



**UNIVERSIDADE ESTADUAL DE
CAMPINAS
FACULDADE DE TECNOLOGIA**



Missão: Formar e aperfeiçoar cidadãos e prestar serviços atendendo às necessidades tecnológicas da sociedade com agilidade, dinâmica e qualidade.

Monitoramento de Batimentos Cardíacos Utilizando Linguagem Python

Alunos: Mathews Lago Joaquim e Matheus De Lima Cerqueira

Orientadora: Talía Simões dos Santos Ximenes

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Luiz Felipe Galeffi - CRB 8/10385

J57m Joaquim, Mathews Lago, 1996-
Monitoramento de batimentos cardíacos utilizando linguagem Python /
Mathews Lago Joaquim, Matheus de Lima Cerqueira. – Limeira, SP : [s.n.], 2021.

Orientador: Talia Simões dos Santos Ximenes.
Trabalho de Conclusão de Curso (graduação) – Universidade Estadual de
Campinas, Faculdade de Tecnologia.

1. Eletrocardiografia. 2. Python (Linguagem de programação de computador).
3. Análise de dados. I. Ximenes, Talia Simões dos Santos, 1980-. II. Cerqueira,
Matheus de Lima, 1994-. III. Universidade Estadual de Campinas. Faculdade de
Tecnologia. IV. Título .

Informações adicionais, complementares

Palavras-chave em inglês:

Electrocardiography

Python (Computer program language)

Data analytics

Titulação: Bacharel

Banca examinadora:

Talia Simões dos Santos Ximenes [Orientador]

Francisco José Arnold

Ivan de Oliveira

Data de entrega do trabalho definitivo: 20-12-2021

Agradecimentos

Em primeiro lugar nós agradecemos a todos os familiares que nos ajudaram e estiveram ao nosso lado durante toda a etapa da nossa graduação, sem o imprescindível apoio dos mesmos nós não conseguiríamos chegar tão longe.

Gostaríamos também de agradecer a todos nossos amigos e amigas que nos ajudaram nessa etapa tão importante de nossas vidas, que foram as pessoas que nos deram suporte e sempre estiveram presente com um ombro amigo e nos incentivando a nunca desistir, pois tudo daria certo no final.

Resumo

As doenças cardiovasculares são as doenças líderes em morte, não só no Brasil, mas ao redor do mundo como um todo. Muitos fatores contribuem para isso, como o envelhecimento da população, o aumento do sedentarismo e o aumento do estresse em função do dinamismo do mundo moderno. Dito isso, o objetivo deste trabalho é o da leitura de dados obtidos através de um eletrocardiograma e a identificação dos valores da frequência cardíaca. É através da identificação da frequência cardíaca que se torna possível a emissão de um alerta mostrando a frequência mínima, a máxima e a média dentro do período de amostragem. Sendo, assim, de grande uso para o tratamento de dados cardíacos. A linguagem de programação utilizada no trabalho, é a linguagem Python. A linguagem Python foi escolhida por sua grande gama de bibliotecas disponíveis para uso, além de ser uma das difundidas ao redor do mundo para a utilização de leitura, tratamento e interpretação de quaisquer tipos de dados, sendo assim a linguagem líder quando o assunto referido é a análise de dados e ciência de dados.

Palavras Chave: Eletrocardiograma, Python, Análise de dados, Doenças cardiovasculares, Ciência de dados.

Abstract

Cardiovascular diseases are the leading diseases in death, not only in Brazil, but around the world as a whole. With the aging of the population combined with people becoming more sedentary in conjunction with environments that cause increased stress. That said, the objective of this work is to read data obtained through an electrocardiogram and identify heart rate values. It is through the identification of the heart rate that it is possible to issue an alert showing the minimum, maximum and average frequency within the sampling period. Therefore, it is of great use for the processing of cardiac data. The programming language used at work is the Python language. The Python language was chosen for its wide range of libraries available for use, in addition to being one of the most widespread around the world for the use of reading, processing and interpreting any type of data, thus being the leading language when the subject referred to is data analysis and data science.

Keywords: Electrocardiograms, Python, Data Analytics, Cardiovascular diseases, Data Science.

Lista de Figuras

Figura 1: Orientação dos vetores das despolarizações dos átrios direito e esquerdo e da resultante (SâP).....	13
Figura 2: A Despolarização ventricular representada por 4 vetores cardíacos.....	14
Figura 3: Elementos de um Eletrocardiograma.....	15
Figura 4: Representação do batimento cardíaco.....	16
Figura 5: Medição usando Triângulo de Einthoven.....	17
Figura 6: Demonstração da tela inicial do Visual Studio Code.....	18
Figura 7: Demonstração da tela com o repositório utilizado no Github.....	20
Figura 8: Demonstração de como é o diretório com arquivos advindos do Physionet.....	22
Figura 9: Exemplificação de código para obter os dados do arquivo do Physionet.....	23
Figura 10: Exibição na tela da informação adquirida do Physionet.....	24
Figura 11: O Código presente na função divide_tempo.....	25
Figura 12: Gráfico representando a quantidade de erros que cada método apresentou com relação ao database utilizado.....	27
Figura 13: Gráfico representando estatisticamente a quantidade de erros que cada método possui.....	27
Figura 14: O Código presente na função processa_plota.....	29
Figura 15: O Código presente na função min_max.....	30
Figura 16: Exemplificação de como cada biblioteca é utilizada no programa.....	33
Figura 17: Arquivo .txt gerado com informações sobre o sample 16265.....	34
Figura 18: Janela mostrando os dados do sample 16265.....	35
Figura 19: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 16265.....	35
Figura 20: Arquivo .txt gerado com informações sobre o sample 108.....	36
Figura 21: Janela mostrando os dados do sample 108.....	37
Figura 22: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 108.....	37
Figura 23: Arquivo .txt gerado com informações sobre o sample 421.....	38
Figura 24: Janela mostrando os dados do sample 421.....	39

Figura 25: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 421.....	39
Figura 26: Arquivo .txt gerado com informações sobre o sample 422.....	40
Figura 27: Janela mostrando os dados do sample 422.....	41
Figura 28: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 422.....	41

Lista de Abreviaturas e Siglas

ECG - Eletrocardiograma

BPM - Batimentos por Minuto

mv - MilliVolts

WFDB - *WaveForm DataBase*

DCV - Doenças Cardiovasculares

DIC - Doenças Isquêmicas do Coração

OMS - Organização Mundial da Saúde

AD - Átrio Direito

AE - Átrio Esquerdo

Sumário

- 1. Introdução**
 - 1.1. Cenário Brasileiro e Mundial sobre Doenças Cardiovasculares
 - 1.2. Eletrocardiograma
- 2. Objetivos**
 - 2.1. Objetivos Gerais
 - 2.2. Objetivos Específicos
- 3. Motivação**
- 4. Fundamentação Teórica de um Eletrocardiograma**
- 5. Materiais**
 - 5.1. Software Visual Studio Code
 - 5.2. Python
 - 5.3. Physionet
 - 5.4. GitHub
 - 5.5. *MIT-BIH Arrhythmia Database*
 - 5.6. *MIT-BIH Normal Sinus Rhythm Database*
 - 5.7. *MIT-BIH Malignant Ventricular Ectopy*
- 6. Desenvolvimento**
 - 6.1. Leitura e armazenamento de Dados
 - 6.2. Análise dos Dados
 - 6.3. Demonstração dos Dados
 - 6.4. Resumo
- 7. Resultados e Discussões**
- 8. Conclusões**
- 9. Referências Bibliográficas**

1. Introdução

1.1 Cenário Brasileiro e Mundial sobre Doenças Cardiovasculares

As doenças cardiovasculares (DCV) são a principal causa de morte em homens e mulheres, no Brasil. Sendo que para indivíduos com mais de 30 anos a representatividade chega a ser em 20% dos casos. Segundo dados oficiais do ministério da saúde, no ano de 2009, tivemos 962.931 mortes de pessoas com mais de 30 anos. Destas 962.931 mortes, 95.449 mortes foram causadas por doenças isquêmicas do coração (DIC), e 97.860 mortes foram causadas pelas doenças cerebrovasculares (DCbV). Demonstrando assim o grande impacto que as doenças cardiovasculares possuem em na mortalidade de adultos no Brasil [1].

Segundo a Organização Mundial da Saúde (OMS), os casos mais comuns são de pacientes que tiveram acidente vascular cerebral e doenças cardíacas, sendo estas responsáveis por aproximadamente 31% das mortes globais. Entre essas mortes, 75% ocorrem em países de baixa e média renda e 80% são causadas por ataques cardíacos e derrames. Durante o ano de 2007, somente as DCVs causaram nas Américas 1,5 milhão de mortes [2].

1.2 Eletrocardiograma

De acordo com a Sociedade Brasileira de Cardiologia:

O eletrocardiograma(ECG) é um exame realizado, cujo objetivo é o registro da atividade elétrica cardíaca na superfície do tórax. A qualidade deste método não é obtida por outras técnicas e, sem dúvidas, é o procedimento mais utilizado para auxiliar no diagnóstico das doenças cardíacas. Este método é de simples realização, seguro, reprodutível, podendo ser usado em grandes estudos devido ao seu baixo custo. O eletrocardiograma é considerado padrão ouro para o diagnóstico não invasivo das arritmias e distúrbios de condução, além de ser muito importante nos

quadros isquêmicos coronarianos, constituindo-se em um marcador de doenças do coração. Sua sensibilidade e sua especificidade são maiores para o diagnóstico das arritmias e distúrbios de condução, do que para as alterações estruturais ou metabólicas. Existe, por outro lado, um grande número de anormalidades fisiopatológicas e estruturais que pode ser reconhecido pelo eletrocardiograma, porém, a sobreposição de alterações resultante reduz a especificidade para várias formas de doenças do coração. As alterações da repolarização ventricular são um bom exemplo disso, pois, embora sejam as modificações mais comuns e mais sensíveis, são pouco específicas.[3] (Diretriz de interpretação de eletrocardiograma de repouso, 2003, p. 23)

2. Objetivos

2.1 Objetivos Gerais

Temos por objetivo a elaboração de um programa na linguagem de programação de alto nível, chamada Python. Por meio desta linguagem, a ideia é montar um código capaz de ler e tratar os dados obtidos através de um eletrocardiograma. Após o tratamento e análise dos dados, plotar graficamente a curva relacionada ao batimento cardíaco junto com sua frequência cardíaca.

2.2 Objetivos Específicos

- Estudar a seleção de bibliotecas da linguagem Python para o tratamento dos dados que serão utilizados;
- Desenvolver uma interface *front-end* para demonstrar os gráficos obtidos e a emissão de alertas e dos dados obtidos;
- Estudar e compreender as etapas do funcionamento do batimento cardíaco;

- O entendimento de como funciona o eletrocardiograma, um exame extremamente importante para a monitoração cardíaca;
- Conseguir identificar rapidamente grandes variações de frequências cardíacas.

3. Motivação

A motivação para a realização do trabalho vem de conseguir fazer a leitura, armazenamento e tratamento de dados e depois a demonstração destes dados fisiológicos para o usuário. De modo que estes dados demonstrados possam ser obtidos de quaisquer meios, como *wearables* entre outros. No nosso caso iremos utilizar os dados de um ECG, mas o código consegue ser replicado para *wearables* e outros dispositivos de leitura de dados fisiológicos, neste caso de batimentos cardíacos. Sendo assim podendo ser amplamente utilizado, implementado e melhorado para os fins necessários.

4. Fundamentação Teórica de um Eletrocardiograma

O eletrocardiograma é um dos exames mais usados ao redor do mundo para a realização de análises sobre o funcionamento do coração. Sendo utilizado em todas as frentes de tratamento, partindo de postos de saúde e triagem, até hospitais e centros especializados. Contudo o exame ainda é totalmente dependente da análise das imagens por um profissional especializado. Um eletrocardiograma deve ser analisado por um cardiologista, clínico geral ou um médico que trabalhe com urgências, sendo que estes devem possuir a habilidade de encontrar e diagnosticar

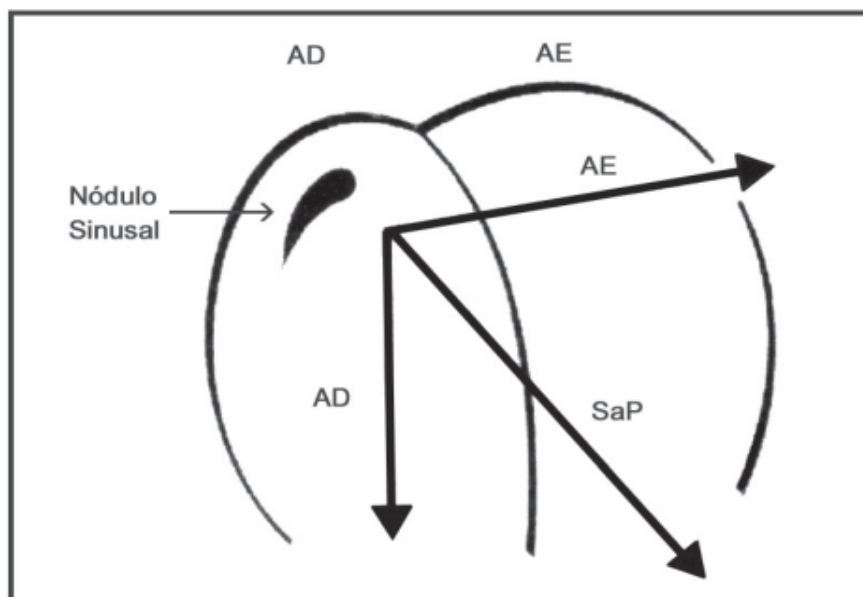
os princípios fisiopatológicos de anormalidades eletrocardiográficas encontradas no exame [3].

O ECG possui diversas vantagens, dentre elas temos que, por se tratar de um exame que demonstra inúmeros problemas no coração, não ser invasivo e possuir rápida realização (principalmente comparado ao exame holter, uma espécie de eletrocardiograma de alta duração para detecção principalmente de arritmias) e também pode ser facilmente empregado em grandes estudos [3].

A análise dedutiva é a base fundamental na interpretação de um ECG e requer o conhecimento do significado do processo de ativação do coração. Ativação ou despolarização cardíaca, em condições normais, tem origem no nódulo sinusal (nódulo de Keith-Flack), região do marca-passo cardíaco, localizado no átrio direito, sendo esta a primeira área do coração a se despolarizar. O estímulo alcança, em seqüência, o átrio esquerdo, o nódulo atrioventricular (nódulo de Aschoff-Tawara), o feixe de His e seus ramos (esquerdo e direito), a rede de Purkinje, os ventrículos e, por fim, se extingue. Durante a atividade cardíaca desencadeada pelo processo de ativação do coração, fenômenos elétricos são originados na despolarização e repolarização, podendo ser registrados pelo eletrocardiógrafo. (J. Feldman, Gerson, 2004, p. 1)

A ativação atrial inicia-se no átrio direito (AD) e se estende ao átrio esquerdo (AE), sendo representados por dois vetores do AD que se orientam para baixo e um pouco para frente, e do AE que se orientam para trás e mais para esquerda. O vetor resultante final, que determina o eixo elétrico, orienta-se para a esquerda, para baixo em paralelo ao plano frontal (Figura 1). A deflexão resultante é denominada ONDA P (J. Feldman, Gerson, 2004, p. 2).

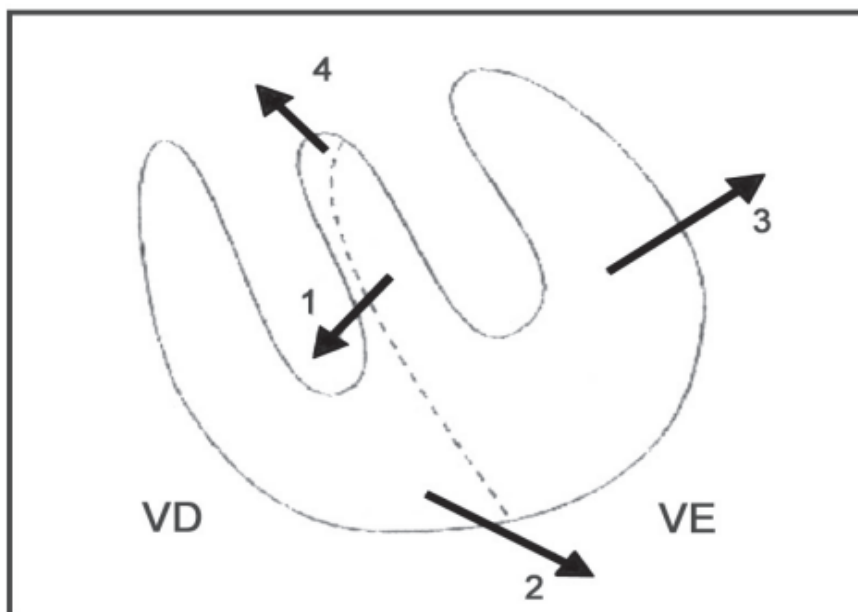
Figura 1: Orientação dos vetores das despolarizações dos átrios direito e esquerdo e da resultante (SâP)



Fonte: [J. Feldman, Gerson, 2004, p. 2]

Ativação ventricular é representada por quatro vetores cardíacos (Figura 2): vetores 1 e 2 correspondem às despolarizações das regiões do septo interventricular; vetores 3 e 4 correspondem às despolarizações das paredes dos ventrículos com amplo predomínio vetorial do ventrículo esquerdo. O vetor resultante final, que determina o eixo elétrico, orienta-se para a esquerda e algo para trás (J. Feldman, Gerson, 2004, p. 2).

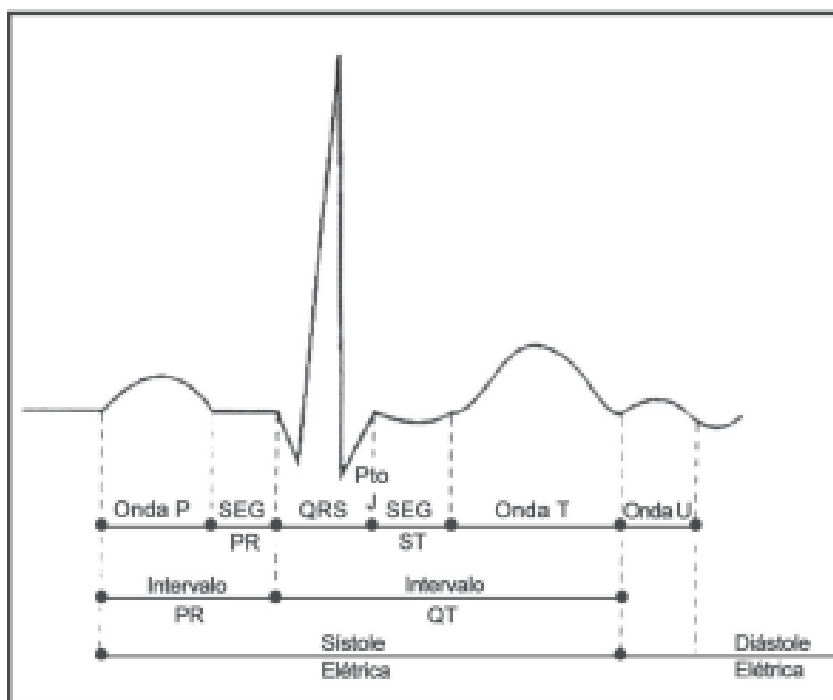
Figura 2: A Despolarização ventricular representada por 4 vetores cardíacos.



Fonte: [J. Feldman, Gerson, 2004, p. 2]

A repolarização ventricular inicia-se ao fim da ativação ventricular. Inicialmente as correntes elétricas desenvolvidas não são intensas e não causam deflexão, sendo inscrita uma linha isoelétrica. A seguir surgem potenciais mais intensos que determinam a inscrição da deflexão resultante, denominada onda T. A repolarização atrial (Onda T) não é identificada no ECG normal. (J. Feldman, Gerson, 2004, p. 2).

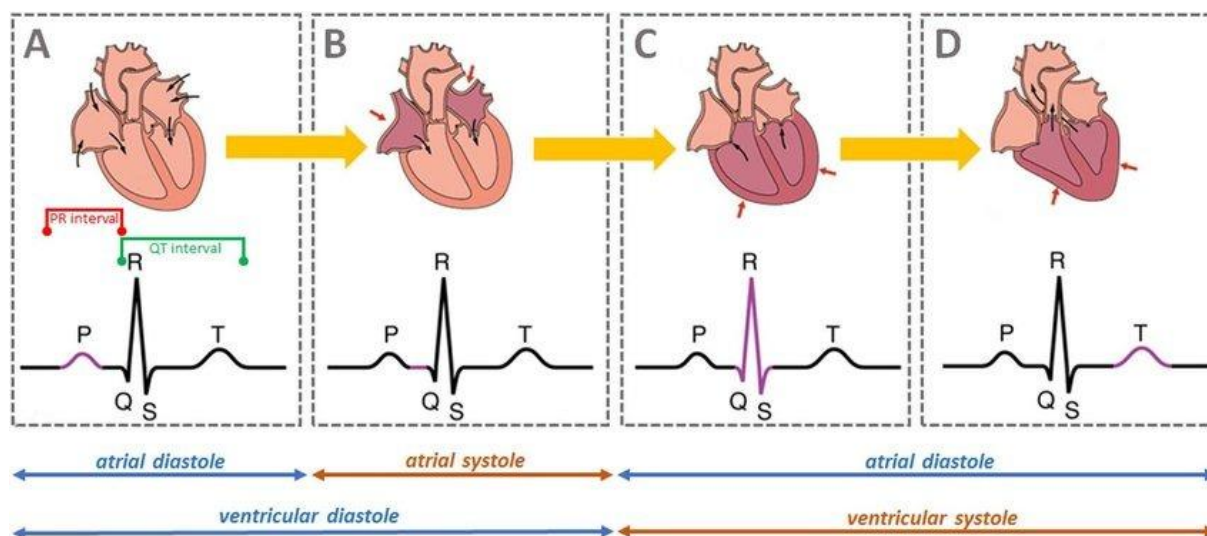
Na Figura 3 tem-se a representação de toda uma onda PQRST em um ECG.

Figura 3: Elementos de um Eletrocardiograma.

Fonte: [J. Feldman, Gerson, 2004, p. 2]

A figura 4 presente abaixo, demonstra melhor o funcionamento do fluxo sanguíneo através do coração durante o ciclo cardíaco em correspondência com o complexo QRS que é observado no ECG: Na parte A temos o enchimento atrial durante a diástole atrial, diástole é o período de relaxamento muscular ou recuperação do músculo cardíaco. Na parte B, temos a Sístole atrial e enchimento ventricular, sístole é o período de contração muscular das câmaras cardíacas. Na parte C temos a contração isovolumétrica ventricular, que é o local onde ocorre o pico R. Em que após o pico R vamos começar a diástole do ventrículo e a sístole do ventrículo. Na parte D temos a sístole ventricular e sendo assim a ejeção de sangue dos ventrículos. Todo este processo representa um batimento cardíaco.[20]

Figura 4: Representação do batimento cardíaco.



Fonte: [T. Vasavana, E.Ferraro, E.Ibrahimb, P. Dixona, J.Gorelikb, C.Williamson, 2018,p.2]

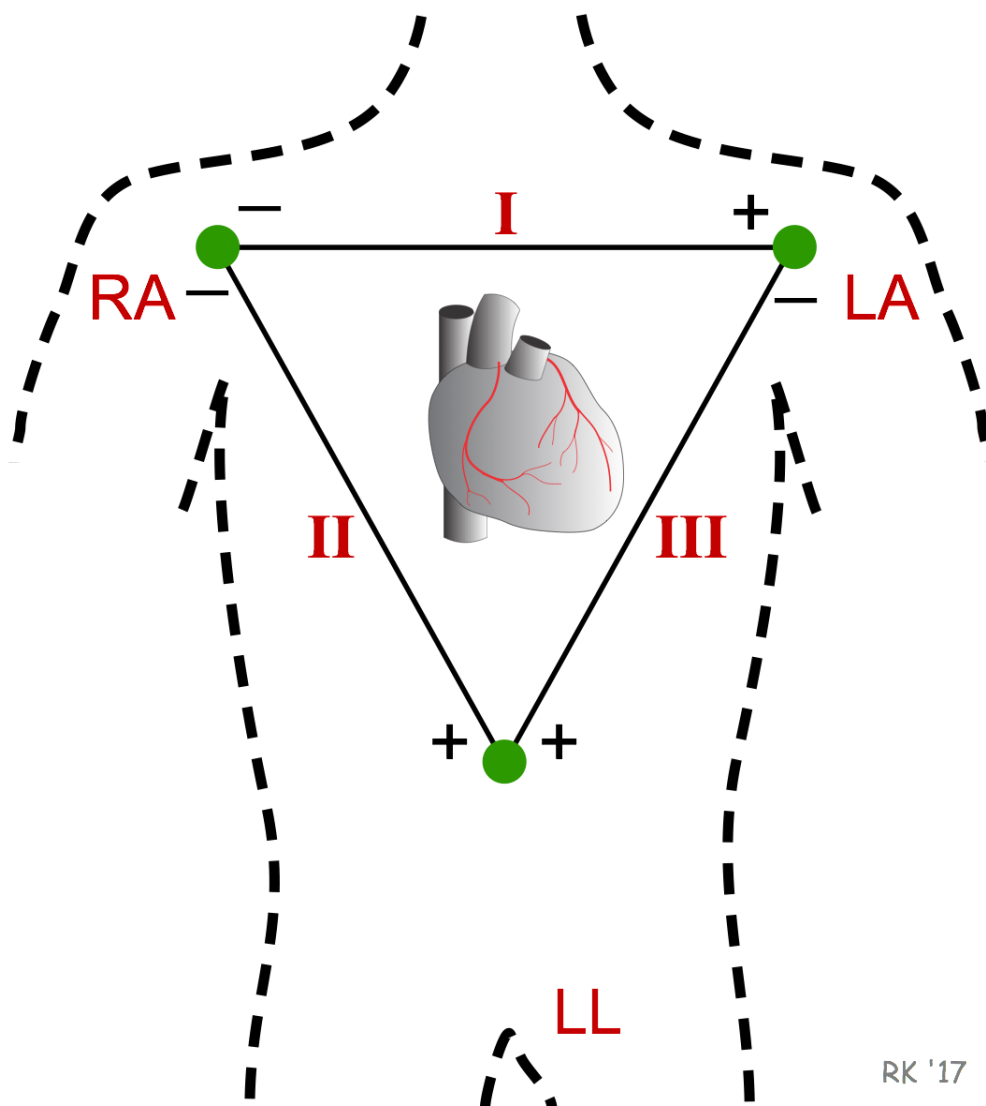
Todo este processo é o correspondente a um batimento cardíaco. Como é possível identificar o maior pico durante toda a onda do ECG, o chamado pico R. Como este pico é o de mais fácil identificação é a partir dele que iremos identificar que ocorreu um batimento cardíaco. A partir da contagem de quantos picos R tivemos no período de um minuto iremos conseguir identificar o número de batimentos por minuto.

Para a realização da medição da diferença de potencial no coração e assim identificar toda a onda temos que os registros bipolares utilizam configurações de eletrodos de membro padrão representadas na figura 5.

A derivação I tem o eletrodo positivo no braço esquerdo e o eletrodo negativo no braço direito e, portanto, mede a diferença de potencial entre os dois braços. Nesta e nas outras duas derivações de membro, um eletrodo na perna direita serve como um eletrodo de referência para fins de registro. Na configuração da derivação II, o eletrodo positivo está na perna esquerda e o eletrodo negativo está no braço direito. A derivação III tem o eletrodo positivo na perna esquerda e o eletrodo negativo no braço esquerdo. Essas três derivações de membro bipolar formam aproximadamente um triângulo equilátero (com o coração no centro) que é chamado de triângulo de Einthoven em homenagem a Willem Einthoven, que desenvolveu o

eletrocardiograma no início do século XX. Se os eletrodos do membro estão ligados à extremidade do membro (punhos e tornozelos) ou na origem do membro (ombro ou coxa) faz pouca diferença na gravação porque o membro pode simplesmente ser visto como um condutor de fio longo originado de um ponto no tronco do corpo. (Richard E. Klabunde, 2017).

Figura 5: Medição usando Triângulo de Einthoven.



Fonte: [Richard E. Klabunde]

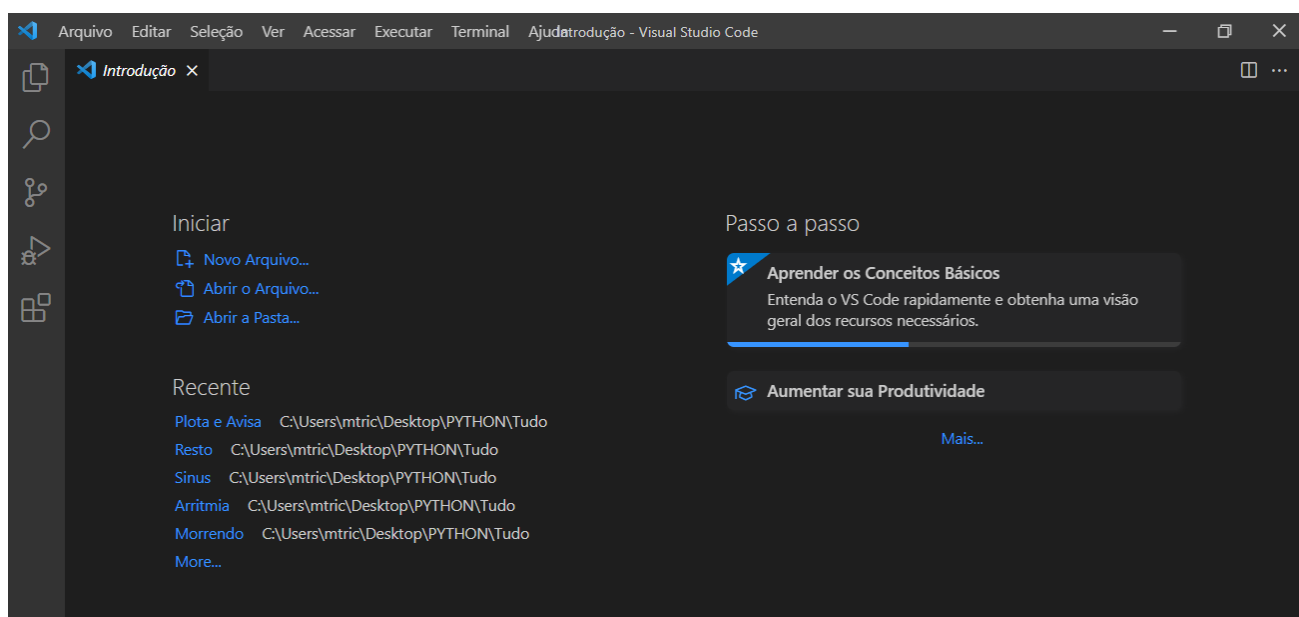
5. Materiais e Métodos

Para o desenvolvimento do projeto foram necessários alguns materiais, dentre eles o software utilizado para a programação, os *databases* utilizados e a linguagem de programação utilizada. Estes materiais estão listados a seguir.

5.1 Software Visual Studio Code

O Visual Studio Code foi o software selecionado para a realização de toda a programação e foi utilizada a linguagem Python. A Figura 4 mostra como é o ambiente de programação no software.

Figura 6: Demonstração da tela inicial do Visual Studio Code



Fonte: Autor

5.2 Python

A linguagem de alto nível que foi utilizada durante todo o projeto foi a linguagem Python. Esta linguagem foi escolhida devido à variedade de bibliotecas que possibilitam inúmeras aplicações. Sendo esta linguagem uma das mais utilizadas ao redor do mundo atualmente.

5.3 Physionet

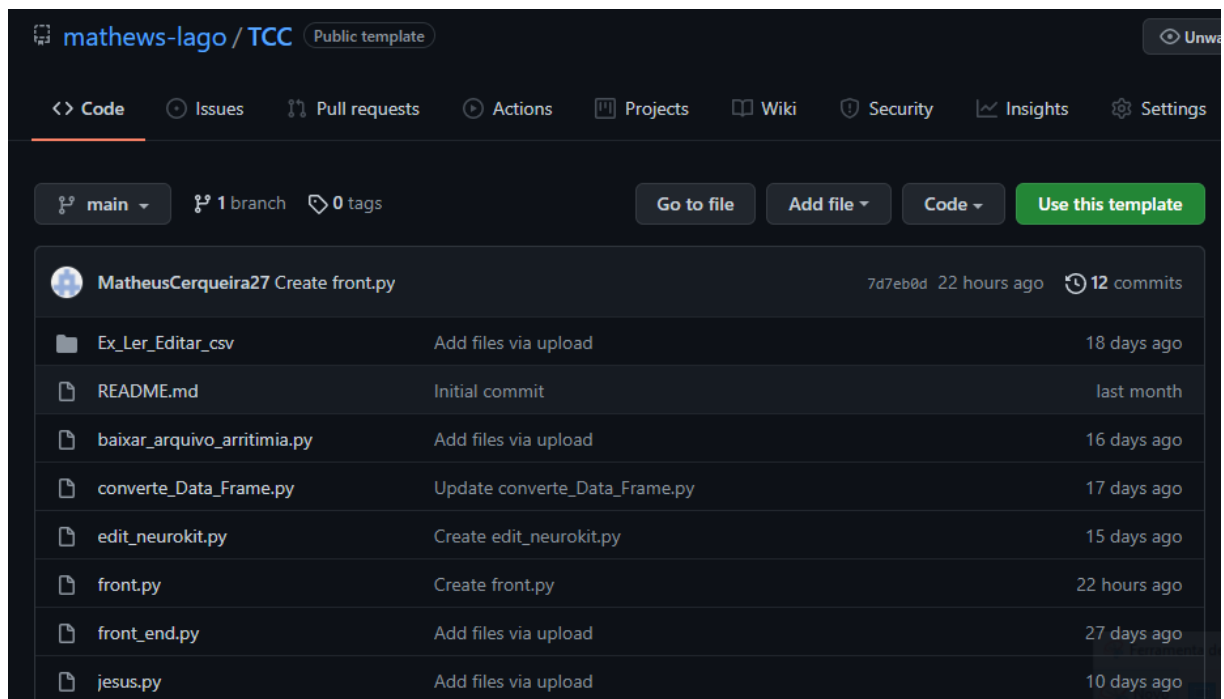
Physionet é um fórum online onde se tem a divulgação de arquivos com sinais biomédicos gravados, e onde é possível acessar códigos abertos que são utilizados para analisá-los. Ele foi criado pelos pesquisadores do Beth Israel Deaconess Medical Center de Boston, Boston University, McGill Universidade e o MIT (*Massachusetts Institute of Technology*), no dia 1 de agosto de 1999 [4].

Os dados do ECG, foram obtidos através do Physionet, que foi escolhido por possuir facilidade para manuseio, além de muitos estudos publicados que utilizam os dados com *open access*. Dessa forma, contém dados que são utilizados para estudo ao redor de todo o mundo.

5.4 GitHub

O GitHub é um site onde é possível criar diretórios públicos ou privados para o compartilhamento de arquivos. Por estarmos em ambientes de ensino à distância ele foi utilizado como um repositório em que era possível os dois membros acessassem e conseguissem manipular os arquivos. A Figura 7 mostra o ambiente do diretório que era utilizado pelos autores do trabalho de conclusão de curso.

Figura 7: Demonstração da tela com o repositório utilizado no GitHub



Fonte: Autor

5.5 MIT-BIH Arrhythmia Database

O *MIT-BIH Arrhythmia Database*, possui 48 *samples* de meia hora de registros de ECG ambulatorial. Estes *samples* foram obtidos a partir de 47 indivíduos, que estavam sendo estudados pelo Laboratório de arritmia BIH, no período entre 1975 e 1979 [17].

Estas gravações foram digitalizadas com uma frequência de 360 amostras por segundo. Com um canal de resolução de 11 bits em uma faixa de 10 mV. Além disso, dois ou mais cardiologistas realizaram anotações independente de cada registro. Quaisquer desacordos foram resolvidos para obter as anotações de referência legível pelo computador [17].

5.6 MIT-BIH Normal Sinus Rhythm Database

O *MIT-BIH Normal Sinus Rhythm Database*, possui 18 registros de ECG, que são de longo prazo. De indivíduos que foram encaminhados para o Laboratório de Arritmia do Hospital Beth Israel de Boston, que agora é conhecido como o Centro Médico Beth Israel Deaconess. Estes indivíduos não apresentaram arritmias significativas. Dentre eles tem-se 5 homens, cujas idades estão entre 26 e 45 anos, e 13 mulheres, no qual as idades variam entre 20 e 50 anos [18].

5.7 MIT-BIH Malignant Ventricular Ectopy

O *MIT-BIH Malignant Ventricular Ectopy*, possui registros de ECGs com meia hora de duração de 22 indivíduos que experimentaram episódios de taquicardia ventricular sustentada, flutter ventricular e fibrilação ventricular [19].

6. Desenvolvimento

O trabalho é dividido em 3 etapas muito importantes e distintas entre si. No primeiro momento tem-se a obtenção e pré-tratamento dos dados. Nesta etapa recebe-se os arquivos do tipo .atr, .dat, .hea e .xws para então passá-los para um formato que seja legível para o código. Em seguida, faz-se o tratamento do sinal adquirido e armazenado. Nesta etapa processa-se o sinal de forma a identificar todas as variáveis e informações consideradas importantes. Como última etapa é feita a demonstração dos dados obtidos através de gráficos e a apresentação em tela do que foi observado com um possível alerta.

6.1 Leitura e armazenamento de Dados

Primeiramente é necessário receber os dados contendo os registros feitos com ECG. Os dados utilizados estão disponíveis publicamente na Physionet. Neste trabalho utilizou-se os dados presentes em três distintos *databases* para, dessa forma, demonstrar o funcionamento do código nos três ambientes. Os *databases* selecionados foram os:

- *MIT-BIH Arrhythmia Database*
- *MIT-BIH Normal Sinus Rhythm Database*
- *MIT-BIH Malignant Ventricular Ectopy*

Os arquivos do Physionet vem separados por paciente estudado. Os dados e anotações, na maioria dos bancos de dados do PhysioBank, são armazenados no formato Waveform Database (WFDB). Seguindo este padrão, para cada paciente, os registros de forma de onda são disponibilizados em 4 tipos de arquivos. Sendo estes no formato .atr, .dat, .hea e .xws. Assim como demonstrado na Figura 8.

Figura 8: Demonstração de como é o diretório com arquivos advindos do Physionet

Nome	Data de modificação	Tipo	Tamanho
mitdbdir	16/11/2021 21:21	Pasta de arquivos	
x_mitdb	16/11/2021 21:21	Pasta de arquivos	
100.atr	29/07/1992 21:21	Arquivo ATR	5 KB
100.dat	30/07/1992 01:36	Arquivo DAT	1.905 KB
100.hea	30/07/1992 01:36	Arquivo HEA	1 KB
100.xws	12/12/1999 17:28	Arquivo XWS	1 KB

Fonte:Autor

O arquivo .hea (cabeçalho) é um arquivo de texto curto que descreve os sinais (incluindo o nome ou URL do arquivo de sinal, formato de armazenamento, número e tipo de sinais, frequência de amostragem, dados de calibração, características do

digitalizador, duração do registro e hora de início). A maioria dos registros inclui um ou mais arquivos de anotação binários (no exemplo, .atr denota um arquivo de anotação). Os registros incluem um arquivo binário .dat (sinal), que é onde contém as amostras digitalizadas de um ou mais sinais; estes arquivos podem ser muito grandes, devido à enorme quantidade de dados [6].

Como os dados estão no formato .dat, se faz necessário convertê-los para um padrão mais adequado para ser analisado. Para isso, utilizou-se a biblioteca em Python chamada *wfdb* (*Waveform Database*).

A biblioteca *wfdb* é de código aberto, pertence ao PhysioToolkit, ela é utilizada para leitura e análise de dados do PhysioBank e está disponível gratuitamente para, GNU / Linux, Mac OS / X, MS-Windows, Solaris e muitos outros sistemas operacionais populares. Neste trabalho, a biblioteca foi importada para o Python operando no sistema operacional Windows [6].

Para a leitura dos arquivos utilizou-se a função *rdsamp*. Por meio desta função é possível escolher os arquivos do paciente desejado e então retornar um objeto do tipo ND-array contendo uma tabela com as leituras do ECG. Um *ndarray*, um *array* que possui mais de uma dimensão, sendo nesse caso de duas dimensões. E um *dictionary* contendo informações sobre as medições como a frequência, unidade de medida das leituras e informações sobre os pacientes. Neste código, optou-se por trabalhar com os arquivos do padrão Waveform no mesmo diretório que o programa. A seguir tem-se um exemplo do código para leitura do arquivo e um exemplo dos dados obtidos pela função. O *sample* utilizado foi o de número 100, que está presente no *MIT-BIH Arrhythmia Database*.

Figura 9: Exemplificação de código para obter os dados do arquivo do Physionet.

```
1 import wfdb
2 data = wfdb.rdsamp("100") #tupla
3 tabela = data[0] #tabela ndarray
4 dicionario = data[1]
5 print(data)
```

Fonte: Autor

Figura 10: Exibição na tela da informação adquirida do Physionet

```
(array([[ -0.145, -0.065],
        [ -0.145, -0.065],
        [ -0.145, -0.065],
        ...,
        [ -0.675, -0.365],
        [ -0.765, -0.335],
        [ -1.28 ,  0.    ]]), {'fs': 360, 'sig_len': 650000, 'n
_sig': 2, 'base_date': None, 'base_time': None, 'units': ['m
V', 'mV'], 'sig_name': ['MLII', 'V5'], 'comments': ['69 M 10
85 1629 x1', 'Aldomet, Inderal']})
```

Fonte: Autor

Após a extração dos dados, cria-se um *Data Frame* para poder acessar o valores do *array*. Para a leitura e processamento optou-se por utilizar a biblioteca *Pandas*.

Esta biblioteca é um pacote que fornece estruturas rápidas, flexíveis e expressivas projetadas para trabalhar com dados. Ele tem como objetivo ser o bloco de construção fundamental de alto nível para fazer análises de dados práticos e do mundo real em Python. Além disso, tem o objetivo mais amplo de se tornar a ferramenta de análise/manipulação de dados de código aberto mais poderosa e flexível disponível em qualquer idioma [7].

Após a conversão dos dados em *Data Frame* realizamos a transformação dele em um arquivo *.csv*, para poder ser manipulado mais facilmente. Este arquivo *.csv* fica salvo no mesmo diretório no qual o código está sendo utilizado.

Logo após a criação do *.csv* é necessário dividir o arquivo em espaços menores de tempo, para ter análises mais precisas e também devido ao pouco poder de processamento que os nossos computadores possuem. Visto que conforme usamos maiores intervalos de tempo se tem mais dados e é necessário maior poder de processamento e o código demorava mais para poder rodar. No primeiro instante pegamos o arquivo e extraímos a primeira meia hora de dados. Para que depois consigamos retirar 1 minuto, e será esse 1 minuto que será processado na próxima etapa.

Para selecionar a janela de meia hora e depois de 1 minuto do arquivo principal é utilizada a função *epochs* que vem da biblioteca Neurokit2.

A biblioteca Neurokit2 é um pacote amigável que fornece acesso fácil a rotinas avançadas de processamento de biossinais. Ela foi criada com o intuito de que pesquisadores e médicos sem amplo conhecimento de programação ou processamento biomédico de sinais possam analisar dados fisiológicos com apenas duas linhas de código [8].

A função *epochs*, atua com mudanças fisiológicas que ocorrem imediatamente em resposta a um evento. Portanto, ela pode ser utilizada para realizar a análise de uma época específica. Nesta situação, a análise é baseada na época. Uma época é um pequeno pedaço do sinal fisiológico, que é bloqueado para um estímulo específico e, portanto, os sinais fisiológicos de interesse são segmentados no tempo de acordo [9].

Para realizar a quebra dentro da função, é necessário saber qual o *sampling rate*, que é obtido através do *dictionary* retornado do *rdsamp*. A partir desta frequência, calcula-se quantas linhas teremos de leitura no período de 1 minuto. E assim os dados são divididos em um pequeno arquivo contendo 1 minuto de dados do ECG.

A Figura 11 exemplifica o código, em que a função *divide_tempo* realiza a divisão dos dados. Os dados retornados por esta função são analisados utilizando a função *ecg_process* da biblioteca *neurokit2*, que será abordada a seguir. Por este motivo são salvos em formato *.csv*, assim podem ser utilizados para a análise e cálculo dos batimentos cardíacos no minuto.

Figura 11: O Código presente na função *divide_tempo*.

```
def divide_tempo(dados,fs):  
    epochs1 = nk.epochs_create( dados, events=[0, 60*fs],  
                               sampling_rate=fs,  
                               epochs_start=600,  
                               epochs_end=660)  
  
    dados_1min = epochs1['1'].to_csv('422_min1.csv') #Cria arquivo csv com a metade da tabela  
    dados_1min = pd.DataFrame(pd.read_csv('422_min1.csv')) #Cria Dataframe a partir do csv
```

Fonte: Autor

6.2 Análise dos Dados

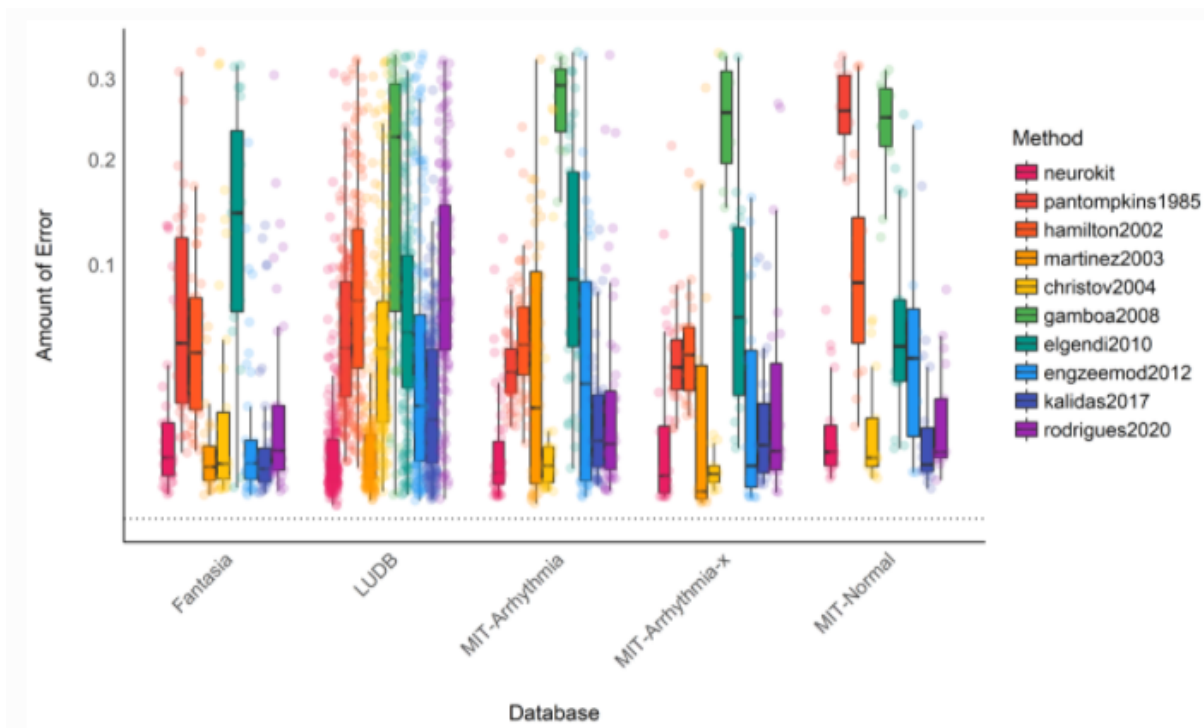
Existem várias metodologias que podem ser aplicadas para o encontro dos picos R e da variação de batimento cardíaco, de forma a poder identificar possíveis doenças cardiovasculares. A metodologia escolhida para o processamento de sinais é a do próprio Neurokit2. Essa escolha ocorre pois, como documentado pela própria biblioteca, ela é a que apresenta a menor quantidade de erros no processamento de dados de um ECG [10].

A documentação da biblioteca demonstra que, ao realizar uma comparação entre os métodos disponíveis para o processamento, o Neurokit2 é o que apresenta a menor quantidade de erros [10].

No estudo, compara-se alguns outros métodos de detecção dos picos R. O métodos analisados são Pan Tompkins criado no ano de 1985, Hamilton criado no ano de 2002, Martinez criado no ano de 2003, Christov criado no de 2004, Gamboa criado no ano de 2008, Elgendi criado no ano de 2010, Engemold criado no ano de 2012, Kalidas criado no ano de 2017 e Rodrigues criado no ano de 2020. O teste foi realizado com diversos tipos de dados de ECGs, sendo estes, os de *Glasgow University Database*, *MIT-BIH Arrhythmia Database*, *MIT-BIH Normal Sinus Rhythm Database*, *Fantasia Database*, *Lobachevsky University Electrocardiography Database* [10].

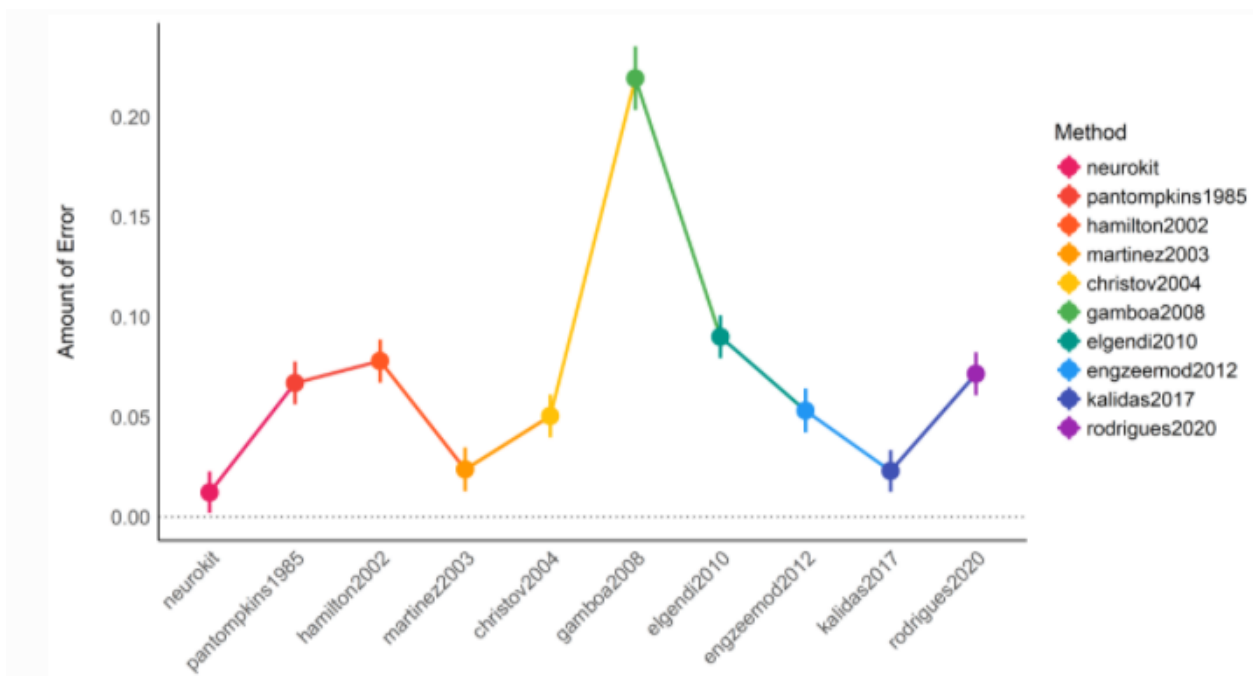
Na Figura 12 é possível visualizar que após o processamento dos dados do *database*, o método do neurokit está entre os métodos que apresentam a menor quantidade de erros. A precisão é calculada como a distância absoluta da localização dos picos R “verdadeiros” originais. Como tal, quanto mais próximo de zero, melhor será a precisão [10].

Figura 12: Gráfico representando a quantidade de erros que cada método apresentou com relação ao *database* utilizado.



Fonte: [Neurokit2. Benchmarking of ECG Preprocessing Methods]

Figura 13: Gráfico representando estatisticamente a quantidade de erros que cada método possui.



Fonte:[Neurokit2. Benchmarking of ECG Preprocessing Methods]

Na Figura 13, tem-se o gráfico que mostra que os métodos neurokit, kalidas2017 e martinez2003 são os métodos mais precisos para detectar os picos R. Este padrão de resultados difere um pouco de Porr & Howell (2019), onde é descrito que os métodos engzeemod2012, elgendi2010, kalidas2017 são os métodos mais precisos, enquanto os métodos de christov2004, hamilton2002 e pantompkins1985, são considerados os piores métodos. As possíveis discrepâncias podem ocorrer devido às diferenças entre os dados e as análises, visto que na pesquisa realizada foram usados mais bancos de dados [10].

Visto a acurácia dos métodos, optamos pelo método Neurokit, para realizar a detecção dos picos R e a frequência dos batimentos cardíacos.

Para a análise dos dados em si, se faz necessário alguns passos. Primeiramente, é preciso ler os dados que estão contidos dentro dos arquivos .csv de 1 minuto que foram obtidos na etapa anterior. Então, após a geração dos dados, estes dados serão armazenados em uma nova variável.

Para o processamento dos dados existem diversas maneiras de identificação dos picos R dentro de uma onda em um ECG. Usaremos a função `ecg_process()`, pois através do processamento dessa função que está presente na biblioteca Neurokit2, se torna possível o encontro dos picos R, além de diversas outras medidas de extrema importância.

Explicitada a seleção do método de análise, precisamos, além do método selecionado, o *sampling rate* no qual o sinal foi criado. Essa informação é obtida através do `rdsamp` utilizado na etapa anterior, em que o *dictionary* gerado nos dá a informação do mesmo.

Na chamada da função `ecg_process`, são inseridos como parâmetro os dados referentes a janela de 1 minuto selecionada em formato Data Frame e o *sampling rate* do exame realizado. Como *default*, a função já utiliza o método neurokit. Realiza-se o processamento dos dados e são retornadas diversas informações, dentre elas as mais importantes, como a detecção do sinal tratado, a detecção de picos R e a própria frequência cardíaca. A partir destas informações será gerado um arquivo .csv com as informações salvas. Para que ele possa ser consultado e a partir das informações obtidas consigamos entender melhor o que está acontecendo com o indivíduo durante o exame.

A Figura 14 mostra a função `processa_plota`, que está sendo chamada. É nela que será realizado o processamento dos dados. Também é o local onde é acionada a função `ecg_plot` que irá plotar o gráfico. Gráfico este que mostra o sinal processado com a presença dos picos R e a variação de frequência no período de 1 minuto. Ao final a função `processa_plota` retorna o sinal do eletrocardiograma processado.

Figura 14: O Código presente na função `processa_plota`

```
def processa_plota(dados_1min,fs):  
    # Process ecg  
  
    ecg_signals, info = nk.ecg_process(dados_1min["ECG(mv)"], sampling_rate=fs,method="neurokit")  
    plot = nk.ecg_plot(ecg_signals,sampling_rate=fs)  
    plt.show()  
    |  
    return ecg_signals
```

Fonte: Autor

É a partir do `.csv` que se tem todas as informações do exame no período, que é gerado um novo arquivo `.csv` com os dados da coluna "ECG_Rate". São os dados desta coluna que iremos utilizar para obter a frequência máxima, frequência média e também a frequência mínima. Para a realização disto, primeiramente iremos alocar em uma variável o tamanho do número de linhas. Esta variável será utilizada para poder pegarmos o seu comprimento e gerar um laço de repetição `for` que irá iterar por todas os dados do arquivo e realizar a leitura e alocação da frequência mínima na variável `rate_min`, e a alocação da frequência máxima na variável `rate_max`.

Em seguida, realiza-se o cálculo da frequência média do período, utilizando o comando chamado `.mean`. Este comando realiza o cálculo da frequência média dentro daquele período de 1 minuto que está sendo analisado.

Na Figura 15 tem-se a demonstração de como identificamos a menor frequência e como é encontrada a maior frequência no período de 1 minuto. Utilizou-se um laço `for` para iterar os dados processados e armazenar os valores de `rate_max` e `rate_min`.

Figura 15: O Código presente na função min_max

```
def min_max(hrv_1min,sig_len,rate_max,rate_min,x):  
    for i in range(int(sig_len)):  
        x = hrv_1min['ECG_Rate'][i]  
        if x < rate_min:  
            rate_min = x  
        if x > rate_max:  
            rate_max = x  
    return rate_min,rate_max
```

Fonte: Autor

Depois do cálculo da frequência iremos encontrar o número de batimentos por minuto. Para a realização desta análise será necessário usar o arquivo .csv, processado com 1 minuto. Neste arquivo as linhas reconhecidas como picos R na onda do batimento cardíaco são identificadas pelo valor “1” na coluna “R_Peaks”. Ao realizar a conta de quantos picos R ocorreram no período de 1 minuto se torna possível identificar o número de batimentos cardíacos naquele minuto.

O cálculo do número de batimentos cardíacos é de extrema importância, pois é através dele que se identificam variações dos níveis que são considerados normais, ocasionando assim o encontro de doenças cardiovasculares por profissionais especializados.

Exemplos de doenças que podem ser encontradas são Bradicardia e taquicardia sinusal entre outras.

Estas e outras informações serão usadas na etapa de demonstração dos dados, em que iremos conseguir plotar o gráfico com o sinal processado, demonstrar toda a variação de frequência no período e também a demonstração dos dados obtidos de variação de frequência e o número de batimentos cardíacos por minuto.

6.3 Demonstração dos Dados

Como última etapa da metodologia serão apresentados os dados obtidos. Os dados obtidos são apresentados de duas maneiras. Primeiramente realiza-se a plotagem de dois gráficos em uma janela gerada usando a função `ecg_plot`. E a segunda parte em que iremos mostrar na tela os valores obtidos de, frequência média, frequência máxima, frequência mínima e os valores de batimentos cardíacos por minuto.

Explicando primeiro a janela, como já dito, teremos dois gráficos. Um gráfico que contém o gráfico apresentado com o sinal já processado, com a presença da identificação dos picos R, enquanto são demonstrados os batimentos cardíacos com a amplitude do sinal em mV (miliVolts). E outro gráfico que possui as informações da variação da frequência cardíaca em função do tempo. Apresentando a frequência mais alta, a mais baixa e a média no período de 1 minuto.

Para esta etapa, temos de seguir alguns passos. Primeiramente, para a criação da janela com a plotagem dos gráficos, nós pegamos o arquivo `.csv`, que contém as informações do sinal processado, lembrando sempre que este sinal foi processado com o intervalo de 1 minuto acerca dos dados grossos obtidos do ECG.

Para a etapa de plotagem se faz necessária a utilização de mais uma biblioteca em python, neste caso a `matplotlib`.

A `matplotlib.pyplot` é uma coleção de funções que fazem com que o `matplotlib` funcione como um programa realizado no MATLAB. Cada função do `pyplot` realiza alguma modificação em uma figura [13].

No `matplotlib.pyplot` diversos estados são preservados quando ocorre o chamado de alguma função. Para que seja mantido o controle da figura atual e a área de plotagem, as funções que realizam as plotagens são direcionadas para os eixos atuais [13].

Então para a plotagem do gráfico é chamada a função `ecg_plot`, que pertence a biblioteca da `Neurokit2`, que utiliza como base a biblioteca `matplotlib`. Esta função recebe dois dados, o sinal processado que foi processado na função `ecg_process` e o *sampling rate*, com a frequência do qual o sinal foi sampleado.

Sendo assim o gráfico que será plotado é de 1 minuto com os dados processados. Nós modificamos a função `ecg_plot`, pois esta mostrava informações que não desejávamos como por exemplo o sinal bruto, pré tratamento. Como não tínhamos o interesse pela demonstração destes gráficos como resultado, optamos por retirá-los. Além de mudanças com relação ao tamanho para melhor encaixe para demonstração dos dados.

Outra demonstração que ocorre é a de uma janela que mostra os dados como, frequência mínima, frequência máxima, frequência média e os batimentos que aconteceram dentro daquele minuto. Para a aparição destas informações na tela iremos usar a biblioteca Python chamada de `tkinter`.

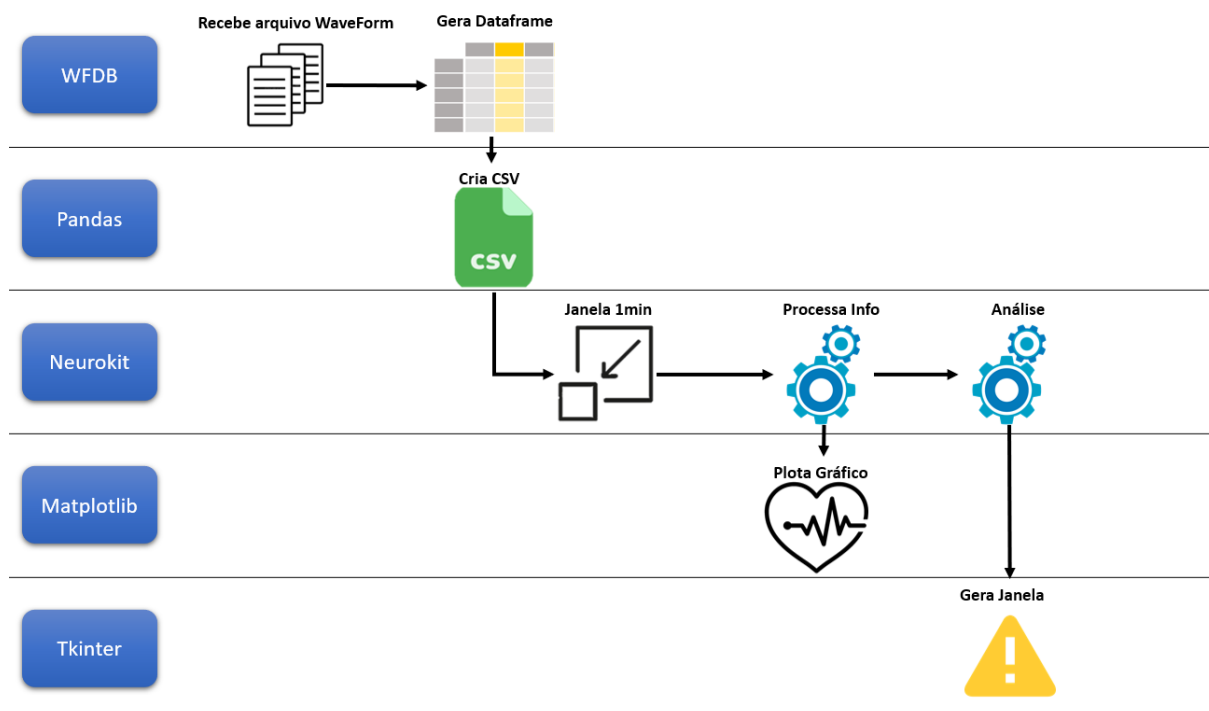
A biblioteca `tkinter` é a interface Python padrão para o kit de ferramentas GUI Tcl / Tk. `Tkinter` suporta uma variedade de versões Tcl / Tk, construídas com ou sem suporte a `thread`. O lançamento oficial do binário Python agrupa Tcl / Tk 8.6 encadeado [14].

É utilizada a biblioteca `tkinter`, visto que, com ela é possível criar uma janela de aviso com o layout desejado e também mostrar as variáveis, além de conseguir criar caixa de texto para adquirir dados, mas esta última opção nós não iremos utilizar.

6.4 Resumo

A Figura 16 explicita de maneira bem simples e visual todo o processo realizado com relação ao uso das bibliotecas.

Figura 16: Exemplificação de como cada biblioteca é utilizada no programa.



Fonte: Autor

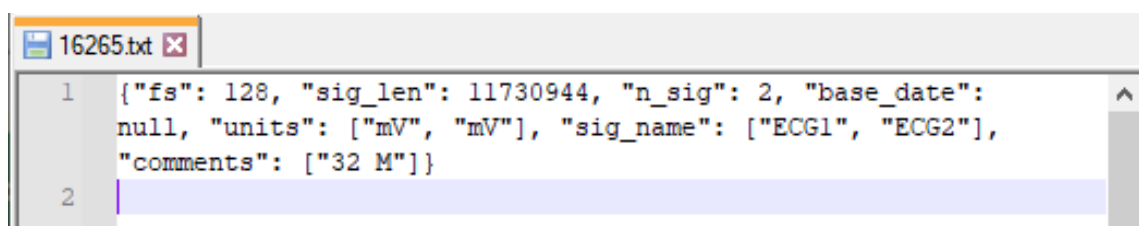
A figura representa de maneira simples todo o processo relativo à metodologia por trás do código. Em que é recebido o arquivo do database no formato *waveform*, então ele é lido e transformado em um dataframe pela biblioteca WFSB. Logo após é utilizada a biblioteca Pandas para realizar a criação do artigo em .csv e também a alocação em variáveis no formato data frame para manipulação. A próxima etapa na biblioteca Neurokit, é realizada a retirada de 1 minuto do código para que esse possa ser processado e a partir dele podemos fazer nossas análises. Para a última etapa, é gerada a janela, em que mostraremos os dados obtidos utilizando a biblioteca Tkinter. E também o gráfico que será gerado com o sinal tratado, junto com a representação dos picos R junto e outro gráfico mostrando a variação da frequência que ocorre no período de 1 minuto.

7. Resultados e Discussões

Para a divulgação dos resultados e depois realizar uma discussão sobre os resultados obtidos, primeiramente se faz necessário o entendimento do processo que foi feito para a obtenção dos resultados. No primeiro momento é necessária a lembrança de onde vem os dados que serão analisados.

O ECG apresentado a seguir é o de código 16265, este está presente no *database*, *MIT-BIH Normal Sinus Rhythm Database*. Como citado anteriormente, os indivíduos incluídos nesta *database* não apresentam arritmias significativas e podem exemplificar o comportamento normal do coração. O indivíduo foi escolhido de maneira aleatória para a utilização neste estudo. As informações que obtém-se do arquivo foram obtidas através da função *rdsamp*, e estão presentes na Figura 17, que é de um *.txt* que foi gerado com as informações do *sample*.

Figura 17: Arquivo *.txt* gerado com informações sobre o *sample* 16265



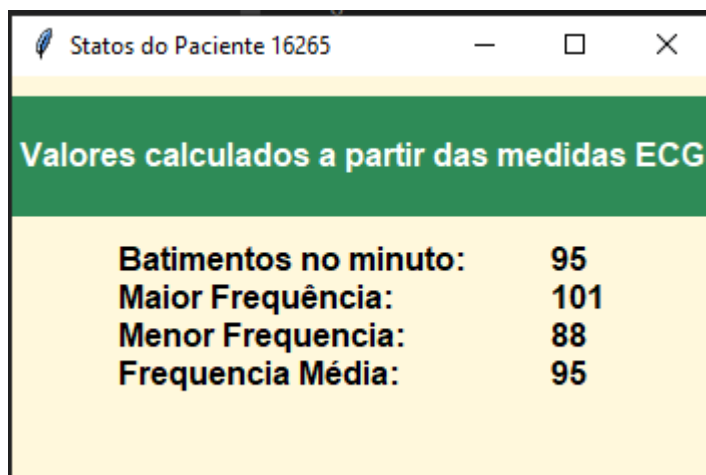
```
1 {"fs": 128, "sig_len": 11730944, "n_sig": 2, "base_date":  
  null, "units": ["mV", "mV"], "sig_name": ["ECG1", "ECG2"],  
  "comments": ["32 M"]}  
2
```

Fonte: Autor

A partir das informações do *txt* é possível identificar a frequência utilizada quando o sinal foi gravado, que foi de 128 Hz. E a sua unidade foi em mV.

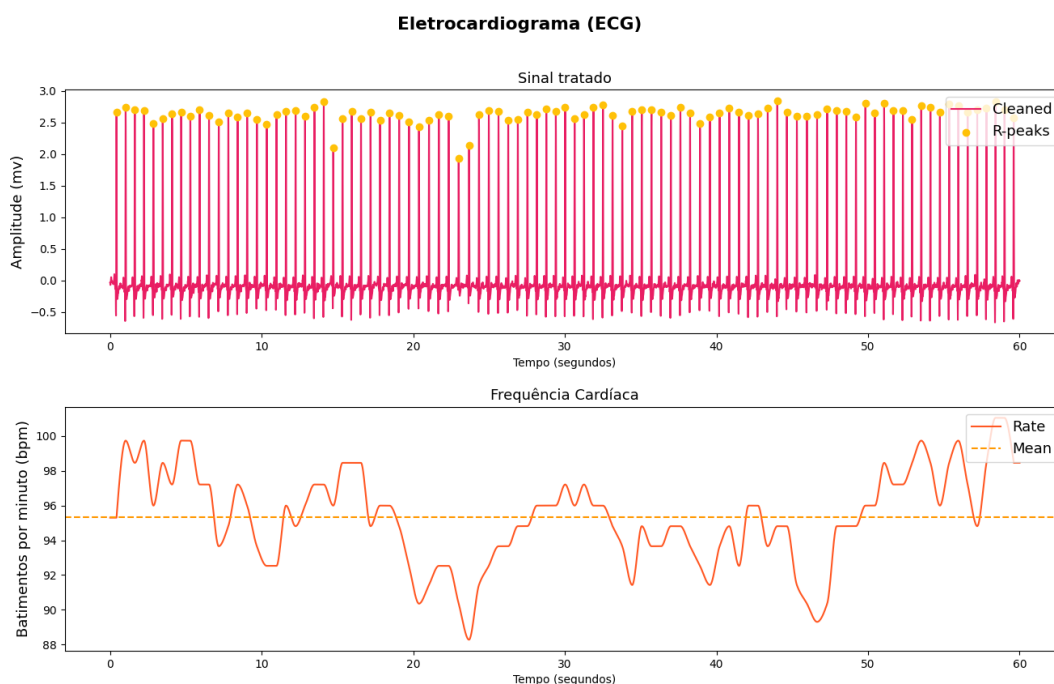
A partir destas informações tratamos os dados com as informações corretas e ao passar por toda a metodologia de obtenção de dados, foi alcançado o seguinte resultado demonstrado nas Figuras 18 e 19.

Figura 18: Janela mostrando os dados do sample 16265



Fonte: Autor

Figura 19: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 16265



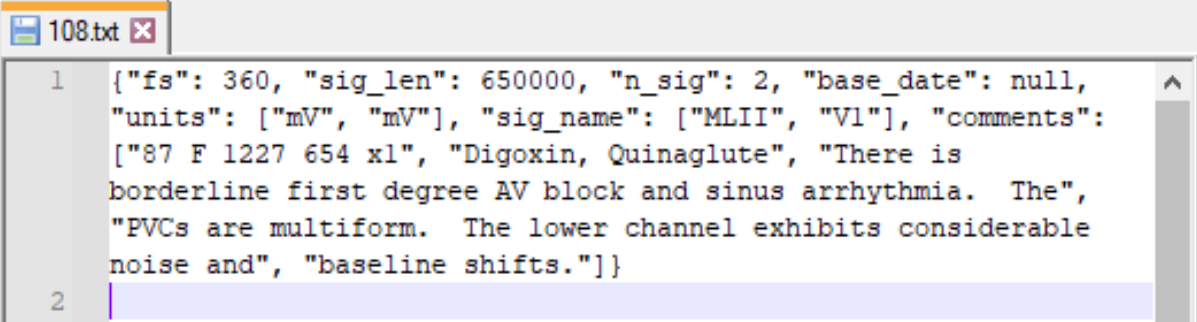
Fonte: Autor

A Figura 19 mostra no gráfico superior o sinal tratado com a identificação dos picos R, durante todo o sinal. Onde é possível identificar que o método encontra com

precisão os picos R durante toda a onda do ECG. E no gráfico inferior é demonstrada a variação da frequência cardíaca. Em que é possível ver visualmente como a frequência com o qual o coração bate no período de um minuto está sempre variando, mas mesmo assim neste caso se encontra dentro dos padrões considerados saudáveis. Em que temos os batimentos por minuto no valor de 95, a frequência máxima no valor de 101 e a frequência mínima no valor de 88. E que a frequência média corresponde ao número de batimentos que tivemos dentro do minuto analisado.

O primeiro ECG que será analisado é o de código 108, que está presente no database, *MIT-BIH Arrhythmia Database*. Ele foi escolhido aleatoriamente para a utilização. As informações que obtém-se do arquivo foram obtidas através da função *rdsamp*, e estão presentes na Figura 17, que é de um txt que foi gerado com as informações do *sample*.

Figura 20: Arquivo .txt gerado com informações sobre o sample 108.



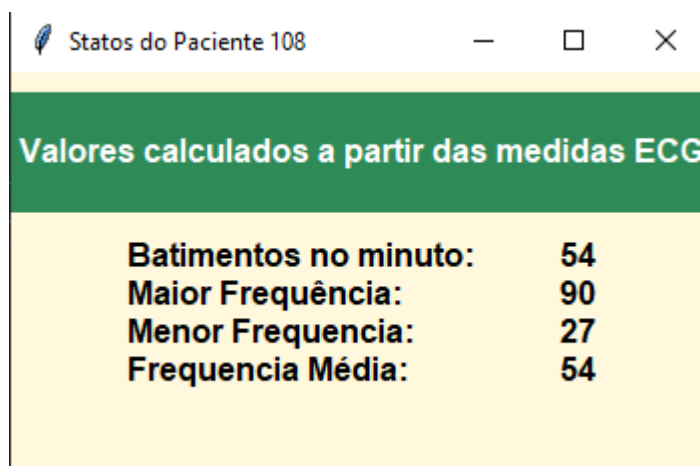
```
1 {"fs": 360, "sig_len": 650000, "n_sig": 2, "base_date": null,
  "units": ["mV", "mV"], "sig_name": ["MLII", "V1"], "comments":
  ["87 F 1227 654 xl", "Digoxin, Quinaglute", "There is
  borderline first degree AV block and sinus arrhythmia. The",
  "PVCs are multiform. The lower channel exhibits considerable
  noise and", "baseline shifts."]}
2
```

Fonte: Autor

A partir das informações do txt é possível identificar a frequência utilizada quando o sinal foi gravado, que foi de 360 Hz. E a sua unidade foi em mV.

A partir destas informações tratamos os dados com as informações corretas e ao passar por toda a metodologia de obtenção de dados, foi alcançado o seguinte resultado, demonstrado pelas Figuras 21 e 22.

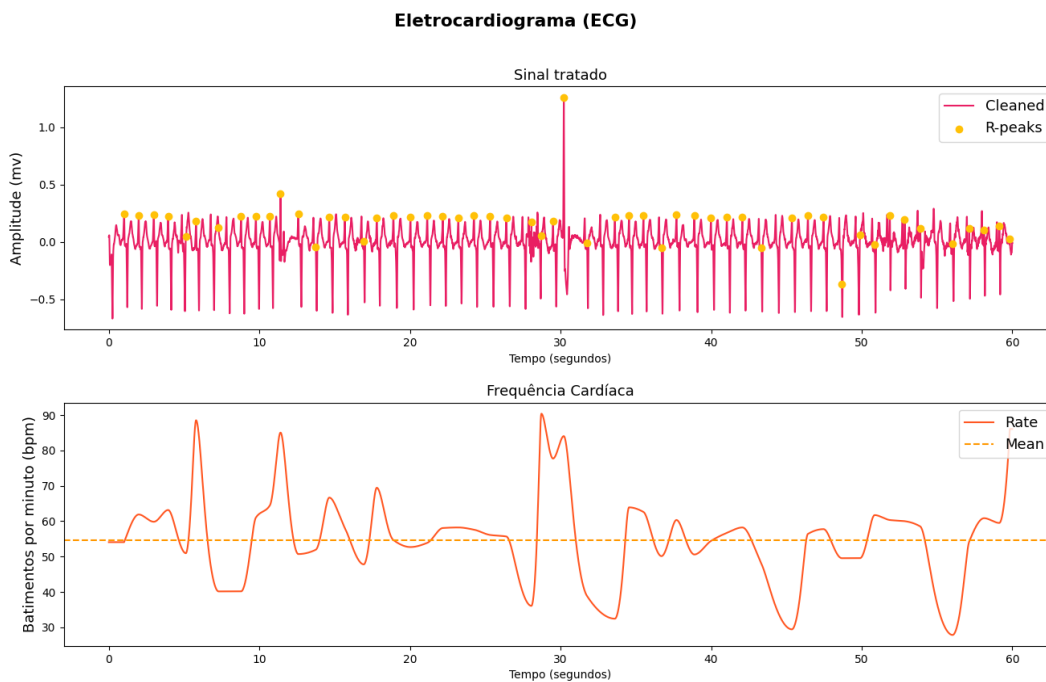
Figura 21: Janela mostrando os dados do sample 108.



Fonte: Autor

Após a execução, o código retorna a tela apresentada na Figura 21, onde apresentam-se os valores calculados para número de batimentos no minuto, maior frequência cardíaca apresentada no intervalo, menor frequência cardíaca apresentada no intervalo, frequência média apresentada no intervalo.

Figura 22: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 108.

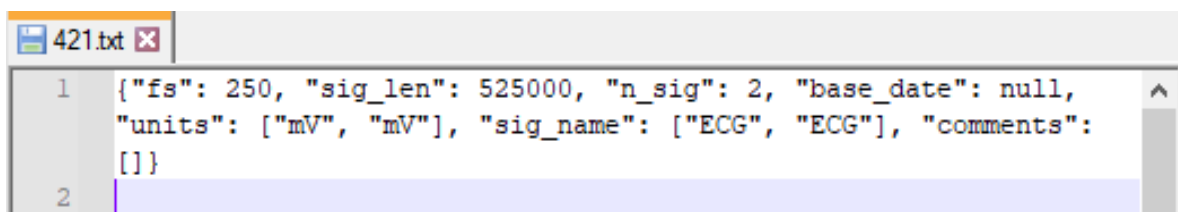


Fonte: Autor

A Figura 22, no gráfico superior, mostra o sinal tratado com a identificação dos picos R, durante todo o sinal. E no gráfico inferior é demonstrada a variação da frequência cardíaca. Em que é possível ver visualmente como a frequência com o qual o coração bate no período de um minuto está sempre variando.

O segundo ECG que será analisado é o de código 421, este está presente no database, *MIT-BIH Malignant Ventricular Ectopy*. Este também foi escolhido de maneira aleatória para a utilização. As informações que obtém-se do arquivo foram obtidas através da função *rdsamp*, e estão presentes na Figura 20 que encontra-se abaixo, que é de um txt que foi gerado com as informações do *sample*.

Figura 23: Arquivo .txt gerado com informações sobre o sample 421



```
421.txt x
1 {"fs": 250, "sig_len": 525000, "n_sig": 2, "base_date": null,
  "units": ["mV", "mV"], "sig_name": ["ECG", "ECG"], "comments":
  []}
2
```

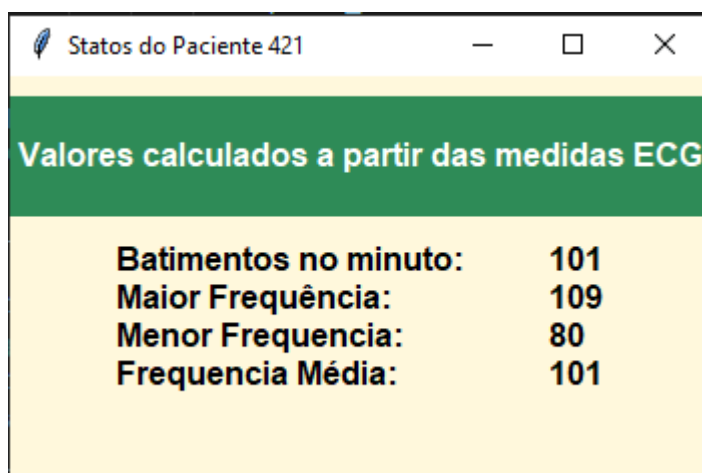
Fonte: Autor

A partir das informações do txt é possível identificar a frequência utilizada quando o sinal foi gravado, que foi de 250 Hz. E a sua unidade foi em mV.

A partir destas informações tratamos os dados com as informações corretas e ao passar por toda a metodologia de obtenção de dados, foi alcançado o seguinte resultado, que está demonstrado nas Figuras 24 e 25.

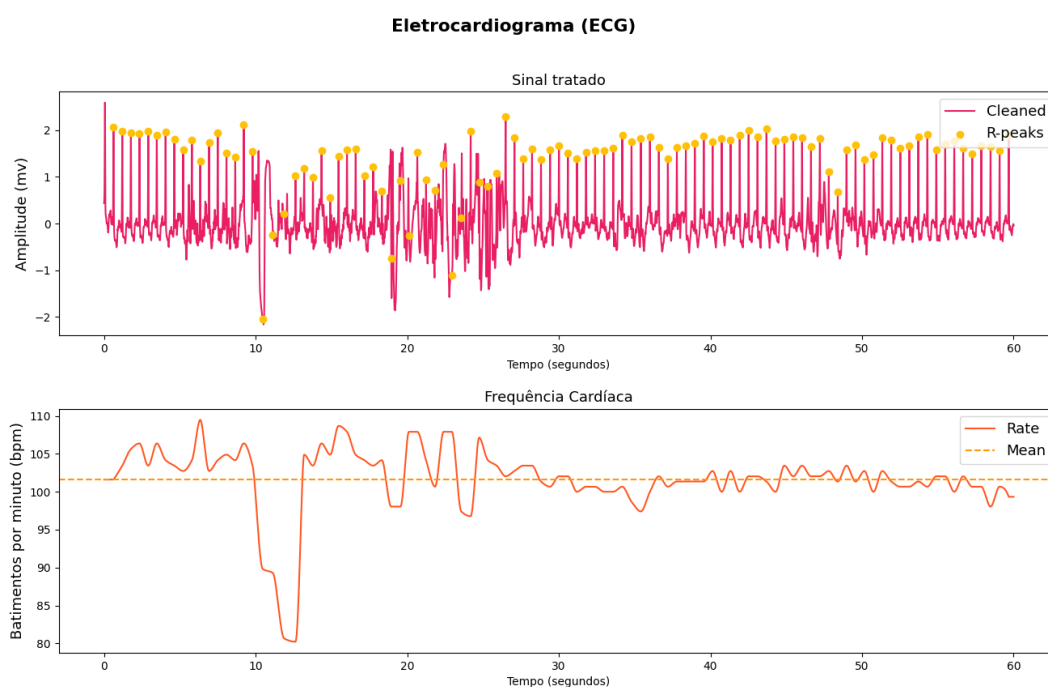
Na Figura 24 é possível identificar que no período analisado de um minuto, os dados obtidos foram de uma frequência cardíaca que pode ser considerada alta. Os dados retornados podem ser utilizados para facilitar a análise de um profissional da saúde.

Figura 24: Janela mostrando os dados do sample 421.



Fonte: Autor

Figura 25: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 421.



Fonte: Autor

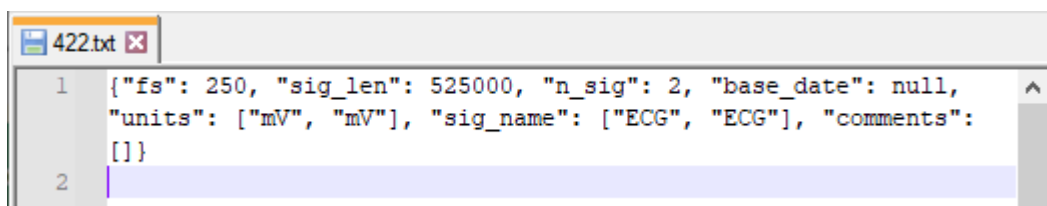
A Figura 25 mostra no gráfico superior o sinal tratado com a identificação dos picos R, durante todo o sinal. E no gráfico abaixo é demonstrada a variação da frequência cardíaca. Em que é possível ver visualmente como a frequência com o

qual o coração bate no período de um minuto está sempre variando. No gráfico de cima é possível identificar que o método possui certa dificuldade de alocar todos os picos R de maneira exata, ele identifica os picos mas às vezes os aloca um pouco antes ou posterior a realmente a presença do pico. Enquanto que com o sinal mais perto de seguir um padrão, ele consegue identificar mais facilmente os picos R.

Também é possível ver no gráfico que é encontrada uma frequência média de 101 batimentos por minuto, junto com uma frequência máxima de 109 e uma frequência mínima de 80.

O terceiro ECG que será analisado é o de código 422, este está presente no *database*, *MIT-BIH Malignant Ventricular Ectopy*. Este também foi escolhido de maneira aleatória para a utilização. As informações que obtém-se do arquivo foram obtidas através da função *rdsamp*, e estão presentes na Figura 23, que é de um arquivo .txt gerado com as informações do *sample*.

Figura 26: Arquivo .txt gerado com informações sobre o sample 422

A screenshot of a text editor window titled '422.txt'. The window displays two lines of JSON data. Line 1 contains a JSON object with the following fields: 'fs' (250), 'sig_len' (525000), 'n_sig' (2), 'base_date' (null), 'units' (['mV', 'mV']), 'sig_name' (['ECG', 'ECG']), and 'comments' ([]). Line 2 is empty. The text is displayed in a monospaced font on a light background.

```
1 {"fs": 250, "sig_len": 525000, "n_sig": 2, "base_date": null,
  "units": ["mV", "mV"], "sig_name": ["ECG", "ECG"], "comments":
  []}
2
```

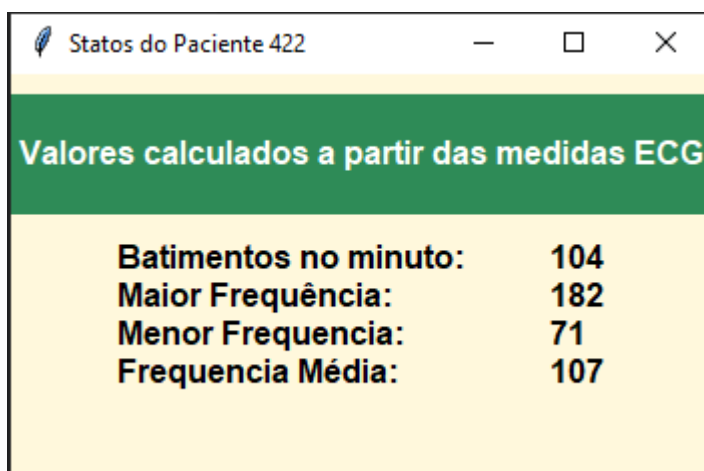
Fonte: Autor

A partir das informações do .txt é possível identificar a frequência utilizada quando o sinal foi gravado, que foi de 250 Hz, igual ao sinal anterior. E a sua unidade foi em mV.

A partir destas informações tratamos os dados com as informações corretas e ao passar por toda a metodologia de obtenção de dados, foi alcançado o seguinte resultado, apresentado nas Figuras 27 e 28.

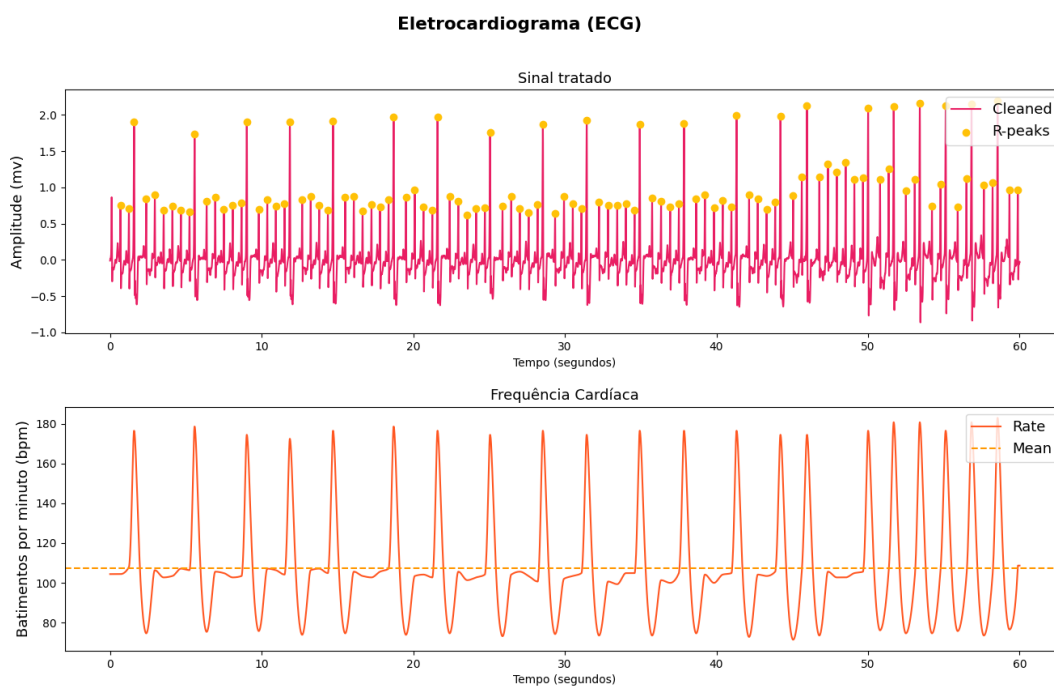
Com a Figura 27 é possível identificar que no período analisado de um minuto, os dados obtidos foram de uma frequência cardíaca considerada alta.

Figura 27: Janela mostrando os dados do sample 422.



Fonte: Autor

Figura 28: Gráficos mostrando a onda ECG e a variação de frequência no período sobre os dados do sample 422.



Fonte: Autor

A Figura 28 mostra no gráfico superior o sinal tratado com a identificação dos picos R, durante todo o sinal. E no gráfico abaixo é demonstrada a variação da frequência cardíaca. Em que é possível ver visualmente como a frequência com o

qual o coração bate no período de um minuto está sempre variando. No gráfico de cima é possível identificar que em comparação com o *sample* 421, neste caso existe uma maior facilidade de detecção dos picos R. Visto que mesmo com uma grande variação da frequência, esta é uma variação que se mantém durante todo o período. Sendo assim mais fácil para o método conseguir alocar todos os picos R de maneira exata, assim como está demonstrado no gráfico de cima.

Neste *sample*, também é possível identificar que a frequência média difere do número de batimentos que aconteceram no minuto, isso se dá devido ao batimento irregular do coração. Mas esse valor não difere tanto da quantidade de batimentos cardíacos que aconteceram dentro de 1 minuto.

Também é possível ver no gráfico inferior que é encontrada uma frequência média de 107 batimentos por minuto, junto com uma frequência máxima de 182 e uma frequência mínima de 71.

8. Conclusões

Durante todo o trabalho houveram várias etapas, a primeira etapa foi conseguir compreender todos os conceitos por trás de um eletrocardiograma e qual sua importância na utilização de identificação de possíveis problemas cardíacos. Nesta etapa conseguimos compreender o funcionamento do coração com todo o seu ciclo, além de como é importante conseguir identificar quaisquer alterações no seu batimento.

Como segunda etapa tivemos de adquirir todo o conhecimento na linguagem Python, para poder realizar todo o código. Esta etapa foi extremamente desafiadora, visto que cada biblioteca que usamos possui suas peculiaridades e maneiras corretas de utilizá-las. E conseguimos compreender o porquê do Python ser uma linguagem cada vez mais utilizada ao redor do mundo para a ciência de dados.

Após a leitura dos dados, seu armazenamento foi realizado em formato de arquivo .csv para ser um arquivo mais leve e de fácil manuseio. Seguido da divisão

em partes pequenas, para assim poder processar e identificar os dados que necessitamos e assim poder mostrar para o usuário na forma de alerta identificando possíveis arritmias e grandes variações de frequências.

Logo após o decorrer do código e a apresentação dos dados que foram obtidos é possível retirar algumas conclusões acerca do objetivo do trabalho que era a realização de um código em Python que conseguisse ler os dados de um eletrocardiograma. Para poder realizar análise e mostrar os dados obtidos.

Sendo assim, o código cumpre todos os requisitos, de leitura, análise e demonstração para o usuário do que encontrou. Os únicos erros possíveis de acontecer seria uma aproximação errada no método selecionado, em que ele não conseguisse identificar corretamente os picos da onda R, ou encontrasse com muitos erros nos cálculos realizados, mas como visto, o método utilizado é o que apresentou menos erros. Portanto, existe uma boa aproximação do sinal real conseguindo assim uma grande quantidade de acertos no encontro dos dados independente do tipo de sinal que está sendo analisado.

Referências Bibliográficas

[1] A. de Padua Mansur, D. Favarato. Mortalidade por Doenças Cardiovasculares no Brasil e na Região Metropolitana de São Paulo: Atualização 2011. Instituto do Coração (InCor) – HCFMUSP, São Paulo, SP - Brasil, 2011

[2] A. K. da S. Freire, N. C. Costa Alves, E. J. P. Santiago, A. Saraiva Tavares ,D. Da Silva Teixeira, I. Alencar Carvalho, M. C. Pimentel de Melo, M. Negro-Dellacqua. Panorama no Brasil Das Doenças Cardiovasculares Dos Últimos Quatorze Anos Na Perspectiva Da Promoção à Saúde.Revista Saúde e Desenvolvimento| vol.11, n.9, 2017

[3] PASTORE, CA et al.Diretriz de Interpretação de Eletrocardiograma de Repouso. Arq Bras Cardiol volume 80, (suplemento II), 2003

[4] George B. Moody, Roger G. Mark, Ary L. Goldberger. Harvard-MIT Division of Health Sciences and Technology and Massachusetts Beth Israel Deaconess Medical Center. IEEE Engineering in Medicine and Biology, Maio/Junho 2001

[5] J. Feldman, Gerson P. Goldwasser. Eletrocardiograma: Recomendações para a sua interpretação. Universidade Federal do Rio de Janeiro, Faculdade de Medicina Souza Marques, Universidade Gama Filho. Revista da SOCERJ - Out/Nov/Dez 2004

[6] Physionet. About Records. Disponível em: <<https://archive.physionet.org/physiobank/physiobank-intro.shtml>> Acesso em 27/11/2021

[7] Pandas. Package overview. Disponível em: <https://pandas.pydata.org/docs/getting_started/overview.html>. Acesso em 27/11/2021

[8] Neurokit2. The Python Toolbox for Neurophysiological Signal Processing. Disponível em: <<https://neurokit2.readthedocs.io/en/latest/introduction.html#>>. Acesso em 28/11/2021

[9] Neurokit2. Functions. Disponível em: <https://neurokit2.readthedocs.io/en/latest/functions.html?highlight=EPOCHS#neurokit2.epochs.epochs_plot> Acesso em 28/11/2021

[10] Neurokit2. Benchmarking of ECG Preprocessing Methods. Disponível em: <https://neurokit2.readthedocs.io/en/latest/studies/ecg_preprocessing.html?highlight=MIT#mit-bih-arrhythmia-database> Acesso em 28/11/2021

[11] Rede D'or São Luiz. Doenças > Bradicardia. Disponível em: <<https://www.rededorsaoluiz.com.br/doencas/bradicardia>>. Acesso em 28/11/2021

[12] .Rede D'or São Luiz. Doenças > Taquicardia. Disponível em:
<<https://www.rededorsaoluiz.com.br/doencas/taquicardia>>. Acesso em 28/11/2021

[13] Matplotlib. Pyplot . Disponível em:
<<https://matplotlib.org/stable/tutorials/introductory/pyplot.html>>. Acesso em
28/11/2021

[14] Python. tkinter — Python interface to Tcl/Tk. Disponível em:
<<https://docs.python.org/3/library/tkinter.html>>. Acesso em 28/11/2021

[15] Kaggle. Getting Started. Disponível em:
<<https://www.kaggle.com/getting-started/449160>>. Acesso em 29/11/2021

[16] Bob Hayes. Programming Languages Most Used and Recommended by Data Scientists. Disponível em:
<<https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/>>. Acesso em 27/11/2021

[17] Physionet. MIT-BIH Arrhythmia Database. Disponível em:
<<https://physionet.org/content/mitdb/1.0.0/>>. Acesso em 30/11/2021

[18] Physionet.MIT-BIH Normal Sinus Rhythm Database. Disponível em:
<<https://physionet.org/content/nsrdb/1.0.0/>>. Acesso em 30/11/2021

[19] Physionet.MIT-BIH Malignant Ventricular Ectopy Database. Disponível em:
<<https://physionet.org/content/vfdb/1.0.0/>>. Acesso em 30/11/2021

[20] ResearchGate. Heart and bile acids – Clinical consequences of altered bile acid metabolism. Disponível em:
<https://www.researchgate.net/figure/The-blood-flow-through-the-heart-during-the-cardiac-cycle-in-correspondence-with-the-QRS_fig2_322289204>. Acesso em
17/12/2021

[21] Richard E. Klabunde. Cardiovascular Physiology Concepts. Disponível em: <<https://www.cvphysiology.com/Arrhythmias/A013a>> Acesso em 17/12/2021