UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Felipe do Nascimento

Desenvolvimento de um software para análise da variabilidade do ritmo cardíaco

Florianóplis
2019

Felipe do Nascimento

Desenvolvimento de um software para análise da variabilidade do ritmo cardíaco

Trabalho de conclusão de curso submetida ao Programa de Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia Elétrica.
Orientador: Prof. Jefferson Luiz Brum Marques, Ph.D.

Florianóplis
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Nascimento, Felipe do
Desenvolvimento de um software para análise da variabilidade do ritmo cardíaco / Felipe do Nascimento ; orientador, Jefferson Luiz Brum Marques, 2019.
91 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Elétrica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Elétrica. 2. Software. 3. Variabilidade de ritmo cardíaco. 4. Engenharia Biomédica. I. Marques, Jefferson Luiz Brum. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Felipe do Nascimento

Desenvolvimento de um software para análise de variabilidade do ritmo cardíaco

Este Trabalho foi julgado adequado como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica e aprovado, em sua forma final, pela Banca Examinadora

Florianópolis, 06 de dezembro de 2019.

Documento assinado digitalmente
Jean Vianei Leite
Data: 10/02/2020 15:51:41-0300
CPF: 003.474.909-80

Prof. Jean Vianei Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:

Documento assinado digitalmente
Jefferson Luiz Brum Marques
Data: 08/02/2020 10:33:37-0300
CPF: 457.624.060-91

Prof. Jefferson Luiz Brum Marques, Ph.D.
Orientador
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Caio Venancio Duarte Carvalho
Data: 10/02/2020 09:45:35-0300
CPF: 067.413.665-98

Prof. Caio Venancio Duarte Carvalho, Esp.
Instituto Fedral do Norte de Minas Gerais

Documento assinado digitalmente
Sandra Cossul
Data: 09/02/2020 23:42:30-0300
CPF: 026.490.140-16

Eng.ª, Sandra Cossul, MSc.
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus amigos que sempre me apoiaram durante o período desse curso e aos meus pais.

AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais, Odete e Clenemar, por sempre terem piorizado a minha educação e me apoiado a seguir em frente e não parar até alcançar meus objetivos.

Gostaria também de agradecer a todo o apoio dado pela minha irmã, Gabrieli, que sempre vinha ver se estava tudo bem comigo e sempre me ajudou quando não estava.

Aos meus melhores amigos, Jordano e Alessandra, que nunca negaram me dar um conselho ou me acalmar em momentos de ansiedade. Também gostaria de agradecer os meus amigos de Florianópolis, principalmente a Carla, a Caroline e o Rafael, que foram a minha família durante esses seis anos de faculdade.

E por último e não menos importante, gostaria de agradecer ao professor Jefferson por ser meu orientador nessa etapa final do curso, e a banca, a engenheira Sandra e o professor Caio.

“A tarefa não é tanto ver
o que ninguém viu, mas
pensar o que ninguém ainda
pensou sobre aquilo que todo
mundo vê”
(Arthur Schopenhauer)

RESUMO

Esse trabalho teve como objetivo a implementação de um *software* de análise de sinal de eletrocardiograma pelo método de variabilidade do ritmo cardíaco. Com o avanço das tecnologias médico-hospitalares e os estudos na área de medicina e engenharia biomédica, novos métodos de análise estão sendo empregados a sinais conhecidos, de modo a facilitar e agilizar a identificação de patologias e possíveis anormalidades na saúde dos pacientes. Nesse âmbito, um dos métodos que está se popularizando é o método de análise de variabilidade de ritmo cardíaco, onde são observadas as diferenças de espaço temporal entre duas ondas R do eletrocardiograma. Deste modo, a metodologia adotada para o trabalho é a contrução de um *software* com a utilização de bibliotecas externas do Python que englobe todo o desenvolvimento do pré processamento do sinal de eletrocardiograma, da obtenção dos intervalos entre as ondas R e do cálculo dos parâmetros dos métodos de análise no domínio do tempo, no domínio da frequência e os parâmetros não lineares.

Palavras-chave: *Software*. Variabilidade de frequência cardíaca. Python.

ABSTRACT

The main focus of this work was to implement a software for electrocardiogram signal analysis using heart rate variability. With the advancement of medical-hospital technologies and studies in the field of medicine and biomedical engineering, new methods of analysis are being employed with known signals, in order to facilitate and expedite the identification of pathologies and possible abnormalities in patients' health. In this context, one of the methods that is becoming more popular is the heart rate variability analysis method, wich considers the time defference between two consecutives R waves of the electrocardiogram. Thus, the methodology adopted for the work is the construction of a Python language software wich includes the entire development of the electrocardiogram signal preprocessing, the obtaining of the R-wave intervals and the calculation of the required parameters, in the time domain, frequency domain and non-linear parameters.

Keywords: Software. Heart rate variability. Python.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

CMRR	*Common Mode Rejection Ratio*
CVNNi	*Coefficient of variation*
CVSD	*Coefficient of variation of the standard deviation*
ECG	Eletrocardiograma
FFT	*Fast Fourier Transform*
FIR	*Finite Impulse Response*
GUI	*Graphical User Interface*
HRV	*Heart Rate Variability*
PSD	*Power Spectral Density*
RMSSD	*Root of the mean square of successive differences*
SDNN	*Standard deviation of the NN interval*
SNA	Sistema Nervoso Autônomo
SNC	Sistema Nervoso Central
SNP	Sistema Nervoso Priférico
SNR	*Signal-noise Ratio*
TINN	*Triangular Interpolation of NN*

SUMÁRIO

1 INTRODUÇÃO

Variabilidade do ritmo cardíaco, ou *Heart Rate Variability* (HRV), é a variação entre os intervalos de tempo entre dois batimentos cardíacos consecutivos. Essa medida está sendo cada vez mais empregada em análises de doenças cardíacas e as suas relações com o sistema nervoso, principalmente o Sistema Nervoso Autônomo (SNA). Esse método pode ser utilizado na identificação de doenças do coração como arritmias e riscos de paradas cardiovasculares (SHAFFER; GINSBERG, 2017), e também de doenças relacionadas ao sistema nervoso como depressão, ansiedade, bipolaridade e estresse (HAGE *et al.*, 2017).

Além disso, o uso do método de HRV está se popularizando nos meios esportivos, sendo empregado em treinos focados para esportes específicos, como ciclismo e análise de desempenho dos atletas (LASS *et al.*, 2019), na identificação de inflamações (P.WILLIAMS *et al.*, 2019), em estudos das consequências de falência do fígado causada por cirrose (MANI *et al.*, 2009) e até mesmo no prognóstico de câncer (KLOTER *et al.*, 2018).

Hoje em dia existem alguns *softwares* utilizados para esse tipo de análise, como o Kubios HRV (KUBIOS, 2016) e o Nevrokard aHRV (NEVROKARD, 1987). Esses *softwares* foram desenvolvidos para fins comerciais, de utilização em análises hospitalares, sendo muitas vezes confusos, de difícil interpretação e com informações demais. A partir disso, a importância desse trabalho é o desenvolvimento de uma ferramenta de análise de eletrocardiograma que utilize o método de maneira rápida e de fácil entendimento, utilizando linguagem de programação gratuita e de modo open source, para ser usado em pesquisas e modificado de acordo com a necessidade do usuário.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Esse trabalho de conclusão de curso tem por objetivo unir diferentes bibliotecas externas do *Python* em um único *software*, de modo que possa ser feita a análise de sinais de eletrocardiograma com o método de variabilidade do ritmo cardíaco.

1.1.2 Objetivos Específicos

1) Ler e entender a literatura sobre eletrocardiogramas e o método de análise de variabilidade do ritmo cardíaco, juntamente com todos os parâmetros envolvidos.

2) Estudar as bibliotecas externas do *Python* e definir o fluxo de funcionamento do *software*.

3) Implementar o *layout* da janela principal do *software* e interação com o usuário.

4) Implementar os códigos de processamento dos sinais e análise dos dados obtidos.

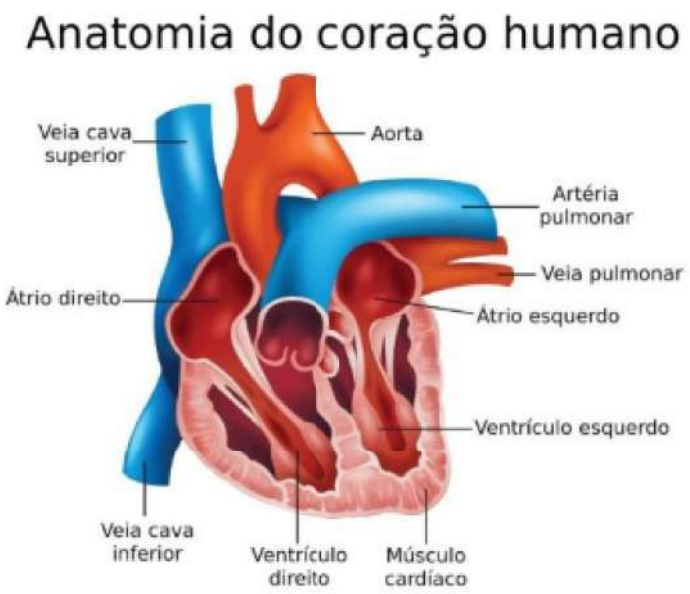
2 FUNDAMENTAÇÃO TEÓRICA

2.1 ANÁLISES BIOLÓGICAS

2.1.1 Coração

O coração é um órgão “oco”, dividido em quatro câmaras – dois átrios (superiores), receptores de sangue, e dois ventrículos (inferiores), ejetores de sangue – responsável por bombear o sangue oxigenado para o corpo e o não-oxigenado para o pulmão. A divisão horizontal das câmaras se dá por duas válvulas: a válvula mitral, que separa o átrio esquerdo do ventrículo esquerdo, e a válvula tricúspide, que separa o átrio direito do ventrículo direito (NOBLE, 2002). A Figura 1 mostra a relação entre cada uma destas estruturas.

Figura 1 – Anatomia do coração humano.



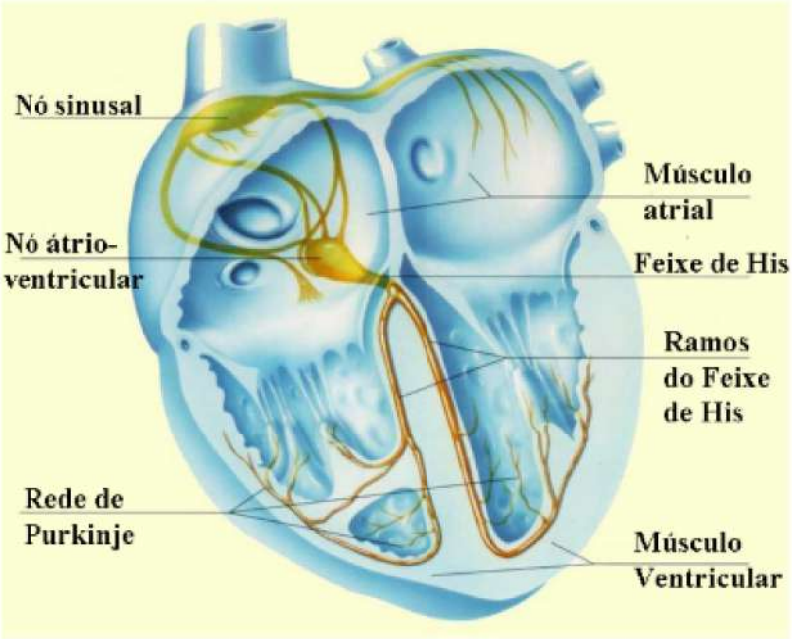
Fonte: Dextro (2015)

Para que haja um eficaz bombeamento de sangue, é necessária uma sincronia entre a contração e o relaxamento muscular nas câmaras cardíacas. Essa sincronia se deve ao nó sinusal, ou sinoatrial, localizado no átrio direito, que envia sinais elétricos para a rede elétrica do coração (NOBLE, 2002). Essa rede é composta pelo nó sinoatrial, o nó atrioventricular, o feixe de His e a rede de Purkinje, conforme indicado na Figura 2.

O sinal eletroquímico produzido pelo nó sinoatrial se propaga até o nó atrioventricular por meio das vias internodais. Em seguida, o sinal é dividido nos feixes de His direito e esquerdo, que por sua vez se dividem em feixes posterior e anterior. Cada um desses ramos principais se ramifica dentro do músculo cardíaco no que se denomina

rede de Purkinje, nos ventrículos. Quando o impulso elétrico atinge o feixe de His e a rede de Purkinje, há a contração do músculo ventricular, causando a ejeção do sangue pelas vias superiores.

Figura 2 – Rede elétrica do coração.



Fonte: O. F. d. Souza (2018)

Cada uma das etapas descritas anteriormente pode ser observada com o auxílio de um eletrocardiógrafo, aparelho responsável por captar esses sinais elétricos em um exame denominado Eletrocardiograma (ECG).

2.1.2 Sinal de ECG

O eletrocardiograma é um dos procedimentos mais utilizado para diagnóstico de problemas cardíacos, pois é simples, indolor, seguro e acima de tudo não-invasivo.

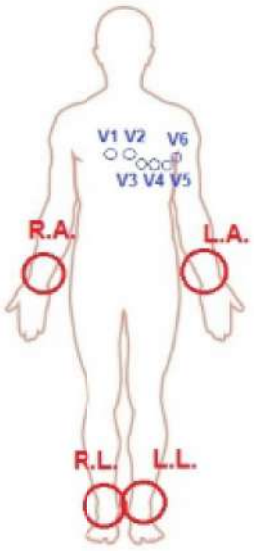
O eletrocardiógrafo capta os sinais elétricos do coração através de eletrodos colocados em lugares específicos na pele do paciente, de modo a conseguir medir a diferença de potencial entre os pontos.

Um exame de eletrocardiograma clínico comum utiliza um total de 10 eletrodos para medir 12 diferentes derivações dos sinais elétricos cardíacos (RAMOS; B. S. SOUZA, 2007). Quatro dos eletrodos (indicados na Figura 3) são posicionados nas duas pernas (L.L. perna esquerda e R.L. perna direita) e nos dois braços (L.A. braço esquerdo e R.A. braço direito) do paciente, de modo a conseguir as derivações bipolares, ou frontais, do ECG.

A derivação I é a diferença de potencial entre o braço direito e o esquerdo, enquanto a derivação II é a diferença entre o braço direito e a perna esquerda e a III

entre o braço esquerdo e a perna esquerda. Nesse caso, a perna direita é utilizada como referência, sinalizando o aterramento, ou a realimentação do sistema.

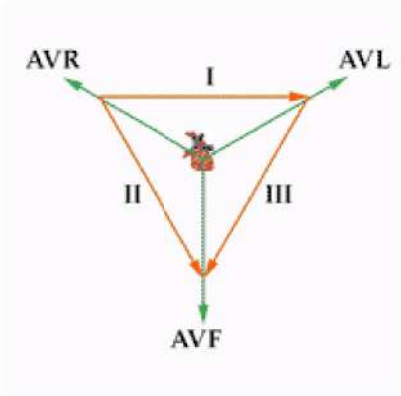
Figura 3 – Posição dos Eletrodos no ECG.



Fonte: Autor (2019)

As derivações bipolares são responsáveis pela construção do triângulo de Eithoven, mostrado na Figura 4.

Figura 4 – Triângulo de Eithoven.

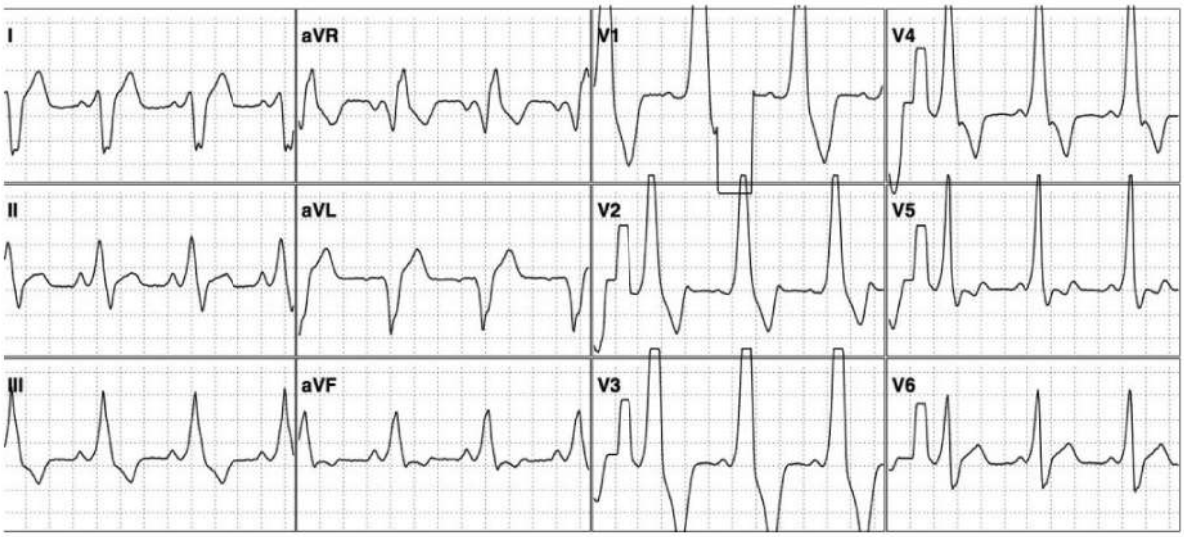


Fonte: EKG (2019)

Já aVR, aVL e aVF são denominadas de derivações unipolares aumentadas. As abreviações vêm do inglês e significam amplified *Vector Right*, *Left* e *Foot*, indicando o sentido dos vetores que saem do triângulo.

Os eletrodos posicionados no peito do paciente formam as derivações precordiais, e são responsáveis por mostrar a diferença de potencial entre o eletrodo e a referência, ou seja, eles medem a tensão elétrica no ponto onde o eletrodo está localizado. Com base nessas diferenças de potencial, pode-se traçar as ondas do sinal de eletrocardiograma (ver Figura 3). Na Figura 5 são mostrados todos os sinais das derivações mencionadas anteriormente. Ao observá-lo, pode-se concluir que as ondas das diferentes derivações apresentam variações em formato, por indicarem vetores de tensão distintos.

Figura 5 – Eletrocardiograma completo.



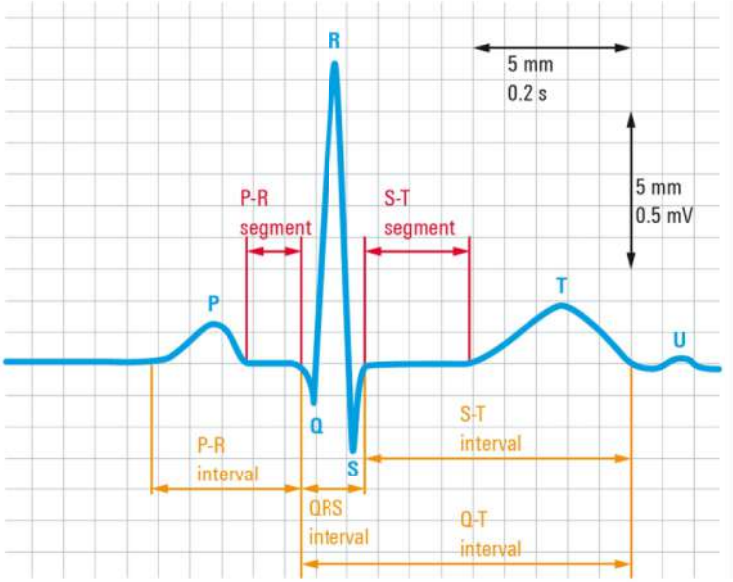
Fonte: Alves (2017)

As análises feitas nesses sinais seguem um modelo de onda cardíaca simplificado, que indica as etapas do movimento cardíaco. O modelo de onda é apresentado na Figura 6 e será utilizado para explicar os conceitos utilizados na análise de variabilidade de frequência cardíaca.

Segundo Alves (2017), pode-se separar o sinal de ECG em três partes principais: a onda P, o complexo QRS e a onda T. A onda P indica o impulso elétrico dado pelo nó sinoatrial e a despolarização dos átrios. A despolarização é a entrada de íons de cálcio nas células do músculo cardíaco e a saída de íons de potássio. O segmento P-R indica a condução elétrica do sinal do nó sinoatrial ao nó atrioventricular. O complexo QRS indica a despolarização ventricular causada pela saída do sinal elétrico do nó atrioventricular, sendo a onda R a que representa a despolarização da maior parte da massa dos ventrículos. A onda R será importante quando for abordado o conceito de variabilidade de frequência cardíaca, pois é com a diferença nos intervalos de tempo entre essas ondas que ocorrem os cálculos necessários para a análise. A repolarização atrial acontece nesse período também, onde os íons de potássio voltam às células

musculares dos átrios. O sangue é bombeado pelos ventrículos no segmento S-T e a onda T indica a repolarização dos ventrículos.

Figura 6 – Eletrocardiograma.



Fonte: Schwarz (2019)

Segundo Morsch (2018), uma frequência cardíaca normal em adultos apresenta uma faixa entre 50 e 100 bpm (batimentos por minuto), ou seja, algo em torno de 0,8 Hz. A onda P tem cerca de 0,1 segundo de duração e tensão na casa de 0,15 mV. O intervalo P-R é de 0,12 a 0,2 segundo. O complexo QRS tem duração de cerca de 0,06 segundo e amplitude de 1,2 mV. O segmento S-T tem duração de aproximadamente 0,12 segundo e a onda T de 0,2 segundo com amplitude próxima a 0,25 mV.

2.1.3 Sistema nervoso

O sistema nervoso representa o controle interno do corpo humano, sendo formado por um conjunto de nervos e células denominadas neurônios. A comunicação entre os neurônios se dá por meio de impulsos elétricos, as sinapses.

Esse sistema pode ser dividido em duas partes, o Sistema Nervoso Central (SNC) e o Sistema Nervoso Priférico (SNP). O SNC é composto pelo cérebro, cerebelo, tronco encefálico e medula espinal, formando o conjunto responsável pelos pensamentos, memórias e ações do corpo humano. Já o SNP é responsável por conectar os nervos marginais a uma das estruturas do SNC.

O sistema nervoso periférico ainda pode ser dividido em dois, o Sistema Nervoso Somático e o SNA. O somático é responsável pelas ações voluntárias, enquanto o autônomo regula as ações involuntárias, como a respiração e o batimento cardíaco. Ainda dentro do sistema autônomo, podemos classificar as ações em dois subsistemas,

o sistema nervoso simpático e o parassimpático. O simpático tem a função de estimular os órgãos, ou seja, faz a aceleração dos batimentos cardíacos, enquanto o parassimpático faz o contrário, desacelerando o coração. Os neurônios cardíacos são os únicos neurônios sujeitos à influência de ambos os sistemas simpático e parassimpático do SNA (ARAUJO; LAUKKANEN, 2009).

2.2 VARIABILIDADE DE FREQUÊNCIA CARDÍACA

2.2.1 História

John e Beatrice Lacey são conhecidos como os percursores da base de estudos utilizada em variabilidade de frequência cardíaca atualmente. O casal recebeu um prêmio da *American Psychological Association* por seus trabalhos feitos nos anos 70 em identificar a comunicação existente entre o coração e o cérebro, pelas vias neurais, responsável por inibir ou facilitar a atividade elétrica desses órgãos (B. C. LACEY; J. I. LACEY, 1978). Suas pesquisas em psicofisiologia mostraram que há um *feedback* interno entre o sistema nervoso autônomo e o sistema cardíaco, como por exemplo, quando a frequência cardíaca está desacelerada, o tempo de reação do indivíduo aumenta, ou seja, ele pensa mais rápido com o coração trabalhando mais devagar.

Esses estudos continuaram a evoluir e, em 1974, os pesquisadores franceses Gahery e Vigier examinaram a influência do nervo vago – responsável pela transmissão de sinais elétricos entre o coração e o cérebro – na resposta elétrica do cérebro (GAHERY; VIGIER, 1974). Para isso, eles estimularam o nervo vago em gatos e perceberam que a resposta elétrica do cérebro era reduzida em aproximadamente metade da atividade sem o estímulo do nervo. Desse modo, ficou provado que o nervo vago tem ação no sistema parassimpático, diminuindo as atividades cerebrais quando estimulado.

Também na década de 70, foi iniciada a pesquisa sobre a diferença dos intervalos R-R a curto prazo nas neuropatias autonômicas em pacientes diabéticos (EWING *et al.*, 1985).

Seguindo as pesquisas, na década de 1990 foram feitos inúmeros avanços relacionados a HRV. Em 1991, o doutor J. Andrew Armour introduziu pela primeira vez o conceito de neurocardiologia, o qual indica que o coração possui um “cérebro” funcional, independente do cérebro craniano (ARMOUR, 1991). Em 1992, o físico Dan Winter quantizou o conceito de coerência cardíaca como sendo o espectro de potência da forma de onda do ECG – conceito utilizado atualmente para os estudos de HRV. Esse conceito levou, ainda na década de 90, a estudos da influência das emoções no espectro de potência do ECG (MCCARTY; ATKINSON; TILLER *et al.*, 1995).

No início dos anos 2000, o desenvolvimento dos estudos de neurocardiologia levou a um maior entendimento dessa rede neural associada ao coração. O sistema

nervoso cardíaco apresenta cerca de 40 mil neurônios e engloba o nodo sinoatrial, o nodo atrioventricular e neuritos acessórios. Esses neuritos podem ser sinais de entrada do sistema autônomo, de nervos simpáticos ou parassimpáticos (MCCARTY; ATKINSON; BRADLEY, 2004).

2.2.2 Método

A variabilidade do ritmo cardíaco é a variação do valor do intervalo entre duas ondas R de um eletrocardiograma (intervalo R-R). Como visto anteriormente (seção 2.1.3), os nervos simpáticos estimulam o coração, diminuindo o intervalo R-R, enquanto o parassimpático aumenta esse intervalo.

Essa variabilidade muda ao decorrer da vida, tanto com a idade quanto com a presença de algumas doenças. Exercícios físicos regulares ajudam a aumentar a HRV, o que faz com que a pessoa se torne mais saudável e menos susceptível a doenças cardíacas, diminuindo inclusive a taxa de morte pós ataque cardíaco (KLEIGER *et al.*, 1987). Fatores como respiração, pressão, temperatura e pensamentos ansiosos são responsáveis por modificar essa variabilidade (HAGE *et al.*, 2017).

O método de HRV pode prover uma variedade de parâmetros, dependendo da aplicação necessária. Esses parâmetros podem ser separados em três grandes grupos: domínio do tempo, domínio da frequência e parâmetros não-lineares.

2.2.2.1 Domínio do tempo

Determina a frequência cardíaca em um momento específico ou intervalos de frequência em uma faixa temporal. Esse método utiliza diretamente a onda adquirida por um eletrocardiógrafo, cujo modelo foi mostrado anteriormente (seção 2.1.2).

O complexo QRS é detectado pelo eletrocardiógrafo, e então o sinal é processado de modo a determinar o intervalo N-N (*normal to normal*), que consiste no intervalo entre dois complexos QRS seguidos. Ao intervalo N-N é dado o nome de frequência cardíaca instantânea.

Esses intervalos sofrem influência de diversos fatores externos ao coração, como a sensação de dia e noite, a respiração, a sensação de perigo e o humor, e fornecem dados importantes a serem analisados. Esses dados são divididos em duas categorias:

- Dados Estatísticos

Um dado estatístico importante a ser adquirido é o desvio padrão dos intervalos N-N, *Standard deviation of the NN interval* (SDNN). Como o desvio padrão muda de acordo com o tamanho do número de amostras utilizadas, é necessária uma padronização do tempo de exame de ECG utilizado, caso queira ser feita uma comparação desses valores para diferentes procedimentos.

O valor do desvio padrão é calculado com a Equação 1, onde N é o número de intervalos R-R, RRi é o valor do intervalo e $\overline{RRi}$ é a média.

$$SDNN = \sqrt{(\frac{1}{N-1}) * \sum_{j=1}^{N} (RR_{ij} - \overline{RR_i})^2} \quad (1)$$

Além do desvio padrão, pode ser calculado o *Root of the mean square of successive differences* (RMSSD) – que é a raiz quadrada da média do quadrado das diferenças consecutivas entre intervalos N-N, um dos métodos que melhor analisa a atividade do sistema parassimpático do SNA, fortemente utilizado em estudos sobre epilepsia (DEGIORGIO *et al.*, 2010). Outros dados estatísticos importante são o NN50, que é o número de intervalos maiores que 50ms seguidos, e o pNN50 que é a proporção de NN50 dividida pelo número total de intervalor N-N. O RMSSD e o pNN50 estão relacionados de modo direto, ou seja, quanto maior o RMSSD, maior o pNN50.

O RMSSD é calculado utilizando a Equação 2.

$$RMSSD = \sqrt{(\frac{1}{N-1}) * \sum_{j=1}^{N-1} (\overline{RRi_{j+1} - Ri_j})^2} \quad (2)$$

O SDNN está relacionado às atividades simpática e parassimpática de longa duração. Já o RMSSD e o pNN50 estão relacionados à atividade parassimpática e podem ser coletados em períodos menores de tempo.

Além disso, podem ser calculadas algumas relações entre os parâmetros estatísticos, como o *Coefficient of variation of the standard deviation* (CVSD) e o *Coefficient of variation* (CVNNi), que são calculados respectivamente como ($RMSSD$/média) e ($SDNN$/média).

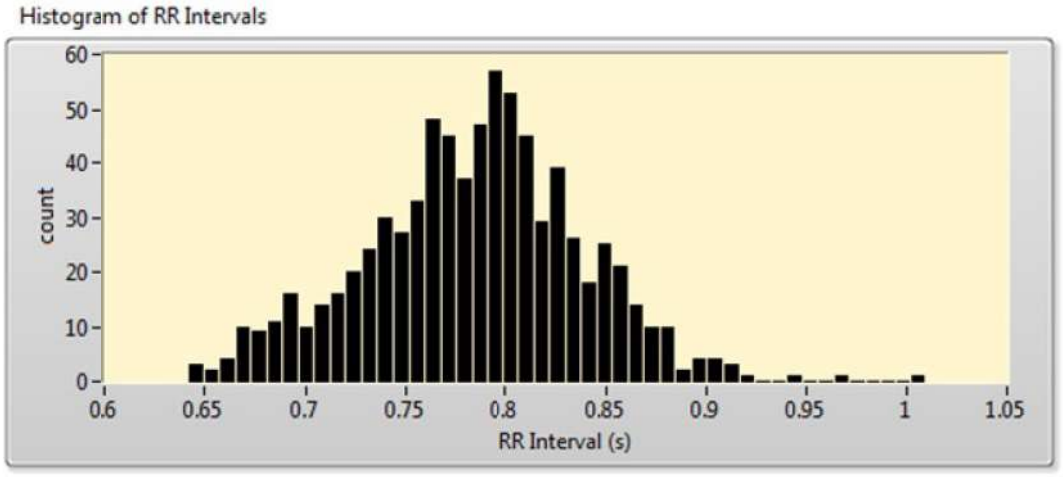
- Dados Geométricos

As análises geométricas são feitas utilizando um padrão gráfico dos intervalos (Figura 7). A partir da análise gráfica, pode-se obter dados importantes utilizando formulações simples. Essas formulações podem ser feitas através da largura da janela de distribuição dos dados, pela interpolação do gráfico com alguma figura geométrica ou com a aproximação do gráfico com gráficos padrões pré-estabelecidos.

O método geométrico mais utilizados é o índice triangular. Ele pode ser calculado utilizando a integral da densidade de distribuição dividida pelo máximo dessa densidade. Somam-se todos os valores de intervalos N-N discretos e divide-se pela altura máxima da curva da densidade de distribuição.

A densidade de distribuição é construída utilizando os valores discretos dos intervalos N-N e arranjando-os em uma curva que mostra o valor do intervalo e o número de vezes que esse intervalo foi observado.

Figura 7 – Distribuição dos intervalos.



Fonte: Roy e Ghatak (2013)

Essa função de densidade de distribuição também pode ser interpolada com um triângulo, chamada de *Triangular Interpolation of NN* (TINN). A base do triângulo é aproximada pela largura da função e o ponto de máximo à altura do triângulo.

2.2.2.2 Domínio da Frequência

As análises feitas no domínio da frequência consistem na construção da função de densidade espectral, *Power Spectral Density* (PSD), da onda de ECG, utilizando a transformada rápida de Fourier (*Fast Fourier Transform* (FFT)) como método não-paramétrico e o modelo autorregressivo como método paramétrico.

Os métodos paramétricos e não-paramétricos têm suas vantagens. A FFT é de simples implementação e alta velocidade de processamento de dados, enquanto os métodos paramétricos apresentam uma melhor separação das frequências, o que facilita no pós-processamento e na separação de cada banda. A desvantagem desse método é a necessidade de adequar os modelos à complexidade do uso necessário.

A *Task Force* da *European Society of Cardiology* e da *North American Society of Pacing and Electrophysiology* (CAMM; MALIK, 1996) estabeleceu em 1996 quatro faixas de frequências encontradas na análise espectral cardíaca, classificadas como:

- **High Frequency** (0.15 - 0.4 Hz): componente ligada à atividade do nervo vago no coração (sistema parassimpático).

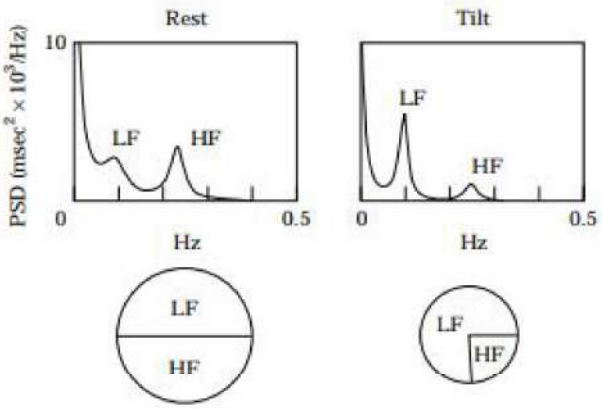
- **Low Frequency** (0.04 - 0.15 Hz): conjunto das atividades do nervo vago e do sistema simpático, com predominância do sistema simpático.

- **Very Low Frequency** (0.0033 - 0.04 Hz) e **Ultra Low Frequency** (<0.0033 Hz): não associadas a nenhuma atividade psicofisiológica.

O método de obtenção dos parâmetros no domínio da frequência é utilizado para medir a densidade de potência de cada uma dessas faixas de frequência, por meio da PSD. Com a curva de densidade de frequência construída, são calculadas as integrais, ou as áreas de cada uma das curvas para a obtenção das potências de cada uma das faixas.

A Figura 8 mostra os resultados da análise no domínio da frequência de um sinal de ECG de um paciente enquanto ele estava deitado (esquerda) e no momento que ele levantou (direita). A partir da análise da imagem podemos aferir que o sistema apresenta uma quantidade de LF e HF balanceada no caso do paciente em descanso e que após levantar, a atividade HF (do sistema parassimpático) diminui (CAMM; MALIK, 1996).

Figura 8 – Análise no domínio da frequência.



Fonte: Camm e Malik (1996)

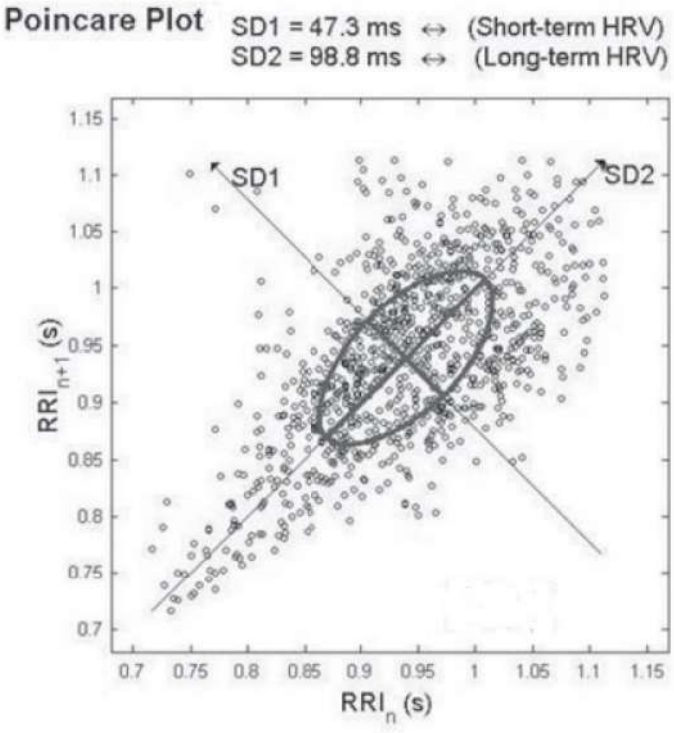
Além das potências dos sinais LF e HF, podem ser calculadas a razão entre as duas grandezas - importante para a análise feita acima, de atividade dos sistemas simpático e parassimpático- e também a potência total do sinal, somando todas as faixas de frequência.

2.2.2.3 Parâmetros não-lineares

Os métodos não-lineares estudados foram o gráfico de Poincaré e a entropia das amostras.

O gráfico de Poincaré/Lorenz é uma exposição geométrica dinâmica dos intervalos R-R, onde cada intervalo é plotado em um dos eixos cartesianos. O primeiro valor de intervalo R-R e plotado no eixo x, o segundo no eixo y, o terceiro no x, e assim por diante, criando o gráfico apresentado na Figura 9.

Figura 9 – Gráfico de Poincaré/Lorenz.



Fonte: Camm e Malik (1996)

As análises feitas com o gráfico de Poincaré são tanto visuais, vendo a concentração de pontos em áreas específicas do plano cartesiano, quanto quantitativas, observando as retas SD1 e SD2, que são relacionadas a dispersão do HRV de curta e longa duração, respectivamente (VANDERLEI *et al.*, 2009).

A análise visual do gráfico de Poincaré é feita observando a figura formada pelos pontos, descritas por Tulppo *et al.* (1998) como três tipos: cometa, torpedo e complexo parabólico. O primeiro, indica uma maior dispersão dos pontos quanto maior for o intervalo R-R e significa um HRV normal; o segundo, com menor dispersão e sem o aumento do espaçamento entre os pontos com o aumento do intervalo R-R; e o terceiro, com grupos de pontos fora do corpo principal do diagrama.

2.2.3 Requisitos para o HRV

Algumas especificações na obtenção do sinal de eletrocardiograma devem ser observadas, de modo a coletar um sinal em que possa ser aplicado o método de HRV. Para uma boa análise de HRV, é necessária uma boa visualização do complexo QRS, como mostrado anteriormente, então o equipamento utilizado como eletrocardiógrafo deve ter um filtro passa baixas em uma frequência de aproximadamente 200 Hz (valor utilizado em equipamentos comerciais como o ECG 12S-PC da empresa Ecafix e o Car-

dioCare 2000 da Bionet). Também, valores como razão sinal-ruído (*Signal-noise Ratio* (SNR)), razão de rejeição de modo comum (*Common Mode Rejection Ratio* (CMRR)) e largura de banda dos componentes utilizados na construção do equipamento devem ser observados.

Além disso, condições de aplicação do exame são utilizadas para cada método, de modo a diminuir o erro. Para eletrocardiogramas de pequena duração – como já mencionado anteriormente, de 5 a 30 minutos – é preferível a utilização de métodos de HRV no domínio da frequência. Para a padronização da utilização dos métodos, foi escolhido o tempo de 5 minutos de exame com o paciente em repouso para a aplicação (CAMM; MALIK, 1996).

Já o método no domínio do tempo é ideal para os eletrocardiogramas longos – geralmente 24 horas de exame, com pelo menos 18 horas de dados coletados e que incluam o período noturno. Em exames de ECG longos, são observadas as mudanças no comportamento do coração no período em que o paciente está acordado e dormindo, sendo então uma boa opção utilizar os dados de SDNN e RMSSD para fazer as análises.

2.3 CONCEITOS COMPLEMENTARES

2.3.1 Transformada rápida de Fourier

A FFT consiste em uma transformação de um sinal discreto no domínio do tempo em domínio da frequência utilizando relações trigonométricas. A regra utilizada para a transformação é apresentada na equação (3).

$$F(u) = \frac{1}{N} * \sum_{x=0}^{N-1} f(x) * e^{-2j(\pi)ux/N} \tag{3}$$

O valor de N representa o número de amostras total do sinal, f(x) é o valor da amostra que está sendo calculada e F(u) é a função da transformada aplicada naquela amostra.

Mais especificamente nesse trabalho, foi usado método de Welch (CONCEIÇÃO *et al.*, 2016), ou periodograma, para encontrar a PSD. Esse método baseia-se na transformada de Fourier, mas aplicada a trechos do sinal, e calculada a média dos quadrados de cada trecho.

$$P_x[k] = \frac{1}{M} * |\sum_{m=0}^{M-1} X_{STFT}[k,m]|^2, 0 \leq k \leq N-1 \tag{4}$$

Ou seja, são calculadas as transformadas de Fourier para janelas do sinal que está sendo transformado. Essas janelas são sobrepostas, somadas e em seguida retirada a média.

2.3.2 Filtro FIR

Finite Impulse Response (FIR) são filtros digitais com resposta finita ao impulso, como o nome indica. Foi escolhida a utilização desse tipo de filtro para o projeto por eles apresentarem estababilidade garantida e uma baixa oscilação em frequência (FERNANDES, 2017).

Os filtros FIR são caracterizados pela equação (5).

$$y(n) = \sum_{i=0}^{P} h_i x(n-i) \tag{5}$$

Com P sendo a ordem, y(n) a resposta, h_i os coeficientes do filtro e x(n) a amostra a ser filtrada.

2.3.3 Método de Malik

O método utilizado para a eliminação dos batimentos ectópicos é o método de Malik (CAMM; MALIK, 1996), onde é avaliada a variação do valor absoluto entre um ponto e outro com a relação:

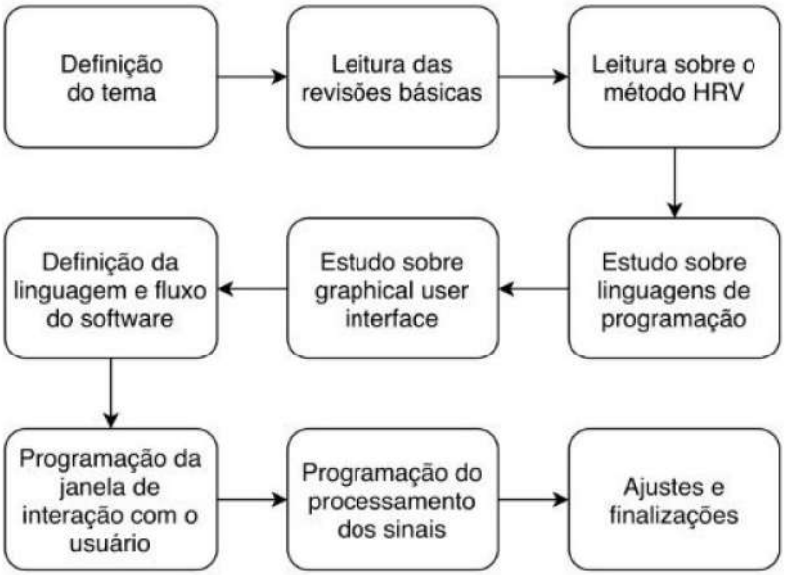
$$outlier = abs(rrinterval - nextrrinterval) <= 0.2 * rrinterval \tag{6}$$

Ou seja, se a diferença entre um intervalo e outro é menor do que 0.2 * o intervalo avaliado, ele é um batimento ectópico.

3 METODOLOGIA

A metodologia adotada para a realização do trabalho segue o fluxo indicado abaixo, na Figura 10.

Figura 10 – Fluxo do trabalho.



Fonte: Autor (2019)

Primeiramente, em reunião com o orientador, foi escolhido o tema do projeto, sendo sugerida a implementação de um *software* que utilizasse o método de variabilidade de frequência cardíaca para a análise de sinal de eletrocardiograma.

A partir desse ponto, foi feita a pesquisa de artigos relacionados ao tema, como os diversos usos do método, os requisitos do *hardware* de coleta dos dados e do *software*, e uma visão geral de como seria montado o fluxo do programa.

Na etapa de estudo sobre as linguagens de programação, foram consideradas duas possíveis linguagens, *Python* e JAVA. Em cima disso, foram levantados os pontos positivos e negativos de cada uma delas: *Python* tem inúmeras bibliotecas que facilitariam a parte de processamento dos sinais e os cálculos necessários, mas uma maior dificuldade na montagem do *layout*; JAVA é uma linguagem mais simples e de fácil compreensão e implementação, mas seria perdida a facilidade do uso de bibliotecas. Foi optado por fazer o trabalho em *Python*, pela familiaridade do autor em relação à linguagem.

Em seguida, com a linguagem escolhida, foram estudados os modos de criação de uma *Graphical User Interface* (GUI) usando *Python*, as bibliotecas e ferramentas que seriam necessárias tanto para a implementação da janela de interação com o

usuário quanto para a programação do processamento e então foi montado o fluxo de funcionamento do *software*, como mostrado no Anexo A.

Podem ser abertos dois tipos de arquivo, com o ECG ou com a onda RR pronta. Caso tenha sido aberto o arquivo contendo o ECG, pode-se plotar o gráfico do sinal. Essa plotagem leva em consideração a amplitude do sinal, geralmente amostrada em milivolts, e o tempo. Para o cálculo do tempo foi levada em consideração a frequência de amostragem do sinal, ou seja, o tempo que aparece no eixo x da plotagem seguiu a seguinte relação:

$$T = \frac{1}{f} \tag{7}$$

Com T igual ao tempo entre amostras e f igual a frequência de amostragem do sinal.

Como as ondas do ECG, cuja frequência varia de 0,05 a 100Hz (seção 2.1.2), sofrem muita interferência da rede (no caso do Brasil, 60Hz), é necessário eliminar a ação dessa interferência e de suas duas primeiras harmônicas - 60, 120 e 180Hz. Além disso, para eliminar o nível DC, é necessário filtrar a componente de frequência 0Hz.

Como comentado anteriormente, foi escolhida a utilização de um filtro FIR. Devem ser então aplicados quatro filtros FIR ao sinal, um passa faixa com frequências de 10 e 40Hz, e três rejeita faixas com frequências de 58 a 62Hz, 118 a 122Hz e 178 a 182Hz, para eliminar as componentes de frequência indicadas anteriormente, com ordem escolhida de modo a satisfazer a atenuação do sinal.

Após a filtragem do ECG, pode ser feita a detecção dos picos, correspondentes à onda R de cada ciclo, e o posterior cálculo das distâncias, em milisegundos, entre cada um desses picos. Isso determinará o sinal RR utilizado nas análises seguintes. Caso tenha sido escolhida a abertura do sinal RR no começo do *software*, o fluxo de funcionamento continuará desse ponto.

Para a correção do sinal RR, foram utilizados dois métodos, a retirada de *outliers* e de batimentos ectópicos. Batimentos ectópicos são contrações prematuras que causam uma onda elétrica antes do batimento cardíaco normal (MITCHELL, 2017). Para a retirada dos batimentos ectópicos foi utilizado o método de Malik (seção 2.3.3).

Para a extração dos *outliers*, que podem ser causados por movimento do paciente durante o exame, assim como ruídos de alta amplitude no sistema, foi escolhida a utilização do intervalo interquartil como referência. Esse método utiliza a mediana do sinal e o intervalo de espalhamento dos dados em torno dessa mediana. Para isso, foram calculados os limites superior e inferior dos *outliers* como sendo:

$$LMax = mediana + 1.5 * IQR \tag{8}$$

$$LMin = mediana - 1.5 * IQR \quad (9)$$

E então foram extraídos os pontos que ficassem fora desse intervalo.

Para todos os pontos retirados pelas correções, foi feita uma interpolação para completar as lacunas onde esses pontos ficavam, usando a média dos valores anterior e posterior ao ponto retirado.

Após feitas as correções necessárias no sinal RR, puderam ser calculados os parâmetros das análises temporal, em frequência e não-lineares descritos na seção 2.2.2, assim como a plotagem dos gráficos interessantes à essas análises.

Por último, se faz necessária a organização e exportação dos resultados obtidos pelo *software*, de modo a conseguir uma melhor visualização dos sinais e parâmetros. A princípio, a exportação é feita em dois tipos diferentes de arquivo, .txt de modo simples e .pdf com os gráficos encontrados, no modelo de relatório.

3.1 RECURSOS UTILIZADOS

3.1.1 PyCharm

Para esse projeto foi utilizada a versão gratuita do PyCharm (JETBRAINS, 2017), um IDE (*Integrated Development Envoirement*), ou seja, um ambiente de desenvolvimento integrado, que é um *software* desenvolvido para agrupar ferramentas e características de linguagens de programação, a fim de facilitar a implementação de códigos.

Na Figura 11 é apresentada a tela inicial quando é criado um novo projeto *Python* utilizando o PyCharm. Na sessão *Project* encontrada à esquerda do IDE está localizado o projeto, nomeado na imagem como HRVOficial, juntamente com o local onde o projeto está armazenado. Logo abaixo tem-se a pasta venv, onde estão armazenadas todas as bibliotecas necessárias para a compilação dos códigos do projeto, chamadas de bibliotecas raíz do PyCharm (*library root*).

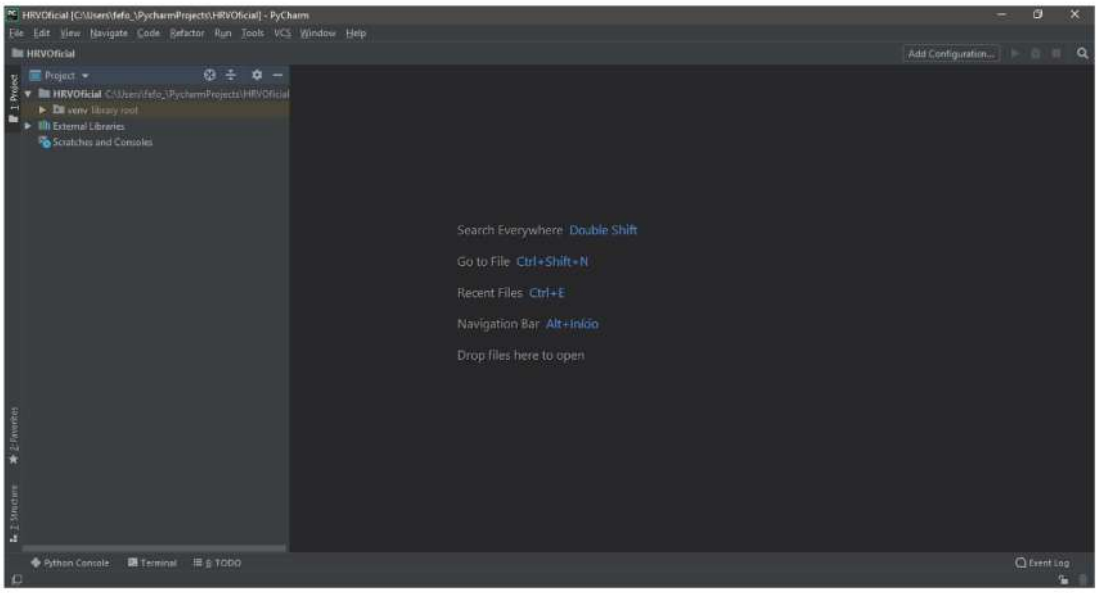
Além disso, por essa sessão podem ser acessadas as bibliotecas externas (*External Libraries*) utilizadas no projeto. As bibliotecas externas são conjuntos de códigos prontos que podem ser utilizados para diversas funcionalidades, e serão melhor explicadas ao decorrer desse capítulo.

Para a criação de um novo *script Python*, clica-se com o botão direito na pasta do projeto, em seguida em *New, Python File*, e, depois de nomear o arquivo, é apresentado um espaço na tela onde pode-se colocar o código, como mostrado na Figura 12.

3.1.2 PyPI

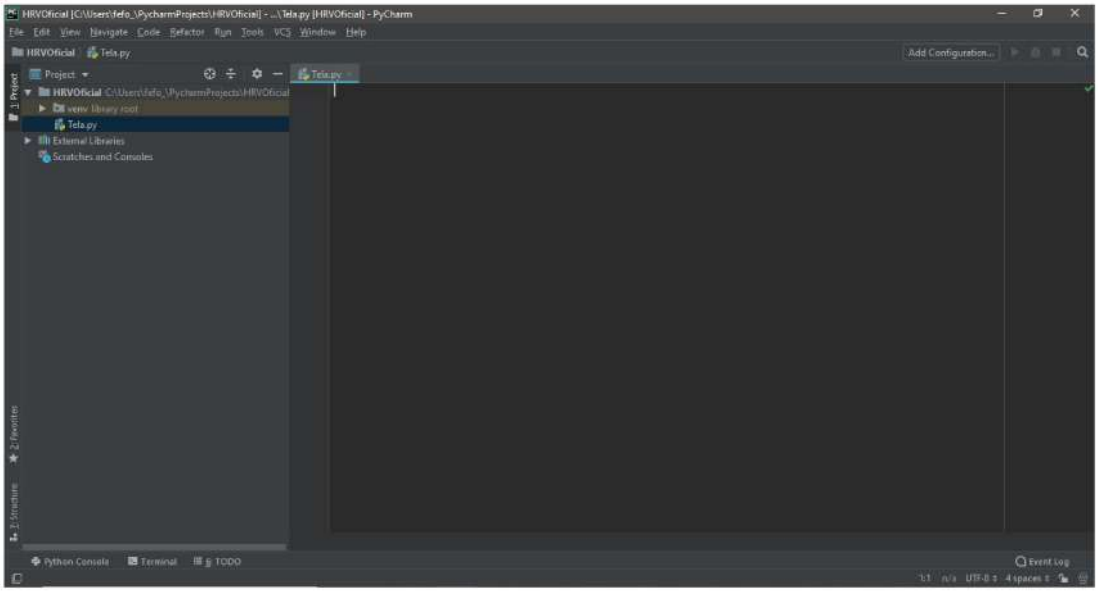
Python Package Index (PyPI) é um catálogo de códigos abertos escritos em Pyhton, com o qual pode ser feito o *upload* e o *download* das bibliotecas externas

Figura 11 – Ambiente do PyCharm.



Fonte: Autor (2019)

Figura 12 – Novo arquivo .py.



Fonte: Autor (2019)

mensionadas anteriormente, códigos prontos para as mais diversas aplicações na linguagem (PYPI. . ., s.d.).

Para instalar um *package* (pacote, ou biblioteca) utilizando o PyPI é usado o Terminal do PyCharm, apresentado abaixo (Figura 13), com o comando *pip install <nome da biblioteca>*. Por exemplo, para a instalação da biblioteca *NumPy*, digitou-se

"pip install NumPy" e o programa faz o *download* e instalação do pacote no projeto. Esse método foi utilizado para a instalação de todas as bibliotecas externas.

Figura 13 – PYPI

Fonte: Autor (2019)

3.1.3 Bibliotecas Externas

3.1.3.1 NumPy

NumPy é um pacote essencial para programação científica, que conta com funções de processamento de *arrays* e álgebra. Além disso, pode ser usado para armazenamento, transformando qualquer tipo de dado em *np-arrays*, facilitando a utilização e acoplamento de diversos outros pacotes (NUMPY. . ., s.d.).

3.1.3.2 TKinter

TKinter é uma biblioteca de desenvolvimento de GUI assim como o PyQT5 utilizado nesse projeto. É bastante utilizada em interfaces gráficas que fazem interação com banco de dados (TKINTER. . ., s.d.).

3.1.3.3 MatPlotLib

Essa biblioteca é responsável por plotagens em 2D de gráficos de linha, *scatterplots*, histogramas, espectros de potência, entre outros. É um dos pacotes mais utilizados em plotagem em *Python* (MATPLOTLIB. . ., s.d.).

3.1.3.4 BioSPPy

Biosignal Processing in Python (BioSPPy) é uma *toolbox* utilizada em processamento de sinais biológicos, com diversos módulos de processamento de biossinais, como ECG, EEG, filtragem e análise em frequência. Dentre as funções presentes na biblioteca para análise de ECG, estão filtragem, detecção de picos e cálculo de batimentos cardíacos (BIOSPPY. . ., s.d.).

3.1.3.5 PyHRV e HRV-Analysis

PyHRV (PYHRV..., s.d.) é uma *toolbox* de análise de variabilidade de ritmo cardíaco para *Python*, que utiliza algumas outras bibliotecas mencionadas anteriormente, como *NumPy*, *MatPlotLib* e BioSPPy, e mais algumas extras, como SciPy, Nolds e Spectrum, para fazer as análises de variabilidade de ritmo cardíaco. Além da PyHRV, a HRV-Analysis (HRV-ANALYSIS..., s.d.) também se encaixa no projeto, mas apenas para os cálculos dos parâmetros, ao final do processamento do sinal.

3.1.3.6 FPDF

FPDF é uma biblioteca externa responsável pela criação de arquivos pdf para *Python*. Foi utilizada no projeto na exportação dos resultados numéricos dos cálculos, caso seja escolhida a exportação nesse tipo de arquivo (FPDF..., s.d.).

3.1.3.7 PyQt5 e Qt Designer

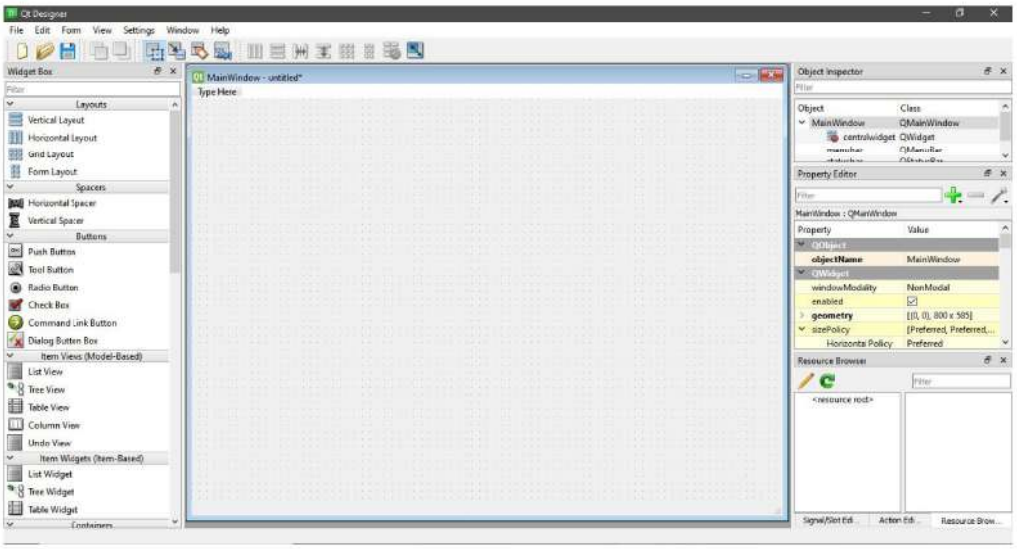
Para a implementação do *layout* do *software*, onde ocorre a interação com o usuário, foi utilizada a biblioteca PyQt5 para *Python*. PyQt5 é um conjunto de bibliotecas linkadas de modo a facilitar a construção de GUIs utilizando linguagem *Python*. A biblioteca foi produzida pela empresa *Riverbank Computing*, utilizando *scripts* e extensões já existentes da *QT Company* (PYQT5..., s.d.).

Mais especificamente para esse projeto, foi utilizada a ferramenta *Qt Designer*, uma *Qt tool* que facilita a construção de GUIs, por apresentar o método de *drag-and-drop*, onde o usuário apenas arrasta os componentes que serão utilizados no *software* para o espaço reservado para a janela principal. Para instalar as bibliotecas necessárias nessa etapa do projeto, foi utilizado o comando *pip install* comentado anteriormente, fazendo o *download* tanto do pyqt5 quanto do pyqt5-tools, onde se encontram as ferramentas extras da biblioteca – incluindo o *Qt Designer*.

Na Figura 14 é mostrada a tela inicial do *Qt Designer*, onde pode-se ver a janela principal do *software* que será construído - *MainWindow*. À esquerda (na seção *Widget Box*) estão todos os componentes para que possam ser arrastados e aplicados à janela, e à direita estão algumas informações úteis no projeto.

Os componentes são divididos em categorias, de acordo com a funcionalidade, como mostrado na Figura 15. *layouts* são as formas de distribuição que os componentes vão apresentar na janela, sendo eles *layout* Horizontal, Vertical, *Grid* ou *Form*. *Spacers* são espaçadores horizontal e vertical, responsáveis por centralizar ou distanciar elementos em um *layout*. *Containers* são espaços que podem ser reservados na janela para futuras aplicações, como um espaço para *Widget* genérico, onde pode-se programar o *Widget* no código de processamento. As demais categorias englobam to-

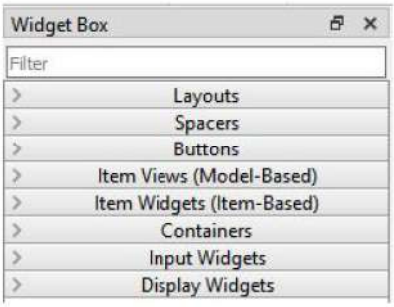
Figura 14 – Qt Designer.



Fonte: Autor (2019)

dos os elementos interativos do *software*, como botões, textos e demais componentes que serão melhor desenvolvidos ao decorrer desse capítulo.

Figura 15 – Menu do Qt Designer.



Fonte: Autor (2019)

A seção *Property Editor* do *Qt Designer* é onde podem ser aplicadas as configurações dos componentes. Todas as configurações necessárias para os componentes do *software* se encontram nessa seção, como largura, altura, política de redimensionamentos, fonte das palavras, tamanho da letra, entre outras. É nessa seção também que pode ser mudado o nome do objeto, em *objectName*, colocando-se o nome da variável desejada, para facilitar a posterior programação da parte de processamento.

3.2 PARÂMETROS COLETADOS

Os parâmetros obtidos pelo *software* são:

Análise temporal:

- Gráficos de RR e NN (imagem)
- Histograma do sinal no tempo (imagem)
- Média dos intervalos NN (intervalos RR normalizados)
- SDNN - desvio padrão entre intervalos NN
- RMSSD – raiz quadrada da média da soma dos quadrados das diferenças entre intervalos adjacentes
- SDSD – desvio padrão entre as diferenças de intervalos adjacentes
- Mediana dos intervalos NN
- NNi50 - número de intervalos acima de 50ms
- pNNi50 – porcentagem dos intervalos acima de 50ms em relação ao total
- NNi20 - número de intervalos acima de 20ms
- pNNi20 – porcentagem dos intervalos acima de 20ms em relação ao total
- *Range* - Diferença entre o maior e menor valor entre NN
- CVSD – coeficiente de variação de diferenças consecutivas (RMSSD/média)
- CVNNi – coeficiente de variação (SDNN/média)
- Média dos batimentos cardíacos
- Batimentos máximos
- Batimentos mínimos
- Desvio padrão dos batimentos
- Parâmetros geométricos - índice triangular (integral da densidade de distribuição dividida pela máxima da função densidade) e TINN (largura da base do triângulo formado pelo histograma de distribuição de intervalos)

Análise em frequência:

- Espectro de frequência (imagem)
- Potência total do sinal
- VLF – *very low frequency* (0.003 a 0.04 Hz)
- LF – *low frequency* (0.04 a 0.15 Hz)
- HF – *high frequency* (0.15 a 0.4 Hz)
- Razão entre LF e HF
- LF normalizado
- HF normalizado

Análise não-linear:

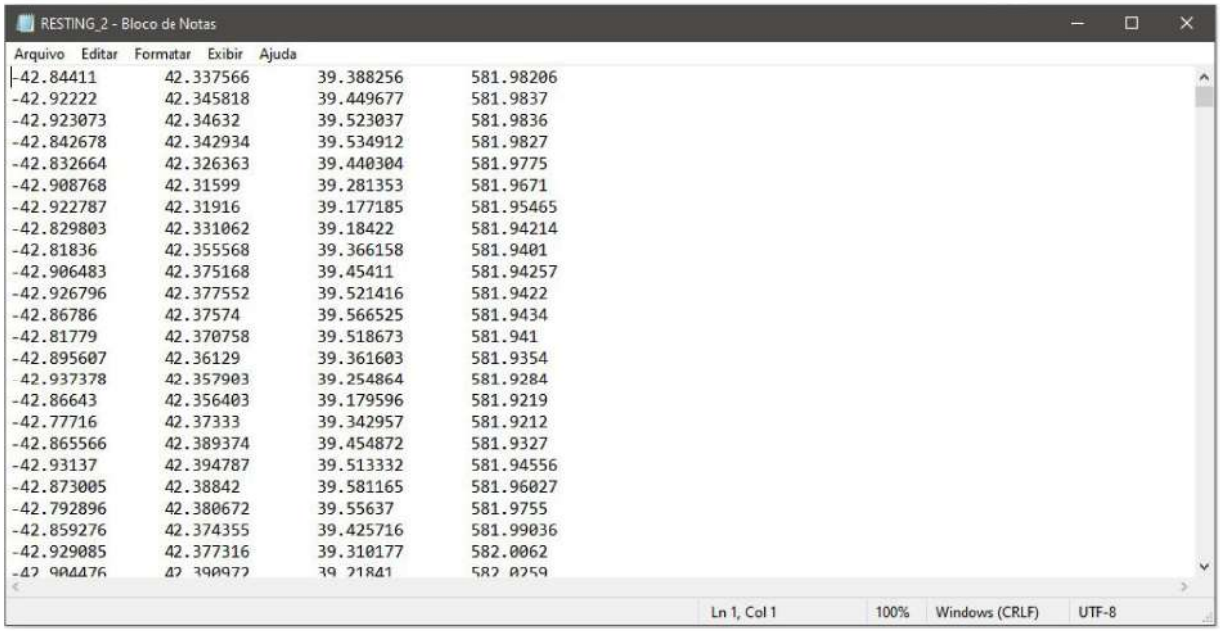
- Plot de Poincaré/Lorenz (imagem)
- SD1 e SD2 – desvio padrão na reta y=x no plot de Poincaré e o desvio padrão na reta perpendicular a ela
- Razão SD2/SD1
- CSI – *cardiac sympathetic index*
- CVI – *cardiac vagal index*
- Entropia do sinal NN

4 DESENVOLVIMENTO

4.1 IMPLEMENTAÇÃO DO CÓDIGO

Primeiramente, analisando os arquivos de teste, é possível observar a necessidade de configurar o arquivo .txt (Figura 16) caso seja requerida a abertura do ECG, pois deve ser apontada a coluna dentro do arquivo onde se encontram os resultados do eletrocardiograma, uma vez que raramente se obtém o sinal de modo isolado. Ao configurar o sinal de ECG também é selecionada a frequência de amostragem do sinal, um parâmetro importante para futuros passos no processamento, como análise no tempo. Caso seja escolhido abrir o arquivo com os intervalos RR, não é necessária essa configuração, visto que a correção dos intervalos RR e as análises não levam em consideração a frequência de amostragem do sinal.

Figura 16 – Arquivo de teste.



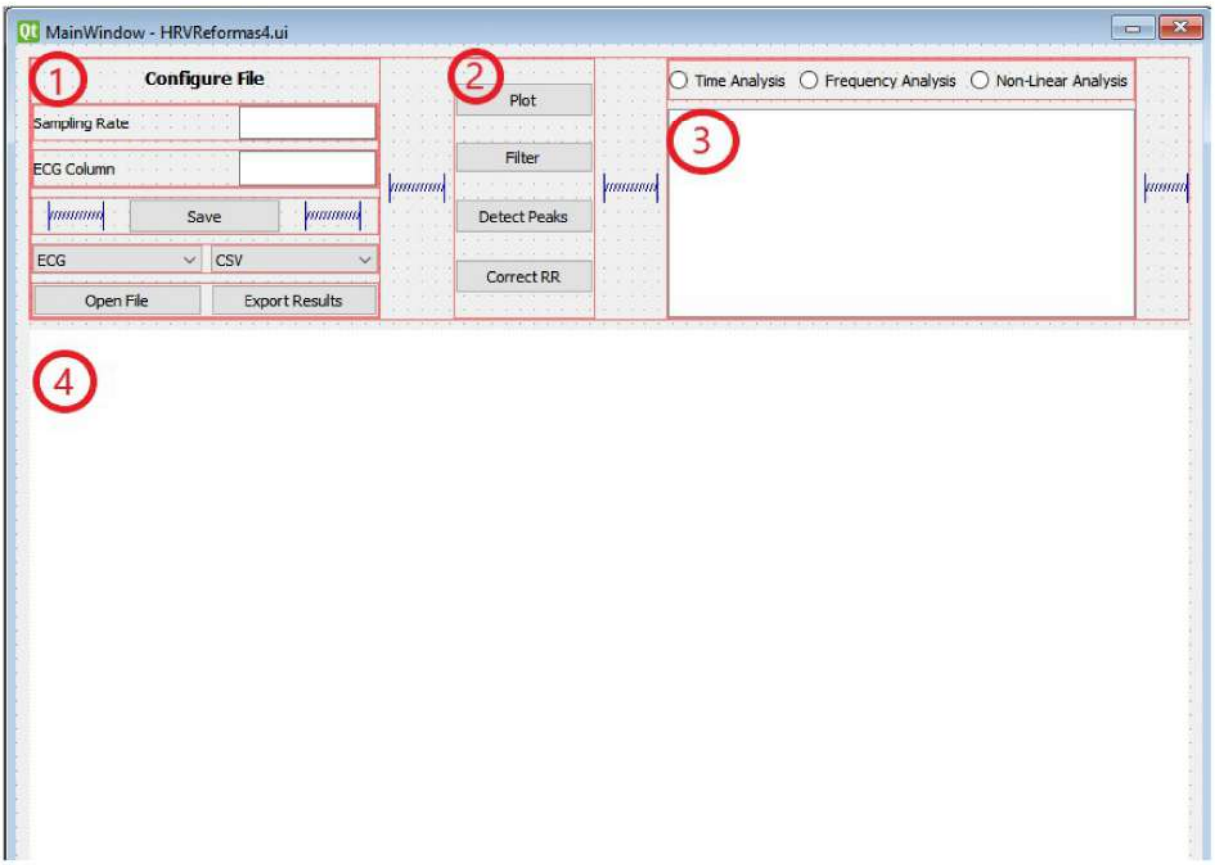
Fonte: Autor (2019)

4.1.1 Construção do *layout* da janela principal

Usando o *drag-and-drop* do *QT Designer*, foram alocados cada um dos componentes do *software* de acordo com a Figura 17. Todas as configurações inseridas no *Qt Designer* podem ser alteradas posteriormente, quando o código gerado for utilizado no PyCharm.

Na imagem pode-se ver quatro áreas principais da GUI, sendo elas:

Figura 17 – Main Window.



Fonte: Autor (2019)

1. Arquivos externos: como mencionado anteriormente, se faz necessária a configuração do arquivo de ECG antes da abertura, sendo essa seção do *software* responsável por essa etapa.

   A frequência de amostragem do sinal, utilizada em vários cálculos, que geralmente tem valores na faixa de 500 a 1000 Hz, pode ser alterada dependendo do sinal, utilizando o primeiro campo de texto dessa seção, o *QTextEdit* ao qual foi dado o nome de **SampRate**. Já o segundo campo é responsável por configurar a coluna do arquivo .txt onde se encontra o sinal de ECG, caso haja mais sinais salvos no mesmo arquivo, como PPG, e o *QTextEdit* foi denominado **Coluna**. Ao clicar no botão *"Save"*, que é um *Push Button*, esses elementos de cálculo são salvos e utilizados nas funções seguintes.

   Em seguida, tem-se a abertura e exportação dos arquivos, com suas respetivas opções de escolha. Primeiro, a abertura do arquivo para análise, com uma *QComboBox* e as opções ECG e RR, para que o usuário escolha que tipo de sinal deseja abrir, seguido de outro *Push Button* para fazer a abertura do arquivo.

Ao lado, a *Combo box* com as opções CSV, TXT ou PDF, para que o usuário escolha em que tipo de arquivo quer exportar os resultados encontrados pelo *software*, também seguido de um *Push Button* para a aplicação da função.

2. Botões: os botões utilizados para a interação do usuário com a interface foram agrupados em ordem de utilização. Para isso foram escolhidos *Push Buttons*. Primeiramente o botão *"Plot"*, responsável por construir o gráfico do sinal de ECG não processado na área designada aos gráficos. O botão *"Filter"* é responsável por fazer a filtragem do sinal, eliminando o nível DC e a ação de ruídos. *"Detect peaks"* será utilizado para detectar os picos do sinal de ECG filtrado, assim como calcular os intervalos RR e plotar o sinal RR na área de gráficos. O último botão, *"Correct RR"*, aplica as correções mencionadas anteriormente, *outliers* e batimentos ectópicos, e mostra a onda corrigida na área dos gráficos.

3. Resultados: nessa área do *software* será mostrada uma prévia dos resultados calculados pelo programa, em forma de tabela. Para migrar entre esses diferentes tipos de dados e observar os resultados na tela principal do *software*, foi escolhida a utilização de *Radio Buttons*, botões de seleção única, ou seja, quando *Time Analysis* está selecionada, os elementos de cálculos temporais aparecerão na tabela localizada logo abaixo dos botões. Caso *Frequency Analysis* for então selecionado, serão mostrados esses elementos e assim por diante. Para a apresentação dos dados foi alocada uma área com um *QTableWidget*, uma *widget* de tabela para o Qt. Ao clicar cada um dos *Radio Buttons*, além dos resultados numéricos mostrados na tabela, também serão abertos os gráficos importantes para essa análise, como o histograma, o espectro de potência e o gráfico de Poincaré/Lorenz.

4. Gráficos: além dos resultados numéricos, os gráficos utilizados durante o processamento também são mostrados na janela principal do *software*. Gráficos de ECG não processado e filtrado são mostrados no espaço de cima, enquanto o gráfico com a onda de intervalos RR e NN é mostrado em baixo. Para colocar as figuras dos gráficos na apresentação da tela principal foi utilizado um *QWidget* genérico, apenas delimitando o espaço onde os gráficos ficariam, e a inserção das imagens nesses espaços se dá ao longo do programa, com a utilização do *MatPlotLib*.

Além dos elementos apresentados nas seções, foram utilizados alguns espaçadores horizontais para melhor aproveitamento do espaço da janela e para questões de estética. Dentro de cada uma das seções também foram utilizados *layouts* horizontais e verticais para que ficassem nas posições necessárias, e, ao final, todas as seções foram alinhadas utilizando o *grid layout*.

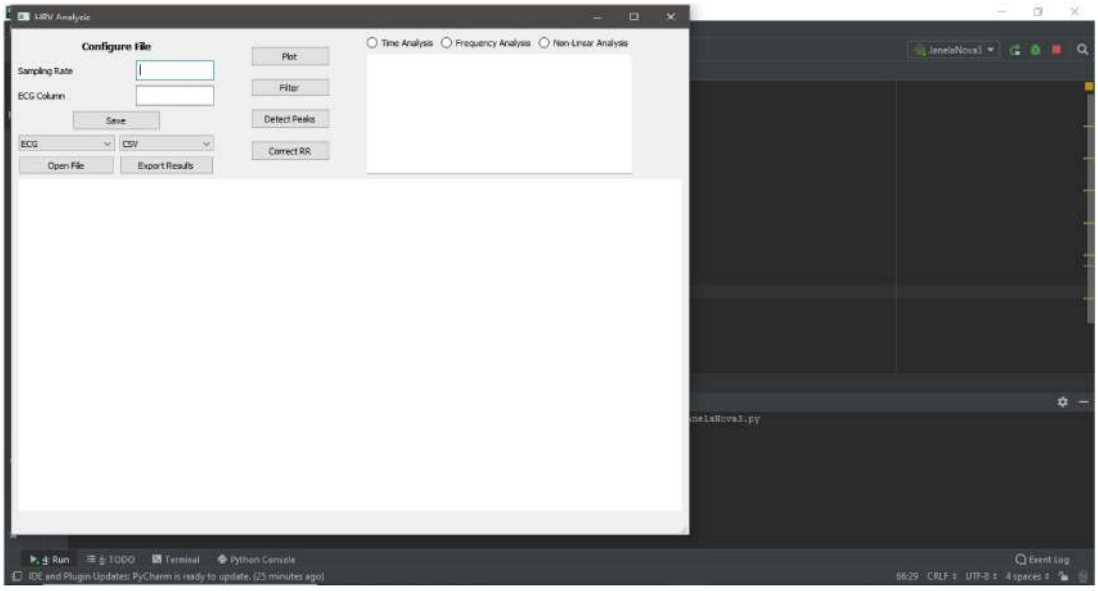
Com a janela principal montada, bastou salvar o projeto em *File, Save as...* e salvar o arquivo .ui que o *Qt Designer* cria com o nome do projeto, por exemplo HRVOficial.ui. Esse arquivo .ui é então traduzido pelo PyCharm em forma de código *Python*.

4.1.2 Código de processamento

Voltando ao PyCharm, para abrir o arquivo .ui salvo do *Qt Designer*, utilizou-se o comando *"pyuic5 –x <nome do arquivo>.ui -o <nome do arquivo>.py"* no Terminal. A extensão pyuic5 é a responsável pela integração do *Qt Designer* com a utilização do PyQt5 no programa, fazendo a tradução do arquivo .ui para *Python*, e os comandos –x de *extract* e –o de *open* são utilizados para extrair o código *Python* do arquivo .ui e salvá-lo em um novo *script* .py no PyCharm.

Após a extração, tem-se o programa em *Python* que gera a janela construída no *QT Designer*. O programa consiste em uma *class* com todos os elementos que compõem a janela, suas posições, tamanhos e distribuição, mas sem as ações necessárias para que o *software* faça alguma coisa. Para fins de teste, bastou executar o arquivo .py criado utilizando o *Run* do PyCharm e ver se a janela construída no *Qt Designer* aparece (Figura 18).

Figura 18 – Main Window construída.



Fonte: Autor (2019)

4.1.2.1 Configure File

Na primeira área apresentada anteriormente, tem-se os campos de configuração do arquivo de ECG. De modo a coletar os parâmetros inseridos pelo usuário de frequência de amostragem e coluna do ECG, foram utilizados *QTextEdits* e *Labels* que indicam o que cada um dos campos representa.

Para salvar os textos inseridos pelo usuário em uma variável, utilizou-se a função **toPlainText( )**, herdada da *class QTextEdit*. Essa função salva o texto escrito no campo em formato *string*. Então essa *string* deve ser transformada em int para que possa ser utilizada nos cálculos, e finalmente armazenada em uma nova variável. Abaixo é mostrado como fica a linha utilizada para essa etapa.

```
self.sampling_rate = int(self.SampRate.toPlainText())
```

O mesmo foi feito para a variável da coluna e, em seguida, ambos os campos são apagados com a função **clear( )**, para que o usuário saiba que os dados foram salvos.

```
self.SampRate.clear()
self.Column.clear()
```

Para que haja o armazenamento, é necessária a utilização do botão *Save* que se encontra logo abaixo dos campos de texto, cuja função **get_values( )** é responsável pelas manipulações citadas acima. O método de aderir uma função a um *Push Button* é o mesmo ao decorrer de todos os botões do *software*, com a seguinte chamada.

```
self.bSave.clicked.connect(self.get_values)
```

Ou seja, quaisquer *Push Buttons* podem ser conectados a uma função utilizando **<nome do botão>.clicked.connect(<nome da função>)**, que será chamada ao clicar no botão. Essas chamadas devem ser inseridas na função *setupUi* criada pelo *Qt Designer*, ao importar o código.

Logo abaixo do botão *Save* tem-se os campos de abertura do arquivo e exportação dos resultados. Por *default*, foi escolhido deixar ECG e PDF como as primeiras opções nas *Combo Boxes*. Ao clicar no botão *Open File*, a função **open_file( )** é chamada e o texto atual das *combo boxes* é gravado em uma variável por meio da função **currentText( )** herdada da *class QComboBox*.

```
self.tipo_de_arquivo = self.comboBox.currentText()
```

Então o texto, nesse caso "ECG", é salvo na variável **self.tipo_de_arquivo** e utilizado na abertura dos arquivos. O programa, por sua vez, faz a distinção do tipo de arquivo a ser aberto utilizando esse texto. Foram usadas funções *if* para cada um dos casos, ECG e RR, e dentro dos *if*s de cada um deles, foi adicionado o código de abertura dos arquivos. No exemplo, com a opção ECG na *Combo Box*, o programa vai para o *if* do ECG.

```
if self.tipo_de_arquivo == 'ECG':
```

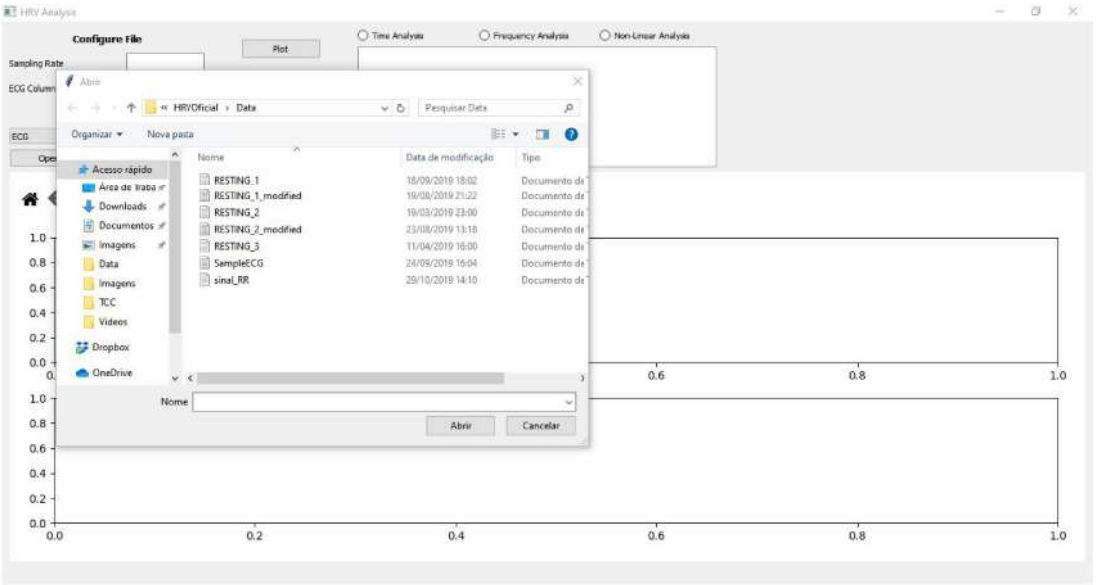
Para que o usuário escolha o arquivo a ser aberto, foi importada a biblioteca

TKinter, e dela foi usada a função **filedialog.askopenfilename( )**, responsável por abrir uma caixa de diálogo onde o usuário pode navegar pela memória do computador em busca do arquivo. Para facilitar a utilização, foi colocada uma pasta inicial de procura, onde se encontram os arquivos contendo os ECG e os RR, resultando na linha a seguir.

```
self.archive = filedialog.askopenfilename(initialdir='Data')
```

O argumento initialdir da função vem de *initial directory*, ou seja, a pasta que é aberta inicialmente ao clicar no botão (Figura 19).

Figura 19 – Abertura do arquivo.



Fonte: Autor (2019)

Depois que o usuário escolhe o arquivo que quer abrir, esse arquivo é alocado à variável **self.archive** e em seguida, utilizando a biblioteca *NumPy* e a função **loadtxt( )**, o conteúdo do arquivo é carregado a uma variável signal, de formato *np.array*.

```
self.signal = np.loadtxt(self.archive, usecols=(self.column - 1))
```

Nesse ponto se faz necessário que o usuário tenha configurado o arquivo antes de abri-lo, pois essa linha utiliza a coluna do ECG escolhida anteriormente, no argumento **usecols**. Foi configurado para que a coluna seja **(self.column - 1)** pois as colunas do *np.array* começam [0], ou seja, se o usuário quer que a terceira coluna seja aberta, no arquivo será a coluna de número [2].

Nesse ponto do programa também a calculado o tempo total do sinal, considerando a frequência de amostragem. Como o arquivo aberto apresenta as amostras coletadas a cada (1/frequência de amostragem) segundos, o tempo total do sinal pode ser calculado simplesmente pegando o tamanho total do *array* e dividindo pela frequência de amostragem. Esse passo é importante para cálculos e plotagens futuras.

self.tempo = ((np.array(range(len(self.signal)))) / self.sampling_rate)

Por exemplo, em um arquivo de teste usado nas análises, tem-se 250.000 amostras, com frequência de amostragem de 500 Hz, então o tempo total do sinal é de 500 segundos.

No caso da troca de arquivo para RR, muda-se a *Combo box* para o texto "RR", que segue o mesmo princípio de funcionamento, mas ao invés de carregar o *np.array* na variável **self.signal**, ele é carregado à variável **self.rpeaks**.

self.rpeaks = np.loadtxt(self.archive)

As funções responsáveis pela exportação dos resultados serão mostradas ao final do processo, na seção 3.3.2.7, seguindo o passo a passo de utilização do *software*.

4.1.2.2 Plot

Seguindo com o esquemático de funcionamento do *software*, tem-se a plotagem do sinal carregado. O objetivo do projeto é que o sinal seja mostrado na tela do *software*, e, segundo pesquisas rápidas em sites de programação, foi considerada a utilização da biblioteca *MatPlotLib* para que isso acontecesse.

Para as plotagens na tela do *software*, foi utilizado o *QWidget* genérico mencionado anteriormente, que, ao chamar a função **plot_signal( )**, conectada ao botão *Plot*, faz uso de uma *class* externa para a construção dos gráficos. Foi criado um segundo arquivo .py com nome mplwidget.py responsável pela contrução do espaço onde ficam os gráficos na janela, onde foram colocados todos os detalhes dos gráficos, como a separação dos *subplots* inferior e superior, as margens e a adição do *toolbar* responsável pelas ferramentas de zoom. O código completo dessa configuração, assim como o código completo do *software*, se encontra na seção de anexos ao final do texto (Anexo D).

Foram utilizadas as funções **clear( ), plot( ), set_title( ), set_xlabel( ), set_ylabel( ) e draw( )** nessa ordem, para que o gráfico seja mostrado na tela do *software*, no espaço superior.

4.1.2.3 Filter

Analisando as bibliotecas pesquisadas que abordavam análise de ECG, foi observado que na biblioteca BioSPPy havia o uso de filtros digitais para o processamento desse tipo de sinal, então foi escolhido utilizar a estrutura dada pela biblioteca. Como comentado anteriormente, essa biblioteca utiliza um filtro FIR (*Finite Impulse Response*), de modos passa faixa e rejeita faixa nas frequências escolhidas.

A função do filtro construída com o BioSPPy, **filter_signal( )** usa como argumentos o sinal de entrada do filtro, nesse caso o sinal de ECG não processado *self.signal*, o tipo do filtro, FIR, se ele é passa baixa, passa alta, passa faixa ou rejeita faixa, a ordem do filtro, que, seguindo recomendação do autor da biblioteca é calculada a par-

tir da frequência de amostragem como sendo 0,3 * sampling_rate, as frequências de passagem do filtro e a frequência de amostragem em si.

Como apontado anteriormente, foram utilizados quatro filtros seguidos, sendo o primeiro mostrado acima, um passa faixa com frequências de 10 a 40 Hz, e três rejeita faixas, para eliminar a interferência da rede de 60 Hz e suas primeiras duas harmônicas. Nos três filtros seguintes, as únicas mudanças feitas foram a troca do argumento *"band"* de *'bandpass'* para *'bandstop'* e a *frequency* de [10, 40] para [58, 62], [118, 122] e [178, 182], resultando no sinal filtrado.

4.1.2.4 Detect Peaks

A biblioteca SciPy apresenta uma função de detecção de picos, denominada **find_peaks( )**, que foi utilizada nesse passo do processamento. Ela encontra os picos de modo simples, por meio de comparação com os valores vizinhos.

```
self.peaks, _ = find_peaks(self.filtered_signal, height=0.3)
```

A função tem como argumentos o sinal de entrada, o qual será analisado para o cálculo dos picos, e a altura mínima necessária para um ponto ser considerado um pico. Como visto na seção 2.1.2, as ondas P e T do eletrocardiograma têm amplitudes médias de 0.15 e 0.25 mV, então basta configurar a função com uma altura mínima de 0.3 mV para que sejam desconsideradas essas ondas, sendo então marcadas as ondas R de cada ciclo.

Após os picos serem detectados, é necessária uma manipulação desses dados para obter a onda RR. A função encontra as posições no eixo x onde se encontram os picos, ou seja, se houver um pico na posição 200, outro na posição 300 e assim por diante, o resultado da função é um array de modo [200,300,400...]. Para encontrar a onda RR é feita a diferença entre essas posições.

```
for i in range(len(self.peaks) - 1):
self.rpeaksList.append(self.peaks[i + 1] - self.peaks[i])
```

Desse modo, é criada uma lista à qual é adicionada a diferença entre duas posições de picos seguidas, durante toda a duração do sinal de entrada. Com essa lista, basta transformá-la de volta em um *np.array* e multiplicar pelo fator de transformação de amostras em tempo comentado anteriormente, de acordo com as linhas seguintes de código.

```
self.rpeaksamostra = np.asarray(self.rpeaksList)
self.rpeaks = self.rpeaksamostra * (1000/self.sampling_rate)
```

Agora na variável **self.rpeaks** tem-se um *np.array* com os valores de tempo em milissegundos de cada um dos intervalos entre duas ondas R seguidas.

4.1.2.5 Correct RR

Algumas correções foram aplicadas ao sinal RR, de modo a evitar os erros comentados na seção 3.1, como batimentos ectópicos e *outliers*. Para essas correções, foi utilizada a biblioteca HRV-Analysis.

Para eliminar os *outliers*, foi usada a função descrita na seção 3.1 como limites superior e inferior, e a função **remove_*outliers*( )** da biblioteca. Essa função compara os valores de RR indicados no sinal com os limites impostos pelo intervalo interquartil, e, caso o ponto esteja fora do limite, ele é subsituído por um valor vazio. Como argumentos da função, foram utilizados os parâmetros calculados de mediana e intervalo interquartil. A mediana foi calculada com a biblioteca *NumPy* com a função **median( )** e o intervalo interquartil com a função **iqr( )** da biblioteca SciPy, na seção *stats*.

```
self.medianaR = np.median(self.rpeaks)
self.rriqr = iqr(self.rpeaks)
self.rpeaksWO=hrvanalysis.remove_outliers (self.rpeaks,
low_rri=(self.medianaR - (1.5*self.rriqr)),
high_rri=(self.medianaR + (1.5*self.rriqr)))
```

Após a remoção dos *outliers*, é feita a interpolação dos pontos retirados, usando a média entre o ponto anterior e posterior. Para isso foi utilizada a função **interpolatenan_values( )** do HRV-Analysis.

Com os *outliers* eliminado e interpolados, é feita a retirada dos batimentos ectópicos com a regra de Malik descrita na seção 2.1.3, e então uma nova interpolação dos valores retirados.

```
self.rpeaksWE = hrvanalysis.removeectopic_beats(self.interpolRWO,
method="malik")
```

Com a onda RR corrigida, são aplicadas as análises de HRV mencionadas na seção 2.2.2.

4.1.2.6 Cálculo dos parâmetros

Para calcular os parâmetros de tempo, frequência e não-lineares correspondentes a onda RR corrigida, foram utilizadas as funções da biblioteca HRV-Analysis **get_time_domain_features( ), get_geometrical_features( ), get_frequency_domain_features( ), get_poincare_plot_features( ), get_csi_cvi_features( ) e get_sampen( )**. Essas funções retornam um *dictionary* com as *keys* sendo os nomes dos parâmetros para cada uma das análises e os values sendo os resultados dos cálculos.

As funções **get_time_domain_features( )** e **get_geometrical_features( )** fazem uso da biblioteca *NumPy* para cálculos estatísticos, como média, mediana, desvio padrão, valor absoluto, histograma e diferença entre os intervalos, enquanto usa funções nativas do *Python*, como *max, min, sum* e *len* para complementar os cálculos.

A função **get_frequency_domains( )** utiliza o método de Welch mencionado na seção 2.1.3 para o cálculo dos parâmetros, por meio da biblioteca *NumPy* e das funções **trapz** e **logical_and**.

Os parâmetros não-lineares são obtidos com **get_poincare_plot_features( )** usando a biblioteca *NumPy* e as funções *diff, std,* e *sqrt* para calcular SD1 e SD2, que são passados para a função **get_csi_cvi_features( )**. Além disso, é utilizada a biblioteca *Nolds* para o cálculo da entropia feito pela função **get_sampen( )**.

Esses resultados são então mostrados na tela do *software* por meio de uma tabela, *QTableWidget*, e alternados entre os três diferentes tipos de análise com a utilização de *Radio Buttons*. Para isso, foi implementada uma cadeia de *if*s com a função **isChecked( )** herdada da *class* de *Radio Buttons*, para averiguar qual dos botões está selecionada.

A inserção dos dados na tabel se dá pelo seguinte método.

```
self.Tabela.clear()
self.Tabela.setItem(0, 0, QTableWidgetItem('Mean NN'))
self.media_nn = round(self.resultsTime.get("mean_nni", ), 2)
self.Tabela.setItem(0, 1, QTableWidgetItem(str(self.media_nn)))
self.Tabela.setItem(0, 2, QTableWidgetItem('ms'))
```

Primeiro, é limpada a tabela, apagando os dados colocados nela anteriormente, se existir algum. Em seguida é declarado o nome da variável, e colocado na primeira coluna, como *'Mean NN'* no exemplo. Como os dados retornados pelas funções do HRV-Analysis são no formato *dict*, pode-se pegar os valores separados de cada um dos parâmetros, que são arredondados para ficarem com apenas duas casas decimais antes de serem mostrados na tabela. Esses valores são então alocados à segunda coluna, enquanto na terceira se encontram as unidades de medida.

Além dos dados numéricos, para cada uma das análises é plotado um gráfico que auxilia tanto na visualização dos dados, quanto em outros tipos de análise. Na análise temporal é plotado o histograma da onda RR, usado nos cálculos geométricos, a partir da função **hist** do *MatPlotLib*.

Já na análise em frequência, é utilizada a função **plot_psd** do HRV-Analysis para plotar a função de densidade do espectro de potência. E por fim, na análise não-linear, é plotado o gráfico de Poincaré/Lorenz, de onde são calculados SD1 e SD2.

4.1.2.7 Exportação dos resultados

Foi utilizada uma *Combo box*, como mostrado anteriormente para a abertura dos arquivos, de modo que o usuário consiga escolher em que tipo de arquivo ele quer exportar os resultados numéricos obtidos pelo *software*. As opções de exportação são em arquivo txt, PDF e CSV, mas foram apenas desenvolvidas as duas primeiras no âmbito desse trabalho, deixando a exportação do arquivo em CSV para os trabalhos

futuros. Além disso, são exportadas as imagens principais relacionadas com a análise, como a onda RR e a RR corrigida, o histograma, o gráfico de densidade de potência e o gráfico de Poincaré/Lorenz.

Para organizar esses arquivos exportados, quando um arquivo ECG ou RR é carregado ao *software*, o programa cria uma pasta com o nome do arquivo aberto, onde são alocados todos os resultados obtidos na análise mencionados anteriormente. Para isso foi utilizada o código a seguir.

```
self.name = self.archive.split("/")[-1]
if not os.path.exists(self.name + '/'):
os.makedirs(self.name + '/')
```

Como a abertura do arquivo se dá pela variável **self.archive** e é no formato **"C://<pasta>/<arquivo>"** pode-se facilmente conseguir o nome do arquivo separando os elementos a cada barra "/" e pegando o último. O resultado dessa linha é o nome do arquivo aberto, salvo na variável **self.name**. Em seguida o código inspeciona para ver se já existe uma pasta com o nome do arquivo aberto e, caso não exista, cria essa pasta.

Em seguida, são salvas as figuras mencionadas anteriormente dentro dessa pasta, por meio da função **savefig** do *MatPlotLib*, por meio da linha abaixo, que é acrescentada após a plotagem dos sinais, apenas trocando o nome do arquivo png criado para cada um dos gráficos.

```
plt.savefig(self.name + '/' + 'RR_signal.png')
```

Agora utilizando a mesma lógica explicada anteriormente para a *Combo box* da abertura dos arquivos, escolhe-se se os dados numéricos serão exportados em txt ou PDF. Para o caso do txt, foi implementado um código que faz a exportação direta dos dicionários com os resultados. Esse método é recomendado para os usuários que entendem do programa e sabem o que é cada uma das siglas dos valores apresentados. Para isso foram utilizadas funções nativas do *Python*, como *open* e *print*.

No caso do PDF, foi usada a biblioteca FPDF, com as funções **add_page, set_font** e **cell** para fazer a inserção dos dados no arquivo PDF. Nesse caso, foram colocados os nomes completos das variáveis, para que um usuário que não tenha conhecimento do programa possa avaliar os resultados numéricos de maneira fácil. Então, foram colocados os nomes das variáveis como *strings*, seguidos dos valores e unidades em uma célula com a função *cell*, e definidas a largura (*width*) e altura(*height*) das células, assim como as bordas(*border*) e se é necessário ir para a linha seguir (ln), como pode ser visto na linha de código a seguir.

```
pdf.cell(width= 100, heigth= 10, txt='Mean NN intervals: ' + str(self.media_nn) + '
ms', ln=1, border=1)
```

Esse processo foi repetido para todas as variáveis obtidas na análise.

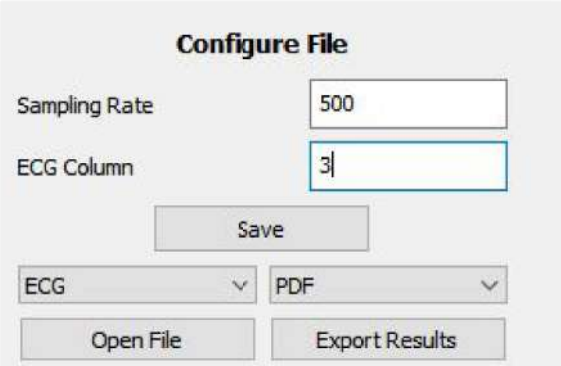
5 RESULTADOS

5.1 TESTE COMPLETO

Para aferir o funcionamento do *software*, foi feito um teste completo utilizando um arquivo de ECG, coletado a 500 Hz, durante aproximadamente 8 minutos. Todas as imagens e resultados numéricos obtidos pelo *software* são mostrados a seguir. Para isso, foi criada uma pasta chamada *Data* dentro da pasta do projeto do *software*, onde foram colocados os arquivos ECG e o arquivo RR de teste.

Primeiramente foi feita a abertura do arquivo, como indicado no capítulo anterior, utilizando a primeira área do *software*. Foi colocado no campo *Configure File* a frequência de amostragem do sinal (**Sampling Rate = 500**) e a coluna onde se encontrava o ECG dentro do arquivo .txt (**ECG Column = 3**), como mostrado na Figura 20. Isso faz com que as variáveis ligadas a essas caixas de texto (**self.sampRate** e **self.column**) sejam salvas, ao clicar no botão *Save*. Os dados são apagados da caixa de texto para mostrar que foram salvos.

Figura 20 – Configure File.



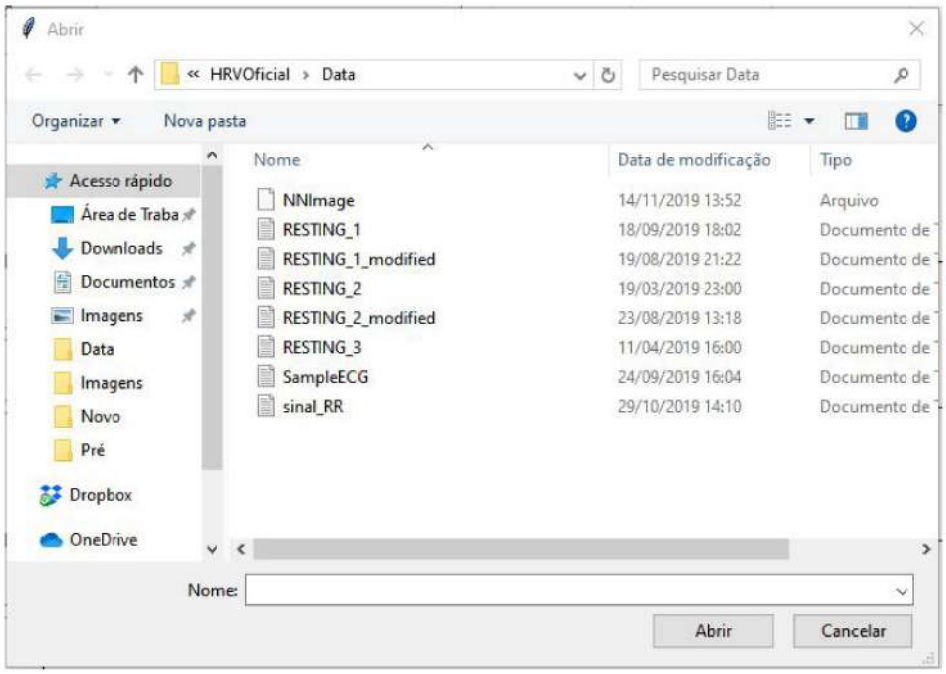
Fonte: Autor (2019)

Em seguida, com a *Combo box* selecionada em ECG, foi clicado o botão *Open File*, para que a janela de interação com a memória do computador fosse aberta, como mostrado na Figura 21, onde é selecionado o arquivo **RESTING_1.txt**, que é salvo na variável de sinal de entrada (**self.signal**). Nesse ponto também é válido perceber que a pasta com o nome do arquivo já foi criada dentro da pasta do projeto, onde serão salvos os resultados obtidos pelo *software*.

Com o arquivo aberto, pode ser feita a plotagem do sinal no campo dos gráficos da janela do software, utilizando o botão *Plot*.

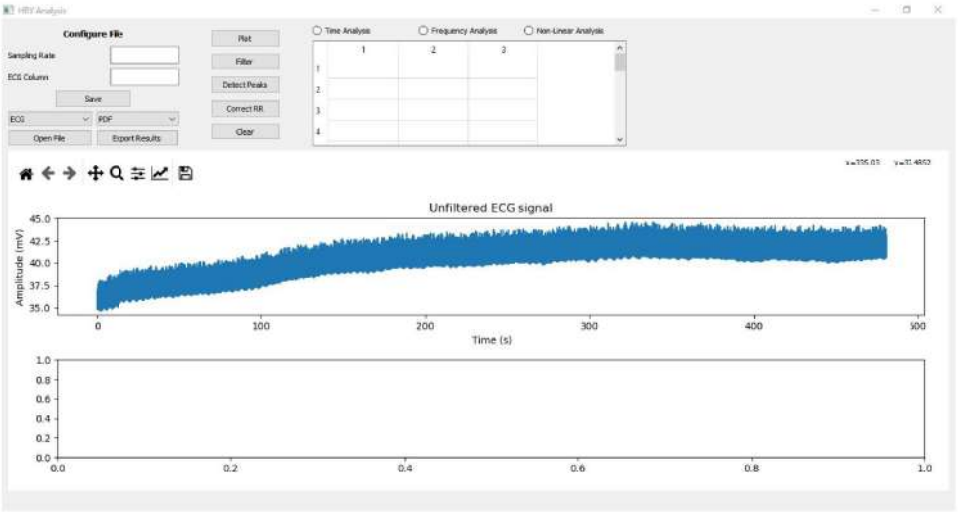
Pode-se perceber na Figura 22 que o sinal de ECG coletado pelo eletrocardiógrafo apresenta tanto variação da média de valores (curva) quanto um nível DC - pois o valor dos dados está centrado em cerca de 40 mV e não em 0 mV, como o esperado.

Figura 21 – Abertura do Arquivo.



Fonte: Autor (2019)

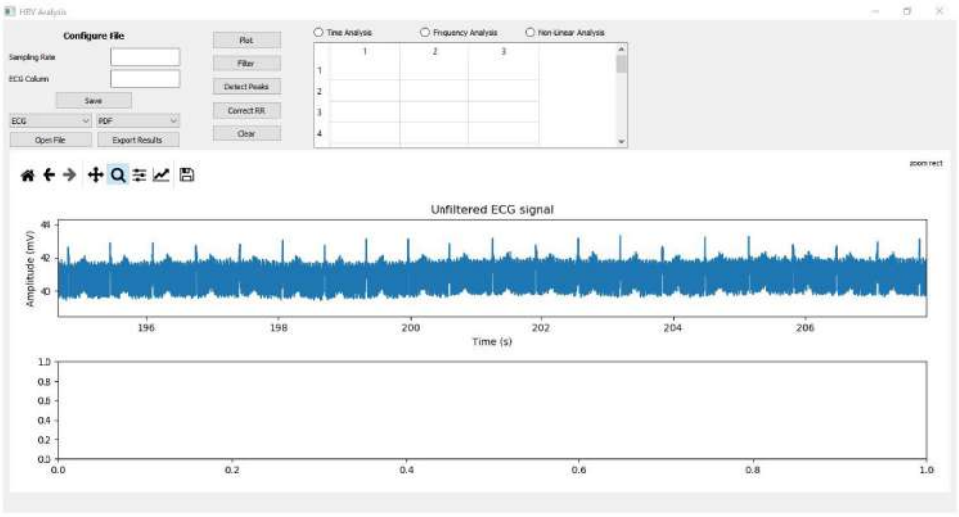
Figura 22 – Plot.



Fonte: Autor (2019)

Com o auxílio da ferramenta de zoom, pode-se também perceber a interferência do ruído no sinal (Figura 23) não sendo possível distinguir as ondas P e T do eletrocardiograma. A onda R ainda é visível, mas esse sinal está sujeito a erros.

Figura 23 – Plot: Ruído.

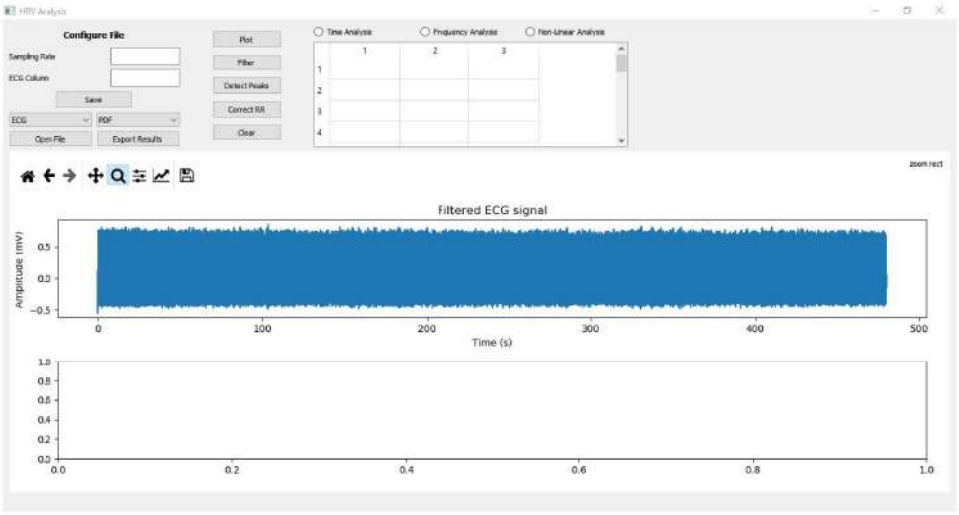


Fonte: Autor (2019)

Para corrigir esses erros, é utilizado o botão *Filter*, para fazer a filtragem do sinal. Como comentado anteriormente, são usados quatro filtros FIR, que devem agir sobre o sinal de ECG não processado, eliminando a curva do sinal, o nível DC e o ruído.

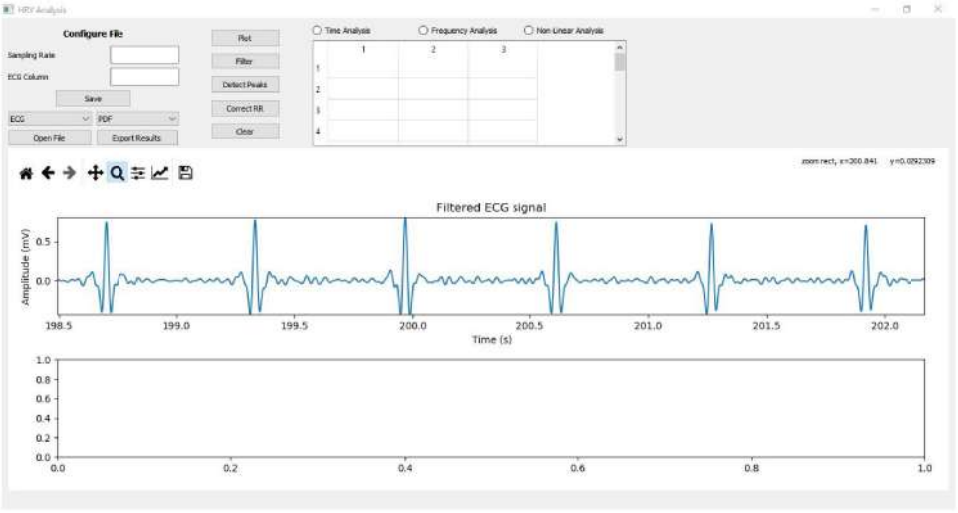
Na Figura 24, percebe-se a eliminação do nível DC - com o sinal agora centrado em 0 mV - e da curva apresentada no sinal antes da filtragem. Novamente com a ferramenta de zoom, pode-se ver a eliminação de grande parte do ruído, sendo possível localizar as ondas P e T no sinal (Figura 25).

Figura 24 – Filter.



Fonte: Autor (2019)

Figura 25 – Filter: Zoom.

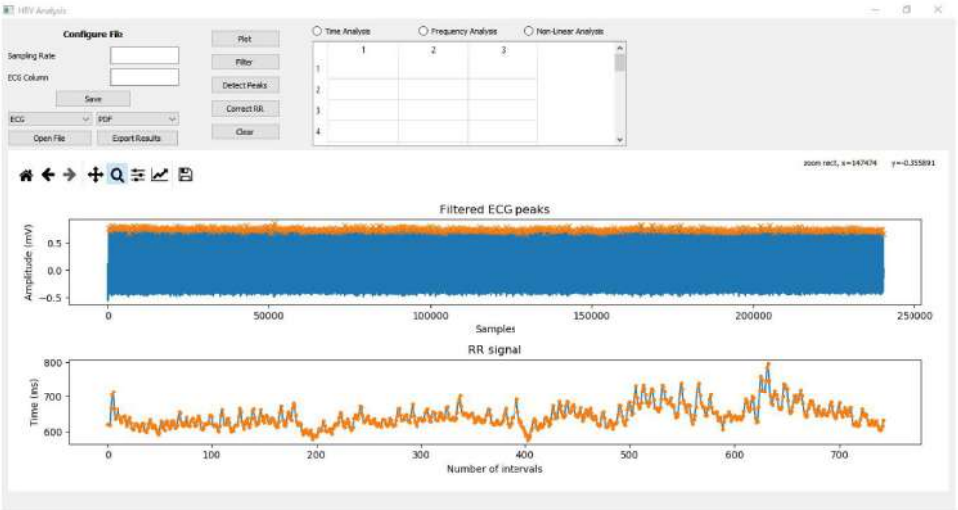


Fonte: Autor (2019)

Posterior à filtragem, pode ser feita a detecção dos picos do ECG filtrado e a construção do sinal RR, com o uso do botão *Detect Peaks*.

Na Figura 26 vê-se os dois gráficos, acima o sinal ECG filtrado seus picos sinalizados e abaixo o sinal RR. Nessa etapa houve dificuldade em relação ao sinal superior, visto que a biblioteca *MatPlotLib* mostrou o gráfico com o eixo x em relação às amostras e não ao tempo. Mas a detecção dos picos e a contrução da onda RR foram satisfatórios.

Figura 26 – Detect Peaks.



Fonte: Autor (2019)

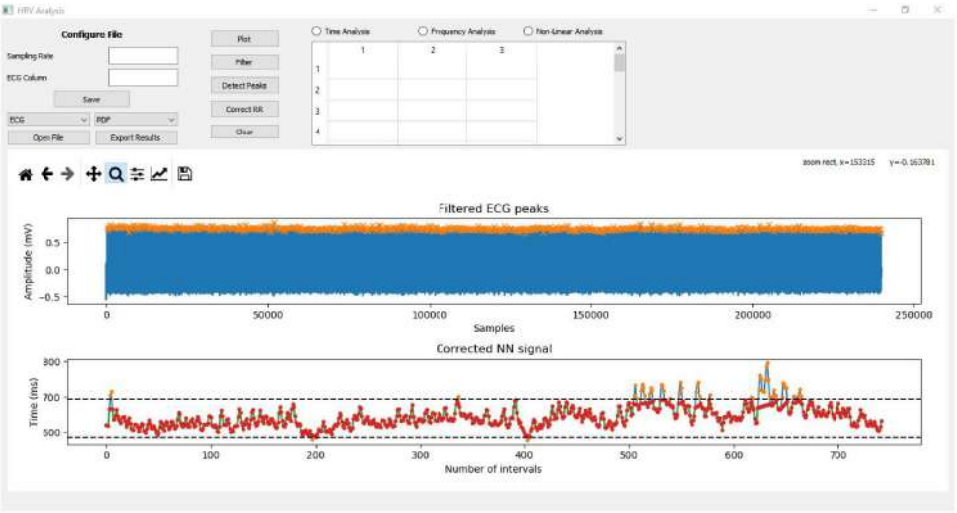
Após essa etapa, é feita a correção da onda RR com os métodos apresentados

anteriormente, retirada dos batimentos ectópicos com o método de Malik e retirara dos *outliers* com o uso do intervalo interquartil. Isso é feito pelo uso do botão *Correct RR*.

A onda corrigida é apresentada no gráfico inferior, juntamente com a onda não corrigida e os limites calculados para a eliminação dos *outliers* (Figura 27). Isso foi feito para que pudesse ser observada a diferença entre uma onda e outra e possíveis erros cometidos pelo programa.

Pode-se perceber na Figura 28 que há a eliminação de dois grandes grupos de outliers entre as amostras 500 e 700, seguindo o método de eliminação, possivelmente causados por movimento dos equipamentos de ECG durante a coleta de dados.

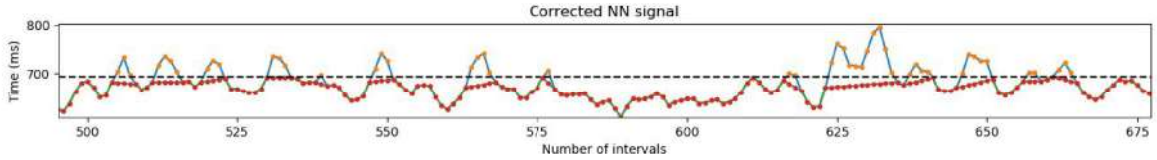
Figura 27 – Correct RR.



Fonte: Autor (2019)

Dado um zoom na área problemática, fica evidente um possível erro de escolha de parâmetros para a eliminação de *outliers*. Uma possível correção para esse erro é apresentada na seção de trabalhos futuros no capítulo 5.

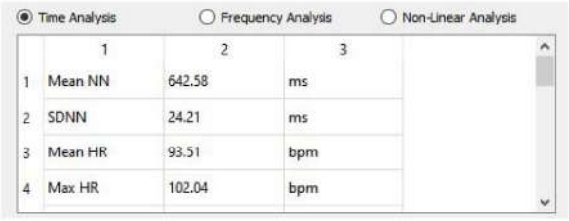
Figura 28 – Correct RR: Zoom.



Fonte: Autor (2019)

A partir da onda corrigida, são obtidos os parâmetros numéricos, alocados na tabela da área 3 do *software*, como mostrado na Figura 29 abaixo.

Figura 29 – Tabela com os resultados numéricos.



Fonte: Autor (2019)

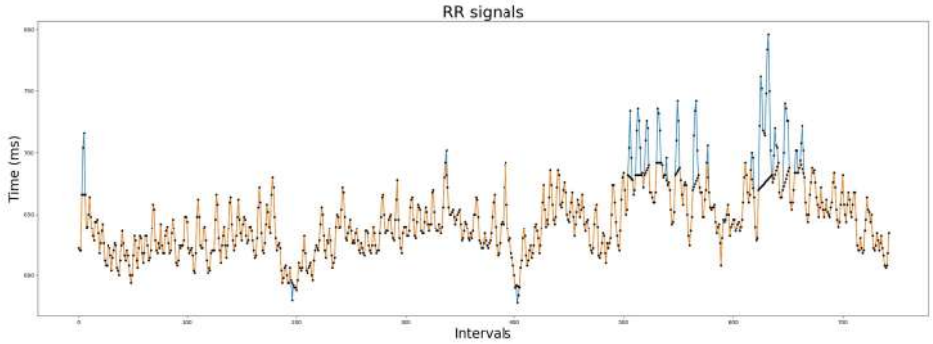
Para a visualização de todos os dados obtidos, basta variar entre os três *Radio Buttons* e usar o *scroll* do *QWidgetTable*. A tabela completa do arquivo de teste será mostrada na parte de exportação do arquivo em pdf.

Agora com PDF selecionado na *Combo box*, clicou-se em *Export* para exportar os resultados numéricos em formato PDF para a pasta com os resultados. Para o teste também foi selecionado o formato TXT e clicado novamente em *Export* para exportar esse tipo de arquivo, que também é alocado na pasta de resultados. Ambos os arquivos criados pelo *software* estão anexados ao final desse documento (Anexos B e C).

Entrando na pasta com os resultados, tem-se seis arquivos. Os dois exportados acima, PDF e txt com os resultados numéricos, e mais quatro imagens em formato PNG. Essas imagens foram salvas ao decorrer dos passos de execução do *software*, sendo elas as ondas RR e NN juntas, o histograma criado na análise temporal dos dados, o gráfico PSD e o gráfico de Poincaré/Lorenz.

Na Figura 30 são mostradas as ondas RR e NN, com os respectivos pontos representando os intervalos.

Figura 30 – Ondas RR e NN.

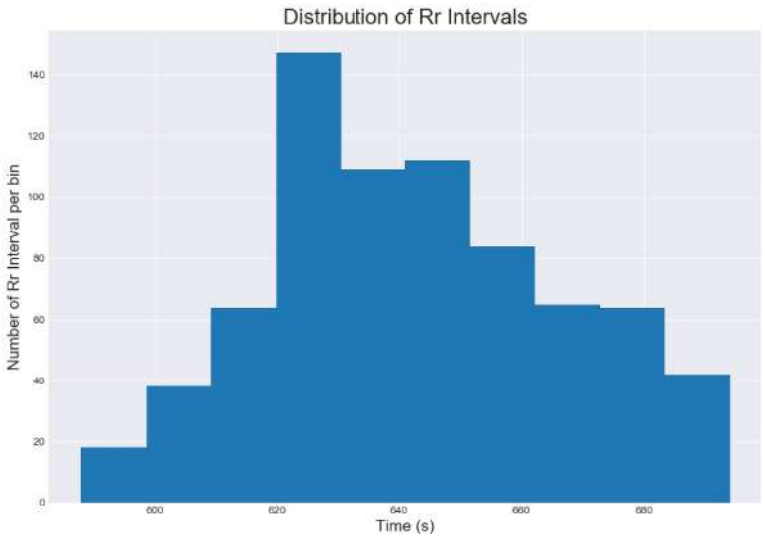


Fonte: Autor (2019)

Na Figura 31 é apresentado o Histograma dos intervalos RR. O histograma

representa a distibuição dos intervalos RR, mostrando no eixo x o tempo de cada intervalo e no eixo y quantas vezes aquele intervalo foi encontrado na amostra. Esse gráfico é utilizado para as análises geométricas de *triangular index* e TINN.

Figura 31 – Histograma.



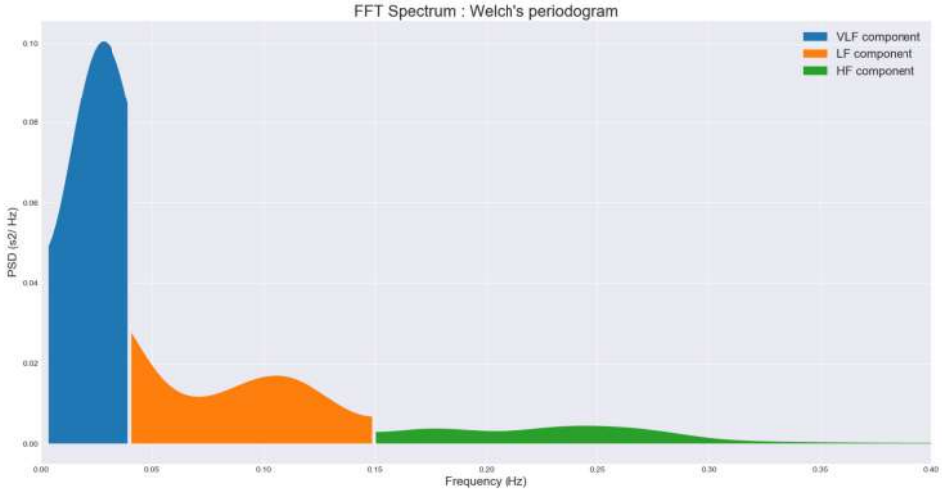
Fonte: Autor (2019)

Na Figura 32 é mostrado o espectro de potência do sinal RR, construído com o método de Welch.

O PSD é usado para calcular as potências de cada uma das faixas de frequência apresentadas anteriormente, *Very Low Frequency, Low Frequency e High Frequency*, indicando as atividades simpática e parassimpática do paciente durante o exame.

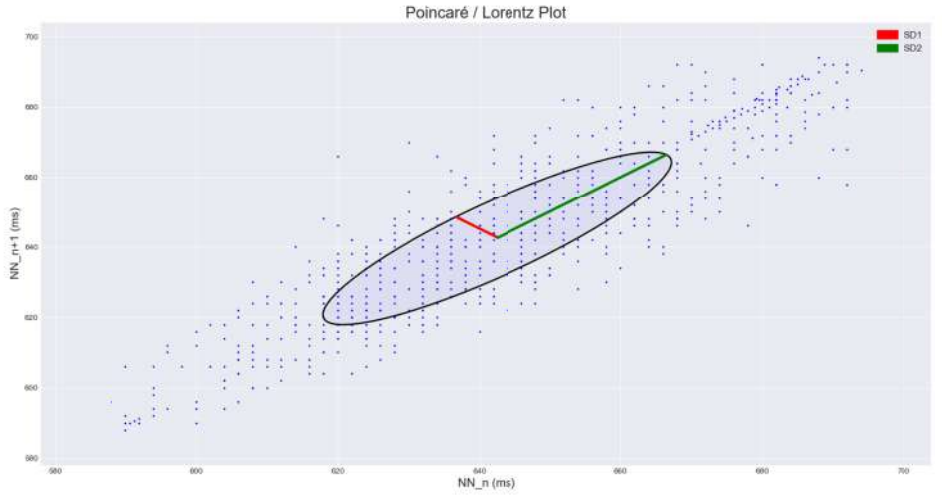
E por último, na Figura 33 é mostrado o gráfico de Poincaré/Lorenz, de onde são retirados os dados de espalhamento dos pontos, SD1 e SD2.

Figura 32 – PSD.



Fonte: Autor (2019)

Figura 33 – Poincaré.



Fonte: Autor (2019)

5.2 SEGUNDO TESTE

Rodando o programa mais uma vez, com um segundo arquivo de teste, podem ser feitas algumas comparações entre os resultados obtidos nos dois arquivos, de modo a ser melhor analisada a concordância dos resultados.

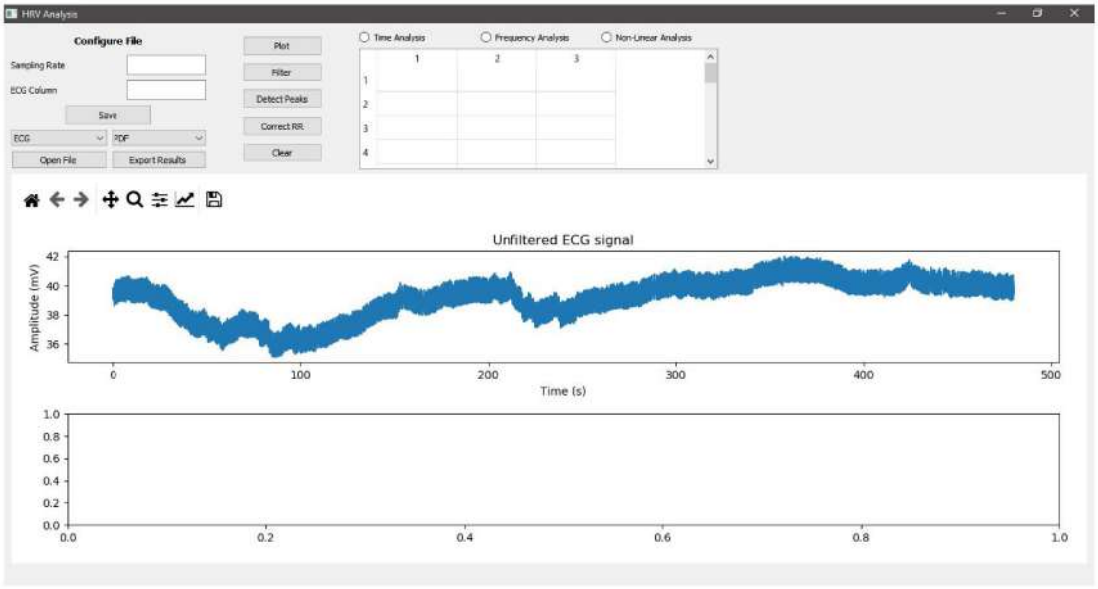
Para o segundo teste, foi utilizado um segundo arquivo de eletrocardiograma,

com mesma frequência de amostragem de 500 Hz.

A partir da plotagem do ECG não filtrado (Figura 34), pode-se perceber uma maior influência da interferência da rede nesse caso, com maior variação da onda em relação ao eixo do valor médio.

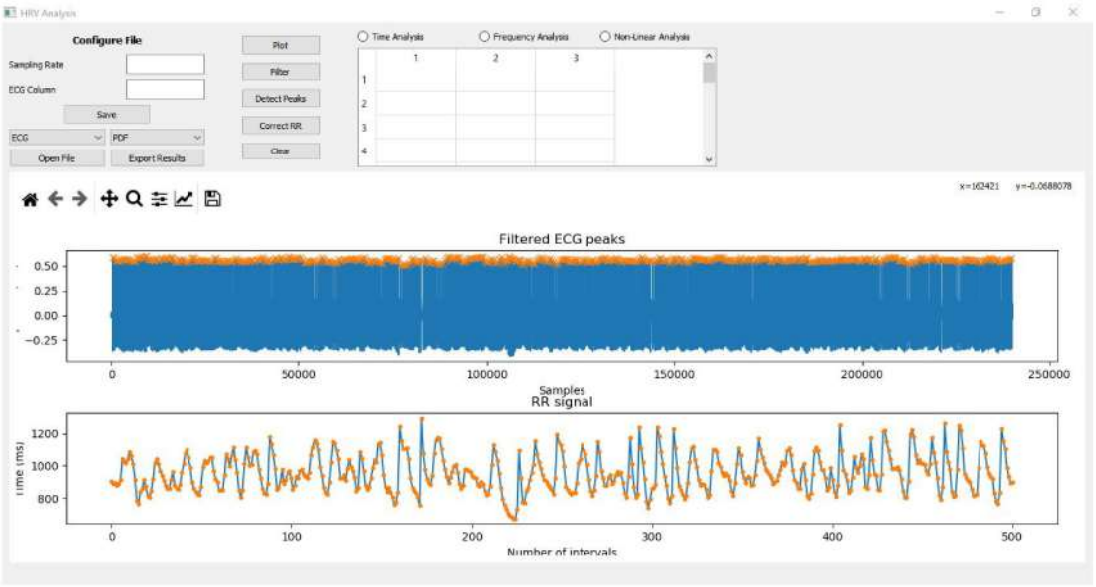
Os gráficos com zoom e do ECG filtrados se assemelham com os apresentados anteriormente, então não serão mostrados. Na Figura 35, pode-se perceber a diferença da onda RR obtida por esse teste, em relação ao anterior, principalmente no valor médio dos pontos e na variação dos intervalos.

Figura 34 – ECG não filtrado (teste 2).



Fonte: Autor (2019)

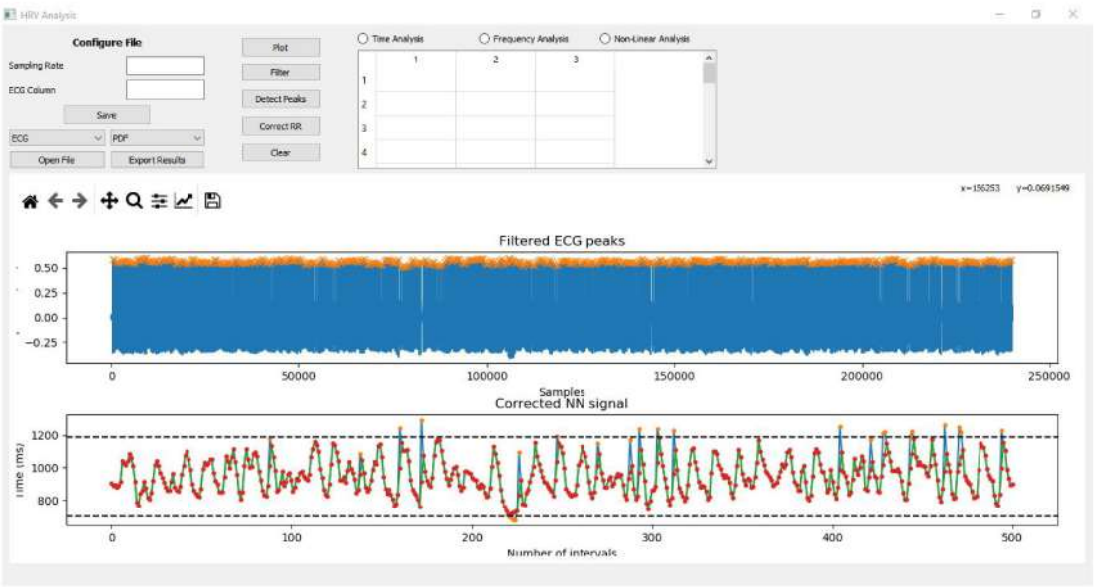
Figura 35 – Onda RR (teste 2).



Fonte: Autor (2019)

A partir da correção da onda RR nota-se a principal diferença entre os dois testes, a não eliminação do grande grupo de outliers que aconteceu no último teste (Figura 36).

Figura 36 – Onda RR corrigida (teste 2).

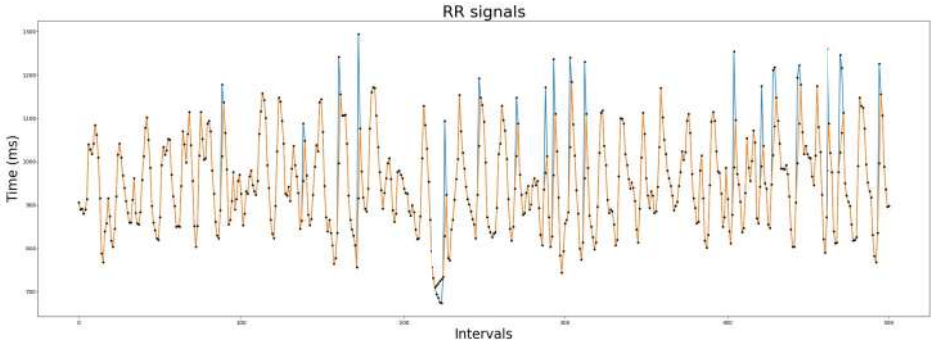


Fonte: Autor (2019)

Nesse caso foram eliminados alguns pontos identificados pelo software como sendo batimentos ectópicos e poucos outliers causados pelos limites superior e inferior.

Na Figura 37 é mostrado o arquivo com o gráfico salvo pelo software na pasta de resultados, para uma melhor visualização do que foi dito anteriormente. Uma discussão entre os dois resultados obtidos será feita na seção Discussões (Seção 5.1).

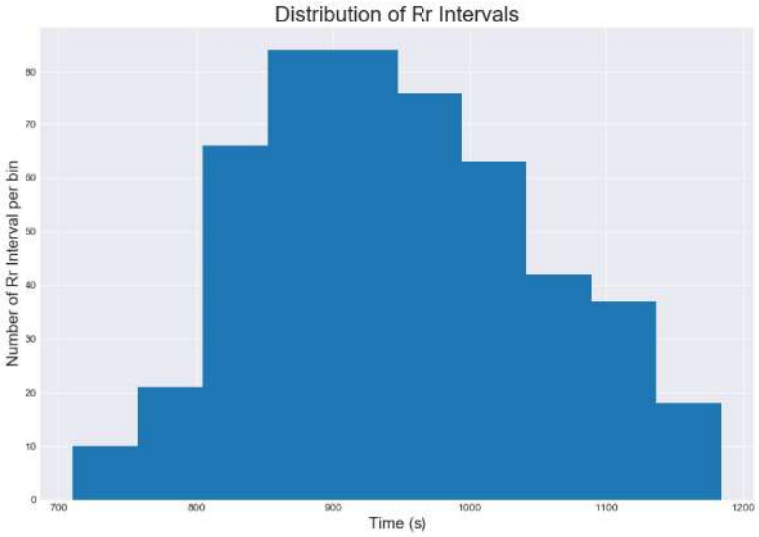
Figura 37 – Ondas RR (teste 2).



Fonte: Autor (2019)

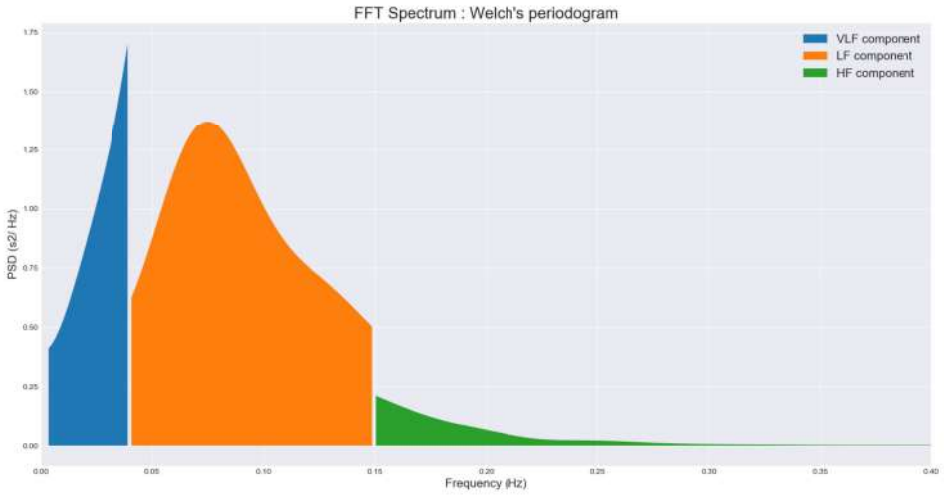
A Figura 38 mostra o histograma do segundo teste, assim como a Figura 39 mostra o PSD e a Figura 40 o Poincaré.

Figura 38 – Histograma (teste 2).



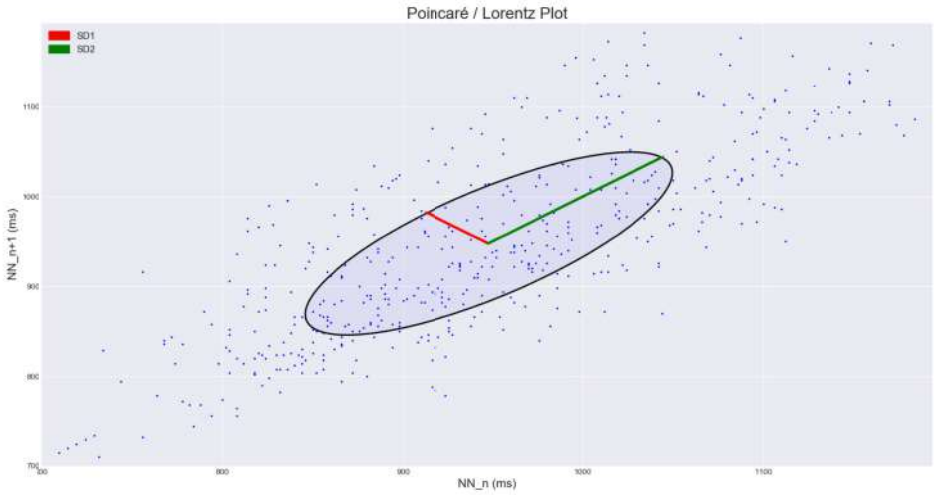
Fonte: Autor (2019)

Figura 39 – PSD (teste 2).



Fonte: Autor (2019)

Figura 40 – Poincaré (teste 2).



Fonte: Autor (2019)

6 CONSIDERAÇÕES FINAIS

6.1 DISCUSSÕES

Analisando os dados obtidos nas duas simulações, pode-se chegar a algumas conclusões sobre os sinais.

Primeiramente a questão da eliminação dos outliers, que foi muito maior no primeiro caso, como comentado anteriormente. Esse erro deve ser corrigido utilizando, por exemplo, um método de janelamento, onde os outliers são calculados em trechos, considerando a variação da mediana local dentro das janelas.

Segundo, percebe-se pelos gráficos de PSD que o segundo arquivo apresenta uma área maior de LF, ou seja, uma maior atividade do sistema simpático, com um possível aumento no ritmo cardíaco, o que não é confirmado pelos dados numéricos obtidos, mostrando um intervalo maior de tempo entre os batimentos em média, e, por consequência, uma frequência cardíaca menor. Mas ao mesmo tempo, o sinal mostra um desvio padrão elevado, o que pode levar a conclusão de que ele varia de valor muitas vezes, causando o aumento no gráfico de LF.

Observando ainda as tabelas com os dados numéricos, pode-se perceber mais uma vez a grande variabilidade dos valores do segundo teste, com os espalhamentos do gráfico de Poincaré, muito maiores do que o primeiro teste feito. Isso pode levar a uma conclusão de que a pessoa do segundo teste é mais saudável, com uma maior variação de ritmo cardíaco e ao mesmo tempo com os batimentos mais desacelerados em repouso.

6.2 QUANTO AO ALCANCE DOS OBJETIVOS

O intuito inicial desse trabalho era o desenvolvimento de um *software standalone*, que pudesse ser utilizado em várias plataformas, e que apresentasse os resultados da análise de variabilidade do ritmo cardíaco de maneira fácil e rápida.

Quanto a leitura da literatura sobre o assunto, os estudos se mostraram satisfatórios para o entendimento básico dos processos envolvidos, assim como os interesses na área e as novas tecnologias e usos que estão sendo desenvolvidos. Os estudos sobre *Python* e suas bibliotecas também foram satisfatórios, de modo a aprender a utilizar diversas bibliotecas, a integração entre elas, o entendimento entre os diferentes tipos de variáveis e os modos de construção de *graphical user interfaces*.

O *software* trabalha de maneira satisfatória, fazendo o pré processamento do sinal de ECG, assim como a análise dos intervalos RR e os cálculos dos parâmetros ligados à variabilidade de ritmo cardíaco. Com o processo sendo feito em passo a passo, fica fácil utilizar e entender o fluxo de funcionamento do programa, assim como os resultados.

6.3 TRABALHOS FUTUROS

Para continuar o desenvolvimento do *software*, primeiro deve-se estudar mais algumas bibliotecas de plotagem de gráficos e sinais, para que seja corrigido o erro na escala do eixo x mostrado no gráfico de detecção dos picos. Além disso, um estudo mais aprofundado do código de plotagem do gráfico de PSD feito pela biblioteca HRV-Analysis.

Também se faz necessário um estudo de métodos de correção dos pontos RR eliminados, de modo que o usuário consiga decidir se um ponto retirado pelo software é realmente um *outlier* ou um batimento ectópico, de modo a diminuir os erros de cálculo aproximados pelo programa.

Para aumentar a independência do software em relação aos conhecimentos no assunto, pode ser implementado um modo de análise dos dados obtidos, como identificar uma faixa normal de valores ou fazer algumas assunções dos valores obtidos, dizendo algumas causas e doenças que podem estar relacionadas a um dado específico.

Por último, deve ser implementada a exportação dos resultados em arquivo CSV. Seria interessante a exportação de múltiplos testes em um único arquivo CSV, como um modelo de banco de dados para as análises dos pacientes.

REFERÊNCIAS

ALVES, L. **Bloqueio do Ramo Esquerdo (BRE) – É grave? Mata? Saiba Tudo!** [*S.l.*: *s.n.*], 2017. http://sopronocoracao.com/bloqueio-do-ramo-esquerdo-bre/. [Online; acessado em 26-11-2019].

ARAUJO, C. G.; LAUKKANEN, J. A. Músculo Cardíaco e Músculo Esquelético Conectados pelo Sistema Nervoso Autônomo. **Arquivo Brasileiro de Cardiologia**, v. 112, n. 6, 2009.

ARMOUR, J. A. M. D. Intrinsic Cardiac Neurons. **Journal of Cardiovascular Electrophysiology**, 1991. Disponível em: https://doi.org/10.1111/j.1540-8167.1991.tb01330.x.

BIOSPPY. [*S.l.*: *s.n.*]. https://biosppy.readthedocs.io/en/stable/. [Online; acessado em 28-11-2019].

CAMM, A. J.; MALIK, U.K M. Heart rate variability: Standards of measurement, physiological interpretation, and clinical use. **European Heart Journal**, p. 354–381, 1996. Disponível em: https://www.escardio.org/static_file/Escardio/Guidelines/Scientific-Statements/guidelines-Heart-Rate-Variability-FT-1996.pdf.

CONCEIÇÃO, P. O. *et al.* Estimação espectral do sinal de vibração para o monitoramento do desgaste do dressador de ponta única. **Matéria (Rio de Janeiro)**, v. 21, n. 4, 2016. Disponível em: http://dx.doi.org/10.1590/s1517-707620160004.0079.

DEGIORGIO, C. M. *et al.* RMSSD, a Measure of Heart Rate Variability, Is Associated With Risk Factors For SUDEP: The SUDEP-7 Inventory. **Epilepsy Behav**, 2010. DOI: 10.1016/j.yebeh.2010.06.011.

DEXTRO, Rafael Barty. **Coração**. [*S.l.*: *s.n.*], 2015. https://www.infoescola.com/anatomia-humana/coracao/. [Online; acessado em 26-11-2019].

EKG. **Derivações do Eletrocardiograma**. [*S.l.*: *s.n.*], 2019. https://pt.my-ekg.com/generalidades-ecg/derivacoes-ecg.html. [Online; acessado em 26-11-2019].

EWING, D. J. *et al.* The value of cardiovascular autonomic function tests: 10 years experience in diabetes. **Diabetic Care**, 1985.

FERNANDES, C. R. R. Avaliação e Melhoria da Qualidade de Sinais ECG Adquiridos em Sistemas Vestíveis. **Faculdade de Engenharia da Universidade do Porto**, 2017.

FPDF. [*S.l.*: *s.n.*]. https://pyfpdf.readthedocs.io/en/latest/. [Online; acessado em 28-11-2019].

GAHERY, Y.; VIGIER, D. Inhibitory effects in the cuneate nucleus produced by vago-aortic afferent fibers. **Brain Research**, v. 75, n. 2, p. 241–246, 1974. Disponível em: https://doi.org/10.1016/0006-8993(74)90744-6.

HAGE, B. *et al.* Low cardiac vagal tone index by heart rate variability differentiates bipolar from major depression. **The World Journal of Biological Psychiatry**, v. 20, p. 359–367, 2017. Disponível em: https://doi.org/10.1080/15622975.2017.1376113.

HRV-ANALYSIS. [*S.l.*: *s.n.*]. https://pypi.org/project/hrv-analysis/. [Online; acessado em 28-11-2019].

JETBRAINS. **PyCharm**. [*S.l.*: *s.n.*], 2017. https://www.jetbrains.com/pycharm/. [Online; acessado em 28-11-2019].

KLEIGER, R. E. *et al.* Decreased heart rate variability and its association with increased mortality after acute myocardial infarction. **Am J Cardiol**, 1987. DOI: 1;59(4):256-62.

KLOTER, E. *et al.* Heart Rate Variability as a Prognostic Factor for Cancer Survival - A Systematic Review. **Frontiers in Physiology**, 2018. Disponível em: doi:10.3389/fphys.2018.00623.

KUBIOS. **Kubios HRV**. [*S.l.*: *s.n.*], 2016. https://www.kubios.com/support/. [Online; acessado em 26-11-2019].

LACEY, B. C.; LACEY, J. I. Two-way communication between the heart and the brain: Significance of time within the cardiac cycle. **American Psychologist**, v. 33, n. 2, p. 99–113, 1978. Disponível em: https://doi.org/10.1037/0003-066X.33.2.99.

LASS, A. D. *et al.* The Effect of an Indoor Cycling Session on Heart Rate Variability in Triathletes. **Journal of Exercise Physiology**, v. 22, n. 4, p. 112–119, 2019. ISSN 1097-9751.

MANI, A.R. *et al.* Decreased heart rate variability in patients with cirrhosis relates to the presence and degree of hepatic encephalopathy. **American Journal of Physiology**, v. 296, n. 2, 2009. Disponível em: doi:10.1152/ajpgi.90488.2008.

MATPLOTLIB. [*S.l.*: *s.n.*]. https://matplotlib.org/. [Online; acessado em 28-11-2019].

MCCARTY, R.; ATKINSON, M.; BRADLEY, R. T. Electrophysiological Evidence of Intuition: Part 1. The Surprising Role of the Heart. **The Journal of Alternative and Complementary Medicine**, v. 10, n. 1, 2004. Disponível em: https://doi.org/10.1089/107555304322849057.

MCCARTY, R.; ATKINSON, M.; TILLER, W. A. *et al.* The effects of emotions on short-term power spectrum analysis of heart rate variability. **Journal of Cardiovascular Electrophysiology**, 1995. Disponível em: doi:10.1016/S0002-9149(99)80309-9.

MITCHELL, L. B. Batimentos ventriculares prematuros (Batimentos ventriculares ectópicos, contrações ventriculares prematuras). **Libin Cardiovascular Institute of Alberta**, 2017.

MORSCH, J. A. **Ondas do eletrocardiograma: como funciona o ECG e como interpretar**. [*S.l.*: *s.n.*], 2018. https://telemedicinamorsch.com.br/blog/ondas-do-eletrocardiograma. [Online; acessado em 26-11-2019].

NEVROKARD. **Nevrokard – flexible, versatile, professional software tools for serious research of the autonomic nervous system sleep apnea in humans**. [*S.l.*: *s.n.*], 1987. http://www.nevrokard.eu/. [Online; acessado em 26-11-2019].

NOBLE, D. Modeling the heart–from genes to cells to the whole organ. **Science**, v. 295, p. 1678–1682, 2002.

NUMPY. [*S.l.*: *s.n.*]. https://numpy.org/. [Online; acessado em 28-11-2019].

P.WILLIAMS, D. *et al.* Heart rate variability and inflammation: A meta-analysis of human studies. **Brain, Behavior, and Immunity**, v. 80, p. 119–226, 2019. Disponível em: https://doi.org/10.1016/j.bbi.2019.03.009.

PYHRV. [*S.l.*: *s.n.*]. https://pypi.org/project/pyhrv/. [Online; acessado em 28-11-2019].

PYPI Python Community. [*S.l.*: *s.n.*]. https://pypi.org. [Online; acessado em 28-11-2019].

PYQT5. [*S.l.*: *s.n.*]. https://pypi.org/project/PyQt5/. [Online; acessado em 28-11-2019].

RAMOS, A.P.; SOUZA, B. S. **Eletrocardiograma: princípios, conceitos e aplicações**. [*S.l.*: *s.n.*], 2007. http://files.samuzeiro.webnode.com.br/200000056-a4b30a5acf/ecg.pdf. [Online; acessado em 26-11-2019].

ROY, B.; GHATAK, S. Métodos não-lineares para avaliar mudanças na variabilidade da frequência cardíaca em pacientes com diabetes tipo 2. **Arquivos Brasileiros de Cardiologia**, v. 101, n. 4, 2013. Disponível em: http://dx.doi.org/10.5935/abc.20130181.

SCHWARZ, ROHDE. **Capturando pequenos sinais de ECG em aplicações médicas**. [*S.l.*: *s.n.*], 2019. https://www.rohde-schwarz.com/br/aplicativos/capturando-pequenos-sinais-de-ecg-em-aplica-es-m-dicas-cartao-de-

aplicativo _ 56279 - 152385 . html ? rusprivacypolicy = 0. [Online; acessado em 26-11-2019].

SHAFFER, F.; GINSBERG, J. P. An Overview of Heart Rate variability Metrics and Norms. **Front Public Health**, p. 1–17, 2017. DOI: 10.3389/fpubh.2017.00258. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990/.

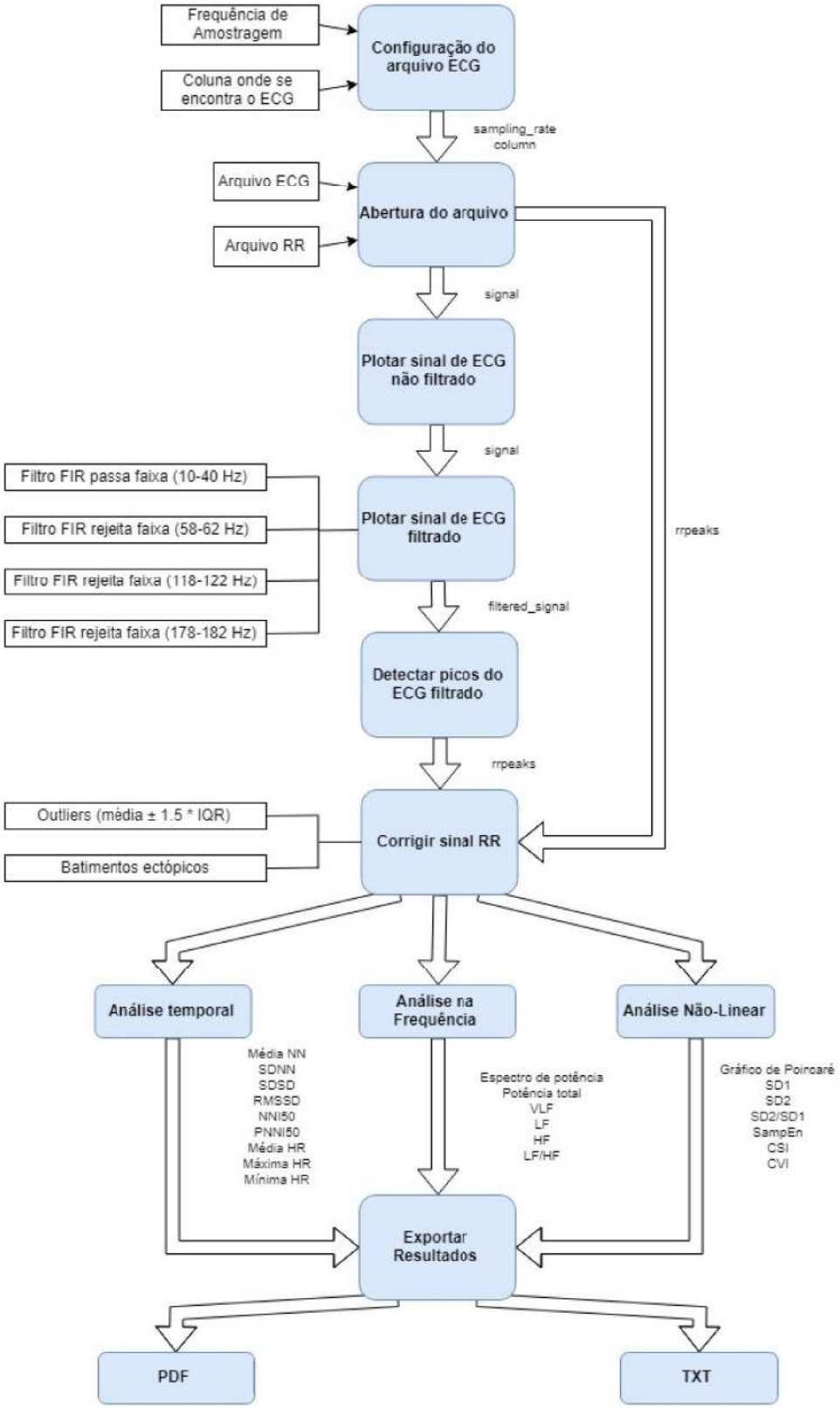
SOUZA, O. F. de. **Arritmia Cardíaca**. [*S.l.*: *s.n.*], 2018. http://arritmia.med.br/. [Online; acessado em 26-11-2019].

TKINTER. [*S.l.*: *s.n.*]. https://docs.python.org/3/library/tkinter.html. [Online; acessado em 28-11-2019].

TULPPO, M. P. *et al.* Vagal modulation of heart rate during exercise: effects of age and physical fitness. **American Journal of Physiology**, 1998.

VANDERLEI, L. C. M. *et al.* Basic notions of heart rate variability and its clinical applicability. **Brazilian Journal of Cardiovascular Surgery**, v. 24, n. 2, p. 205–217, 2009.

ANEXO A – FLUXOGRAMA DO SOFTWARE



Fonte: Autor (2019)

ANEXO B – ARQUIVOS PDF EXPORTADOS

RESTING_1.txt

Time analysis results

Mean NN intervals: 642.58 ms
SDNN: 24.21 ms
Mean heart rate: 93.51 bpm
Maximun heart rate: 102.04 bpm
Minimun heart rate: 86.46 bpm
Standard diviation of the heart rate: 3.51 bpm
RMSSD: 11.11 ms
NNi 50: 0 beats
Percentage of NNi 50: 0.0 %
NNi 20: 44 beats
Percentage of NNi 20: 5.92 %
Triangular index: 7.14
SDSD: 11.11 ms
Median NN intervals: 640.0 ms
Range NN intervals: 106.0 ms
CVSD: 0.02
CVNNi: 0.04

Fonte: Autor (2019)

Frequency analysis results

Total power: 246.41 ms²/Hz
Very Low Frequency: 65.74 ms²
Low Frequency: 97.67 ms²
High Frequency: 82.99 ms²
LF/HF ratio: 1.18
Normalized LF: 54.06 nu
Normalized HF: 45.94 nu

Non-Linear analysis results

SD1: 7.86 ms
SD2: 33.33 ms
SD2/SD1: 4.24
CVI: 3.62
CSI: 4.24
Modified CSI: 565.25
Sample Enthropy: 1.19

Fonte: Autor (2019)

RESTING_2.txt

Time analysis results

Mean NN intervals: 947.78 ms
SDNN: 101.38 ms
Mean heart rate: 64.03 bpm
Maximun heart rate: 84.51 bpm
Minimun heart rate: 50.68 bpm
Standard diviation of the heart rate: 6.87 bpm
RMSSD: 67.79 ms
NNi 50: 215 beats
Percentage of NNi 50: 42.91 %
NNi 20: 376 beats
Percentage of NNi 20: 75.05 %
Triangular index: 29.47
SDSD: 67.79 ms
Median NN intervals: 940.0 ms
Range NN intervals: 474.0 ms
CVSD: 0.07
CVNNi: 0.11

Fonte: Autor (2019)

Frequency analysis results

Total power: 8643.16 ms²/Hz
Very Low Frequency: 739.27 ms²
Low Frequency: 6573.04 ms²
High Frequency: 1330.84 ms²
LF/HF ratio: 4.94
Normalized LF: 83.16 nu
Normalized HF: 16.84 nu

Non-Linear analysis results

SD1: 47.98 ms
SD2: 135.1 ms
SD2/SD1: 2.82
CVI: 5.02
CSI: 2.82
Modified CSI: 1521.61
Sample Enthropy: 1.24

Fonte: Autor (2019)

ANEXO C – ARQUIVO TXT EXPORTADO PARA RESTING_1.TXT

'mean_nni': 642.5800807537012
'sdnn': 24.21475949152013
'sdsd': 11.110233778123801
'nni_50': 0
'pnni_50': 0.0
'nni_20': 44
'pnni_20': 5.921938088829071
'rmssd': 11.110245548795817
'median_nni': 640.0
'range_nni': 106.0
'cvsd': 0.01729005595032495
'cvnni': 0.037683644757736535
'mean_hr': 93.50559038807943
'max_hr': 102.04081632653062
'min_hr': 86.45533141210375
'std_hr': 3.5087068572626596
'triangular_index': 7.144230769230769
'tinn': None
'lf': 97.67476698613021
'hf': 82.99298024510395
'lf_hf_ratio': 1.1769039585958523
'lfnu': 54.063200810888304
'hfnu': 45.936799189111696
'total_power': 246.41039057974052
'vlf': 65.74264334850636
'sd1': 7.861420884565456
'sd2': 33.33027476844508
'ratio_sd2_sd1': 4.239726540259831
'csi': 4.239726540259831
'cvi': 3.622459924718163
'Modified_csi': 565.2450021197168
'sampen': 1.1911528839457348

ANEXO D – APÊNDICE

Para acesso ao código completo desenvolvido durante o trabalho de conclusão de curso, enviar email para fefonascimento@gmail.com com o assunto "Código Python Análise HRV