



**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO**

RODRIGO NICOLE DE ALCÂNTARA SILVA
rodrigonicole@hotmail.com
Matrícula (21953324)

Simulador de ECG para monitores multiparâmetros

BRASÍLIA
2022



RODRIGO NICOLE DE ALCÂNTARA SILVA
rodrigonicole@hotmail.com

Simulador de ECG para monitores multiparâmetros

Trabalho de Conclusão de Curso (TCC) apresentado
como um dos requisitos para a conclusão do curso de
Engenharia Elétrica do CEUB– Centro Universitário de
Brasília

Orientador (a): **MsC. Francisco Javier de Obaldia
Diaz**

BRASÍLIA
2022



RODRIGO NICOLE DE ALCÂNTARA SILVA
rodrigonicole@hotmail.com

Simulador de ECG para monitores multiparâmetros

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia Elétrica do CEUB – Centro Universitário de Brasília

Orientador (a): **MsC. Francisco Javier de Obaldia Diaz**

Brasília, 2022.

BANCA EXAMINADORA

MsC. Francisco Javier de Obaldia Diaz
Orientador (a)

MsC. Luciano Henrique Duque
Examinador (a)

MsC. Nilo Sergio Soares Ribeiro
Examinador (a)

Simulador de ECG para monitores multiparâmetros

ECG simulator for multiparameter monitors

aluno¹, orientador², Primeiro examinador³, Segundo examinador⁴

RESUMO

A simulação de sinais biológicos por meio de gráficos ou valores é o recurso utilizado para testes de monitores multiparâmetros, eletrocardiógrafo, oxímetros de pulso, outros aparelhos de uso médico hospitalar e seus acessórios. Os principais sinais vitais produzidos pelo nosso corpo e captados pelo monitor são: sinais cardíacos; saturação de oxigênio na corrente sanguínea; pressão arterial invasiva; pressão arterial não invasiva; e temperatura. O simulador de eletrocardiograma – ECG, é aplicado em testes de monitores multiparâmetros ou eletrocardiógrafo após manutenção corretiva ou reparo na assistência técnica, bem como na manutenção preventiva periódica e calibração anual realizado pela engenharia clínica no ambiente hospitalar, com o objetivo de atestar o correto funcionamento do monitor antes de conectá-lo ao paciente. Isso é possível devido a possibilidade do operador ajustar os parâmetros disponibilizando na saída sinais de qualidade e repetibilidade conhecidos. Este artigo apresenta o estudo e modelamento dos principais circuitos integrantes deste tipo de simulador. O sinal escolhido foi o sinal cardíaco por estar presente em mais equipamentos (monitores, eletrocardiógrafo, desfibriladores e outros) e comumente ser representado por parte gráfica e numérica. Para o desenvolvimento serão utilizados o software de simulação Matlab e o Arduino Due.

Palavras-chave: Eletrocardiograma. Simulador. Arduino. Matlab. Wavelet.

Abstract

The simulation of biological signals through graphs or values is the resource used for testing multiparameter monitors, electrocardiographs, pulse oximeters, other devices for hospital medical use and their accessories. The main vital signs produced by our body and captured by the monitor are: heart signs; oxygen saturation in the bloodstream; invasive blood pressure; non-invasive blood pressure; and temperature. The electrocardiogram simulator - ECG, is applied in tests of multiparameter monitors or electrocardiograph after corrective maintenance or repair in technical assistance, as well as in periodic preventive maintenance and annual calibration carried out by clinical engineering in the hospital environment, with the objective of attesting to the correct functioning of the monitor before connecting it to the patient. This is possible due to the possibility for the operator to adjust the parameters, providing signals of known quality and repeatability at the output. This article presents the study and modeling of the main circuits of this type of simulator. The chosen signal was the cardiac signal, as it is present in more equipment (monitors, electrocardiographs, defibrillators and others) and is commonly represented graphically and numerically. For the development, Matlab simulation software and Arduino Due will be used.

keywords: Electrocardiogram. Simulator. Arduino. Matlab. Wavelet.

¹ UniCEUB, Rodrigo Nicole de Alcântara Silva.

² UniCEUB, Francisco Javier de Obaldia Diaz.

³ UniCEUB, primeiro examinador.

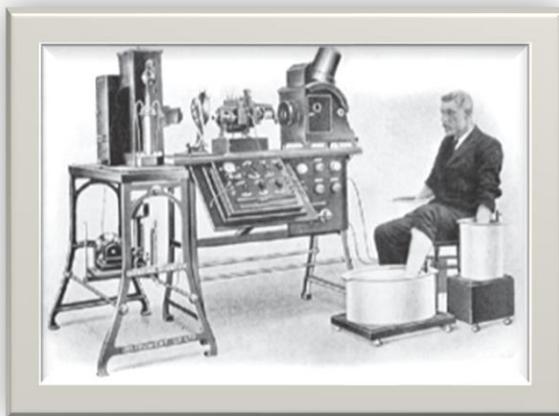
⁴ UniCEUB, segundo examinador.

1 INTRODUÇÃO

O coração, responsável pela circulação sanguínea, é formado por células com função mecânica ou elétrica. As células do miocárdio ao serem estimuladas eletricamente reagem produzindo contração. De forma sucinta, os estímulos associados aos eletrólitos contidos nos fluidos corporais criam um fluxo de corrente, que por sua vez é expresso em milivolts (mV) em um eletrocardiograma (doravante chamada ECG). A frequência cardíaca pode ser afetada por arritmias, resultado de distúrbios ocorridos na formação ou condução dos impulsos, podendo acontecer em ambos (AEHLERT, 2019).

Quando o coração está em repouso sua frequência é de 60 a 100 batimentos/minuto. O ECG é uma ferramenta importante no diagnóstico de arritmias, possibilitando avaliar variações na frequência, regularidade, amplitude e configuração. O sinal captado reflete a atividade elétrica das células miocárdicas, por meio das ondas vistas no ECG. A figura 1 demonstra o primeiro eletrocardiógrafo construído por Einthoven onde os elétrodos de captação são três recipientes com solução salina aquosa (THALER, 2013).

Figura 1 Eletrocardiógrafo de Einthoven



Fonte: ditada (M., 2013)

A GE Healthcare é exemplo de empresa que desde 1965 investe em desenvolvimento de ferramentas de cardiologia diagnóstica. Com portfólio de sistemas de ECG projetados para melhorar a precisão clínica somada às ferramentas avançadas de análise (HEALTHCARE, 2021).

Na década de 80 a Hewlett-Packard – HP busca aprimorar a impressão de gráficos no projeto Digital Control Simplified X-Y Plotter Electronics fazendo uso do processador Z80 de forma customizada (HP, 1981).

No Brasil são fundadas a Cmos Drake em 1989 (DRAKE, 2021) e Dixtal Biomedica em 1994, também com produtos na linha de ECG. A Dixtal foi vendida a Philips em 2014 (HUDSON, 2008).

Outro exemplo é a linha Fluke Biomedical produz o ProSim 8, é um simulador de paciente, importado, capaz de produzir sinais de: ECG, fetal, arritmia, respiração, temperatura, pressão invasiva (IBP) e não invasiva (NIBP), saída cardíaca, cateterismo cardíaco e oximetria (SPO2). Produz relatórios de teste, entre outros recursos avançados de simulação de fácil configuração (FLUKE CORPORATION, 2021).

No Brasil tem a R&D MEDIQ, fabricante da linha HS-30, que permite a simulação de sinais de ECG, arritmias, respiração, oximetria e temperatura. Também é possível configurar relatórios e laudos de calibração de acordo com as normas brasileiras ABNT (R&DMEDIQ, 2021).

Com o objetivo de modelar e atestar o correto funcionamento do simulador de eletrocardiograma, faz-se necessário uma especificação dos componentes que irão compor tal simulador. Assim, para o hardware, o componente principal será o Arduino por sua flexibilidade, disponibilidade de documentação, integração com outros programas (ARDUINO, 2018).

Já para o tratamento, processamento do sinal e desenvolvimento do software será utilizada a ferramenta MATLAB.

O MATLAB, conhecido no meio acadêmico, é um produto da Math Works, criada em 1984 por Jack Little – engenheiro electricista e cientista da computação, e Cleve Moler – matemático e cientista da computação. O desenvolvimento surgiu a partir da demanda por um sistema de maior capacidade e produtividade, em contrapartida do que era oferecido pelo mercado fundamentado em Fortran e C. A ferramenta desenvolvida traz diversos aplicativos de suporte a cálculos, gráficos, simuladores, modelagem, processamento de imagem, processamento de sinais, sistemas de controle tipo PID (Matlab MathWorks, 2021).

2 REVISÃO BIBLIOGRÁFICA

Para o desenvolvimento do trabalho foi necessário realizar pesquisas na literatura científica médica sobre o funcionamento do coração, sensoriamento de sinais vitais, especificamente os associados à frequência cardíaca, batimentos do coração e pressão arterial. Estes aspectos, assim como os componentes eletrônicos necessários para a conversão e processamento dos sinais, base para o desenvolvimento do simulador, são apresentados a seguir.

2.1. Coração

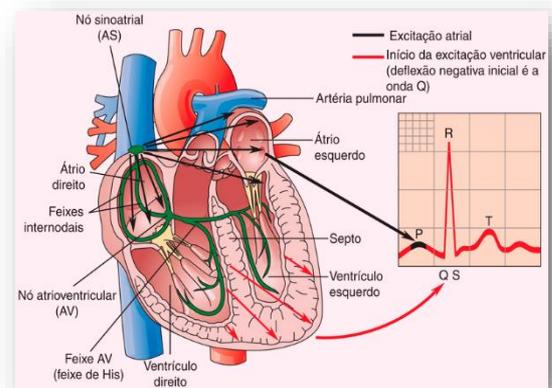
2.1.1 Princípios cardíacos

O coração é mais do que sístole e diástole, partes do bombeamento. Sua complexidade compreende diversos processos elétricos, biológicos, mecânicos e químicos. Recebe sinais do sistema nervoso por meio de fibras, o sistema nervoso autônomo simpático resumidamente é responsável pelo aumento da frequência cardíaca, pressão

arterial e débito cardíaco. Já as fibras parassimpáticas são responsáveis pelo retardo dos batimentos. Consome oxigênio, eletrólitos e outros compostos presentes no sangue durante seus processos. Os lados: direito, responsáveis pela circulação do sangue venoso e esquerdo pelo bombeamento do sangue arterial, são separados pelo septo. A quantidade de sangue bombeada é denominada débito cardíaco, em um adulto saudável é aproximadamente 5L/min (AEHLERT, 2019).

A polarização e despolarização produzem diferença de potencial (DDP), o aparelho de eletrocardiograma capta esse estímulo na ordem de milivolt (mV). O resultado é a onda exibida na figura 2, onde é possível observar a propagação (por meio das setas) do sinal partindo do nó sinusal, passando pelos átrios até os ventrículos e sua variação nas transições fica demonstrada no gráfico (AEHLERT, 2019).

Figura 2 Esquemático do sistema condução cardíaco



Fonte: (AEHLERT, 2019)

Ciclo cardíaco ou frequência designa a repetição dos movimentos causados pela contração e relaxamento dos músculos cardíacos, ditados pelo nó sinoatrial (THALER, 2013).

Em condições normais é esperado um

sinal menor que 25 mV e período de 0,1 s na onda P durante a despolarização e a contração atrial (LUTHRA, 2012).

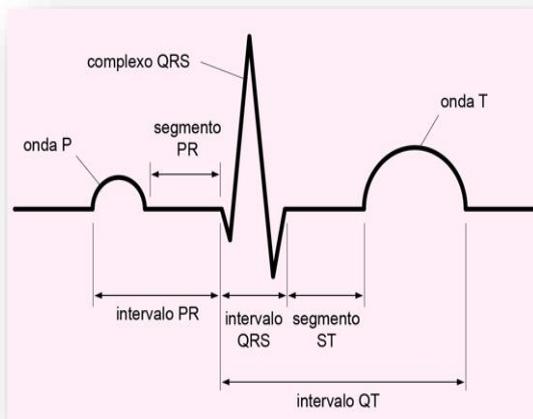
O sinal do segmento PR é contínuo e o ECG se mostra silencioso (THALER, 2013).

A onda Q deve apresentar período de 0,04 s e ser menor que 25% da onda R. A onda R na derivação V1 não deve exceder 0,4 mV e em V6 ser inferior a 2,5 mV. Por fim a onda S não deve exceder 0,7 mV. O segmento QRS apresenta período de 0,08 s (LUTHRA, 2012).

A onda T apresenta amplitude de 0,5mV e período de 0,4s e marca o início da repolarização ventricular (LUTHRA, 2012).

O sinal é composto por segmentos e intervalos, resultado da variação da amplitude e períodos durante o ciclo cardíaco explicitados na figura 3 (THALER, 2013).

Figura 3 O sinal cardíaco propriamente dito no modelo sinusal



Fonte: (THALER, 2013)

Cada trecho da onda elétrica representa o movimento mecânico ocorrido no coração (RULLI, 2020).

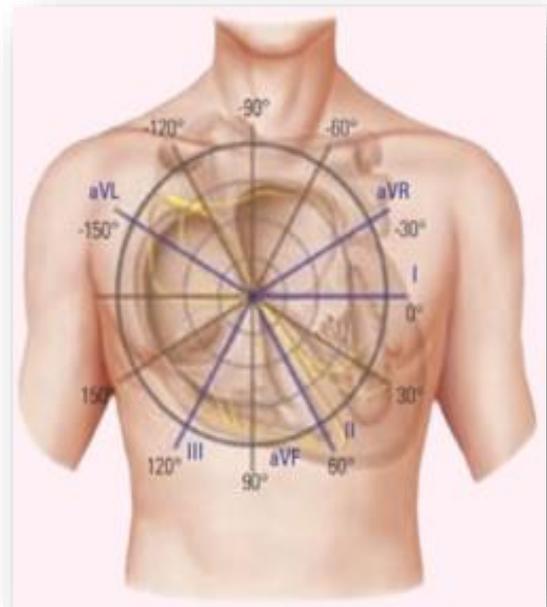
O coração é tridimensional e a análise depende dos pontos de observação escolhidos, ou seja, o observador tomando por base pontos pré-definidos mostrados na figura 4 terá condições de iniciar avaliação

do comportamento cardíaco. É importante ressaltar que apesar do objeto observado ser o mesmo, as derivações deverão estar corretamente identificadas e quanto mais derivações melhor a capacidade de conclusão (RULLI, 2020).

Os sinais são obtidos por meio de elétrodos ou garras jacarés instalados no corpo do paciente, sendo no mínimo 03 pontos. Para o correto contato, é imprescindível a limpeza da pele, ausência de pelos e presença de gel para melhorar o contato dos elétrodos (MOREIRA, 2016).

A figura 4 mostra as coordenadas das derivações e ao fundo o coração correlacionado suas partes com a derivação (RIERA e UCHIDA, 2011):

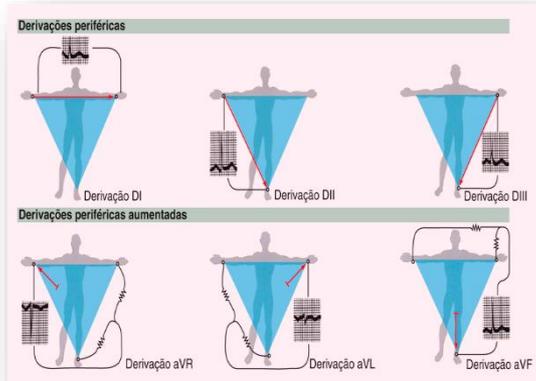
Figura 4 Plano frontal



Fonte: (RIERA e UCHIDA, 2011)

A derivação é o registro da diferença de potencial entre dois elétrodos, na figura 5 é exibido o posicionamento do elétrodo e os respectivos sinais esperados (AEHLERT, 2019).

Figura 5 Ondas x derivação

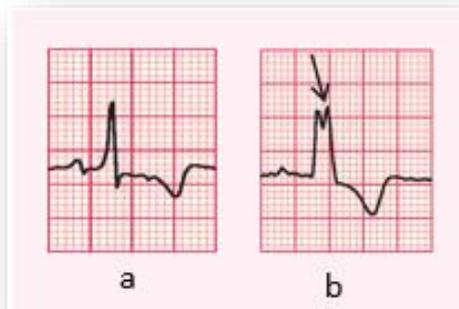


Fonte: (AEHLERT, 2019)

2.1.2. Arritmias

O sinal cardíaco pode sofrer deformações causadas por arritmias, como nas figuras 6 e 7 a seguir:

Figura 6 a) Hipertrofia ventricular esquerda e b) Hipertrofia ventricular direita, figuras adaptadas

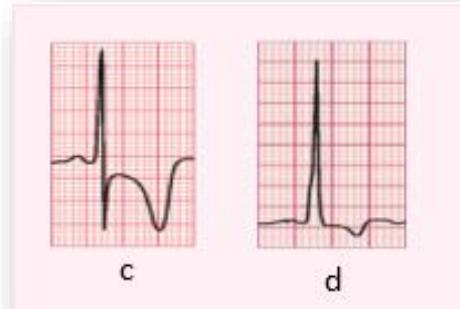


Fonte: (HAMPTON, 2014)

Considerando o ritmo sinusal das figuras apresentadas e de forma resumida temos (Hampton, 2014):

- 6 a) Onda R dominante em V1; ondas T invertidas nas derivações V1 – V4;
- 6 b) Intervalo PR discretamente prolongado; inversão de onda T em D1, Avl e V5-V6;

Figura 7 c) Bloqueio do ramo esquerdo e d) Hipertrofia ventricular esquerda, figuras adaptadas



Fonte: (HAMPTON, 2014)

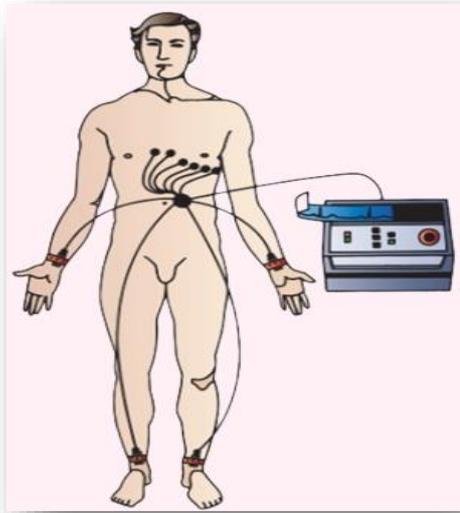
- 7 c) Onda T invertidas acentuadas nas derivações V3 – V6; Onda T invertida em V4;
- 7 d) Ondas R altas e S profundas; ondas T invertidas nas derivações D1, Avl e V5 – V6.

Tendo em vista alguns distúrbios fisiológicos, é necessário a escolha de filtros capazes de extrair ruídos e interferências provenientes da rede elétrica ou mesmo do ambiente para correta separação do sinal.

2.2. Funcionamento do ECG

O eletrocardiograma da figura 8 capta os sinais cardíacos a partir de eletrodos não invasivos instalados no paciente em repouso seguindo os protocolos-padrão de posicionamento. Este procedimento é importante para garantir a qualidade do registro, de modo a permitir a comparação entre exames (THALER, 2013).

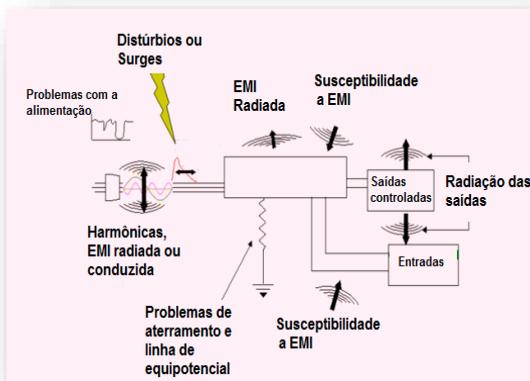
Figura 8 Distribuição dos elétrodos



Fonte: (THALER, 2013)

Normalmente alimentado pela rede elétrica alternada, ficando exposto a Interferências Eletromagnéticas (EMI), como mostra a figura 9. Esses são sinais indesejados, gerados por motores, acionamentos por chaveamento de cargas, lâmpadas fluorescentes, aparelhos de micro-ondas, equipamentos de comunicação, descargas atmosféricas, descargas eletrostáticas e outros (CASSIOLATO, 2021).

Figura 9 Fontes de distúrbios

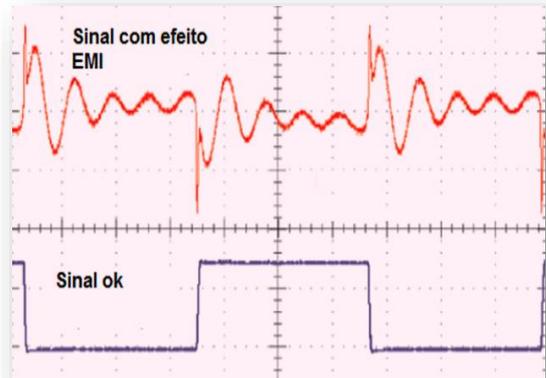


Fonte: (CASSIOLATO, 2021)

Variação de tensão; frequência harmônica; falta de conectividade com

aterramento, desacoplamentos, cargas não lineares são efeitos perturbadores de sinal, representados na figura 10 (CASSIOLATO, 2021)

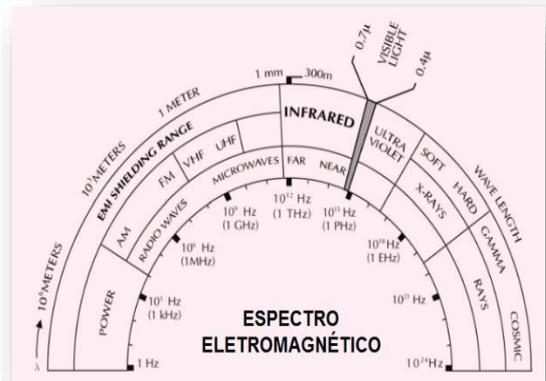
Figura 10 Efeitos da EMI



Fonte: (CASSIOLATO, 2021)

Os ruídos são sinais indesejados e o conhecimento prévio das frequências homologadas como mostrado na figura 11, é importante para saber quais dispositivos ou outras fontes passíveis de interferir no funcionamento (CASSIOLATO, 2021).

Figura 11 Espectro de frequência



Fonte: (CASSIOLATO, 2021)

Em um adulto a frequência cardíaca é em média entre 60 a 100 batimentos por minuto (bpm), que trazendo para o Sistema Internacional de Unidades (SI) equivale ao intervalo de 1 a 1,67Hz. Os filtros são necessários no tratamento do sinal com

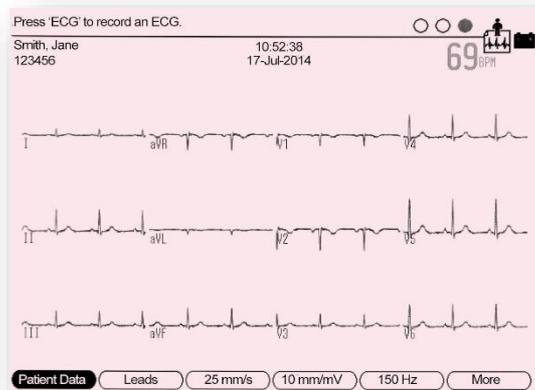
objetivo de separar eventuais ruídos com frequências maiores que a esperada, além de poder extrair a frequência da rede elétrica (50Hz ou 60Hz) (MOREIRA, 2016).

A *International Electrotechnical Commission* (IEC) estabelece padrões de fabricação operação próximo a fonte de radiofrequência (RF) e EMI descritos na EN60601-1-2 determinando os limites tolerados. A IEC 60601 também estabelece padrões para equipamentos hospitalares, peças e acessórios, bem como desempenhos essenciais que atendam o sistema elétrico médicos IEC 60601-1-1 (IEC, 2021).

Tomando como exemplo o ECG Marca GE modelo MAC600 exibido na figura 13 (GEHEALTHCARE, 2019):

- É capaz de amplificar magnitude dos sinais captados de 2,5 mm/mV até 20 mm/mV;
- Seu filtro atenua ruídos e possui opção de seleção de frequências de corte de 20 Hz, 40 Hz, 100 Hz e 150 Hz;
- Segue os padrões IEC 60601 de segurança;
- Imprimir relatório com gráfico e informações gerais, figura 12;
- Permite salvar exames em formato XML e PDF.

Figura 12 Sinal captado e processado



Fonte: Editada Manual de operação GE

Figura 13 MAC600



Fonte: Editada Manual de operação GE

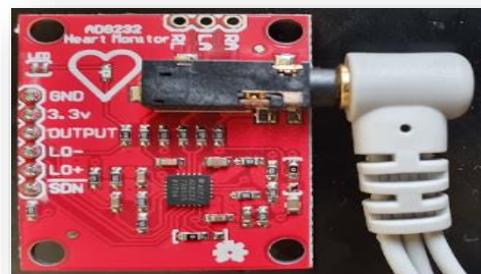
Os registros obtidos servem de base para desenvolvimento de emuladores. Os emuladores utilizados em testes de monitores cardíacos, dispensa a necessidade de pacientes durante os testes, garante a repetibilidade do sinal, permite simulações controladas e outros.

2.3. Conversor A/D e D/A

O sinal produzido pelo paciente é analógico e contínuo. O monitor através de conversor analógico para digital (A/D) transforma em sinal discreto no tempo, a amostra é quantizada em 2^n níveis possíveis, sendo n o número de bits de representação, por fim disponibiliza em sua saída o sinal digitalizado e formatado (HAUPT & DACHI, 2018).

O CI AD8232 da figura 14 possui em seu encapsulamento amplificadores operacionais para extrair os ruídos, amplificar e filtrar o sinal, por fim o sinal passa por conversor A/D interno. Este circuito foi utilizado para captação do sinal do simulador Fluke Prosim.

Figura 14 Módulo ECG AD8232 para Arduino



Fonte: O autor

Para o simulador é necessário fazer a conversão do sinal digital em sinal analógico por meio de conversor digital para analógico. No mercado há diversos conversores D/A encapsulados em circuitos integrados (CI). Os principais critérios de escolha são valor de fundo de escala e resolução. Quanto mais bits melhor a resolução (HAMPTON, 2014).

A tabela 1 é exemplo de operação do conversor D/A BCD 8421, onde apresenta a relação entre o sinal de entrada digital em bits e sua saída analógica em tensão (CAPUANO e IDOETA, 2019).

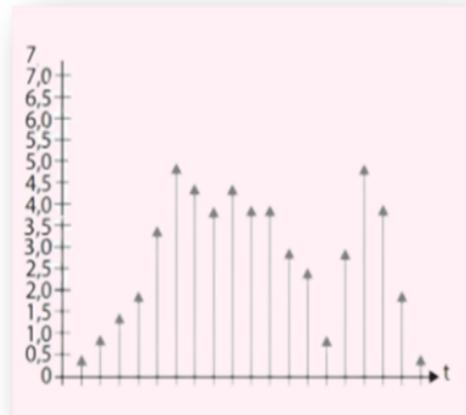
Tabela 1 Conversão D/A

Entrada digital				Saída analógica
A	B	C	D	V (mV)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Fonte: (CAPUANO e IDOETA, 2019)

Para um conversor de 4 bits teremos 16 níveis na saída de tensão demonstrados na figura 15. Ao traçar uma linha ligando as pontas da seta é possível ter a dimensão da qualidade da amostra obtida em relação ao sinal a ser recomposto (HAUPT e DACHI, 2018).

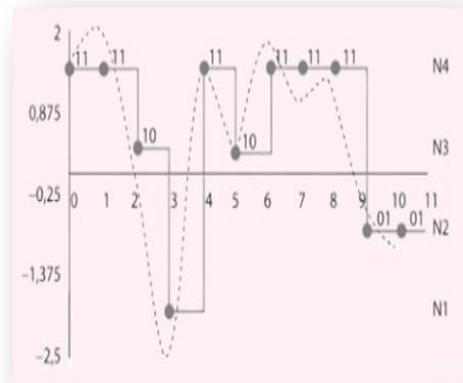
Figura 15 Quantidade de amostras do conversor D/A de 4 bits



Fonte:(HAUPT e DACHI, 2018)

Quando há presença de ruído, representado pela linha tracejada da figura 16, durante aquisição do sinal, não é seguida pela linha contínua devido a percepção dessa variação não ser instantânea (HAUPT e DACHI, 2018).

Figura 16 Sinal com ruído

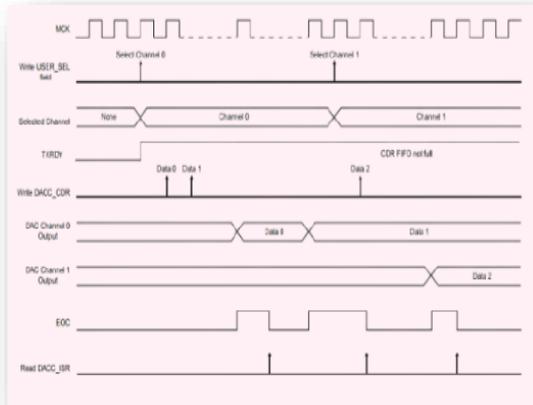


Fonte: (HAUPT e DACHI, 2018)

O microcontrolador Atmel AVR da família SAM3X series conta com duas portas D/A integradas com resolução de 12 bits em seu encapsulamento, conta com instruções RISC, 84 MHz de frequência de operação, 512 KB de memória Flash e 100 KB de SRAM. Seus canais são independentes, possui Trigger e utiliza o sistema FIFO (*First In First Out*) de

conversão. Na figura 17 é demonstrado a sequência de conversão do sinal (Atmel Corporation, 2015).

Figura 17 Sequência de conversão do sinal



Fonte: (Atmel Corporation, 2015)

2.4. Arduino

O microcontrolador Arduino é uma plataforma open *source* de prototipagem eletrônica e utiliza microcontroladores Atmel AVR em suas placas. Além disso, foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino (THOMSEN, 2014).

O Arduino foi criado em 2005 na Itália, no Instituto de Design Interativo de Ivrea pelos pesquisadores: David Cuartielles, David Mellis, Massimo Banzi e Tom Igoe, Gianluca Martino. Concebido para ser uma ferramenta acessível, de custo baixo, porém funcional (ARDUINO, 2018). E hoje incorporado a softwares como o Proteus, Matlab e outros.

Na tabela 2 é apresentado a relação dos Arduinos e suas portas, nela é possível ver quais portas IO disponíveis (Matlab MathWorks, 2021).

Tabela 2 Relação de modelos do Arduinos

Blocks	Target Hardware					
	Uno	Nano 3.0	Leonardo	Micro	Mega 2560/ADK	Due
Analog Output	N/A	N/A	N/A	N/A	N/A	DACC, DAC1
Digital Input	0-13	0-13	0-19	0-19	0-53	0-53
Digital Output	0-13	0-13	0-19	0-19	0-53	0-53
Analog Input	0-5	0-7	0-11	0-11	0-15	0-11
External Interrupt	2,3	2,3	0-3,7	0-3,7	2,3,10-21	0-53
SPI Slave Select (SS)	0-10	N/A	0-19	0-19	0-49, 53	4,10,52

Fonte: Adaptado (Matlab MathWorks, 2021)

O Arduino Due é um microcontrolador de 32 bits dos mais rápidos da família ARM, dentre os motivos está da escolha (THOMSEN, 2014).

Especificações do fabricante (Atmel Corporation, 2015):

- O *clock* de 84 MHz;
- Modos de manipulação das *threads*;
- Três estágios de *pipeline*;
- Arquitetura com diversos periféricos integrados, permitindo busca simultânea de instruções com ciclos de carga e armazenamento de dados;
- Baixa latência de entrada e saída;
- Alta performance do processador de 32 bits;
- Utiliza a tecnologia AMBA de alta velocidade e baixa latência de acesso a memória;
- Tem memória reservada proteção, para garantir a segurança de operações críticas;
- 100 KB de SRAM e 512 KB de memória *Flash*.

O Arduino Due possui diversos métodos de comunicação (Atmel Corporation, 2015):

- Interface Serial Periférica (SPI);
- Interface Two-Wire (TWI);
- Transmissão e recepção universal

assíncrona (UART);

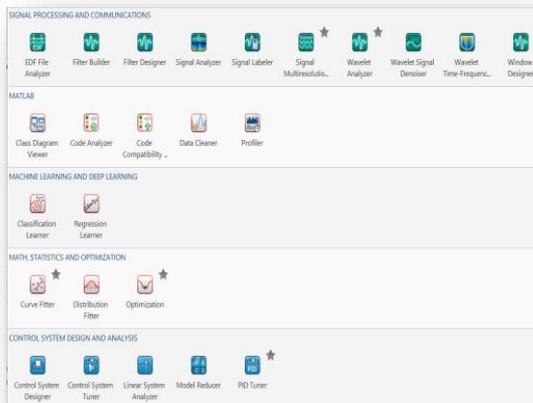
- Transmissão e recepção universal síncrona e assíncrona (USART);
- USB;
- Ethernet.

2.5. Matlab

O Matlab é uma poderosa ferramenta de cálculo e disponibiliza diversos cursos em seu site. Possui em sua base modelos de filtros, simuladores, interface com shields como Arduino e Raspberry. Tanto pelo Matlab quanto pela sua toolbox é possível comunicação com o Arduino (Matlab MathWorks, 2021).

O usuário pode desenvolver suas próprias funções, utilizar funções pré-existentes do sistema ou os seus aplicativos que abrange diversas áreas de conhecimento (Matlab MathWorks, 2021):

Figura 18 Menu de APP



Fonte: (Matlab MathWorks, 2021)

2.6. Sinais de ECG

Partindo do sinal segmentado é possível reconstituir o sinal de ECG e através do modelamento matemático reconstituí-lo (MADEIRO, 2013).

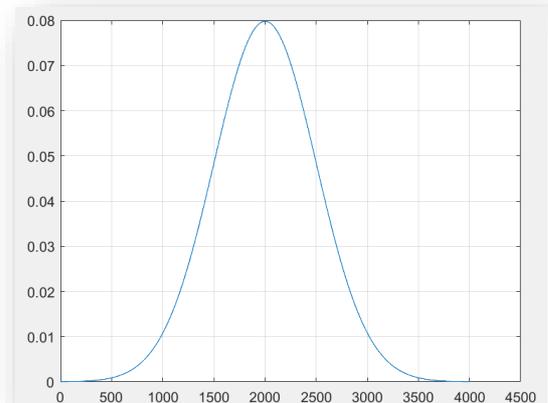
As ondas P e T podem ser expressas com modelamento de funções gaussianas. A fórmula (Eq. 1) utilizada para onda P é

(MADEIRO, 2013):

$$G(x) = A_G \cdot \frac{1}{\sqrt{2\pi}\sigma_1} \cdot \exp\left(\frac{-x^2}{2\sigma_1^2}\right) \quad \text{Eq. 1}$$

E sua onda correspondente é mostrada na figura 19:

Figura 19 Gaussiana de representação da onda P



Fonte: Elaborada pelo autor

Na onda T há uma inclinação resultante da repolarização dos ventrículos, diante disso, é necessário a multiplicação da tangente pela gaussiana (MADEIRO, 2013), como na Eq. 2:

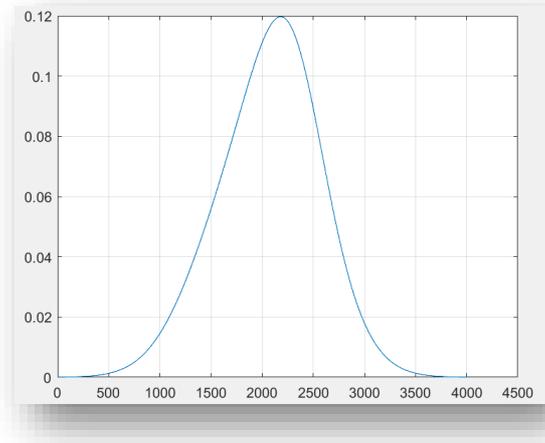
$$\rho(x) = f(G(x)) = \tan(\alpha) \cdot G(x) \quad \text{Eq. 2}$$

A conclusão da onda T ocorre na fórmula dada pela Eq. 3 a seguir (MADEIRO, 2013):

$$G'(x) = G(x - \rho(x)) \quad \text{Eq. 3}$$

Seu gráfico é exibido na figura 20:

Figura 20 Onda T



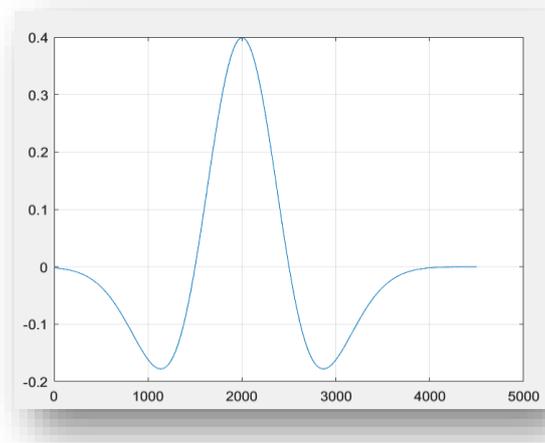
Fonte: Elaborada pelo autor

Para a terceira onda necessária que representa o complexo QRS temos a *wavelet*-mãe popularmente conhecida como Chapéu-Mexicano derivado de segunda ordem da gaussiana. Sua fórmula é dada na Eq. 4 (MADEIRO, 2013):

$$\psi(t) = \frac{1}{\sqrt{2\pi}} (1 - t^2) e^{-\frac{t^2}{2}} \quad \text{Eq. 4}$$

Seu gráfico é mostrado na figura

Figura 21 Chapéu-Mexicano, Complexo QRS



Fonte: Elaborado pelo autor

2.7. Wavelets

As transformadas de Wavelets ganharam destaque no processamento de sinais na engenharia, medicina e física

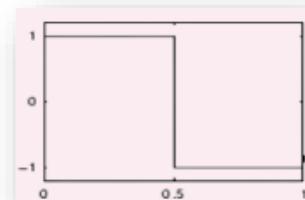
quântica devido a sua capacidade de representação e análise de alta complexidade de variáveis no domínio da frequência e do tempo (MADEIRO, 2013).

A partir de 1991 com Daubechies, volta a ganhar destaque na decomposição de sinal, levando vantagem sobre a FFT devido a baixa resolução temporal e resolução de frequência alta. Ideal para análise de sinais com grandes variações de duração, decomposição de sinal e construção de filtros (Diniz, Silva, & Netto, 2014).

Na análise com a transformadas de Wavelets encontram-se vários tipos de ondas usadas na representação do sinal. A seguir esses tipos de ondas:

Haar: é a wavelet de entrada, sendo a primeira e mais simples; é descontínua; simétrica; assemelha a função degrau; equivale a Daubechies db1, figura 22 (Matlab MathWorks, 2021).

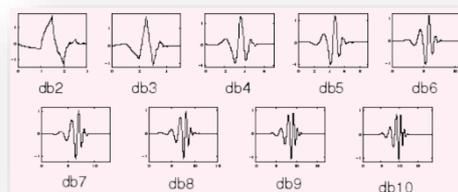
Figura 22 Haar



Fonte: (Matlab MathWorks, 2021)

Daubechies - dbN: possui fase não linear; energia concentrada perto do início de seus apoios; alto número de momento de fuga N para dada largura do suporte, figura 23 (Matlab MathWorks, 2021):

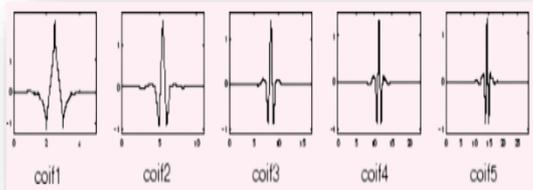
Figura 23 Daubechies



Fonte: (Matlab MathWorks, 2021)

Coiflets - $coifN$: função de escala e as ondas tem o mesmo número de momentos de fuga N , figura 24 (Matlab MathWorks, 2021);

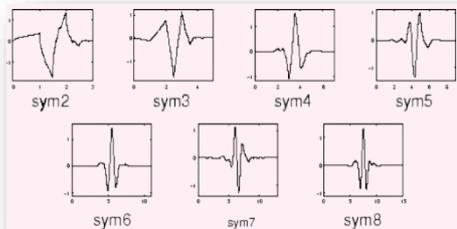
Figura 24 Coiflets



Fonte: (Matlab MathWorks, 2021)

Symlets - $symN$: menos assimétrica; apresenta semelhanças com a Daubechies, figura 25 (Matlab MathWorks, 2021)

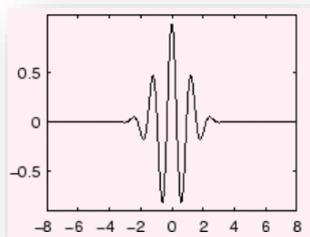
Figura 25 Symlets



Fonte: (Matlab MathWorks, 2021)

Morlet - $morl$: esta wavelet não tem função de escala, mas é explícita. Figura 26 (Matlab MathWorks, 2021):

Figura 26 Morlet

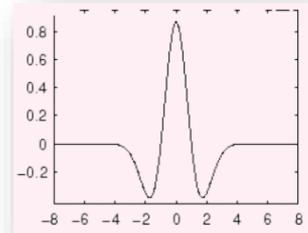


Fonte: (Matlab MathWorks, 2021)

Chapéu-Mexicano - $mexh$: não tem função escalar e é derivada de uma função

gaussiana. Também conhecida como Ricker wavelet, figura 27 (Matlab MathWorks, 2021):

Figura 27 Chapéu-Mexicano



Fonte: (Matlab MathWorks, 2021)

A tabela 3 apresenta as principais características das wavelets:

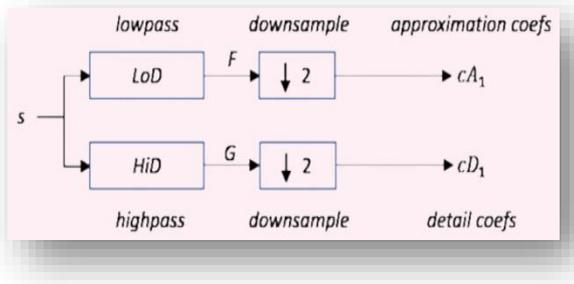
Tabela 3 Famílias wavelets e suas propriedades

Propriedades/ Tipos	Haar	dbN	coifN	symN	morl	mexh
Infinitamente regular					✓	✓
Com suporte compacto						
ortogonal	✓	✓	✓	✓	✓	✓
Simetria	✓				✓	✓
Análise ortogonal	✓	✓	✓	✓		
Filtros FIR	✓	✓	✓	✓		
Transformada contínua	✓	✓	✓	✓	✓	✓
Transformada discreta	✓	✓	✓	✓		
Algoritmo rápido	✓	✓	✓	✓		

Fonte: Adaptada pelo autor (Matlab MathWorks, 2021)

O sinal de tamanho N a cada nível é decomposto em duas partes resultantes da convolução de s ao passar pelos filtros passa baixa e passa alta, produzindo o coeficiente de aproximação (cAN) e coeficiente de detalhe(cDN), com máximo de estágios representados por \log_2N o algoritmo é expresso na figura 28 (Matlab MathWorks, 2021).

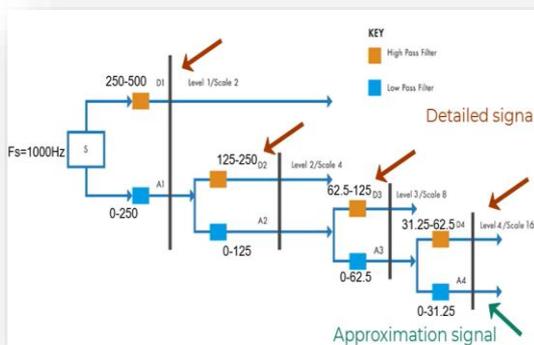
Figura 28 Decomposição



Fonte: (Matlab MathWorks, 2021)

A figura 29 e a tabela 4, a seguir, expressam as divisões ocorridas nas passagens de nível. A cada divisão a amplitude do sinal dobra de tamanho (Matlab MathWorks, 2021).

Figura 29 Exemplo da divisão do sinal



Fonte: (Matlab MathWorks, 2021)

Tabela 4 Escala x Frequência

Escala Wavelet	2	4	8	16
Frequência Equivalente (Feq)	Feq/2	Feq/4	Feq/8	Feq/16

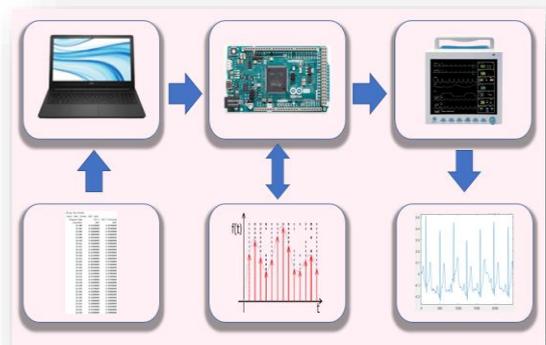
Fonte: Adaptado pelo autor (Matlab MathWorks, 2021)

O Matlab permite fazer filtros digitais, aplicar ganho, ajuste de offset e vários outros recursos nativos para melhora da qualidade do sinal. Quanto maior a quantidade de amostras melhor será exibição do sinal, em contrapartida demanda maior capacidade de processamento.

3 METODOLOGIA DO TRABALHO

Este trabalho é uma pesquisa aplicada que tem por objetivo implementar um simulador capaz de reproduzir o sinal cardíaco por meio do Arduino Due conhecido pelos técnicos e engenheiros dando-lhes a flexibilidade de seleção das frequências de testes com sinal com menor índice de ruído e repetibilidade. Antes de liberar o monitor ou eletrocardiógrafo para aplicação no paciente é necessário a submissão a rotina de testes, dentre eles a utilização do simulador para aferição da frequência e formato de onda conhecido.

Figura 30 Diagrama de conexão



Fonte: Elaborado pelo autor

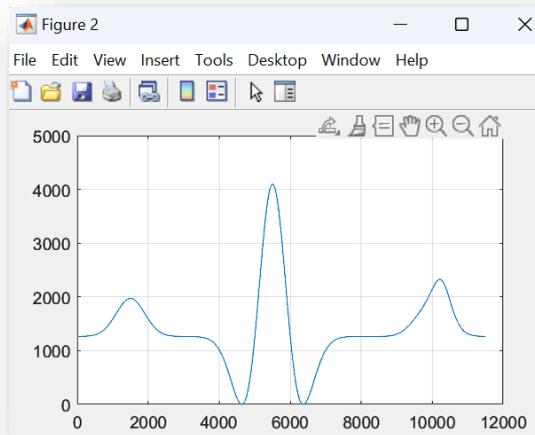
Foram realizados ensaios com sinais captados de diversas fontes, a pesquisa visou encontrar o sinal que menos demandasse manipulações, com pontos de amostras equidistantes e melhor qualidade do ponto de vista de ruídos. As fontes estudadas são as seguintes:

- Sinais tabulados obtidos através do site <https://archive.physionet.org/cgi-bin/atm/ATM>, (PhysioNet, 2021), em arquivos formatados nas extensões .TXT e .XLSX. Alguns modelos de sinais apresentam pequenos ruídos, porém há opção com sinal filtrado.;
- Como mencionado anteriormente,

foi realizada a captação do sinal utilizando o simulador da Fluke e a shield de ECG do Arduino, no entanto a resolução de apenas 8 bits limitou a qualidade do sinal;

- Por fim, para obtenção do sinal foram utilizados recursos matemáticos com base em teses e o Matlab. Por meio da concatenação dos gráficos formada por uma Gaussiana representando a onda P, o complexo QRS descrito pelo Chapéu Mexicano e a onda T, Gaussiana + reta tangente, obteve-se para o simulador do projeto, o resultado apresentado na figura 31:

Figura 31 Sinal sintetizado



Fonte: Elaborado pelo autor

Superada a pesquisa de qualidade do sinal, o próximo passo é a pesquisa quantitativa com a medição e comparação do sinal resultante.

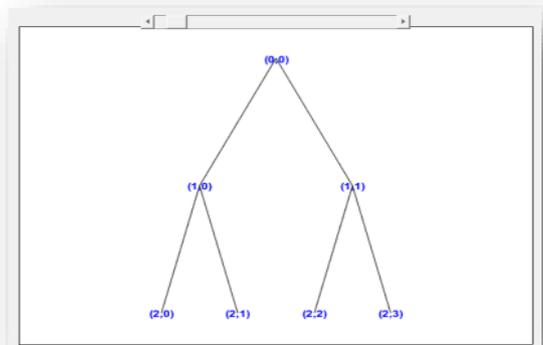
De posse do sinal sintetizado que apresentou melhor desempenho para os testes, foram aplicados ganhos proporcionais buscando melhor reprodução da onda e aproveitamento da amplitude/resolução da porta analógica do Arduino. A verificação da qualidade foi feita com aplicativo Wavelet *Analayzer* do Matlab. A ferramenta oferece diversos recursos, dentre eles:

- Escolha da wavelet, neste caso foi

utilizada a *Haar* com base nas possibilidades de manipulação contínua e discreta, simetria, rapidez do algoritmo e outras;

- Obtenção de árvore de complexidade, sendo possível avaliar qual a melhor árvore ou melhor nível. No projeto, o nível de melhor desempenho foi o 2 como mostrado na figura 32:

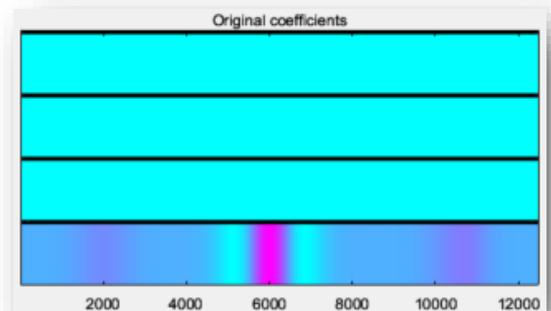
Figura 32 Resultante da Best Tree



Fonte: Elaborada pelo autor

- Avaliação do comportamento da frequência por meio do espectro de cores, figura 33:

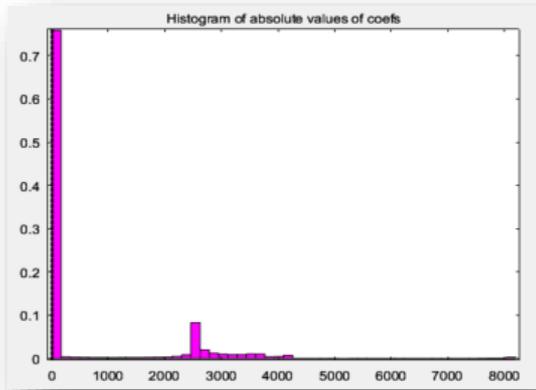
Figura 33 Espectro de frequência



Fonte: Elaborada pelo autor

- O gráfico do espectro de frequência permite visualizar o comportamento do sinal e suas componentes, como visto na figura 34:

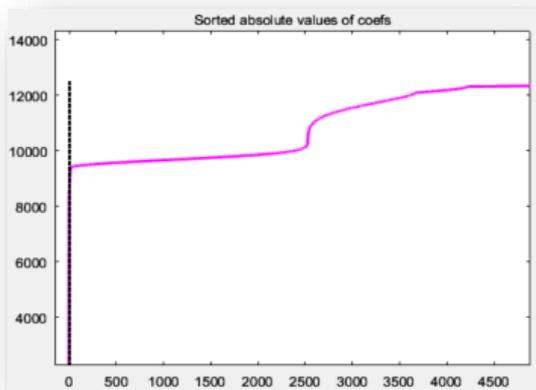
Figura 34 FFT do sinal de ECG



Fonte: Elaborada pelo autor

- A Figura 35 mostra o comportamento do sinal, ideal para dimensionamento e criação de filtro:

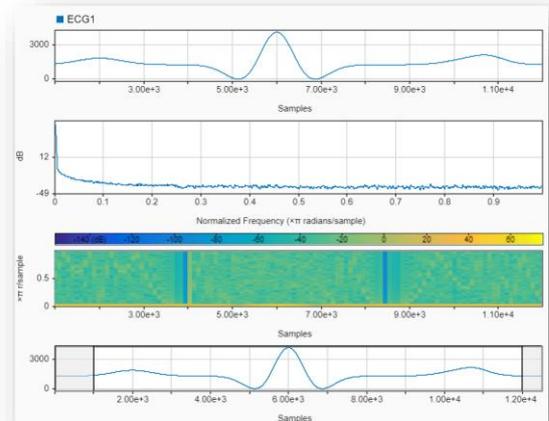
Figura 35 Concatenação entre as ondas



Fonte: Elaborada pelo autor

O ponto de concatenação mostrado nos sinais das figuras podem ser melhor analisados com o módulo de análise de sinal nativo do Matlab, figura 36:

Figura 36 Analisador de sinal do Matlab

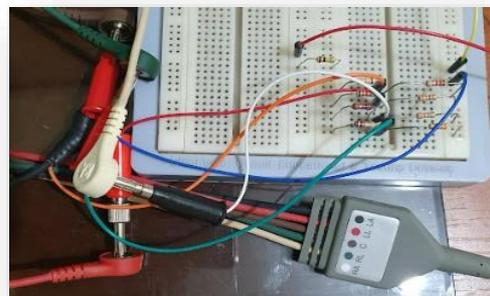


Fonte: O autor

Após adequação do sinal de ECG com 12503 amostras, a segunda etapa foi validar o funcionamento da porta D/A do Arduino Due. Para realização dos testes foram utilizados: um monitor multiparametros, o Arduino Due, e o computador com a IDE do Arduino.

Para a conexão foi necessário circuito intermediário para redução da tensão de saída da porta DAC0 (até 3,6 volts) para compatibilidade com a entrada do monitor (25 mV), neste caso trata-se de um divisor de tensão. A figura 37 apresenta esse circuito:

Figura 37 Interface Simulador x Monitor

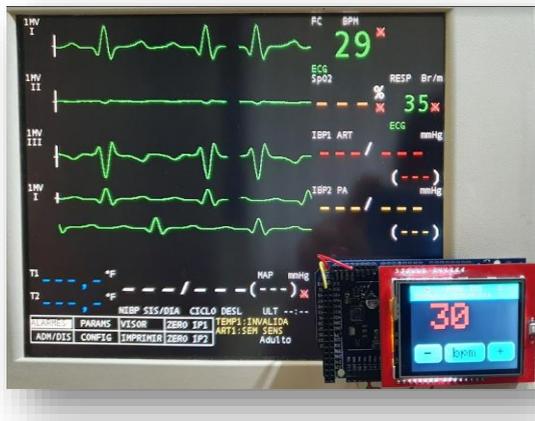


Fonte: Elaborada pelo autor

O sinal com 12503 serviu de base para derivações das frequências. Quanto maior for o número de amostras, mais próximo será do sinal real.

Esse fato implica em atenção especial ao código de programa, apesar da utilização de um display para interface com usuário, figura 38, o foco foi a disponibilidade do sinal na porta analógica evitando rotinas desnecessárias e separando o programa principal de sub-rotinas.

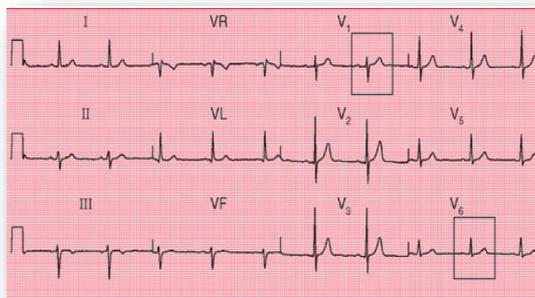
Figura 38 Monitor x Simulador



Fonte: Elaborada pelo autor

O formato das ondas esperado para um ECG normal é apresentado na figura 39:

Figura 39 ECG normal



Fonte: (Hampton, 2014)

Para a realização dos ensaios foi utilizado um monitor que recentemente passou por certificação. O relatório aponta a utilização do simulador da Fluke, também certificado por laboratório homologado pelo INMETRO de modo a fechar a cadeia de rastreabilidade (a certificação ocorre ao menos 01 vez ao ano

ou toda vez que o dispositivo passar por manutenção).

Como exibido na figura 38, o valor ajustado no display do simulador está inferior a 5%. Para a obtenção de tal resultado, foram realizados ensaios utilizando a função *milis* do Arduino para estimar o tempo médio de execução de cada loop, redução de instruções deixando somente as estritamente necessárias, evitando a repetição de cálculos e particionando o programa em funções de modo que o processador passe a maior parte do tempo na reprodução do sinal. Os resultados dessas ações são representados nas tabelas 5 e 6.

Ao observar os resultados obtidos com a função *milis* é possível constatar a estabilidade do processamento do Arduino. O valor mínimo expresso na tabela variou apenas de uma unidade na maioria dos ensaios; apenas na frequência de 120 bpm não ocorreu variação. A tabela 5 mostra a frequência selecionada, o tempo em milissegundos gasto para execução de 1 bpm considerando o maior tempo de ciclo, o tempo real esperado e a diferença percentual entre eles mostrado na tabela 5:

Tabela 5 Tempo de execução x Tempo estimado

Seleção (bpm)	Execução (ms)	Tempo (ms)	Erro
20	3107 + 1	3000	4%
30	2072 + 1	2000	4%
40	1554 + 1	1500	4%
50	1243 + 1	1200	4%
60	1036 + 1	1000	4%
70	888 + 1	857	4%
80	777 + 1	750	4%
90	691 + 1	667	4%
100	622 + 1	600	4%
110	565 + 1	545	4%
120	518 + 0	500	4%

Fonte: Elaborado pelo autor

Na tabela 6 tem se o valor da frequência selecionado pelo operador, o valor lido pelo monitor e sua variação percentual

tomando como base o valor selecionado. O critério utilizado no sistema de metrologia, onde são efetuados **no mínimo 3** leituras para levantamento do erro, desvio padrão e variância não foi possível implementar pois o valor lido se manteve fixo.

Tabela 6 Valor selecionado x Valor lido

<u>Seleção (bpm)</u>	<u>Valor Lido</u>	<u>Erro</u>
20	20	0%
30	29	-3%
40	39	-3%
50	48	-4%
60	58	-3%
70	67	-4%
80	77	-4%
90	87	-3%
100	96	-4%
110	106	-4%
120	115	-4%

Fonte: Frequência selecionada x Valor lido

Considerando os valores dos erros obtidos temos:

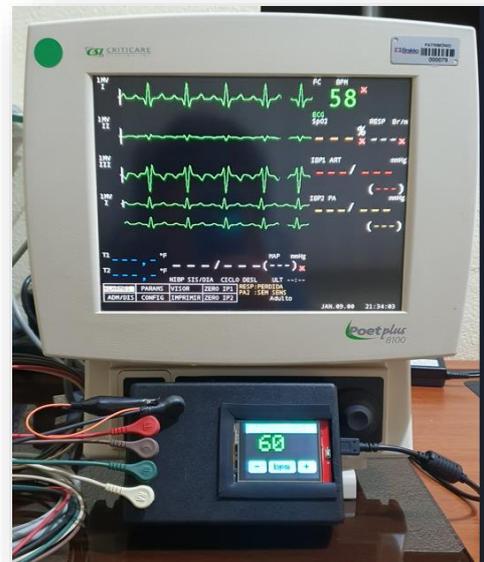
Tabela 7 Estatísticas do erro

<u>Tipo</u>	<u>Valor</u>
Tamanho	11
Mínimo	-4%
Máximo	0%
Amplitude	4%
Média	-3%
Desvio	0,01150
Variância	0,00013

Fonte: Elaborado pelo autor

Uma vez concluídos os testes, o circuito divisor de tensão foi montado em placa de circuito impresso perfurada, Arduino Due, display LCD e conexões para o cabo de ECG do simulador foram acondicionados em caixa patola, conforme mostrado na figura 40:

Figura 40 Simulador, protótipo



Fonte: Elaborado pelo autor

A alimentação do circuito pode ser através da porta USB do computador ou utilizando fonte externa.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

A qualidade do sinal é crítica quando se fala em monitores, por se tratar de sinais de baixa amplitude e sujeitos a distorções ou falhas que podem culminar em diagnóstico errado do paciente. Diante disso é importante contar com simulador de qualidade e confiança durante o processo de aferição.

O Matlab é uma excelente ferramenta de desenvolvimento, foi fundamental para implementação, ajustes e validação do sinal de ECG. Para o projeto foi adquirida a versão estudante juntamente com módulo das Wavelets. No sinal tabulado obtido na internet e no sinal captado de simulador apresentou ótimos recursos de filtros para tratamento. No entanto ao utilizar o sinal sintetizado, os filtros aplicados não produziram resultados significativos. Já a manipulação do sinal na concatenação das ondas, amplificação e outros deram

resultados bastante satisfatórios. Além de possibilitar a compressão do sinal.

Os resultados obtidos apontam ser viável a implementação, principalmente no que se refere a estabilidade e repetibilidade do sistema apresentado. A validação final é condicionada ao envio a um laboratório de metrologia reconhecido pelo INMETRO para ensaios dos valores e formatos de onda.

Como ferramenta, a utilização do Arduino Due apresenta limitação quando comparado a simuladores comerciais; estes possuem outros módulos de simulação dos sinais vitais que permitem a reprodução simultânea; e no caso do protótipo é possível desenvolver demais módulos, porém a reprodução é sequencial. O que não chega a ser um agravante, visto que no processo de calibração o técnico faz 3 leituras sequencias de um único parâmetro.

Apesar do protótipo estar funcional, um dos pontos de melhora decorre na implementação de eletrônica analógica, com uso de amplificador operacional para melhor isolamento entre as partes e utilização de capacitores na atenuação dos *ripples* permitindo a redução de amostras a serem processadas.

5 CONSIDERAÇÕES FINAIS

Os objetivos iniciais foram alcançados e as dificuldades na transição de sinal digital para analógico e vice-versa foram superados com conhecimentos adquiridos ao longo do curso de engenharia. O Matlab possibilitou o processamento e análise do sinal, reduzindo a carga do Arduino.

Os conceitos obtidos na seleção/produção do sinal, tratamento (filtragem, amplificação e demais adequações) e processamento são replicáveis, possibilitando a utilização de outros *hardwares*.

Diante do exposto, o protótipo pode evoluir para uma ferramenta de menor custo e se mostrando como opção para testes de bancada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, também a minha família pelo apoio e suporte ao longo dessa caminhada. Aos amigos que estiveram comigo nesses anos. Por fim ao CEUB e professores pelo conhecimento compartilhado.

REFERÊNCIAS

- Aehlert, B. J. (2019). *Manual de ECG* (6ª ed.). GEN Guanabara Koogan.
- Arduino. (2018). *Arduino*. Obtido em 9 de Setembro de 2021, de <https://www.arduino.cc>: <https://www.arduino.cc/en/Guide/Introduction>
- Atmel Corporation. (2015). *SAM3X / SAM3A Series, Datasheet*.
- Capuano, F. G., & Idoeta, I. V. (2019). *Elementos de Eletrônica Digital* (42ª ed.). São Paulo: Érica.
- Cassiolato, C. (2021). *Smar Technoly Company*. Obtido em 30 de Outubro de 2021, de <https://www.smar.com/>: <https://www.smar.com/brasil/artigo-tecnico/emi-interferencia-eletromagnetica-em-instalacoes-industriais-e-muito-mais>
- Diniz, P. S., Silva, E. A., & Netto, S. L. (2014). *Processamento Digital de Sinais*. BOOKMAN EDITORA LTDA.
- Drake, C. (2021). *Cmos Drake*. Obtido em 17 de Outubro de 2021, de <https://cmosdrake.com.br>: <https://cmosdrake.com.br/quem-somos/>
- Fluke Corporation. (2021). *Ferramentas de qualidade de energia*. Obtido em 18 de Abril de 2021, de <http://www.fluke.com>: <https://www.flukebiomedical.com/products/biomedical-test-equipment/patient-monitor-simulators/prosim-8-vital-signs-patient-simulator>
- GEHealthcare, W. (2019). *Sistema de Análise de ECG em Repouso, Manual do operador* (2047426-022 Revisão G ed., Vol. Versão do Programa 1.0).
- Hampton, J. R. (2014). *ECG na Prática* (6ª ed.). GEN Guanabara Koogan.
- HAUPT, A. G., & DACHI, É. P. (2018). *Eletrônica Digital*. EDGAR BLUCHER.
- Haupt, A., & Dachi, É. (2018). *Eletrônica Digital* (1ª ed.). Edgard Blucher Ltda.
- HEALTHCARE, G. (2021). <https://www.gehealthcare.com.br>. Obtido

- em 27 de Outubro de 2021, de GE Healthcare:
<https://www.gehealthcare.com.br/Products/Diagnostic%20ECG/Resting%20ECG>
- HP, H. P. (Outubro de 1981). HEWLETT PACKARD JOURNAL. *HEWLETT PACKARD JOURNAL*, 32.
- Hudson, A. (2008). *Estadão*. Obtido em 12 de Outubro de 2021, de <https://economia.estadao.com.br>:
<https://economia.estadao.com.br/noticias/negocios,philips-compra-dixtal-em-busca-de-crescimento-na-area-de-saude,171889>
- IEC. (2021). Obtido em Outubro de 2021, de <https://www.iec.ch/>:
<https://www.iec.ch/homepage>
- Luthra, A. (2012). *ECG Made Easy* (4ª ed.). Nova Delhi: Jaypee Brothers Medical Publishers.
- M., S. (2013). *Learn ECG in a Day*. Jaypee Brothers Pvt. Ltd.
- Madeiro, J. P. (2013). Fortaleza, Ceará, Brasil.
- Matlab MathWorks. (2021). *Matlab MathWorks*. Obtido em 03 de Agosto de 2021, de <https://www.mathworks.com>:
https://www.mathworks.com/support.html?s_tid=gn_supp
- Moreira, T. (2016). <https://www.embarcados.com.br/>. Obtido em 25 de 04 de 2021, de Embarcados:
<https://www.embarcados.com.br/projeto-eletrocardiografo/>
- PhysioNet. (11 de 2021). *PhysioNet*. Obtido em 27 de Agosto de 2021, de <https://archive.physionet.org>:
<https://archive.physionet.org/cgi-bin/atm/ATM>
- PhysioNet. (12 de 2021). *PhysioNet*. Obtido em 27 de Agosto de 2021, de <https://archive.physionet.org>:
<https://archive.physionet.org/cgi-bin/atm/ATM>
- R&DMEDIQ. (2021). <https://www.rdmediq.com.br/>. Obtido em 17 de Abril de 2021, de <https://www.rdmediq.com.br/>:
https://www.rdmediq.com.br/simulador_h30/
- Riera, A. P., & Uchida, A. (2011). *Eletrocardiograma, Teoria e Prática*. Barueri, SP: Manoele.
- Rulli, D. E. (2020). *Médico na Prática*. Obtido em 24 de Abril de 2021, de <https://www.youtube.com/channel/UCXI3HeNrkbAy1Bm7CrvEoQ>:
<https://www.youtube.com/watch?v=3IXzxbqbrWg&t=61s>
- Thaler, M. S. (2013). *ECG Essencial, Eletrocardiograma na Prática Diária* (7ª ed.). Artmed.
- Thomsen, A. (2014). *Filipeflop*. Obtido em 17 de Outubro de 2021, de <https://www.filipeflop.com/>:
<https://www.filipeflop.com/blog/o-que-e-arduino/>

Anexo

Matlab

```
clear
clc
s = 1; % Desvio padrão
Fs = 500; % Frequência de amostragem
t11 = -4; % Início da Gaussiana
t12 = 4; % Fim da Gaussiana
x = t11: 1/Fs: t12; % Intervalo

%--- Decomposição da fórmula-----%
num = -(x).^2; % Numerador da
exponencial
den = 2 * s ^ 2; % Denominador da
exponencial
Ag = 0.2; % Ganho da função
a = 1/ (sqrt (2 * pi * s ^2)); % Fração
multiplicadora da exponencial

%--- Formula reconstituída -----%
Gx = Ag * (a * exp (num/ den)); %
Forma de onda P
% figure
% plot(x,Gx)

r = (tan (pi/ 2.3)); % Reta tangente

px = r.*Gx; % Coeficiente para geração da
onda

x2 = x - px;

num = -(x2).^2;
Ag = 0.3;
Gx1 = Ag * (a * exp (num/ den)); %
Derivada para produção da onda T

% --- Complexo QRS -----%
t2 = -4: 1/Fs: 5;
b = 1/sqrt (2*pi);
c = 1 - (t2.^2);
d = exp (-t2.^2/2);

Gx2 = (b * c) .* d ;

%---Agrupamento dos sinais
ECG = horzcat (Gx ,Gx2);
```

```
ECG = horzcat (ECG ,Gx1);
```

%---Ajuste para operação em 12 bits no Arduino

```
t = length(ECG);
t1 = 1:1:t;
nBit = (2 ^ 12) - 1;
menorvalor = min (ECG);
maiorvalor = max(ECG);
```

```
for i = 1: t
    ECG1 (i,1) = round ((ECG(i) -
menorvalor)*(nBit/(maiorvalor -
menorvalor)));
end
```

Arduino

```
//Declaração dos pinos LCD TFT
#define LCD_CS A3 // Display Chip
Select pino Analogico 3
#define LCD_CD A2 // Display
Command/Data pino Analogico 2
#define LCD_WR A1 // Display LCD
Write pino Analogico 1
#define LCD_RD A0 // Display LCD
Read pino Analogico 0
#define LCD_RESET A4 // Reset pino
Analogico
```

```
//Pressão no touch
#define MINPRESSURE 200
#define MAXPRESSURE 1000
```

```
//Referência da tela de toque
// X
#define TS_MINX 129
#define TS_MAXX 886
// Y
#define TS_MINY 856
#define TS_MAXY 140
```

```
//Pinos Touchscreen
#define YP A3
#define YM 9
#define XM A2
#define XP 8
```

```
// Inicialização das bibliotecas
```



```
#include "ECG60.h"
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;
#include <TouchScreen.h>

//Variáveis global
int pixel_x, pixel_y; //Touch_getXY()
updates global vars
int valor = 20;
int i;
bool Press;
int tamanho;
int tempo1;
int passo;

TouchScreen ts = TouchScreen(XP, YP,
XM, YM, 300);
Adafruit_GFX_Button btn_Menos,
btn_Mais, btn_bpm, btn_valor;
//Definição dos botões

void Teclado();// Função futura
void visor ();
unsigned long testFillScreen();

bool Touch_getXY(void)
{
  TSPoint p = ts.getPoint();
  pinMode(YP, OUTPUT);
  pinMode(XM, OUTPUT);
  digitalWrite(YP, HIGH);
  digitalWrite(XM, HIGH);
  bool pressed = (p.z > MINPRESSURE
&& p.z < MAXPRESSURE);
  if (p.z > ts.pressureThreshhold) {
    pixel_x = map(p.y, TS_MAXX,
TS_MINX, 0, 320 );
    pixel_y = map(p.x, TS_MAXY,
TS_MINY, 0, 240 );
  }
  return pressed;
}

void setup(void) {
  Serial.begin(115200);
  tft.reset();
  uint16_t ID = tft.readID(); //
  tft.begin(ID);
  tft.setRotation(3); //Posição paisagem
```

```
pinMode (DAC0, OUTPUT); //Pinos de
saída de sinal
pinMode (DAC1, OUTPUT);
analogWriteResolution(12);

//Cabeçalho
tft.fillScreen(BLACK);
tft.fillRect(0, 0, 320, 40, CYAN);
tft.setTextColor(RED);
tft.setTextSize(2);
tft.setCursor(20, 2);
tft.println(" * SINAL ECG * ");
tft.print("Ajuste a frequencia em bpm:
");

//Escrita do valor selecionado
tft.setTextColor(RED);
tft.setTextSize(10);
tft.setCursor(60, 50);
tft.print(valor);

// Criação dos botões de seleção de
frequência
btn_Menos.initButton(&tft, 50, 190, 80,
50, WHITE, CYAN, BLACK, " - ", 4); //
Botão Menos
btn_Mais.initButton(&tft, 270, 190, 80,
50, WHITE, CYAN, BLACK, " + ", 4); //
Botão Mais
btn_bpm.initButton(&tft, 160, 190, 120,
50, WHITE, CYAN, BLACK, " bpm ",
4);
btn_Menos.drawButton(false);
btn_Mais.drawButton(false);
btn_bpm.drawButton(false);

tamanho = sizeof(amplitude)/ sizeof(int);
// Definição do tamanho do vetor a ser
utilizado.

}

void loop (void) {
  Press = false;
  passo = valor/10;
  while (Press == false) {
    tempo1 = millis();
    for (i = 1; i < tamanho; i = i + passo) {
      Serial.println(amplitude[i]);
```

```

    analogWrite (DAC0,amplitude[i]);
    if (Press == false) {
        Teclado ();
    }
}
tempo1 = (millis() - tempo1);
Serial.print("Time: ");
Serial.println(tempo1);
}
}

void visor(void) {
    tft.fillRect(30, 50, 320, 100, BLACK);
//Apaga o valor selecionado
anteriormente.
    tft.setTextColor(GREEN); //Cor da letra
do valor selecionado
    tft.setTextSize(10); //Tamanho da letra
do valor selecionado
    tft.setCursor(60, 50); //Posicionamento
do cursor para início da escrita
    tft.print(valor); //Valor selecionado.
    loop ();
}

void Teclado () {
    bool down = Touch_getXY();
    btn_Menos.press(down &&
btn_Menos.contains( pixel_x, pixel_y));
    btn_Mais.press(down &&
btn_Mais.contains( pixel_x, pixel_y));
    if (btn_Menos.justPressed()) { //
Decrementa a frequência
        btn_Menos.drawButton(true);
        pinMode(XM, OUTPUT);
        pinMode(YP, OUTPUT);
        valor = valor - 10;
        if (valor < 20) {
            valor = 20;
        }
        btn_Menos.drawButton(false);
        visor();
    }
    if (btn_Mais.justPressed()) { //
Incrementa a frequência
        btn_Mais.drawButton(true);
        pinMode(XM, OUTPUT);
        pinMode(YP, OUTPUT);
        valor = valor + 10;
        btn_Mais.drawButton(false);
        visor();
    }
    return;
}
}

```