

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA DA COMPUTAÇÃO**

GIAN LUCA SILVA MELLA

Um Modelo para Monitoramento de Sinais Vitais com BITalino

São Leopoldo

2019

GIAN LUCA SILVA MELLA

Um Modelo para Monitoramento de Sinais Vitais com BITalino

Trabalho de Conclusão de Curso
apresentado como requisito parcial para
obtenção do título de Graduado em
Engenharia da Computação, pelo Curso de
Engenharia da Computação da
Universidade do Vale do Rio dos Sinos -
UNISINOS

Orientador: Prof. Dr. Cristiano André da Costa

São Leopoldo

2019

RESUMO

O presente trabalho descreve o desenvolvimento de um protótipo para aquisição de sinais vitais provenientes do *kit* de desenvolvimento *BITalino* através de um aplicativo Android. O processamento dos dados adquiridos é realizado através de uma *WebApi* hospedada na nuvem que garante a persistência dos registros em banco de dados, a filtragem do sinal obtido pelo *kit* e a criação de um prontuário eletrônico que segue o padrão a norma mundial *openEHR* garantindo interoperabilidade com outros sistemas que reconhecem a norma. Todas as operações são visualizadas através do aplicativo Android, dentre elas está a possibilidade de cadastrar pacientes, executar e visualizar exames, criar e visualizar prontuários. A operabilidade do modelo proposto foi apurada através de testes funcionais e integração básica com o sistema *EHRServer* que armazena e cria versões de prontuários eletrônicos. Os resultados obtidos com os testes foram positivos e mostraram que há muito a ser explorado e acrescentado no modelo proposto.

Palavras-chave: Dispositivos Móveis, Monitoramento de Sinais Vitais, OpenEHR, Bitalino.

SUMÁRIO

RESUMO	2
1 INTRODUÇÃO	8
1.1 Motivação	9
1.2 Objetivos	9
1.2.1 Objetivo Geral	9
1.2.2 Objetivos Específicos	10
1.3 Organização do Trabalho	10
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 Wearable	11
2.2 Bluetooth	11
2.3 BITalino	12
2.4 OpenSignals	14
2.5 Processamento Digital de Sinais	15
2.5 Eletromiografia	16
2.6 Resposta Galvânica da Pele	17
2.7 Eletrocardiografia	17
2.8 Eletroencefalografia	17
2.9 openEHR	18
2.10 EHRServer	19
3 TRABALHOS RELACIONADOS	21
3.1 Soluções de Baixo Custo para Implementação de Internet das Coisas na Área da Saúde	21
3.1.1 Arquitetura e Implementação	22
3.1.2 Aplicativo	23
3.2 WHMS4 - Um Sistema de Monitoramento Remoto de Pacientes de Forma Integrada, Flexível, Escalável e com Baixo Custo	23
3.2.1 Arquitetura e Implementação	24
3.2.2 Aplicativo	25
3.3 UbHeart - Um Modelo para Monitoramento de Sinais Vitais do Coração Baseado em Ciência da Situação e Computação Ubíqua	26
3.3.1 Arquitetura e Implementação	26
3.3.2 Aplicativo	27

3.4 MobileVJ: A mobile app for a novel wearable human sensing system	28
3.4.1 Arquitetura e Implementação	28
3.4.2 Aplicativo	29
3.5 Análise Comparativa	30
4. MODELO PROPOSTO	33
4.1 Visão Geral do Modelo	33
4.2 Camada Cliente	34
4.3 Camada de Serviço	35
4.3.1 Processamento dos Sinais	36
5. METODOLOGIA DE AVALIAÇÃO	41
5.1 Implementação	42
5.1.2 Android	42
5.1.3 Filtragem do Sinal	49
5.1.4 openEHR.....	52
5.1.5 .NET Core	56
5.1.6 SQL Server.....	58
5.1.7 Azure	59
5.1.8 Github.....	60
5.2 Análise de Resultados	60
6. CONCLUSÃO	67
REFERÊNCIAS	68
APÊNDICE A – XML DE PRONTUÁRIO ELETRÔNICO	74

LISTA DE FIGURAS

Figura 1 - BITalino.....	14
Figura 2 - Interface do OpenSignals.....	15
Figura 3 - <i>Template</i> de Ficha de Atendimento	19
Figura 4 - Arquitetura preliminar proposta por JUCHEM	22
Figura 5 - Arquitetura preliminar do WHMS4.....	24
Figura 6 - Arquitetura do UbHeart	27
Figura 7 - <i>Kit VitalJacket</i>	29
Figura 8 - Visão geral do modelo proposto	34
Figura 9 - Camada de Processamento de dados	35
Figura 10 - Mapeamento das frequências do plano-s para o plano-z utilizando transformação bilinear.....	38
Figura 11 - Fluxo de Ações do Aplicativo	43
Figura 12 – Tela de Cadastro de Pacientes	44
Figura 13 – Tela de Listagem de Pacientes	44
Figura 14 – Tela de Seleção de Dispositivo e Execução de exame.....	45
Figura 15 – Tela de Visualização de Exame Eletrocardiograma	46
Figura 16 – Tela de Visualização de Exame Eletromiografia	47
Figura 17 – Tela de Visualização de Exame Resposta Galvânica da Pele	47
Figura 18 – Tela de Visualização de Exame Eletroencefalografia	48
Figura 19 – Tela de Exames Realizados.....	48
Figura 20 – Exemplo de Aliasing: (a) Espectro do sinal, (b) espectro do sinal com teorema de Nyquist, (c) espectro do sinal com aliasing	49
Figura 21 - Filtragem do sinal de Eletrocardiograma.....	50
Figura 22 - Filtragem do sinal de Resposta Galvânica da Pele.....	50
Figura 23 - Filtragem do sinal de Eletromiografia.....	51
Figura 24 - Filtragem do sinal de Encefalografia.....	51
Figura 25 - Principais classes de arquétipos.....	52
Figura 26 - Software Ocean Template Designer	53
Figura 27 – Tela de exibição de validação do pronturário	55
Figura 28 - Componentes lógicos do CQRS	57
Figura 29 – Visualização do Swagger	58
Figura 30 – Modelo ER do Banco de Dados	59
Figura 31 - Painel de Recursos do Azure.....	59

Figura 32 - Lista de Modelos do EHRServer	60
Figura 33 - Lista de EHRs do EHRServer	61
Figura 34 - Confirmação de persistência do prontuário eletrônico pelo Insomnia	62
Figura 35 - Detalhamento do EHR no EHRServer	63
Figura 36 - Detalhamento do Versionamento do EHR no EHRServer	64
Figura 37 - Tela de detalhamento do prontuário eletrônico	65

LISTA DE TABELAS

Tabela 1 - Especificações do BITalino	13
Tabela 2 - Avaliação dos Trabalho Relacionados	31
Tabela 3 - Polinômios de Butterworth	37
Tabela 4 - Características dos Sensores do <i>BITalino</i>	40
Tabela 5 – Correspondência entre exame e canal de comunicação do <i>BITalino</i>	45
Tabela 6 - Arquétipos escolhidos para o template	54

1 INTRODUÇÃO

No mundo contemporâneo o celular virou um artefato essencial na vida das pessoas. Conforme levantamento feito pela empresa de estatísticas *Statista* com dados de 2016, o brasileiro utiliza o celular em média quatro horas e quarenta e oito minutos por dia (ARMSTRONG, 2017) e este tempo é dividido entre centenas de aplicativos com os mais diversos propósitos e que não necessariamente exploram todo o potencial que a tecnologia móvel possui aliado à sua constante evolução. Atualmente existem dispositivos no mercado com uma boa eficiência energética e com processadores atingindo a frequência de *clock* de 2,0GHz (SNAPDRAGON, 2018) o que resulta em capacidade suficiente para executar aplicativos mais complexos junto a periféricos conectados ao *smartphone* fornecendo melhores experiências ao usuário final.

Aliado ao uso do celular existe um aumento de aplicativos nas lojas virtuais dispostos a auxiliar o usuário a cuidar da sua saúde. (POHL, 2018) Dentre eles destacam-se programas para dietas, monitores de passos, carteiras de vacinas, monitoradores de sono, alertas para consumo de água, entre outros que aproveitam a mudança comportamental do ser humano para um estilo de vida mais saudável (NOVAIS, 2018), pois segundo uma pesquisa realizada pelo Ministério da Saúde Brasileiro a prática de atividades físicas aumentou 24,1% de 2009 a 2017 o que contribui com a motivação para o desenvolvimento de aplicativos do gênero. (BRASIL, 2018)

Mesmo com os avanços tecnológicos e o aumento do número de pessoas na busca por uma vida saudável, segundo levantamento feito pela empresa Royal Philips tanto profissionais da área da saúde quanto a população em geral expressam pouca confiança nos sistemas de saúde, sendo 20% da população e 8% dos profissionais da área os que concordam que o sistema funciona adequadamente. (HOSPITALAR, 2018) Porém, com relação a tecnologia é evidente a flexibilidade dos profissionais em aceitar o uso da mesma para análise de exames e cuidados conectados a saúde, cerca de 86% dos profissionais acredita que a tecnologia é uma forte aliada na prevenção de questões médicas. (HOSPITALAR, 2018)

Naturalmente o campo de pesquisas acadêmicas e tecnológicas na área da saúde vem em crescente expansão desde 2013, pois segundo Tibes, Dias e Zem-Mascarenhas (2014, p. 476) a popularização do uso de smartphones e tablets auxilia

na expansão destas publicações, tanto que este trabalho mostra o estado da arte com quatro publicações escolhidas entre quatorze outras que visavam melhorar o sistema de saúde, a vida dos pacientes em estado de risco e o trabalho dos profissionais da área em relação aos cuidados do ser humano.

Contribuindo a este campo de pesquisas, este trabalho visa propor um modelo para assessorar médicos e/ou enfermeiros no diagnóstico de irregularidades na saúde de pacientes de maneira rápida e segura junto a um aplicativo e um *kit* de sensores para leitura de sinais vitais.

1.1 Motivação

A motivação deste trabalho é ilustrada por Rocha et al. (2017, p. 2),

Os smartphones permitem a utilização de uma gama de aplicativos, e o mercado disponibiliza os mais variados tipos, desde os que possuem a função de entretenimento até aqueles que buscam orientar pacientes e profissionais de saúde quanto ao cuidado e manutenção da saúde.

A oportunidade de estudar um modelo que poderá auxiliar nas atividades dos profissionais da saúde, também dos pacientes, aliado a tecnologia móvel completa a motivação deste trabalho.

1.2 Objetivos

Em seguida serão apresentados o objetivo geral e os objetivos específicos que dirigirão o desenvolvimento do modelo proposto neste trabalho.

1.2.1 Objetivo Geral

Com base nos levantamentos e pesquisas realizadas no cenário nacional, este trabalho tem como objetivo propor um modelo computacional móvel que apoiará os profissionais da área da medicina no monitoramento da saúde de pacientes e na análise dos seus sinais vitais, utilizando um conjunto de sensores atrelados a um dispositivo móvel com um aplicativo capaz de interpretar e armazenar os dados provenientes dos sensores para análise médica.

1.2.2 Objetivos Específicos

- a) estudar o cenário atual de monitoramento de sinais vitais em dispositivos móveis;
- b) compreender o funcionamento do *kit* de desenvolvimento *BITalino* com sensores para leitura de sinais vitais;
- c) desenvolver um *software mobile* que auxilie na leitura dos dados fornecidos pelos sensores;
- d) analisar o protótipo desenvolvido.

1.3 Organização do Trabalho

A fundamentação teórica descrita no capítulo 2 apresentará os principais conceitos abordados neste trabalho, além de tecnologias empregadas para construção do modelo proposto no capítulo 4 que tem como objetivo preencher lacunas deixadas nas soluções apresentadas nos trabalhos descritos no capítulo 3.

O capítulo 5 consiste na discussão sobre o desenvolvimento do modelo proposto e da análise dos resultados obtidos até o momento, fechando com a conclusão como último capítulo do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção será apresentado os principais conceitos que serão explorados na construção e desenvolvimento do modelo proposto. A estrutura do capítulo começará explicando sobre vestíveis (*wearables*) seguindo da tecnologia de comunicação *Bluetooth*, sobre o kit de desenvolvimento *BITalino* e seus sensores.

2.1 Wearable

Um computador vestível (*wearable*) deve ser incorporado ao espaço pessoal do seu usuário, sem limitar seus movimentos ou dificultar sua mobilidade corporal, além disso o dispositivo deve possuir uma performance computacional capaz de auxiliar em atividades cognitivas ou motoras. O funcionamento de um dispositivo vestível é mais interativo visto que o mesmo deve aprender informações sobre quem o usa e sobre o ambiente onde é utilizado, fazendo com que essa seja a principal diferença entre um dispositivo vestível e um computador convencional ou um *smarthphone*. (DONATI, 2004, p. 94)

Uma das tecnologias empregadas na comunicação de computadores vestíveis é a rede sem fio *Bluetooth*.

2.2 Bluetooth

Atualmente a especificação de rede de comunicação sem fio *Bluetooth* é utilizada em diversas finalidades, principalmente no pareamento entre celular e dispositivos ou em controles de videogames. A regulamentação da rede é feita pelo consórcio *Bluetooth Special Interest Group* (SIG) fundado em 1998 englobando empresas como IBM, Ericson, 3Com, Microsoft. Tradicionalmente, o *Bluetooth* é utilizado em redes locais (LAN - *Local Area Network*) e redes pessoais (PAN - *Personal Area Network*). (CALDAS NETO, 2011)

Para a tecnologia de comunicação do *Bluetooth* se fez necessária a utilização de uma frequência de rádio aberta e aceita no mundo inteiro, possibilitando que a tecnologia fosse utilizada em qualquer lugar do planeta. A frequência de 2,45 GHz é a faixa que mais se adequa a esta necessidade, na qual opera a faixa ISM (*Industrial, Scientific, Medical* – Industrial, Científico, Médico). Por esta ser a mesma frequência do

Wifi, o Bluetooth está sujeito as mesmas dificuldades e problemas de interferência desta faixa. (OLIVEIRA, 2017)

Dentre os dispositivos capazes de se comunicarem via *Bluetooth* existe o *BITalino* que apresenta um conjunto de sensores para monitoramento de sinais vitais.

2.3 BITalino

Conforme documentação disponível no site do projeto, o *BITalino (r)evolution* consiste a nova geração da plataforma *BITalino* e é definido como um kit de ferramentas de hardware e software projetado especificamente para lidar com sinais corporais e está disponível em três versões que se comunicam através da tecnologia *Bluetooth* e as mesmas também estão disponíveis com comunicação *Bluetooth Low Energy*. (BITALINO, 2018)

Nas versões disponíveis para venda é possível adquirir o *BITalino (r)evolution Board* que possui uma placa *all-in-on* e com sensores pré conectados e prontos para uso. Também existe o *BITalino (r)evolution Freestyle* que além de ter as mesmas características que o *BITalino (r)evolution Board*, facilita a construção de uma plataforma *wearable* customizada já que as peças de conexão com os sensores são separadas e possibilitam soldar os componentes da maneira que forem necessários. O *BITalino (r)evolution Plugged* também possui as mesmas características do *BITalino (r)evolution Board*, com o diferencial de haver uma conexão cabeada entre os sensores e placa principal. (BITALINO, 2018)

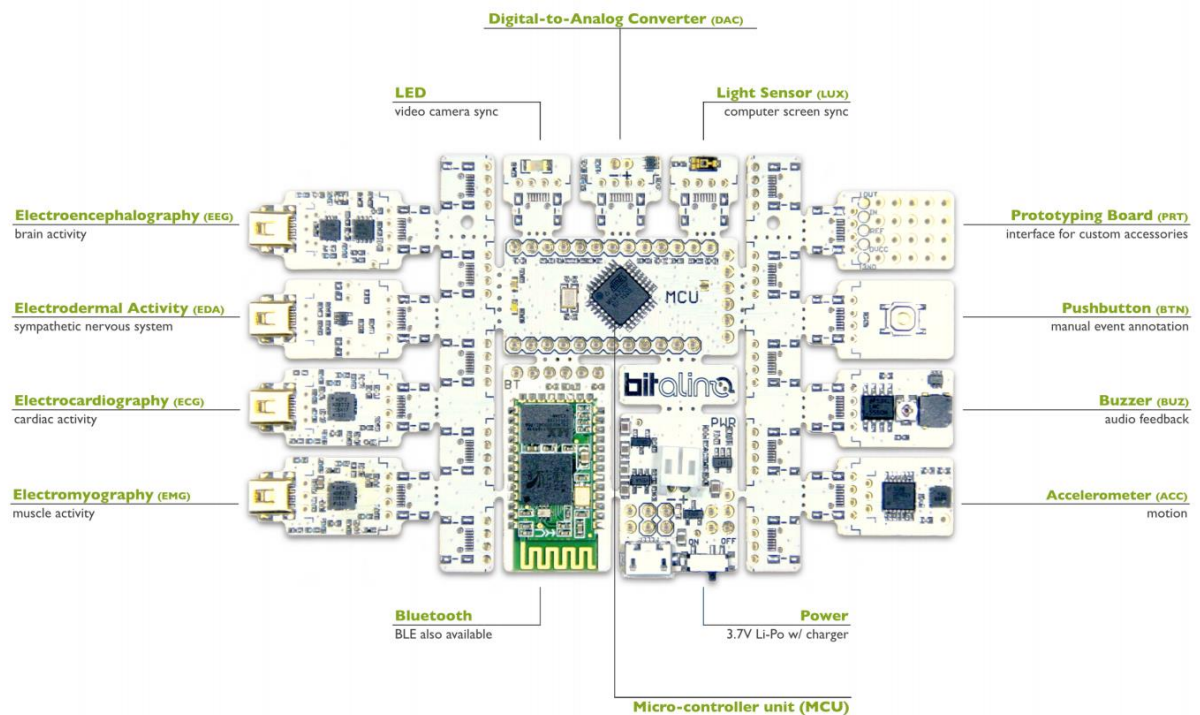
No kit de hardware do *BITalino (r)Evolution*, apresentado na Figura 1, estão presentes canais de comunicação para sensores de eletromiografia (*Electromyography - EMG*), resposta galvânica da pele, (*Galvanic Skin Response - GSR*), Eletrocardiografia (*Electrocardiography - ECG*), fotopletismografia (*Photoplethysmography - FPG*) além de acelerômetro e fotodiodo. As demais características do *kit* de desenvolvimento são apresentadas na Tabela 1. (REIGONES; GASPAR; GARCIA, 2017)

Tabela 1 - Especificações do BITalino

Módulo	Características
Eletromiografia (EMG)	Número de Eléttodos: 3
	CMRR: -110 dB
	Impedância de Entrada: 1000GΩ3pF
	Limites: 0 - 3,3 mV
	Ganho: 1000
	Largura de Banda: 10 - 400 Hz
	Resolução: 10 bits
Resposta galvânica da pele (GSR)	Número de Eléttodos: 2
	Largura de Banda: 0 - 3 Hz
	Limites: 1 MΩ
	Resolução: 10 bits
Luz	Gama do espectro eletromagnético: 360-970 nm
Eletrocardiograma (ECG)	Número de Eléttodos: 2
	CMRR: 110 dB
	Impedância de Entrada: 1000GΩ3pF
	Limites: 0 - 3 mV
	Ganho: 1100
	Largura de Banda: 0,5 - 40 Hz
	Resolução: 10 bits
Acelerômetro	Largura de Banda: 0 - 50 Hz
	Limites: ± 3g
Comunicação	Bluetooth 2.0
Dimensões	105x60x6mm

Fonte: (REIGONES; GASPAR; GARCIA, 2017)

Figura 1 - BITalino



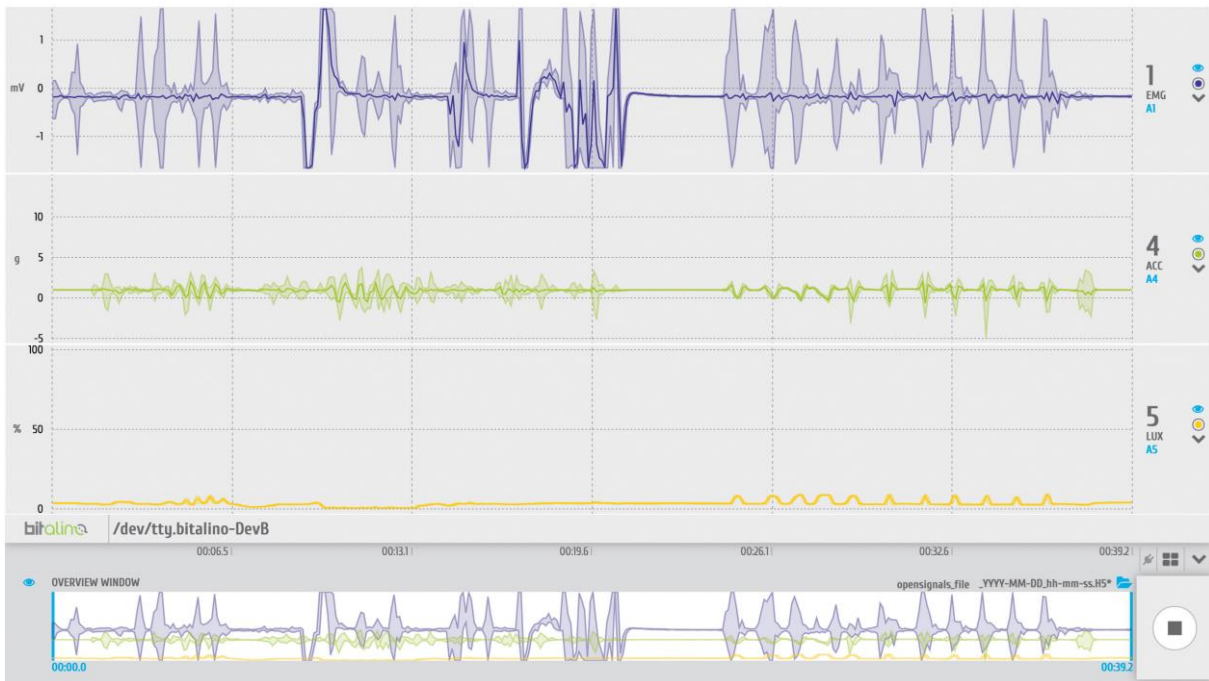
Fonte: Adaptado de BITalino (2018)

Os dados obtidos pelos sensores podem ser visualizados através de um *software* gratuito, cujo nome é *OpenSignals* e disponibilizado no site do *BITalino*.

2.4 OpenSignals

O *OpenSignals* é um *software* de código aberto desenvolvido pela empresa Plux e escrito na linguagem de programação *Python* que permite extrair e visualizar os dados dos sensores do *BITalino* em tempo real, um exemplo disso está na Figura 2. Com o *OpenSignals* é possível, inclusive, efetuar a leitura de sinais de múltiplos canais do *BITalino*, gerar relatórios e gravar os dados obtidos. (OPENSINGALS, 2018)

Figura 2 - Interface do OpenSignals



Fonte: Adaptado de BITalino (2018)

2.5 Processamento Digital de Sinais

De acordo com Santana, Nascimento e Santos (2017) a aquisição de sinais digitais a partir de um sinal analógico consiste em três etapas: a captação do sinal, o seu condicionamento e a digitalização. A primeira etapa se refere a transdução da forma elétrica do sinal, já na segunda etapa são feitas operações como a amplificação do sinal e filtragem do sinal analógico. A última etapa, quando o sinal é digitalizado, é o resultado de uma sequência de amostras digitalizado por uma interface.

Para possibilitar o registro digital de todas as frequências analisadas no processamento do sinal, deve-se respeitar o Teorema de Nyquist, que sugere uma taxa de amostragem de no mínimo duas vezes o valor a frequência máxima alcançada pelo sinal analógico. Caso a taxa de amostragem seja menor que a frequência de Nyquist, ocorre o efeito *aliasing*, em que uma alta frequência é medida equivocadamente como sendo a frequência mais baixa. (SILVA, 2016)

Geralmente sinais eletrofisiológicos, que são capturados através de eletrodos metálicos em diferentes partes do corpo humano, possuem uma amplitude baixa, na ordem de centenas ou milhares de microvolts. Por esta razão se faz uso de amplificadores diferenciais na etapa de condicionamento do sinal, pois tais

amplificadores são sensíveis a diferença de potencial entre dois pontos, além de permitir a redução do ruído da rede elétrica, comumente de 60Hz. (SANTANA; NASCIMENTO; SANTOS, 2017)

Ainda citando Santana, Nascimento e Santos (2017), afim de extrair apenas a parte desejável do sinal biomédico, é efetuada a filtragem do sinal para facilitar sua análise. Os filtros geralmente utilizados são os seguintes:

- Filtro passa-baixa: remove componentes senoidais acima de determinada frequência;
- Filtro passa-alta: é utilizado para permitir a passagem de partes do sinal acima de uma determinada frequência;
- Filtro passa-faixa: com este filtro é permitido a passagem de componentes do sinal dentro de uma faixa de frequência;
- Filtro rejeita-faixa: remove parte do sinal dentro de determinada faixa de frequência.

Os sinais vitais que podem ser obtidos através dos sensores do *BITalino*, como eletromiografia, resposta galvânica da pele, eletrocardiografia e eletroencefalografia podem receber algum tipo de processamento para melhor interpretação de um profissional da área da saúde.

2.5 Eletromiografia

Segundo Oncins (2006, p. 157), a eletromiografia (EMG) é um exame que submerge a detecção e os registros potenciais elétricos nas fibras musculares registrando a atividade muscular em microvolts (mV) e em décimos de segundo, através da utilização de eletrodos bipolares, quase sempre descartáveis, na região correspondente a cada músculo da superfície da pele. É um exame que não apresenta nenhum risco ao paciente.

Logo, a eletromiografia é o registro elétrico de um músculo utilizado para avaliação do padrão de atividade do muscular, podendo ser registrado os potenciais de ação que ocorrem através da resposta de um músculo a partir de uma ativação voluntária do mesmo. (MALTA; CAMPOLONGO; BARROS; OLIVEIRA, 2006)

Completando a definição, conforme Caparelli (2007, p. 4), o sinal eletromiográfico é gerado pelas fibras musculares quando excitadas por neurônios motores, resultando na soma de vários potenciais de ação.

2.6 Resposta Galvânica da Pele

A Resposta Galvânica da Pele (*Galvanic Skin Response - GSR*), ou Atividade Eletrodérmica (*Electrodermal Activity - EDA*), corresponde a variação dos aspectos elétricos da pele que são relacionados com alguns tipos de reações do sistema nervoso simpático. Para medir a resposta galvânica da pele é utilizado um sensor composto de eletrodos que detectam a mudança de condutividade na pele. (SOARES, 2016)

Neste exame, a condutância da pele é medida em reação a estímulos e é empregado em estudos relacionados a questões gerenciais ou estratégicas por possibilitar a detecção de emoções e respostas ao estímulo aplicado. (SHIGAKI, 2017)

Conforme escrito por Pereira (2010, p.3) a resposta galvânica da pele é utilizada para descrever as alterações psicológicas na forma de atividade elétrica, através da variação das glândulas sudoríparas.

2.7 Eletrocardiografia

De acordo com Figueiredo (2018 p. 16), a eletrocardiografia (ECG) é um exame apontado pelo eletrocardiógrafo que através do registro do sinal elétrico do coração auxilia o diagnóstico de doenças cardíacas além de servir como base para um tratamento preventivo de infarto de um paciente. Tanto a contração quanto o relaxamento do coração são causados pela despolarização e repolarização elétrica das células cardíacas, por esta razão a leitura do sinal elétrico cardíaco é uma forma de obter informações do coração.

2.8 Eletroencefalografia

A transmissão de sinais elétricos do cérebro realizada pelas células nervosas traz como seus dependentes todas as funções, incluindo sensibilidade, emoções e raciocínio. Logo, o conjunto de potenciais elétricos capturados na região do encéfalo são capturados pelo eletroencefalograma através de eletrodos de superfície colocados em áreas diferentes do cérebro, em que é medido a diferença de potencial entre um eletrodo e outro. (CAPARELLI, 2007)

Segundo Ricardo, (2012 p. 23), o eletroencefalograma (EEG) é um exame de extrema importância na epileptologia e serve como prognóstico para detecção de alterações no estado de consciência e coma.

Os dados de leitura dos exames podem ser convertidos em um formato de uma norma mundialmente utilizada para melhorar ainda mais a interpretação e interoperabilidade, chamada *openEHR*.

2.9 openEHR

Segundo Alves (2017, p. 22) a norma *openEHR* consiste em um conjunto de especificações e ferramentas livres para criação de registros de saúde. Neste conjunto, é seguido um padrão de modelagem de dados separados em camadas de informação e conhecimento e disponibilizada em uma arquitetura orientada a serviços. A representação dos conceitos clínicos é disposta na camada de informação e na camada de conhecimento fica a cargo dos profissionais a área da saúde modelarem as características mais convenientes para o registro da saúde.

A principal vantagem da utilização do *openEHR* é a independência tecnológica, já que, usando entidades de modelo de informação, o conhecimento clínico é estruturado além de ser um padrão adotado por países como Argentina, Brasil, Alemanha, Japão, Nova Zelândia, Portugal, Espanha, Eslováquia e Suécia que o utilizam em pesquisas e até mesmo em soluções. (ALVES, 2017)

Na abordagem do *openEHR* os dados são modelados em conformidade a um conjunto de conceitos pré-estabelecidos que são chamados de arquétipos e definem a camada de conhecimento. Estes arquétipos são expressados em uma linguagem chamada ADL (*Archetype Definition Language* - Linguagem de Definição de Arquétipo) e podem ser recuperados por meio de consultas realizadas na linguagem AQL (*Archetype Query Language* - Linguagem de Consulta de Arquétipo) (MARTINS, 2016)

Figura 3 - *Template* de Ficha de Atendimento

The image displays a medical record template. On the left, a grid labeled 'Vários Arquétipos' shows multiple instances of different data entry forms. On the right, a detailed view of the 'Exame físico' (Physical Exam) form is shown. It includes the following sections and fields:

- Exame físico** (Physical Exam)
- Sinais Vitais** (Vital Signs)
 - Pressão Sanguínea** (Blood Pressure)
 - Hora do evento: 23/06/2008 11:08
 - Sistólica: 189 mm[Hg]
 - Diastólica: 110 mm[Hg]
- Estado** (Status)
 - Posição: Em pé
- Temperatura Corpórea** (Body Temperature)
 - Hora do evento: 23/06/2008 11:08
 - Temperatura: 39.2 °F
- Protocolo** (Protocol)
 - Local da medição: Oral
- Peso Corporal** (Body Weight)
 - Hora do evento: 23/06/2008 11:08
 - Peso: kg
 - Estado: (dropdown menu)

Fonte: Adaptado de ALVES (2017)

No padrão *openEHR* são utilizados *templates* para expressar um conjunto de arquétipos e estes podem ser obtidos junto ao repositório internacional CKM (*Clinical Knowledge Manager* - Gerenciador de Conhecimento Clínico) abastecido pela fundação *openEHR* que também disponibiliza ferramentas para sua utilização e faz a gestão das normas utilizadas. A Figura 3 ilustra um *template* de um exame físico que agrega os arquétipos de sinais vitais, temperatura corpórea, peso corporal entre outros. (ALVES, 2017)

Uma maneira de validar os arquivos gerados no formato que atende esta norma é através de uma ferramenta de código aberto chamado *EHRServer*.

2.10 EHRServer

O *EHRServer* consiste em um repositório de informação clínica gratuito, que e além de ser compatível com a norma *openEHR*, respeita a arquitetura orientada a serviço (REST API) e através dela se possibilita o armazenamento de dados clínicos no formato XML ou JSON (*JavaScript Object Notation* - Notação de Objetos JavaScript) garantindo a uniformização dos dados através da base de dados EHR.

Outra funcionalidade do *EHRServer* é a construção de *queries* através da interface de utilização do cliente, permitindo consultas elaboradas. (CORREIA, 2016)

Dentro do *EHRServer* é possível submeter *templates* customizados que suportarão os dados clínicos e cada paciente possui vários registros EHR (*Electronic Health Record* - Prontuário Eletrônico do Paciente) que são estruturados através de versões e de uma estrutura de dados que persiste informações de quem criou o registro, quando o submeteu, além do armazenamento do dado clínico em si. (CORREIA, 2016)

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados um resumo e uma análise realizada em cima dos trabalhos relacionados ao tema proposto que foram encontrados através de intensa pesquisa utilizando as palavras-chave “*monitoring vital signs*” e “*app vital signs*” no buscador Google Acadêmico, no portal de periódicos da CAPES e no repositório TCC da Unisinos. Foram encontrados trinta e oito documentos relacionados ao tema, dos quais quatorze foram selecionados para leitura e apenas quatro deles ocuparam o espaço de trabalhos relacionados, pois dentre os demais, estes quatro apresentam um modelo similar ao idealizado no desenvolvimento deste trabalho e assim possibilitam um comparativo mais interessante, além de possuírem protótipos e conceitos compatíveis com dispositivos móveis, também abordado neste trabalho

O objetivo principal do estudo dos trabalhos relacionados é expor os pontos positivos e negativos de cada trabalho compondo assim um comparativo para auxiliar na conciliação do modelo a ser definido mais a frente, com a intenção de contribuir com o que ainda não foi totalmente explorado além de aproveitar o que há de bem aplicado e desenvolvido em soluções de monitoramento remoto de sinais vitais junto com a utilização de vestíveis na área da saúde. Desta maneira, o resultado da análise e da comparação de cada trabalho relacionado enriquecerá a construção do modelo proposto que será descrito no próximo capítulo.

3.1 Soluções de Baixo Custo para Implementação de Internet das Coisas na Área da Saúde

No trabalho “*Soluções de Baixo Custo para Implementação de Internet das Coisas na Área da Saúde*” (JUCHEM, 2015) o autor descreve um modelo utilizando conceitos e práticas existentes na Internet das Coisas empregando uma solução de baixo custo em maternidades que monitora, com diversos sensores, a saúde de um recém-nascido. O monitoramento prevê a medição da frequência cardíaca, frequência respiratória e temperatura corporal, posteriormente os dados lidos pelos sensores são enviados para um software que é responsável por trata-los e enviá-los através da ethernet em linguagem HTML (*HyperText Markup Language*, ou Linguagem de

Marcação de Hipertexto) para um aplicativo escrito na plataforma iOS, responsável pela exibição das informações de maneira legível e amigável ao usuário final.

3.1.1 Arquitetura e Implementação

A arquitetura proposta pelo autor assegura basicamente uma placa controladora, um módulo ethernet, um circuito de leitura, um conjunto sensorial, o aplicativo iOS e um roteador, como pode ser visto na Figura 4.

Figura 4 - Arquitetura preliminar proposta por JUCHEM



Adaptado de: JUCHEM (2015)

No que diz respeito a placa controladora da arquitetura do projeto, foram utilizados dois *Arduino Uno*, sendo um para a comunicação com o módulo ethernet e o outro para comunicação com o conjunto de sensores. A decisão de utilizar dois microcontroladores ocorreu devido a limitação no número de pinos do *Arduino Uno*

para comunicação SPI, sendo necessário então comunicar as duas placas através do protocolo I2C. O módulo *ethernet* escolhido pelo autor foi o ENC28J60 do fabricante *Microchip*, devido a sua compatibilidade com a placa controladora e por fornecer um endereço de IP na rede visível a qualquer dispositivo que possa se conectar na rede *ethernet*. (JUCHEM, 2015)

O conjunto sensorial para medição dos sinais vitais utilizado se chama *e-Health* e é fabricado pela empresa *Cooking Hacks*, esta plataforma tem total compatibilidade com a placa controladora Arduino Uno por ser construída no modelo de *shield*, o que significa que é uma placa de expansão de hardware para *Arduino*. Dos 10 sensores biométricos disponíveis na plataforma, foram utilizados somente o sensor de pulsioximetria, o sensor de temperatura corporal e o sensor de frequência respiratória. A implementação do software foi realizada dentro da plataforma Arduino com auxílio da biblioteca *e-Health* que interpreta os dados lidos pelos sensores e os converte em valores de fácil compreensão para um profissional da saúde. (JUCHEM, 2015)

3.1.2 Aplicativo

Os dados oriundos da leitura da plataforma de sensores foram exibidos para o usuário final através de um aplicativo desenvolvido apenas para o sistema operacional iOS utilizando a ferramenta XDK da Intel para prototipagem das telas e do código. (JUCHEM, 2015)

3.2 WHMS4 - Um Sistema de Monitoramento Remoto de Pacientes de Forma Integrada, Flexível, Escalável e com Baixo Custo

O modelo proposto pelo autor deste trabalho é um sistema de monitoramento de sinais vitais que, juntamente com um conjunto de sensores unidos à roupa do paciente, facilitará o acompanhamento médico de pacientes, especialmente de idosos. Para manter aspectos ubíquos no projeto, foi utilizada uma camiseta com o conjunto de sensores e uma placa controladora com bateria para fornecer informações do paciente periodicamente. (RIOS; BEZERRA, 2014)

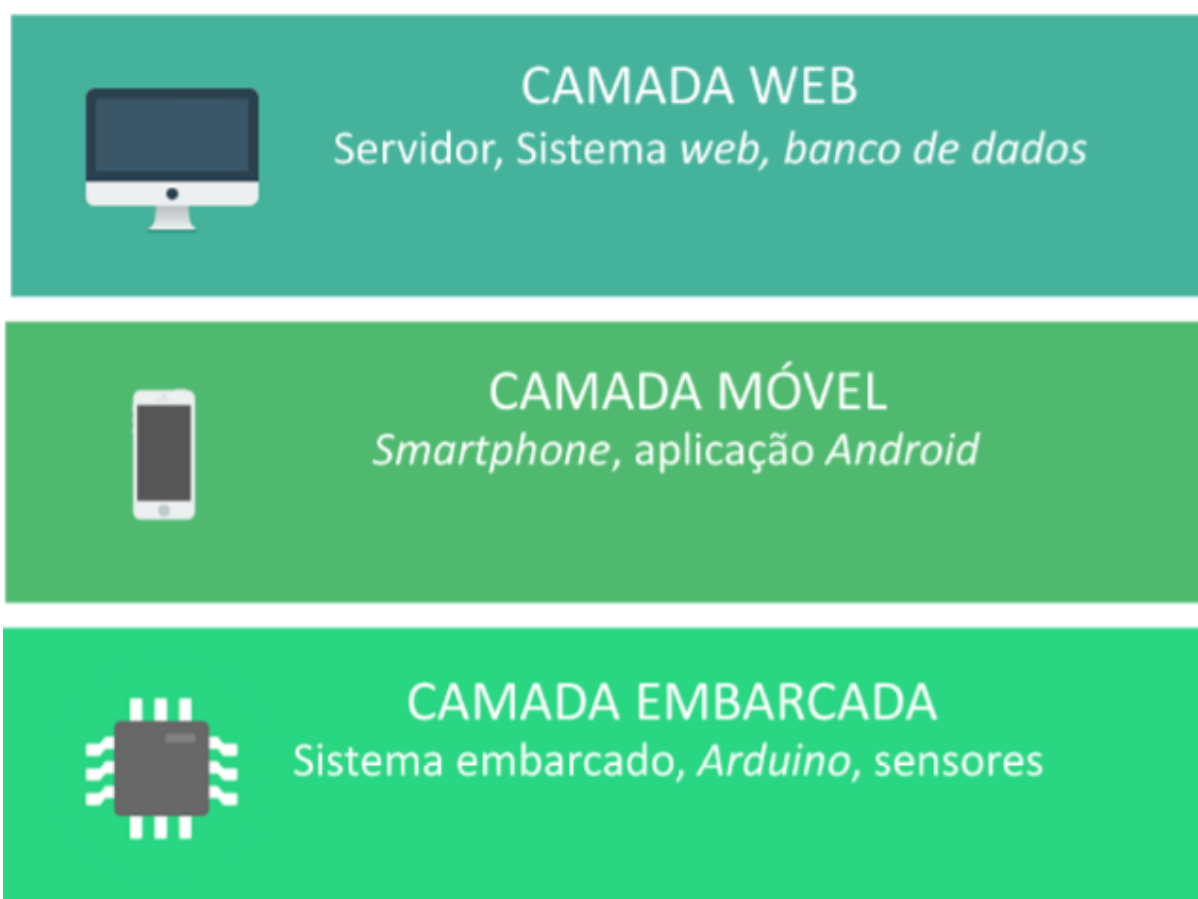
A utilização de uma tecnologia embarcada vestível, impulsiona o objetivo principal do trabalho no monitoramento do estado de saúde de idosos que estejam enfrentando algum tipo de tratamento ou até mesmo em asilos, enviando os dados

lidos através da internet para uma interface simples, intuitiva e que possam ser lidos por um profissional através de uma plataforma *web*. (RIOS; BEZERRA, 2014)

3.2.1 Arquitetura e Implementação

O sistema de monitoramento WHMS4 (*Wearable Health Monitoring Simple Secure and Scalable System* ou Sistema de Monitoramento de Saúde Vestível Simples Seguro e Escalável) desenvolvido pelo autor é uma composição de três subsistemas integrados, sendo eles representados pela Figura 5.

Figura 5 - Arquitetura preliminar do WHMS4



Adaptado de: RIOS; BEZERRA (2014)

Cada camada possui uma responsabilidade distinta, sendo que a “Camada Embarcada” corresponde ao sistema embarcado vestível que possui os sensores distribuídos para coleta de sinais vitais. Na “Camada Móvel” se concentra o aplicativo que será utilizado nos *smartphones* além do envio dos dados para a “Camada Web” que por sua vez apresenta uma aplicação como sistema final para os usuários para

visualização e gerenciamento das informações obtidas pelos sensores. (RIOS; BEZERRA, 2014)

Dentro da estrutura do sistema *web* desenvolvido, os usuários são cadastrados e possuem diferentes perfis de acesso, sendo eles médico, auxiliar/enfermeiro e administrador. Os usuários com perfil de médico são responsáveis por definir o tratamento do paciente e cadastrar os parâmetros que serão utilizados como comparação com os dados fornecidos pela leitura dos sensores e com este comparativo é possível avaliar a situação de cada um. Além dos valores obtidos pela leitura do conjunto de sensores, o sistema *web* também recebe a identificação do paciente e sua localização. (RIOS; BEZERRA, 2014)

A placa controladora escolhida pelo autor para interpretação da leitura dos sensores é da plataforma *Arduino Pro Micro*, que se trata de um modelo pequeno com dimensões de 3,30 x 1,77 cm. e junto com um módulo *Bluetooth* é responsável pela comunicação do microcontrolador com a aplicação móvel. A leitura e o processamento dos dados fornecidos pelos sensores são realizados de maneira contínua através do software desenvolvido para o sistema embarcado. (RIOS; BEZERRA, 2014)

3.2.2 Aplicativo

A aplicação móvel desenvolvida neste projeto é exclusivamente para o sistema operacional Android e funciona como agente intermediário enviando os dados entre o sistema embarcado e o sistema *web* através de recursos nativos como *Wifi*, *Bluetooth* e GPS. A comunicação entre o sistema embarcado e o aplicativo ocorre através de pareamento via *Bluetooth*, em que a aplicação recebe os dados enviados pelo Arduino e posteriormente os envia para o sistema *web* através da interface *Wifi*. (RIOS; BEZERRA, 2014)

O sistema *web* é a camada mais complexa do projeto, pois é responsável por persistir os dados do paciente e dos sinais vitais na base de dados além de fornecer um dashboard para visualização de informações gerais sobre os pacientes monitorados. Foi escolhida a linguagem de programação PHP para confecção da camada *web* junto com um sistema de gerenciamento de banco de dados com a ferramenta *MySQL*. (RIOS; BEZERRA, 2014)

3.3 UbHeart - Um Modelo para Monitoramento de Sinais Vitais do Coração Baseado em Ciência da Situação e Computação Ubíqua

Visando contemplar as premissas básicas da computação ubíqua, o autor do trabalho “*UbHeart - Um Modelo para Monitoramento de Sinais Vitais do Coração Baseado em Ciência da Situação e Computação Ubíqua*” desenvolveu um projeto para monitoramento de sinais vitais de pacientes com insuficiência cardíaca, com uma arquitetura que possibilite acesso a computação em qualquer momento e lugar para medir sinais vitais a todo instante, além de transformar a captação dos sinais de maneira confortável ao usuário do sistema. (ROCHA, 2016)

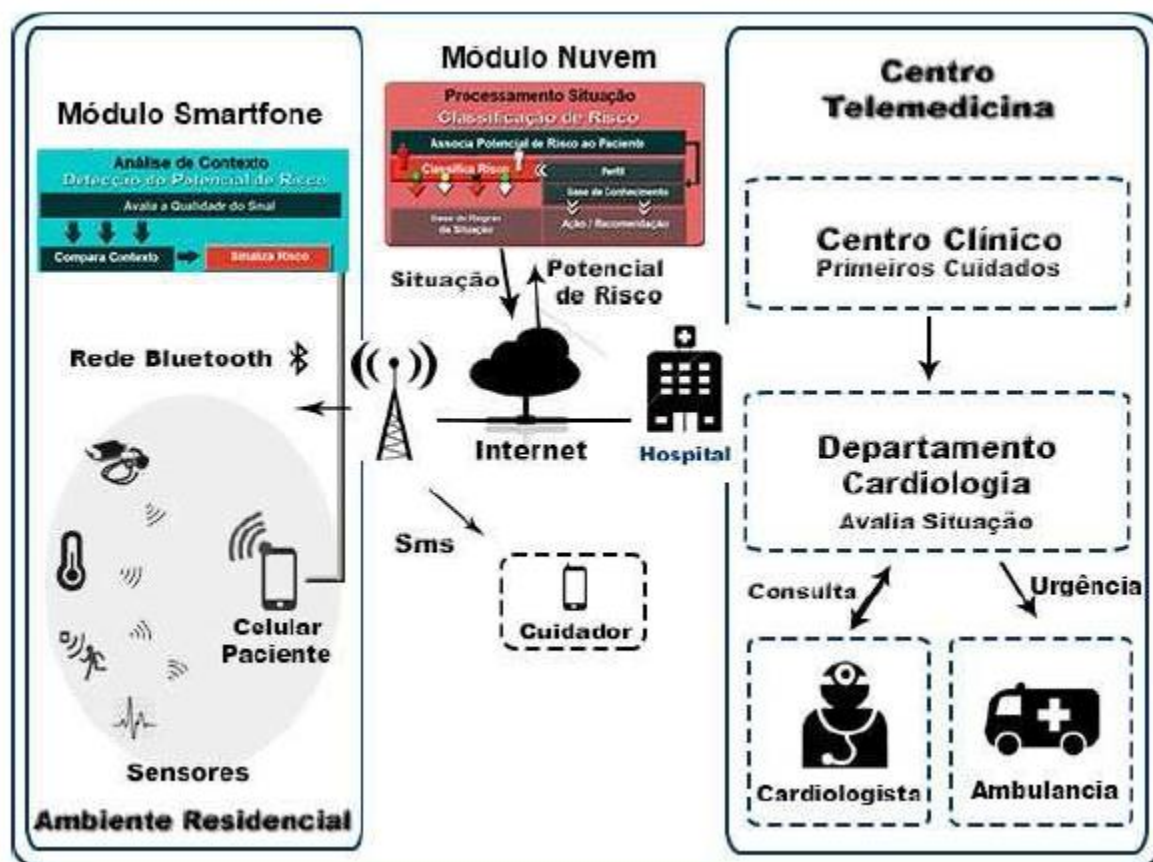
3.3.1 Arquitetura e Implementação

O sistema do *UbHeart* consiste em receber os sinais vitais empregando sensores em área corporal conectados a uma rede *Bluetooth* que, posteriormente, são utilizados para montar um histórico de saúde do paciente e enviar ao hospital através da *Internet*. O *smarthphone* do próprio paciente é utilizado para informar o perfil do paciente e receber os dados lidos pelos sensores. (ROCHA, 2016)

Os sinais vitais enviados pelo *smarthphone* do paciente para a *Internet* também são fonte de dados para que um sistema analise a situação de saúde baseada no histórico e na degradação dos sinais vitais e assim, com essas informações chegando ao hospital é possível que um médico responsável garanta uma assistência mais rápida ao paciente. (ROCHA, 2016)

O *UbHeart* foi um sistema desenvolvido em módulos e cada um com uma responsabilidade única. Dentro do Módulo Nuvem existe um componente chamado Processamento de Situação, em que são realizados os processamentos dos sinais vitais, e é responsável pela classificação de risco e definição da situação de saúde do paciente, também pelas regras e ações do sistema. A Figura 6 ilustra a visão da arquitetura geral do sistema. (ROCHA, 2016)

Figura 6 - Arquitetura do UbHeart



Adaptado de: ROCHA (2016)

A classificação de risco do paciente é realizada em três níveis, sendo alto, médio e baixo e de acordo com o nível são realizadas ações que podem variar de recomendações ao paciente até mesmo acionar seu cuidador. (ROCHA, 2016)

Foi utilizado a ferramenta de banco de dados *SQL Lite* junto com um sistema *Android* para desenvolver a implementação do módulo de análise do potencial de risco, o módulo de processamento da situação foi simulado através do sistema operacional *Gnu/Linux Debian* com banco de dados *Postgres* e por fim, o módulo de regras de ação e determinação de comportamento do sistema foi desenvolvido na linguagem *Python*. (ROCHA, 2016)

3.3.2 Aplicativo

O protótipo do aplicativo do *UbHeart* desenvolvido na plataforma *Android* e nele é possível definir o perfil do paciente, chamar o cuidador, informar pessoais com peso,

altura, idade, sexo do paciente e verificar status e o nível de risco atual. (ROCHA, 2016)

3.4 MobileVJ: A mobile app for a novel wearable human sensing system

O trabalho “*MobileVJ: A mobile app for a novel wearable human sensing system*” ou “MobileVJ: Um aplicativo móvel para um novo sistema wearable de detecção humana” (ALEXEENKO, 2016) propões a criação de um sistema de monitoramento de sinais vitais em tempo real aproveitando conceitos de arquitetura M2M (*Machine to Machine*), enviando os dados para análise de profissionais da saúde. (ALEXEENKO, 2016)

O propósito do sistema *MobileVJ* é monitorar sinais vitais em bombeiros e militares durante suas missões. O conjunto de sensores acoplados ao usuário se comunica através da tecnologia *Bluetooth* com um *smarthphone* e fornece dados para um aplicativo *Android* que os envia para um servidor que, por sua vez, os armazena em uma base de dados para serem visualizados pelo analista profissional da saúde. (ALEXEENKO, 2016)

3.4.1 Arquitetura e Implementação

O conjunto de sensores escolhidos pelo autor do trabalho para monitoramento dos sinais vitais se chama *VitalJacket*, visto na Figura 7, que é um dispositivo conectado a uma camiseta e possibilita 72 horas seguidas de monitoramento de diferentes sinais vitais como ECG, temperatura, respiração, postura, saturação de oxigênio. (ALEXEENKO, 2016)

Figura 7 - Kit VitalJacket



Adaptado de: ALEXEENKO (2016)

O monitoramento dos sinais vitais só acontece se o dispositivo móvel detectar que existe conexão com a Internet e que foi detectado um pareamento com o conjunto de sensores via *Bluetooth*. As informações obtidas pelos sensores são enviadas no formato *JSON* para um servidor e todas as mensagens trafegadas no aplicativo são armazenadas em uma base de dados *noSQL* e os dados que mais relevantes para o monitoramento são exibidos em tempo real para o profissional da saúde que estiver acompanhando a leitura. (ALEXEENKO, 2016)

Para auxiliar o desenvolvimento de uma aplicação de tempo real, foi feito uso da plataforma *RabbitMQ* dentro do servidor para gerenciamento das mensagens enviadas pelo sistema. Esta plataforma permite o tráfego de mensagens através de um sistema de produtor/consumidor, neste caso, o consumidor das informações seria o profissional da saúde que acompanha os dados dos sensores por um computador, se conectando diretamente no servidor, e o consumidor seria o militar que está sendo monitorado. (ALEXEENKO, 2016)

3.4.2 Aplicativo

Para a comunicação com o conjunto de sensores foi utilizada a tecnologia *Bluetooth* que via pareamento, se conecta ao aplicativo escrito na plataforma Android. O aplicativo permite a escolha de quais sensores o usuário deseja receber a leitura, inclusive de diferentes conjuntos. (ALEXEENKO, 2016)

3.5 Análise Comparativa

A leitura e compreensão dos trabalhos relacionados resultou em uma comparação entre cada um que será utilizada para compor o modelo proposto descrito no próximo capítulo.

Ao analisar o trabalho “*Soluções de Baixo Custo para Implementação de Internet das Coisas na Área da Saúde*” (JUCHEM, 2015) observou-se que o autor obteve um resultado dentro do seu esperado, pois mesmo centralizando seu projeto para o uso em maternidades, foi constatado, junto com um profissional da área da saúde, que seu uso pode ser ampliado para qualquer paciente de qualquer idade ou quadro clínico já que o modelo é de fácil manuseio e interpretação do usuário final.

O trabalho “*WHMS4 - Um Sistema de Monitoramento Remoto de Pacientes de Forma Integrada, Flexível, Escalável e com Baixo Custo*” (RIOS; BEZERRA, 2014) compõe uma solução bem estruturada, pois foi construída com uma divisão de camadas bem clara e cada camada possui uma responsabilidade única. O autor do projeto também se preocupou com a realidade de países emergentes e a dificuldade de investimentos na área da saúde para priorizar uma solução de baixo custo, tanto a nível de *hardware* quanto ao uso de *software* livre, em comparação com as soluções já existentes. O sistema desenvolvido também é configurável, pois permite parametrização de sinais vitais para cada paciente e teve resultados satisfatórios em testes de escalabilidade e testes de carga. Outro ponto positivo é a definição de perfis de acesso ao sistema que impede que o usuário execute uma função ou leia um dado do qual não deva ter conhecimento.

No projeto *UbHeart* (ROCHA, 2016) notou-se que a preocupação maior do autor é a análise de situação que o sistema efetua, pois nele foi empregado maior esforço e complexidade para cálculo de padrões e índice de degradação de taxa de comprometimento computada após medição dos sinais vitais. A utilização de um conjunto de sensores para a leitura em si dos sinais vitais não fez parte do trabalho, sendo utilizada uma base de dados para efetuar simulações com valores de sinais estabelecidos pela Sociedade Brasileira de Cardiologia.

O sistema *MobileVJ* (ALEXEENKO, 2016) tem um diferencial que permite a conexão de múltiplos conjuntos de sensores junto ao aplicativo, além de ter desempenhado um bom resultado no monitoramento de pessoas em repouso e durante exercícios de levantamento de pesos, mostrando a diferença de sinais vitais

em cada situação. A maneira como foi desenvolvida a arquitetura do sistema mostrou que, quando conectado em uma rede EDGE, o fluxo de mensagens não é constante devido à baixa velocidade da rede, mas em demais conexões o sistema atendeu ao seu propósito. Uma brecha deixada pelo projeto é a falta de uma interface amigável para acompanhamento dos sinais vitais pelos profissionais da saúde, seja em um aplicativo, um sistema *web* ou *mobile*.

A seguir, a Tabela 2 tem como objetivo apresentar um comparativo entre os trabalhos relacionados, em que foram levados em consideração aspectos como usabilidade dos protótipos, tecnologias empregadas e funcionalidades.

Tabela 2 - Avaliação dos Trabalho Relacionados

	Soluções de Baixo Custo para Implementação de Internet das Coisas na Área da Saúde (JUCHEM, 2015)	WHMS4 (RIOS; BEZERRA, 2014)	UbHeart (ROCHA, 2016)	MobileVJ (ALEXEENKO, 2016)
Meios de acesso	Dispositivo móvel	Dispositivo móvel e computador	Dispositivo móvel	Dispositivo móvel e computador
Interface com o usuário	Oferecido pelo dispositivo	Oferecido pelo dispositivo	Oferecido pelo dispositivo	Oferecido pelo dispositivo ou teclado e mouse
Tecnologias de programação	Arduino, Objective-C e Intel XDK	Arduino, Android, PHP, MySQL	Android, Linux, Postgres, Python, SQL Lite	Android, RabbitMQ, MongoDB
Conjunto de sensores	e-Health	Sensor de temperatura e batimentos cardíacos	Não possui	VitalJacket
Modelo de comunicação com conjunto de sensores	SPI	Wifi e Bluetooth	Não possui	Bluetooth
Interface para o profissional da área da saúde	Sim, dentro do aplicativo	Sim, em uma plataforma Web	Sim, dentro do aplicativo	Sim, se conectando ao servidor RabbitMQ
Funcionalidades	Monitorar frequência cardíaca, respiratória, temperatura corporal e saturação de oxigênio no sangue do paciente.	Cadastrar médico, paciente e enfermeiro; monitorar a temperatura, os batimentos cardíacos e a localização do paciente; verificar todos os médicos e pacientes cadastrados.	Monitorar sinais vitais e classificar grau de risco do paciente; enviar informações do paciente ao hospital ou ao médico.	Conectar-se a múltiplos conjuntos de sensores; enviar leitura em tempo real

Fonte: Elaborada pelo autor

Observando as lacunas deixadas por cada trabalho, como a ausência de uma interface para visualização do histórico dos sinais vitais do paciente no próprio aplicativo, um filtro para eliminar ruídos indesejáveis do sinal e pequena exploração dos sensores de sinais vitais, o modelo apresentado a seguir visa entregar uma solução de leitura de sinais vitais para dispositivos móveis que permitirá ao profissional da saúde efetuar exames em paciente, como será mostrado no próximo capítulo.

4. MODELO PROPOSTO

O modelo proposto neste trabalho tem como objetivo principal auxiliar profissionais da área da saúde no diagnóstico de anomalias cardíacas, no sistema nervoso ou muscular de pacientes em qualquer situação de risco, protegendo seu estado de saúde e permitindo que um médico analise seus sinais vitais ao longo do tempo para sugerir um tratamento caso necessário.

Avaliando os trabalhos relacionados que serviram como guia para a construção do modelo, será proposto aqui uma forma de obter os sinais vitais de um paciente, além de apenas os batimentos cardíacos através do *kit* de desenvolvimento *BITalino* com a utilização de um *smartphone* que possibilitará a comunicação com os sensores e um aplicativo que será construído com finalidade de ser um instrumento de envio dos dados para um servidor de processamento.

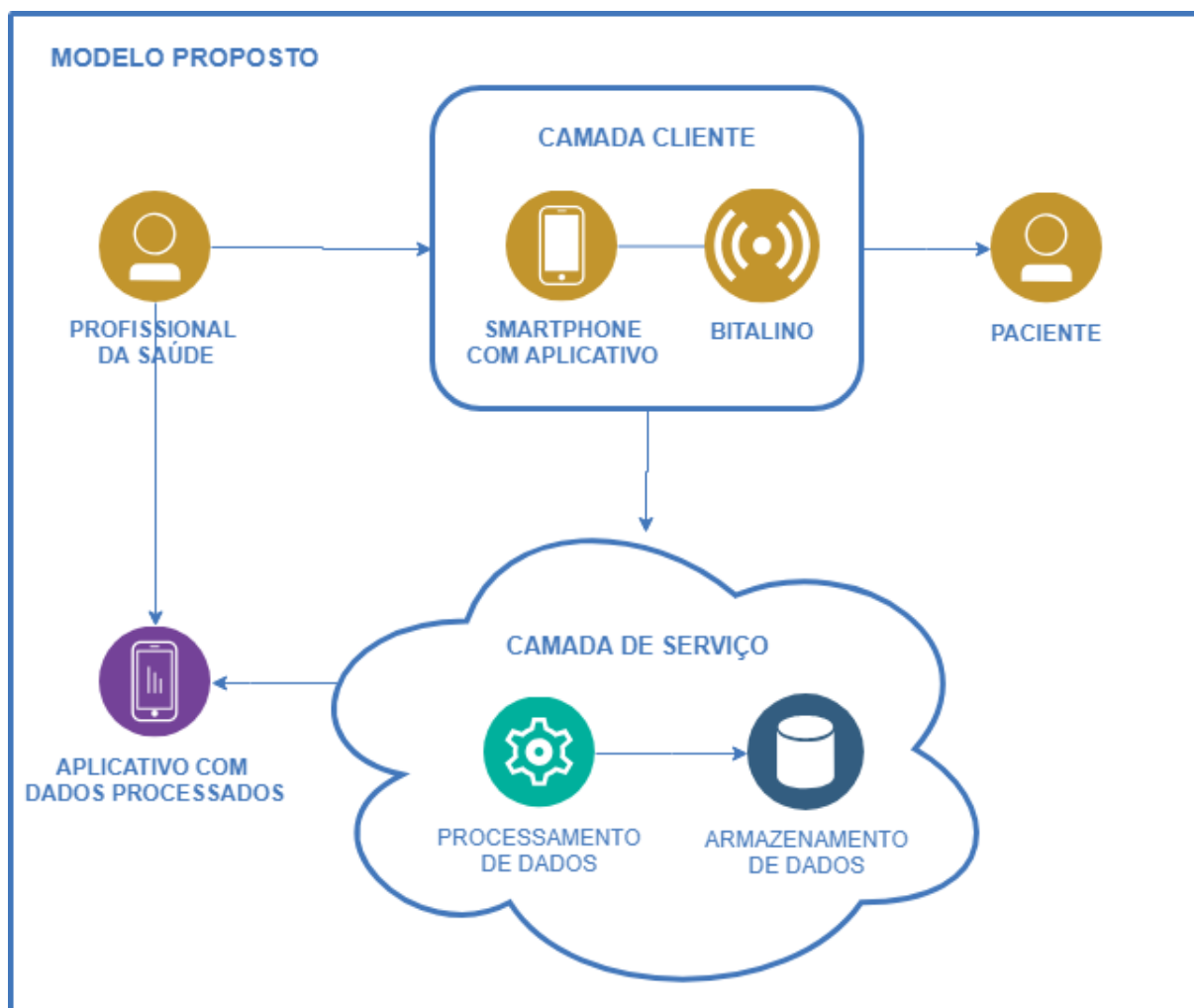
A diversidade de sensores disponíveis no *BITalino* será essencial para avaliar problemas nervosos ou musculares com o exame de eletromiografia e detectar possíveis problemas cardíacos com os sensores de eletrocardiografia. A interface gráfica que será construída dentro do próprio aplicativo terá a intenção de facilitar o procedimento de avaliação dos exames, visto que não será necessário que o profissional acesse um computador ou outro dispositivo para averiguar os sinais do paciente.

4.1 Visão Geral do Modelo

De maneira geral o profissional da saúde deverá possuir um *smartphone* que usará para parear via *Bluetooth* o conjunto de desenvolvimento do *BITalino*. O aplicativo desenvolvido irá realizar o pareamento dos dispositivos, receberá os dados de leitura dos sensores e os enviará para um servidor que realizará o processamento do sinal recebido e posteriormente permitirá que um médico avalie as informações no próprio aplicativo. A Figura 8 ilustra o fluxo de utilização deste modelo descrito.

O modelo que será desenvolvido não tem como objetivo a construção de *hardware* dedicado para leitura dos sinais vitais já que o *kit* de desenvolvimento do *BITalino* oferece este suporte.

Figura 8 - Visão geral do modelo proposto



Fonte: Elaborado pelo autor

4.2 Camada Cliente

A camada cliente corresponderá ao aplicativo que será desenvolvido e utilizado num *smartphone* juntamente com o *kit* do *BITalino*.

O aplicativo será construído em plataforma híbrida para possibilitar compatibilidade com os principais sistemas operacionais de dispositivos móveis do mercado e terá como finalidade sincronizar via *Bluetooth* com os sensores do *kit*, receber seus dados e os enviar para a camada de processamento assim que o dispositivo móvel estiver conectado à *internet*.

Além do envio dos dados, o aplicativo permitirá o cadastro de pacientes e será o meio pelo qual o profissional da saúde poderá visualizar por uma interface gráfica

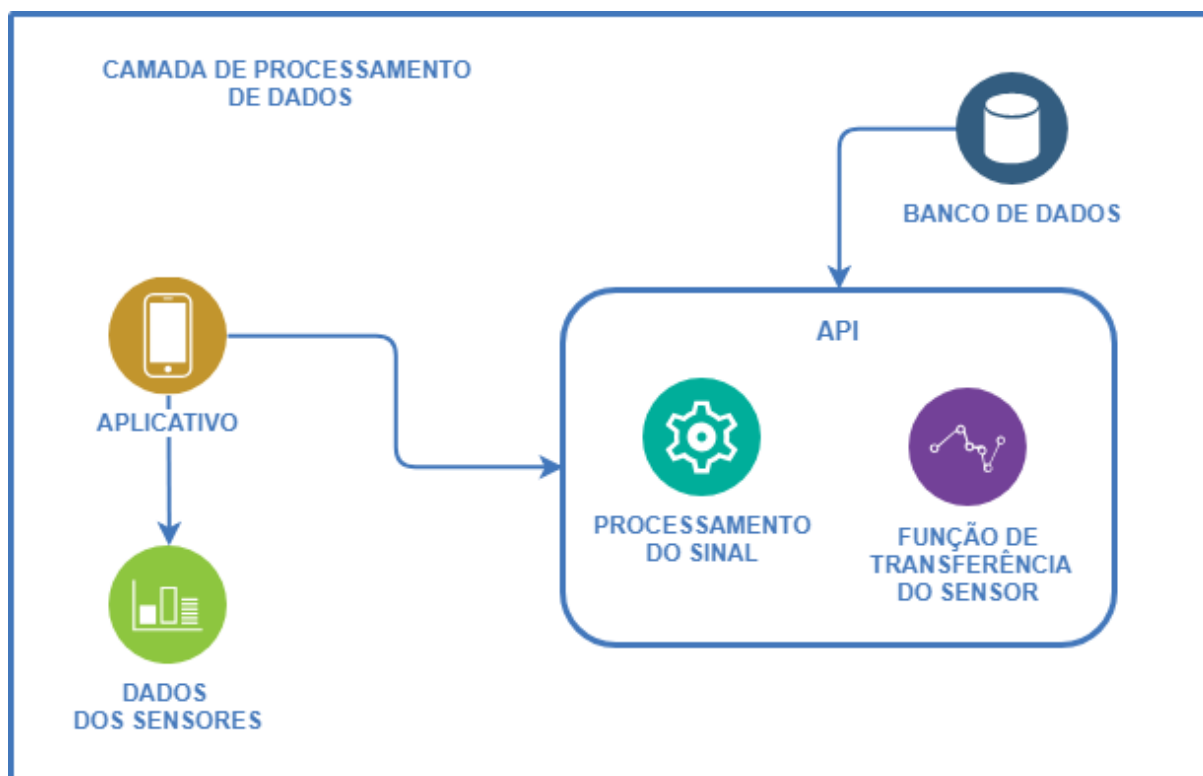
os sinais vitais obtidos e já processados de todos os pacientes que estão sob seus cuidados, possibilitando o agrupamento de exames por paciente.

4.3 Camada de Serviço

O processamento dos dados recebidos pela camada cliente será realizado em uma API (*Application Programming Interface* - Interface de Programação de Aplicativos) hospedada em um servidor e será responsável pelo processamento dos sinais e a comunicação com o banco de dados que armazenará as informações do paciente, caracterizando assim a arquitetura como um modelo *Service-oriented architecture* (SOA) em que as principais funcionalidades são disponibilizadas no formato de serviços e como componentes reutilizáveis. (ZANUZ; FILIPPETTO; CRESPO, 2007)

Todos os dados processados serão armazenados em um banco de dados também hospedado em um servidor para gerar um histórico de informações do paciente que está sob ação dos exames e permitir que um médico avalie as modificações dos seus sinais vitais ao longo do tempo. A Figura 9 mostra um diagrama deste processo.

Figura 9 - Camada de Processamento de dados



Fonte: Elaborado pelo autor

Os sinais obtidos pelos sensores do *BITalino* não são submetidos a nenhum condicionamento de sinal quando interpretados pelo microcontrolador do kit, portanto a camada de processamento de serviço deverá ser responsável por efetuar o condicionamento e a filtragem necessária para cada sinal, além de aplicar a função de transferência de cada sensor que produziu os dados para exibir as unidades corretas de cada medida efetuada. A filtragem do sinal será realizada utilizando um filtro Butterworth em *software*.

4.3.1 Processamento dos Sinais

A filtragem dos sinais dos sensores implementará a aproximação do filtro *Butterworth*, pois segundo Rathke (2011, p. 59) trata-se de uma aproximação monotônica que tem como função eliminar sinais de alta frequência, superiores ao sinal de interesse.

Conforme Nilsson e Riedel (1999, p. 418) a função de transferência de um filtro Butterworth passa-baixa analógico e de ganho unitário é expressa por

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (\omega/\omega_c)^{2n}}} \quad (1)$$

Nesta equação temos:

- n corresponde a ordem do filtro;
- A frequência de corte é simbolizada por ω_c e expressa em rad/s;
- O denominador é praticamente igual a 1 quando $\omega < \omega_c$;

Seguindo o raciocínio expressado por Nilsson e Riedel (1999, p. 418), e partindo do princípio que N é um número complexo, $|N|^2 = NN^*$, onde N^* é o complexo conjugado de N . Portanto a função de transferência pode ser escrita como

$$|H(j\omega)|^2 = H(j\omega)H(-j\omega) \quad (2)$$

Como $s^2 = j\omega$ e $s^2 = -\omega^2$, temos

$$|H(j\omega)|^2 = H(s)H(-s) \quad (3)$$

$$|H(j\omega)|^2 = \frac{1}{1 + \omega^{2n}} = \frac{1}{1 + (\omega^2)^n} = \frac{1}{1 + (-s^2)^n} = \frac{1}{1 + (-1)^n s^{2n}} \quad (4)$$

Ou ainda

$$H(s)H(-s) = \frac{1}{1 + (-1)^n s^{2n}} \quad (5)$$

A determinação do valor de $H(s)$ para um dado valor de n consiste em determinar as raízes do polinômio $1 + (-1)^n s^{2n} = 0$. No caso de um filtro de segunda ordem, $n = 2$, é necessário encontrar as raízes do polinômio $1 + (-1)^2 s^4 = 0$ e com o auxílio de um programa obtêm-se o resultado da Equação 6. (NILSSON; RIEDEL, 1999)

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1} \quad (6)$$

Os polinômios de *Butterworth* até oitava ordem já normalizados para $\omega_c = 1 \text{ rad/s}$ são encontrados na Tabela 3.

Tabela 3 - Polinômios de Butterworth

#	Polinômio de Butterworth de ordem n
1	$(s + 1)$
2	$(s^2 + \sqrt{2}s + 1)$
3	$(s + 1)(s^2 + s + 1)$
4	$(s^2 + 0,765s + 1)(s^2 + 1,848s + 1)$
5	$(s + 1)(s^2 + 0,618s + 1)(s^2 + 0,618s + 1)$
6	$(s^2 + 0,518s + 1)(s^2 + \sqrt{2}s + 1)(s^2 + 1,932s + 1)$
7	$(s + 1)(s^2 + 0,445s + 1)(s^2 + 1,247s + 1)(s^2 + 1,802s + 1)$
8	$(s^2 + 0,390s + 1)(s^2 + 1,111s + 1)(s^2 + 1,663s + 1)(s^2 + 1,962s + 1)$

Fonte: Adaptado de NILSSON; RIEDEL, 1999

De acordo com Nilsson e Riedel (1999), quanto maior a ordem do filtro *Butterworth*, mais as suas características se aproximam de um filtro a passa baixa ideal, ou seja, para altos valores de n o ganho permanece constante e próxima da frequência de corte e o valor da ordem pode ser determinado através da seguinte equação:

$$n = \frac{-0,05A_s}{\log_{10}(\omega_s/\omega_p)} \quad (7)$$

Desta equação temos as seguintes variáveis:

- A_s corresponde ao ganho em decibéis;
- ω_s representa a frequência;
- ω_p representa a frequência de corte.

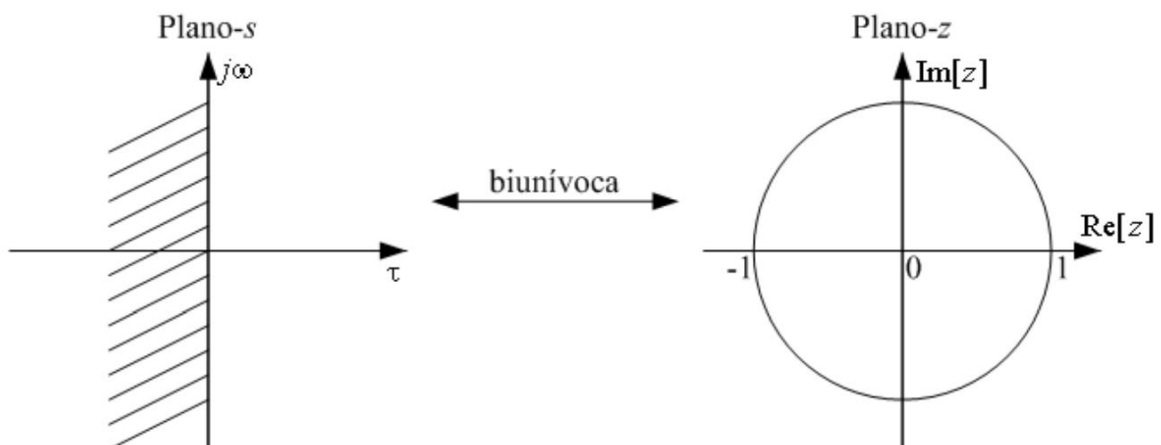
Outra maneira de descobrir a ordem do filtro é através da equação mostrada por Schaumann e Valkenburg (2001), em que α_{min} corresponde a atenuação mínima e α_{max} a atenuação máxima desejada na filtragem.

$$n = \frac{\log[(10^{\frac{\alpha_{min}}{10}} - 1)/(10^{\frac{\alpha_{max}}{10}} - 1)]}{2\log_{10}(\omega_s/\omega_p)} \quad (8)$$

De acordo com Moreira e Silva (2014), o método da transformação bilinear permite a transformação de uma função de transferência de tempo contínuo em uma função de transferência em tempo discreto e tal método será necessário para implementar a filtragem digital do sinal.

A transformação bilinear consiste em uma transformação algébrica entre as variáveis s e z mapeado o eixo, da unidade imaginária $j, j\Omega$ do plano- s com frequência Ω em uma completa revolução da circunferência unitária no plano- z e com frequência ω como ilustrado na Figura 10. (FERNANDES, 2006)

Figura 10 - Mapeamento das frequências do plano- s para o plano- z utilizando transformação bilinear.



Segundo Chaparro (2011) a transformação bilinear mapeia todo o plano s em um plano z , do mesmo modo, um filtro analógico estável com polos no plano s gerará um filtro discreto também estável:

- z para plano s :

$$s = K \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad K = \frac{2}{T}$$

- s para o plano z :

$$z = \frac{1 + s/K}{1 - s/K}$$

(9)

A frequência analógica Ω e a frequência discreta ω de acordo com a transformação bilinear são relacionadas por: (CHAPARRO, 2011)

$$\Omega = \frac{2}{T} \tan \left(\frac{\omega}{2} \right)$$

(10)

Ou em relação a ω ,

$$\omega = 2 \arctan \left(\frac{\Omega T}{2} \right)$$

(11)

Conforme Lathi (2006, p. 513), a função tangente causa uma compressão na frequência, devido sua não linearidade. Esta compressão é chamada de *warping* em frequência e é responsável por distorcer a transformação, porém ajustando a variável T uma frequência pode ser discretizada garantindo ser o mais precisa possível.

No método da transformação bilinear todo o espectro de analógico de frequências $-\infty < \Omega < \infty$ é organizado em uma volta sobre a circunferência unitária em $-\pi \leq \omega < \pi$, logo as frequências digitais ω são restritas ao intervalo $-\pi \leq \omega < \pi$, deste modo as especificações digitais devem ser normalizadas para o intervalo $[-\pi, \pi)$. (DINIZ, 2014)

De acordo com a documentação dos sensores do *kit* de desenvolvimento do *BITalino*, foi construída a Tabela 4 para mostrar de maneira clara a faixa de frequência de operação de cada sensor, junto com sua função de transferência, que serão necessárias para cálculo dos filtros passa-faixa.

Tabela 4 - Características dos Sensores do *BITalino*

Sensor	Faixa de Frequência (Hz)	Função de Transferência
<i>Eletrmiografia (EMG)</i>	25 – 480	$EMG(V) = \frac{\left(\frac{ADC}{2^n} - \frac{1}{2}\right) \times VCC}{G_{EMG}}$
<i>Eletrocardiografia (ECG)</i>	0,5 – 40	$ECG(V) = \frac{\left(\frac{ADC}{2^n} - \frac{1}{2}\right) \times VCC}{G_{ECG}}$
<i>Eletroencefalografia (EEG)</i>	0,8 – 48	$EEG(V) = \frac{\left(\frac{ADC}{2^n} - \frac{1}{2}\right) \times VCC}{G_{EEG}}$
<i>Resposta Galvânica da Pele (EDA)</i>	0 – 2,8	$EDA(\mu S) = \frac{\left(\frac{ADC}{2^n}\right) \times VCC}{0,132}$

Fonte: Elaborado pelo autor

Em relação as funções de transferência, temos as seguintes variáveis:

- ADC corresponde ao valor obtido diretamente pelo sensor;
- VCC é a tensão de operação de 3.3V para todos os sensores;
- n é o número de bits do canal, em que os quatro primeiros canais do *BITalino* usam 10 bits de resolução ($n = 10$), já os dois últimos usam resolução de 6 bits ($n = 6$);
- G_{EMG} corresponde ao ganho do sensor que é equivalente a 1009;
- G_{ECG} corresponde ao ganho do sensor que é equivalente a 1100;
- G_{EEG} corresponde ao ganho do sensor que é equivalente a 40000;

Após a filtragem do sinal e a aplicação da função de transferência haverá uma conversão, no nível da camada de serviço, dos dados para o formato XML (*Extensible Markup Language* - Linguagem de Marcação Extensível) ou ADL (*Archetype Definition Language* - Linguagem de Definição de Arquétipo) compatíveis com o padrão *openEHR*, para criação de registros eletrônicos de saúde modulares e interoperáveis e armazenados na base da dados para exportação quando solicitado pelo usuário.

5. METODOLOGIA DE AVALIAÇÃO

A avaliação do modelo proposto terá como objetivo confirmar a possibilidade de sua utilização como protótipo e acontecerá através das seguintes etapas:

- a) Construção de um aplicativo capaz de obter os dados de leitura dos sensores do *BITalino* e enviar estes dados para processamento.
- b) Testes funcionais envolvendo os módulos de toda a aplicação com finalidade de avaliar a estrutura projetada para suportar o protótipo;
- c) Testes de usabilidade do aplicativo desenvolvido;
- d) Validação do modelo clínico gerado junto ao *EHR Server*.

Conforme Rios e Bezerra (2014) os testes funcionais tem por finalidade confirmar a possibilidade de utilizar o protótipo conforme seus requisitos, portanto seus módulos serão testados num primeiro momento separadamente utilizando testes unitários.

Além do teste funcional, a avaliação de usabilidade tem como propósito traduzir a facilidade de uso de um objeto de diversos tipos, já que a facilidade do uso do protótipo pode ser o seu real motivo de destaque em relação a um concorrente. Assim é possível detectar as dificuldades enfrentadas pelo utilizador do protótipo e diferenças entre o notar a diferença do modo de utilização na visão do programador e o comportamento real do utilizador do sistema. (REIS, 2017)

Durante o teste de usabilidade, afim de garantir uma variedade de dados para análise e conversão, serão executados testes em pelo menos cinco pessoas voluntárias de idade e sexo diferentes, o que contribuirá com o enriquecimento da base de dados da aplicação.

5.1 Implementação

Partindo da disponibilidade de diversas APIs (*Application Programming Interface* - Interface de Programação de Aplicativos) no site do BITalino, escritas pela comunidade e em diversas linguagens de programação, foi selecionada para realização de testes de comunicação com o *kit* de desenvolvimento a API escrita na linguagem *C#*, pois com ela seria possível construir um aplicativo utilizando a tecnologia *Xamarin* da *Microsoft*.

O *framework Xamarin* permite que através de uma única linguagem de programação e um único IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) sejam escritas aplicações multi plataformas, ou seja, destinadas a mais de um sistema operacional por meio do *framework .NET*. A vantagem do *Xamarin* está na performance de desenvolvimento, já que não é necessário o aprendizado de linguagens de programação nativas de cada sistema operacional para qual a aplicação será fornecida. (MARTINS, 2018)

Contudo, ao efetuar testes de comunicação entre o *kit* de desenvolvimento *BITalino* e o aplicativo escrito em *Xamarin* para o sistema operacional Android, como prova de conceito, eram obtidos erros em tempo de execução no momento em que havia uma tentativa de conexão *Bluetooth* com o *BITalino* o que motivou a construção de um aplicativo na linguagem *Java* utilizando o *Android SDK* e o ambiente de desenvolvimento *Android Studio*.

Todas as operações de leitura, escrita e manipulação de dados como filtragem e aplicação da função de transferência dos sensores são realizadas com apoio de uma *WebApi* desenvolvida na linguagem *.NET Core*, pois assim o núcleo do modelo fica descentralizado possibilitando que outras aplicações, como um *web site* ou outro aplicativo escrito em um linguagem diferente, também consumam a lógica desenvolvida na *WebApi*.

5.1.2 Android

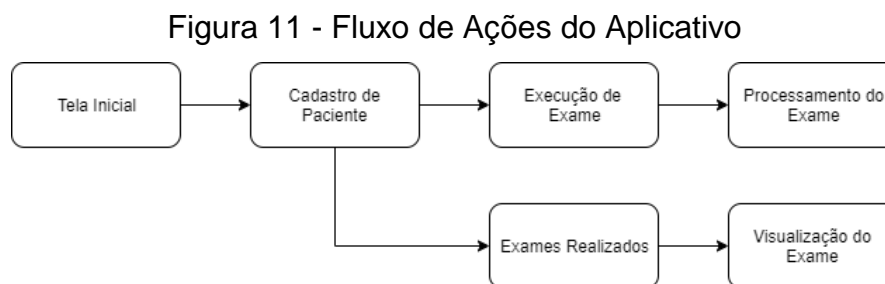
Os primeiros passos para confecção do aplicativo foram direcionados a comunicação entre o *BITalino* e a API disponibilizada no repositório *BITalinoWorld* na plataforma de versionamento de código *Github*. Esta API foi desenvolvida utilizando a versão 23 do *Android SDK* e possui uma breve documentação que possibilita a

implementação de todas as funcionalidades de um dispositivo *BITalino*, tais como comunicação, aquisição de dados e estado do dispositivo.

O *Android SDK* é consiste em um software que é suportado nos sistemas operacionais *Windows*, *Mac OS* e *Linux* e possui todas as ferramentas utilitárias, *API* na linguagem *Java* e emuladores que simulam o celular para construção de aplicativos *Android* (LECHETA, 2015)

A critérios de disponibilidade, os testes de desenvolvimento do aplicativo foram realizados em um dispositivo da empresa *LG* cujo modelo é *K10 LTE* e opera o sistema operacional *Android 6.0*.

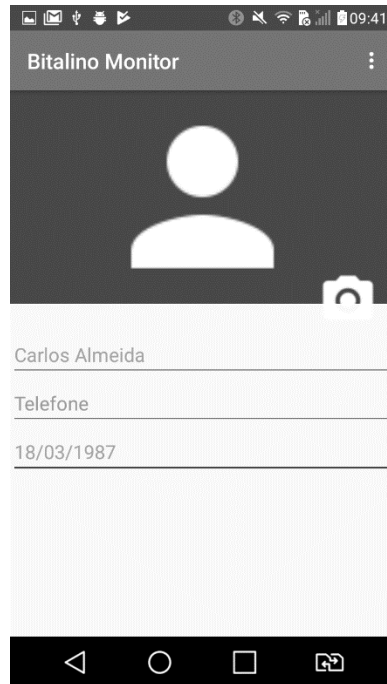
Com a comunicação entre o aplicativo e o *BITalino* atuando, iniciou-se a construção das telas do sistema e do fluxo de navegação que é ilustrado pela Figura 11.



Fonte: Elaborado pelo autor

Das operações do *BITalino*, foram implementadas a aquisição de dados via exame de eletrocardiograma, resposta galvânica da pele, eletromiografia e eletroencefalograma e os testes dos exames foram realizados conforme orientação da posição dos eletrodos no *datahseet* de cada sensor. Todas as operações são previamente atreladas a um paciente que é incluído no sistema através da tela de cadastro de paciente, mostrada pela Figura 12, em que é possível informar um nome, a data de nascimento, o telefone e inserir uma foto do paciente. Esta tela pode facilmente conter os mais variados campos com informações pessoais a medida que forem necessários.

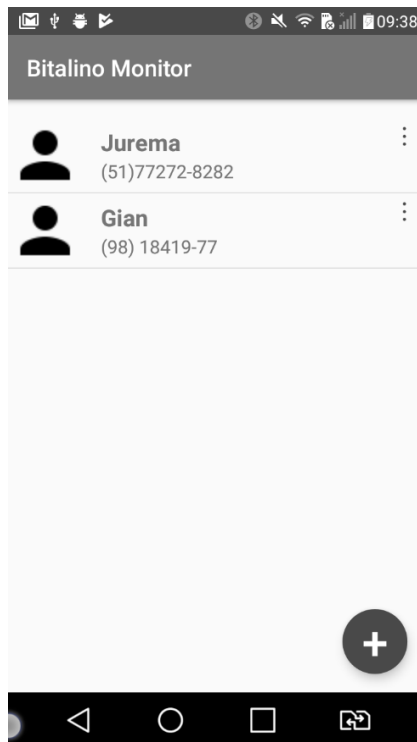
Figura 12 – Tela de Cadastro de Pacientes



Fonte: Elaborado pelo autor

Após o cadastramento do paciente ele é listado na tela de listagem de pacientes como mostra a Figura 13.

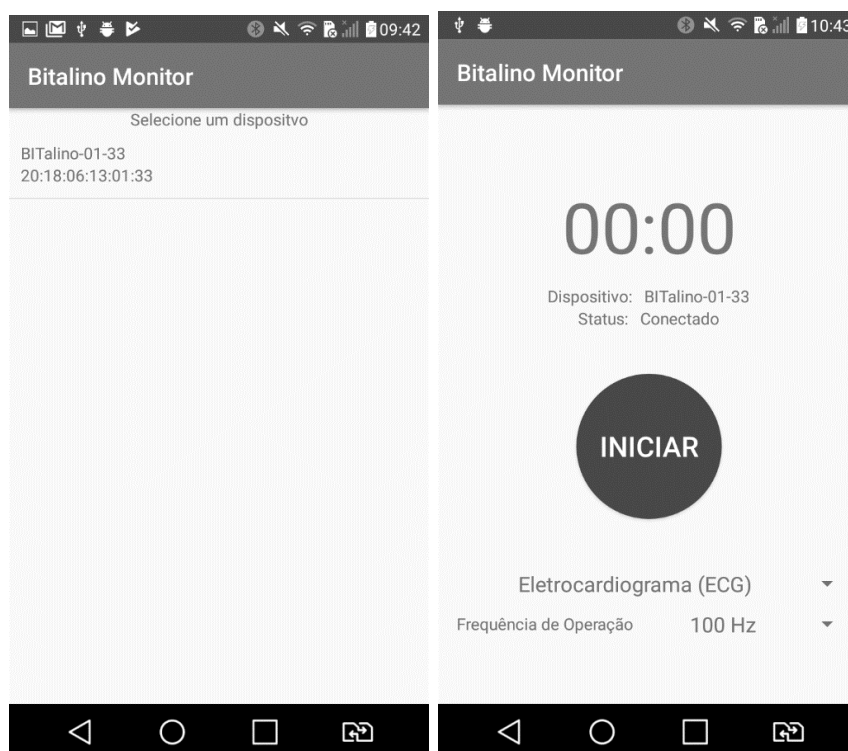
Figura 13 – Tela de Listagem de Pacientes



Fonte: Elaborado pelo autor

Com o paciente cadastrado é permitido a vinculação de um exame ao mesmo através da tela de execução de exame, sendo antes necessário selecionar um dispositivo *BITalino* para que seja feito o emparelhamento via *Bluetooth* com o aplicativo, como exibido na Figura 14.

Figura 14 – Tela de Seleção de Dispositivo e Execução de exame



Fonte: Elaborado pelo autor

Nesta tela é necessário selecionar o tipo de exame a ser executado, pois cada exame corresponde a um canal de comunicação do *BITalino*, como explicado na Tabela 5. Para níveis de teste, é possível também selecionar a frequência de leitura do sinal que obrigatoriamente deve ser 1 Hz, 10 Hz, 100 Hz ou 1000 Hz.

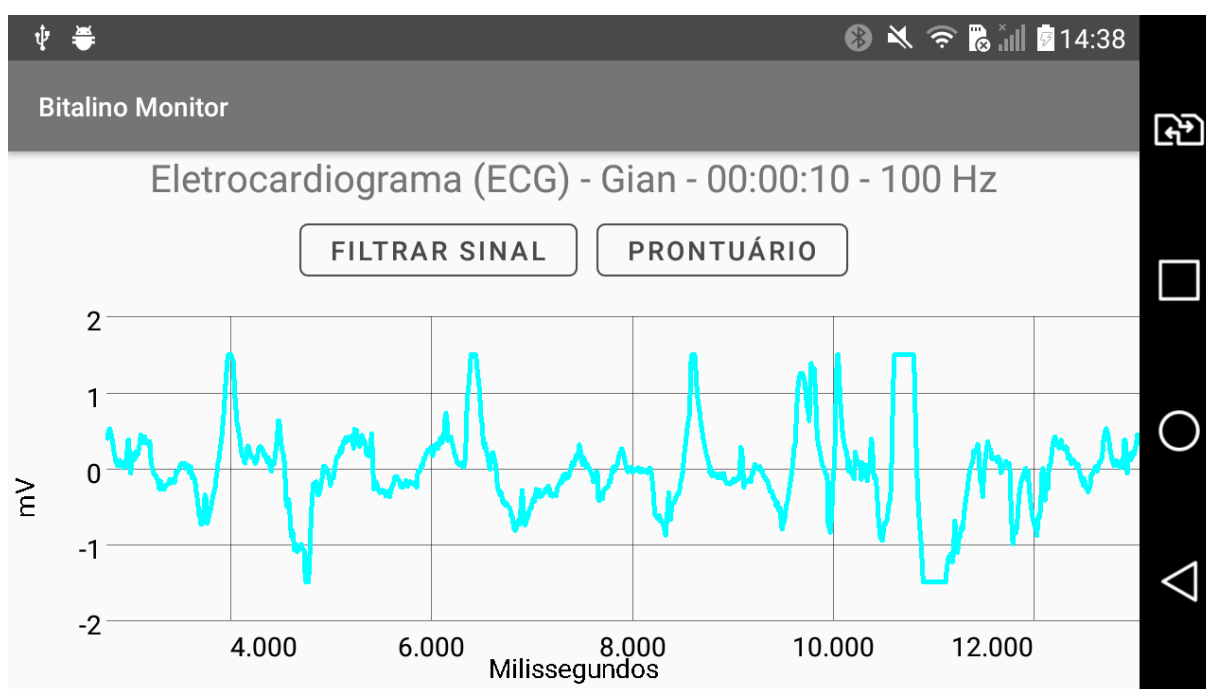
Tabela 5 – Correspondência entre exame e canal de comunicação do *BITalino*

Exame	Canal
Eletromiografia (EMG)	0
Eletrocardiograma (ECG)	1
Resposta Galvânica da Pele (EDA)	2
Eletroencefalografia (EEG)	3

Fonte: Elaborado pelo autor

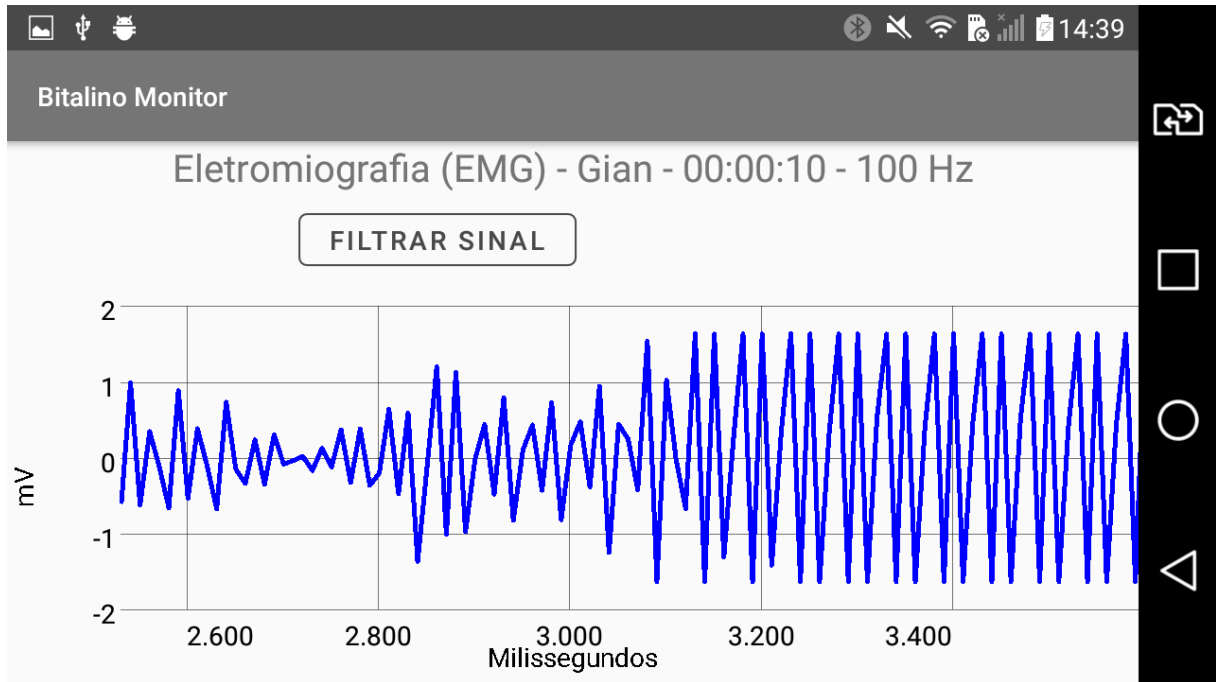
Foi implementado nesta tela um cronometro progressivo para monitoramento do tempo de execução do exame, sendo possível realizar pausas através do botão “Iniciar/Parar” que tem a ação de adquirir os dados de leitura em tempo real. Ao clicar na ação de “Parar” é habilitado um novo botão com o título “Processar Exame”, que salva os dados de forma como adquiridos no banco de dados através da *WebApi* e direciona o usuário para a tela de visualização do exame, em que é exibido um gráfico com o sinal adquirido já processado, o nome do paciente, do exame realizado, a frequência de execução e o tempo de duração, como mostra as Figura 15, 16, 17 e 18.

Figura 15 – Tela de Visualização de Exame Eletrocardiograma



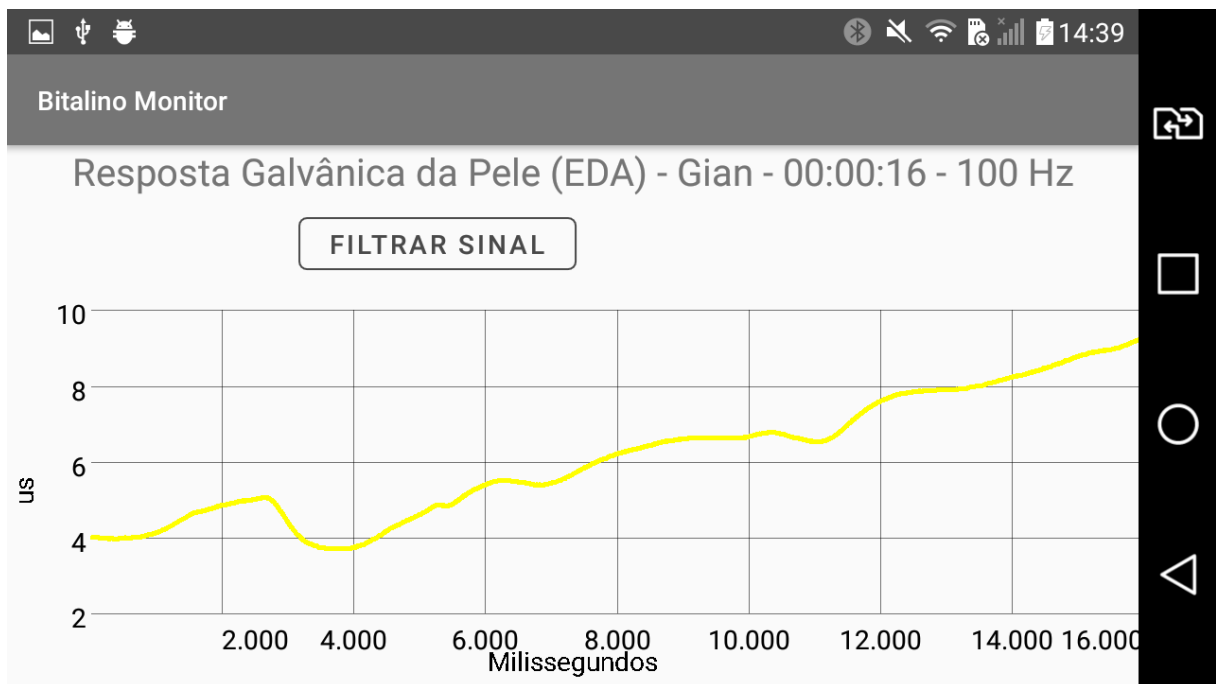
Fonte: Elaborado pelo autor

Figura 16 – Tela de Visualização de Exame Eletromiografia



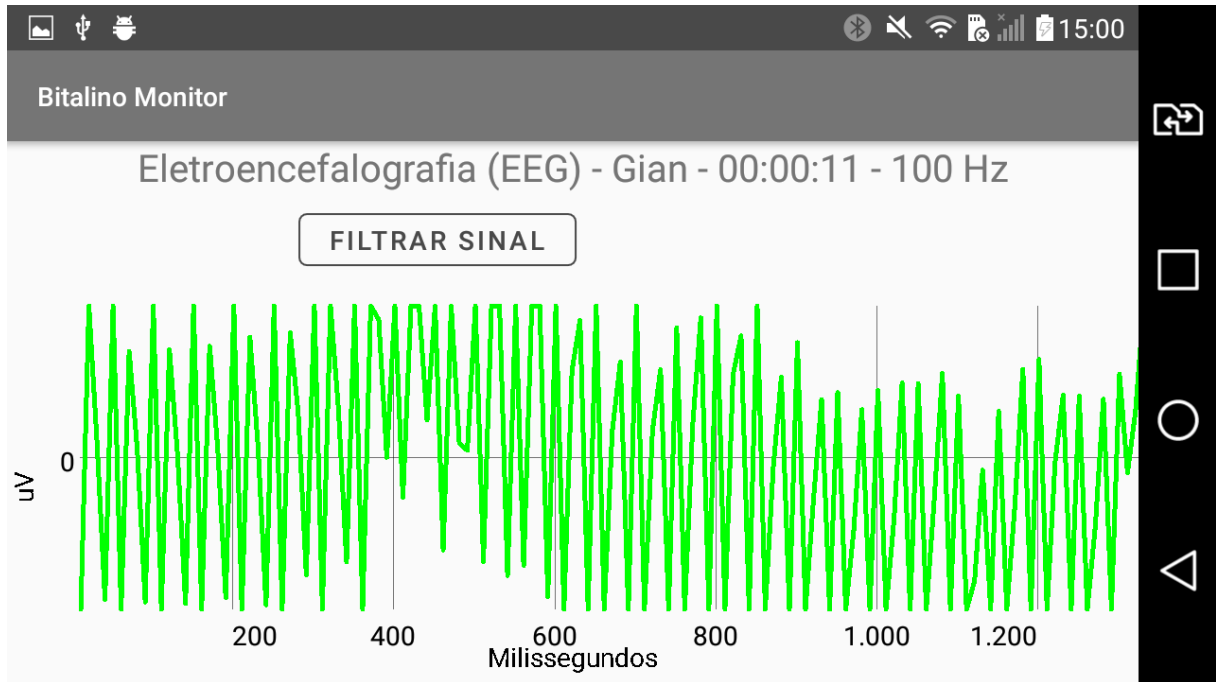
Fonte: Elaborado pelo autor

Figura 17 – Tela de Visualização de Exame Resposta Galvânica da Pele



Fonte: Elaborado pelo autor

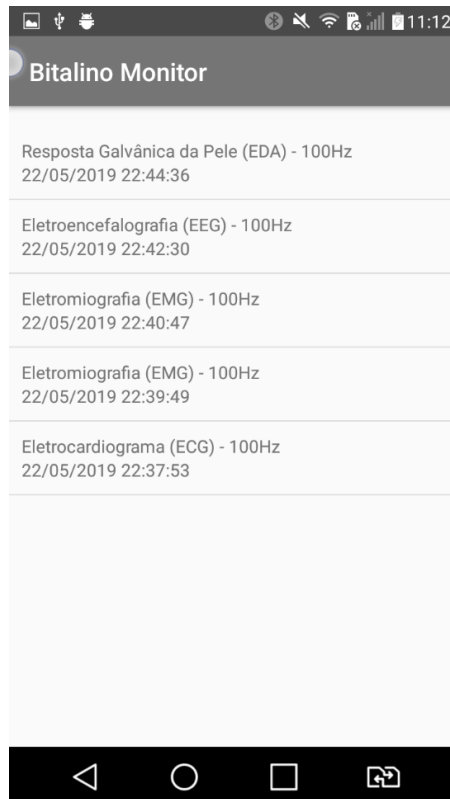
Figura 18 – Tela de Visualização de Exame Eletroencefalografia



Fonte: Elaborado pelo autor

Após este processo é possível visualizar novamente o exame na listagem de exame que cada paciente possui, conforme Figura 19.

Figura 19 – Tela de Exames Realizados

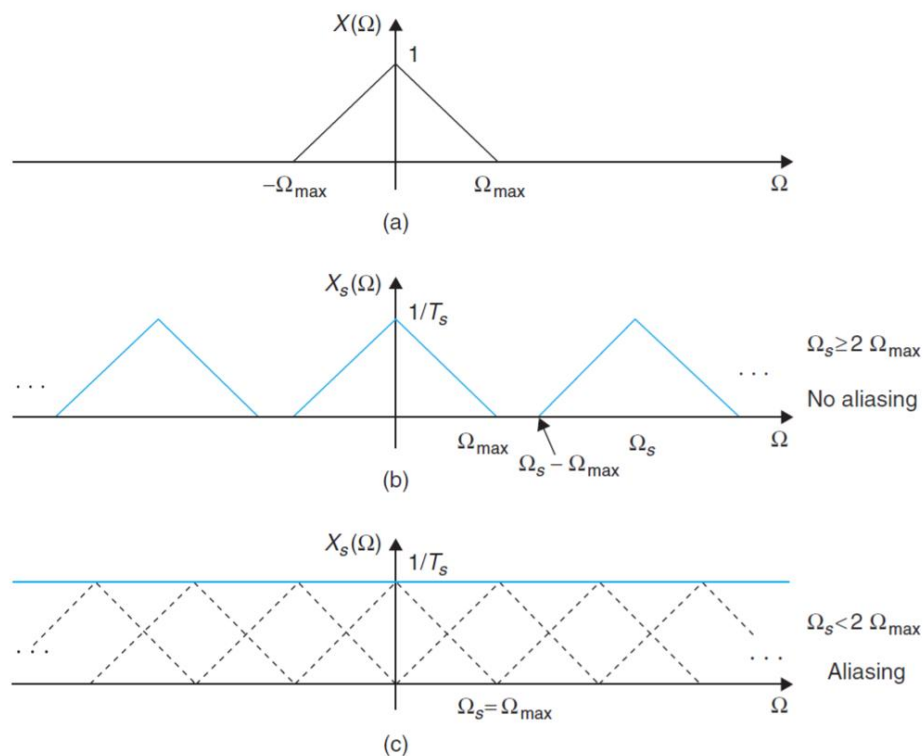


Fonte: Elaborado pelo autor

5.1.3 Filtragem do Sinal

Inicialmente para evitar o comportamento de *aliasing*, que corresponde a uma indesejada sobreposição de espectro como mostrado na Figura 20, foi utilizado o teorema de *Nyquist* que define que a frequência de amostragem deve ser duas vezes a máxima frequência que será amostrada. (CHAPARRO, 2011)

Figura 20 – Exemplo de Aliasing: (a) Espectro do sinal, (b) espectro do sinal com teorema de Nyquist, (c) espectro do sinal com aliasing



Fonte: Adaptado de CHAPARRO, 2011

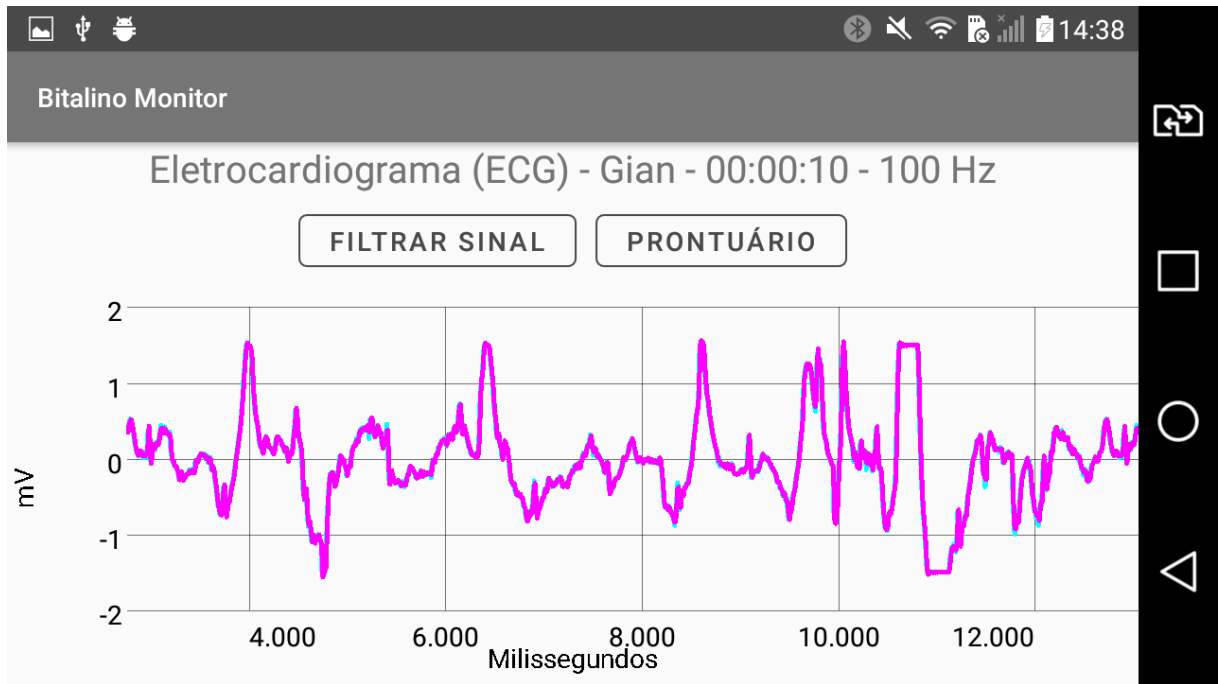
Logo a frequência de corte normalizada é definida pela equação

$$\omega = \frac{f_{\text{corte}}}{f} \times 2 \quad (12)$$

Foi implementado um filtro de ordem 2 conforme realizado também nos trabalhos de Rathke (2011) e Schwarz (2009), considerando o período como $T = \pi$ durante a transformação bilinear. A ordem do filtro determinará sua taxa de ganho ou atenuação em função da frequência, portanto a complexidade do filtro cresce conforme maior for sua ordem.

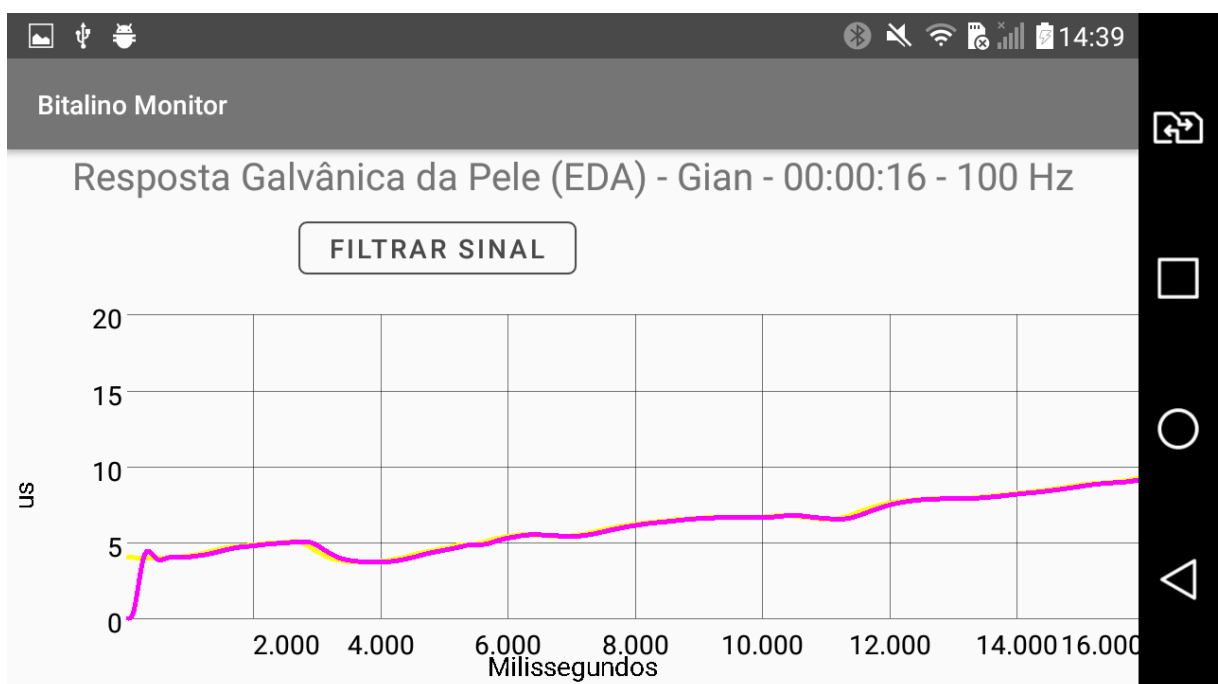
No aplicativo foi adicionado um botão na tela de exibição gráfica do sinal com o nome “Filtrar Sinal” e quando acionado realiza a busca dos dados obtidos previamente com a leitura dos sensores, realiza a filtragem do sinal e exibe um gráfico sobreposto conforme ilustrado nas Figuras 21, 22, 23 e 24.

Figura 21 - Filtragem do sinal de Eletrocardiograma



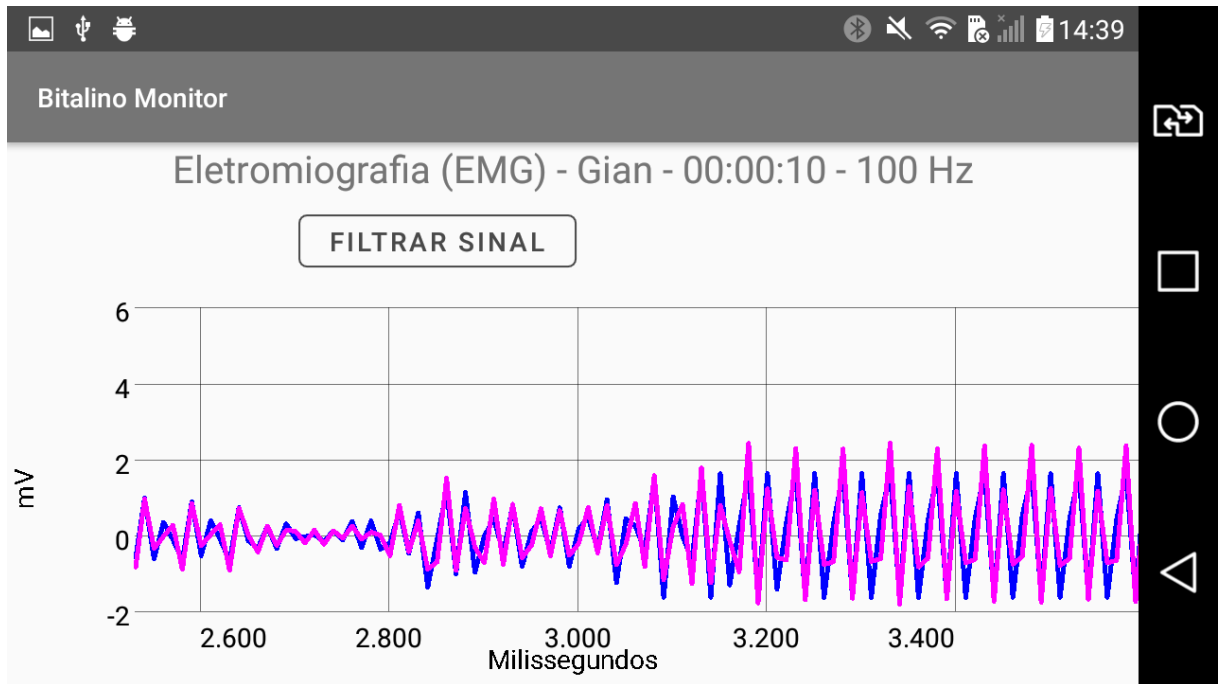
Fonte: Elaborado pelo autor

Figura 22 - Filtragem do sinal de Resposta Galvânica da Pele



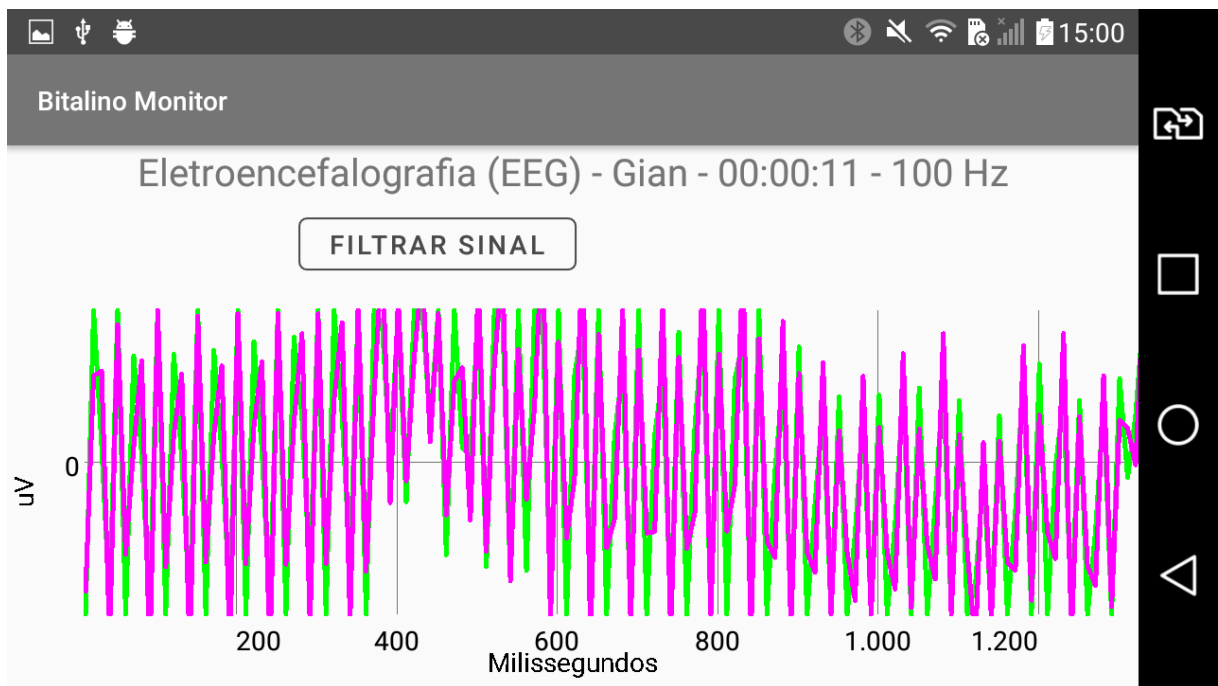
Fonte: Elaborado pelo autor

Figura 23 - Filtragem do sinal de Eletromiografia



Fonte: Elaborado pelo autor

Figura 24 - Filtragem do sinal de Encefalografia



Fonte: Elaborado pelo autor

5.1.4 openEHR

Antes de construir um modelo de prontuário eletrônico que seja estruturado de acordo com os modelos de arquétipos do *openEHR*, é necessário conhecer melhor a estrutura dos arquétipos.

Os arquétipos são modelos eletrônicos computáveis, estruturados e detalhados, de um conceito clínico e para garantir a interoperabilidade cada arquétipo deve ser único e abrangente o suficiente para servir a qualquer situação de uso, sendo possível inclusive traduzi-los em diferentes idiomas. Os arquétipos são desenvolvidos por profissionais da saúde e disponibilizados em repositórios como o *openEHR CKM - Clinical Knowledge Manager*. Nesta plataforma cada arquétipo possui diferentes estados, como rascunho, revisão, publicado, rejeitado, entre outros. (BACELAR; CORREIA, 2015)

Para atender diferentes necessidades da informação, cada arquétipo possui uma classe como pode ser visto na Figura 25.

Figura 25 - Principais classes de arquétipos



Fonte: Adaptado de BACELAR, CORREIA, 2015

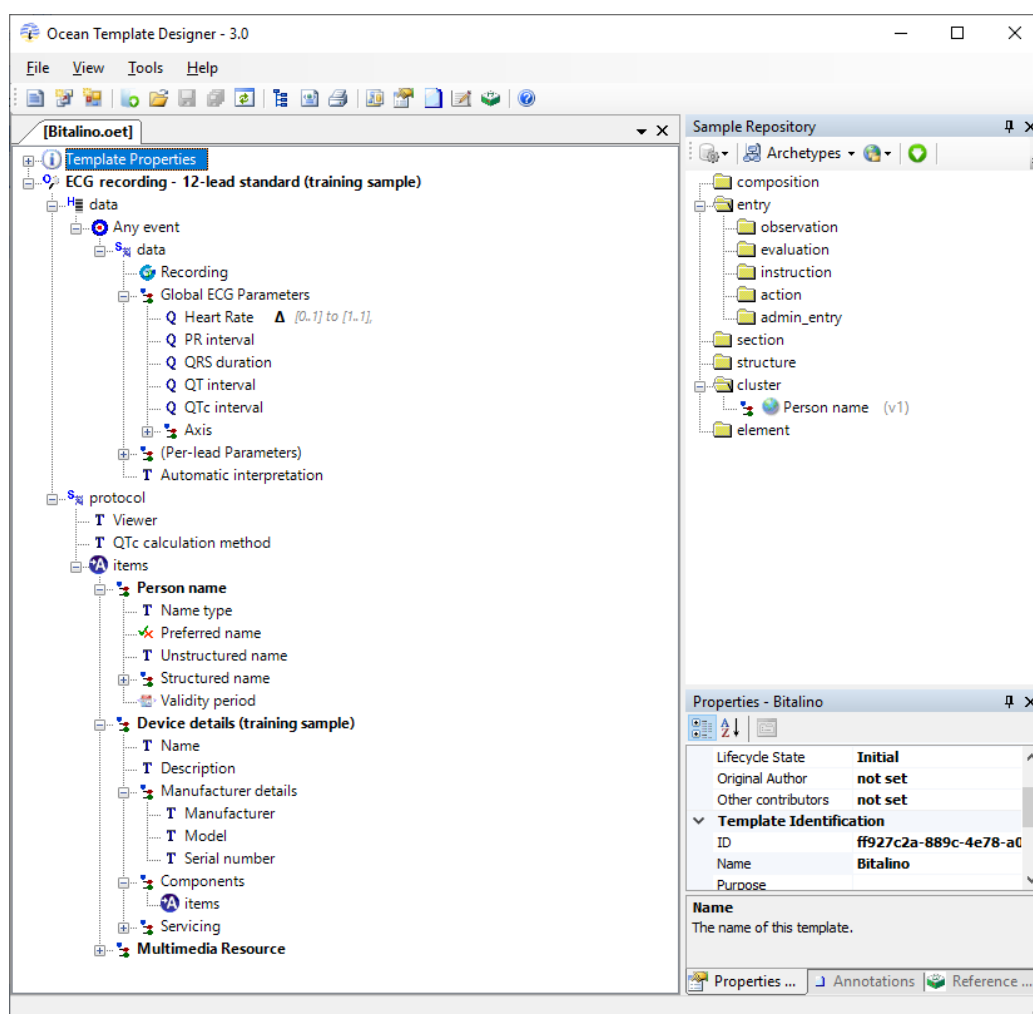
Conforme Bacelar e Correia (2015), a definição de cada classe pode ser esclarecida como:

- Composição (*Composition*): corresponde a um formulário, por exemplo, o formulário de alta.
- Seção (*Section*): é equivalente a um cabeçalho que organiza informações de uma composição.
- Entradas (*Entries*): conceitos clínicos ou administrativos que preenchem uma composição.

- *Clusters*: é um fragmento de arquétipo que pode ser utilizado em diversos contextos de atendimento clínico.
- *Observação (Observation)*: registro de informação de tudo que é dito pelo paciente, resultado de testes, exames ou medidas.
- *Instrução (Instruction)*: informação sobre como dar seguimento aos cuidados de saúde do paciente.
- *Ação (Action)*: são resultados de uma atividade clínica como uma intervenção, por exemplo, um procedimento cirúrgico.
- *Avaliação (Evaluation)*: registra opiniões e diagnósticos dos profissionais da saúde sobre o quadro clínico do paciente.

Para compor os arquétipos julgados necessários no projeto, foi construído um *template* utilizando o software *Ocean Template Designer* da empresa *Ocean Health Systems* como visto na Figura 26.

Figura 26 - Software Ocean Template Designer



Fonte: Elaborado pelo autor

Um *template* é um arquivo que sustenta vários arquétipos de diferentes classes, agrupados em um arquétipo do tipo Composição e pode ser utilizado para dar origem a formulários, relatórios ou mensagens. A Tabela 6 descreve os arquétipos utilizados para construção do *template* cujo nome escolhido foi “*bitalino_monitor.pt.v1*”. A descrição dos templates está disponível no openEHR *CKM*.

Tabela 6 - Arquétipos escolhidos para o template

Arquétipo	Tipo	Descrição	Identificador
ECG result	Observação	Medição da atividade elétrica gerada pelo coração.	openEHR-EHR-OBSERVATION.ecg_result.v0
Result Report	Composição	Documento para comunicar informações a outras pessoas sobre o resultado de um teste ou avaliação.	openEHR-EHR-COMPOSITION.report-result.v1
Medical Device	Cluster	Um instrumento, aparelho, implante, material ou similar, utilizado na prestação de cuidados de saúde	openEHR-EHR-CLUSTER.device.v1
Person Name	Cluster	Detalhes do nome pessoal de um indivíduo, provedor ou terceiro.	openEHR-EHR-CLUSTER.person_name.v1

Fonte: Elaborado pelo autor

Até o presente momento não há arquétipos para uso de exames como Eletroencefalografia, Resposta Galvânica da Pele e Eletromiografia, portanto os dados deste tipo de exame não serão convertidos para o formato de arquétipo.

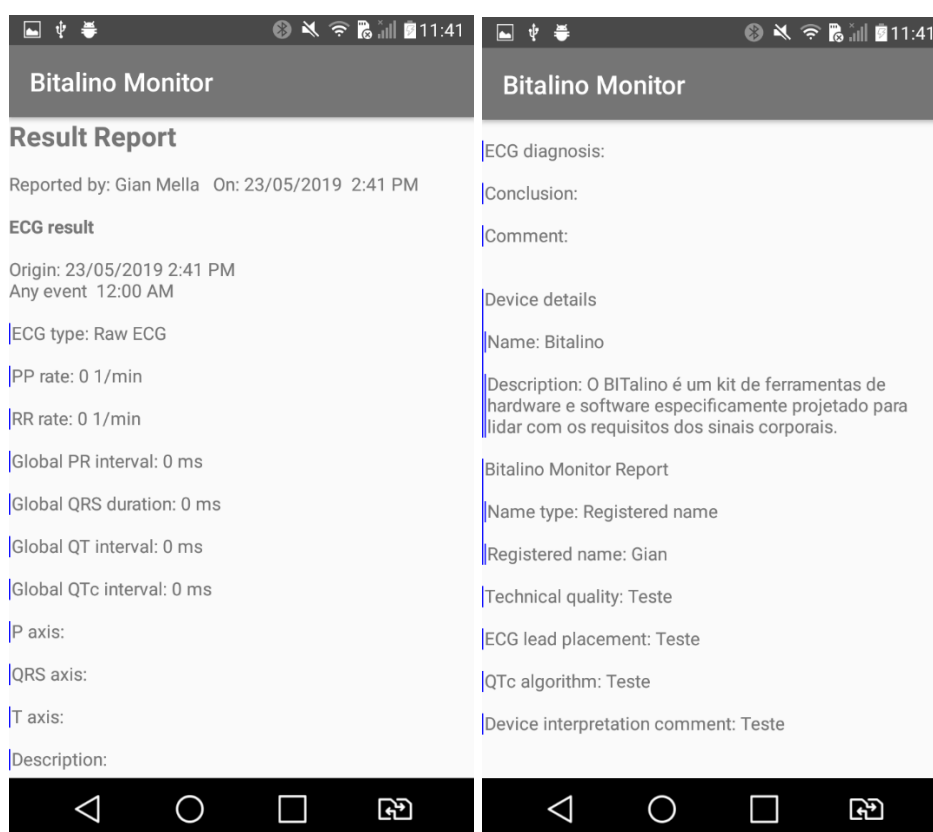
Para preenchimento das informações dos arquétipos foi utilizado o *framework openEHR.NET* que permite através da linguagem *C#* compor objetos *Reference*

Model (RM) e *Archetype Model (AM)* além de serializa-los no formato XML (*Extensible Markup Language - Linguagem de Marcação Extensível*).

Para validar se a estrutura da composição está de acordo com o especificado nos arquétipos e se depender do acesso a um outro sistema que também utilize arquétipos, a empresa *Ocean Informatics* disponibiliza um arquivo no formato XSL (*eXtensible Stylesheet Language - Linguagem de Folha de Estilos Extensível*) que transforma o XML gerado na linguagem HTML (*HyperText Markup Language - Linguagem de Marcação de Hipertexto*). Este arquivo consiste em uma folha de estilos com recomendações para definir a apresentação dos dados e a transformação do XML, garantindo que a estrutura de dados do arquivo XML está de acordo com o esperado pelo arquivo XML e assim garantindo a interoperabilidade desejada.

A Figura 27 exibe o prontuário gerado pela transformação citado anteriormente dentro do aplicativo.

Figura 27 – Tela de exibição de validação do prontuário



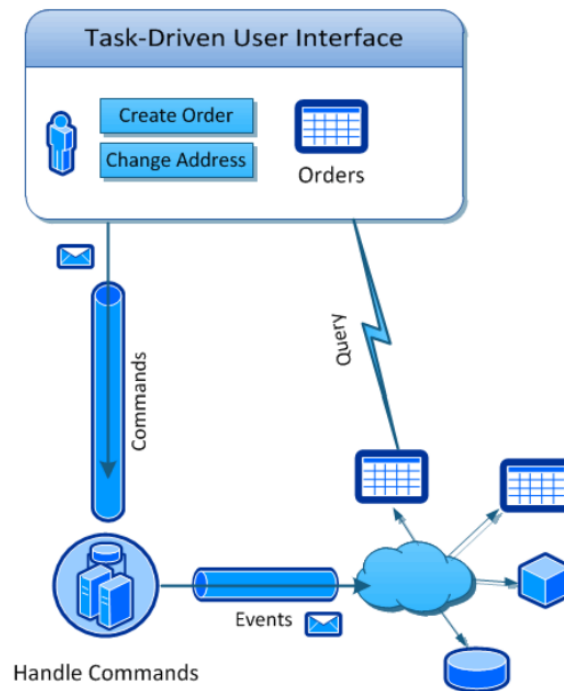
Fonte: Elaborado pelo autor

5.1.5 .NET Core

Como evolução da plataforma *.NET*, o *.NET Core* surgiu totalmente redesenhado e baseado em pacotes pequenos *NuGet*, ou seja, arquivos compactados que contêm códigos compilados com funcionalidades de fácil atualização e localização. A plataforma *.NET* é modular, permitindo o uso de bibliotecas de terceiros e podendo rodar em sistemas operacionais *Windows*, *Mac OS* e *Linux*. O *.NET Core MVC* é um *framework* de desenvolvimento de aplicações *web* criado pela Microsoft que combina o conceito de modelo-visão-controlador (MVC) e na construção da *API* permite a utilização de conceitos como extensão de métodos, expressões lambda, tipos dinâmicos e de expressões em *LINQ* (*Language Integrated Query* – Consulta de Linguagem Integrada). (LIZZONI, 2018)

A *WebApi* desenvolvida neste modelo centraliza a lógica de comunicação com o banco de dados e operações CRUD (*Create Read Update Delete*), assim como as operações importantes como a filtragem e a aplicação da função de transferência do sinal obtido pelo *BITalino*. Foi codificada seguindo o padrão arquitetural *CQRS* (*Command Query Responsibility Segregation* - Segregação de Responsabilidade de Consulta de Comando). Segundo Ferreira (2012), o padrão de desenvolvimento de software *CQRS* é caracterizado pela separação de papéis e responsabilidades dentro do sistema, como comandos, queries e eventos. A Figura 28 apresenta este conceito bem como foi aplicado no desenvolvimento da *WebApi*

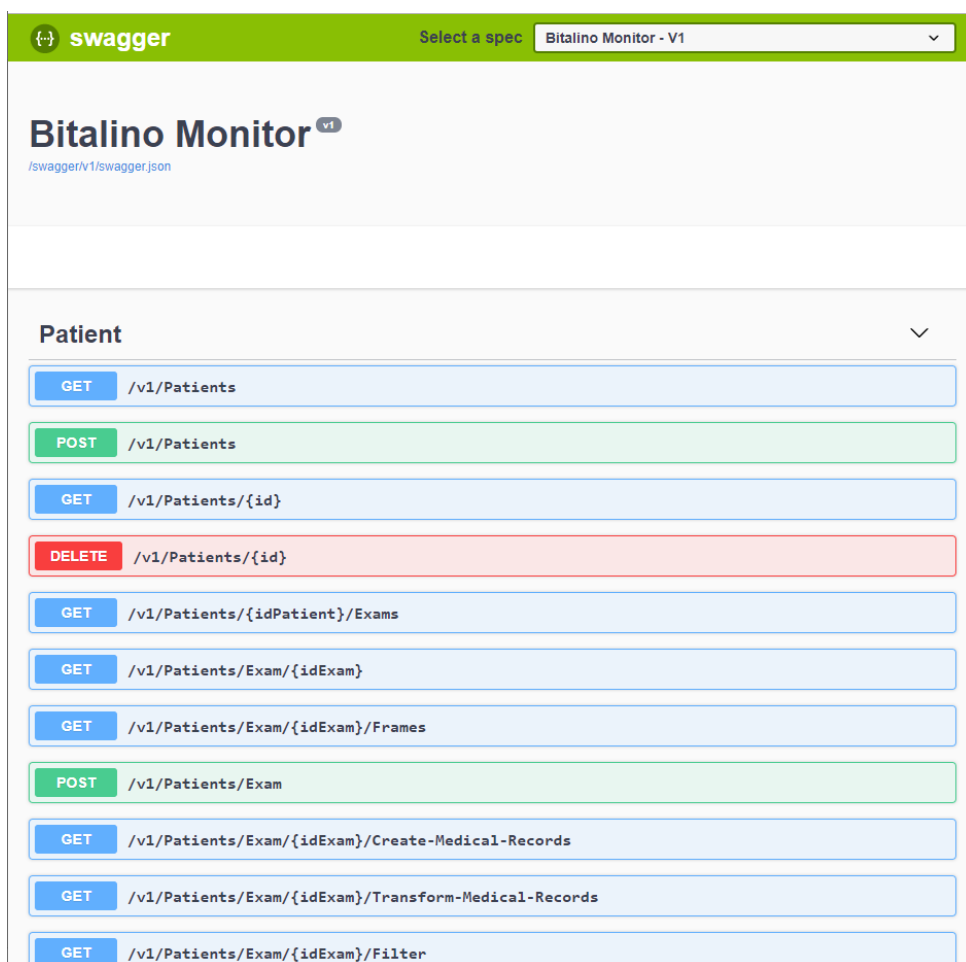
Figura 28 - Componentes lógicos do CQRS



Fonte: Adaptado de FERREIRA, 2012

Como a *WebApi* atende a arquitetura de comunicação REST (*Representational State Transfer* - Transferência de Estado Representacional) foi gerada sua documentação automaticamente pelo *framework open source Swagger* que descreve recursos da *WebApi* como os *endpoints* e os parâmetros de entrada e objetos retorno de cada rota disponível, disponibilizados em uma interface gráfica que pode ser vista na Figura 29.

Figura 29 – Visualização do Swagger



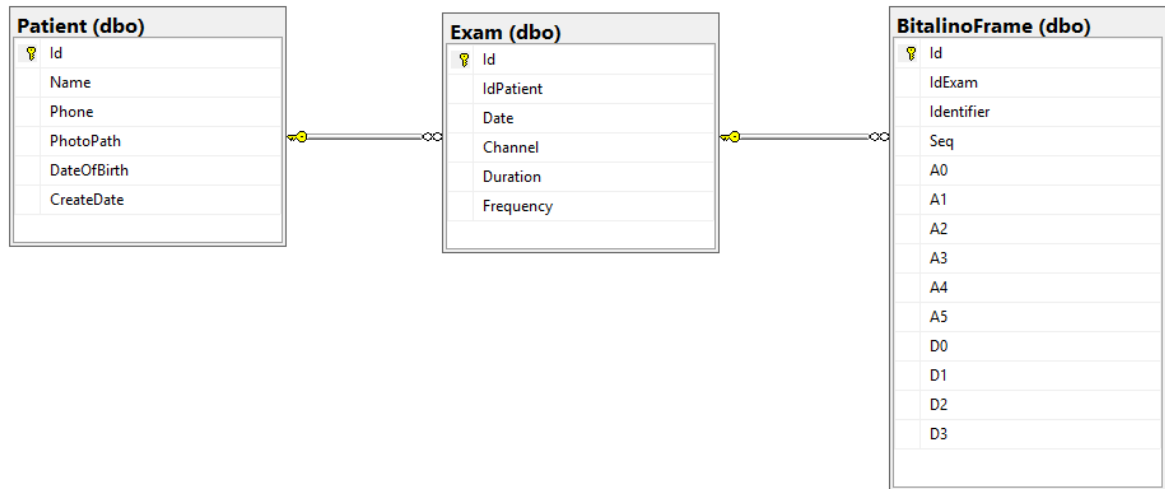
Fonte: Elaborado pelo autor

5.1.6 SQL Server

A base de dados implementada no modelo utiliza o sistema gerenciador de banco de dados relacional *Microsoft SQL Server*. Com o este gerenciador de banco de dados é possível conectar-se a outros bancos de dados, importar e exportar relatórios, criar tarefas, entre outras demandas (OLIVEIRA, 2018)

O diagrama de entidade-relacionamento (DER) implementado do banco de dados é apresentado pela Figura 30.

Figura 30 – Modelo ER do Banco de Dados



Fonte: Elaborado pelo autor

5.1.7 Azure

A infraestrutura que hospeda a camada de processamento do modelo está distribuída na plataforma *Microsoft Azure* que consiste em um conjunto de ferramentas e serviços de nuvem para soluções de *TI*. A Figura 31 mostra o painel de recursos do Azure com a estrutura gerada para atender o modelo proposto.

Figura 31 - Painel de Recursos do Azure

Painel > Todos os recursos					
Todos os recursos					
Assinaturas: Azure for Students Starter					
Filtrar por nome...					
Todos os grupos de recursos					
Todos os tipos					
Todos os locais					
Todas as marcações					
Nenhum agrupamento					
5 itens					
Mostrar os tipos ocultos					
	NOME	TIPO	GRUPO DE RECURSOS	LOCALIZAÇÃO	ASSINATURA
<input type="checkbox"/>	bitalino-monitor	SQL Server	bitalino-monitor	Brazil South	Azure for Students Starter
<input type="checkbox"/>	bitalino-monitor-database (b...	Banco de dados SQL	bitalino-monitor	Brazil South	Azure for Students Starter
<input type="checkbox"/>	bitalino-monitor-web-api	Application Insights	bitalino-monitor	South Central US	Azure for Students Starter
<input type="checkbox"/>	bitalino-monitor-web-api	Serviço de Aplicativo	bitalino-monitor	Brazil South	Azure for Students Starter
<input type="checkbox"/>	plano-serviço-brasil-gratuito	Plano do Serviço de Aplicativo	bitalino-monitor	Brazil South	Azure for Students Starter

Fonte: Elaborado pelo autor

5.1.8 Github

Para contribuir com trabalhos futuros e ampliação do modelo desenvolvido, os códigos fontes da *WebApi* desenvolvida em *C#* e do aplicativo desenvolvido em *Android* estão disponibilizados na plataforma *Github*, que consiste em uma ferramenta de desenvolvimento que permite o compartilhamento e versionamento de código, entre outras funcionalidades que contribuem essencialmente para o desenvolvimento de *software*.

5.2 Análise de Resultados

Foi escolhida a ferramenta *EHRServer*, desenvolvido pela empresa *CaboLabs*, para a validação do prontuário eletrônico gerado através da *WebApi*, já que esta ferramenta é gratuita e permite a construção de um repositório de dados clínicos em um servidor, permitindo também o armazenamento e gerenciamento de documentos clínicos. (CLOUDEHRSERVER, 2019)

Inicialmente foi necessário submeter o *template* criado para atender a necessidade do projeto através da tela de Modelos, conforme Figura 32. Este passo é fundamental já que posteriormente será enviado ao *EHRServer* um arquivo XML contendo o prontuário eletrônico que, no servidor, será convertido e validado conforme estrutura esperada pelo *template*.

Figura 32 - Lista de Modelos do EHRServer

Template ID	Concept	Language	UID	Archetype ID	Version	
Templates accessible with your current language						
bitalino_monitor.en.v1	Bitalino Monitor	pt	a0ed9e0c-ca36-44d5-aec6-eca51b3770fc	openEHR-EHR-COMPOSITION.report-result.v1	1	Template indexes Archetype indexes
Templates not accessible with your current language						
These templates won't be shown on the Query Builder. You should load templates with the same language as your that you will use, change your language from the login screen, or share the templates with organizations that other languages are used.						
vital_signs_summary.en.v1	Vital Signs Summary	en	544ca1ed-d6f0-4446-9250-0dfda06a8db1	openEHR-EHR-COMPOSITION.health_summary.v1	1	Template indexes Archetype indexes

Fonte: Elaborado pelo autor

O próximo passo na validação do arquivo é a criação de um registro eletrônico de saúde através da tela de *EHRs* que é apresentada na Figura 33. O prontuário gerado será vinculado a este registro em seguida.

Figura 33 - Lista de EHRs do EHRServer

UID	Date Created	Subject	Organization
431fb248-b65f-435a-a7ab-db88d26cce58	2019-05-30 16:08:46.0	86167765-6f93-4fa7-a470-e0f9fbf3ea5a	e9d13294-bce7-44e7-9635-8e906da0c914

Fonte: Elaborado pelo autor

Para persistir o arquivo XML ao *EHRServer*, foi seguido a recomendação da *CaboLabs* que sugere utilizar o *software Insomnia* que consiste em uma ferramenta que permite a realização de testes em APIs REST através de mensagens. A empresa *CaboLabs* disponibiliza gratuitamente um *Script* contendo as chamadas já estruturadas para a API do *EHRServer*, sendo somente necessário substituir o conteúdo do arquivo XML no corpo da mensagem desejada.

No Apêndice A é possível verificar um exemplo de XML gerado pela WebApi no que diz respeito a um eletrocardiograma.

Através da rota “/api/v1/ehrs/{%uid%}/compositions” em que “{%uid%}” é substituído pelo UID corresponde ao gerado na criação do EHR, o arquivo XML é persistido e a confirmação do ocorrido pode ser visualizado na interface do *Insomnia* como mostra a Figura 34.

Figura 34 - Confirmação de persistência do prontuário eletrônico pelo Insomnia

201 Created TIME 1.56 s SIZE 1113 B

Preview ▾ Header 4 Cookie Timeline

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <commitResult>
3   <type>AA</type>
4   <message>Versions successfully committed to EHR 431fb248-b65f-435a-a7ab-db88d26cce58</message>
5   <versions>
6     <version>
7       <uid>99fccca0c-fa74-4d33-9f01-d01f9ad1968d::CABOLABS_EHR::1</uid>
8       <precedingVersionUid />
9       <lifecycleState>532</lifecycleState>
10      <commitAudit>
11        <timeCommitted>2019-05-30 20:32:11</timeCommitted>
12        <committer>
13          <namespace>local</namespace>
14          <type>PERSON</type>
15          <value />
16          <name>Dr. Cristiano</name>
17        </committer>
18        <systemId>CABOLABS_EHR</systemId>
19        <changeType>CREATION</changeType>
20      </commitAudit>
21      <data>
22        <uid>e7493792-5125-490d-8ca2-49045b2383c4</uid>
23        <category>event</category>
24        <startTime>2019-05-30 00:00:00</startTime>
25        <subjectId>86167765-6f93-4fa7-a470-e0f9fbf3ea5a</subjectId>
26        <ehrUid>431fb248-b65f-435a-a7ab-db88d26cce58</ehrUid>
27        <templateId>bitalino_monitor.en.v1</templateId>
28        <archetypeId>openEHR-EHR-COMPOSITION.report-result.v1</archetypeId>
29        <lastVersion>true</lastVersion>
30        <organizationUid>e9d13294-bce7-44e7-9635-8e906da0c914</organizationUid>
31        <parent>99fccca0c-fa74-4d33-9f01-d01f9ad1968d::CABOLABS_EHR::1</parent>
32      </data>
33    </version>
34  </versions>
35 </commitResult>

```


Fonte: Elaborado pelo autor

No final deste processo é possível localizar o prontuário no *EHRServer* através da tela de detalhes do EHR como exibido na Figura 35.

Figura 35 - Detalhamento do EHR no EHRServer

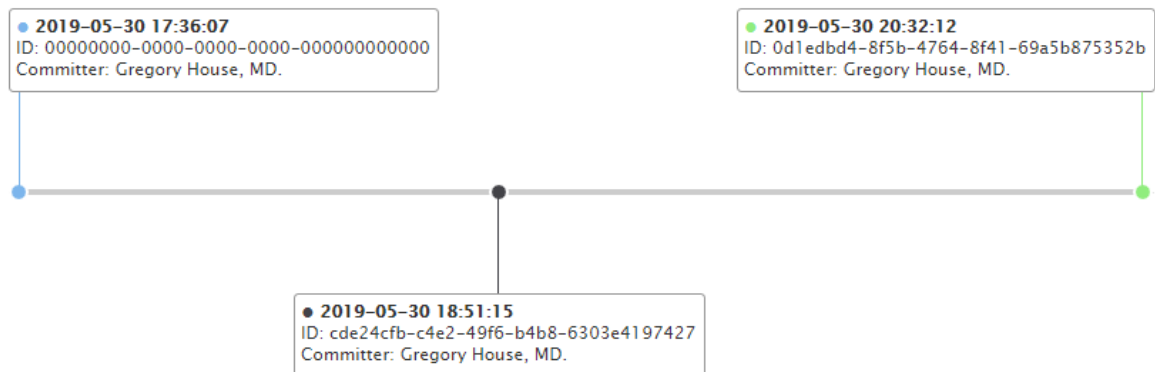
EHR Details

UID	431fb248-b65f-435a-a7ab-db88d26cce58
Subject UID	86167765-6f93-4fa7-a470-e0f9fbf3ea5a
Organization	e9d13294-bce7-44e7-9635-8e906da0c914
Date Created	30/05/2019 16:08:46 UTC
System ID	CABOLABS_EHR_SERVER

 Remove

Contribuições


Contribuições





Fonte: Elaborado pelo autor

Na Figura 36 é apresentado o versionamento do prontuário eletrônico, conforme estrutura criada pelo próprio *EHRServer*.

Figura 36 - Detalhamento do Versionamento do EHR no EHRServer

Time Committed		Committer			
2019-05-30 17:36:07.0		Gregory House, MD.			
UID	Creation Date	Template	Type	Change Type	
23187bc9-541b-4bf6-a5aa-d72af72f4558::CABOLABS_EHR::1	2019-05-30 00:00:00.0	bitalino_monitor.en.v1	openEHR-EHR-COMPOSITION.report-result.v1	CREATION	

Time Committed		Committer			
2019-05-30 18:51:15.0		Gregory House, MD.			
UID	Creation Date	Template	Type	Change Type	
69dc9436-bb42-4af3-ae09-fbcd34d97eef::CABOLABS_EHR::1	2019-05-30 00:00:00.0	bitalino_monitor.en.v1	openEHR-EHR-COMPOSITION.report-result.v1	CREATION	

Time Committed		Committer			
2019-05-30 20:32:12.0		Gregory House, MD.			
UID	Creation Date	Template	Type	Change Type	
99fcca0c-fa74-4d33-9f01-d01f9ad1968d::CABOLABS_EHR::1	2019-05-30 00:00:00.0	bitalino_monitor.en.v1	openEHR-EHR-COMPOSITION.report-result.v1	CREATION	

Fonte: Elaborado pelo autor

Ao clicar no ícone que simboliza uma folha são exibidos os dados do prontuário, similar ao exibido na interface do aplicativo, este processo é ilustrado pela Figura 37.

Figura 37 - Tela de detalhamento do prontuário eletrônico

UID	Creation	Template	Type	Change
Result Report				Reported by: Dr. Cristiano
				On: 30/05/2019 5:30 PM
ECG result				
Origin: 30/05/2019 5:30 PM				
Any event 5:30 PM				
ECG type: Raw ECG				
PP rate: 0 1/min				
RR rate: 0 1/min				
Global PR interval: 0 ms				
Global QRS duration: 0 ms				
Global QT interval: 0 ms				
Global QTc interval: 0 ms				
P axis: Sem Valor				
QRS axis: Sem Valor				
T axis: Sem Valor				
Clinical information provided: Sem Valor				
Device interpretation: Sem Valor				
Finding: Sem Valor				
ECG diagnosis: Sem Valor				
Conclusion: Sem Valor				
Comment: Sem Valor				
Dispositivo				
Name: Bitalino				
Description: O BITalino é um kit de ferramentas de hardware e software especificamente projetado para lidar com os requisitos dos sinais corporais.				
Paciente				
Name type: Registered name				
Registered name: Gian				
Technical quality: Teste				
ECG lead placement: Teste				
QTc algorithm: Teste				
Device interpretation comment: Teste				

Fonte: Elaborado pelo autor

Contudo, este método foi utilizado apenas para comprovar a interoperabilidade do prontuário gerado mesmo que nele, a maioria dos dados inseridos não sejam dinâmicos. O *BITalino* disponibiliza os dados dos exames em função do tempo, sendo microvolts por milissegundos no caso do eletrocardiograma, porém a tradução deste formato de dados para os dados necessários no arquétipo *ECG result* é de grande complexidade no que se refere ao processamento digital de sinais, prevendo cálculos para identificação dos batimentos, intervalo de batimentos, entre outros. Este tipo de processamento claramente ilustra uma evolução deste trabalho bem como a

implementação ideal do método de comunicação com o *EHRServer*, que pode ser implementado em software dentro da *WebApi* construída, pois uma comunicação mais eficiente com o *EHRServer* proverá um forte enriquecimento no modelo proposto, permitindo um armazenamento do histórico do paciente a parte da base de dados da *WebApi*.

6. CONCLUSÃO

Mediante a leitura dos artigos recentes sobre o tema proposto e a análise feita em cima do estado da arte, construiu-se neste trabalho um modelo para leitura de sinais vitais que se acredita que contribuirá com a melhora do processo de monitoramento de pacientes por profissionais da área da saúde utilizando um *kit* de desenvolvimento que futuramente possibilitará outros trabalhos relacionados a este tema.

Para comprovar a funcionalidade e o nível de contribuição do modelo proposto, foi construído um protótipo que envolve a confecção de um aplicativo *Android* e de uma camada de serviço estruturada na nuvem, que contempla uma *WebApi* e a base de dados, além da utilização do *kit* de desenvolvimento do *BITalino* para prover a leitura de sinais vitais.

O modelo proposto neste trabalho serve como uma prova de conceito de um sistema amplo, escalável e complexo, pois nele está presente a compatibilidade com um *kit* de desenvolvimento robusto que se mostrou eficiente na aquisição de sinais vitais e ainda possui outros sensores e saídas analógicas não explorados nesta fase do projeto. Também é presente no modelo a centralização em servidor de todas as regras de negócio e comunicação com banco de dados, possibilitando que diferentes clientes possam consumi-las, seja um portal, intranet ou aplicativos escritos para outros sistemas operacionais. Através do processo de melhoria contínua, a implementação de novas funcionalidades na camada de serviços e no aplicativo promovem um crescimento imensurável a nível de software, abrindo caminhos para trabalhos futuros que possam explorar a manipulação dos sinais digitais para conversão de dados dos arquétipos, a implementação de novos arquétipos disponibilizados pela comunidade médica, a segurança do tráfego de informações entre cliente e servidor, a persistência de dados em cache para processamento futuro prevendo a utilização do modelo em áreas remotas, e até mesmo a calibragem dos sensores.

Até o presente momento os objetivos deste trabalho foram concluídos mostrando que o modelo é capaz de auxiliar profissionais da saúde por se mostrar completamente funcional.

REFERÊNCIAS

ALEXEENKO, Pavel. **MobileVJ: A mobile app for a novel wearable human sensing system**. 2016. 75 p. Dissertação (Mestrado Integrado em Engenharia Informática e Computação) - Faculdade de Engenharia, Universidade do Porto, Porto, 2015. Disponível em: <<https://repositorio-aberto.up.pt/handle/10216/85754>>. Acesso em: 17 set. 2018.

ALVES, Jeam Martins. **Utilização do modelo MVC no desenvolvimento de aplicações E-saúde utilizando o padrão openEHR**. 2017. 80 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Tecnológica Federal do Paraná, Curitiba, 2017. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/9204>>. Acesso em: 26 nov. 2018.

ARMSTRONG, Martin. Smartphone Addiction Tightens Its Global Grip. 2017. Disponível em: <<https://www.statista.com/chart/9539/smartphone-addiction-tightens-its-global-grip/>>. Acesso em: 17 out. 2018.

BACELAR, Gustavo Marísio; CORREIA, Ricardo João Cruz. **As bases do openEHR**. Porto, Portugal: [s. n.], 2015. Disponível em: <https://www.researchgate.net/publication/282869250_As_Bases_do_openEHR>. Acesso em: 1 maio 2019.

BITALINO (r)evolution Board Kit Data Sheet. Disponível em: <http://bitalino.com/datasheets/REVOLUTION_BITalino_Board_Kit_Datasheet.pdf>. Acesso em: 28 out. 2018.

BITALINO **Java Application Programming Interface - Documentation - Android API**. [S. l.]. Disponível em: <<https://bitalino.com/docs/android-api.pdf>>. Acesso em: 6 maio 2019.

BRASIL, Governo do. **Brasileiro está buscando vida mais saudável, indica pesquisa**. 2018. Disponível em: <<http://www.brasil.gov.br/noticias/saude/2018/06/brasileiro-esta-buscando-vida-mais-saudavel-indica-pesquisa>>. Acesso em: 17 out. 2018.

CALDAS NETO, Carlos Castelo Branco. **Desenvolvimento de um leitor de cartões inteligentes para dispositivos móveis com comunicação Bluetooth**. 2011. 159 p. Dissertação (Programa de Pós-Graduação em Mecatrônica) - Escola Politécnica, Universidade Federal da Bahia, Salvador, Bahia, 2011. Disponível em: <<https://repositorio.ufba.br/ri/handle/ri/21395>>. Acesso em: 08 nov. 2018.

CAPARELLI, Thiago Bruno. **Design and development of a multichannel biotelemetry system for ECG, EEG and EMG signal acquisition**. 2007. 58 f. Dissertação (Mestrado em Engenharias) - Universidade Federal de Uberlândia, Uberlândia, 2007. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14636>>. Acesso em: 12 nov. 2018.

CLOUDEHRSERVER Clinical Data Management and Sharing Platform. Disponível em: <<https://cloudehrserver.com/>>. Acesso em: 30 maio. 2019.

CORREIA, João de Sá Balão Calisto. **Indexação de Documentos Clínicos**. 2016. Dissertação (Mestrado Integrado em Engenharia Informática e Computação) - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, Portugal, 2016. Disponível em: <<https://repositorio-aberto.up.pt/handle/10216/85738>>. Acesso em: 30 maio 2019.

DINIZ, P. S. R. **Processamento digital de sinais: projeto e análise de sistemas**. Porto Alegre: Bookman, 2014. Disponível em: <<https://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000004956&lang=pt-br&site=eds-live>>. Acesso em: 22 maio. 2019.

DONATI, Luisa Paraguai. **Computadores vestíveis: convivência de diferentes espacialidades**. In: Revista Conexão - Comunicação e Cultura [online], v.3, n.6, pp. 93-102. ISSN: 2178-2687, 2004. Disponível em: <<http://goo.gl/AElqnM>>. Acesso em: 07 nov. 2018.

FERNANDES, Ricardo Souza Monteiro. **Projeto de filtros digitais transicionais Cauver-Chebyshev inverso**. 2006. Dissertação (Mestre em Engenharia Elétrica) - Universidade Federal de Santa Catarina, Florianópolis, 2006. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/89506>>. Acesso em: 5 maio 2019.

FERREIRA, Carlos Miguel Brites da Silva. **Padrão CQRS para sistemas distribuídos de larga escala**. 2012. 161 f. Dissertação (Mestrado) - Curso de Grau de Mestre em Engenharia Informática, Área de Especialização em Arquiteturas, Sistemas e Redes, Instituto Superior de Engenharia do Porto, Instituto Superior de Engenharia do Porto, Porto, 2012. Disponível em: <<http://recipp.ipp.pt/handle/10400.22/2690>>. Acesso em: 7 maio 2019.

FIGUEIREDO, Natanael de Souza. **Aplicação de Filtros de Savitzky-Golay no Processamento de Sinais de Eletrocardiografia**. 2018. 60 p. Dissertação (Programa de Pós-Graduação em Engenharia Elétrica) - Faculdade de Engenharia, Universidade Federal de Itajubá – UNIFEI, Itajubá, MG, 2018. Disponível em: <<https://repositorio.unifei.edu.br/xmlui/handle/123456789/1562>>. Acesso em: 19 out. 2018.

HOSPITALAR. **Para brasileiro, sistema de saúde ainda é frágil, revela estudo**. 2017. Disponível em: <<https://www.hospitalar.com/pt/editorias/121-saude/1169-para-brasileiro-sistema-de-saude-ainda-e-fragil-revela-estudo>>. Acesso em: 17 out. 2018.

JUCHEM, Gabriel Ott. **Soluções De Baixo Custo Para Implementação De Internet Das Coisas Na Área Da Saúde**. 2015. 74 p. Trabalho de Conclusão (Bacharel em Engenharia da Computação) - Unisinos, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2015. Disponível em: <<http://tcc.unisinos.br/DocuWare/Platform/WebClient/1/Integration?fc=bce33608-1522-4c98-bed0-861e104c7051&did=619&p=V>>. Acesso em: 11 set. 2018.

LATHI, B. P. **Sinais e sistemas lineares**. [s. l.], 2006. Disponível em: <<https://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000011521&lang=pt-br&site=eds-live>>. Acesso em: 22 maio. 2019.

LECHETA, Ricardo R. **Google Android - 5ª Edição**. [S. l.]: Novatec, 2015.

LIZZONI, Luciano; FEIDEN, Aldi; FEIDEN, Armin. PLAFIR: **Aplicativo web para planejamento financeiro rural**. Biblios, Pittsburgh, n. 73, p. 91-104, oct. 2018. Disponível em: <http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1562-47302018000400006&lng=es&nrm=iso>. Acesso em: 6 maio 2019.

MALTA, Juliana; CAMPOLONGO, Gabriel Denser; DE BARROS, Tarley Eloy Pessoa; OLIVEIRA, Reginaldo Perilo de. **Eletromiografia aplicada aos músculos da mastigação**. Acta ortop. bras. [online]. 2006, vol.14, n.2, pp.106-107. ISSN 1413-7852. <<http://dx.doi.org/10.1590/S1413-78522006000200011>>. Acesso em: 05 nov. 2018.

MARCHETTI, Paulo Henrique; DUARTE, Marcos. **Instrumentação em Eletromiografia**. 2006. 29 p. Nota Técnica - Laboratório de Biofísica, Escola de Educação Física e Esporte, Universidade de São Paulo, São Paulo, Brasil, 2006. Disponível em: <<http://demotu.org/pubs/emg.pdf>>. Acesso em: 19 nov. 2018.

MARTINS, B. **Persistência de dados clínicos baseados no openEHR: uma abordagem orientada por recursos limitados**. 2016. 106 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Goiás, Goiânia, 2016. Disponível em: <<https://repositorio.bc.ufg.br/tede/handle/tede/6645>>. Acesso em: 12 nov. 2018.

MARTINS, CARLOS MANUEL MIRANDA. **Xamarin vs. Android**. 2018. Dissertação (Engenharia Informática) - Instituto Superior de Engenharia do Porto, Porto, Portugal, 2018. Disponível em: <<http://recipp.ipp.pt/handle/10400.22/12405>>. Acesso em: 22 abr. 2019.

MARTINS, Rui Miguel Silva. **Desenvolvimento de um sensor de fotopletismografia para monitorização cardíaca para aplicação no pulso**. 2010. 90 p. Dissertação (Mestre em Engenharia Biomédica) - Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, Portugal, 2010. Disponível em: <<https://estudogeral.sib.uc.pt/handle/10316/14086>>. Acesso em: 05 nov. 2018.

MOREIRA, Carlos Noel; DA SILVA, Fernando Silvestre. **Automação da discretização de controladores PID e filtros digitais**. *Revista Ciência e Tecnologia*, [S.l.], v. 17, n. 31, dez. 2014. ISSN 2236-6733. Disponível em: <<http://www.revista.unisal.br/sj/index.php/123/article/view/358>>. Acesso em: 04 maio 2019.

NILSSON, James W.; RIEDEL, Susan A. **Circuitos Elétricos**. 5. ed. [S.l.]: Ltc, 1999. 560 p.

NOVAIS, Jennifer. **Os 9 melhores Aplicativos para Cuidar da Saúde**. 2018. Disponível em: <<https://www.cissamagazine.com.br/blog/melhores-aplicativos-saude>>. Acesso em: 17 out. 2018.

OLIVEIRA, Luiz Augusto Santos de; HORST, Pedro dos Santos. **Sistema de monitoramento de embarcação utilizando bluetooth em smartphone**. 2017. 52 f. Trabalho de Conclusão de Curso (Tecnologia em Mecatrônica Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2017. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/9899>>. Acesso em: 19 nov. 2018.

OLIVEIRA, Moacir de Souza et al. **BANCO DE DADOS NO-SQL X BANCO DE DADOS SQL**. South American Development Society Journal, [S.l.], v. 4, n. 11, p. 298, ago. 2018. ISSN 2446-5763. Disponível em: <<http://www.sadsj.org/index.php/revista/article/view/16>>2. Acesso em: 07 maio 2019.

ONCINS, Maristella Cecco; FREIRE, Regina Maria Ayres de Camargo; MARCHESAN, Irene Queiroz. Mastigação: análise pela eletromiografia e eletrognatografia. Seu uso na clínica fonoaudiológica. **Distúrbios da Comunicação**, São Paulo, v. 18, n. 2, p. 155-165, ago. 2006. Disponível em: <<https://revistas.pucsp.br/index.php/dic/article/view/11781>>. Acesso em: 17 out. 2018.

OPENSIGNALS (r)evolution Software Data Sheet. Disponível em: <http://bitalino.com/datasheets/OpenSignals_Datasheet.pdf>. Acesso em: 28 out. 2018.

OPENSIGNALS. Disponível em: <<https://plux.info/index.php/en/11-team/11-opensignals>>. Acesso em: 27 out. 2018.

POHL, Markus. **325,000 mobile health apps available in 2017 – Android now the leading mHealth platform**. Disponível em: <<https://research2guidance.com/325000-mobile-health-apps-available-in-2017>>. Acesso em: 09 nov. 2018.

RATHKE, Juliano Elesbão. **Sistema de processamento de sinais biomédicos: módulos didáticos de aquisição de ECG, EMG, EOG e conversão analógico-digital de biosinais**. 2008. 162 p. Dissertação de Mestrado (Programa de Pós-Graduação em Engenharia Elétrica) - Centro Tecnológico, Universidade Federal de Santa Catarina, Santa Catarina, Brasil, 2008. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/92056>>. Acesso em: 19 nov. 2018.

REIGONES, Ana Rita; GASPAR, Pedro Dinis; GARCIA, Nuno M.. **Sistema de monitorização de biosinais de gado baseado em tecnologias da informação e comunicação & eletrónica (TICE)**. In: ICEUBI2017 - International Congress on Engineering University da Beira Interior, 2017, Covilhã, Portugal. A Vision of the Future. Covilhã, Portugal: [s.n.], 2017. Disponível em: <https://www.researchgate.net/publication/321808640_Information_and_communication_technologies_electronics_ICTE-based_livestock_biosignals_monitoring_system>. Acesso em: 26 out. 2018.

REIS, Adriana Patrícia Monteiro. **Sistema de monitorização para serviço de urgência em ambiente hospitalar - Estágio na Sensing Future**. 2017. 112 p. Dissertação (Mestre em Instrumentação Biomédica) - Instituto Politécnico de Coimbra, Universidade de Coimbra, Coimbra, Portugal, 2017. Disponível em: <<https://comum.rcaap.pt/handle/10400.26/19000>>. Acesso em: 27 nov. 2018.

RICARDO, João Adilson Gama. **ELETROENCEFALOGRAMA NA NEUROEMERGÊNCIA: Impacto na avaliação de pacientes com alteração do estado de consciência e em pacientes na fase hiperaguda do AVC**. 2012. 176 f. Tese (Doutorado em Ciências) - Faculdade de Ciências Médicas, Universidade Estadual de Campinas, Campinas, 2012. Disponível em: <<http://repositorio.unicamp.br/handle/REPOSIP/313705>>. Acesso em: 12 nov. 2018.

RIOS, Thiago de Souza; BEZERRA, Romildo Martins da Silva. **WHMS4 - Um Sistema Integrado de Monitoramento Remoto de Saúde**. 2014. 20 p. Artigo (Graduação Tecnológica em Análise e Desenvolvimento de Sistemas - IFBA) - Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA - ADS, Salvador, Bahia, 2014. Disponível em: <<http://sedici.unlp.edu.ar/handle/10915/42270>>. Acesso em: 22 set. 2018.

ROCHA, Cristofe Coelho Lopes da. **UbHeart - Um modelo para monitoramento de sinais vitais do coração baseado em ciência da situação e computação ubíqua**. 2016. 85 p. Dissertação (Mestrado em Computação Aplicada) - Unisinos, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2016. Disponível em: <<http://repositorio.jesuita.org.br/handle/UNISINOS/5396>>. Acesso em: 05 set. 2018.

ROCHA, Fernanda Suzart da et al. Uso de APPS para a Promoção dos Cuidados à Saúde. **II Seminário de Tecnologias Aplicadas em Educação e Saúde**, Brasil, p. 1-10, jan. 2017. Disponível em: <<http://revistas.uneb.br/index.php/staes/article/view/3832>>. Acesso em: 09 nov. 2018.

SANTANA, Maurício B.; NASCIMENTO, André A.; SANTOS, Leandro B. Sistema de Aquisição de Sinais de EMG e ECG para Dispositivos Android. **II Seminário de Tecnologias Aplicadas em Educação e Saúde**, Salvador, Bahia, p. 1-1, jan. 2017. Disponível em: <<http://www.revistas.uneb.br/index.php/staes/article/view/3831>>. Acesso em: 14 nov. 2018.

SCHAUMANN, Rolf; VALKENBURG, Mac. E. Van. **Analog Filter Design**. New York: Holt, Rinehart and Winston, 2001.

SCHWARZ, Leandro. **Artigo de Revisão: Eletrocardiograma**. Revista do Departamento Acadêmico de Eletrônica, Santa Catarina, p. 3-19, 2009. Disponível em: <<http://ilhadigital.florianopolis.ifsc.edu.br/index.php/ilhadigital/article/view/4>>. Acesso em: 22 maio 2019

SHIGAKI, Helena Belintani; GONÇALVES, Carlos Alberto; DOS SANTOS, Carolina Pantuza Vilar. NEUROCIÊNCIA DO CONSUMIDOR E NEUROMARKETING: POTENCIAL DE ADOÇÃO TEÓRICA COM A APLICAÇÃO DOS MÉTODOS E TÉCNICAS EM NEUROCIÊNCIA. **Revista Brasileira de Marketing – ReMark**, São Paulo, v. 16, n. 4, p. 439-453, dez. 2017. Disponível em: <<http://www.revistabrasileiramarketing.org/ojs-2.2.4/index.php/remark/article/viewArticle/3427>>. Acesso em: 18 out. 2018.

SILVA, Edenize Pinheiro da. **Monitoramento e aquisição de dados das condições de uma instalação elétrica trifásica baseado no Arduino**. Orientador: Ivaldo Ohana. 2016. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Campus Universitário de Tucuruí, Universidade Federal do Pará, Tucuruí, 2016. Disponível em: <<http://bdm.ufpa.br/jspui/handle/prefix/837>>. Acesso em: 21 maio 2019.

SNAPDRAGON 821 Mobile Platform. 2018. Disponível em: <<https://www.qualcomm.com/products/snapdragon/processors/821>>. Acesso em: 17 out. 2018.

SOARES, Rodrigo Teixeira. **SENSORES DE BIOFEEDBACK EM JOGOS ELETRÔNICOS: UM ESTUDO TEÓRICO E PRÁTICO**. 2006. 56 p. TRABALHO DE GRADUAÇÃO (Curso de Graduação em Engenharia de Controle e Automação) - Faculdade de Tecnologia, Universidade de BRASÍLIA, BRASÍLIA, 2016. Disponível em: <<http://bdm.unb.br/handle/10483/16692>>. Acesso em: 30 out. 2018.

TIBES, Chris Mayara dos Santos; DIAS, Jessica David; ZEM-MASCARENHAS, Silvia Helena. Aplicativos móveis desenvolvidos para a área da saúde no Brasil: revisão integrativa da literatura. **Revista Mineira de Enfermagem**, Belo Horizonte, Minas Gerais, v. 18.2, p. 471-478, jan. 2014. Disponível em: <<http://www.reme.org.br/artigo/detalhes/940>>. Acesso em: 09 nov. 2018.

VENTURI, Simoni. **Transformada Z**. 2016. 126 f. Dissertação (Mestrado Profissional em Matemática) - Universidade Tecnológica Federal do Paraná, Curitiba, 2016.

ZANUZ, Luciano; FILIPPETTO, Alexsandro S.; CRESPO, Sergio. **Service-Oriented Architecture: Teoria e Prática**. 2007. 6 p. Artigo (Programa de Pós-Graduação em Computação Aplicada) - Computação Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo, Brasil, 2007. Disponível em: <https://www.researchgate.net/publication/267254103_Service-Oriented_Architecture_Teoria_e_Pratica>. Acesso em: 12 nov. 2018.

APÊNDICE A – XML DE PRONTUÁRIO ELETRÔNICO

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<versions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.openehr.org/v1">
  <version
    xsi:type="ORIGINAL_VERSION">
    <contribution>
      <id
        xsi:type="GENERIC_ID">
          <value>0d1edbd4-8f5b-4764-8f41-69a5b875352b</value>
          <scheme>EHR</scheme>
        </id>
        <namespace>EHR::COMMON</namespace>
        <type>CONTRIBUTION</type>
      </contribution>
      <commit_audit>
        <system_id>CABOLABS_EHR</system_id>
        <committer
          xsi:type="PARTY_IDENTIFIED">
            <external_ref>
              <id
                xsi:type="HIER_OBJECT_ID">
                  <value>EHR</value>
                </id>
                <namespace>DEMOGRAPHIC</namespace>
                <type>PERSON</type>
              </external_ref>
              <name>Dr. Cristiano</name>
            </committer>
            <time_committed>
              <value>20190530T173057,1688975-0300</value>
            </time_committed>
            <change_type>

```

```
<value>Creation</value>
<defining_code>
  <terminology_id>
    <value>openehr</value>
  </terminology_id>
  <code_string>249</code_string>
</defining_code>
</change_type>
<description>
  <value>Teste</value>
</description>
</commit_audit>
<uid>
  <value>f8b6ac2b-643f-44f4-bb53-075b4382ca33::EHR::1</value>
</uid>
<data
  xsi:type="COMPOSITION"
  archetype_node_id="openEHR-EHR-COMPOSITION.report-result.v1">
  <name>
    <value>Result Report</value>
  </name>
  <archetype_details>
    <archetype_id>
      <value>openEHR-EHR-COMPOSITION.report-result.v1</value>
    </archetype_id>
    <template_id>
      <value>bitalino_monitor.en.v1</value>
    </template_id>
    <rm_version>1.0.1</rm_version>
  </archetype_details>
  <language>
    <terminology_id>
      <value>ISO_639-1</value>
    </terminology_id>
```

```
<code_string>en</code_string>
</language>
<territory>
  <terminology_id>
    <value>ISO_639-1</value>
  </terminology_id>
  <code_string>en</code_string>
</territory>
<category>
  <value>event</value>
  <defining_code>
    <terminology_id>
      <value>openehr</value>
    </terminology_id>
    <code_string>433</code_string>
  </defining_code>
</category>
<composer
  xsi:type="PARTY_IDENTIFIED">
  <name>Dr. Cristiano</name>
</composer>
<context>
  <start_time>
    <value>20190530T173057,0144448-0300</value>
  </start_time>
  <setting>
    <value>Hospital X</value>
    <defining_code>
      <terminology_id>
        <value>openehr</value>
      </terminology_id>
      <code_string>secondary medical care</code_string>
    </defining_code>
  </setting>
```

```
</context>
<content
  xsi:type="OBSERVATION"
  archetype_node_id="openEHR-EHR-OBSERVATION.ecg_result.v0">
  <name>
    <value>ECG result</value>
  </name>
  <archetype_details>
    <archetype_id>
      <value>openEHR-EHR-OBSERVATION.ecg_result.v0</value>
    </archetype_id>
    <template_id>
      <value>bitalino_monitor.en.v1</value>
    </template_id>
    <rm_version>1.0.1</rm_version>
  </archetype_details>
  <language>
    <terminology_id>
      <value>ISO_639-1</value>
    </terminology_id>
    <code_string>en</code_string>
  </language>
  <encoding>
    <terminology_id>
      <value>IANA_character-sets</value>
    </terminology_id>
    <code_string>UTF-8</code_string>
  </encoding>
  <subject
    xsi:type="PARTY_IDENTIFIED">
    <name>Dr. Cristiano</name>
  </subject>
  <protocol
    xsi:type="ITEM_TREE"
```

```
archetype_node_id="at0003">
<name>
  <value>Tree</value>
</name>
<items
  xsi:type="CLUSTER"
  archetype_node_id="openEHR-EHR-CLUSTER.device.v1">
  <name>
    <value>Dispositivo</value>
  </name>
  <archetype_details>
    <archetype_id>
      <value>openEHR-EHR-CLUSTER.device.v1</value>
    </archetype_id>
    <template_id>
      <value>bitalino_monitor.en.v1</value>
    </template_id>
    <rm_version>1.0.1</rm_version>
  </archetype_details>
  <items
    xsi:type="ELEMENT"
    archetype_node_id="at0001">
    <name>
      <value>Name</value>
    </name>
    <value
      xsi:type="DV_TEXT">
      <value>Bitalino</value>
    </value>
  </items>
  <items
    xsi:type="ELEMENT"
    archetype_node_id="at0002">
    <name>
```

```

    <value>Description</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>O BITalino é um kit de ferramentas de hardware e software
especificamente projetado para lidar com os requisitos dos sinais corporais.</value>
    </value>
  </items>
</items>
<items
  xsi:type="CLUSTER"
  archetype_node_id="openEHR-EHR-CLUSTER.person_name.v1">
  <name>
    <value>Paciente</value>
  </name>
  <archetype_details>
    <archetype_id>
      <value>openEHR-EHR-CLUSTER.person_name.v1</value>
    </archetype_id>
    <template_id>
      <value>bitalino_monitor.en.v1</value>
    </template_id>
    <rm_version>1.0.1</rm_version>
  </archetype_details>
  <items
    xsi:type="ELEMENT"
    archetype_node_id="at0006">
    <name>
      <value>Name type</value>
    </name>
    <value
      xsi:type="DV_TEXT">
      <value>Registered name</value>
    </value>
  </items>

```



```
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0006">
  <name>
    <value>Registered name</value>
  </name>
  <value
    xsi:type="DV_CODED_TEXT">
    <value>Gian</value>
    <defining_code>
      <terminology_id>
        <value>local</value>
      </terminology_id>
      <code_string>at0020</code_string>
    </defining_code>
  </value>
</items>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0102">
  <name>
    <value>Technical quality</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Teste</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0097">
  <name>
```

```
<value>ECG lead placement</value>
</name>
<value
  xsi:type="DV_TEXT">
  <value>Teste</value>
</value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0025">
  <name>
    <value>QTc algorithm</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Teste</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0095">
  <name>
    <value>Device interpretation comment</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Teste</value>
  </value>
</items>
</protocol>
<data
  archetype_node_id="at0001">
  <name>
    <value>Event Series</value>
```

```
</name>
<origin>
  <value>20190530T173057,1343564-0300</value>
</origin>
<events
  xsi:type="POINT_EVENT"
  archetype_node_id="at0002">
  <name>
    <value>Any event</value>
  </name>
  <time>
    <value>20190530T173057,1338679-0300</value>
  </time>
  <data
    xsi:type="ITEM_TREE"
    archetype_node_id="at0005">
    <name>
      <value>Tree</value>
    </name>
    <items
      xsi:type="ELEMENT"
      archetype_node_id="at0100">
      <name>
        <value>ECG type</value>
      </name>
      <value
        xsi:type="DV_TEXT">
        <value>Raw ECG</value>
      </value>
    </items>
    <items
      xsi:type="ELEMENT"
      archetype_node_id="at0094">
      <name>
```

```
<value>PP rate</value>
</name>
<value
  xsi:type="DV_QUANTITY">
  <magnitude>0</magnitude>
  <units>1/min</units>
  <precision>0</precision>
</value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0013">
  <name>
    <value>RR rate</value>
  </name>
  <value
    xsi:type="DV_QUANTITY">
    <magnitude>0</magnitude>
    <units>1/min</units>
    <precision>0</precision>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0012">
  <name>
    <value>Global PR interval</value>
  </name>
  <value
    xsi:type="DV_QUANTITY">
    <magnitude>0</magnitude>
    <units>ms</units>
    <precision>0</precision>
  </value>
```

```
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0014">
  <name>
    <value>Global QRS duration</value>
  </name>
  <value
    xsi:type="DV_QUANTITY">
    <magnitude>0</magnitude>
    <units>ms</units>
    <precision>0</precision>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0007">
  <name>
    <value>Global QT interval</value>
  </name>
  <value
    xsi:type="DV_QUANTITY">
    <magnitude>0</magnitude>
    <units>ms</units>
    <precision>0</precision>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0008">
  <name>
    <value>Global QTc interval</value>
  </name>
  <value
```

```
xsi:type="DV_QUANTITY">
  <magnitude>0</magnitude>
  <units>ms</units>
  <precision>0</precision>
</value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0020">
  <name>
    <value>P axis</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Sem Valor</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0021">
  <name>
    <value>QRS axis</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Sem Valor</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0022">
  <name>
    <value>T axis</value>
  </name>
```

```
<value
  xsi:type="DV_TEXT">
  <value>Sem Valor</value>
</value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0096">
  <name>
    <value>Clinical information provided</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Sem Valor</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0009">
  <name>
    <value>Device interpretation</value>
  </name>
  <value
    xsi:type="DV_TEXT">
    <value>Sem Valor</value>
  </value>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0101">
  <name>
    <value>Finding</value>
  </name>
  <value
```

```
    xsi:type="DV_TEXT">
      <value>Sem Valor</value>
    </value>
  </items>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0081">
  <name>
    <value>ECG diagnosis</value>
  </name>
  <value
    xsi:type="DV_TEXT">
      <value>Sem Valor</value>
    </value>
  </items>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0089">
  <name>
    <value>Conclusion</value>
  </name>
  <value
    xsi:type="DV_TEXT">
      <value>Sem Valor</value>
    </value>
  </items>
</items>
<items
  xsi:type="ELEMENT"
  archetype_node_id="at0090">
  <name>
    <value>Comment</value>
  </name>
  <value
    xsi:type="DV_TEXT">
```



```
        <value>Sem Valor</value>
      </value>
    </items>
  </data>
</events>
</data>
</content>
</data>
<lifecycle_state>
  <value>complete</value>
  <defining_code>
    <terminology_id>
      <value>openehr</value>
    </terminology_id>
    <code_string>532</code_string>
  </defining_code>
</lifecycle_state>
</version>
</versions>
```