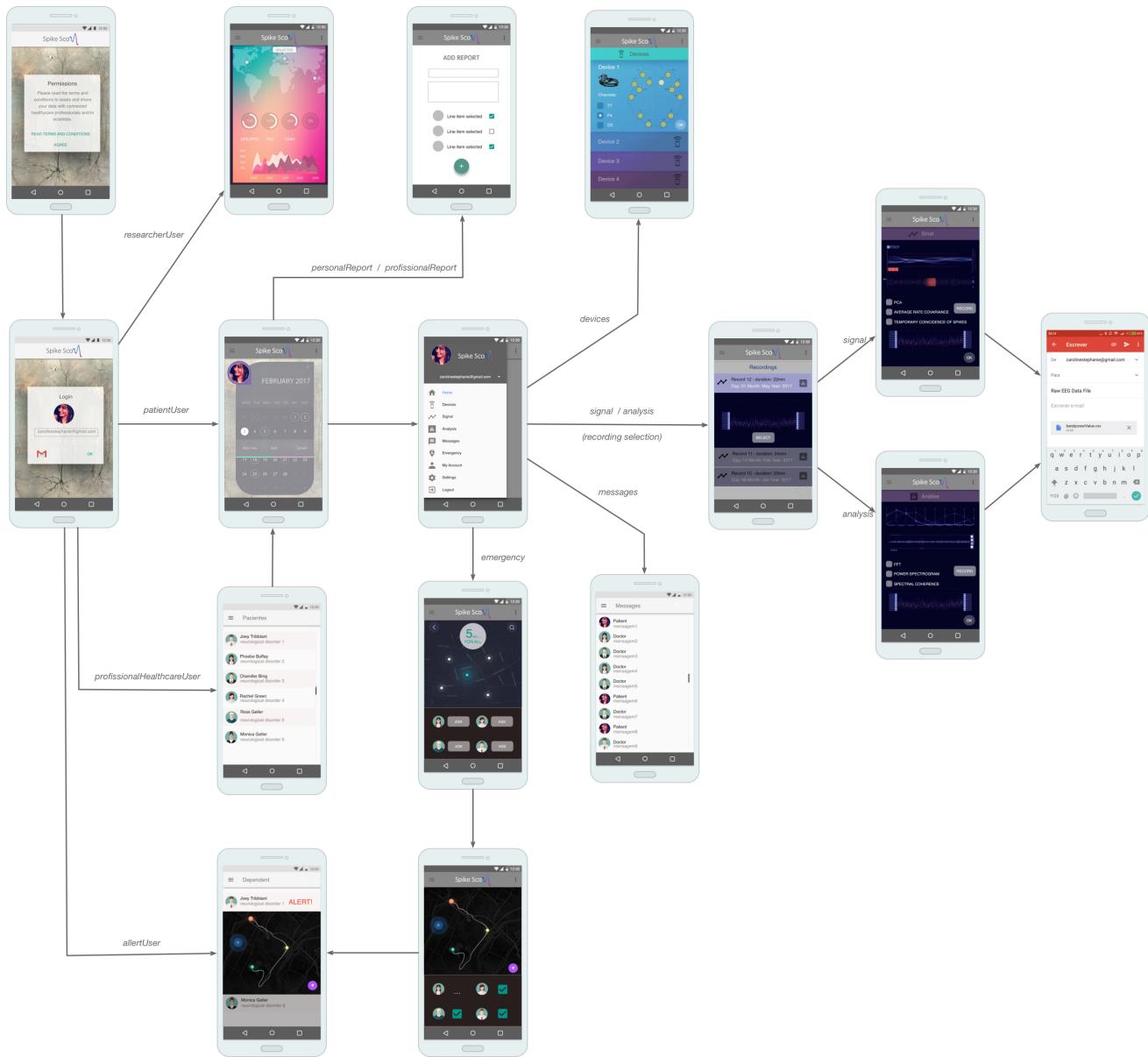
4.2.4 Fluxo de Telas

O fluxo de telas exibe a relação da navegação do sistema referente a cada usuário. Inicialmente é solicitado a autorização dos termos de uso para o usuário realizar o cadastro no sistema. O *login* do usuário é feito posteriormente e o sistema detecta o tipo de vínculo: "pesquisador"(*researcherUser*), "paciente"(*patientUser*), "profissional da saúde"(*professionalHealthcareUser*) ou "alertas"(*allertUsers*). Caso seja "pesquisador", será redirecionado para a tela de estatísticas gerais gerados pelo sistema. Caso seja o "paciente", o mesmo será conduzido para a tela inicial com o calendário atualizado com os registros destacados em cada dia. Caso seja o profissional da saúde, primeiramente será exibido a tela de seleção do paciente para posteriormente ser direcionado para a tela inicial dos registros pessoais e clínicos. Caso seja um usuário "alerta", a tela seguinte será da lista de pacientes vinculados e do mapa com a localização do usuário paciente quando ativar a emergência.

A partir do menu principal, é possível realizar selecionar a tela para detecção de *devices* externos, sinais, análises, mensagens e emergência. No caso das telas de sinais e análises, é

necessário antes selecionar o sinal desejado para a manipulação. Posteriormente é possível enviar o dado por *email* para os usuários.

Figura 4.14 – Fluxo da relação entre as telas do sistema *Spike Scan*



Fonte: Autoria Própria.

4.3 Implementação (Object-Oriented Programming)

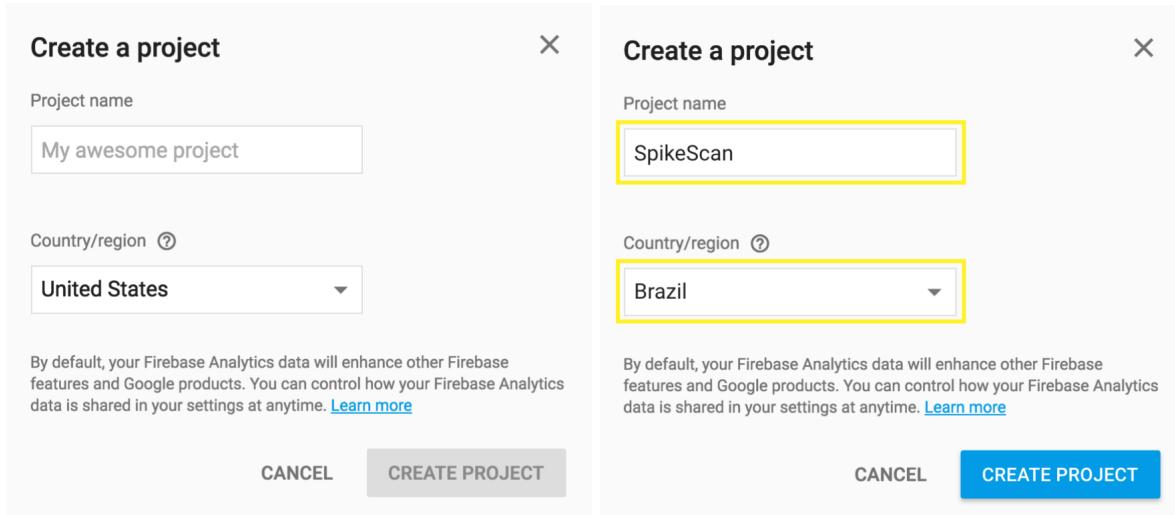
Nessa etapa são descritas as fases de implementação do sistema: conexão do *Android* com *Firebase*, autenticação do usuário, seleção do paciente, menu principal, seleção de registros, interação entre usuários, visualização do sinal registrado em tempo real, gravação dos dados gerados, transferência do registro do sinal, consulta do envio do sinal e visualização do arquivo.

4.3.1 Conexão do Android Studio com o Firebase

Neste item descreve-se a criação do banco de dados para vincular os dados transmitidos do aplicativo. Foi necessário os seguintes passos:

- Criar conta no "firebase.google.com" e ir para o console;
- Criar novo projeto;
- Preencher o campo "Projeto" com o mesmo nome criado no Projeto no Android Studio. Nesse caso utilizamos "SpikeScan" e no campo abaixo, na mesma janela, selecionar seu país (*Figura4.18*);

Figura 4.15 – Criar Projeto Firebase



Fonte: <<https://console.firebaseio.google.com>>

- No Dashboard, clicar no ícone "Add Firebase to Android" (*Figura4.19*).

Figura 4.16 – Selecionar Android

Fonte: <<https://console.firebaseio.google.com>>

- Na janela seguinte, preencher o primeiro campo com o nome do pacote igual ao criado no Android Studio (Figura 4.20);

Figura 4.17 – Inserir Pacote e SHA-1

The image displays two side-by-side screenshots of the 'Add Firebase to your Android app' wizard. Both screenshots are at step 1: 'Enter app details'.
 Left Screenshot:
 - Package name: `com.yourapp.android`
 - App nickname (optional): `Freemium Android App`
 - Debug signing certificate SHA-1 (optional): `00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00`
 Right Screenshot:
 - Package name: `com.spikescan.spikescan`
 - App nickname (optional): `Freemium Android App`
 - Debug signing certificate SHA-1 (optional): `E9:27:26:D6:B2:1D:ED:69:9B:F0:26:A8:1F:30:AD:C9:8C:03:4C:68`

Fonte: <<https://console.firebaseio.google.com>>

O segundo campo é opcional, caso deseje-se um nome diferente do projeto para a interface do aplicativo. O último campo do código SHA-1 é necessário para acessar alguns serviços como *links*, *invites*, ou *Google Sign-in*, fornecidos pelo *Firebase*. O número precisa ser gerado em cada máquina e tem algumas formas de identificar. Uma delas é abrir seu projeto no Android Studio, na janela do "Gradle Console" (painel lateral) -> "Tasks" -> "android" -> "Sign in Report" e irá aparecer o SHA1 (Figura 4.21).

Figura 4.18 – Localização do SHA-1

```

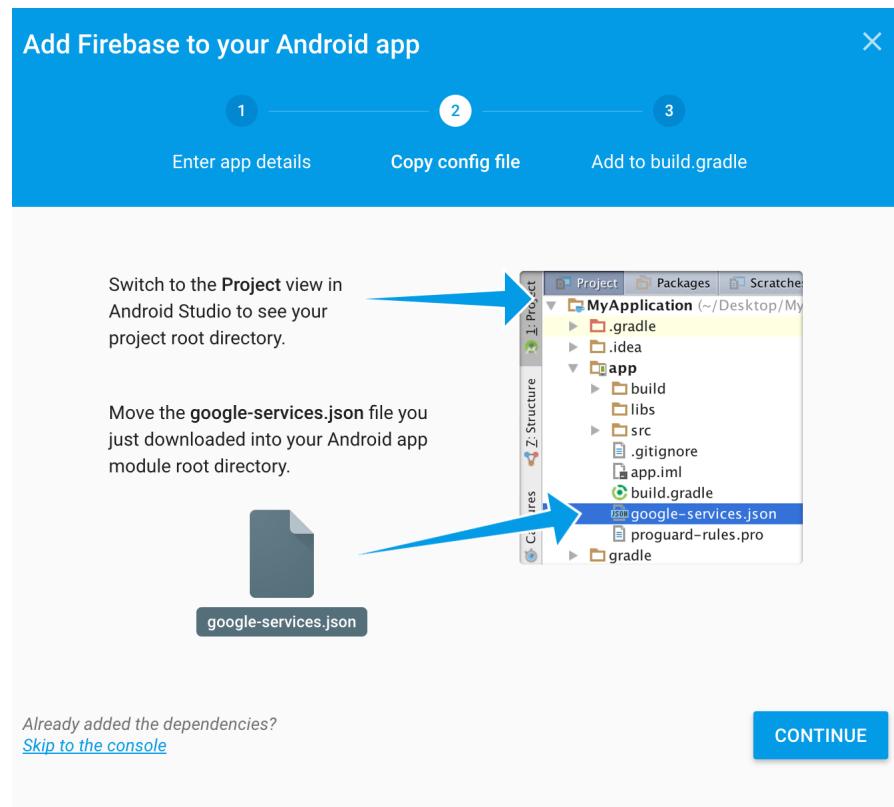
Variant: debug
Config: debug
Store: /Users/stephanie/.android/debug.keystore
Alias: AndroidDebugKey
MD5: 3E:B3:87:3A:07:77:70:B2:1D:DF:74:DF:1C:7B:1C:2F
SHA1: E9:27:26:D6:B2:1D:ED:69:9B:F0:26:A8:1F:30:AD:C9:8C:03:4C:68
Valid until: Thursday, October 11, 2046

```

Fonte: Android Studio Console

- Ao seguir para o próximo passo, automaticamente será gerado o arquivo de configurações JSON (google-services.json), adicione na pasta "app" do seu projeto do AndroidStudio (*Figura 4.22*);

Figura 4.19 – Salvar Arquivo JSON

Fonte: <https://console.firebaseio.google.com>

- No passo seguinte, será necessário adicionar a dependência e o plugin do *Firebase*

no Android Studio. Para a dependência será necessário abrir o arquivo do Gradle no nível de Projeto (<project>/build.gradle) e acrescentar a seguinte linha de comando *(Figura 4.23)*:

Figura 4.20 – Adicionar Dependência

```
buildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:3.0.0'
    }
}
```

Fonte: Android Studio Console

- Para o plugin, abra o arquivo do Gradle no nível de App <project>/<app-module>/build.gradle e acrescente a seguinte linha de comando *(Figura 4.24)*:

Figura 4.21 – Adicionar Plugin

```
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Fonte: Android Studio Console

- Para finalizar, clique no botão se sincronizar o *Gradle* no Android Studio para atualizar as informações inseridas *(Figura 4.25)*.

Figura 4.22 – Adicionar Plugin



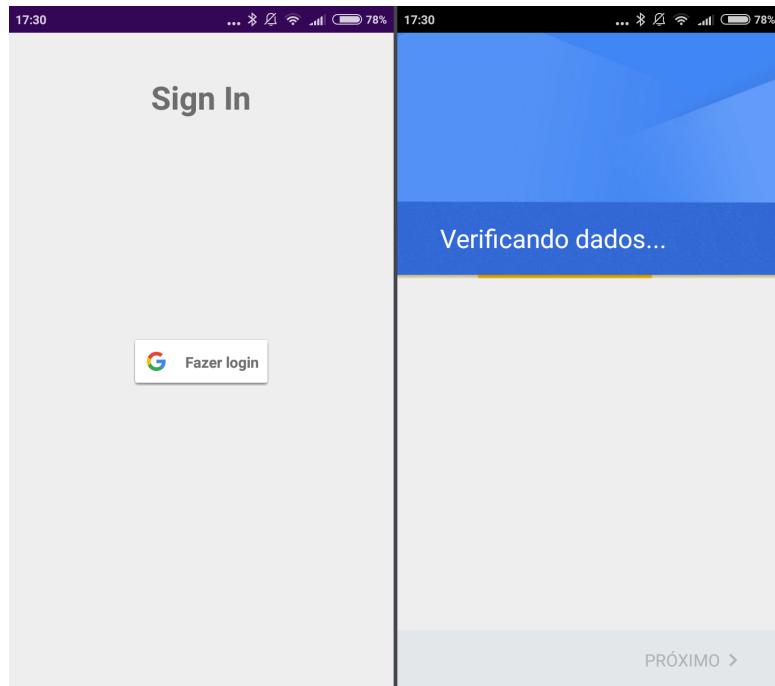
Fonte: Android Studio Console

4.3.2 Autenticação do Usuário

A autenticação do usuário foi feita utilizando o serviço "Authentication" do *Firebase*. Foi implementada a tela "*activity_sign_in.xml*", com os elementos de uma *label* com identificação *Sign In*, um botão para login pelo *Gmail* (arquivo do código relacionado: "*SignInAc-*

tivity.java". É verificado pelo *Google* se já existe uma prévia autorização dos dados de *login* e senha no dispositivo para o acesso imediato (Figura 4.23).

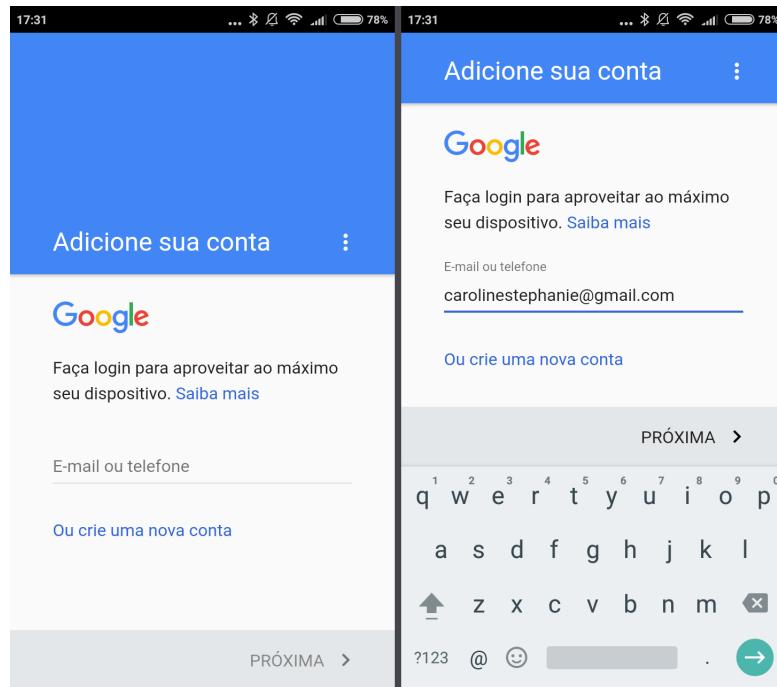
Figura 4.23 – Telas de autenticação do usuário: *login* e verificação de autorização prévia dos dados de acesso (usuário e senha).



Fonte: Autoria Própria.

Em seguida, caso os dados de usuário e senha não estejam previamente autorizados, o sistema exibe a tela do *Google* com o campo para digitação do *login* (*email* ou telefone) (Figura 4.24).

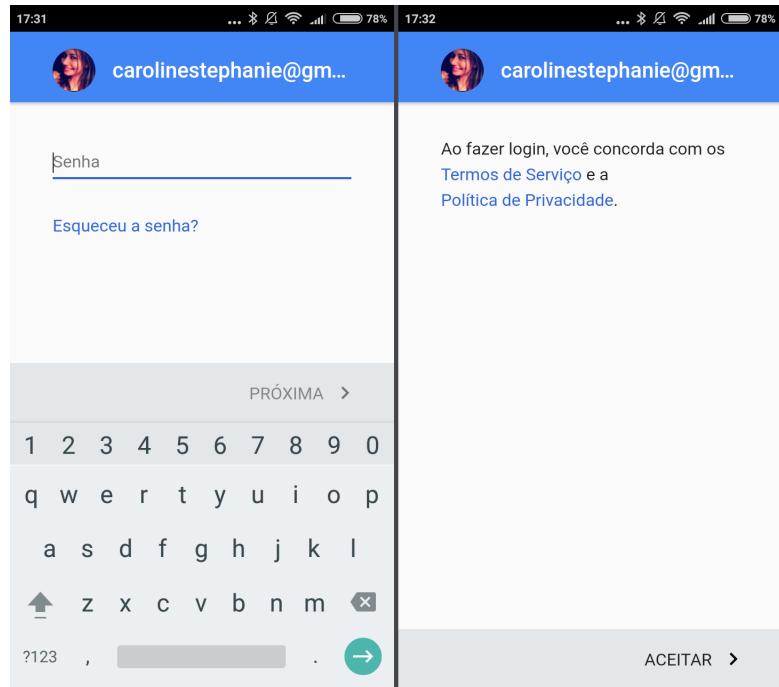
Figura 4.24 – Telas de autenticação do usuário: inserção de *email* ou telefone e preenchimento do email como exemplo.



Fonte: Autoria Própria.

Em seguida, é solicitada a senha do usuário e posteriormente a tela de confirmação com os termos de uso (Figura 4.25).

Figura 4.25 – Telas autenticação do usuário:

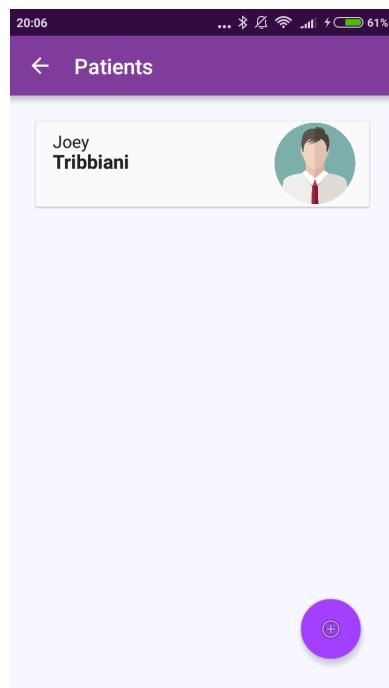


Fonte: Autoria Própria.

4.3.3 Seleção de Pacientes

Quando o usuário *healthcareProfessionalUser* inicia o tratamento de um paciente que utiliza o sistema, ele solicita autorização do vínculo para o usuário paciente (*patientUser*), e por sua vez, quando aprovado, o *patientUser* é adicionado à lista. Assim, o profissional da saúde pode selecionar o paciente desejado, consultar e registrar informações necessárias (Figura 4.26).

Figura 4.26 – Lista de pacientes vinculados ao usuário profissional da saúde

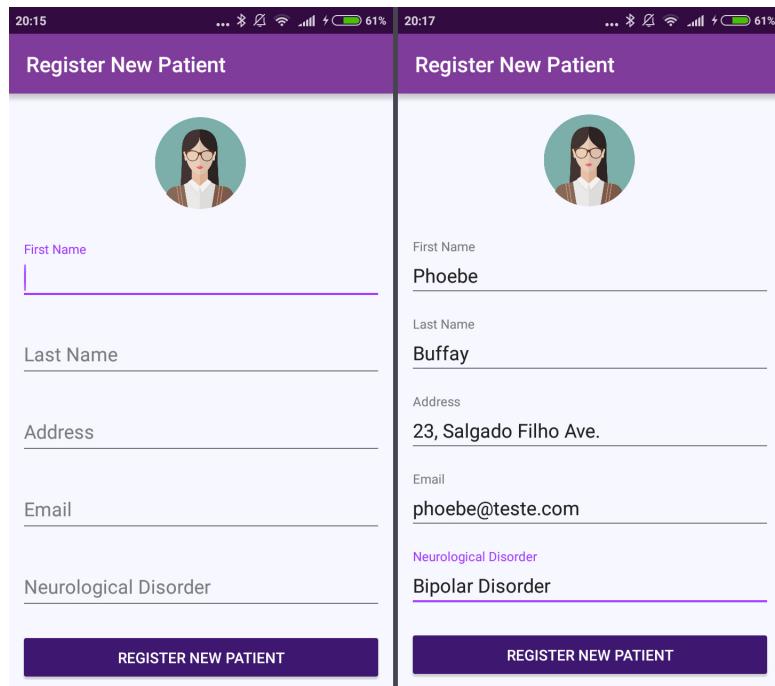


Fonte: Autoria Própria.

4.3.4 Cadastro do Paciente

O usuário *healthcareProfessionalUser* pode adicionar e editar os pacientes vinculados (Figura 4.27).

Figura 4.27 – Cadastro de Paciente

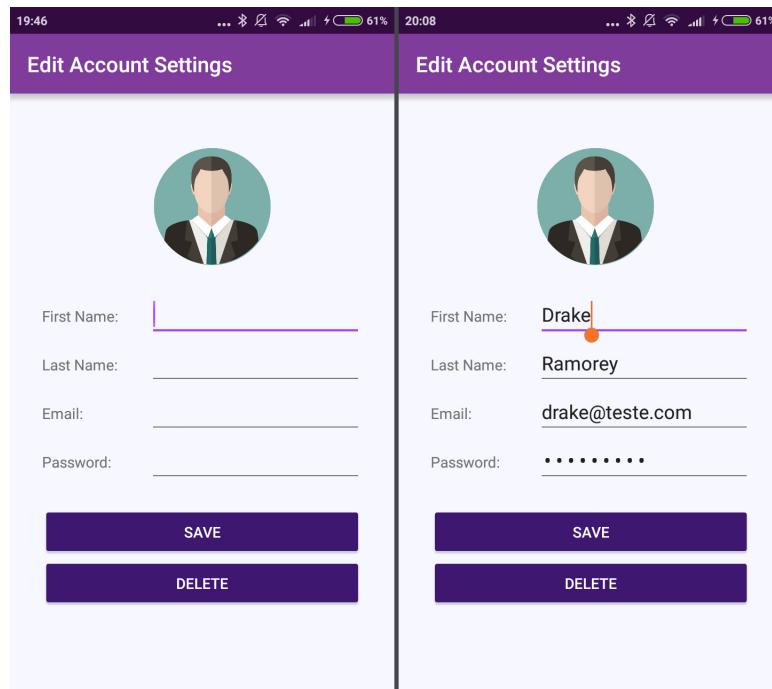


Fonte: Autoria Própria.

4.3.5 Cadastro do Profissional de Saúde

As telas abaixo representam a edição de campos registrados pelo usuário: First Name, Last Name, email e password.

Figura 4.28 – Edição de cadastro de profissional da saúde

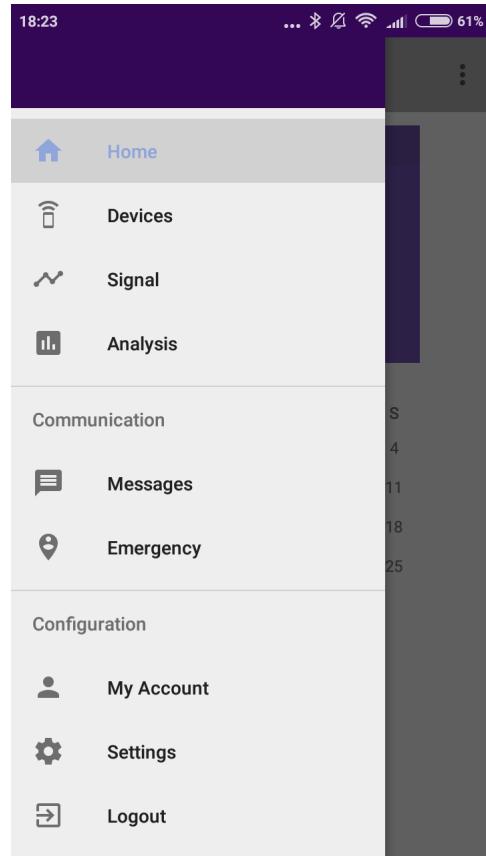


Fonte: Autoria Própria.

4.3.6 Implementação do Menu Principal

O menu principal é deslizante lateral, utilizando o *Navigation Drawer*, como definido no template. As opções no menu foram agrupadas em principais (*Home*, *Devices*, *Signal*, *Analysis*), Comunicação (*Messages*, *Emergency*) e de Configurações (*MyAccount*, *Settings* e *Logout*) (Figura 4.29). Passos para criar um *Navigation Drawer* podem ser consultados na página oficial do *Android* (<https://developer.android.com/training/implementing-navigation/nav-drawer.html>).

Figura 4.29 – Menu Principal: *Home*, *Devices*, *Signal*, *Analysis*, *Messages*, *Emergency*, *My Account*, *Settings*.

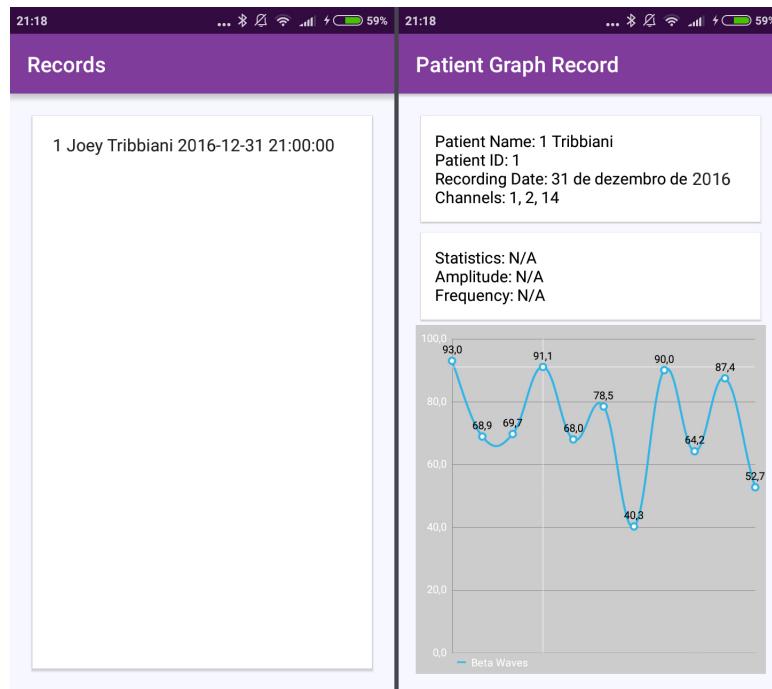


Fonte: Autoria Própria.

4.3.7 Seleção de Registros

O sistema grava os arquivos dos dados gerados em forma de lista. Ao selecionar o arquivo, é aberto o gráfico dinâmico e o conteúdo resumido da gravação (Figura 4.30)

Figura 4.30 – Lista de Gravações Arquivadas

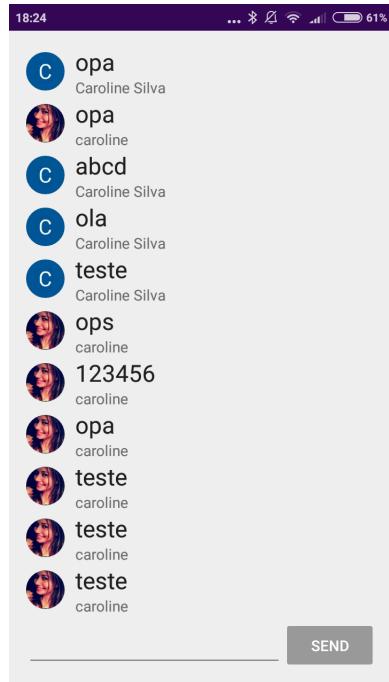


Fonte: Autoria Própria.

4.3.8 Interação entre Usuários

A tela de interação entre usuários permite o envio e recebimento de mensagens instantâneas, sincronizadas e armazenadas na nuvem utilizando o *Google Cloud* com os envolvidos no monitoramento do tratamento do paciente.

Figura 4.31 – Exemplo de troca de mensagens entre usuários conectados no sistema vinculados ao acompanhamento remoto de paciente.

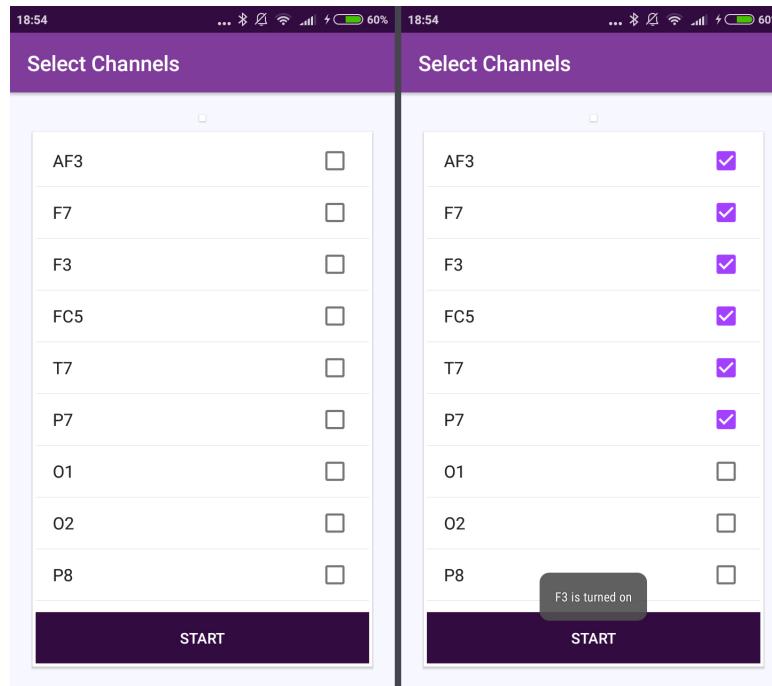


Fonte: Autoria Própria.

4.3.9 Seleção de Canais

Tela para seleção de canais de eletroencefalografia (EEG), para uso com dados artificiais (gerados pelo sistema). Nesse caso, foi utilizado o sistema internacional 10-20 para identificação dos canais ($A1, A2, T3, T4, T5, T6, O1, O2, P3, Pz, P4, C3, Cz, C4, F3, Fz, F4, F7, F8, Fp1$ e $Fp2$) (Figura 4.32).

Figura 4.32 – Seleção dos canais desejados pelo usuário referentes ao sistema internacional 10-20.

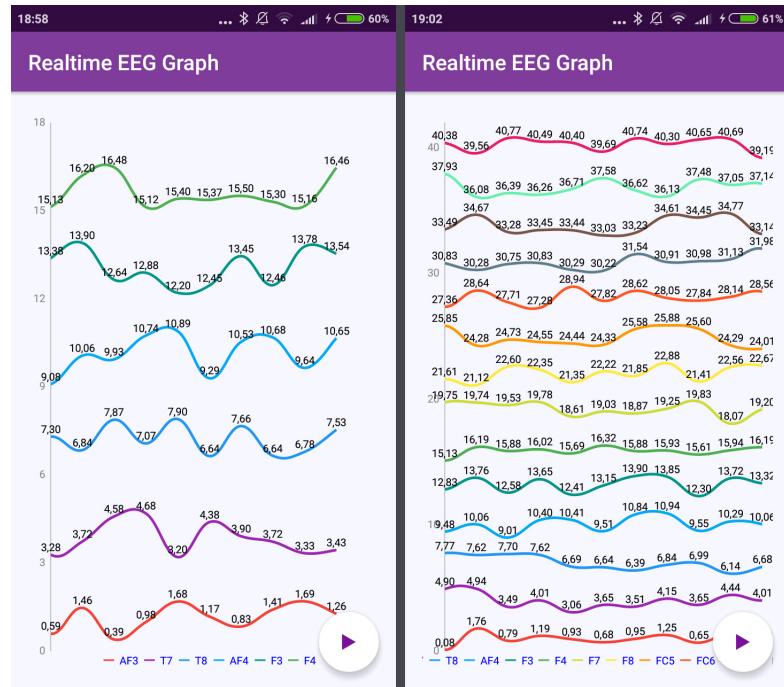


Fonte: Autoria Própria.

4.3.10 Visualização do Sinal em Tempo Real

Após a seleção dos canais desejados, o sistema gera um grupo de dados e simula o *plot* com os dados artificiais com os canais desejados pelo usuário (Figura 4.33).

Figura 4.33 – Visualização dos sinais artificiais nos canais selecionados.

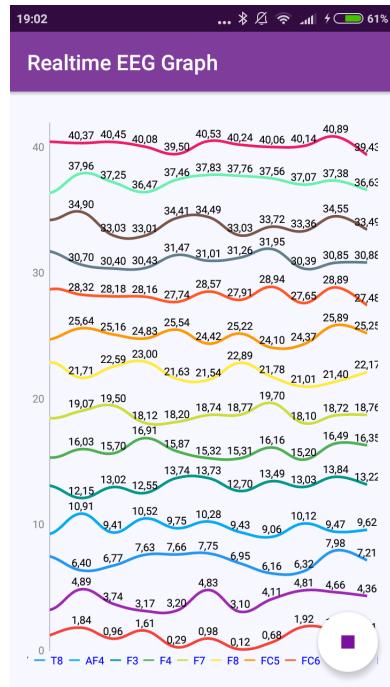


Fonte: Autoria Própria.

4.3.11 Gravação dos Sinais

Nesse momento, é possível a gravação dos dados artificiais plotados pressionando o botão inferior direito da tela, modificando o ícone típico de *play* para o *stop* (Figura 4.34), identificando o início da gravação dos dados. O usuário pode observar a continuidade da gravação visualmente até decidir parar a gravação apertando no mesmo botão.

Figura 4.34 – Gravação dos Sinais por Visualização de Canais

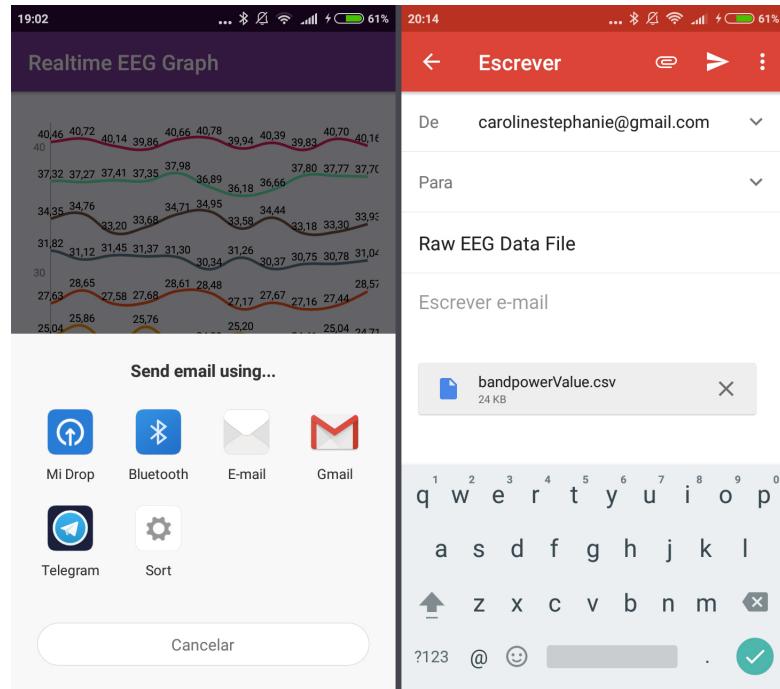


Fonte: Autoria Própria.

4.3.12 Envio da Gravação por Email

Ao confirmar a parada da gravação, o sistema oferecerá ao usuário a opção de envio por *email* do arquivo com extensão *.csv* que contém os dados gravados, conforme a *(Figura 4.35)*. Caso o usuário decida enviar, a tela seguinte (*(Figura 4.36)*) aparece solicitando o endereço do destinatário já com o arquivo em anexo *bandpowerValue.csv* e com um assunto do e-mail preenchido por padrão *Raw EEG Data File*.

Figura 4.35 – Envio da Gravação dos dados por Email



Fonte: Autoria Própria.

4.3.13 Visualização do arquivo enviado

O usuário receberá o arquivo por *email* com os dados articiais gravados, sendo exibidos em um arquivo com extensão ".csv" com duas colunas, a primeira com os canais selecionados e a segunda com seus respectivos dados (Figura 4.36).

Figura 4.36 – Visualização do arquivo com extensão ".csv" com os dados artificiais gravados e enviados pelo usuário.

The screenshot shows a CSV file titled "bandpowerValue.csv" with 26 rows of data. The columns are labeled "A" and "B". Column A contains channel names and numbers, and column B contains numerical values. The data is as follows:

	A	B
1	Channel	Theta
2	AF3	0.2856781159
3	T7	3.949505634
4	T8	7.934334682
5	AF4	10.2011306
6	F3	13.2693655
7	F4	16.17638444
8	F7	18.14139538
9	F8	21.85180087
10	FC5	25.76918662
11	FC6	28.66801766
12	P7	30.74108232
13	P8	34.73175524
14	O1	36.61186568
15	O2	40.48856318
16	AF3	0.283851448
17	T7	4.062433439
18	T8	7.77552416
19	AF4	9.093884977
20	F3	12.60366312
21	F4	15.20976751
22	F7	18.88448239
23	F8	21.35276633
24	FC5	25.72593148
25	FC6	27.7446186
26	P7	31.64829221

Fonte: Autoria Própria.

4.4 Validação do Sinal

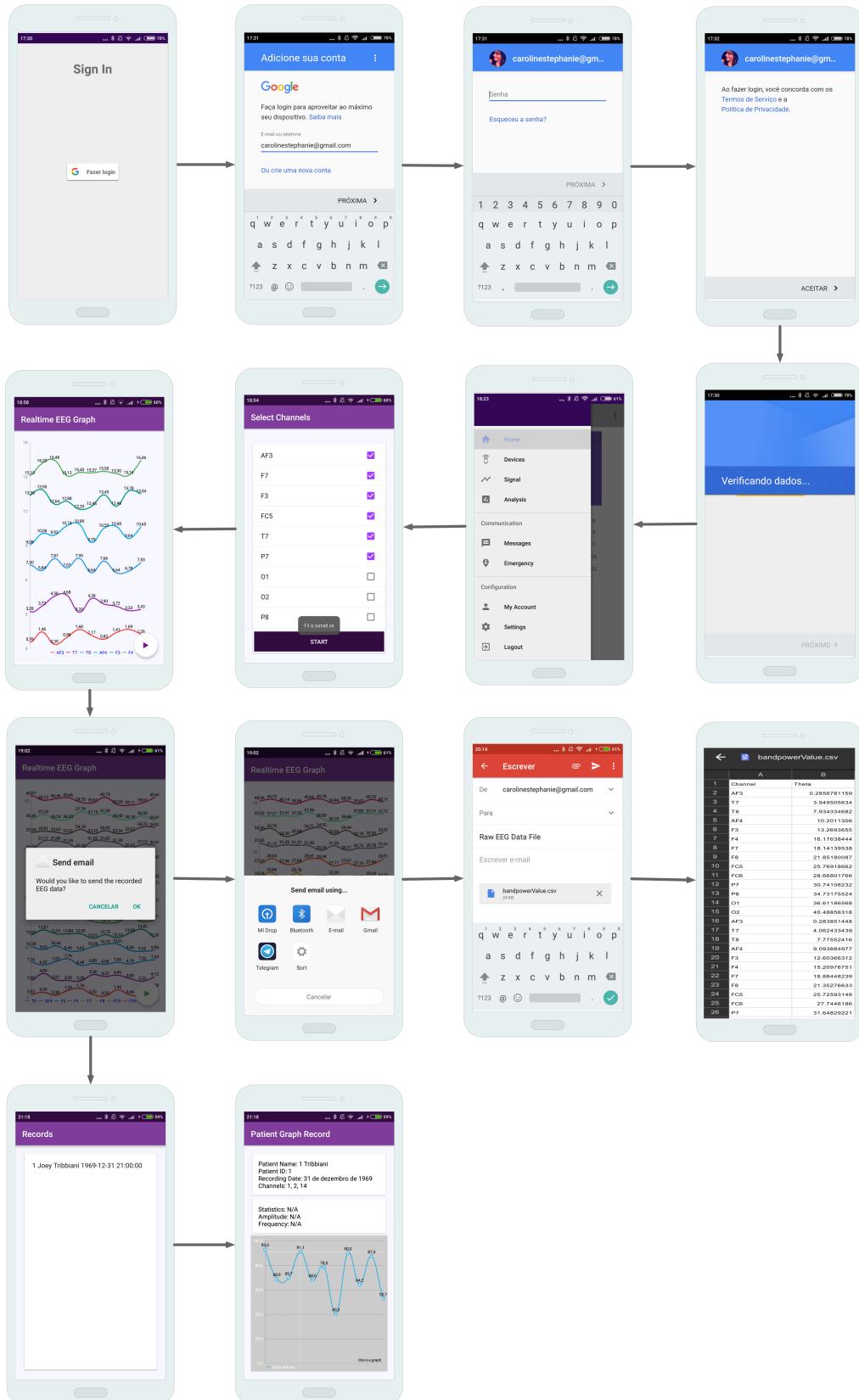
Para exemplo de validação da transmissão dos dados artificiais (gerados pelo sistema), pode ser seguido a seguinte sequência das telas implementadas:

1. Autenticação do usuário;
2. *Menu Principal*;
3. *Signals*;
4. Seleção de canais;
5. Visualização do *plot* no gráfico;
6. Gravação dos dados gerados;
7. Envio do dado gravado por *email*;
8. Visualização do arquivo no gráfico.

No exemplo, o usuário realiza o login no sistema *Spike Scan*, passando pelas telas de inserção de dados de *login* e senha, aceitação dos termos de uso. Posteriormente é acessado o item *Signal* através do *menu principal*, onde possibilita primeiramente a seleção dos canais e posteriormente a visualização do *plot* no gráfico em tempo real e a gravação desses dados gerados. Ao solicitar a gravação, o sistema *Spike Scan* oferece também o envio dos dados por *email*. O fluxo da condução desses passos está exemplificado na imagem abaixo (Figura 4.37).

Nesse capítulo foram apresentados as etapas executadas para o planejamento e o desenvolvimento do sistema, finalizando o conteúdo desse trabalho. Em seguida estão descritas as conclusões gerais e possíveis trabalhos futuros.

Figura 4.37 – Validação do sinal com os dados artificiais gravados e enviados pelo usuário.



Fonte: Autoria Própria.

Conclusão e Trabalhos Futuros

O sistema teve o propósito de diminuir o tempo de espera na comunicação entre os pacientes e os profissionais da saúde, de manter um histórico rico e atualizado de informações particulares a cada condição, possibilitar fácil acesso aos dados de forma confiável, gerar um aumento da qualidade de vida de pacientes e o fornecer dados específicos para pesquisas.

Foram realizados o planejamento e o desenvolvimento de um sistema computacional para processar e transmitir sinais eletrofisiológicos, com finalidade de monitoramento remoto de pacientes com desordens neurológicas por profissionais da saúde. Como resultado desse trabalho, foram implementados da conexão do *Android* com o *Firebase*, a autenticação dos usuários, a seleção de pacientes, a implementação do *menu* principal do sistema, a listagem dos registros dos dados do paciente (possibilitando a seleção e a visualização do resumo do registro), a interação entre os usuários vinculados, a visualização do sinal registrado em tempo real através do *plot* dos dados em gráfico (com opção de filtro dos canais), a gravação dos dados gerados, a transferência do registro do sinal, a consulta do envio do sinal e a visualização do arquivo enviado.

Como trabalhos futuros é sugerido a elaboração dos termos de uso voltados para a segurança dos dados do paciente seguindo as leis de privacidade da informação como por exemplo a *Health Insurance Portability and Accountability Act of 1996 (HIPAA)*, a implementação de estatísticas pelo usuário pesquisador (*researcherUser*), registro clínico com anexo aos exames, registro pessoal com opções de atributos pré-estabelecidos pela desordem neurológica específica, o uso de *SDK's* de dispositivos de captação de sinais compatíveis para integrar ao sistema, envio de alertas, implementação de diferentes análises, o uso do serviço *Firebase Analytics* para mensurar a efetividade do uso dos dados e testes em clínicas com profissionais e pacientes que já possuem acompanhamento presencial.

Referências

- AUERSWALD, P. Healthcare in the home. *Nursing management*, v. 39, n. 12, p. 1–36, 2015. ISSN 15388670. Citado na página 7.
- BEN-ARI, M. *Principles of Concurrent and Distributed Programming (2Nd Edition) (Prentice-Hall International Series in Computer Science)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006. ISBN 032131283X. Citado na página 4.
- BLABE, C. H.; GILJA, V.; CHESTEK, C. a.; SHENOY, K. V.; ANDERSON, K. D.; HENDERSON, J. M. Assessment of brain-machine interfaces from the perspective of people with paralysis. *Journal of neural engineering*, IOP Publishing, v. 12, n. 4, p. 043002, 2015. ISSN 1741-2552. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/26169880>>. Citado na página 4.
- BUZSÁKI, G.; ANASTASSIOU, C. a.; KOCH, C. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature reviews. Neuroscience*, Nature Publishing Group, v. 13, n. 6, p. 407–20, 2012. ISSN 1471-0048. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/22595786>>. Citado na página 11.
- CHAPIN, J. K.; MOXON, K. a.; MARKOWITZ, R. S.; NICOLELIS, M. a. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience*, v. 2, n. 7, p. 664–670, 1999. ISSN 1097-6256. Citado na página 2.
- CHAURASIYA, R. K.; LONDHE, N. D.; GHOSH, S. An efficient P300 speller system for Brain-Computer Interface. *Proceedings of 2015 International Conference on Signal Processing, Computing and Control, ISPCC 2015*, p. 57–62, 2016. Citado na página 2.
- Comitê Gestor da Internet no Brasil. *Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros*. [S.l.: s.n.], 2015. 424 p. ISBN 9788555590313. Citado na página 5.
- CURADO, M. R.; COSSIO, E. G.; BROETZ, D.; AGOSTINI, M.; CHO, W.; BRASIL, F. L.; YILMAZ, O.; LIBERATI, G.; LEPSKI, G.; BIRBAUMER, N.; RAMOS-MURGUIALDAY, A. Residual upper arm motor function primes innervation of paretic forearm muscles in chronic stroke after Brain-Machine interface (BMI) training. *PLoS ONE*, v. 10, n. 10, p. 1–17, 2015. ISSN 19326203. Citado 2 vezes nas páginas 2 e 14.
- DRONGELEN, W. van. Signal Processing for Neuroscientists References. *Signal Processing for Neuroscientists: An Introduction to the Analysis of Physiological Signals*, n. 1975, p. 297–300, 2011. Disponível em: <<http://www.sciencedirect.com/science/book/9780123708670>>. Citado na página 11.
- FLEURY. *Fleury fecha parceria com IBM Watson Health na América Latina*. 2016. <<http://www.fleury.com.br/noticias/Pages/>

fleury-fecha-parceria-com-IBM-watson-health-america-latina.aspx>. Citado na página 8.

HARRIS, K. D. Neural signatures of cell assembly organization. *Nature reviews. Neuroscience*, v. 6, n. 5, p. 399–407, 2005. ISSN 1471-003X. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/15861182>>. Citado na página 11.

HASHIMOTO, Y.; USHIBA, J.; KIMURA, A.; LIU, M.; TOMITA, Y. Change in brain activity through virtual reality-based brain-machine communication in a chronic tetraplegic subject with muscular dystrophy. *BMC neuroscience*, v. 11, n. 1, p. 117, 2010. ISSN 1471-2202. Disponível em: <<http://www.biomedcentral.com/1471-2202/11/117>>. Citado na página 2.

KOUFI, V.; MALAMATENIOU, F.; VASSILACOPOULOS, G.; PRENTZA, A. An Android-enabled mobile framework for ubiquitous access to cloud emergency medical services. *Proceedings - IEEE 2nd Symposium on Network Cloud Computing and Applications, NCCA 2012*, p. 95–101, 2012. ISSN 2333-2549. Citado na página 6.

LONG, J.; CARMENA, J. A statistical description of neural ensemble dynamics. *Frontiers in Computational Neuroscience*, v. 5, p. 52, 2011. ISSN 1662-5188. Disponível em: <<http://journal.frontiersin.org/article/10.3389/fncom.2011.00052>>. Citado na página 11.

LUU, T. P.; HE, Y.; BROWN, S.; NAKAGOME, S.; CONTRERAS-VIDAL, J. L. A closed-loop brain computer interface to a virtual reality avatar: Gait adaptation to visual kinematic perturbations. *2015 International Conference on Virtual Rehabilitation (ICVR)*, IOP Publishing, v. 13, n. 3, p. 30–37, 2015. ISSN 1741-2560. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7358598>>. Citado na página 2.

MANYIKA, J.; CHUI, M.; BUGHIN, J.; DOBBS, R.; BISSON, P.; MARRS. Disruptive technologies: Advances that will transform life, business, and the global economy. *McKinsey Global Institute*, n. May, p. 163, 2013. Disponível em: <[http://www.mckinsey.com/insights/business\technology/disruptive\technologies\\\$delimiter](http://www.mckinsey.com/insights/business\technology/disruptive\technologies\$delimiter)>. Citado na página 7.

MILLER, G. Is pharma running out of brainy ideas? *Science*, v. 329, n. July 2010, p. 502–504, 2010. ISSN 0036-8075. Citado na página 2.

MOLINARI, C. Does the Accountable Care Act aim to promote quality, health, and control costs or has it missed the mark? Comment on "Health system reform in the United States". *International journal of health policy and management*, v. 2, n. 2, p. 97–9, 2014. ISSN 2322-5939. Disponível em: <<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3952545&tool=pmcentrez&rendertype=ab>>. Citado na página 7.

MONKS, K. *This talking, electric, self-driving bus is coming to a city near you*. 2016. <<http://edition.cnn.com/2016/10/20/world/ollis-electric-bus/>>. Citado na página 4.

NICOLELIS. *Interface Cérebro-Máquina*. 2016. <<https://www.youtube.com/watch?v=OvqHP7elwTM>>. Citado 3 vezes nas páginas 2, 13 e 14.

- OUYANG, E. B. B. M. I. f. C. a. R. A. W.; CASHION, K.; ASARI, V. K. A. I. P. R. W. S. f. C.; AIPR, A. . I. Electroencephelograph based brain machine interface for controlling a robotic arm. *Wenjia Ouyang*, 2013. ISSN 15505219. Disponível em: <[PANZERI, S.; BRUNEL, N.; LOGOTHEΤIS, N. K.; KAYSER, C. Sensory neural codes using multiplexed temporal scales. *Trends in Neurosciences*, Elsevier Ltd, v. 33, n. 3, p. 111–120, 2010. ISSN 01662236. Disponível em: <<http://dx.doi.org/10.1016/j.tins.2009.12.001>>. Citado na página 11.

PAVONE, E. F.; TIERI, G.; RIZZA, G.; TIDONI, E.; GRISONI, L.; AGLIOTI, S. M. Embodying Others in Immersive Virtual Reality: Electro-Cortical Signatures of Monitoring the Errors in the Actions of an Avatar Seen from a First-Person Perspective. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, v. 36, n. 2, p. 268–79, 2016. ISSN 1529-2401. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/26758821>>. Citado na página 2.

PRINCE, D.; EDMONDS, M.; SUTTER, A.; CUSUMANO, M.; LU, W.; ASARI, V. Brain machine interface using Emotiv EPOC to control robai cyton robotic arm. *Proceedings of the IEEE National Aerospace Electronics Conference, NAECON*, v. 2016-March, p. 263–266, 2016. ISSN 23792027. Citado na página 2.

Quian Quiroga, R.; PANZERI, S. Extracting information from neuronal populations: information theory and decoding approaches. *Nature reviews. Neuroscience*, v. 10, n. 3, p. 173–85, 2009. ISSN 1471-0048. Disponível em: <\[http://www.nature.com/doifinder/10.1038/nrn2578\\\$\\delimiter"026E30F\\\$nhttp://www.ncbi.nlm.nih.gov/pubmed/19229240\]\(http://www.nature.com/doifinder/10.1038/nrn2578\$\\delimiter\)>. Citado na página 11.

RADU, A.; COSTAN, A.; IANCU, B.; DADARLAT, V.; PECULEA, A. Intercloud platform for connecting and managing heterogeneous services with applications for e-health. *Proceedings - 2015 8th Romania Tier 2 Federation Grid, Cloud and High Performance Computing in Science, ROLCG 2015*, 2015. Citado na página 6.

RATHORE, M. M.; AHMAD, A.; PAUL, A. The Internet of Things based Medical Emergency Management using Hadoop Ecosystem. p. 1–4, 2015. Citado na página 7.

ROSSANT, C.; KADIR, S. N.; GOODMAN, D. F. M.; SCHULMAN, J.; BELLUSCIO, M.; BUZSAKI, G.; HARRIS, K. D. Spike sorting for large, dense electrode arrays. *bioRxiv*, v. 19, n. 4, p. 015198, 2015. ISSN 1097-6256. Disponível em: <\[http://biorxiv.org/content/early/2015/02/16/015198\\\$\\delimiter"026E30F\\\$nhttp://biorxiv.org/content/early/2015/02/16/015198.full.pdf\]\(http://biorxiv.org/content/early/2015/02/16/015198\$\\delimiter\)>. Citado na página 11.

SANTOS, V. Lopes-dos; CONDE-OCAZIONEZ, S.; NICOLELIS, M. A. L.; RIBEIRO, S. T.; TORT, A. B. L. Neuronal assembly detection and cell membership specification by principal component analysis. *PLOS ONE*, Public Library of Science, v. 6, n. 6, p. 1–16, 06](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp={&}arnumber=6749312&matchBoolean=true&rowsPerPage=30&searchField=Search{_}All&queryText=(p{_}Title:)

2011. Disponível em: <<http://dx.doi.org/10.1371%2Fjournal.pone.0020996>>. Citado na página 11.

SANTOS, V. Lopes-dos; RIBEIRO, S.; TORT, A. B. L. Detecting cell assemblies in large neuronal populations. *Journal of Neuroscience Methods*, Elsevier B.V., v. 220, n. 2, p. 149–166, 2013. ISSN 01650270. Disponível em: <<http://dx.doi.org/10.1016/j.jneumeth.2013.04.010>>. Citado na página 11.

SHEDEED, H. A. Brain EEG Signal Processing For Controlling a Robotic Arm. p. 152–157, 2013. Citado na página 2.

WHO. *What are neurological disorders?* 2016. <<http://www.who.int/features/qa/55/en/>>. Citado 2 vezes nas páginas 1 e 13.

ANEXO A – Manuscrito

Spike Scan – A Computational System for Monitoring Patients with Neurological Disorders using Internet of Things

Caroline Stéphanie Cabral Silva^{1*}, Cicilia Raquel Maia Leite^{1,2}, Renan Cipriano Moioli¹.

¹Edmond and Lily Safra International Institute of Neuroscience, Santos Dumont Institute, Macaíba/RN Brazil.

²Departamento de Informática da Universidade do Estado do Rio Grande do Norte (UERN).

*carolinestephanie@gmail.com

***Correspondence:** Dr. Renan Cipriano Moioli, Research fellow at the Edmond and Lily Safra International Institute of Neuroscience, Santos Dumont Institute, Macaíba/RN, 59280-000, Brazil. E-mail: moioli@isd.org.br.

Abstract

This work describes Spike Scan, a computational system for the processing and transmission of electrophysiological data using internet of things (IoT), for remote monitoring (home care) of patients, promoting the increase of quality of life and the decentralization of healthcare. The system proposes an alternative way for monitoring treatments due to the high death rate and limiting conditions derived from neurological disorders. It was used an object-oriented programming approach (OOP) and unified modeling language (UML) for planning, Java programming language, and the integrated development environment (IDE) Android Studio for the Android platform on mobile devices, Firebase services such as "Authentication" for user login, "Storage" for data storage and "Realtime Database" for message interaction between related users. A simulation of the acquisition of a set of artificial data (data generated by the system), the visualization of the data recorded in real time plot and the transmission of the data to the users were developed.

Keywords: IoT, mHealth, Android, home care, electrophysiological signals.

I. Introduction

Recent trends in healthcare innovation use participation from the user in the healthcare delivery process¹. Mobile health (mHealth) refers to the use of mobile devices in administering healthcare services², is one way to facilitate the communication between users and sharing medical data. According to National Institute of Health (NIH) in the announcement of their first mHealth Summer Institute in 2011, mobile technologies “have the potential to transform medical research and enable health care providers to more rapidly and accurately assess biological processes, behavior, attitudes, and the environment”³. It provides digital technologies that allows patients and medical professionals to participate in the related process in a practice manner, playing an increasingly important role in healthcare, self-surveillance and self-care practices⁴.

Neurological disorders represent a remarkable share of costs in hospital systems, low quality of life and dependence on individuals. It is estimated that 1/3 of the population suffers with some chronic condition, and that 15% this work is 70% of non-health care costs⁵. These patients need to face an illness for a long period of

time, and can last years of treatment. Epilepsy for example is a common neurologic disorder that affects about 2% of population, in Brazil there are more than one million of people that suffers with this condition⁶. The Neurology department of São Lucas Hospital manages around 200 to 300 appointments each month. Due the high costs, as well the number of patients we should look for better ways to increase the quality of lives of this people.

This paper presents Spike Scan, a smartphone and cloud based platform for long-term remote patient monitoring, for EEG signal analysis and spike detection, which can provide to patients a more comfortable monitoring experience at home.

II – Spike Scan Overview

It is a modular system for the acquisition of electrophysiological data, aiming at integrating the different signal capture devices offered in the market today, as well as storing, processing and transmitting information to possible users of the system: patients, healthcare professionals, researcher's and alerts users.

The Spike Scan platform comprises the following components:

- Record relevant patient condition data;
- Plot the data captured by external device, enabling online and remote monitoring;
- Selection of attributes for the filtering of the desired signal;
- Transmission of information generated by the system to users;
- Selection of recording and storage of partial data;
- Remote communication between users;
- Alerts to registered external users.

A – Resources for users

The system presented enables interaction between 4 types of users: *PatientUser (PU)*, *HealthcareProfessionalUser (HU)*, *ResearcherUser (RU)* and *AllertUser (AU)*.

The *PU* user can register in the independent system of connection with some *HU*, being able to register personal reports / self-evaluation of symptoms and detailed

observations considered relevant for registration in the history of the specific condition.

HU can add *PU* users to their individual patient follow-up list, participate in the history record of the patient's condition by clinical record, can add examinations performed at the clinic, consult, process, analyze recorded data, request remote use of device for online monitoring or recording for later reference, send and receive instant messages.

The *RU*, if authorized, may have access from more general information, such as the number of users with a specific condition and or data and partial information generated, processed, analyzed by *PU* and / or *HU* users and made available for research purposes according to with authorization in pre-established terms, protecting the user's identity and other relevant characteristics for research.

The user named *AU* may receive instant messages if the system detects any changes considered to be critical to the monitored online signal and classified as emergency, or to the patient's direct emergency request, *PU*, thereby being aware of the current condition and location of a linked user.

B - System Applications

The developed system has the purpose of generating the following benefits to users:

- Remote patient care;
- Optimization of the time in waiting of patients in clinics and hospitals;
- Minimization of unnecessary physical displacement to access digital information;
- Data recording in everyday situations;
- Practical communication between patient and health professional;
- Assistance to scientific research, providing real data in a safe way;
- Dynamicity and synchrony in the access to generated data;
- Decreased duplication of information in health facilities;
- Speed of alert transmission to users in a predetermined emergency situation.

C - Specific tools

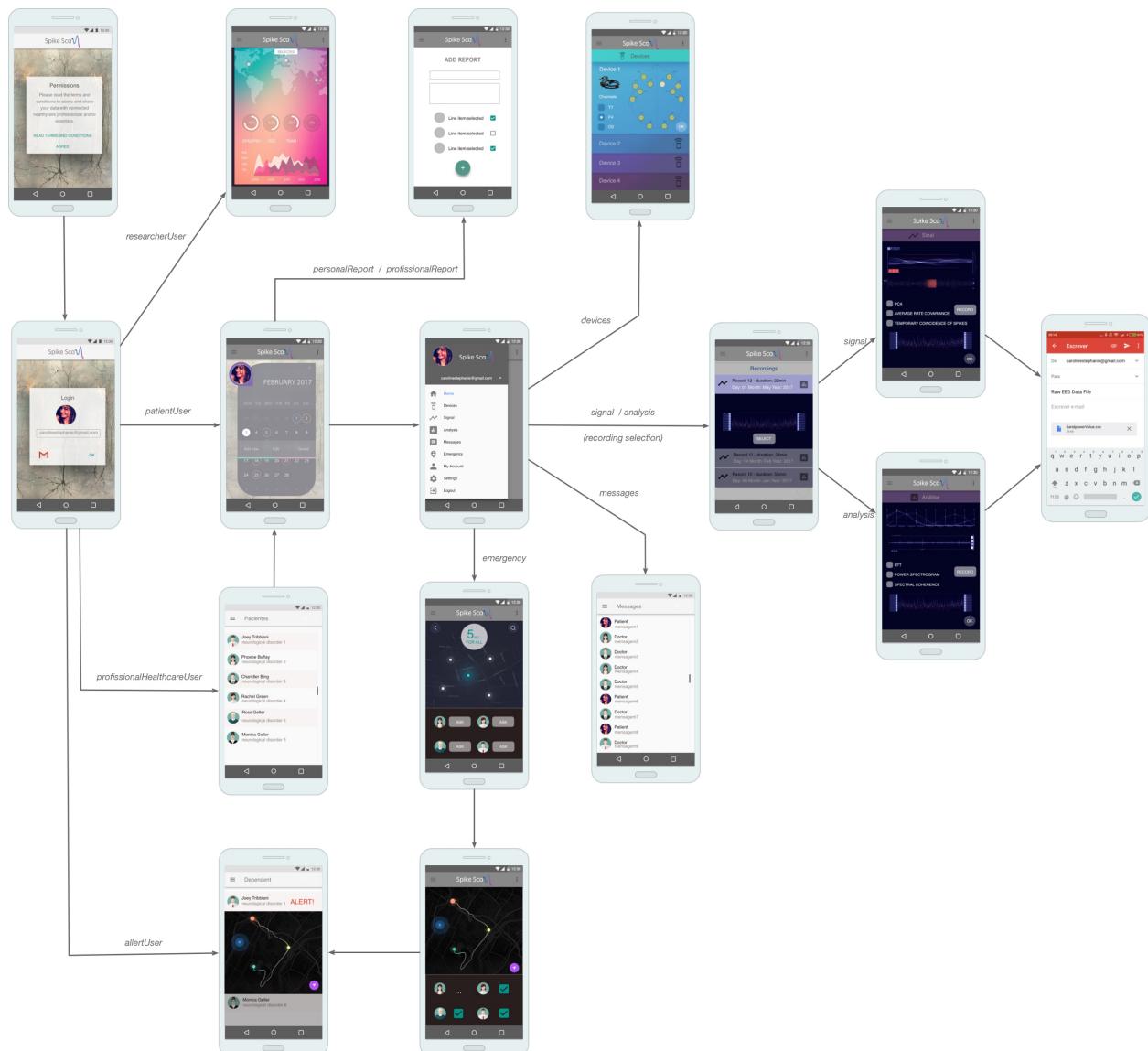
As it concerns the development of a system for remote monitoring, we use Android as an example for the implementation, being easy to use and more affinity for users. Because Android uses Java as the programming language. The specific tools required for the development of the system in question were:

- JDK in Java Standard Edition (SE) 8;
- Android Studio version 2.2.3;
- Version 6.0, API level 23, Marshmallow;
- Google APIs Intel x86 Atom System Image;
- Sources for Android 23;
- Android SDK Build Tools;
- Android SDK Tools;
- Android SDK Platform-Tools;
- Documentation for Android SDK;
- Android Support Repository;
- Google Repository;
- Google USB Driver;
- Intel x86 Emulator Accelerator (HAXM), Emulator Accelerator.

D - Screenflow

The screen flow displays the system navigation relationship for each user. Initially the authorization of the terms of use is requested for the user to register in the system. The user login is done later and the system detects the type of link: "researcherUser", "patient" (patientUser), "health professional" (professionalHealthcareUser) or "alerts" (alertUsers). If you are a "researcher" you will be redirected to the general statistics screen generated by the system. If it is the "patient", it will be taken to the home screen with the calendar updated with the records posted each day. If it is the health professional, the patient selection screen will first be displayed and then be directed to the initial screen of the personal and clinical records. If you are an "alert" user, the next screen will be from the list of paired patients and the map with the location of the patient user when activating the emergency.

From the main menu, it is possible to select the screen for detection of external devices, signals, analysis, messages and emergency. In the case of signal and analysis screens, it is necessary to select the desired signal for manipulation. Later it is possible to send the data by email to the users.



III – Conclusion

The purpose of the system was to reduce the waiting time in communication between patients and health professionals, to maintain a rich and up-to-date history of private information for each condition, to provide easy access to data reliably, to generate an increase in quality And provide specific data for research.

The planning and development of a computer system to process and transmit electrophysiological signals was carried out with the purpose of remote monitoring of patients with neurological disorders by health professionals. As a result of this work, the implementation of the \textit{Android} connection with the \textit{Firebase}, the authentication of the users, the selection of patients, the implementation of the system main menu, the listing of the (Allowing the selection and visualization of the record summary), the interaction between the linked users, the visualization of the signal registered in real time through the graph plot (with filter option of the channels) , Recording the generated data, transferring the signal register, sending the signal and viewing the sent file.

Acknowledgments

We would like to thank our partners at Edmond and Lily Safra International Institute of Neuroscience (IINELS) and Department of Informatics of the State University of Rio Grande do Norte (UERN) in Brazil for their support on this research.

References

1. Handel MJ. "mHealth (mobile health) -- using apps for health and wellness". EXPLORE: J. of Sci H 2011;7:265-261;
2. Kazi AM, Jafri LA. The use of mobile phones in polio eradication. 2015. <http://www.who.int/bulletin/volumes/94/2/15-163683/en/>;
3. National Institutes of Health launches summer institute on mHealth. <http://www.nih.gov/news/health/feb2011/od-28.htm>;
4. Lupton, D. 2012. "M-Health and Health Promotion: The Digital Cyborg and Surveillance Society". Social Theory & Health 10 (3): 229–244;
5. World Health Organization. "What are neurological disorders?", 2016. <http://www.who.int/features/qa/55/en/>;
6. A. Melo; E.M. Yacubian; M.L. Nunes. "Crises epilépticas e epilepsias ao longo da vida: 100 questões práticas", São Paulo, Brazil, 2006;