CENTRO UNIVERSITÁRIO FEI
CÉSAR NOGUEIRA SILVA
JEFFERSON ALMEIDA MATOS
FERNANDA FERREZI LOPES

AQUISIÇÃO DE ELETROCARDIOGRAMA E CLASSIFICAÇÃO DE ARRITMIAS POR *MACHINE LEARNING*

São Bernardo do Campo
2022

CÉSAR NOGUEIRA SILVA
JEFFERSON ALMEIDA MATOS
FERNANDA FERREZI LOPES

AQUISIÇÃO DE ELETROCARDIOGRAMA E CLASSIFICAÇÃO DE ARRITMIAS POR *MACHINE LEARNING*

Trabalho de Conclusão de Curso apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Elétrica.
Orientado por: Profa. Dra. Maria Claudia F. de Castro.

São Bernardo do Campo

2022

Silva, César Nogueira.
Aquisição de Eletrocardiograma e Classificação de Arritmias por "Machine Learning" / César Nogueira Silva, Jefferson Almeida Matos, Fernanda Ferrezi Lopes. São Bernardo do Campo, 2022.
73 f. : il.

Trabalho de Conclusão de Curso - Centro Universitário FEI.
Orientadora: Prof.ª Dra. Maria Claudia Ferrari de Castro.

1. ECG. 2. "Machine Learning". 3. Arritmia. I. Matos, Jefferson Almeida. II. Lopes, Fernanda Ferrezi. III. Castro, Maria Claudia Ferrari de, orient. IV. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

CÉSAR NOGUEIRA SILVA
JEFFERSON ALMEIDA MATOS
FERNANDA FERREZI LOPES

AQUISIÇÃO DE ELETROCARDIOGRAMA E CLASSIFICAÇÃO DE ARRITMIAS POR *MACHINE LEARNING*

Trabalho de Conclusão de Curso apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia Elétrica.
Orientado por: Profa. Dra. Maria Claudia F. de Castro.

Comissão julgadora

Profa. Dra. Maria Claudia Ferrari de Castro
Orientador e presidente

Prof. Dr. Marcelo Antonio Pavanello
Examinador (1)

Prof. Dr. Salvador Pinillos Gimenez
Examinador (2)

São Bernardo do Campo
2022

Aos nossos familiares, amigos e professores, que nos ajudaram durante esta grande trajetória.

“Et ideo homo per hoc in sapientia proficit, quae est propria eius perfectio prout est rationalis. (Assim, o homem progride na sabedoria que é sua perfeição própria enquanto racional)”.

***(*Tomás de Aquino)**

RESUMO

Esse projeto apresenta uma proposta de um dispositivo para monitoração cardíaca pessoal. O dispositivo foi desenvolvido aplicando técnicas de Inteligência Artificial (aprendizado de máquina) para o reconhecimento de arritmias. O dispositivo informará, por meio de um e-mail, a detecção da possível arritmia à um médico ou pessoa credenciada, de modo que possa agilizar ou antecipar, um atendimento de emergência. Segundo a classificação da AAMI (*Association for the Advancement of Medical Instrumentation*) foram selecionadas 4 classes, sendo Normal (N), Supraventricular ectópico (SVE), Ventricular ectópico (VE) e Fusão de normal e ventriculares (F). Utilizando a base de dados *MIT-BIH Arrhythmia Database* e o aplicativo (*app) Classification Learner* do Matlab, para o treinamento, foi possível investigar vários modelos, sendo que os melhores foram o Ensemble (SubspaceKNN) e o SVM (Cubic SVM) com acurácias de 94% e 94,1%, respectivamente. Esses modelos foram selecionados para a etapa de teste, tendo o modelo Ensemble (SubspaceKNN) obtido a melhor acurácia (74,4%) e posteriormente sendo utilizado para a implementação no aplicativo de interface do usuário. Desta forma, sua implantação contribui para o desenvolvimento de inovação científico-tecnológica, à medida que o equipamento apresenta características que ainda não foram totalmente implementadas no mercado e causa impacto no âmbito social, tornando-o acessível, sem a necessidade de um serviço de monitoração contínuo contratado pelo paciente.

Palavras-chave: ECG. *Machine Learning*. Arritmia. MATLAB.

ABSTRACT

This project presents a proposal for a device for personal cardiac monitoring. The device was developed by applying Artificial Intelligence (machine learning) techniques for the recognition of arrhythmias. The device will inform, by means of an e-mail, the detection of the possible arrhythmia to a doctor or accredited person, so that he can speed up or anticipate an emergency care. According to the AAMI (Association for the Advancement of Medical Instrumentation) classification, 4 classes were selected, being Normal (N), Supraventricular ectopic (SVE), Ventricular ectopic (LV) and Fusion of normal and ventricular (F). Using the MIT-BIH Arrhythmia Database and the application (app) Classification Learner from Matlab, for training, it was possible to investigate several models, the best of which were the Ensemble (SubspaceKNN) and the SVM (Cubic SVM) with accuracies of 94% and 94.1%, respectively. These models were selected for the test stage, and the Ensemble model (SubspaceKNN) obtained the best accuracy (74.4%) and was later used for implementation in the user interface application. In this way, its implementation contributes to the development of scientific-technological innovation, as the equipment presents characteristics that have not yet been fully implemented in the market and causes an impact on the social sphere, making it accessible, without the need for a monitoring service. continuous contracted by the patient.

Keywords: ECG. Machine learning. Arrhythmia. MATLAB

LISTA DE ILUSTRAÇÕES

Figura 1: Átrio direito (Vista lateral direita) 17
Figura 2: Ventrículo direito (Vista anterior)..... 17
Figura 3: Estrutura Elétrica do coração 19
Figura 4: Elementos do eletrocardiograma (normal)..... 20
Figura 5: Etapas técnicas computadorizadas. 22
Figura 6: Diagrama de blocos do sistema..... 27
Figura 7: Diagrama de blocos funcional do AD8232 28
Figura 8: Diagrama esquemático do amplificador de instrumentação do AD8232 29
Figura 9: Filtro passa-alta implementado no AD8232 29
Figura 10: Esquemático para o filtro passa-baixas. 30
Figura 11: Módulo AD8232 31
Figura 12: Diagrama de circuito do módulo AD8232 31
Figura 13: Arduino UNO..... 32
Figura 14: Diagrama de circuito Arduino UNO 32
Figura 15: Circuito completo do sistema..... 33
Figura 16: Fluxograma da metodologia de processamento de dados 34
Figura 17: Gravações utilizadas para dados de teste e treinamento 36
Figura 18: ECG da gravação 100 do banco de dados. 37
Figura 19: Resultado do treinamento dos algoritmos utilizando o *Classifcation Learner* 38
Figura 20: Matriz de confusão de treino Ensemble (Subspace KNN) 39
Figura 21: Matriz de confusão de treino SVM (Cubic SVM) 39
Figura 22: Matriz de confusão de teste Ensemble (Subspace KNN) 40
Figura 23: Matriz de confusão de teste SVM (Cubic SVM) 40
Figura 24: Modo *design*..... 41
Figura 25: Modo de visão de código 42
Figura 26: Design final do aplicativo 43
Figura 27: Fluxograma da lógica do aplicativo 43
Figura 28: Executável do aplicativo final..... 44
Figura 29: Fluxograma da aquisição do ECG 45
Figura 30: Configurações das senhas de aplicativo do Gmail..... 46
Figura 31: Alocação dos eletrodos no paciente na derivação MLII 47

Figura 32: Indicação para estado de repouso do paciente 48
Figura 33: Dispositivo integrado 49
Figura 34: Interface de usuário após realizar o autoexame 50
Figura 35: Exemplo de e-mail recebido pelo profissional da saúde 51

LISTA DE TABELAS

Tabela 1: Rótulos do banco de dados e sua descrição 35
Tabela 2: Quantidade de amostras por rótulo após filtragem dos dados 37
Tabela 3: Quantidade de amostras por rótulo após equalização 38

SUMÁRIO

1 INTRODUÇÃO 12

2 OBJETIVOS 14

3 FUNDAMENTAÇÃO TEÓRICA 15

3.1 ANATOMIA DO CORAÇÃO 15

3.1.1 Sistema Elétrico do coração 18

3.2 MONITORAMENTO DE SINAIS CARDÍACOS 20

3.2.1 Características do Eletrocardiograma 20

3.3 ARRITMIA CARDÍACA 21

3.4 INTELIGÊNCIA ARTIFICIAL E *MACHINE LEARNING* 22

3.4.1 Algoritmos de aprendizado de máquina 23

4 REVISÃO BIBLIOGRÁFICA 25

4.1 CLASSIFICAÇÃO DE ARRITMIA POR *MACHINE LEARNING* 25

5 METODOLOGIA 27

5.1 VISÃO GERAL DO SISTEMA 27

5.2 PROJETO DE *HARDWARE* 27

5.2.1 Diagrama de blocos 27

5.2.2 Descrição funcional e configuração dos principais componentes 28

5.2.3 Diagrama de interligação dos principais blocos do *hardware* 33

5.3 PROJETO DE *SOFTWARE* 33

5.3.1 Inteligência computacional para classificação de sinais de ECG 34

5.3.2 Algoritmo de aquisição de ECG 38

5.3.3 Interface de usuário 40

5.4 ESTUDOS DE TÉCNICAS PARA AQUISIÇÃO DE ECG 44

5.5 MODO DE USO 46

6 RESULTADOS 48

7 CONCLUSÕES 52

8 PERSPECTIVAS FUTURAS E SUGESTÕES DE MELHORIAS 53

REFERÊNCIAS 54

APÊNDICE A – CÓDIGO DESENVOLVIDO PARA PREPARAÇÃO DO BANCO DE DADOS 58

APÊNDICE B – CÓDIGO PARA CÁLCULO DA ACURÁCIA DE TESTE APÓS EXPORTAÇÃO DOS MODELOS 65
APÊNDICE C – CÓDIGO PARA INTERFACE DE USUÁRIO 66

1 INTRODUÇÃO

Segundo a Organização Pan-Americana da Saúde (OPAS), com base nos dados da Organização Mundial da Saúde (OMS), as doenças cardiovasculares são a principal causa de morte no mundo. Estima-se que 17,9 milhões de pessoas morreram por doenças cardiovasculares em 2016, representando 31% de todas as mortes no mundo. Destes óbitos, estima-se que 85% ocorreram devido a ataques cardíacos e acidentes vasculares encefálicos, sendo mais de três quartos em países de baixa e média renda (OPAS, [s.d.]).

Segundo a Sociedade Brasileira de Arritmias Cardíacas (SOBRAC), mais de 20 milhões de pessoas sofrem de arritmia cardíaca. Arritmia Cardíaca é uma condição caracterizada por alterações no ritmo do coração, em que os batimentos podem se tornar mais lentos, mais rápidos, irregulares ou desordenados. Esse quadro compromete o bombeamento de sangue para o corpo e, em casos extremos, pode levar à morte súbita. No Brasil, um em cada dez brasileiros têm algum tipo de arritmia, sendo responsável por mais de 320 mil mortes súbitas todos os anos (SOBRAC, 2019).

A última década tem mostrado uma mudança na abordagem do cuidado com a saúde, com a integração de recursos de monitoramento de sinais vitais a telefones celulares, relógios e outros dispositivos portáteis. Em particular, a pandemia da COVID-19 causou um aumento na demanda por dispositivos desse tipo (JABULAS, 2021). Contudo, a maioria dos dispositivos de mercado apenas informa o valor dos parâmetros, sem oferecer uma análise dos mesmos. O monitoramento confiável e preciso, bem como a atuação em tempo real quando necessária, requisitos obrigatórios em algumas situações, está presente nos dispositivos de inteligência artificial. Contudo, o custo dessa tecnologia ainda é proibitivo para a grande maioria. Em 2013, a empresa Nexcor criou o primeiro monitor cardíaco portátil inteligente (DARAYA, 2013). O aparelho é capaz de detectar problemas no coração do usuário e transmitir os dados para uma central de atendimento ou hospital através da rede de celular. A proposta era minimizar ao máximo o impacto das doenças cardíacas previamente, além de diminuir os custos hospitalares. Em casos de arritmias, isquemias, infartos ou qualquer alteração no batimento cardíaco, o equipamento emitiria um alerta a central de processamento, a qual avisaria ao médico, empresa de resgate ou parentes sobre o ocorrido, de modo a oferecer o devido atendimento a tempo. A locação do equipamento poderia custar, a época, até R$600,00 por semana (DARAYA, 2013). Atualmente, um equipamento e serviço similar é oferecido pela CardioWeb (SMART LOOPER, 2021). O SmartLooper coleta e envia os traçados de eletrocardiograma (ECG) de

forma contínua para um servidor na nuvem, utilizando telefonia móvel embarcada, ou até mesmo de forma opcional via celular do paciente, mantém o paciente constantemente monitorado e permite o diagnóstico precoce dos eventos a qualquer momento e em qualquer lugar. Mas ainda hoje este é tido como um serviço nível *premium*, acessível para poucos (SMART LOOPER, 2021).

Dessa forma, esse projeto apresenta uma proposta de um dispositivo para monitoração cardíaca pessoal, aplicando técnicas de Inteligência Artificial (aprendizado de máquina), para o reconhecimento de arritmias. Será enviado automaticamente por *e-mail*, seu nome e o tipo de arritmia detectada, para um hospital, SAMU (Serviço de Atendimento Móvel de Urgência) ou pessoa credenciada ao requisitar um pedido de socorro médico, agilizando o atendimento de emergência.

2 OBJETIVOS

O objetivo geral deste trabalho é o desenvolvimento de um dispositivo para monitoração cardíaca pessoal e detecção de arritmias. Como objetivos específicos, apresenta-se a necessidade da investigação de técnicas de aprendizado de máquina (*machine learning*) capazes de classificar as arritmias com precisão adequada e do desenvolvimento de um equipamento característico, que atenda uma relação custo-benefício em comparação aos existentes no mercado. Desta forma, sua implantação irá contribuir para o desenvolvimento de inovação científico-tecnológica, à medida que o equipamento apresenta características que ainda não foram totalmente implementadas no mercado e causa impacto no âmbito social, tornando-o acessível, sem a necessidade de um serviço de monitoração contínuo contratado pelo paciente.

3 FUNDAMENTAÇÃO TEÓRICA

Para entendimento das características que serão tratadas e analisadas, apresenta-se previamente um estudo conceitual acerca destas.

Este capítulo aborda os principais conceitos relacionados ao desenvolvimento do projeto: anatomia do coração e monitoramento cardíaco pelo eletrocardiograma, as características das principais arritmias cardíacas, inteligência artificial e *machine learning*.

3.1 ANATOMIA DO CORAÇÃO

Dentre diversos órgãos do corpo humano, o coração é aquele que possui a função de bombear o sangue, gerando um fluxo pulsátil que circula para os pulmões e para o corpo, permitindo a troca de substâncias nos tecidos e a manutenção da homeostasia corporal. Localizado no mediastino, um espaço entre os pulmões, tem aproximadamente o tamanho de uma mão fechada, com cerca de 12cm de comprimento, 8cm de largura e 6cm de espessura, e pesando na faixa de 280 a 340g nos homens e de 230 a 280g nas mulheres (DANTAS, 2011).

De modo geral, o coração pode ser descrito como um sistema que recebe o sangue venoso pelas veias cavas inferior e superior. Em seguida, o sangue passa para o átrio direito e depois para o ventrículo direito, que imediatamente bombeia o sangue para o tronco pulmonar, ramifica para as artérias pulmonares (direita e esquerda) e finalmente aos vasos capilares que irrigam o pulmão. A oxigenação do sangue é feita pelos alvéolos, logo após retorna para o átrio esquerdo e passa para o ventrículo esquerdo, que bombeia o sangue para a aorta ascendente, através da válvula aórtica, e então para todo o organismo (DUTRA et al., 2009).

Denomina-se ciclo cardíaco os eventos que ocorrem no coração, desde o início de um batimento até o próximo, ocorrendo em duas etapas: sístole e diástole (contração e relaxamento do coração), que podem, por sua vez, ser divididas em cinco partes: sístole atrial, contração ventricular isovolumétrica, ejeção ventricular, relaxamento isovolumétrico e enchimento ventricular (CARMO, 2021).

No processo denominado sístole atrial, as válvulas atrioventriculares (AV) se abrem e permitem a passagem do sangue dos átrios para os ventrículos, ao mesmo tempo que as válvulas semilunares mantêm-se fechadas à fim de evitar a ejeção prematura para o interior dos grandes vasos. Neste momento, os átrios se contraem juntamente com a abertura da válvula mitral e da válvula tricúspide, fazendo com que o sangue flua para as câmaras cardíacas inferiores (CARMO, 2021).

Na segunda fase do ciclo denominada contração isovolumétrica, as válvulas semilunares e atrioventriculares permanecem fechadas. Neste momento, os ventrículos iniciam uma contração, as válvulas AV são fechadas e a pressão intraventricular aumenta rapidamente. O volume ventricular neste momento permanece constante (CARMO, 2021).

Durante a ejeção ventricular, as válvulas semilunares se abrem e as válvulas atrioventriculares permanecem fechadas. Este fato ocorre devido à pressão dos ventrículos exceder a pressão nos troncos arteriais, fazendo com que as válvulas sejam abertas forçadamente. A pressão do ventrículo esquerdo supera a pressão aórtica, e a pressão no ventrículo direito supera a pressão da artéria pulmonar (CARMO, 2021).

Conforme a sístole progride, a ejeção do volume sanguíneo diminui, e quando o volume sanguíneo chega de 70 a 90ml, a sístole termina. (CARMO, 2021)

No relaxamento isovolumétrico, todas as válvulas cardíacas se fecham. Desta forma, não há sangue passando entre as câmaras cardíacas ou para fora do coração. A pressão interna cai para abaixo de 120mmHg, fazendo com que as válvulas semilunares se fechem. Conforme a pressão decai, as pressões atriais superam as ventriculares, fazendo com que as válvulas atrioventriculares se abram. Durante este evento, o sangue está sendo bombeado ao redor do corpo (CARMO, 2021).

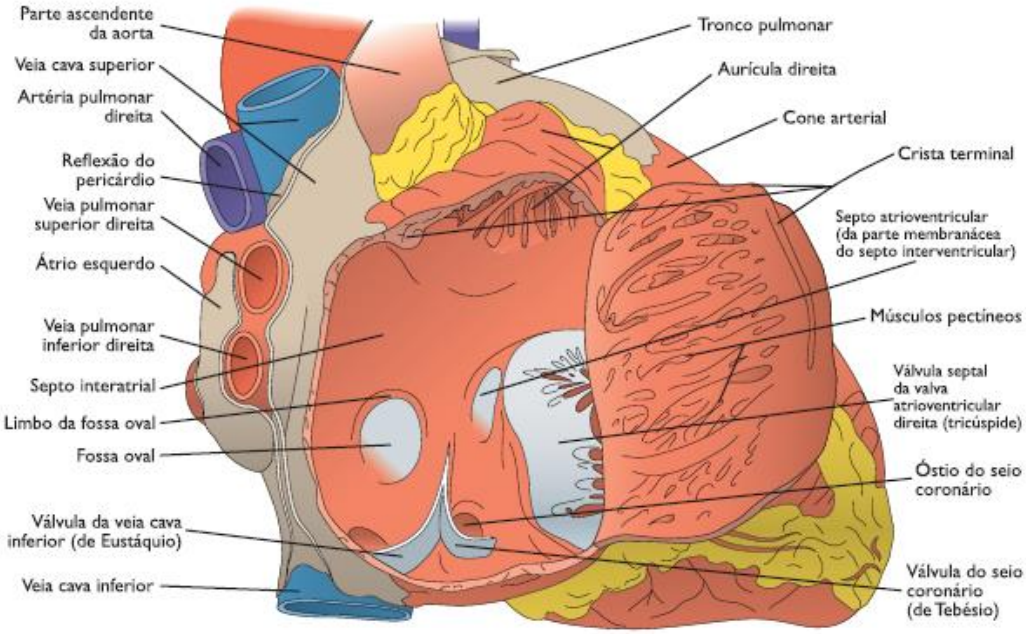
Em um último estágio, ocorre o enchimento ventricular, onde os ventrículos começam a encher-se devido ao sangue proveniente dos átrios, ou seja, as válvulas semilunares estão fechadas, enquanto as válvulas tricúspide e mitral estão abertas. Estando relaxados, os ventrículos são preenchidos em cerca de setenta por cento de suas capacidades máximas. (CARMO, 2021)

As válvulas atrioventriculares começam a se fechar à medida que a pressão atrial reduz e o volume ventricular aumenta. Então, deste modo os átrios começam a encher-se novamente.

O ritmo sinusal varia com uma frequência entre 50 e 100 bpm (LAPA, 2016).

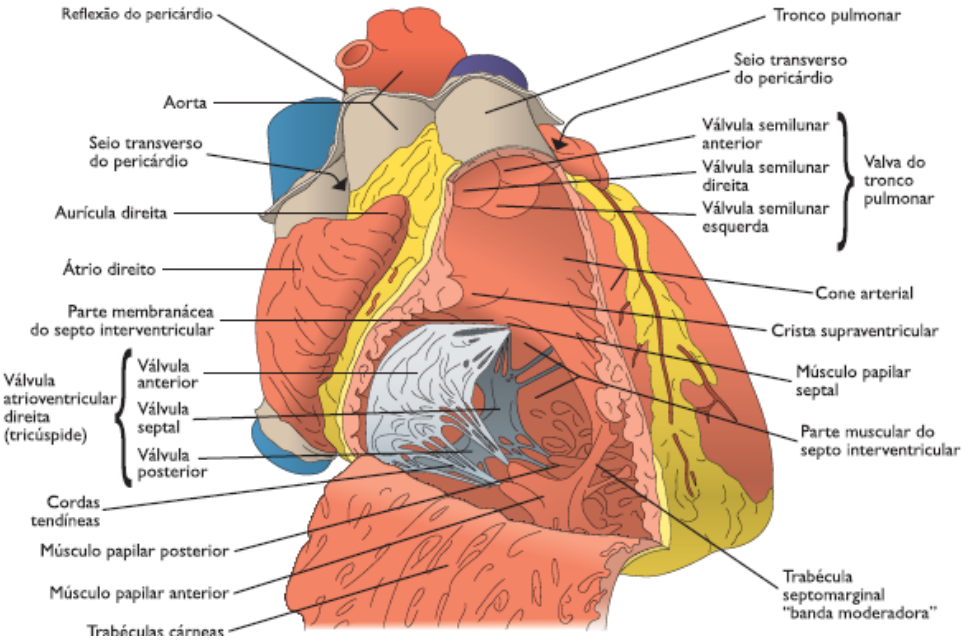
As figuras 1 e 2, mostram representações do coração, evidenciando suas partes.

Figura 1: Átrio direito (Vista lateral direita)



Fonte: (DUTRA, et al. 2009).

Figura 2: Ventrículo direito (Vista anterior).



Fonte: (DUTRA, et al. 2009).

3.1.1 Sistema Elétrico do coração

O coração possui dois tipos de célula: miocárdicas, também conhecidas por células funcionais, que quando estimuladas eletricamente se contraem; e as células marcapassos, que são responsáveis pelo estímulo elétrico. Existem alguns tecidos que são especializados em gerar e conduzir impulsos elétricos através do coração, sendo eles: nó sinodal (nó SA), nó atrioventricular (nó AV), feixe de His e fibras de Purkinje (SILVA, [s.d]).

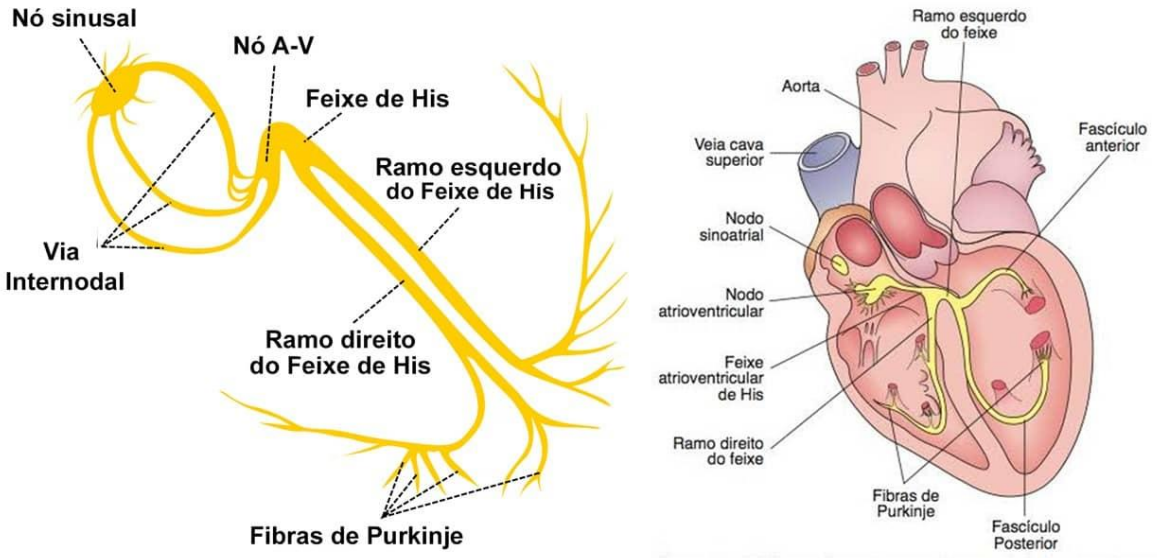
O controle da atividade cardíaca é realizado pelo sistema nervoso simpático e parassimpático, que inervam grande parte do coração. (SILVA, [s.d])

A figura 3 demonstra a estrutura elétrica do coração.

O nó SA está localizado na parede posterior do átrio direito, próximo as veias cavas. O nó AV está localizado na porção inferior do septo interatrial. O feixe de His está no septo interventricular, de modo que se divide no interior da parede dos ventrículos, tratando-se, portanto, da fibra de Purkinje, a qual causa a contração simultânea dos ventrículos (SILVA, [s.d]).

A regulação do ritmo do coração ocorre no nó SA, considerado o marca passo do coração. As membranas do nó SA são muito permeáveis ao sódio, de tal modo que ao passar para o interior das fibras, faz com que o potencial da membrana em repouso passe para o valor positivo até atingir o seu limiar, transformando então em potencial de ação. Este impulso é propagado pelos átrios provocando sua contração. Alguns centésimos de segundo depois, o impulso atinge o nó AV, retardando o impulso para os ventrículos, para dar tempo a passagem de sangue. Em seguida, o impulso é propagado pelo feixe de His e Purkinje aos ventrículos, para dar início à contração (SILVA, [s.d])

Figura 3: Estrutura Elétrica do coração



Fonte: (SILVA, 2018).

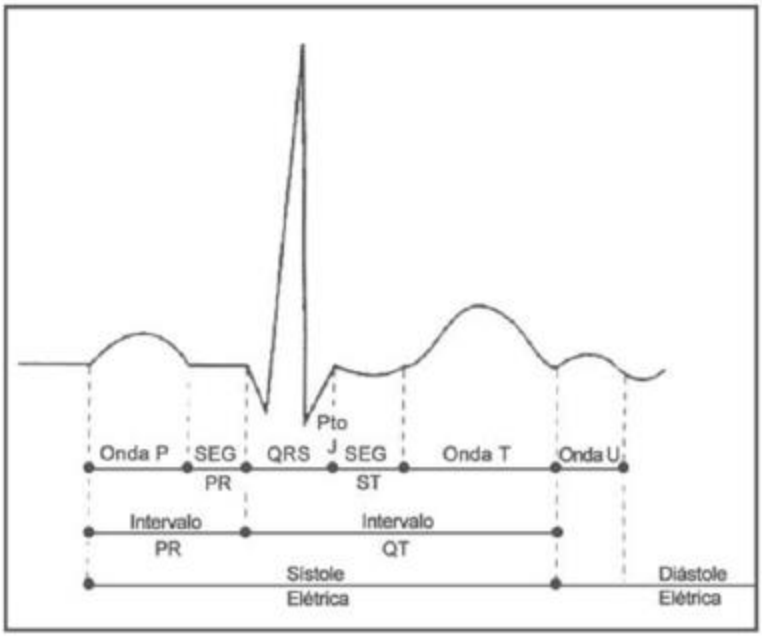
3.2 MONITORAMENTO DE SINAIS CARDÍACOS

Esta seção tem o objetivo de discorrer sobre as principais características de um sinal de eletrocardiograma.

3.2.1 Características do Eletrocardiograma

O Eletrocardiograma (ECG) é o registro da atividade elétrica do coração através de eletrodos sobre a pele. Desta forma, o ECG é composto por uma série de ondas geradas por cada fase do ciclo cardíaco. A figura 4 mostra o traçado clássico do ECG, evidenciando as ondas P, Q, R S e T, com o eventual aparecimento da onda U. As linhas entre as ondas, denominadas segmentos, também possuem relevância para a análise do sinal ECG (ESPINDOLA, 2018).

Figura 4: Elementos do eletrocardiograma (normal)



Fonte: (FELDMAN e GOLDWASSER, 2004) apud (ESPINDOLA, 2018)

- **Onda P:** despolarização atrial, responsável pela contração dos átrios.
- **Segmento PR:** passagem do estímulo elétrico pelo nodo átrio-ventricular.
- **Intervalo PR:** intervalo entre o início da onda P e início do complexo QRS. Indica a velocidade de condução entre os átrios e os ventrículos, assim como o tempo de condução do impulso elétrico desde o nodo sinusal até os ventrículos.
- **Complexo QRS:** evidencia a despolarização e o início da contração dos ventrículos.

- **Segmento ST:** período de inatividade entre a despolarização e o início da repolarização ventricular.
- **Onda T:** fase de repolarização dos ventrículos, em que as fibras ventriculares começam a relaxar.
- **Intervalo QT:** conjunto da despolarização e repolarização dos ventrículos.
- **Intervalo RR:** intervalo entre duas ondas R. Corresponde a frequência de despolarização ventricular, utilizado para o cálculo da frequência cardíaca.

3.3 ARRITMIA CARDÍACA

Arritmia cardíaca ou disritmia cardíaca é uma cardiopatia caracterizada por alterações do ritmo ou da frequência dos batimentos cardíacos. As arritmias cardíacas podem resultar de qualquer anomalia nos diferentes componentes deste circuito elétrico e podem classificar-se de acordo com as alterações que provocam na frequência cardíaca (taquicardia, bradicardia, pausas) (MARGATO, 2021).

Nesta subseção, será abordado as cinco principais arritmias cardíacas segundo a AAMI (Association for the Advancement of Medical Instrumentation): Normais (N); supraventriculares ectópicos (SVP); ventriculares ectópicos (PVC); fusão de normais e ventriculares (FVN); desconhecido ou com marca-passo (FPN).

- **Normal e bloqueio (N)**

Composto por batimentos normais, bloqueio de ramo esquerdo, bloqueio de ramo direito e batimento de escape atrial. (CARDOZO, 2018)

Os batimentos normais seguem os padrões eletrocardiográficos descritos no item 3.2.1.

O bloqueio de ramo direito e esquerdo envolve a interrupção do sinal elétrico cardíaco pelos ramos, provocando uma alteração na onda QRS com duração superior a 120ms. Não há sintomas característicos de sua ação no corpo humano, e, normalmente, não provocam grandes complicações. (MITCHELL , [s.d.])

- **Supraventriculares ectópicos (SVE)**

Composto por extrassístole atrial, extrassístole atrial com aberrância de condução e extrassístole juncional. As extrassístoles são batimentos cardíacos extras, originados nos átrios ou ventrículos, que alternam com o ritmo cardíaco normal. Originam batimentos cardíacos irregulares e podem causar sintomas como palpitações, tonturas, sensação de mal-estar torácico etc. (CARDOZO, 2018)

- **Ventriculares ectópicos (VE)**

Batimentos ventriculares ectópicos são originados de uma região anormal do miocárdio cuja classificação pode ser extrassístole ou escape. (CARDOZO, 2018)

A extrassístole possui duração anormal, geralmente superior a 120 milissegundos, a onda T possui contração negativa contrária aos batimentos normais. (DE SOUSA E SILVA, 2012)

- **Fusão de normais e ventriculares (F)**

Composto por batimentos de fusão. (CARDOZO, 2018)

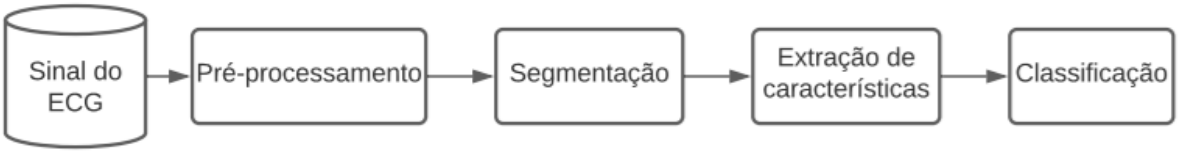
- **Desconhecido ou com marca-passo (D)**

Composto por parassístole ventricular, fibrilação atrial, fusão de ritmos e batimentos não classificados. (CARDOZO, 2018)

3.4 INTELIGÊNCIA ARTIFICIAL E *MACHINE LEARNING*

Em geral, as técnicas computadorizadas para classificação de arritmias cardíacas abrangem as etapas mostradas na Figura 5. (CARDOZO, 2018)

Figura 5: Etapas técnicas computadorizadas.



Fonte: Autores.

A primeira etapa de pré-processamento tem como objetivo filtrar ruídos externos e estimar a localização do complexo QRS. A etapa de segmentação tem como objetivo detectar os limites do complexo QRS e a onda P e T. Já a etapa de extração de características, obtém informações relevantes do sinal, capazes de distinguir os tipos analisados, além de reduzir a dimensionalidade da matriz de dados originais. Uma técnica interessante, no caso do ECG, é a transformada de Wavelet, pela sua capacidade de análise em tempo-frequência. (CARDOZO, 2018)

Por fim, temos a etapa de classificação por *machine learning*, cuja abrangência de métodos aplicáveis ao objetivo proposto deste trabalho é vasta. O aprendizado de máquina tem a finalidade de analisar uma grande quantidade de dados estruturados.

3.4.1 Algoritmos de aprendizado de máquina

Os algoritmos de aprendizado de máquina utilizam dados existentes e aplicam diversas regras para formular uma equação que descreva a relação destes dados. Cada algoritmo apresenta regras específicas, e a equação gerada determina seu desempenho que é relativo para cada base de dados. (CHINNAMGARI, 2019)

Os algoritmos de aprendizado de máquina são comumente categorizados em:

- **Supervisionados**: O algoritmo de aprendizado de máquina é treinado a partir de dados rotulados, produzindo uma função inferida capaz de gerar valores de previsão. O sistema é capaz de comparar a saída obtida com a saída planejada, evidenciando erros e alvos dos sinais em estudo (CHINNAMGARI, 2019).
- **Não-supervisionados**: O algoritmo de aprendizado de máquina é treinado a partir de dados não rotulados nem classificados. O sistema não possui um referencial para calcular a saída correta de um dado de entrada, mas pode extrair inferências a partir de seu estudo cumulativo, descrevendo estruturas desconhecidas (CHINNAMGARI, 2019).
- **Semissupervisionados**: É um método híbrido, pois abrange tanto a técnica supervisionada quanto a não-supervisionada, visto que é utilizado dados rotulados e não rotulados como base para o aprendizado de máquina (CHINNAMGARI, 2019).
- **Aprendizado por reforço**: É um método nem supervisionado nem não-supervisionado, visto que não é utilizado dados rotulados, mas também não são dados não-rotulados devido a identificação prévia no algoritmo dos comportamentos que devem ser mantidos daqueles que não devem durante o aprendizado de máquina. A técnica visa melhorar as estratégias do algoritmo através do feedback de recompensa, denominado de sinal de reforço, tornando o sistema volátil e interativo com o seu ambiente. (CHINNAMGARI, 2019)

A base de dados de eletrocardiogramas enquadra-se em algoritmos de aprendizado supervisionado, pois apresentam sua respectiva classificação de arritmia. (ADDAN, 2019) Algum dos algoritmos mais comuns para essa categoria são:

- ***Decision Tree* (Árvore de decisão)**: É um algoritmo supervisionado que contém nós de decisão até que se encontre a raiz, ou seja, a classificação ideal para o dado de entrada. (ADDAN, 2019)
- **Naïve Bayes:** É um classificador probabilístico baseado no teorema de Bayes. Há muitas variantes: Gaussian naïve Bayes, Multinomial naïve Bayes etc.
- ***Support Vector Machine (SVM)*:** É utilizado para problemas de regressão linear e classificação. Cada item de dados é plotado como um ponto no espaço n-dimensional. A classificação é executada encontrando-se o hiperplano que melhor diferencia as duas classes. (ADDAN, 2019)
- ***K-Nearest Neighbor (KNN)*:** O método de aprendizado classifica os dados de entrada de acordo com a maior proximidade que possui dos dados no diagrama. (ADDAN, 2019)
- ***Ensemble:*** O Aprendizado por Ensembles, é referido à utilização de múltiplos modelos em uma mesma tarefa de aprendizado de máquina. A forma mais simples de utilização, consiste em treinar diversos modelos distintos sobre todos os dados disponíveis para treinamento, aplicando desta forma, todos os testes modelos para cada exemplo de teste. Em modo de classificação, cada modelo terá uma decisão, e podemos decidir a classe resultante. Em tarefas de regressão, podemos tomar uma média das saídas de cada modelo, e tomá-la como resultante do ensemble. (ARAUJO, 2020)

4 REVISÃO BIBLIOGRÁFICA

Nessa seção será mostrado o levantamento das técnicas mais modernas de aquisição de eletrocardiograma e detecção de arritmia por *machine learning*.

4.1 CLASSIFICAÇÃO DE ARRITMIA POR *MACHINE LEARNING*

(KUILA, 2020) realizaram um trabalho de classificação e extração de características de arritmias em sinais de eletrocardiograma, utilizando técnicas de aprendizado de máquina. Neste estudo, utilizaram o módulo NI6008 para aquisição do sinal de ECG filtraram o sinal com filtros digitais com banda de 0,05Hz a 113Hz, e aplicaram as transformadas de Wavelet e de Fourier para o processamento do sinal. Para o treinamento do algoritmo, foi utilizado o banco de dados MIT-BIH Arrhythmia Database (MOODY & MARK, 2001), cujos dados foram coletados desde 1960, e aplicaram a metodologia supervisionada de aprendizado de máquina, dividindo os dados em um conjunto de treinamento e outro de teste. Posteriormente, os autores aplicaram vários algoritmos para encontrar a técnica com melhor acurácia. A pesquisa argumenta que a segmentação do sinal ECG, também interfere na análise e classificação das arritmias, onde diversos parâmetros podem ser utilizados como referência, como: onda T, onda P, detecção do complexo QRS e pico R. Por fim, os resultados mostraram que as técnicas de *Support Vector Machine (SVM), K-Nearest Neighbor (KNN) e Artificial Neural Network (ANN)* foram as mais adequadas para este propósito (KUILA et al, 2020)

Outro trabalho semelhante, desenvolvido por Nikan, Gwadry-Sridhar e Bauer (2017) utilizou os bancos de dados da PhysioNet e MIT-BIH, paraaplicar o aprendizado de máquina supervisionado. Foram selecionadas quatro categorias de sinais disponíveis no banco de dados, sendo eles: Batimentos normais (N), batimentos de fusão (F), batimentos supraventriculares ectópicos (V) e batimentos desconhecidos ou ilegíveis (U) para serem classificados a fim de minimizar os ruídos utiliza um filtro elíptico passa-banda digital, com banda de passagem de 5 à 15Hz que pode ser obtido por uma combinação de um filtro passa-baixa e um filtro passa-alta. A frequência para o filtro passa-baixa, é obtida através da função de transferência das equações 1 e 2, enquanto a frequência utilizada para o filtro passa-alta pode ser obtida por meio da função de transferência das equações 3 e 4 (NIKAN, GWADRY-SRIDHAR e BAUER, 2017).

$$H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}, \quad (1)$$

$$|H(\omega T)| = \frac{sin^2(3\omega T)}{sin^2(\frac{\omega T}{2})} \quad \text{where T is sampling period} \quad (2)$$

$$H(z) = z^{-16} - \frac{(1-z^{-32})}{(1-z^{-1})}. \quad (3)$$

The proposed high-pass filter has the following amplitude response.

$$|H(\omega T)| = \frac{[256 + sin^2(16\omega T)]^{0.5}}{cos(\frac{\omega T}{2})}. \quad (4)$$

Após a aquisição e tratamento do sinal, a técnica de segmentação utilizada pelo grupo foi a detecção de pico R. Desta forma, para cada ECG gravado, foi calculada a média do intervalo de picos R-R consecutivos, e adaptado como duração correspondente de cada batimento.

O grupo propôs ainda a utilização da técnica de algoritmo combinado da transforma da discreta de Wavelet e 1-D LBP (*Local Binary Patterns*). Utilizando 90% do banco de dados para treino e 10% para testes, obtiveram uma acurácia de 98,99% (NIKAN, GWADRY-SRIDHAR e BAUER, 2017).

(MARY, PANDEY, *et al.*, 2021) desenvolveram um sistema de diagnóstico de arritmia em tempo real utilizando SVM como algoritmo de aprendizagem de máquina e a transformada discreta de Wavelet para a extração de padrões. Os autores implementaram a análise de sinal de ECG no microcontrolador da placa ESP32, apresentando uma acurácia de 98%. O algoritmo escolhido detecta, automaticamente, os pontos cruciais para a análise de ECG, denominados como onda P, complexo QRS e onda T.

5 METODOLOGIA

Este capítulo aborda as metodologias definidas para cumprimento dos objetivos deste projeto.

5.1 VISÃO GERAL DO SISTEMA

A fundamentação teórica consolida os pontos que devem ser observados, expandindo o conhecimento sobre o coração, ECG e *Machine Learning.*

Agora torna-se possível a idealização dos *softwares* que regerão o monitoramento ECG, assim como os componentes que caracterizarão os circuitos.

A proposta deste projeto é o desenvolvimento de um dispositivo, no qual são dispostos eletrodos que permitem o monitoramento do sinal de ECG. A fim de otimizar o custo-benefício do sistema, propõe-se que o dispositivo seja capaz de ser conectado à um computador e, que por meio deste, seja possível fazer a análise do sinal de ECG para a uma possível detecção de arritmia cardíaca, enviando o diagnóstico via e-mail para um profissional da saúde.

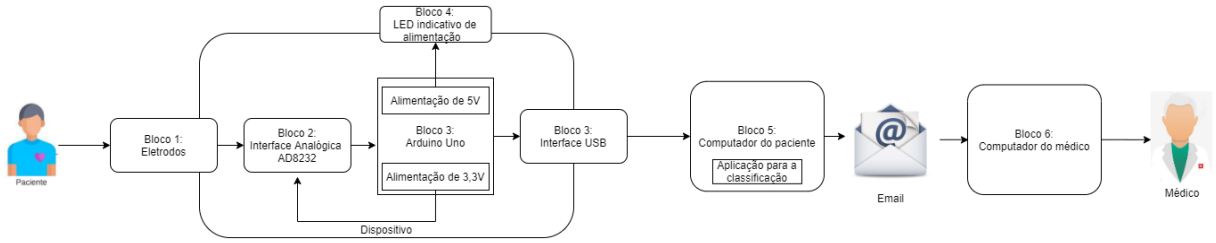
5.2 PROJETO DE *HARDWARE*

Nesta seção será apresentado o projeto de hardware do dispositivo principal, bem como a descrição dos principais componentes que subsidiam o projeto.

5.2.1 Diagrama de blocos

Para implementar o circuito do projeto, foi montado um diagrama de blocos de alto nível que demonstra os principais componentes presentes no projeto. É possível observar esse diagrama na figura 7, evidenciando o AD8232 que é responsável pela aquisição e pré-processamento do ECG e o microcontrolador Arduino Uno responsável pela interface do módulo AD8232 e computador do paciente. Cada bloco será explicado conforme a descrição dos componentes no próximo item.

Figura 6: Diagrama de blocos do sistema.



Fonte: Autores.

5.2.2 Descrição funcional e configuração dos principais componentes

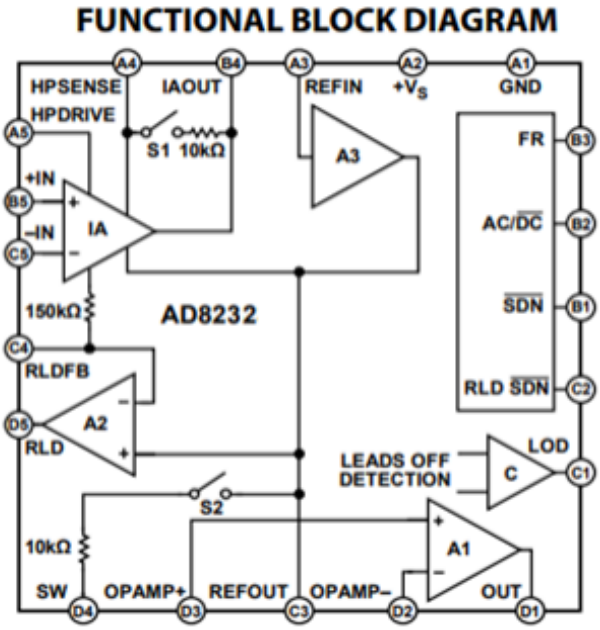
Esta seção tem como objetivo descrever os principais componentes utilizados no projeto.

5.2.2.1 Interface analógica AD8232

O objetivo deste circuito integrado é adquirir, amplificar e filtrar os sinais de ECG, servindo de *front-end* para os conversores analógico-digital do microcontrolador e posterior análise.

O diagrama funcional do AD8232 mostrado na figura 8 apresenta um amplificador de instrumentação (IA) para a amplificação do sinal, um amplificador operacional de propósito geral (A1) que pode ser usado para implementação de filtros com componentes externos e amplificação do parâmetro, um circuito de restauração rápida (A3), um circuito que detecta mau contato nos eletrodos (C) e um circuito de realimentação de perna direita (A2) como principais blocos.(Folha de dados do AD8232)

Figura 7: Diagrama de blocos funcional do AD8232

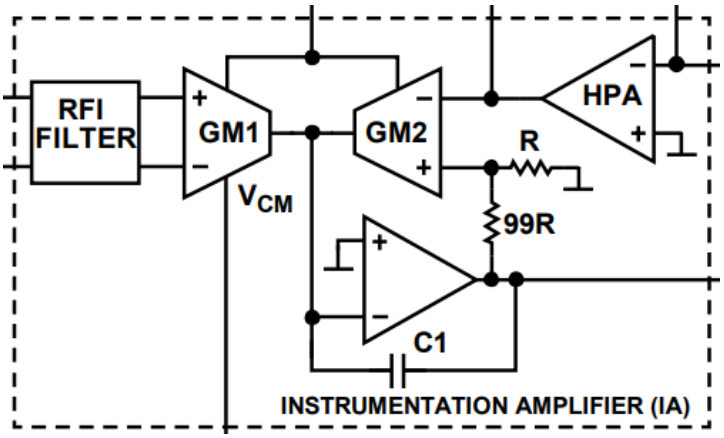


Fonte: Folha de dados do AD8232.

O amplificador de instrumentação é o bloco principal do AD8232, e seu diagrama esquemático é mostrado na figura 9. Este circuito recebe o sinal de ECG através das entradas B5 e C5, passa por um filtro para atenuar interferências de rádio frequência, por dois

amplificadores de transcondutância casados, por um amplificador passa-alta e por um circuito integrador. Este circuito utiliza a resistência e a capacitância de entrada para a implementação de um filtro passa-baixa para a eliminação das interferências de alta frequência. Essa topologia é realimentada por duas vias. A primeira via é através da associação de resistência na saída do integrador que realimenta até a entrada do GM2. A segunda realimentação se dá através do amplificador passa-alta que tem o objetivo de atenuar o nível médio do sinal.

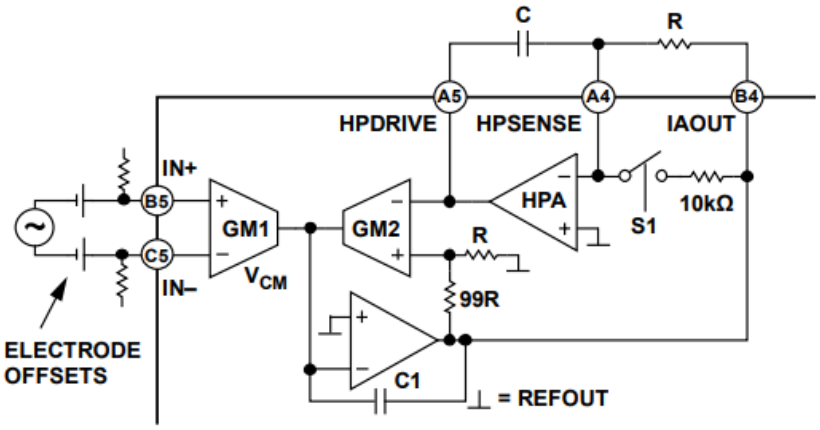
Figura 8: Diagrama esquemático do amplificador de instrumentação do AD8232



Fonte: Folha de dados do AD8232.

Para implementar o módulo, o fabricante disponibiliza algumas equações para a determinação das frequências de corte de filtros necessários para o acondicionamento do sinal.

Figura 9: Filtro passa-alta implementado no AD8232



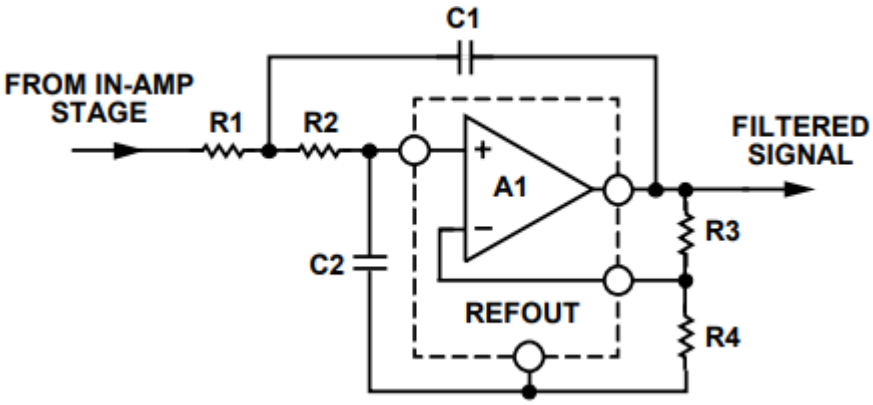
Fonte: Folha de dados do AD8232.

A figura 10 apresenta a montagem de um filtro passa-alta à fim de rejeitar sinais de baixas frequências localizadas entre HPDRIVE e HPSENSE, HPSENSE e IAOUT. A frequência de corte deste filtro é dada pela equação (1).

$$f_c = \frac{100}{2\pi RC} \tag{1}$$

O AD8232 apresenta um amplificador operacional que pode ser utilizado para o controle de ganho e filtragem do sinal. Na aplicação foi implementado um filtro passa-baixa na topologia Sallen-Key, como mostrado na Figura 10.

Figura 10: Esquemático para o filtro passa-baixas.



Fonte: Folha de dados do AD8232.

Para a implementação dos filtros é necessário utilizar a equação (5) para determinar a frequência de corte, a equação (6) para determinar o ganho e equação (7) para determinar o fator de qualidade.

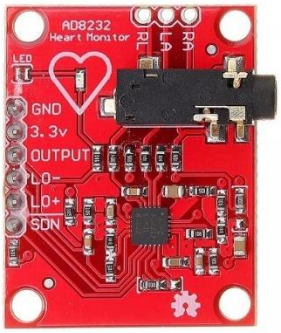
$$f_c = \frac{1}{2\pi\sqrt{R1 \times C1 \times R2 \times C2}} \tag{5}$$

$$Ganho = 1 + \frac{R3}{R4} \tag{6}$$

$$Q = \frac{1}{2\pi\sqrt{R1 \times C1 \times R2 \times C2}} \tag{7}$$

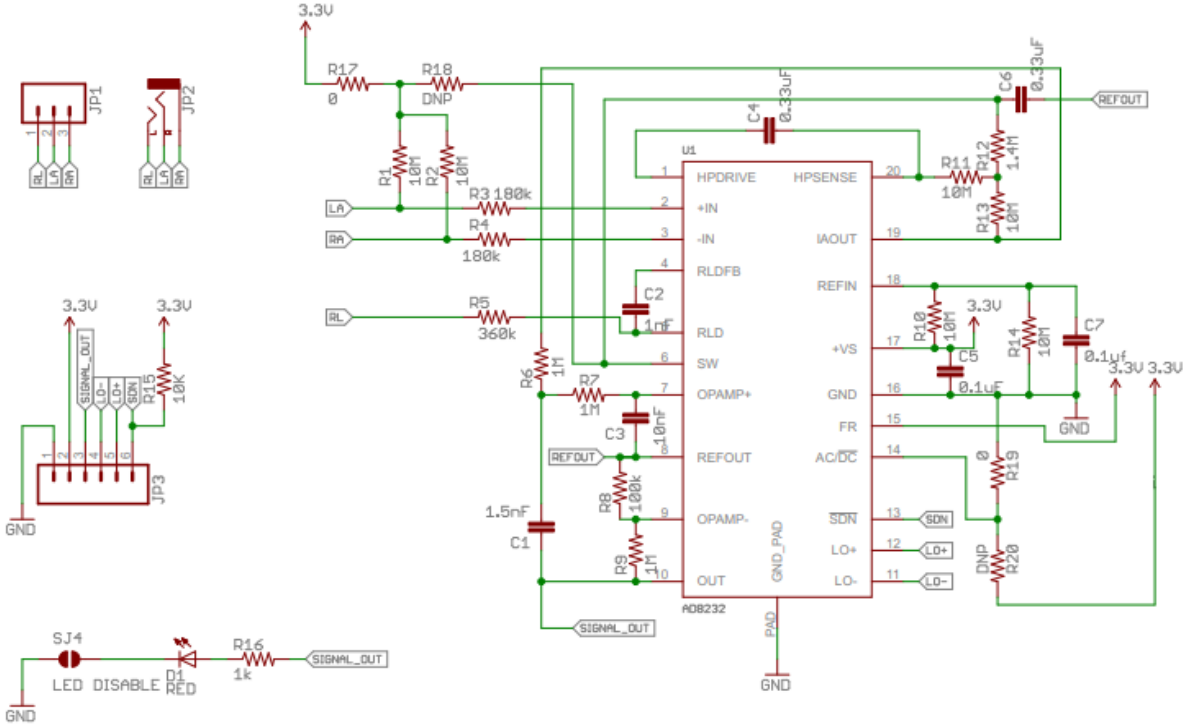
Para a implementação do dispositivo, foi utilizado um módulo do AD8232 para fazer a interface analógica, pela sua praticidade para as conexões, pois manipular o circuito integrado do AD8232 seria muito laborioso pelas dimensões e tecnologia de soldagem do dispositivo smd. Pode-se ver o módulo na Figura 11, bem como seu circuito elétrico na Figura 12.

Figura 11: Módulo AD8232



Fonte: (COELHO, 2020)

Figura 12: Diagrama de circuito do módulo AD8232



Fonte: Folha de dados AD8232.

5.2.2.2 *Microcontrolador Arduino UNO*

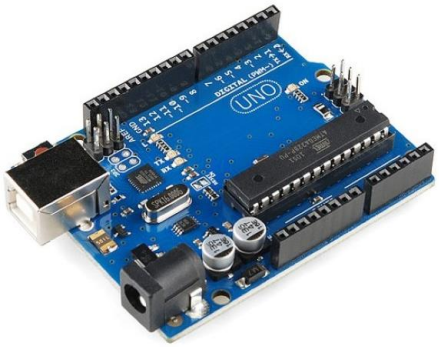
O microcontrolador escolhido tem a funcionalidade de integrar o módulo de interface analógica AD8232 com o computador do paciente, bem como aplicação feita no MATLAB. Foi utilizado o Arduino UNO devido a sua facilidade de integração com o *software* MATLAB e pela presença de portas analógicas para o monitoramento do sinal de ECG.

O Arduino UNO é um microcontrolador *open-source*, que permite a implementação de componentes externos através de entradas e saídas analógicas e digitais. O microprocessador

integrado é o ATmega328, e a tensão de operação da placa é 5V podendo ser alimentada através da conexão USB. Além disso, existem reguladores de tensão que permite a alimentação de módulos externos, sendo características importantes para implementação do projeto.

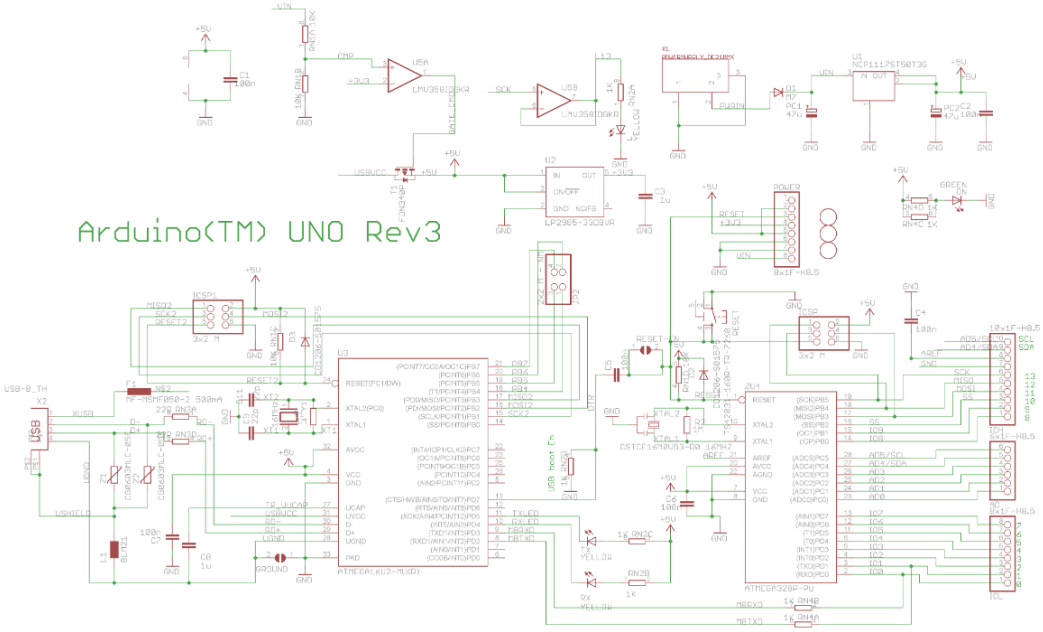
A Figura 13 apresenta a plataforma de computação física do Arduino UNO e a Figura 14 o diagrama de circuito.

Figura 13: Arduino UNO



Fonte: (COELHO, 2020)

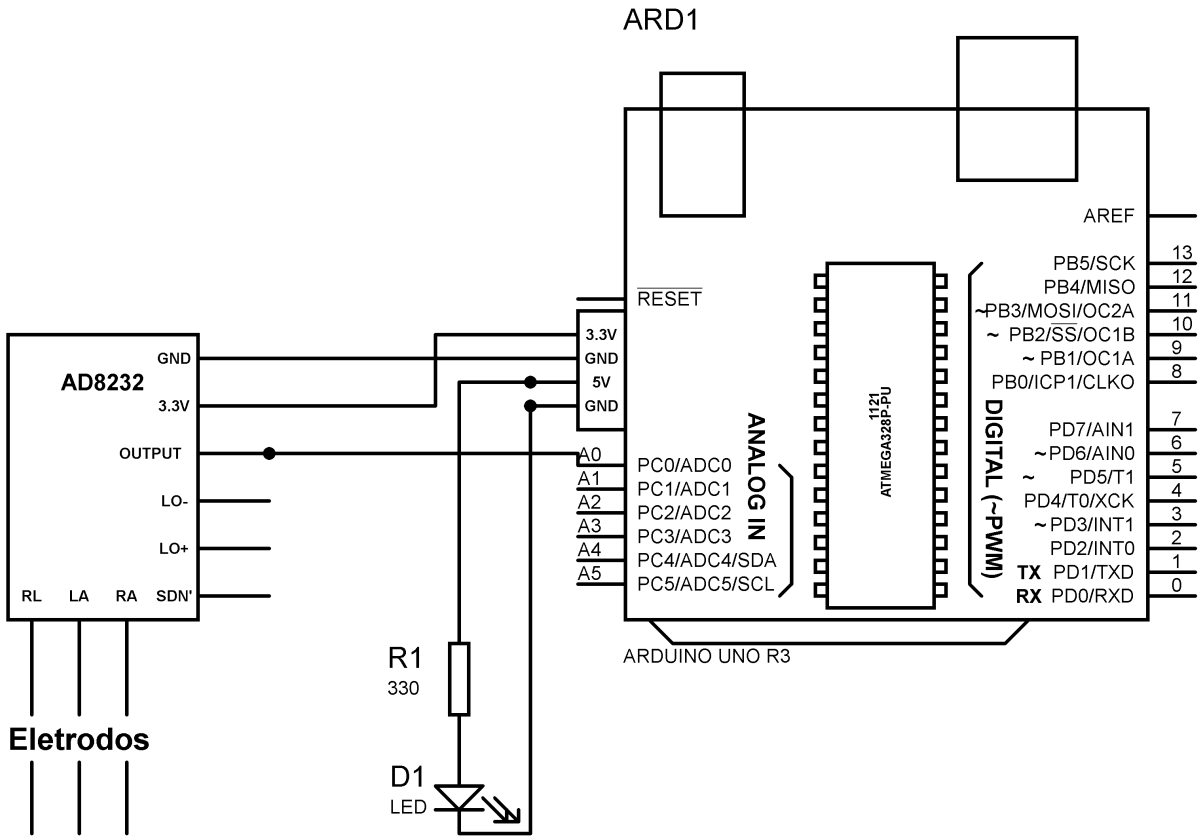
Figura 14: Diagrama de circuito Arduino UNO



Fonte: Folha de dados Arduino Uno.

5.2.3 Diagrama de interligação dos principais blocos do *hardware*

Figura 15: Circuito completo do sistema



Fonte:Autores

A Figura 15 mostra o circuito completo com as interligações dos principais blocos do *hardware.* A primeira topologia observada é a ligação da interface analógica do AD8232 com o microcontrolador. O componente é conectado ao microcontrolador através de dois terminais de alimentação e um terminal de saída analógico, que apresenta o sinal condicionado. O microcontrolador Arduino Uno é utilizado como interface do dispositivo com o computador do paciente, através do conector USB presente na placa. Além disso, há a presença de um LED indicativo de alimentação, utilizando os reguladores de 5V presentes na placa do microcontrolador.

5.3 PROJETO DE *SOFTWARE*

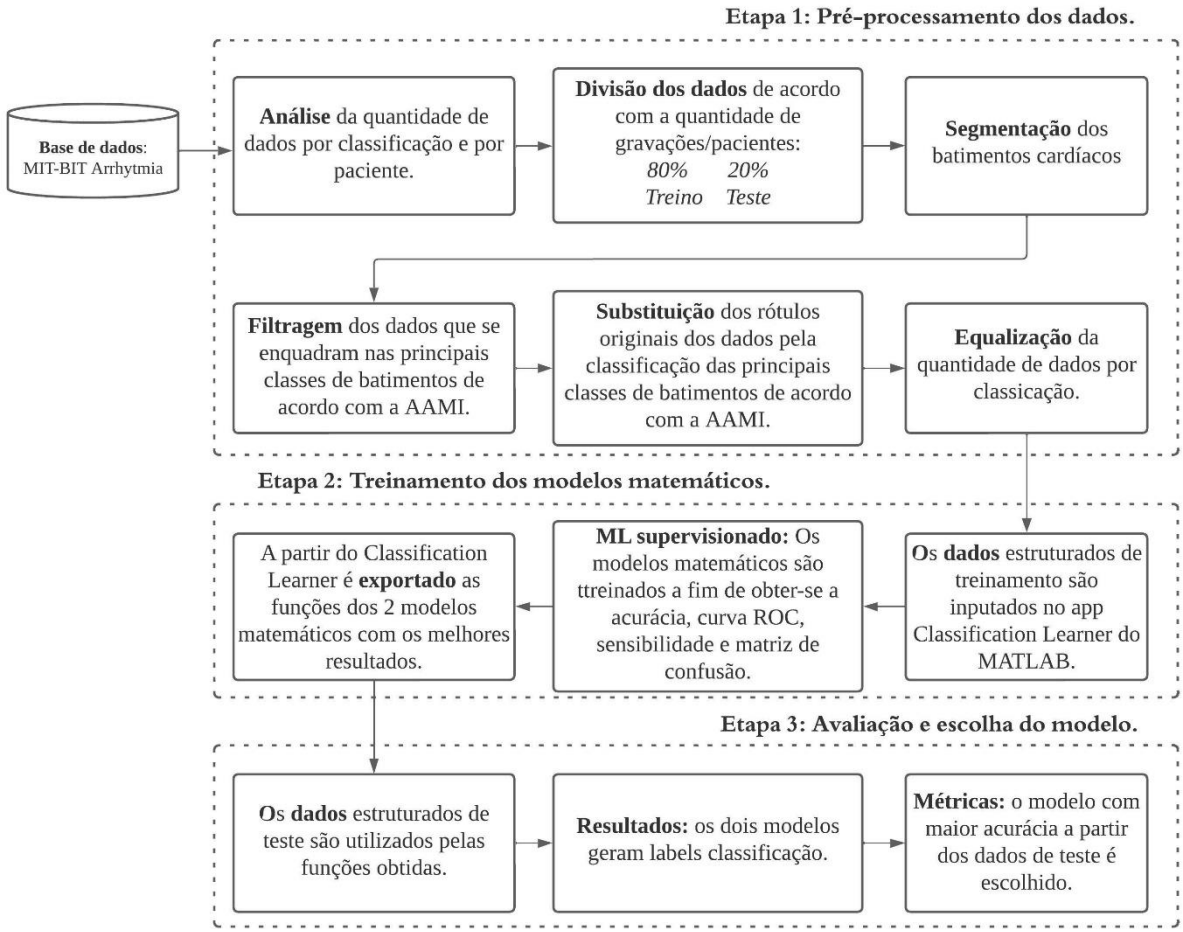
O projeto de software conta com três etapas fundamentais: a primeira refere-se à inteligência computacional desenvolvida para treinamento e teste do sistema de classificação de sinais de ECG. A segunda parte é o algoritmo implementado para aquisição do ECG. E a

terceira etapa é a interface de usuário, onde será apresentado os resultados obtidos através de um aplicativo.

5.3.1 Inteligência computacional para classificação de sinais de ECG

A Figura 16 apresenta um fluxograma, evidenciando as etapas da metodologia empregada. (Apêndice A)

Figura 16: Fluxograma da metodologia de processamento de dados



Fonte: Autores.

5.3.1.1 Base de dados

O banco de dados escolhido é o MIT-BIH Arrhythmia Database (MOODY e MARK, 2001) desenvolvido pelo Laboratório de Arritmias do Hospital Beth Israel de Boston (BIH, do inglês: Beth Israel Hospital) em parceria com o Instituto de Tecnologia de Massachusetts (MIT, do inglês: Massachusetts Institute of Technology – MIT).

Foi obtido em 1975 através de um gravador de fita com dois canais e o sinal foi transmitido para um computador através de um conversor analógico-digital de 11 bits. Depois de digitalizados, todos os registros foram impressos e entregues a dois cardiologistas para avaliação. Mesmo após avaliação, alguns registros permaneceram sem classificação devido à falta de consenso entre os médicos.

O banco contém 48 gravações de diferentes pacientes digitalizados na frequência de 360 amostras por segundo. De acordo com as recomendações da AAMI as 4 gravações que contém *Paced beats* (102, 104, 107, 217) são excluídas neste processo porque não retêm qualidade de sinal suficiente para um processamento confiável (DE ALBUQERQUE, *et al*., 2018)

Ao analisar previamente o conteúdo descritivo do banco de dados (MOODY, 1997), é possível observar que o *data set* é extremamente desbalanceado, e a derivação de aquisição de ECG presente em quase todas gravações é a MLII. Foram considerados os dados cujo rótulo integram 4 dos 5 principais grupos de classes de batimentos de acordo com a AAMI (Association for the Advancement of Medical Instrumentation), sendo:

- Normal (N)
- Supraventricular ectópico (SVE)
- Ventricular ectópico (VE)
- Fusão de normal e ventriculares (F)

A 5ª classe não considerada foi a “Desconhecidos ou com marca-passo” (D), pois possui uma quantidade insignificante de amostras.

A Tabela 1 mostra a equivalência entre as classes do banco de dados MIT-BIH e a classificação de acordo com a AAMI.

Tabela 1: Rótulos do banco de dados e sua descrição

Grupo AAMI	Classe MIT-BIH	Classe MIT-BIH (Descrição)
N	N	*Normal beat*
N	L	*Left bundle branch block beat*
N	R	*Right bundle branch block beat*
N	e	*Atrial escape beat*
N	j	*Nodal (junctional) escape beat*
SVE	A	*Atrial premature beat*
SVE	a	*Aberrated atrial premature beat*
SVE	J	*Nodal (junctional) premature beat*
SVE	S	*Supraventricular premature beat*

VE	V	*Premature ventricular contraction*
VE	E	*Ventricular escape beat*
F	F	*Fusion of ventricular and normal beat*
D	f	*Fusion of paced and normal beat*
D	/	*Paced beat*
D	Q	*Unclassifiable beat*
	[	*Start of ventricular flutter/fibrillation*
	!	*Ventricular flutter wave*
	]	*End of ventricular flutter/fibrillation*
	x	Non-conducted P-wave (blocked APB)
	|	Isolated QRS-like artifact

Fonte: (LIU, SHAO, *et al.*, 2020)

5.3.1.2 Pré-processamento do sinal

Para acessar o conteúdo do banco de dados Physionet nos formatos originais através do MATLAB foi utilizado as funções provenientes do WFDB Toolbox, que podem ser baixadas e estão disponíveis publicamente. (WFDB Toolbox for MATLAB and Octave, 2017) Dessa forma, foi possível dividir o banco de dados entre treino e teste, partindo da premissa de não utilização dos dados da mesma gravação (Figura 17).

Figura 17: Gravações utilizadas para dados de teste e treinamento

Treino

101	106	108	109	111	112	114	115	116	118
119	121	122	123	124	201	202	203	205	207
208	210	214	215	219	220	221	222	223	228
230	231	232	233	234					

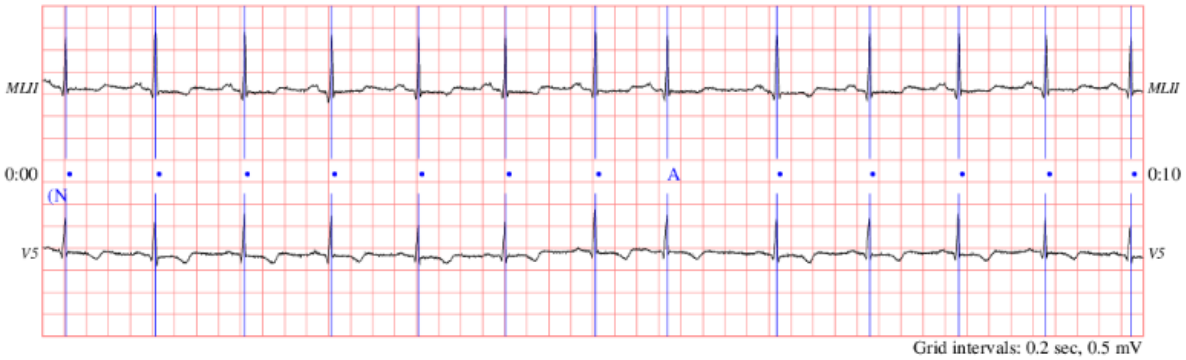
Teste

100	103	105	113	117	200	209	212	213

Fonte: Autores.

Observando os dados plotados (Figura 18) para a derivação MLII da gravação 100, constata-se que a marcação de tipo de batimento do banco de dados refere-se ao pico R do complexo QRS. A partir dos picos, os batimentos cardíacos foram segmentados com uma janela de 300 pontos (R-149 a R+150 pontos) assim como no trabalho feito por (SINGH e TEWARY, 2019).

Figura 18: ECG da gravação 100 do banco de dados.



Fonte: (Physiobank ATM)

Após a filtragem dos dados que se enquadram na classificação AAMI exceto “D”, obteve-se a quantidade de amostras por rótulo conforme a Tabela 2.

Tabela 2: Quantidade de amostras por rótulo após filtragem dos dados

Rótulo	Quantidade de amostras
N	70153
VE	5919
SVE	2296
F	439

Fonte: Autores

Para equalizar a quantidade de amostras, primeiramente os dados em cada linha foram trocados aleatoriamente de ordem, e as classes ‘N’ e ‘VE’ foram reduzidas ao tamanho da terceira maior classe ‘SVE’ (2296), dessa forma, foi considerado amostras de batimentos cardíacos de pacientes variados. Partindo desse pressuposto, a divisão do banco de dados representada na Figura 17 foi definida a partir de uma análise empírica a fim de alcançar o maior número de amostras para a classe ‘SVE’.

Por fim, o *data set* com as informações do sinal apresenta a dimensão 7326X300 *double*, conforme Tabela 3.

Tabela 3: Quantidade de amostras por rótulo após equalização

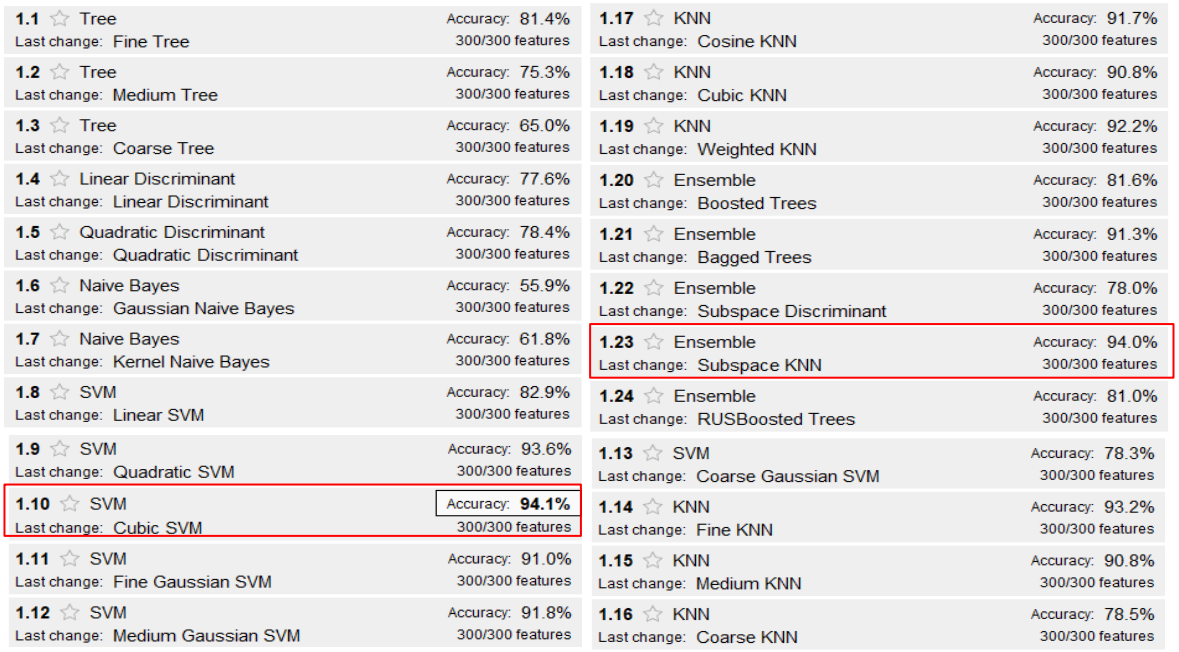
Rótulo	Quantidade de amostras
N	2296
VE	2296
SVE	2296
F	439

Fonte: Autores

5.3.1.3 Treinamento dos modelos

Após o conjunto de treinamento ser inserido no app *Classification Learner*, do MATLAB, os modelos matemáticos foram treinados e apresentaram as acurácias conforme a Figura 19.

Figura 19: Resultado do treinamento dos algoritmos utilizando o *Classifcation Learner*



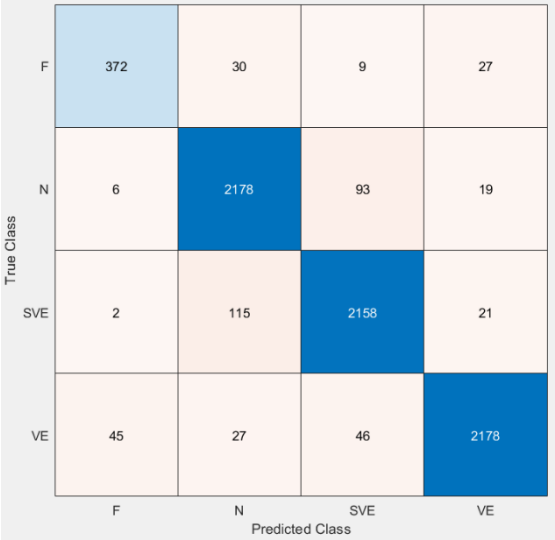
Fonte: Autores.

Os modelos SVM (Cubic SVM) e Ensemble (Subspace KNN) destacados no Figura 19 apresentaram as melhores acurácias.

5.3.2 Algoritmo de aquisição de ECG

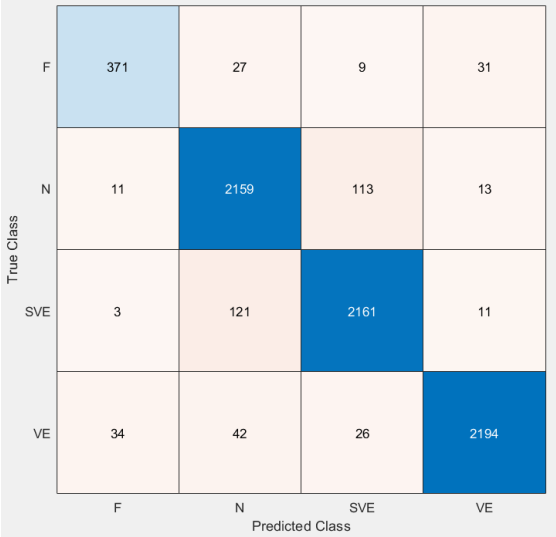
Através do *Classification Learner* é possível extrair as métricas de treinamento conforme Figura 20 e 21.

Figura 20: Matriz de confusão de treino Ensemble (Subspace KNN)



Fonte: Autores

Figura 21: Matriz de confusão de treino SVM (Cubic SVM)



Fonte: Autores

Os modelos matemáticos dos dois algoritmos foram exportados para o *workspace* do MATLAB, e os dados de teste utilizados como input.

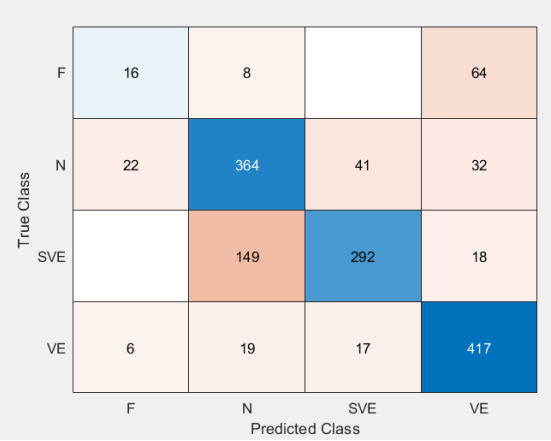
Os dados de teste foram estruturados nos mesmos padrões do treinamento, e o tamanho das amostras foi reduzido para que cada classe tivesse 20% dos dados, apresentando a dimensão 1465X300.

A partir dos resultados foi possível calcular a acurácia e obter a matriz de confusão. (Apêndice B)

$$acurácia = \frac{Total\ de\ acertos * 100}{Total\ de\ itens} \qquad (5)$$

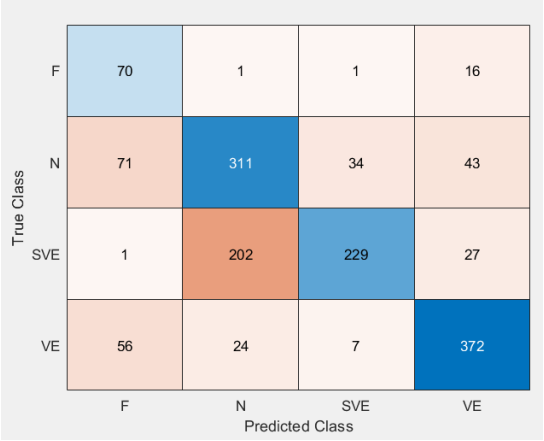
- Ensemble (Subspace KNN): 74,4%
- SVM (Cubic SVM): 67,1%

Figura 22: Matriz de confusão de teste Ensemble (Subspace KNN)



Fonte: Autores.

Figura 23: Matriz de confusão de teste SVM (Cubic SVM)



Fonte: Autores.

As matrizes de confusão dos dois modelos, apresentadas nas figuras 22 e 23,mostram que o modelo matemático que obteve melhor resultado e melhor eficiência no reconhecimento das arritmias foi o Ensemble (Subspace KNN), pois possui maior valor de acurácia de teste.

5.3.3 Interface de usuário

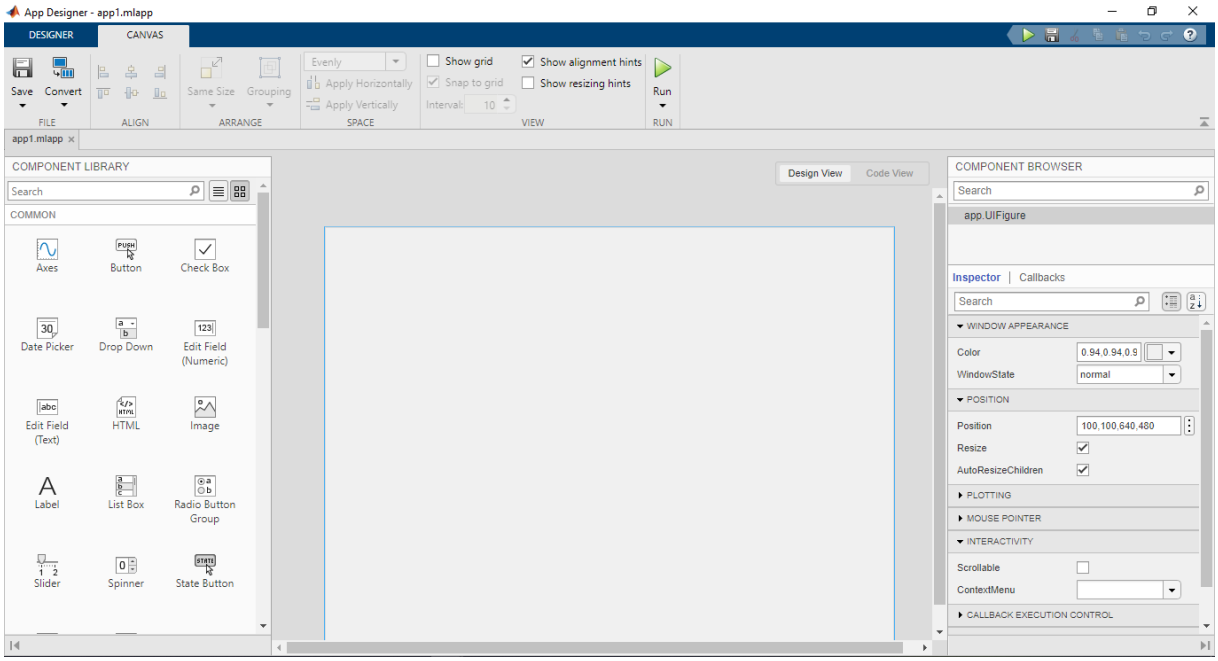
Com o intuito de facilitar a visualização, foi implementado um aplicativo para a interface do usuário e para com o algoritmo de detecção de arritmias. Este aplicativo foi desenvolvido através da plataforma de desenvolvimento do MATLAB, chamado *Appdesigner*.

5.3.3.1 App Designer

O ambiente *Appdesigner* é uma ferramenta disponibilizada no MATLAB para a criação de aplicativos baseados na linguagem de programação característica do próprio MATLAB, com a facilidade de implementação por apresentar as principais funções de maneira gráfica e sem necessidade de um conhecimento avançado em desenvolvimento de aplicações.

A metodologia de utilização é baseada em dois modos principais. O primeiro modo é o modo de design, onde é possível criar a interface gráfica, visualizando em tempo real a tela que o usuário irá observar, como é possível ver na Figura 24.

Figura 24: Modo *design*

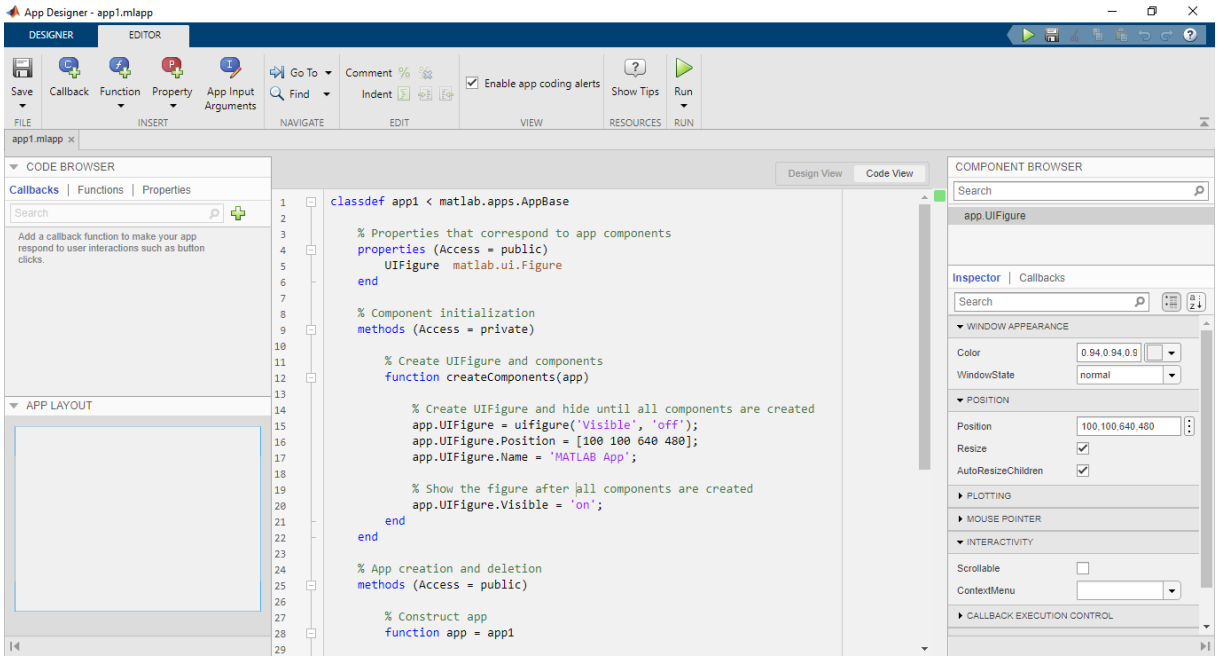


Fonte: Autores

Na Figura 24 é possível ver a tela inicial do aplicativo o *Appdesigner*. Do lado esquerdo é visto a biblioteca de componentes que podem ser adicionados, na parte central é vista a tela que o usuário final irá observar e do lado direito é visto as opções dos componentes que serão adicionados. Na parte superior, são vistas as principais opções de simulação, comumente visto no programa MATLAB.

O segundo modo de utilizar o aplicativo é o modo de visão de código, que pode ser visto na Figura 25, em que é observado o roteiro que descreve a interface, por meio de comandos escritos. Para facilitar o desenvolvimento, o código que descreve o design da aplicação já vem previamente escrito pelo aplicativo e não é possível fazer alterações desta natureza nesse modo de visão. A utilidade desse modo é a possibilidade de criar funções no código para exibir alguma informação de saída para o usuário na tela final ou funções que são acionadas quando o usuário requisita através de uma ação de entrada, como por exemplo o aperto de um botão.

Figura 25: Modo de visão de código



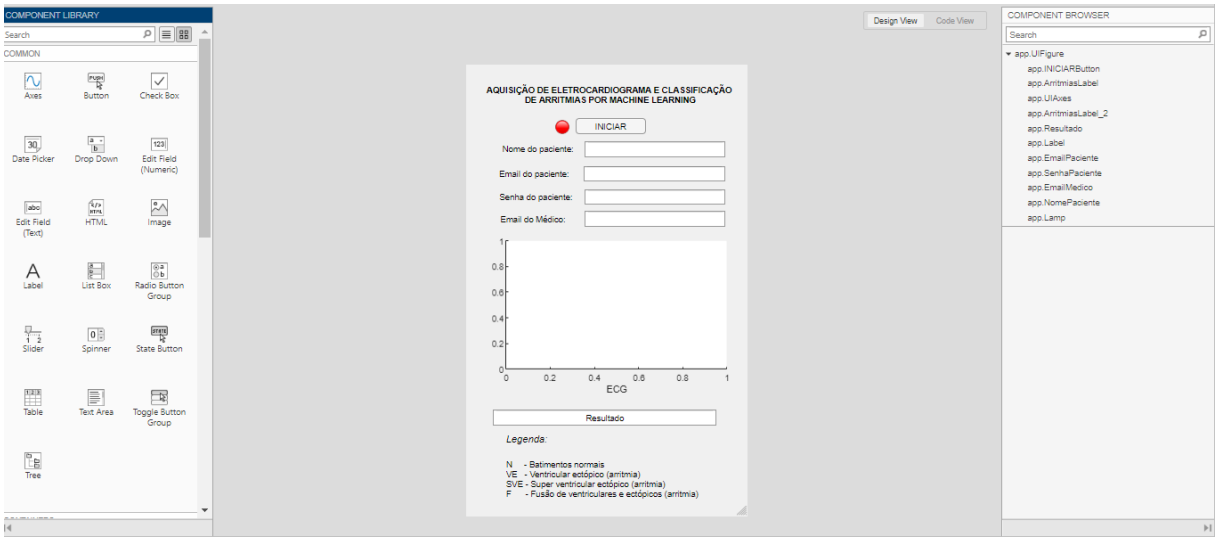
Fonte: Autores

De posse do aplicativo terminado, o *Appdesigner* permite a simulação dentro do ambiente e exportação do projeto final em um aplicativo executável para a utilização sem a necessidade do uso do MATLAB.

5.3.3.2 Desenvolvimento da Interface

A Figura 26 apresenta o design desenvolvido para o aplicativo final para a visualização do usuário. É visto o título do projeto, o botão para o início da aquisição do eletrocardiograma, o LED indicativo, os campos para a entrada das informações do paciente e médico, a área de plotagem do sinal adquirido, o campo para a apresentação do resultado e a legenda dos resultados possíveis.

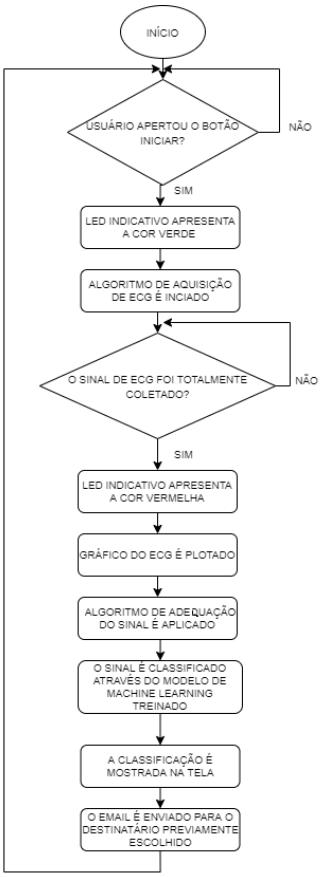
Figura 26: Design final do aplicativo



Fonte: Autores

A Figura 27 apresenta o fluxograma para implementação da lógica de funcionamento da aplicação. (Apêndice C) Este algoritmo é acionado cada vez que o usuário pressiona o botão iniciar.

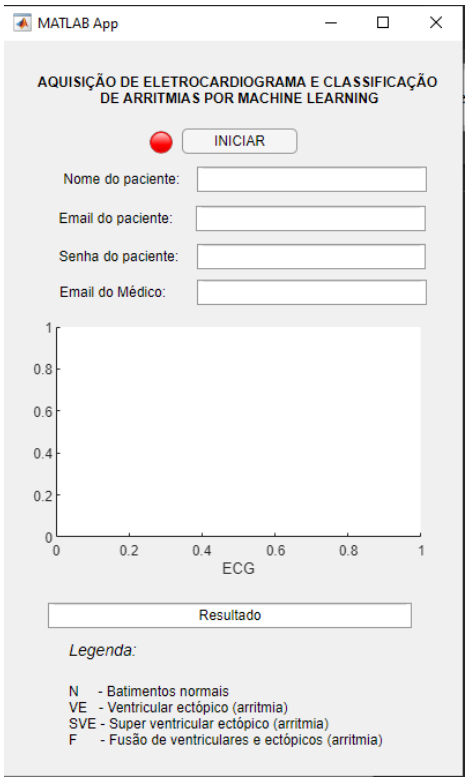
Figura 27: Fluxograma da lógica do aplicativo



Fonte: Autores

O ambiente permite a exportação do aplicativo em formato executável com a necessidade da instalação de um programa gerado pelo próprio *Appdesigner* com as ferramentas e bibliotecas necessárias para o bom funcionamento do algoritmo. A Figura 28 apresenta a visão final do aplicativo, já exportado.

Figura 28: Executável do aplicativo final



Fonte: Autores

5.4 ESTUDOS DE TÉCNICAS PARA AQUISIÇÃO DE ECG

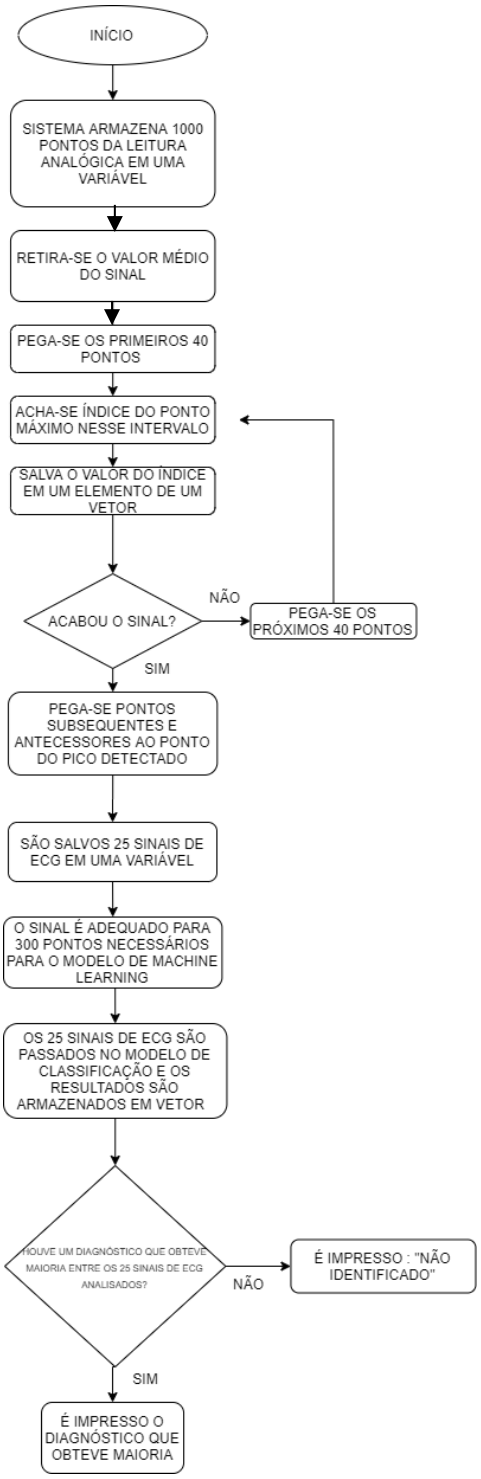
Esta seção tem como objetivo explicar como foi feito a aquisição do sinal de ECG pelo algoritmo do roteiro do MATLAB, bem como o tratamento deste sinal para a sua classificação a partir do modelo de aprendizagem de máquina treinado.

Para a integração do módulo AD8232 conectado ao Arduino Uno, foi utilizado uma biblioteca do MATLAB chamada *Arduino Support from MATLAB* que permite a utilização do microcontrolador através de comandos da linguagem MATLAB.

Com o controle do *hardware* pelo MATLAB, foi possível aplicar a leitura analógica da porta do microcontrolador e armazená-la em uma variável para o tratamento do sinal e aplicação no algoritmo de classificação.

O algoritmo de aquisição de ECG pode ser visto no fluxograma da Figura 29.

Figura 29: Fluxograma da aquisição do ECG



Fonte: Autores

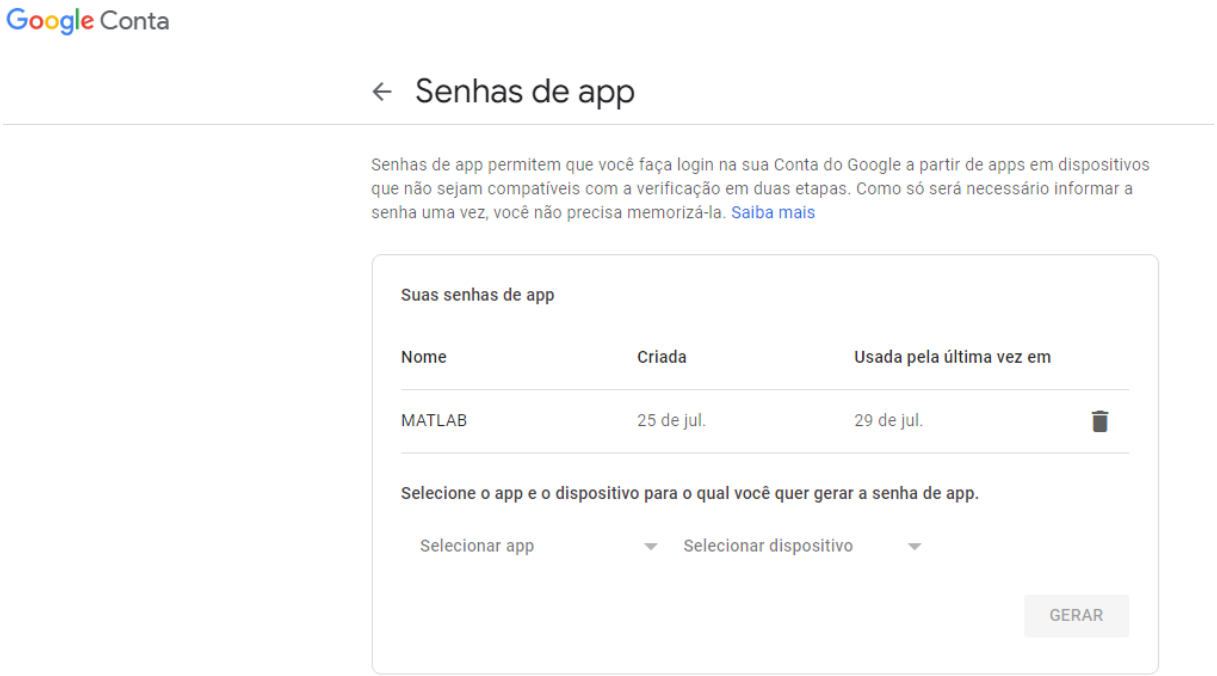
5.5 MODO DE USO

Esta seção tem como objetivo explicar como o paciente deve utilizar o dispositivo e o aplicativo para obter o diagnóstico e enviá-lo para o profissional de saúde estabelecido.

Primeiramente, é necessário a instalação de um ambiente para a aplicação desenvolvida. Este ambiente é disponibilizado pelo aplicativo *Appdesigner*, através da geração de umexecutável para a distribuição para os usuários finais.

Com o aplicativo aberto, o paciente deve informar os dados pessoais mostrados na Figura 28. A senha do usuário deve ser gerada nas configurações do provedor de e-mail do usuário como visto na Figura 30.

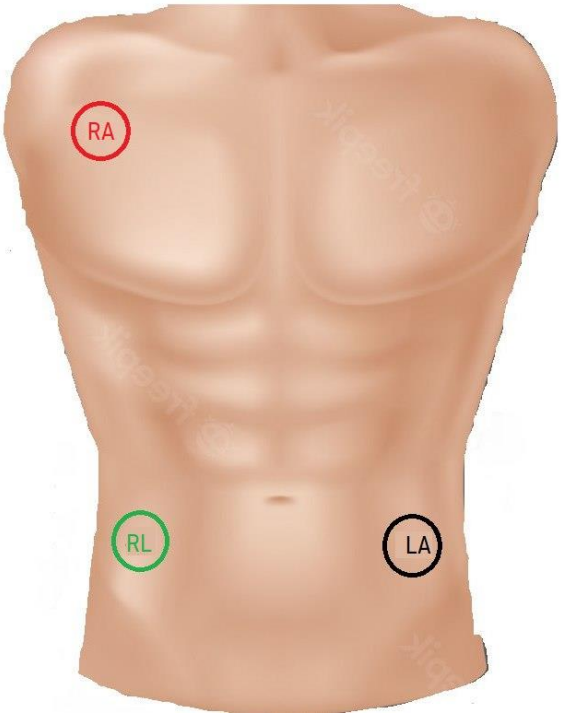
Figura 30: Configurações das senhas de aplicativo do Gmail



Fonte: Autores

Com todos os campos de informação preenchidos, o paciente deve ligar o dispositivo via conexão USB ao computador e conectar os cabos aos eletrodos dispostos como visto na Figura 31, respeitando a derivação MLII, a mesma usada para os dados de treinamento.

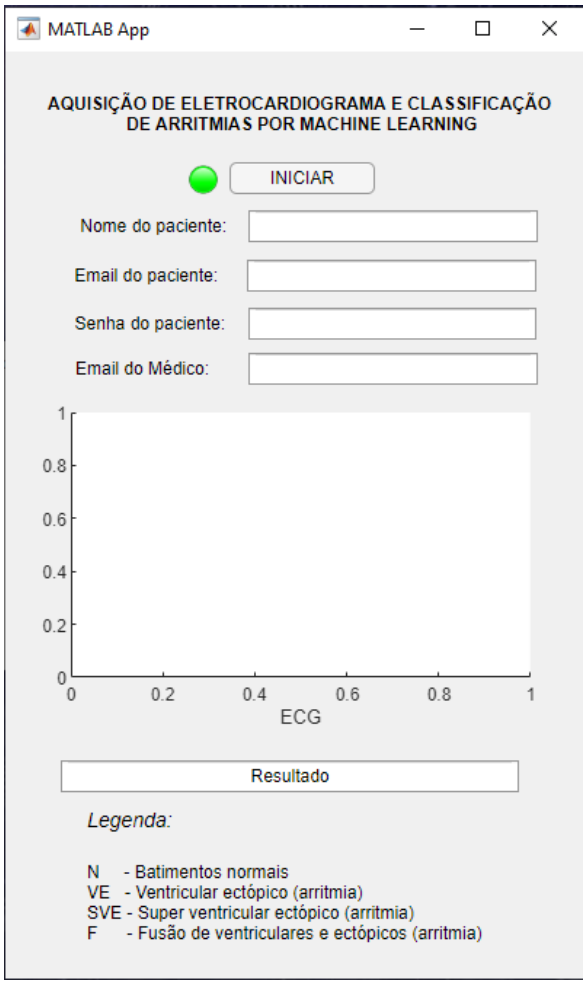
Figura 31: Alocação dos eletrodos no paciente na derivação MLII



Fonte: Autores

Para iniciar o processo, é necessário que o paciente pressione o botão de iniciar e fique o máximo em repouso possível enquanto o algoritmo faz a aquisição do sinal de ECG, pois o movimento pode alterar a medida. Para indicar ao paciente o período em que é necessário repouso, há a presença de um LED indicativo que ficará verde durante o período de aquisição, como mostrado na Figura 32, e ficará vermelho quando o processo for finalizado.

Figura 32: Indicação para estado de repouso do paciente



Fonte: Autores

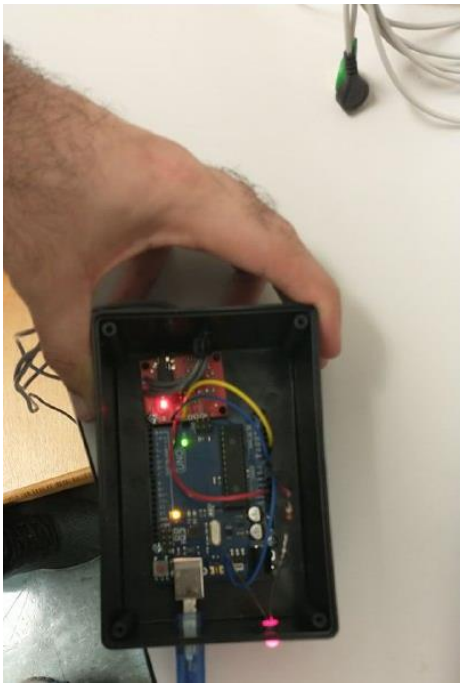
Após essa etapa, o resultado será apresentado na tela e o e-mail será enviado para o profissional de saúde definido.

6 RESULTADOS

Este capítulo tem como objetivo apresentar fotos do projeto final, bem como fotos da aplicação e do e-mail enviado ao profissional da saúde.

A Figura 33 apresenta a imagem do protótipo final. É possível observar os dois principais componentes, o Arduino Uno e o AD8232. Os cabos para os eletrodos acima e o cabo para conexão USB, juntamente com o LED indicativo de alimentação abaixo:

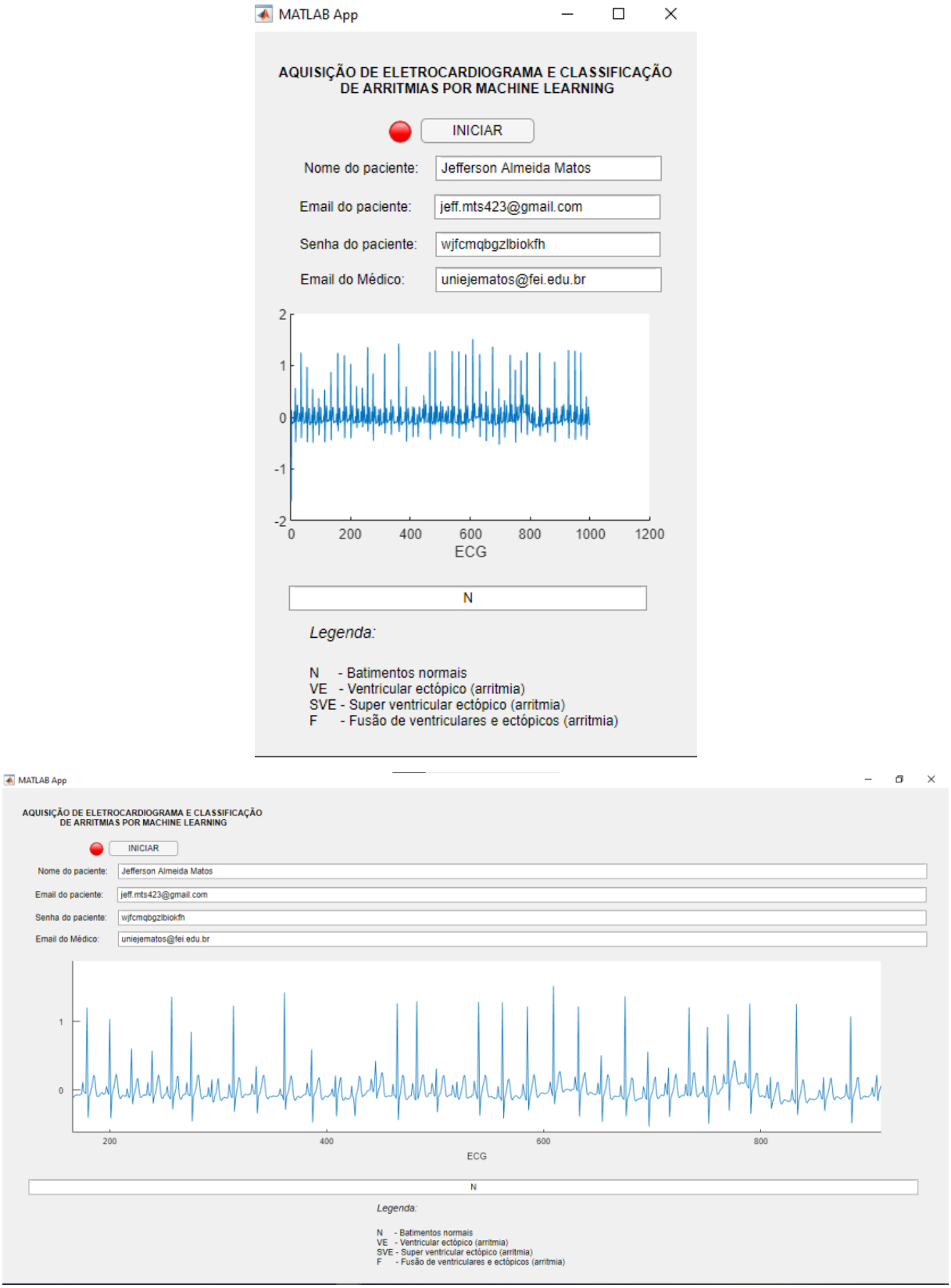
Figura 33: Dispositivo integrado



Fonte: Autores

A seguir, a Figura 34 apresenta a tela da interface de usuário após a realização de um autoexame. Pode-se ver a presença do sinal, juntamente com o ECG plotado e o diagnóstico realizado.

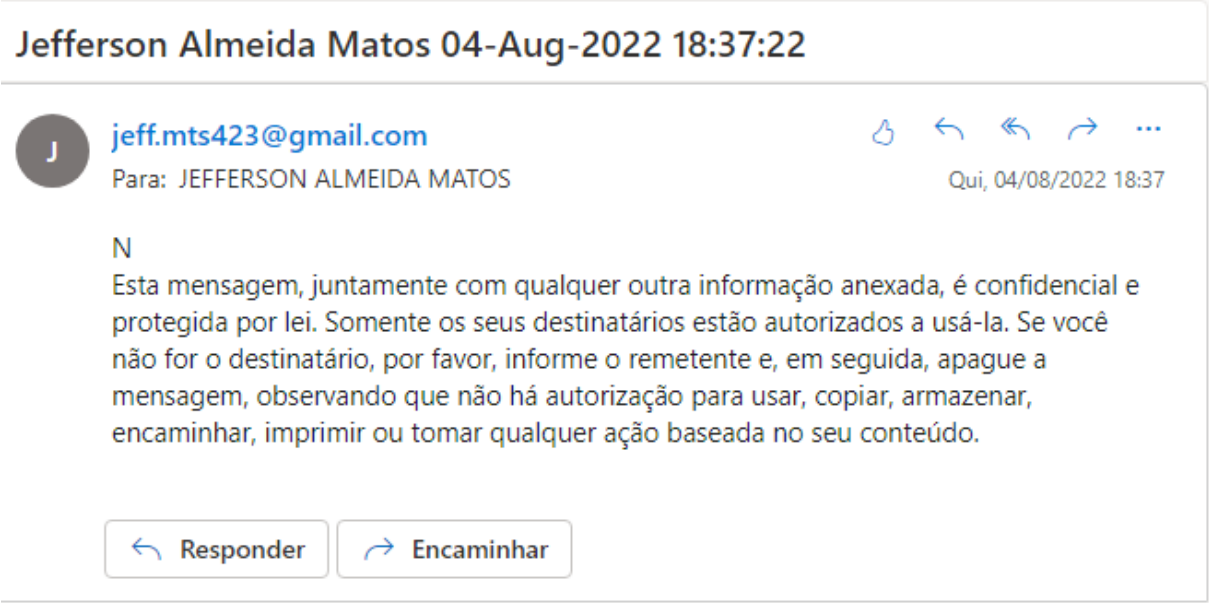
Figura 34: Interface de usuário após realizar o autoexame



Fonte: Autores

A Figura 35 apresenta o e-mail recebido, com o nome do paciente e o horário do autoexame no assunto do e-mail, e o diagnóstico na mensagem do e-mail.

Figura 35: Exemplo de e-mail recebido pelo profissional da saúde



Fonte: Autores

7 CONCLUSÕES

Este projeto propõe um dispositivo de aquisição de sinais de eletrocardiograma e classificação de arritmias através do treinamento do sistema por *machine learning*. O diagnóstico final é enviado através de um e-mail, possibilitando a monitoração remota por um profissional de saúde. O processamento de sinais nos remete ao estudo dos filtros para os sinais cardíacos, onde uma grande seletividade é requerida.

À primeira instância, a necessidade de um dispositivo autônomo, endossa ao critério de estudo para elaboração desta monografia. Contudo, seguindo as metodologias apresentadas, depara-se com a dificuldade de implantação em circuito embarcado para aplicação vestível.

O fato da dificuldade de implantação em microcontrolador, não impossibilita de analisar os métodos de inteligência artificial, bem como o desenvolvimento do dispositivo capaz de adquirir o sinal de ECG e posteriormente realizar sua classificação patológica, a fim de enviá-la às pessoas credenciadas, capazes de antecipar uma causa mais grave ao qual pode levar o paciente à óbito.

Os recursos utilizados, no que competem à ferramenta MATLAB, demonstram-se em demasia, a eficácia para a estruturação do banco de dados, a aplicação para inteligência artificial e o desenvolvimento de interface para o usuário. Todavia, tratando-se de um sinal analógico, isto é, com valores variáveis em um espaço de tempo, encontra-se a objeção de uma escrita quase que contínua, com pontos estritamente demarcados em um vetor, devido a própria limitação de processamento do programa.

Certamente, a falta de alguns pontos pode dificultar a interpretação da inteligência artificial. O programa realiza uma tendência gráfica, cujos valores muitas vezes não representam a idealidade do sinal, por isso, o diagnóstico final é àquele que apresenta a maior dentre as 25 amostras de ciclo cardíaco.

De modo geral, o processo de desenvolvimento do produto é satisfatório, atendendo os objetivos propostos neste projeto. Espera-se que a metodologia abra novas perspectivas de projeto inerentes à engenharia, em específico, o seguimento da biomédica.

8 PERSPECTIVAS FUTURAS E SUGESTÕES DE MELHORIAS

Como perspectivas, pondera-se a necessidade de melhorar a aquisição do sinal ECG, em uma análise mais profunda nos tipos de filtros existentes, bem como métodos para aquisição contínua em software.

Em adição, é sugerido a diminuição ou eliminação do *overfitting* existente, incluindo dados de outros bancos de dados, técnicas mais avançadas de *deep learning* e outra metodologia para divisão do banco.

A criação de um dispositivo vestível que opere independente do processamento de um computador é preferível para a compactação do projeto

Outra sugestão é a implementação de um banco de dados para o armazenamento dos dados clínicos do paciente para uma análise mais precisa do profissional da saúde.

REFERÊNCIAS

ADDAN, D. Unibrasil 2019. **UFPR**, 2019. Disponível em: <https://www.inf.ufpr.br/dagoncalves/IA07.pdf>. Acesso em: 01 ago. 2022.

ARAUJO, R. Aprendizado por Ensembles. **ricardomatsumura**, 2020. Disponível em: <https://ricardomatsumura.medium.com/aprendizado-por-ensembles-7cf80117446>. Acesso em: 05 jun. 2022.

CARDOZO, R. A. **Sistema de Apoio a Diagnóstico de Arritmias Cardíacas**. Dissertação (Mestrado Engenharia Elétrica) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, p. 124. 2018.

CARMO, R. L. Ciclo cardíaco, 2021. Disponível em: <https://www.kenhub.com/pt/library/anatomia/ciclo-cardiaco>. Acesso em: 15 outubro 2021.

CHINNAMGARI, S. K. R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R3.5. Birmingham: Packt, 2019.

COELHO, Í. FlipeFlop. **FlipeFlop**, 2020. Disponível em: <https://www.filipeflop.com/blog/qual-modulo-wifi-esp32-e-ideal-para-meu-projeto/>. Acesso em: 16 Abril 2022.

DANTAS, H. A. D. O. Sistema Cardiovascular. **ULBRA**, 2011. Disponível em: <https://ulbra-to.br/morfologia/2011/08/17/Sistema-Circulatorio>. Acesso em: 14 Outubro 2021.

DARAYA, V. Brasileiros criam monitor cardíaco portátil inteligente, 2013. Disponível em: <https://exame.com/ciencia/brasileiros-criam-monitor-cardiaco-portatil-inteligente/>. Acesso em: 02 Novembro 2021.

DE ALBUQUERQUE, V. H. C. et al. Robust automated cardiac arrhythmia detection in ECG beat signals. **Neural Computing and Application**, p. 1-15, 2018.

DE SOUSA E SILVA, A. F. **Deteção de Extra-Sístoles Ventriculares**. Dissertação (Mestrado Engenharia Biomédica) - Faculdade de Engenharia da Universidade do Porto. Porto, p. 112. 2012.

DUTRA, A. F. et al. Enfermagem em Cardiologia Intervencionista. [S.l.]: Editora dos Editores, 2009. Disponível em: <https://editoradoseditores.com.br/wp-content/uploads/2019/01/Cap.01-HCOR-ECI.pdf>. Acesso em: 15 Novembro 2021.

ESPINDOLA, B. M. **Análise do Sinal de Eletrocardiograma para Detecção de Cardiopatias**. Trabalho de Conclusão de Curso (Graduação Engenharia de Telecomunicação) - Instituto Federal de Santa Catarina. São José, p. 30. 2018.

FELDMAN, J.; GOLDWASSER, G. P. Eletrocardiograma: recomendações para a sua interpretação. **SOCERJ**, v. 14, p. 256, 2004.

GOLDBERGER, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation, v. 101(23):e215-e220, 2000. Disponível em: <http://circ.ahajournals.org/content/101/23/e215.full>. Acesso em: 27 Novembro 2021.

JABULAS, M. Hoje em Dia. **Observatório de Evidências Científicas Covid-19**, 2021. Disponível em: <https://www.hojeemdia.com.br/plural/pandemia-acelera-venda-de-rel%C3%B3gios-e-pulseiras-inteligentes-no-brasil-1.848843>. Acesso em: 18 janeiro 2022.

KUILA, S.; DHANDA, N.; JOARDAN, S. Machine Learning Classification and Feature. **International Journal of Recent Technology and Engineering (IJRTE)**, 16 Junho 2020. p.6598-6605.

LAPA, E. A frequência cardíaca normal fica entre 60 e 100 bpm, certo? Errado! **CardioPapers**, 2016. Disponível em: <https://cardiopapers.com.br/a-frequencia-cardiaca-normal-fica-entre-60-e-100-bpm-certo-errado/>. Acesso em: 27 Novembro 2021.

LIU, X. et al. ECG Arrhythmia Classification using High Order Spectrum and 2D Graph Fourier Transform. **Applied Science**, 09 jul. 2020. 23.

MARGATO, R. Arritmia cardíaca, 2021. Disponível em: <https://www.saudebemestar.pt/pt/medicina/cardiologia/arritmia-cardiaca/>. Acesso em: 20 outubro 2021.

MARY, W. S. et al. Real Time Arrhythmia Diagnostic System. **International Journal of Research Publications and Reviews**, Coimbatore, 2021. pp.1919-194.

MITCHELL , L. B. Bloqueio de Ramo. **msdmanuals**, [s.d.]. Disponível em: <https://www.msdmanuals.com/pt-br/casa/dist%C3%BArbios-do-cora%C3%A7%C3%A3o-e-dos-vasos-sangu%C3%ADneos/arritmias-card%C3%ADacas/bloqueio-de-ramo#:~:text=O%20bloqueio%20de%20ramo%20%C3%A9,esquerdo%20do%20feixe%20de%20His.>. Acesso em: 21 mar. 2022.

MOODY, G. B. MIT-BIH Arrhythmia Database Directory. **Physionet**, 1997. Disponível em: <archive.physionet.org/physiobank/database/html/mitdbdir/records.htm>. Acesso em: 03 ago. 2022.

MOODY, G. B.; MARK, R. G. The impact of the MIT-BIH Arrhythmia Database. **IEEE Eng in Med and Biol.**, v. 20(3):45-50, 2001. Disponível em: <https://physionet.org/content/mitdb/1.0.0/>. Acesso em: 29 Novembro 2021.

NIKAN, S.; GWADRY-SRIDHAR, F.; BAUER, M. Pattern Recognition Application in ECG Arrhythmia Classification. **In Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)**, 2017. pp.48-56. doi: 10.5220/0006116300480056.

OPAS. Doenças cardiovasculares, [s.d.]. Disponível em: <https://www.paho.org/pt/topicos/doencas-cardiovasculares>. Acesso em: 01 Novembro 2021.

PHYSIOBANK ATM. **Physionet**. Disponível em: <https://archive.physionet.org/cgi-bin/atm/ATM>. Acesso em: 01 fev. 2022.

SILVA, A. C. Anatomia e Fisiologia. Sistema cardiovascular. **Capacitação parada Crdiorespiratória Adulto**, [s.d]. Disponível em: <http://www2.eerp.usp.br/Nepien/PCR/car_conducao.html>. Acesso em: 22 Novembro 2021.

SILVA, A. L. Você sabe quando indicar maeca-passo provisório no IAM? **CardioSite**, 2018. Disponível em: <https://www.cardiosite.com.br/voce-sabe-quando-indicar-marca-passo-provisorio-no-iam/>. Acesso em: 01 Novembro 2021.

SINGH, V.; TEWARY, S. Arrhythmia Detection - A Machine Learning based. **5th International Conference for Convergence in Technology (I2CT)**, Pune, 31 Março 2019. pp.1-5.

SMART LOOPER. **CARDIOWEB**, 2021. Disponível em: <https://www.ecgweb.com.br/smart-looper/>. Acesso em: 04 Novembro 2021.

SOBRAC. Campanha coração na batida certa: Brasill em ação pela prevenção e tratamento das arritmias cardíacas e mortes súbitas, 2019. Disponível em: <https://sobrac.org/home/campanha-coracao-na-batida-certa-brasil-em-acao-pela-prevencao-e-tratamento-das-arritmias-cardiacas-e-morte-subia/>. Acesso em: 29 Outubro 2021.

WFDB Toolbox for MATLAB and Octave. **Physionet**, 2017. Disponível em: <https://archive.physionet.org/physiotools/matlab/wfdb-app-matlab/>. Acesso em: 01 Agosto 2021.

APÊNDICE A – CÓDIGO DESENVOLVIDO PARA PREPARAÇÃO DO BANCO DE DADOS

```
%% BOAS PRÁTICAS
% clear all; clc; close all;
tic
%% DIVIDIR AMOSTRAS QUE SERÃO UTILIZADAS PARA TESTE E TREINAMENTO
samplesTrain = [101, 111, 112, 121, 122, 123, 202, 210, 214, 219, 221, 222, 228, 231, 234,...
            106, 108, 109, 114, 115, 116, 118, 119, 124, 201, 203, 205, 207, 208, 215, 220, 223,
230, 232, 233];
samplesTest = [212, 113, 200, 100, 103, 117, 209, 105, 213];

%% EXTRAIR TABELA DE SINAIS COM SUA CLASSIFICAÇÃO (Segmentação dos
batimentos cardíacos)
k=1;
for record = 1:length(samplesTrain) %22 é o tamanho da samplesTrain e samplesTest
   [signalTrain,FsTrain,tmTrain]=rdsamp(num2str(samplesTrain(record)),[],[]);        %Alocar
vetor com sinal em mV
   %Tabela nº da amostra 'ann' e classificação 'annytype'
   [annTrain, anntypeTrain] = rdann(num2str(samplesTrain(record)), 'atr', [],[],[],[]);

   %% MONTA TABELA DE TREINO
   tabelaSinal=[]; % armazrna temporariamente os sinais de cada amostra de ECG para ser
alocado na tabelaDesejada
   for i = 1:length(annTrain)-1
      if annTrain (i) > 150
         amostraInicio = annTrain (i)-149;
         amostraFim = annTrain (i)+150;

         j=1;
         for l = amostraInicio : amostraFim
            tabelaSinal(1, j) = signalTrain(l,1);    %Aloca em cada linha da matriz tabelaSinal os
sinais classificados
            j=j+1;
```

```
            end
            tableTrain{k, 1} = tabelaSinal;
            tableTrain{k, 2} = anntypeTrain(i);   %Classificação
            k=k+1;
            %amostraInicio = amostraFim+1;
        end
    end

end

%% EXTRAIR TABELA DE SINAIS COM SUA CLASSIFICAÇÃO PARA TREINO
k=1;
for record = 1:length(samplesTest)
    [signalTest,FsTest,tmTest]=rdsamp(num2str(samplesTest(record)),[],[]);
    %Tabela nº da amostra 'ann' e classificação 'annytype'
    [annTest, anntypeTest]   = rdann(num2str(samplesTest(record)), 'atr', [],[],[],[]);

    %% MONTA TABELA DE TESTE
    tabelaSinal=[]; % armazrna temporariamente os sinais de cada amostra de ECG para ser alocado na tabelaDesejada
    for i = 1:length(annTest)-1
        if annTest (i) > 150
            amostraInicio = annTest (i)-149;
            amostraFim = annTest (i)+150;

            j=1;
            for l = amostraInicio : amostraFim
                tabelaSinal(1, j) = signalTest(l,1);    %Aloca em cada linha da matriz tabelaSinal os sinais classificados
                j=j+1;
            end
            tableTest{k, 1} = tabelaSinal;
            tableTest{k, 2} = anntypeTest(i);   %Classificação
```

```
        k=k+1;

      end
    end
end

%% FILTRAR DADOS TREINO APENAS CLASSIFICAÇÕES QUE SE ENQUADRAM DENTRO DAS PRINCIPAIS ARRITIMIAS (AAMI)
logicaFiltro = cellfun(@(x) isequal(x,'N')|| isequal(x,'L')|| isequal(x,'R')||...
                    isequal(x,'e')|| isequal(x,'j')|| isequal(x,'A')||...
                    isequal(x,'a')|| isequal(x,'J')|| isequal(x,'S')||...
                    isequal(x,'V')|| isequal(x,'E')|| isequal(x,'F')...
                    , tableTrain(:,2));
tableTrainFiltered = tableTrain(logicaFiltro,:);

%% FILTRAR DADOS TESTE APENAS CLASSIFICAÇÕES QUE SE ENQUADRAM DENTRO DAS PRINCIPAIS ARRITIMIAS (AAMI)
logicaFiltro = {};
logicaFiltro = cellfun(@(x) isequal(x,'N')|| isequal(x,'L')|| isequal(x,'R')||...
                    isequal(x,'e')|| isequal(x,'j')|| isequal(x,'A')||...
                    isequal(x,'a')|| isequal(x,'J')|| isequal(x,'S')||...
                    isequal(x,'V')|| isequal(x,'E') || isequal(x,'F')...
                    , tableTest(:,2));
tableTestFiltered = tableTest(logicaFiltro,:);


%% SUBSTITUIR CLASSIFICAÇÃO BD PELA CLASSIFICAÇÃO AAMI TRAIN
labelTrain = [];
for i=1:length(tableTrainFiltered)
   %sampleSigTrain(i,:)  =  tableTrainFiltered{i,1};          %VARIÁVEL RECEBE AS AMOSTRAS DE ECG
```

```
    if tableTrainFiltered{i,2} == 'N' || tableTrainFiltered{i,2} == 'L' || tableTrainFiltered{i,2} ==
'R' || tableTrainFiltered{i,2} == 'e' || tableTrainFiltered{i,2} == 'j'
        tableTrainAAMI{i,1} = 'N';
        tableTrainAAMI{i,2} = tableTrainFiltered{i,1};
    elseif tableTrainFiltered{i,2} == 'A' || tableTrainFiltered{i,2} == 'a'|| tableTrainFiltered{i,2}
== 'J' || tableTrainFiltered{i,2} == 'S'
        tableTrainAAMI{i,1} = 'SVE';
        tableTrainAAMI{i,2} = tableTrainFiltered{i,1};
    elseif tableTrainFiltered{i,2} == 'V' || tableTrainFiltered{i,2} == 'E'
        tableTrainAAMI{i,1} = 'VE';
        tableTrainAAMI{i,2} = tableTrainFiltered{i,1};
    else
        tableTrainAAMI{i,1} = 'F';
        tableTrainAAMI{i,2} = tableTrainFiltered{i,1};
    end
end
%% SUBSTITUIR CLASSIFICAÇÃO BD PELA CLASSIFICAÇÃO AAMI TEST
for i=1:length(tableTestFiltered)

    if tableTestFiltered{i,2} == 'N' || tableTestFiltered{i,2} == 'L' || tableTestFiltered{i,2} == 'R'
|| tableTestFiltered{i,2} == 'e' || tableTestFiltered{i,2} == 'j'
        tableTestAAMI{i,1} = 'N';
        tableTestAAMI{i,2} = tableTestFiltered{i,1};
    elseif tableTestFiltered{i,2} == 'A' || tableTestFiltered{i,2} == 'a'|| tableTestFiltered{i,2} ==
'J' || tableTestFiltered{i,2} == 'S'
        tableTestAAMI{i,1} = 'SVE';
        tableTestAAMI{i,2} = tableTestFiltered{i,1};
    elseif tableTestFiltered{i,2} == 'V' || tableTestFiltered{i,2} == 'E'
        tableTestAAMI{i,1} = 'VE';
        tableTestAAMI{i,2} = tableTestFiltered{i,1};
     else
         tableTestAAMI{i,1} = 'F';
         tableTestAAMI{i,2} = tableTestFiltered{i,1};
```

```
    end
end
%% QUANTIDADE DE AMOSTRAS POR CLASSIFICAÇÃO TABELA DE TREINO
aux = unique(tableTrainAAMI(:,1));
for i=1:length(aux)
    A=[];
    A = cellfun(@(x) isequal(x,aux{i}), tableTrainAAMI(:,1));
    aux{i,2} = numel(A(A>0));
end

%% DEFINIR A QUANTIDADE DE AMOSTRAS DE N PEGANDO O VALOR DA TERCEIRA MAIOR QUANTIDADE
tamanho=maxk(cell2mat(aux(:,2)),3);
tamanho=tamanho(3);
%% EMBARALHANDO AS AMOSTRAS PARA DIMINUIR O TAMANHO DELAS.
[r c] = size(tableTrainAAMI);
shuffledRow = randperm(r);
tableTrainAAMI = tableTrainAAMI(shuffledRow, :);

 %% TREINO: DIMINUIR QUANTIDADE DE AMOSTRAS CLASSIFICADAS COM NORMAIS PARA A QTD. ARMAAZENADA EM "tamanho"
logicaFiltro = cellfun(@(x) isequal(x,'N') , tableTrainAAMI(:,1));
tableN = tableTrainAAMI(logicaFiltro,:);
logicaFiltro = cellfun(@(x) isequal(x,'SVE') , tableTrainAAMI(:,1));
tableSVE = tableTrainAAMI(logicaFiltro,:);
logicaFiltro = cellfun(@(x) isequal(x,'VE') , tableTrainAAMI(:,1));
tableVE = tableTrainAAMI(logicaFiltro,:);
logicaFiltro = cellfun(@(x) isequal(x,'F') , tableTrainAAMI(:,1));
tableF = tableTrainAAMI(logicaFiltro,:);

%% DIMINUINDO O TAMANHO DAS AMOSTRAS
for i=1:length(aux(:,1))
    if aux{i,2} > tamanho
```

```
        if aux{i,1} == "N"
            tableN = tableN(1:tamanho,:);
         elseif aux{i,1} == "SVE"
            tableSVE = tableSVE(1:tamanho,:);
        elseif aux{i,1} == "VE"
            tableVE = tableVE(1:tamanho,:);
        else
            tableF = tableF(1:tamanho,:);
        end
    end
end

samplesConcat = [tableN(:,2); tableSVE(:,2);tableVE(:,2);tableF(:,2)];
labels = [tableN(:,1); tableSVE(:,1);tableVE(:,1);tableF(:,1)];

for i=1:length(samplesConcat)
    samplesSignal(i,:) = samplesConcat{i,1};
end
%% MANIPULAÇÃO DAS AMOSTRAS DE TESTE:
%% DETERMINAÇÃO O TAMANHO DAS AMOSTRAS DE TESTE.
sizeF = round(0.2*438);
sizeOthers = round(0.2*tamanho);
%% EMBARALHAR AS AMOSTRAS DE TESTE.
[r c] = size(tableTestAAMI);
shuffledRow = randperm(r);
tableTestAAMI = tableTestAAMI(shuffledRow, :);
%% TESTE: DIMINUIR QUANTIDADE DE AMOSTRAS CLASSIFICADAS COM NORMAIS PARA A QTD. ARMAAZENADA EM "tamanho"
logicaFiltro = [];
logicaFiltro = cellfun(@(x) isequal(x,'N') , tableTestAAMI(:,1));
tableN = tableTestAAMI(logicaFiltro,:);
logicaFiltro = cellfun(@(x) isequal(x,'SVE') , tableTestAAMI(:,1));
tableSVE = tableTestAAMI(logicaFiltro,:);
```

```
logicaFiltro = cellfun(@(x) isequal(x,'VE') , tableTestAAMI(:,1));
tableVE = tableTestAAMI(logicaFiltro,:);
logicaFiltro = cellfun(@(x) isequal(x,'F') , tableTestAAMI(:,1));
tableF = tableTestAAMI(logicaFiltro,:);

%% DIMINUINDO O TAMANHO DAS AMOSTRAS
for i=1:length(aux(:,1))
    if aux{i,1} == "N"
        tableN = tableN(1:sizeOthers,:);
     elseif aux{i,1} == "SVE"
        tableSVE = tableSVE(1:sizeOthers,:);
    elseif aux{i,1} == "VE"
        tableVE = tableVE(1:sizeOthers,:);
    else
        tableF = tableF(1:sizeF,:);
    end
end

samplesConcatTest = [tableN(:,2); tableSVE(:,2);tableVE(:,2);tableF(:,2)];
labelsTest = [tableN(:,1); tableSVE(:,1);tableVE(:,1);tableF(:,1)];

for i=1:length(samplesConcatTest)
    samplesSignalTest(i,:) = samplesConcatTest{i,1};
end
```

APÊNDICE B – CÓDIGO PARA CÁLCULO DA ACURÁCIA DE TESTE APÓS EXPORTAÇÃO DOS MODELOS

```
p = [];
p = trainedModelSVM.predictFcn(samplesSignalTest); % trainedModelSVM nome dado ao modelo matemático exportado
count = 0;
for i=1:length(p)
   if isequal(p{i},labelsTest{i})
      count = count + 1;
   end
end
acuraciaTestSVM = count*100/length(p)
figure
confusionchart(labelsTest,p)

p = [];
p = trainedModelENSEMBLE.predictFcn(samplesSignalTest); % trainedModelENSEMBLE nome dado ao modelo matemático exportado

count = 0;
for i=1:length(p)
   if isequal(p{i},labelsTest{i})
      count = count + 1;
   end
end
acuraciaTestENSEMBLE = count*100/length(p)
figure
confusionchart(labelsTest,p)
```

APÊNDICE C – CÓDIGO PARA INTERFACE DE USUÁRIO

```
classdef app < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                    matlab.ui.Figure
        INICIARButton               matlab.ui.control.Button
        ArritmiasLabel              matlab.ui.control.Label
        UIAxes                      matlab.ui.control.UIAxes
        ArritmiasLabel_2            matlab.ui.control.Label
        Resultado                   matlab.ui.control.EditField
        Label                       matlab.ui.control.Label
        EmaildopacienteEditFieldLabel  matlab.ui.control.Label
        EmailPaciente               matlab.ui.control.EditField
        SenhadopacienteEditFieldLabel  matlab.ui.control.Label
        SenhaPaciente               matlab.ui.control.EditField
        EmaildoMdicoEditFieldLabel     matlab.ui.control.Label
        EmailMedico                 matlab.ui.control.EditField
        NomedopacienteEditFieldLabel   matlab.ui.control.Label
        NomePaciente                matlab.ui.control.EditField
        Lamp                        matlab.ui.control.Lamp
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: INICIARButton
        function INICIARButtonPushed(app, event)
            app.Lamp.Color = 'g'
            a = arduino();
            x = 0;
```

```
tic
for k=1:1000
    b = readVoltage(a,'A0');
    x = [x,b];

end
app.Lamp.Color = 'r';

%% EXCLUIR NÍVEL MÉDIO
valorMedio = mean(x);
xNovo = x-valorMedio;

plot(app.UIAxes,xNovo);

%% VARREDURA DO SINAL PARA IDENTIFICAR PICOS
picos=[];
j=1;
k=1;
for i = 40:40:length(xNovo)
    pontoMaximo = max(xNovo(j:i));
    pontoMaximoIndice = find(xNovo(j:i) == pontoMaximo, 1)+40*(k-1);
    picos(k) = pontoMaximoIndice;
    k=k+1;
    j=j+40;
end

%% MONTAR TABELA DE DADOS, CADA LINHA UM COMPLEXO PQRST
tabelaSinal=[];
for i = 1:length(picos)
    amostraInicio = picos(i)-9;
    amostraFim = picos(i)+10;

    j=1;
```

```
            for l = amostraInicio : amostraFim
                tabelaSinal(i, j) = xNovo(1,l);
                j=j+1;
end
end
%% ADEQUAÇÃO DA TABELASINAL PARA 300 PONTOS (VALOR REF DO ALGORITMO)
n=4;
tabelaFinal = tabelaSinal;
for i = 1:n
tabelaSinalZeros = zeros(size(tabelaFinal,1), size(tabelaFinal,2)*2);
tabelaSinalZeros(1:end, 1:2:end) = tabelaFinal;
tabelaSinalZeros(1:end, 2:2:end-2) = (tabelaSinalZeros(1:end, 1:2:end-3)+tabelaSinalZeros(1:end, 3:2:end-1))/2;
tabelaFinal = tabelaSinalZeros(:,1:end-1);
end
amostrasSinal = tabelaFinal(:,1:300);
%% APLICAÇÃO MACHINE LEARNING
load('trainedModel.mat') %Carrega algoritmo já treinado;
resultados = trainedModel.predictFcn(amostrasSinal);
%% MÉDIA DOS RESULTADOS DAS AMOSTRAS
aux = unique(resultados(:,1));
for i=1:length(aux)
A=[];
A = cellfun(@(x) isequal(x,aux{i}), resultados(:,1));
aux{i,2} = numel(A(A>0));
end
maiorValor = maxk(cell2mat(aux(:,2)),2);
if maiorValor(1,1) ~= maiorValor(2,1)
app.Resultado.Value = aux{find(cell2mat(aux(:,2)) == maiorValor(1,1),1),1};
else
app.Resultado.Value ="NÃO IDENTIFICADO";
end
```

```
gmail = app.EmailMedico.Value;
myaddress = app.EmailPaciente.Value;
mypassword = app.SenhaPaciente.Value;
setpref("Internet","E_mail",myaddress);
setpref("Internet","SMTP_Server","smtp.gmail.com");
setpref("Internet","SMTP_Username",myaddress);
setpref("Internet","SMTP_Password",mypassword);
props = java.lang.System.getProperties;
props.setProperty("mail.smtp.auth","true");
props.setProperty("mail.smtp.socketFactory.class",...
"javax.net.ssl.SSLSocketFactory");
props.setProperty("mail.smtp.socketFactory.port","465");
dt = datestr(now);
assunto = app.NomePaciente.Value + " " + dt;
mensagem = app.Resultado.Value
sendmail(gmail,assunto,mensagem);

end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 397 637];
app.UIFigure.Name = 'MATLAB App';

% Create INICIARButton
app.INICIARButton = uibutton(app.UIFigure, 'push');
```

```
app.INICIARButton.ButtonPushedFcn = createCallbackFcn(app, @INICIARButtonPushed, true);
app.INICIARButton.Position = [155 540 100 22];
app.INICIARButton.Text = 'INICIAR';

% Create ArritmiasLabel
app.ArritmiasLabel = uilabel(app.UIFigure);
app.ArritmiasLabel.Position = [57 24 279 58];
app.ArritmiasLabel.Text = {'N - Batimentos normais'; 'VE - Ventricular ectópico (arritmia)'; 'SVE - Super ventricular ectópico (arritmia)'; 'F - Fusão de ventriculares e ectópicos (arritmia)'};

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, '')
xlabel(app.UIAxes, 'ECG')
ylabel(app.UIAxes, '')
app.UIAxes.PlotBoxAspectRatio = [1.73300970873786 1 1];
app.UIAxes.Position = [18 176 359 219];

% Create ArritmiasLabel_2
app.ArritmiasLabel_2 = uilabel(app.UIFigure);
app.ArritmiasLabel_2.FontSize = 14;
app.ArritmiasLabel_2.FontAngle = 'italic';
app.ArritmiasLabel_2.Position = [57 99 73 23];
app.ArritmiasLabel_2.Text = 'Legenda:';

% Create Resultado
app.Resultado = uieditfield(app.UIFigure, 'text');
app.Resultado.HorizontalAlignment = 'center';
app.Resultado.Position = [39 129 315 22];
app.Resultado.Value = 'Resultado';

% Create Label
```

```
app.Label = uilabel(app.UIFigure);
app.Label.HorizontalAlignment = 'center';
app.Label.FontWeight = 'bold';
app.Label.Position = [14 571 378 48];
app.Label.Text = {'AQUISIÇÃO DE ELETROCARDIOGRAMA E CLASSIFICAÇÃO'; ' DE
ARRITMIAS POR MACHINE LEARNING'};

% Create EmaildopacienteEditFieldLabel
app.EmaildopacienteEditFieldLabel = uilabel(app.UIFigure);
app.EmaildopacienteEditFieldLabel.HorizontalAlignment = 'right';
app.EmaildopacienteEditFieldLabel.Position = [43 473 104 22];
app.EmaildopacienteEditFieldLabel.Text = 'Email do paciente:';

% Create EmailPaciente
app.EmailPaciente = uieditfield(app.UIFigure, 'text');
app.EmailPaciente.Position = [167 473 199 22];

% Create SenhadopacienteEditFieldLabel
app.SenhadopacienteEditFieldLabel = uilabel(app.UIFigure);
app.SenhadopacienteEditFieldLabel.HorizontalAlignment = 'right';
app.SenhadopacienteEditFieldLabel.Position = [43 440 109 22];
app.SenhadopacienteEditFieldLabel.Text = 'Senha do paciente:';

% Create SenhaPaciente
app.SenhaPaciente = uieditfield(app.UIFigure, 'text');
app.SenhaPaciente.Position = [168 440 198 22];

% Create EmaildoMdicoEditFieldLabel
app.EmaildoMdicoEditFieldLabel = uilabel(app.UIFigure);
app.EmaildoMdicoEditFieldLabel.HorizontalAlignment = 'right';
app.EmaildoMdicoEditFieldLabel.Position = [43 409 98 22];
app.EmaildoMdicoEditFieldLabel.Text = 'Email do Médico:';
```

```
% Create EmailMedico
app.EmailMedico = uieditfield(app.UIFigure, 'text');
app.EmailMedico.Position = [168 409 199 22];

% Create NomedopacienteEditFieldLabel
app.NomedopacienteEditFieldLabel = uilabel(app.UIFigure);
app.NomedopacienteEditFieldLabel.HorizontalAlignment = 'right';
app.NomedopacienteEditFieldLabel.Position = [47 507 106 22];
app.NomedopacienteEditFieldLabel.Text = 'Nome do paciente:';

% Create NomePaciente
app.NomePaciente = uieditfield(app.UIFigure, 'text');
app.NomePaciente.Position = [168 507 199 22];

% Create Lamp
app.Lamp = uilamp(app.UIFigure);
app.Lamp.Position = [127 540 20 20];
app.Lamp.Color = [1 0 0];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = app

% Create UIFigure and components
createComponents(app)
```

```
% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargout == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```