Universidade Federal
do Rio de Janeiro
Escola Politécnica

DESENVOLVIMENTO DE UM EQUIPAMENTO DE BAIXO CUSTO PARA MEDIÇÃO DE PARÂMETROS DA FREQUÊNCIA RESPIRATÓRIA COM APLICAÇÃO EM NEUROCIÊNCIA COMPORTAMENTAL HUMANA.

Bruno Gomes Reis

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Carlos José Ribas D'Avila
Tiago Arruda Sanchez

Rio de Janeiro
Agosto de 2019

DESENVOLVIMENTO DE UM EQUIPAMENTO DE BAIXO CUSTO PARA MEDIÇÃO DE PARÂMETROS DA FREQUÊNCIA RESPIRATÓRIA COM APLICAÇÃO EM NEUROCIÊNCIA COMPORTAMENTAL HUMANA.

Bruno Gomes Reis

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO.

Examinado por:

Prof. Nome do Primeiro Examinador Sobrenome, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, Ph.D.

Prof. Nome do Terceiro Examinador Sobrenome, D.Sc.

Prof. Nome do Quarto Examinador Sobrenome, Ph.D.

Prof. Nome do Quinto Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2019

Reis, Bruno Gomes

Desenvolvimento de um equipamento de baixo custo para medição de parâmetros da frequência respiratória com aplicação em neurociência comportamental humana./Bruno Gomes Reis. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2019.

XIV, 44 p.: il.; 29, 7cm.

Orientadores: Carlos José Ribas D'Avila

Tiago Arruda Sanchez

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Eletrônica e de Computação, 2019.

Referências Bibliográficas: p. 39 – 39.

1. Behavioral Neuroscience. 2. Respiratory Frequency . 3. Engenharia Eletrônica e de Computação. I. D'Avila, Carlos José Ribas *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Eletrônica e de Computação. III. Título.

Dedico este trabalho às minhas famílias de sangue e coração.

Agradecimentos

Gostaria de agradecer a todos.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletrônico e de Computação.

DESENVOLVIMENTO DE UM EQUIPAMENTO DE BAIXO CUSTO PARA MEDIÇÃO DE PARÂMETROS DA FREQUÊNCIA RESPIRATÓRIA COM APLICAÇÃO EM NEUROCIÊNCIA COMPORTAMENTAL HUMANA.

Bruno Gomes Reis

Agosto/2019

Orientadores: Carlos José Ribas D'Avila
Tiago Arruda Sanchez

Curso: Engenharia Eletrônica e de Computação

O RESUMO SERÁ ESCRITO AQUI

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

LOW COST EQUIPMENT DESIGNED TO MEASURE RESPIRATORY FREQUENCY PARAMETERS APPLIED TO HUMAN BEHAVIORAL NEUROSCIENCE

Bruno Gomes Reis

August/2019

Advisors: Carlos José Ribas D'Avila
Tiago Arruda Sanchez

Course: Electronic Engineering

In this work, we present ...

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\emptyset$	Conjunto vazio, p. 1
$\mathbb{R}$	Conjunto dos números reais, p. 1

Lista de Abreviaturas

COPPE	Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, p. 1
POLI-UFRJ	Escola Politécnica da Universidade Federal do Rio de Janeiro, p. 1

Capítulo 1

Introdução

1.1 Tema

O tema do trabalho é o desenvolvimento de um dispositivo que realize a medição da frequência respiratória dentro de uma faixa sensível para respostas psicofisiológicas. Este dispositivo deve possibilitar o monitoramento e a extração de dados que auxiliarão em estudos os quais correlacionam alguns parâmetros da frequência respiratória com funções psicofisiológicas, como aquelas observadas sob alterações cognitivas, emocionais, sob estresse doenças mentais.

1.2 Delimitação

O objeto de estudo é o desenvolvimento de um protótipo capaz de mensurar comportamentos respiratórios tais quais a frequência de inspiração e expiração, para, a partir desse, analisar a viabilidade de desenvolvimento de um equipamento com baixo custo voltado ao uso científico. As medições terão como finalidade a obtenção de dados, referentes à frequência respiratória do paciente, a serem utilizados em experimentos que vinculem comportamentos saudáveis e suas variantes clínicas à respiração humana. O estudo, limita-se à obtenção desses dados, não abrangendo, a priori, interpretações acerca das correlações obtidas em eventuais medições realizadas, além de possuir, em princípio, finalidade meramente científica, sem conter qualquer estudo sobre viabilidade comercial de produtos que venham a ser desenvolvidos.

1.3 Justificativa

Segundo Sebastião Gusmão [1], a medicina como ciência, baseada na interpretação natural da doença e não em magia e empirismo, como ocorria na medicina arcaica,

tem sua origem no século V a.C com Hipócrates (c. 460-375 a.C). Desde então, análises e estudos sobre o funcionamento do corpo humano, bem como as interações deste com o meio ambiente, vêm sendo realizados em constante evolução. Hoje, sabe-se que o organismo humano é composto de diversas partes que, em conjunto, garantem o seu funcionamento adequado. O corpo é, portanto, um sistema complexo no qual atuam diversas variáveis. Sendo assim, grande parte dos estudos científicos atuais voltam seus métodos e análises ao estudo dos parâmetros que possuem influências para o bom ou mal funcionamento do organismo. Atualmente, sabe-se que diversas doenças psicofisiológicas produzem variações no funcionamento normal do corpo, como alterações na produção de determinados hormônios, no batimento cardíaco, na pressão arterial ou na concentração de CO_2 no sangue.

Estudos recentes demonstram que é possível por exemplo, induzir pânico em ratos apenas alterando a concentração de O_2 ou de CO_2 do ar por eles respirado [2] [3]. A respiração humana é, assim como nos ratos, o processo natural responsável pela troca do CO_2 com o O_2.

Ao realizar uma análise de correspondência (correlação, regressão, inferência etc.) entre determinado comportamento biológico e algum quadro clínico específico, o pesquisador necessita realizar, de alguma maneira, a mensuração das variáveis que compõem o comportamento com acurácia e significado preditivo para a sensibilidade e a especificidade dos resultados. Nesse sentido, o desenvolvimento de um equipamento de baixo custo capaz de coletar dados referentes ao comportamento respiratório torna-se parte essencial à evolução do estudo científico.

Munido dessa motivação, neste trabalho, apresentam-se estudos para viabilidade do desenvolvimento de uma tecnologia com baixo custo capaz de monitorar o comportamento do fluxo respiratório de pacientes, servindo então como base para estudos científicos nas áreas de psicofisiologia e neurociência comportamental.

1.4 Objetivos

O objetivo geral deste estudo é, então, desenvolver um protótipo de equipamento capaz de realizar a aquisição de dados referentes à frequência respiratória em humanos, incluindo parâmetros que possam ser utilizados em pesquisas psicofisiológicas. Desta forma, tem-se como objetivos específicos: (1) Realizar a mensuração da frequência respiratória (2) desenvolver os métodos de medida da frequência respiratória para extrair suas variantes no domínio do tempo e da frequência; e (3) Possibilitar a exportação dos dados para análises futuras.

1.5 Metodologia

A priori, a medição da frequência respiratória seria obtida indiretamente pela variação da temperatura do ar próximo à narina do paciente, uma vez que, em um ambiente controlado, o ar inalado possui uma temperatura inferior à do ar exalado. Contudo, considerando a baixa variação de temperatura para esse tipo de medição, seria inviável a realização da medida diretamente, dada a velocidade de resposta necessária para a obtenção dos dados de forma confiável e a utilização de equipamentos de baixo custo como requisito. Para contornar esse problema, foi utilizada a propriedade de autoaquecimento do termistor, que passou a ser utilizado como um sensor de fluxo. Ou seja, o sistema não mais tenta inferir a frequência respiratória com base no aumento da temperatura ambiente ocasionada pela saída de ar quente do corpo humano, ao contrário, ele aplica uma corrente muito alta no sensor forçando-o a atingir por conta própria uma temperatura ainda maior em seu estado permanente e, no momento de seu encontro com qualquer fluxo de ar resultante da respiração humana, o sensor registra uma alta queda de temperatura, registrando, de forma mais nítida o momento da ação respiratória.

O termistor é um resistor variável à temperatura, escolhido principalmente por suas propriedades de autoaquecimento e pela variação exponencial de sua resistência em relação à mudança na temperatura, contrário a grande parte dos demais sensores que possuem uma relação linear entre essas variáveis. Trabalhar com um sensor de cuja curva característica é exponencial facilita detecções de pequenas variações na temperatura, gerando grandes mudanças na resistência, em contrapartida, adiciona complexidade ao sistema dado que trabalhar com relações lineares é, em geral, mais simples. A escolha do termistor é portanto perfeita porque sua contrapartida sequer implica em complicações para o sistema deste trabalho, dado que, em princípio, não interessa uma medição precisa da temperatura, mas sim o registro dos eventos de inspiração e expiração.

O sistema é composto de uma fonte capaz de entregar ao sensor corrente suficiente para que este entre em estado de autoaquecimento e atinja uma temperatura alta o suficiente para se tornar sensível à variação provocada pelos fluxos de ar, um circuito regulador responsável por filtrar sinais indesejados e controlar a corrente de entrada no termistor, um microcontrolador que irá realizar as medições e um software capaz de exportar os dados para que esses possam ser tratados e estudados em suas aplicações.

1.6 Organização do Trabalho

Nos próximos capítulos serão apresentados em mais detalhes as aplicações, o desenvolvimento e os resultados obtidos com esse trabalho, organizados da seguinte forma:

O capítulo 2 apresentará a motivação para o desenvolvimento do equipamento, as aplicações no campo da medicina e um resumo do funcionamento fisiológico da respiração humana.

O capítulo 3 tratará sobre desenvolvimento do sistema, hardware e software bem como o raciocínio por trás da arquitetura.

O capítulo 4 será destinado à conclusão desse projeto, com o protótipo construído no último ciclo de desenvolvimento, limitações encontradas ao longo das etapas e também possíveis melhorias para trabalhos futuros.

Capítulo 2

Motivação

2.1 Fisiologia da Respiração

A respiração é, em resumo, um ato semiautomático no qual um controle involuntário é exercido por centros respiratórios no tronco encefálico e um controle voluntário é exercido por centros motores de músculos acessórios da respiração no córtex motor.

A inspiração possui como músculo principal o diafragma, que se contrai, desce e expande a cavidade torácica, comprimindo então o conteúdo abdominal e empurrando para fora a parede abdominal. Concorrentemente, os músculos da caixa torácica também expandem o tórax, em especial, os músculos escalenos que percorrem das vértebras cervicais às duas primeiras costelas e os músculos intercostais paraesternais, que possuem um trajeto desde o externo até as costelas. Na medida em que o tórax se expande, a pressão intratorácica diminui, deslocando o ar da árvore traqueobrônquica para os alvéolos, preenchendo os pulmões em expansão. O oxigênio é então difundido para os capilares pulmonares adjacentes, enquanto o dióxido de carbono sai do sangue para os alvéolos.

Quando ocorre a expiração, a parede torácica bem como os pulmões retraem-se, o diafragma relaxa, elevando-se passivamente, enquanto o ar flui para fora do corpo.

Em estado normal, a respiração é tranquila, audível próximo à boca, sendo possível observar apenas os movimentos abdominais com facilidade. Contudo, durante a prática de atividade física, ou em decorrência de determinadas doenças, faz-se necessário um esforço respiratório adicional, recrutando músculos acessórios e demandando ainda mais esforço dos músculos abdominais, que passa a auxiliar também na expiração.

2.2 Padrões Respiratórios

A faixa de frequência padrão para adultos normais é de, aproximadamente, 14 à 20 incursões por minuto e até 44 em lactantes. Em sua normalidade, a inspiração e expiração possuem o mesmo tempo e amplitude, sendo intercalados por uma leve pausa. No momento em que uma dessas características é modificada, surgem os ritmos respiratórios anormais, tais quais a respiração de Cheyne-Stokes, respiração de Biot, respiração de Kussmaul e respiração suspirosa.

Respiração de Cheyne-Stokes: Frequentemente causada por insuficiência cardíaca, hipertensão intracraniana, acidentes vasculares cerebrais e traumatismos cranioencefálicos. Caracteriza-se por uma fase de apneia seguida de incursões inspiratórias cada vez mais profundas até atingir um máximo e, em seguida, decrescer até uma nova pausa. Esse comportamento ocorre devido a variações da tensão de O_2 e CO_2 no sangue. O excesso de CO_2 durante o período de apneia obriga os centros respiratórios a enviar estímulos mais intensos, resultando em um aumento da amplitude dos movimentos respiratórios, em consequência, haverá uma maior eliminação de CO_2 fazendo com que a concentração deste no sangue decaia. Após essa queda, os centros respiratórios recebem o comando inverso, enviando estímulos menores e diminuindo a amplitude dos movimentos respiratórios até que aumente novamente a concentração de CO_2 no sangue e o ciclo se repita.

Respiração de Biot (ou atáxica): As causas dessa respiração são similares às da respiração de Cheyne-Stokes, porém, no ritmo de Biot, a respiração possui duas etapas, uma de apneia seguida de outra com movimentos respiratórios anárquicos quanto ao ritmo e à amplitude. Esse padrão respiratório quase sempre indica um grave comprometimento cerebral.

Respiração de Kussmaul: Costuma ser causada pela acidose (diminuição do PH sanguíneo para menos de 7,35[aumento de H+]), principalmente a diabética. O padrão respiratório é composto de quatro fases: Inspirações ruidosas, gradativamente mais amplas, alternada com inspirações rápidas e de pequena amplitude; Apneia em inspiração; Expirações ruidosas, gradativamente mais profundas, alternadas com inspirações rápidas e de pequena amplitude; E apneia em expiração.

Respiração suspirosa: Normalmente, pode ser traduzida em tensão emocional e ansiedade. Seu padrão ocorre quando o paciente executa uma série de movimentos inspiratórios de amplitude crescente, seguidos de uma expiração breve e rápida.

Além desses padrões respiratórios, existem diversos outros padrões observados (como a Taqpineia, Bradpneia, Respiração obstrutiva e Hiperpneia) que possuem correlação com patologias ou alterações emocionais e, portanto, existe um vasto campo de estudo acerca desse tema.

2.3 Aplicação do equipamento

Motivado pela parceria com o "Laboratório de Neuroimagem e Psicofisiologia", situado no Hospital Universitário Clementino Fraga Filho, ocorre o desenvolvimento deste projeto. Orientado principalmente a estudos nas áreas de psicofisiologia e neurociência comportamental, o desenvolvimento do equipamento visa trazer mais insumos às análises realizadas nos diversos projetos existentes no laboratório. Conforme mencionado na sessão 2.2, existem diferentes padrões respiratórios já observados entre os quais se é possível obter uma relação de correlação com patologias e alterações emocionais e comportamentais. Por derradeiro, a existência de um equipamento capaz de obter dados referentes ao comportamento respiratório transcreve-se em um ganho à neurociência comportamental pelo qual este projeto se justifica.

Capítulo 3

Desenvolvimento

3.1 O funcionamento do termistor

O termistor é um resistor variável à temperatura, característica pela qual seu nome é inspirado (temperatura + resistor), geralmente compostos por uma liga que contém cerâmica e outros polímeros. Suas aplicações mais habituais costumam acontecer em circuitos de monitoramento e controle de temperatura ou como limitador de corrente de partida. São separados, basicamente, em duas categorias:

- NTC (Negative Temperature Coefficient): São termistores que diminuem a sua resistência à medida em que a temperatura aumenta; E
- PTC (Positive Temperature Coefficient): Termistores que aumentam a sua resistência diretamente com o aumento da temperatura.

3.1.1 A equação de Steinhart–Hart

A curva de variação da resistência de um termistor em função de sua temperatura pode ser descrita pela equação de Steinhart–Hart3.1 em sua forma inversa 3.2. A característica logarítmica da resposta com que a resistência varia em função da variação de temperatura (figura 3.1) é, em grande parte das vezes, uma das principais vantagens do seu uso, visto que pequenas variações na temperatura provocam grandes alterações na resistência, fazendo com que o sensor seja mais suscetível a detectar pequenas mudanças na temperatura.

$$\frac{1}{T} = a + b\ln(R) + c(\ln(R))^3 \tag{3.1}$$

Sendo:

- T: Temperatura em Kelvins;
- R: Resistência em ohms (Ω);E

- a,b e c: Coeficientes de Steinhart–Hart, que são variáveis conforme o tipo de material e a construção do termistor.

$$
\begin{aligned}
R &= \exp\left(\sqrt[3]{\beta - \frac{\alpha}{2}} - \sqrt[3]{\beta + \frac{\alpha}{2}}\right), \\
\alpha &= \frac{1}{c}\left(a - \frac{1}{T}\right), \\
\beta &= \sqrt{\left(\frac{b}{3c}\right)^3 + \left(\frac{\alpha}{2}\right)^2}
\end{aligned}
\tag{3.2}
$$

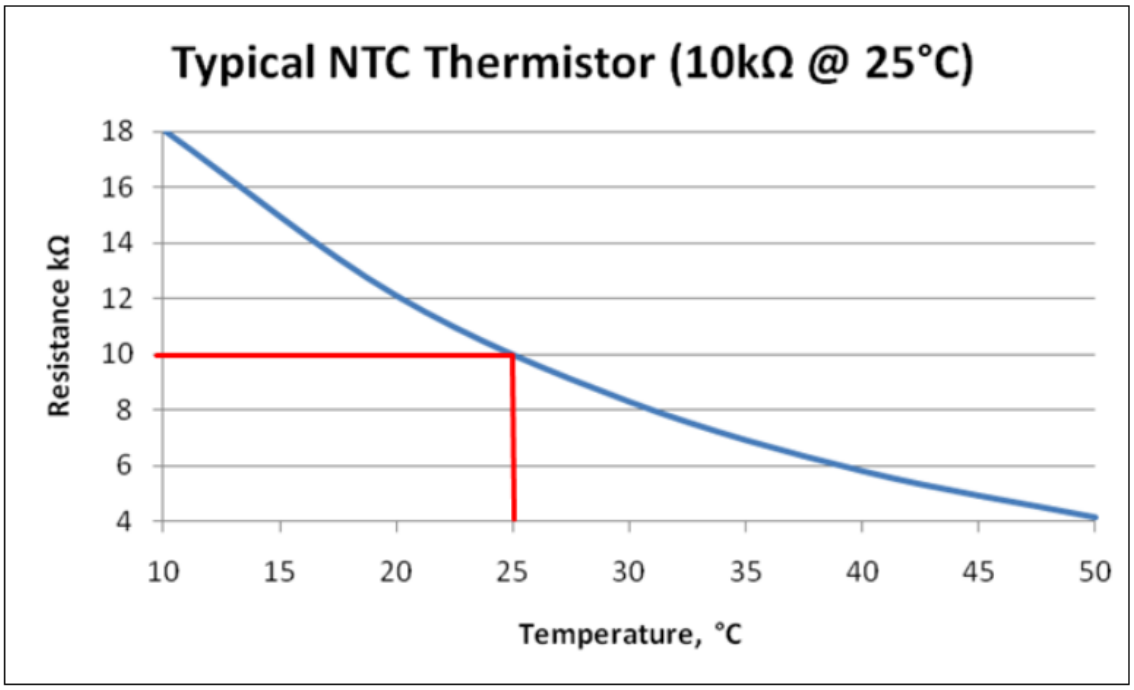

Figura 3.1: Curva de resposta típica em termistores NTC de 10kΩ[1]

3.1.2 O efeito de Autoaquecimento

Como toda resistência, o termistor dissipa energia elétrica na forma de calor. Portanto, ao aplicar uma corrente no sensor, é induzido um efeito de autoaquecimento. A relação da potência elétrica dissipada é dada pela equação 3.3 e a relação entre potência e temperatura pode ser obtida através da equação 3.4. Fazendo $P_E = P_T$, é possível chegar à equação 3.5.

$$P_E = I.V \tag{3.3}$$

sendo:

[1]Fonte: http://www.squids.com.br/arduino/index.php/projetos-arduino/projetos-squids/basico/159-projeto-42-comparando-sensores-de-temperatura-ntc-10k-dht11-e-lm35

- P_E: Potência elétrica dissipada;
- I: Corrente elétrica;E
- V: Tensão entre os terminais.

$$P_T = K(T_{(R)} - T_0) \tag{3.4}$$

sendo:

- P_T: Potência;
- K: Fator de dissipação do termistor;
- $T_{(R)}$: Temperatura em função da resistência;E
- T_0: Temperatura ao redor do termistor.

$$T_0 = T_{(R)} - \frac{V^2}{K.R} \tag{3.5}$$

3.2 A evolução do circuito

3.2.1 O divisor resistivo

Valendo-se da propriedade logarítmica do termistor, uma pequena variação de temperatura ambiente ocasionada pelo processo de expiração ocasionaria uma mudança exponencial no valor da resistência. Por esse motivo, na origem do projeto, era esperada uma medição simples, obtida através da variação de tensão em um divisor resistivo composto por um termistor e uma resistência padrão(Figura: 3.2). Contudo, devido alguns contratempos práticos, diversas mudanças fizeram-se necessárias.

Uma das principais limitações para o desenvolvimento do projeto deve-se à dificuldade na aquisição dos componentes eletrônicos que, quando comprados via internet, necessitavam de um tempo para entrega relativamente alto e, para a compra realizada diretamente na loja física, existe uma limitação na variedade de lojas especializadas na cidade do Rio de Janeiro. Outra complicação relevante é referente à ausência de datasheet, que não é informado no momento da compra e torna-se impossível inferir qual é o datasheet correto apenas observando o componente, que é muito pequeno, sem qualquer informação sobre o fabricante ou o modelo. Dadas essas condições iniciais, foi construído o primeiro divisor resistivo apenas com base na informação de que o sensor adquirido tratava-se de um termistor NTC (do inglês Negative Temperature Coefficient) com resistência em temperatura ambiente de

$10K\Omega$. Foi utilizada uma fonte de alimentação comercial de $12V$ e utilizado diversos valores entre $10K\Omega$ e 330Ω para a resistência $R2$, contudo para nenhum valor de $R2$ era observada qualquer alteração de tensão na medida em que o ar era exalado próximo ao sensor, contrariando ao que era esperado, dado que a queda de tensão em cima do termistor deve ser variável junto à alteração na resistência.

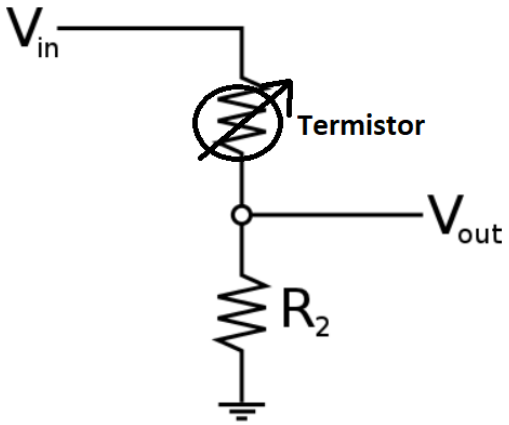

Figura 3.2: Divisor Resistivo

3.2.2 Lei de resfriamento de Newton

A lei de resfriamento de Newton (3.6) indica que a taxa com que um corpo perde calor é proporcional à diferença de temperatura entre o corpo e o meio no qual ele se encontra. Valendo-se desse princípio, nota-se que uma baixa diferença de temperatura resultaria em um maior tempo de resposta do sensor.

$$\frac{dQ}{dt} = h.A.(T(t) - T_{env}) = h.A\Delta T(t) \tag{3.6}$$

Onde:

Q: Energia térmica

t: Tempo

h: Coeficiente de transferência de calor

A: Área de transferência de calor

T: Temperatura do objeto

T_{env}: Temperatura do ambiente

$$T_0 = T(R) - \frac{V^2}{KR} \tag{3.7}$$

Onde:

T_0: Temperatura do meio

$T(R)$: Temperatura do termistor em função de sua resistência

V: Diferença de potencial entre os terminais do termistor

K: Fator de dissipação do termistor

R: Resistência

3.2.3 Medição da variação do Termistor

Para entender melhor o comportamento do termistor e a forma com que ele variava sua resistência devido o efeito do autoaquecimento, sem acesso ao datasheet do mesmo, foram realizadas medições utilizando o divisor resistivo, colocando o termistor em série com um resistor fixo de 330Ω e uma fonte de bancada com fornecimento de tensão variável entre $0V$ e $30V$. A curva de variação pode ser observada no gráfico da figura 3.3.

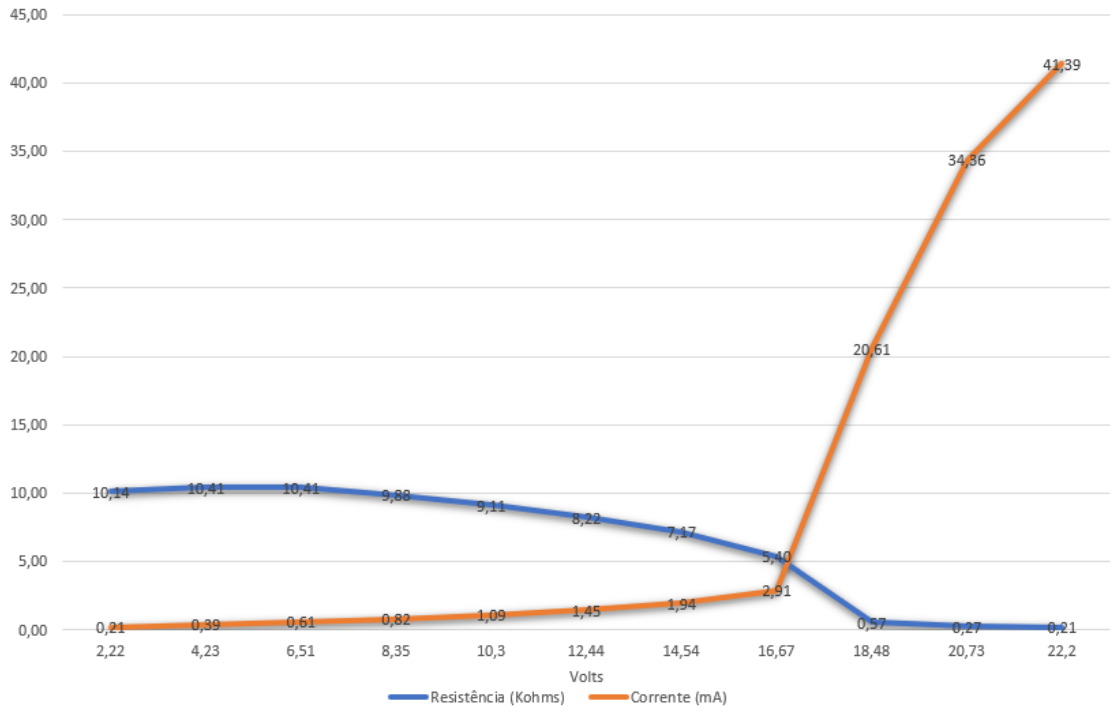

Figura 3.3: Variação do termistor em função da tensão

3.2.4 Circuito utilizando o efeito do autoaquecimetno

Portanto, para contornar o problema obtido devida a lei de resfriamento de Newton, o circuito foi ajustado para utilizar a propriedade de autoaquecimento do termistor (3.7).

Ao aplicar uma alta corrente no sensor, é induzido então um aumento na temperatura deste. Em decorrência desse aumento, é possível observar uma maior diferença de temperatura entre o sensor e o fluxo de ar e, por conseguinte, uma maior taxa para transferência de calor e um menor tempo de resposta por parte do termistor. Contudo, para atingir uma faixa de temperatura sensível, capaz de gerar uma resposta visível ao expor o sensor à respiração, o circuito necessita de uma tensão elevada, acima das entregues por fontes comerciais padrão, que costumam variar entre $5V$ e $12V$, gerando a necessidade de projetar um retificador de tensão capaz de converter a tensão de corrente alternada entregue pela rede elétrica residencial em uma tensão contínua alta o suficiente para fornecer ao termistor a corrente demandada.

Realizando testes de bancada, com um gerador de tensão variando de $0V$ à $30V$ e uma resistência de 330Ω em série com o termistor, foi possível atingir uma temperatura sensível ao sopro, entretanto, ao aplicar uma tensão em torno de $22V$, o sensor demorava um tempo considerável para aquecer novamente, tornando-o inviável para medir o comportamento respiratório dada a frequência do sopro em uma respiração normal. Aumentando a tensão para $28V$, já era possível observar uma atenuação considerável na queda constante da temperatura dado que o termistor era capaz de se aquecer mais rápido. O problema gerado por esse aumento de tensão deve-se ao fato de que, quanto maior é a corrente, maior é o autoaquecimento e, quanto maior a temperatura, menor é a resistência, gerando um aumento ainda maior na corrente passante que, por sua vez, aumenta ainda mais a temperatura até que o sensor atingia um patamar no qual queimava, caso não houvesse nenhum sopro forçando a temperatura a diminuir.

Para suprir o fornecimento de tensão em corrente contínua em torno dos $28V$ foi necessário construir um retificador de tensão, uma vez que as fontes padrão, encontradas no mercado, costumam fornecer tensão menor que a desejada.

– Colocar foto do circuito e do projeto da fonte

3.2.5 Circuito com controle realimentado

Se por um lado, o aumento da tensão era importante para que a temperatura não decaísse constantemente ao iniciar a medição respiratória, por outro, o sistema precisava ser protegido para que a corrente não atingisse um determinado patamar que danificasse o componente. O ajuste mais simples para regular a corrente em um componente costuma ser adicionar uma resistência em série, diminuindo a corrente

por uma mera consequência da lei de ohm (3.8). Contudo, por se tratar de uma resistência variável, a tensão necessária para o aquecimento do termistor em seu estado inicial era maior que a tensão necessária para mantê-lo abaixo de um patamar seguro após a diminuição de sua resistência pelo efeito do autoaquecimento. Foi pensado então em um circuito de controle não linear (Figura: 3.4), que, em teoria, forneceria uma alta tensão para o termistor até que sua resistência variasse e a corrente passante atingisse um patamar determinado.

$$V = R.I \tag{3.8}$$

Onde:

V: Tensão

R: Resistência

I: Corrente

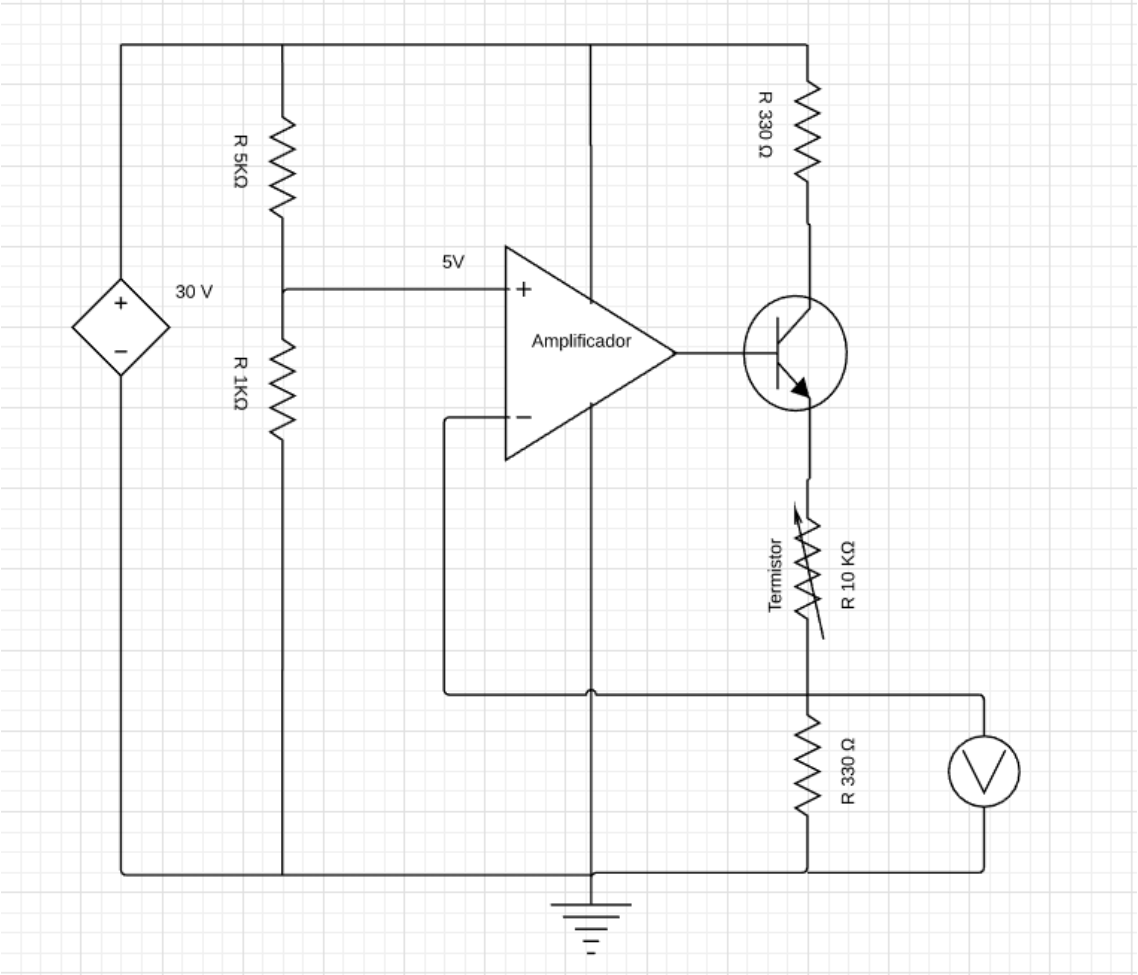

Figura 3.4: Circuito de Controle

O comportamento esperado para o circuito de controle seria o seguinte: O amplificador operacional iria ajustar a tensão de saída na tentativa de igualar as tensões nos dois terminais de entrada, sendo assim, no momento inicial, quando a tensão no terminal de entrada negativo é igual a zero e a do terminal positivo é igual a $5V$, o amplificador aumenta sua saída fazendo com que o transistor entre em saturação e conduza uma corrente no emissor praticamente igual à de referência no coletor, na medida em que a resistência do termistor diminui pelo efeito de autoaquecimento, a tensão de saída aumenta até que esta se iguale à do terminal positivo, ao atingir esse patamar, o transistor sai da zona de saturação e o sistema trabalha pra manter a tensão de saída constante. No momento em que o sensor entra em contato com o fluxo de ar, a resistência do termistor aumenta, diminuindo a tensão na saída e estimulando o sistema de controle a saturar o transistor e induzir o efeito do autoaquecimento no sensor. Após quantizar a tensão da saída através de um microcontrolador, foi possível traçar o gráfico da figura 3.5, no qual aparece nítida a resposta do sistema ao submeter o sensor a um fluxo de ar. Entretanto, ao expor o sensor a um período prolongado de exposição à respiração humana, foi possível observar que o efeito do resfriamento contínuo continuava a ser reproduzido (gráfico da figura 3.6).

Figura 3.5: Resposta em curto prazo circuito realimentado (Tempo no eixo horizontal e valor de tensão quantizado no eixo vertical)

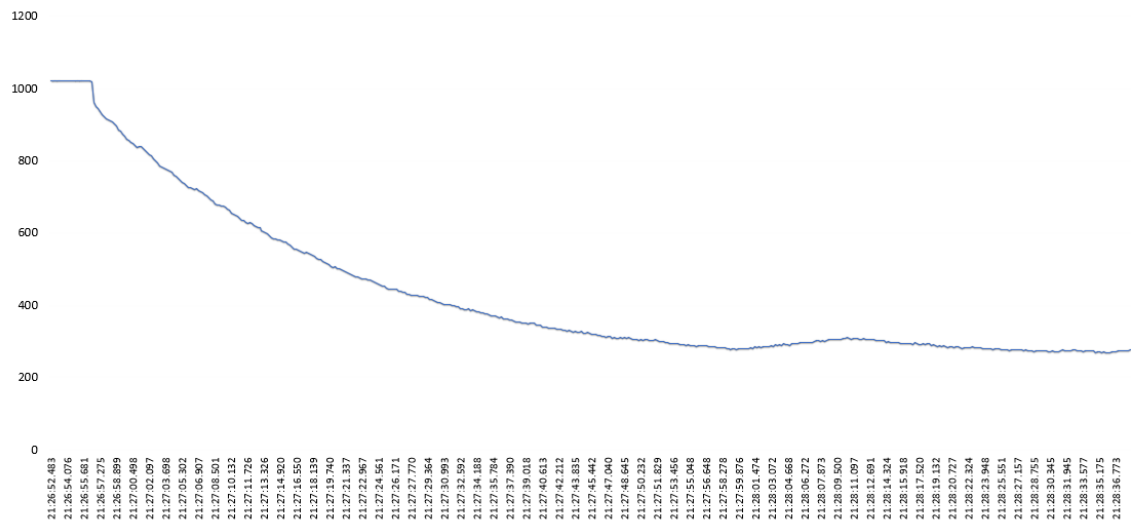

Figura 3.6: Resposta de longo prazo circuito realimentado (Tempo no eixo horizontal e valor de tensão quantizado no eixo vertical)

3.3 O Software

Em paralelo ao desenvolvimento do hardware, foi desenvolvido um software capaz de obter os dados de um microcontrolador através de uma porta serial no computador. Dada a escolha do Arduíno para o desenvolvimento do protótipo de baixo custo, era possível abstrair o desenvolvimento do hardware pressupondo, de antemão, que o circuito deveria ser projetado para entregar à entrada analógica do Arduíno um sinal de tensão variante entre $0V$ e $5V$. Para interpretar esse sinal, o microcontrolador existente no Arduíno (ATmega328) converte o sinal analógico em digital. A resolução do sinal convertido é dada pela equação 3.9, sendo o valor da tensão de referência igual a $5V$ e a quantidade de bits disponível para conversão igual a $10bits$, chegamos a uma resolução de $4,48mV$. Em outras palavras, abstraindo do hardware, o software deveria ser programado para ser capaz de receber uma entrada pela porta serial e ler sinais inteiros variando entre 0 e 1024 (Equivalente a $10bits$) onde cada variação inteira representa $4,48mV$ do sinal de entrada do Arduíno.

$$Resolucao = \frac{V_{ref}}{2^n} \tag{3.9}$$

Onde:

V_{ref}: Tensão de referência

n: Número de bits do conversor

A linguagem de programação escolhida para o desenvolvimento do software foi o Python dada a sua notória ascensão e eficiência para a realização de operações

matemáticas e tratamento de dados científicos, além de ser referência em áreas como inteligência artificial (o que viabilizaria estudos futuros nessa área).

3.3.1 Software de teste de conceitos

Leitura do sinal de entrada

Antes de gerar um sinal simulado para iniciar o efetivamente o desenvolvimento do software, foi desenvolvido um pequeno protótipo para garantir que seria possível a comunicação entre um sinal gerado pelo Arduíno e o software desenvolvido em Python. Basicamente, existem dois softwares atuando simultaneamente para realizar essa leitura. Um, desenvolvido na linguagem de programação C, que fica instalado diretamente no Arduíno e é responsável por realizar a medição periódica e crua do sinal digital, logo após a sua quantização, e escrever um par valor x tempo na porta USB. O segundo é, de fato, o programa escrito em Python que, dentre outras funções realiza a leitura do sinal escrito pelo programa anterior na porta USB e salva os dados em um arquivo .csv, que poderá ser lido futuramente para a realização das operações matemáticas.

Simulação do sinal de entrada

Durante os experimentos com o hardware, foi possível observar um padrão na resposta do sinal analógico quando medido pelo osciloscópio que, em todas as versões do circuito, possuía um comportamento muito similar (imagem 3.8). A forma desse sinal pode ser explicada observando o princípio de funcionamento escolhido para que o sensor de temperatura seja capaz de medir a respiração. Em um cenário ideal, espera-se que o sensor esteja aquecido até que encontre uma temperatura de equilíbrio de operação e, quando em contato com um fluxo de ar da inspiração ou expiração, essa temperatura irá diminuir, aumentando a resistência do sensor e, em consequência, exibindo, no osciloscópio, uma diminuição da tensão. Quando acaba o fluxo de ar (tempo entre a expiração e inspiração), o circuito tentará esquentar novamente o sensor até que ele retorne à temperatura de equilíbrio. É possível também, observar que possuímos duas faixas distintas de amplitude na queda de tensão, o que nos faz concluir que o sensor tenderá a não obter uma resposta indiferente ao tipo de fluxo de ar (inspiração ou expiração) seja pelo fato de que a temperatura do ar na expiração é levemente maior, ou seja devido à construção que pode privilegiar o contato do fluxo do ar com o sensor em um sentido mais que no outro. Nos casos experimentados até então, por exemplo, a queda de tensão decorrente da inspiração era consideravelmente maior que a da expiração.

Considerando o comportamento dos sinais analógicos obtidos nas versões de circuito anteriores e o caráter cíclico natural do comportamento respiratório, o sinal

```python
import serial
import time
import csv
import matplotlib
matplotlib.use("tkAgg")
import matplotlib.pyplot as plt
import numpy as np

ser = serial.Serial('COM6', 9600)
ser.flushInput()

plot_window = 20
y_var = np.array(np.zeros([plot_window]))

plt.ion()
fig, ax = plt.subplots()
line, = ax.plot(y_var)

while True:
    try:
        ser_bytes = ser.readline()
        try:
            decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
            print(decoded_bytes)
            tempo = time.asctime( time.localtime(time.time()) )
        except:
            continue
        with open("test_data.csv","a") as f:
            writer = csv.writer(f,delimiter=",")
            writer.writerow([time.time_ns(),decoded_bytes])
        y_var = np.append(y_var,decoded_bytes)
        y_var = y_var[1:plot_window+1]
        line.set_ydata(y_var)
        ax.relim()
        ax.autoscale_view()
        fig.canvas.draw()
        fig.canvas.flush_events()
    except:
        print("Keyboard Interrupt")
        break
```

Figura 3.7: Leitor de porta USB

de entrada foi simulado como um somatório de senoides com diferentes amplitudes e frequências. A frequência escolhida para o sinal principal foi a de 18 ciclos por minuto, que é a faixa superior da frequência considerada confortável para um adulto normal em repouso. Uma nova senoide com a metade da frequência foi gerada e posta defasada de modo a atenuar os picos referentes à expiração e agravar os relacionados à inspiração e, para a simulação do ruído, foi adicionada mais uma senoide com frequência mais elevada. O sinal simulado pode ser descrita conforme a equação 3.10 e produz o resultado apresentado no gráfico da figura 3.9.

$$F(t) = \sin(0.15 * 2\pi t) + 0.8\sin(\frac{\pi}{2} + 0.3 * 2\pi t) + 0.1\sin(\frac{\pi}{2} + 30 * 2\pi t) \qquad (3.10)$$

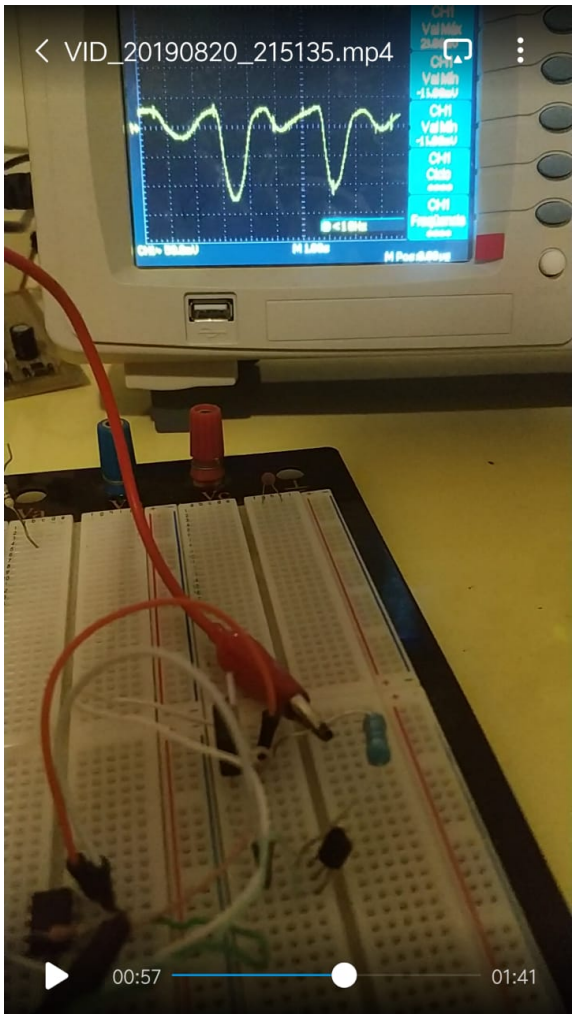

Figura 3.8: Leitura do sinal analógico no osciloscópio

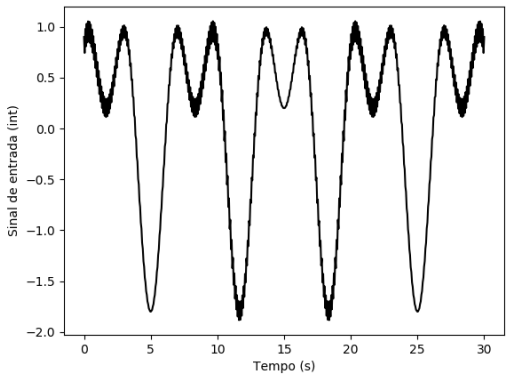

Figura 3.9: Função simulada interpolada

Operações Realizadas

A primeira versão do software foi projetada para ser capaz de realizar as seguintes operações:

1- Ler os dados de uma porta serial via usb que serão enviados por um arduíno (trecho do código descrito na subseção 3.3.1)

2- Criar pastas com o nome e horário das medições para o armazenamento dos arquivos

3- Salvar os dados em um arquivo .csv

4- Ler os dados do arquivo csv e realizar as seguintes operações:

4.1- Plotar diretamente

4.2- Plotar uma interpolação

4.3- Calcular a derivada primeira e segunda e plotar

4.4- Calcular a FFT e plotar

5- Ler o arquivo csv, passar por um filtro IIR passa baixas, salvar um novo csv e realizar as seguintes operações:

5.1- Plotar diretamente

5.2- Plotar uma interpolação

5.3- Calcular a derivada primeira e segunda e plotar

5.4- Calcular a FFT e plotar

6- Ler o arquivo csv, passar por um filtro IIR passa baixas, salvar um novo csv e realizar as seguintes operações:

6.1- Plotar diretamente

6.2- Plotar uma interpolação

6.3- Calcular a derivada primeira e segunda e plotar

6.4- Calcular a FFT e plotar

Calculo das derivadas

Para obter a derivada do sinal de entrada utilizamos o método das diferenças finitas 3.11, uma vez que, ao ser processado pelo programa, o sinal encontra-se discretizado e com uma boa quantidade de pontos devido a alta taxa de amostragem em relação à frequência de variação da tensão provocada pela respiração. O sinal discretizado é armazenado em dois vetores de mesmo tamanho, onde um armazena o valor da tensão e outro o valor do tempo de medição. Ao realizar o somatório da equação 3.11 em todos os pontos dos vetores, conseguimos plotar o gráfico da derivada. Foi criada uma função no programa em Python que recebia como parâmetro os vetores

x e y e retornava o resultado operação supracitada, permitindo assim o reuso do método para o cálculo da derivada segunda.

$$f'(x) = \sum_{X0}^{Xn} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3.11}$$

Onde:

$f'(x)$: Derivada da tensão no ponto x

$X0$: Primeiro valor do vetor de tempo

Xn: Último valor do vetor de tempo

$f(x)$: Valor da tensão no ponto x

Fast Fourier Transform (FFT)

Para obter informações referentes à frequência do sinal respiratório, o sistema calcula e plota um gráfico realizando uma análise de Fourier, convertendo o sinal original, no domínio do tempo, em uma representação no domínio da frequência. Contudo, realizar o cálculo da transformada de Fourier diretamente de sua definição 3.12 implica em operações aritméticas de complexidade $\boldsymbol{O(n^2)}$, entretanto é possível atingir o mesmo resultado utilizando a FFT com uma complexidade de apenas $\boldsymbol{O(n \log(n))}$. Aumentando a performance do software.

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt \tag{3.12}$$

Complexidade de algorítimo

A complexidade de um algorítimo é determinada através da relação entre o tempo e memória gastos para realizar a operação de acordo com o tamanho da entrada. Ou seja, em termos práticos, é a complexidade do algorítimo, aliada à capacidade de processamento do computador, quem determinará a velocidade de funcionamento do sistema na medida em que se aumenta a quantidade de dados na entrada. A forma mais popular de representar a complexidade de tempo na análise de algorítimos é a ***complexidade assintótica***, que ignora fatores constantes e os termos que crescem mais lentamente. Por exemplo, em uma relação onde o tempo de execução $T(N)$ em função da entrada de dados N é de $T(N) = 27N^2 + 45N + 12$, na medida em que N aumenta, a função pode ser simplificada por $T(N) = 24N^2$. Simplificando ainda mais a análise, para prever o quão rápido será o crescimento do tempo de execução em relação ao aumento da entrada, podemos desconsiderar a constante que multiplica o termo de maior grau, uma vez que, tanto para $T(N) = 100000N^2$

quanto para $T(N) = 10N^2$, dobrar a entrada N implicará em quadruplicar o tempo $T(N)$. Representamos $O(g)$ como sendo o limite superior para o tempo gasto por um algorítimo, em outras palavras, $O(g)$ representa a classe de funções que crescem, no máximo, tão rápidas quanto a função g. Ou seja, reduzir o cálculo da FFT de complexidade $\boldsymbol{O(n^2)}$ para $\boldsymbol{O(n\log(n))}$. representa diminuir o tempo de execução, antes limitado pela função em vermelho, para a função em azul no gráfico da figura 3.10.

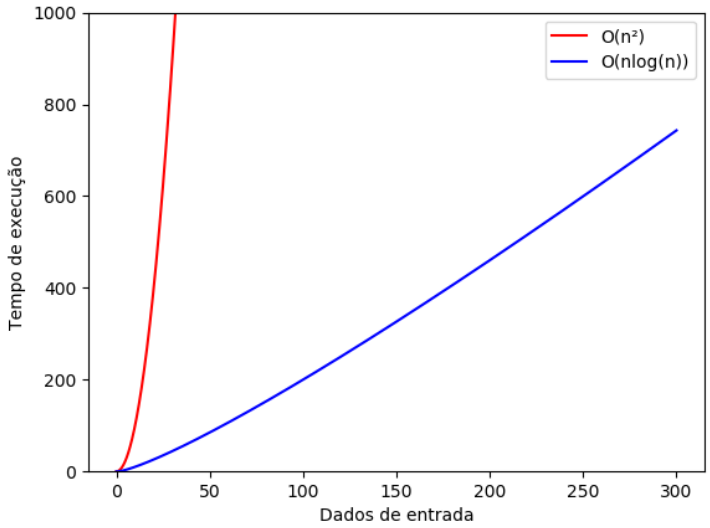

Figura 3.10: Relação entre $\boldsymbol{O(n^2)}$ e $\boldsymbol{O(n\log(n))}$.

Filtro passa-baixas

Com a finalidade de eliminar eventuais ruídos, o sistema é capaz de aplicar filtros passa-baixas, eliminando altas frequências ruidosas e preservando as variações que ocorrem em uma faixa de frequência compatível com a respiração humana (abaixo de 70 ciclos por minuto em atletas). Os filtros digitais são divididos em duas classes FIR e IIR e, para maior liberdade do pesquisador, o sistema oferece as duas opções de filtragem.

IIR (Infinite Impulse Response)

Os filtros IIR são mais rápidos que os FIR. Para esse programa, foi escolhido o filtro de Butterworth, que possui resposta na frequência conforme a figura 3.11. Na medida em que aumentamos a ordem do filtro, conseguimos obter uma atenuação mais ingrime na filtragem 3.12, contudo, adicionamos mais complexidade ao cálculo. Para o experimento, foi escolhida, empiricamente, o filtro de ordem 3. Chamamos

de frequência de corte a frequência na qual o sinal possui atenuação de $-3dB$ e esta foi escolhida de forma a permitir que o sinal na banda de frequências da respiração fosse preservado.

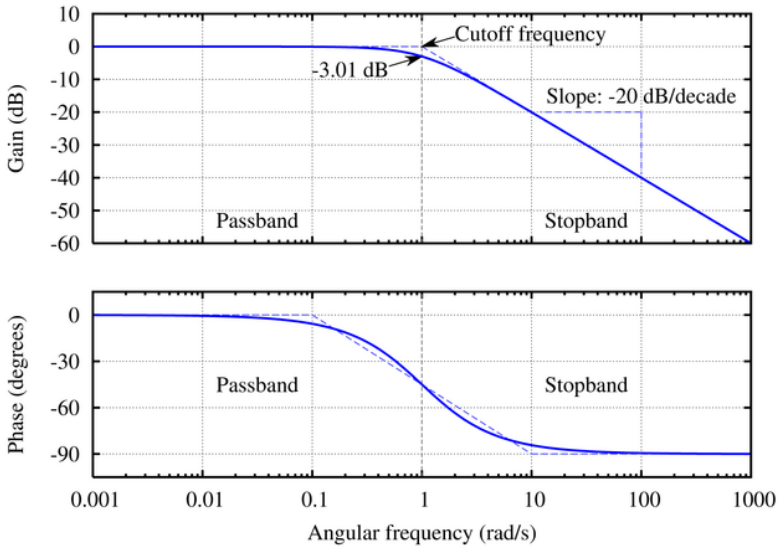

Figura 3.11: Resposta de um filtro Butterworth

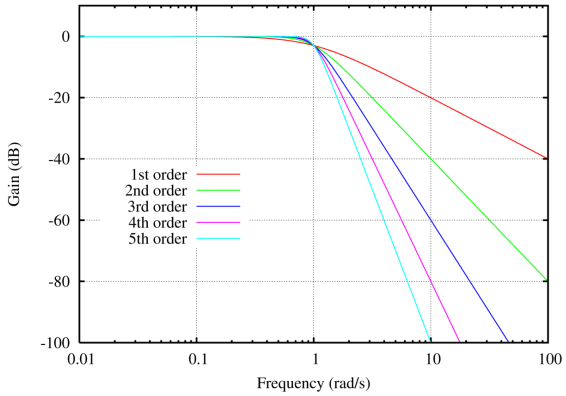

Figura 3.12: Resposta de um filtro Butterworth

FIR Finite Impulse Response

A implementação do filtro FIR foi realizada utilizando o método de janelas. Nessa primeira fase do projeto, foi utilizada a janela de Hamming, que é definida pela equação 3.13 e possui resposta em frequência conforme a figura 3.13.

$$w(n) = 0,54 - 0,46\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \leq n \leq M-1 \tag{3.13}$$

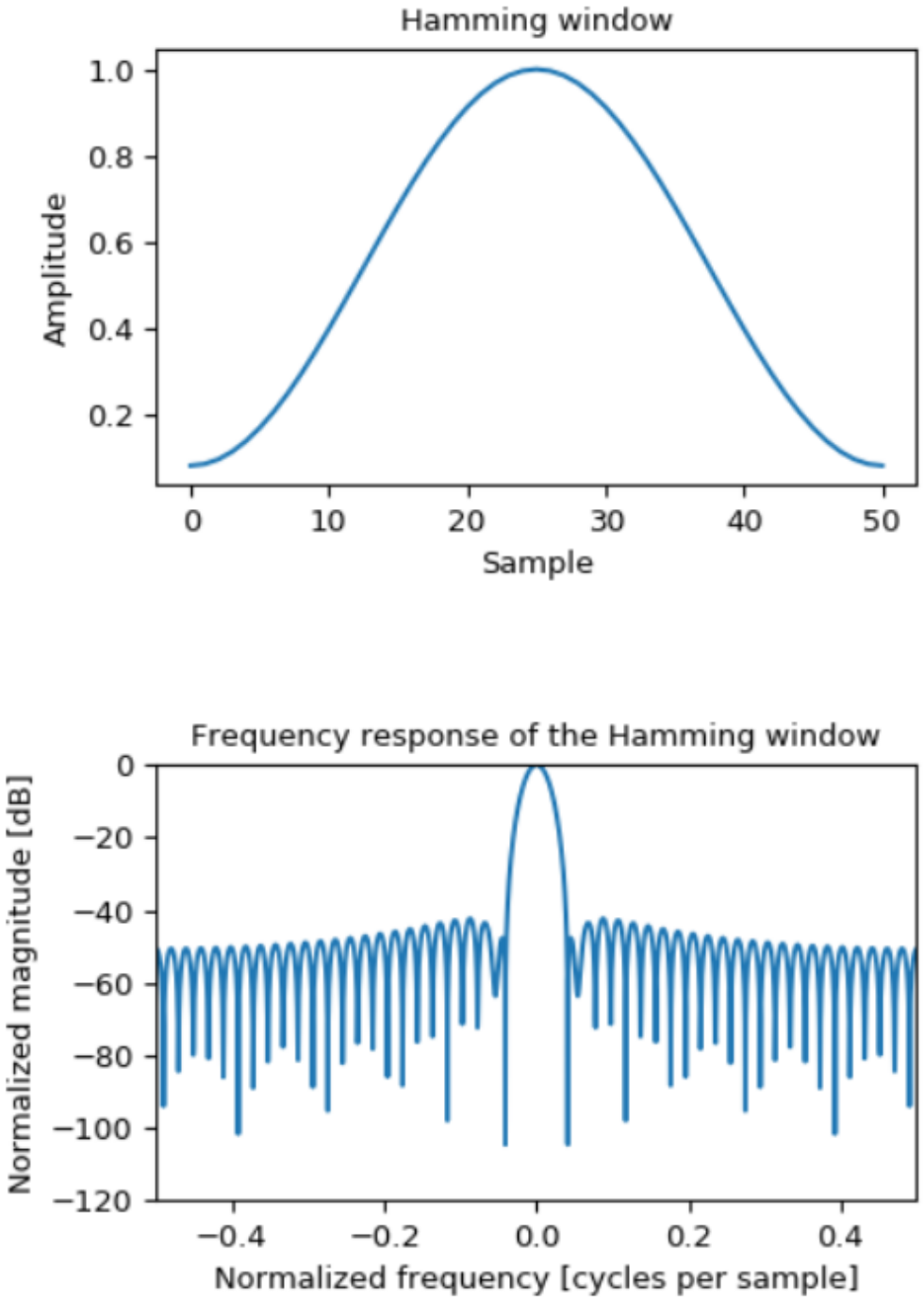

Figura 3.13: Resposta de uma janela de hamming

3.3.2 Evolução do software para um sistema web

Após validado o teste de conceitos, ou seja, após a certeza de que seria factível o desenvolvimento de uma aplicação capaz de ler os dados enviados pelo hardware através de uma porta usb, salvar em um arquivo .csv e realizar as operações ma-

temáticas desejadas para a análise do sinal, foi desenvolvida uma aplicação web, gerando uma interface mais intuitiva e completa, que é capaz de armazenar dados referentes aos experimentos realizados, ao cadastro dos paciente, bem como as medições em um sistema de banco de dados, além de exibir os resultados em um gráfico interativo, com diversas funcionalidades, como aproximar, exportar, deslocar e etc.

Flask

Desde o princípio do desenvolvimento do software, a linguagem escolhida para o processamento dos sinais foi o Python, por estar listada entre as melhores linguagens para o desenvolvimento de aplicações com inteligência artificial [4], sendo esse um campo em potencial para estudos futuros. Para o desenvolvimento de uma aplicação web utilizando o Python, foi escolhido o Flask [5] como framework, por aparentar ser um framework relativamente simples, suficiente para atender aos requisitos do sistema e com potencial escalável para desenvolvimentos futuros.

MongoDB

Atualmente, os bancos de dados são divididos em duas categorias, são elas:

- Banco de dados Relacionais.

- Banco de dados não Relacionais.

Nos bancos de dados relacionais, os dados são armazenados em estruturas denominadas tabelas, onde cada tabela é composta por diversas colunas (atributos) e linhas (dados). Sua linguagem é o SQL (Structured Query Language). Essa categoria de banco de dados fornece uma estrutura consistente, porém rígida, onde cada entidade possui uma tabela e o relacionamento entre as entidades é realizado através de chaves primárias e chaves estrangeiras. Por exemplo, em uma aplicação onde são armazenadas informações na tabela "Paciente"conforme a figura 3.14, os dados armazenados se apresentariam conforme o exemplo da figura 3.15. Se o sistema que estiver utilizando esse modelo de dados precisar evoluir para uma estrutura que armazene as informações sobre endereço de forma mais detalhada, seria necessário criar uma nova tabela "Endereço"com novas colunas e alterar o tipo da coluna "endereço"da tabela "Paciente"para que esta não seja mais uma coluna que armazena o endereço, mas sim que armazene uma chave estrangeira apontando para a chave primária do endereço desejado, conforme as figuras 3.16 e 3.17. Logo, nota-se que, em um cenário onde o sistema já possuía muitos pacientes com seus respectivos endereços cadastrados, seria necessário que cada endereço já existente fosse desmembrado e inserido nas colunas da tabela endereço, adicionando uma complexidade considerável na tentativa de evolução do sistema. Mais ainda, supomos que, em determinado

momento, o sistema precise evoluir de maneira que um paciente possa cadastrar mais de um endereço. Em um banco relacional, seguindo a boa prática de modelagem de banco, seria necessário remover a coluna "endereço"da tabela "Paciente"e criar uma terceira tabela chamada "relPacienteEndereço", por exemplo, onde cada linha é responsável por relacionar pacientes e endereços (figuras 3.18 e 3.19).

Portanto, para projetar um sistema que fosse maleável o suficiente e permitisse uma evolução fluida em projetos futuros, não era desejada a rigidez de um banco de dados relacional, sendo utilizado o banco não relacional. Nessa categoria, as informações são separadas em coleções e não possuem rigidez para seus atributos. Com o sistema construído com esse modelo de dados, se fosse necessário o mesmo processo evolutivo que descrevemos como exemplo para o banco de dados relacional, poderíamos, por exemplo manter os dados de endereço que já foram cadastrados na coleção paciente em um primeiro momento e, para os novos pacientes cadastrados na mesma coleção, armazenar um ou mais ids que apontam para a coleção Endereço (Conforme as figuras 3.20 e 3.21).

Figura 3.14: Tabela Paciente banco relacional

Id	Nome	CPF	Email	Endereço
1	Bruno	123.324.223-28	bruno@poli.ufrj.br	Rua teste 1, centro - Rio de janeiro
2	Tiago	223.544.223-24	tiago@hu.ufrj.br	Rua teste 5, tijuca- Rio de janeiro
3	Carlos	333.544.333-34	carlos@eel.ufrj.br	Rua teste 2, centro - Rio de janeiro
4	Gabriel	443.544.443-44	gabriel@cht.br	Rua teste 2, botafogo - Rio de janeiro

Figura 3.15: Dados exemplo da tabela paciente

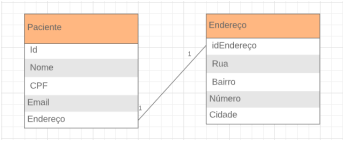

Figura 3.16: Relacionamento paciente x endereço

Paciente

Id	Nome	CPF	Email	Endereço
1	Bruno	123.324.223-28	bruno@poli.ufrj.br	3
2	Tiago	223.544.223-24	tiago@hu.ufrj.br	2
3	Carlos	333.544.333-34	carlos@eel.ufrj.br	1
4	Gabriel	443.544.443-44	gabriel@cht.br	4

Endereço

IdEndereço	Rua	Bairro	Número	Cidade
1	Rua teste	Centro	2	Rio de Janeiro
2	Rua teste	Tijuca	5	Rio de Janeiro
3	Rua teste	Centro	1	Rio de Janeiro
4	Rua teste	Botafogo	2	Rio de Janeiro

Figura 3.17: Dados exemplo das tabelas paciente e endereço

Figura 3.18: Relacionamento paciente x endereço

Paciente

Id	Nome	CPF	Email
1	Bruno	123.324.223-28	bruno@poli.ufrj.br
2	Tiago	223.544.223-24	tiago@hu.ufrj.br
3	Carlos	333.544.333-34	carlos@eel.ufrj.br
4	Gabriel	443.544.443-44	gabriel@cht.br

Endereço

IdEndereço	Rua	Bairro	Número	Cidade
1	Rua teste	Centro	2	Rio de Janeiro
2	Rua teste	Tijuca	5	Rio de Janeiro
3	Rua teste	Centro	1	Rio de Janeiro
4	Rua teste	Botafogo	2	Rio de Janeiro
5	Rua teste	Fundão	30	Rio de Janeiro

relPacienteEndereço

IdPaciente	IdEndereco
1	3
1	5
2	2
3	1
4	4

Figura 3.19: Dados exemplo das tabelas paciente e endereço

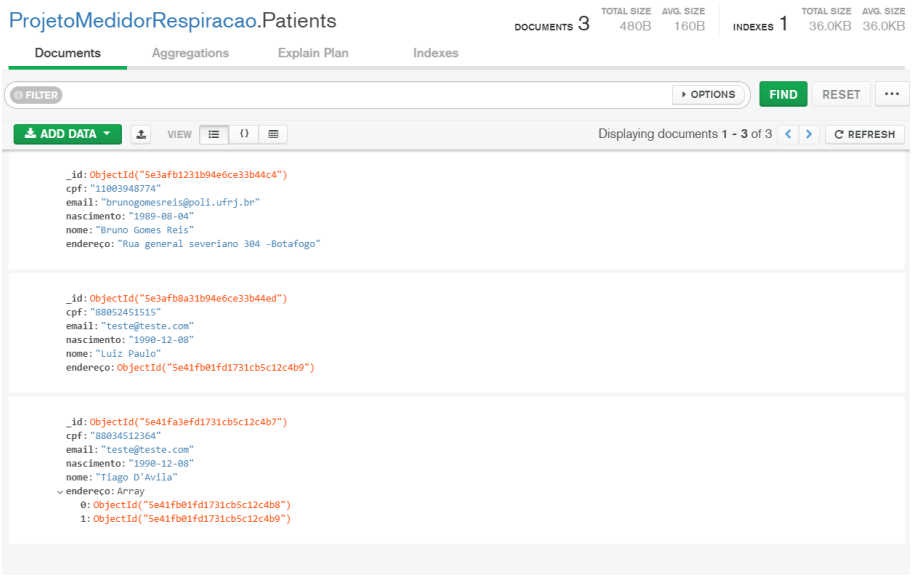

Figura 3.20: Coleção "Paciente"no banco de dados não relacional

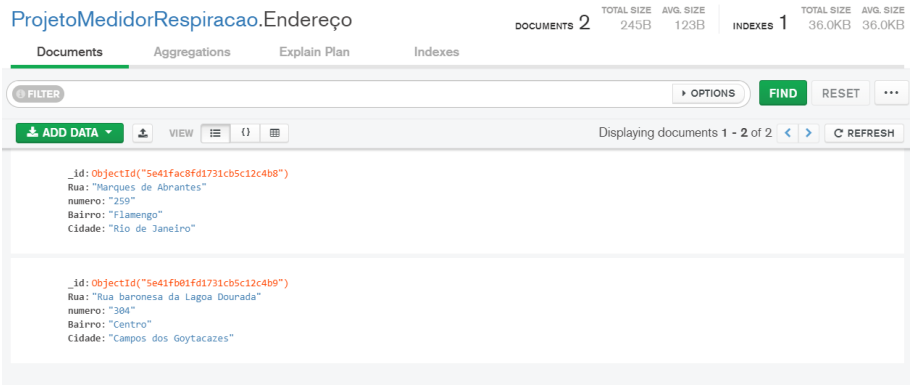

Figura 3.21: Coleção "Endereço"no banco de dados não relacional

Após a escolha do Flask como framework e do MongoDb (Banco de dados não relacional) como base de dados, foi desenvolvido o sistema web. Em seu menu superior é possível acessar 3 telas, são elas:

- Home

- Cadastro

- Experimento

Tela de cadastro de Pacientes

Ao selecionar a opção "Cadastro"no menu superior, o usuário é capaz de registrar um paciente, inserindo as informações "nome", "CPF", "E-mail"e "Data de Nascimento"(Figura 3.22). Após preenchidas, se CPF inserido for um CPF válido, as

informações são armazenadas no banco de dados (Figuras 3.23 e 3.24). Caso contrário, o sistema exibe uma mensagem informando que trata-se de um CPF inválido (Figura 3.25). Para facilitar que o usuário observe a invalidade do CPF, o campo altera de cor dinamicamente, ficando verde caso o CPF seja válido e vermelho caso não seja (Figuras 3.26 e 3.27).

Para validar se trata-se de um CPF válido, o sistema foi desenvolvido utilizando um algorítimo fornecido pelo próprio ministério da fazenda, que consiste, basicamente, em operações utilizando os 9 primeiros dígitos e comparando o resultado dessas operações com os 2 últimos dígitos.

Utilizando, como exemplo, o CPF fictício 023.549.477-12

Validação do primeiro dígito:

Para validar o primeiro dígito, o sistema precisa multiplicar os 9 primeiros dígitos por uma sequência decrescente de 10 à 2. No nosso caso exemplo, teremos $0 * 10 + 2 * 9 + 3 * 8 + 5 * 7 + 4 * 6 + 9 * 5 + 4 * 4 + 7 * 3 + 7 * 2 = 197$ após realizada essa operação, o resultado deve ser multiplicado por $\frac{10}{11}$ e o resto dessa operação deve ser igual ao primeiro dígito da validação. No nosso caso $197 * \frac{10}{11} = \begin{cases} 179, & \text{Quociente} \\ 1, & \text{Resto} \end{cases}$, portanto, o resto está igual ao primeiro dígito dos últimos 2 dígitos de validação.

Validação do segundo dígito:

Passada a validação do primeiro dígito, para que o cpf seja válido, é necessário validar o segundo dígito. Para isso, o sistema precisa utilizar os 9 primeiros dígitos e o primeiro dígito da validação em uma operação similar à realizada na primeira verificação. Todos os números são multiplicados por uma escala que vai de 11 à 2 de acordo com sua posição. Sendo assim, para o nosso exemplo, teremos $0*11+2*10+3*9+5*8+4*7+9*6+4*5+7*4+7*3+1*2 = 240$ e $240*\frac{10}{11} = \begin{cases} 218, & \text{Quociente} \\ 2, & \text{Resto} \end{cases}$. Sendo o resto igual ao último dígito de validação, o CPF está eleito para ser um CPF válido.

Números inválidos conhecidos

Além das operações de validação dos primeiro e segundo dígito, o algorítimo elimina alguns casos de CPFs inválidos conhecidos que podem furar as regras de validação, como é o caso dos CPFs com números repetidos (Ex: 111.111.111-11).

Tela de Experimentos

Ao acionar a opção "Experimento"no menu superior, o usuário é capaz é direcionado para uma tela (Figura 3.28) em que pode escolher um experimento já cadastrado (Figura 3.29) ou cadastrar um novo experimento, além de associar pacientes já cadastrados e fazer o upload de arquivos .csv, que podem conter os dados obtidos pelo hardware 3.3.1 ou dados de qualquer outra origem, desde que estejam divididos

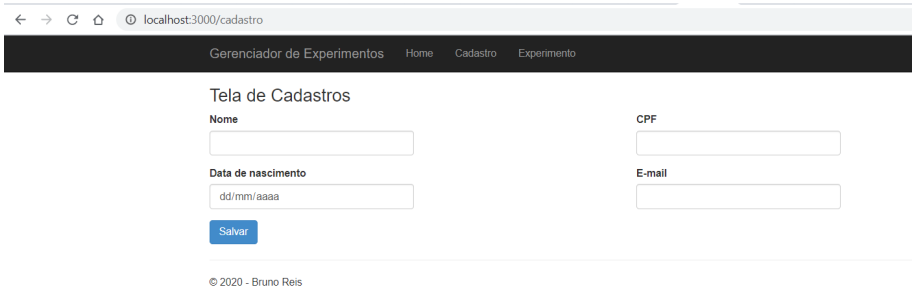

Figura 3.22: Tela de Cadastros

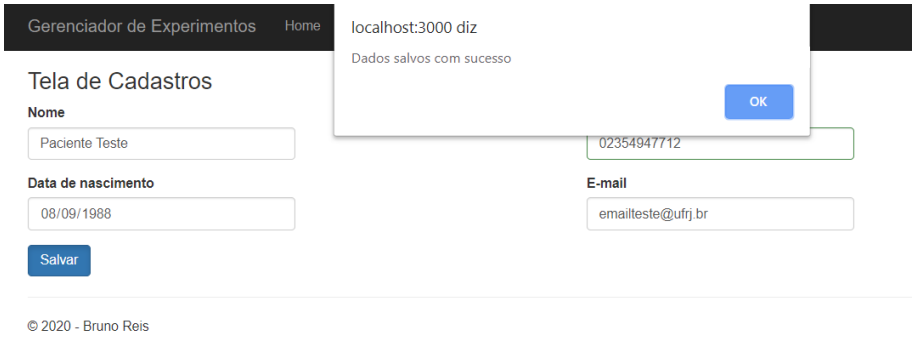

Figura 3.23: Paciente Salvo

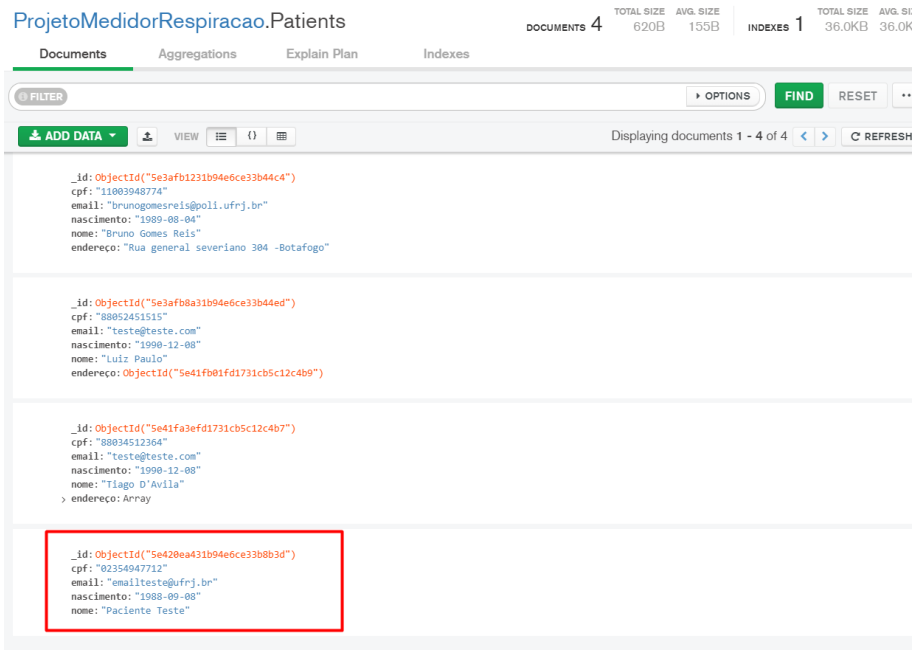

Figura 3.24: Paciente Salvo no Banco de Dados

com o eixo x na primeira coluna e o eixo y na segunda coluna, ambas com cabeçalho e começando a inserção dos dados na linha 2. Ao acionar o botão de upload, o sistema

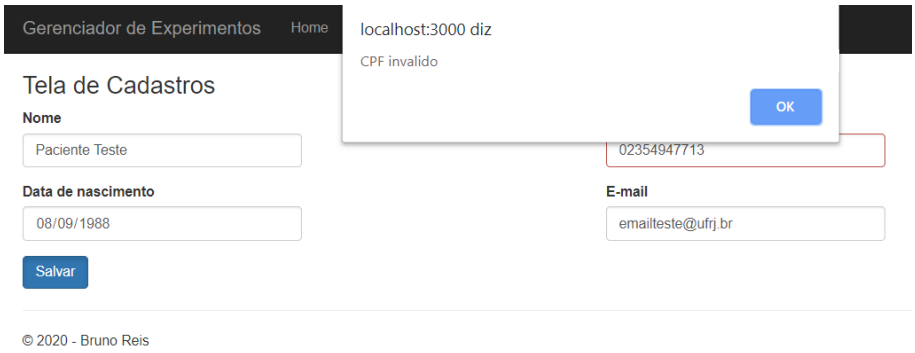

Figura 3.25: Mensagem de Erro ao salvar Paciente

Figura 3.26: Campo com CPF inválido

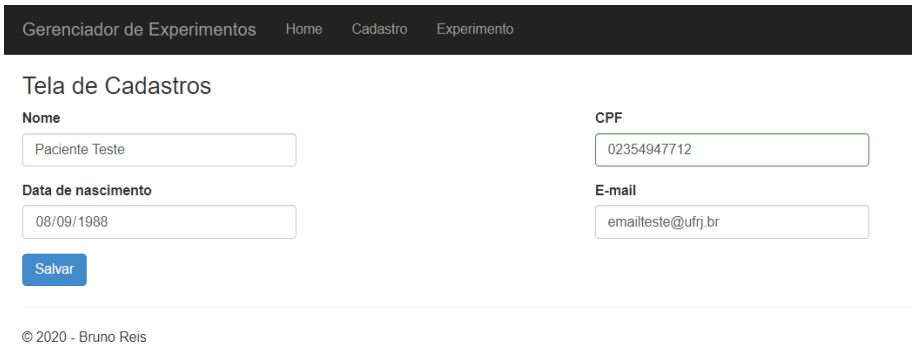

Figura 3.27: Campo com CPF válido

abre uma janela para que o usuário procure e adicione o arquivo .csv armazenado em seu computador local (Figura 3.30).

Tela Principal

Ao acessar a página principal, ou ao selecionar a opção "Home"no menu principal, o sistema irá exibir a tela da figura 3.31. Nessa tela, será exibido um gráfico aleatório e os campos "Experimento"e "Paciente"não estarão preenchidos. Ao apertar em

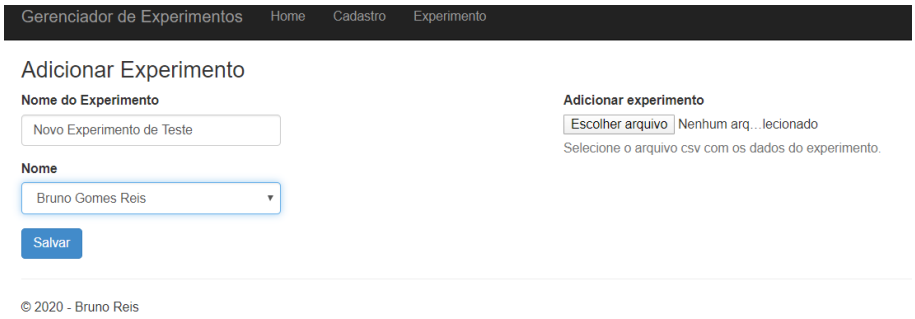

Figura 3.28: Tela de Experimento

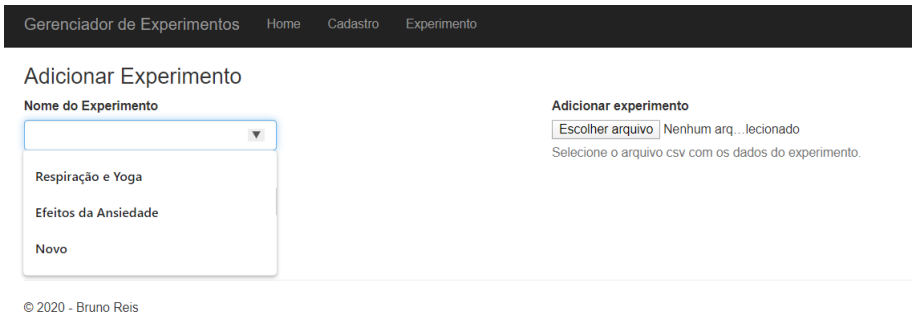

Figura 3.29: Lista de experimentos existentes no banco de dados

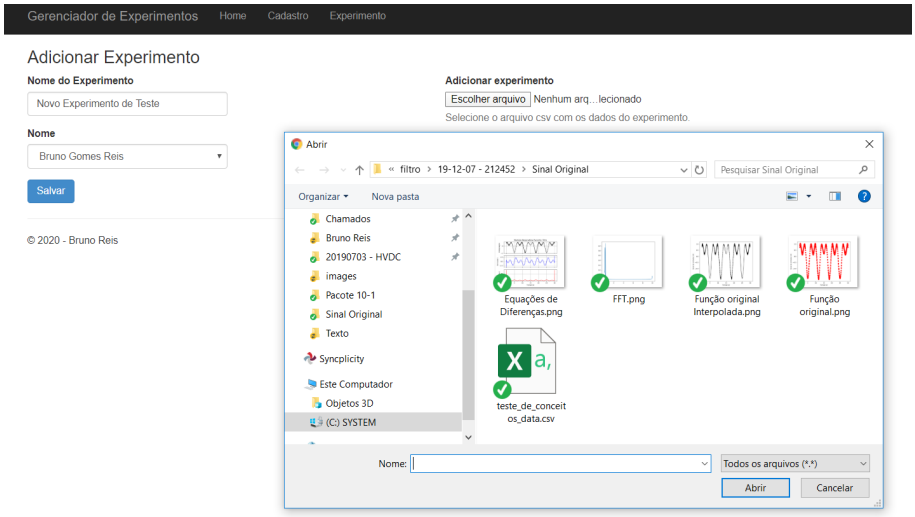

Figura 3.30: Tela para upload de arquivo .csv

algum desses campos, será exibido um modal 3.32 com a lista de experimentos ou pacientes cadastrados para que o usuário possa selecionar qual medição ele deseja analisar. Ao escolher o experimento e o paciente desejado, o usuário pode utilizar qualquer um dos 4 botões localizados na parte inferior da tela para escolher qual análise quer exibir. As opções são:

- - Sinal quantizado
- - Sinal interpolado
- - Análise de Fourier
- - Equação de diferenças

Ao posicionar o mouse em cima dos botões "Equação de Diferenças"e "Análise de fourier"é exibido um aviso com as respectivas formulas (Fuguras 3.33 e 3.34)

É possível também encontrar no canto superior direito uma lista contendo os filtros possíveis de serem utilizados:

- FiltroOriginal
- Filtro IIR
- Filtro FIR

Todas as operações matemáticas realizadas foram descritas na seção 3.3.1.

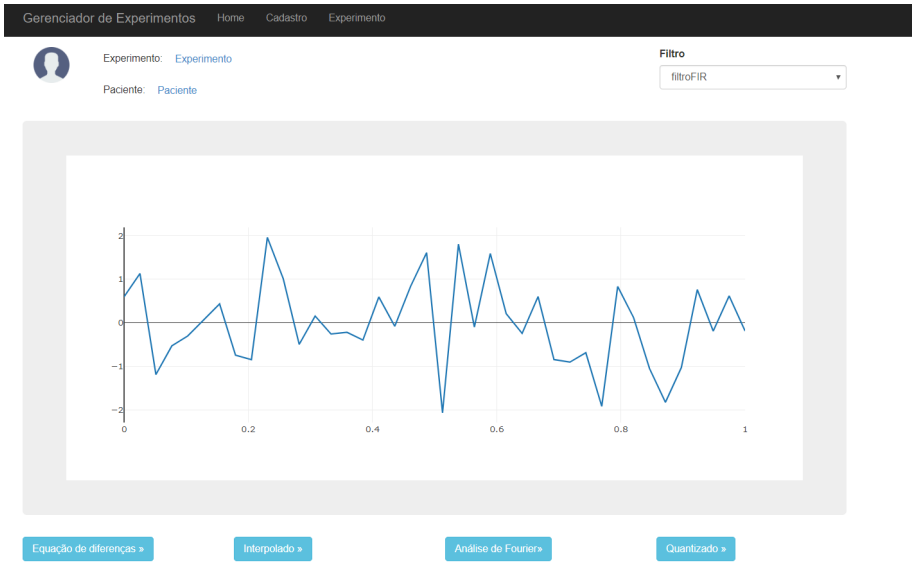

Figura 3.31: Tela Principal

Para facilitar a visualização dos dados, o sistema exibe o valor dos eixos ao posicionar o mouse em cima do local desejado no gráfico (3.35) e, caso necessário, o usuário pode arrastar a tela com o mouse dando um zoom na região desejada (Figuras 3.36 e 3.37).

Por fim, existem algumas funcionalidades úteis no canto superior direito do gráfico, que permitem ao usuário dar zoom in, zoom out, fazer download do gráfico, deslocar o gráfico para esquerda ou direita, exibir linhas no eixo x e y e retornar a escala para o valor original.

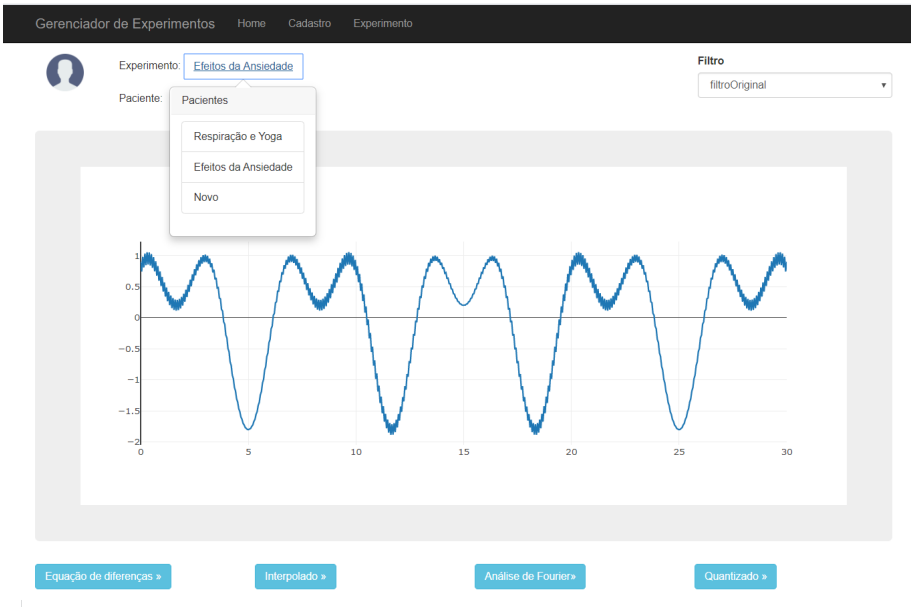

Figura 3.32: Tela Principal

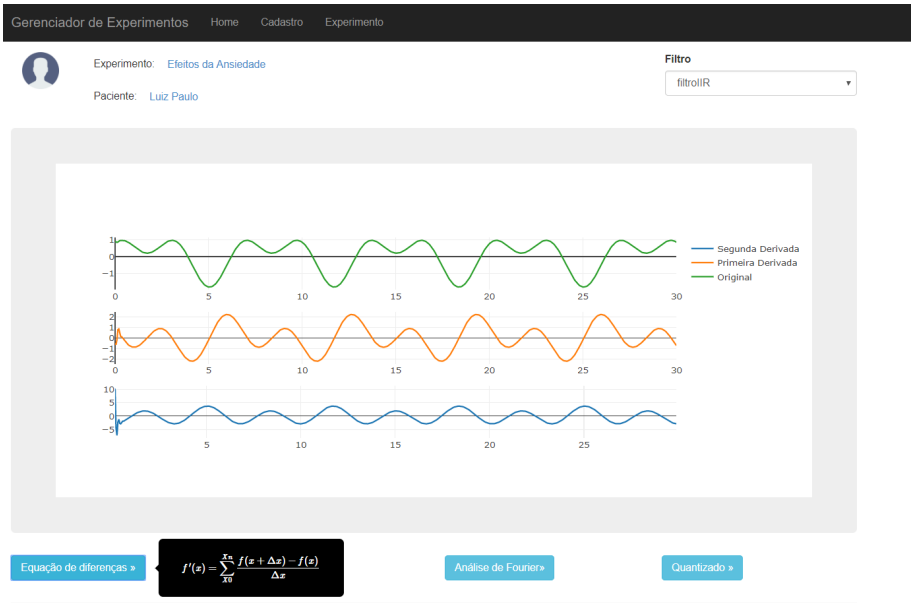

Figura 3.33: Equação de diferenças

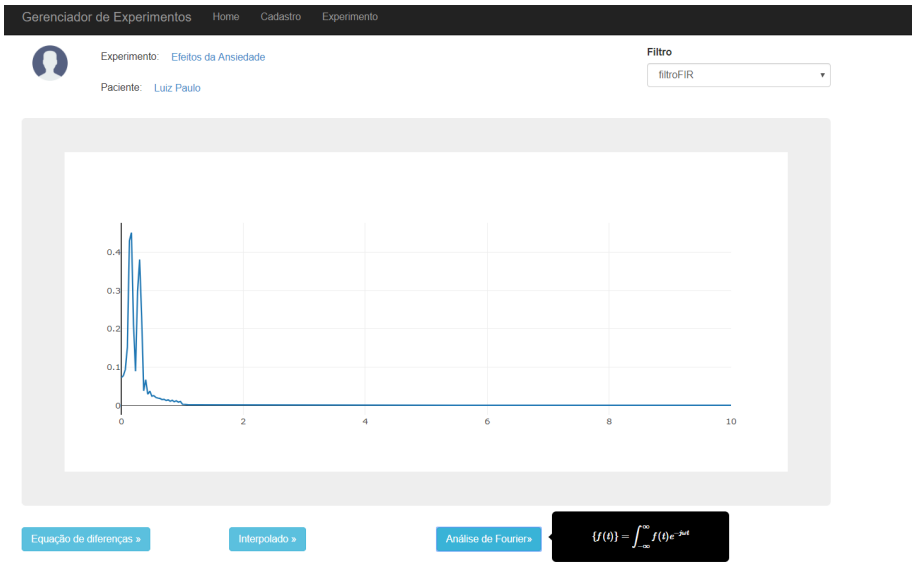

Figura 3.34: Análise de Fourier

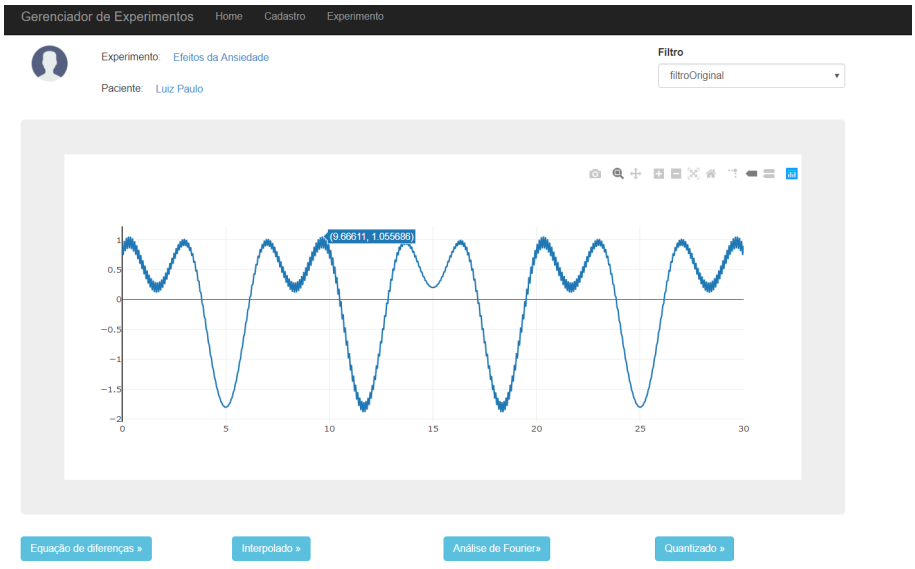

Figura 3.35: Exibindo valor do ponto no gráfico

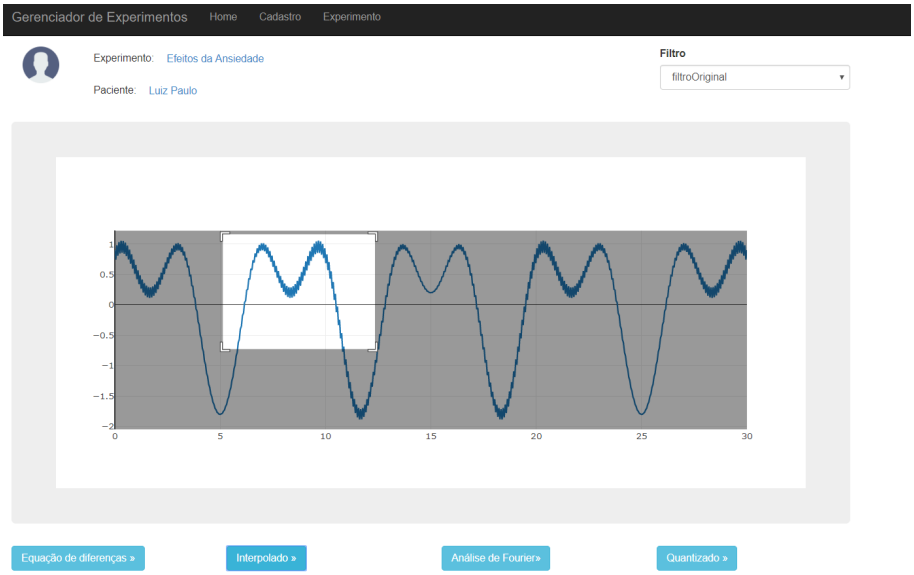

Figura 3.36: Selecionando área do gráfico desejada

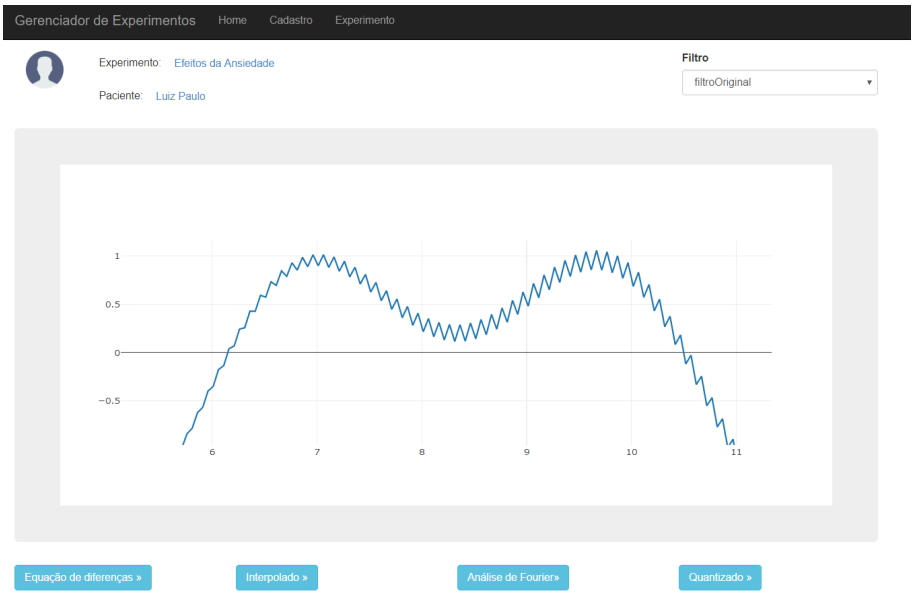

Figura 3.37: zoom na área selecionada

Capítulo 4

Conclusão

4.1 Limitações

As principais limitações deste trabalho são:

4.1.1 Hardware

Em relação ao hardware do projeto, a principal limitação encontrada é referente ao resfriamento do sensor na utilização de longo prazo. A última versão desenvolvida do circuito neste trabalho ainda não possui um sistema de controle eficaz o suficiente para contornar esse problema, tornando-o inviável para a utilização prática na medição respiratória de pacientes em estudos científicos.

4.1.2 Software

A não conclusão de um hardware funcional acarretou em uma limitação por parte do software, que foi desenvolvido a partir de um sinal simulado, sem entrada integrada com o equipamento. Muito embora neste trabalho tenha sido desenvolvido também um software capaz de ler o sinal de entrada e armazenar em um arquivo .csv, o sistema não opera de forma integrada, sendo necessário que o usuário execute um programa para gerar o arquivo .csv e, em seguida, fazer o upload desse arquivo no software responsável pela realização das operações matemáticas.

4.2 Trabalhos futuros

As próximas evoluções do trabalho consistem em desenvolver um sistema de controle capaz de manter a corrente no termistor constante, aumentando o fornecimento de tensão sempre que o sensor resfriar e aumentar sua resistência, diminuindo o fornecimento de tensão sempre que a temperatura do sensor estiver ultrapassando

à necessária para a medição respiratória e obtendo, através do erro, uma medida de tensão que refletirá o comportamento respiratório. Uma vez que esse sistema funcione de forma funcional, realizar a integração entre o sistema, o leitor de entrada e o software de cálculos.

4.3 Considerações finais

O desenvolvimento de um equipamento de baixo custo, capaz de monitorar o comportamento respiratório de pacientes para estudos científicos resulta em uma contribuição significativa para o desenvolvimento da sociedade, uma vez que, diversos estudos científicos poderiam se beneficiar de tal artefato. Em especial, o laboratório de neuroimagem e psicofisiologia, que realiza diversos estudos sobre o comportamento humano, incluindo estudos com análise e medições de dados respiratórios, beneficiaria-se com o desenvolvimento de um equipamento capaz de realizar tais medições por um baixo custo. Muito embora a última versão do hardware desenvolvida não tenha sido satisfatória para o uso prático, este trabalho contribui para o desenvolvimento futuro indicando diversos problemas encontrados, bem como as soluções utilizadas para contorná-los, além de traçar um caminho bem definido para a continuação de seu desenvolvimento. Como entrega concreta e funcional, o trabalho culminou em um software capaz de realizar operações matemáticas de forma fácil e gerar gráficos que poderão ser utilizados pelos pesquisadores em suas análises. A resposta à pergunta fundamental deste trabalho, referente à viabilidade do desenvolvimento de um equipamento de baixo custo capaz de realizar uma medição respiratória confiável, apesar de por hora ainda inconclusiva uma vez que o equipamento não foi, de fato, construído de maneira funcional, tende a ser positiva, avaliando os problemas e análises encontrados neste trabalho, bem como a sua projeção para trabalhos futuros.

Referências Bibliográficas

[1] GUSMÃO, S. "História da Medicina", *JBNC-JORNAL BRASILEIRO DE NEUROCIRURGIA*, v. 15, n. 1, pp. 5–10, 2004.

[2] SPIACCI JR, A., DE OLIVEIRA SERGIO, T., DA SILVA, G., et al. "Serotonin in the dorsal periaqueductal gray inhibits panic-like defensive behaviors in rats exposed to acute hypoxia", *Neuroscience*, v. 307, pp. 191–198, 2015.

[3] SPIACCI JR, A., VILELA-COSTA, H. H., SANT'ANA, A. B., et al. "Panic-like escape response elicited in mice by exposure to CO2, but not hypoxia", *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, v. 81, pp. 178–186, 2018.

[4] "5 Best Programming Languages for Artificial Intelligence in 2020". Disponível em: <https://dev.to/saikrishna_iam/top-5-best-programming-languages-for-artificial-intelligence-in-2020-5ghj>

[5] "Flask". Disponível em: <https://www.palletsprojects.com/p/flask/>.

Apêndice A

Software teste de conceitos

```python
# Esse programa deverá percorrer as seguintes etapas:
#     1- Ler os dados de uma porta serial via usb que serão enviados por um arduíno
#     2- Criar uma pastas com o nome e horário das medições para o armazenamento dos arquivos
#     3- Salvar os dados em um arquivo .csv
#     4- Ler os dados do arquivo csv e realizar as seguintes operações:
#         4.1- Plotar diretamente
#         4.2- Plotar uma interpolação
#         4.3- Calcular a derivada primeira e segunda e plotar
#         4.4- Fazer uma FFT e plotar
#     5- Ler o arquivo csv, passar por um filtro IIR passa baixas, salvar um novo csv e realizar as seguintes operações:
#         5.1- Plotar diretamente
#         5.2- Plotar uma interpolação
#         5.3- Calcular a derivada primeira e segunda e plotar
#         5.4- Fazer uma FFT e plotar
#     6- Ler o arquivo csv, passar por um filtro IIR passa baixas, salvar um novo csv e realizar as seguintes operações:
#         6.1- Plotar diretamente
#         6.2- Plotar uma interpolação
#         6.3- Calcular a derivada primeira e segunda e plotar
#         6.4- Fazer uma FFT e plotar

import numpy as np
import matplotlib.pyplot as plt
import csv
import os
import datetime
from scipy.fftpack import fft
from scipy import signal

#Funções
def gerarCsv(x,y,endereco):
    with open(endereco + "\\teste_de_conceitos_data.csv", mode='w') as csv_file:
        fieldnames = ['time', 'value']
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        writer.writeheader()
        for time in range(0,len(x),1):
            writer.writerow({'time': x[time], 'value': y[time]})

def GerarGráficosDeEquacaoDeDiferencas(x,y,df,df2,xlim,paciente,nomeDoArquivo):

    titulo = 'Medição Respiratória Paciente: ' + paciente
    plt.figure()

    ax1 = plt.subplot(311)
    plt.title(titulo)
    plt.ylabel('Interpolação')
    plt.grid(True)
    plt.plot(x,y,color='black')
    plt.setp(ax1.get_xticklabels(), visible=False)

    ax2 = plt.subplot(312, sharex=ax1)
    plt.grid(True)
    plt.ylabel('Derivada 1')
    plt.plot(x,df, color='blue')
    plt.setp(ax2.get_xticklabels(), visible=False)

    ax3 = plt.subplot(313, sharex=ax1)
    plt.ylabel('Derivada 2')
    plt.xlabel('Tempo (s)')
```

```
    plt.grid(True)
    plt.plot(x,df2, color='red')
    plt.setp(ax3.get_xticklabels())

    plt.savefig(nomeDoArquivo)

def GeraEquacaoDeDiferencas(x,y):
    df =  np.zeros_like(x)       # df/dx
    dx = x[1] - x[0]
    # Internal mesh points
    for i in range(1, len(x) - 1):
        df[i] = (y[i+1] - y[i-1])/(2*dx)
    # End points
    df[0]  = (y[1]  - y[0]) /dx
    df[-1] = (y[-1] - y[-2])/dx
    return df

# Simulando a etapa 1
#Gerando um sinal padrão para testes
N = 600                 #Numero de pontos
T = 0.05                #Taxa de amostragem
x = np.linspace(0.0, N*T, N)
y = np.sin(0.15 * 2.0*np.pi*x)  + 0.8*np.sin(np.pi/2 + 0.3 * 2.0*np.pi*x) +
0.1*np.sin(np.pi/2 + 30 * 2.0*np.pi*x)
y2 = np.sin(0.15 * 2.0*np.pi*x)
y3 = 0.8*np.sin(np.pi/2 + 0.3 * 2.0*np.pi*x)


#Etapa 2
def CriarNovoDiretorio(path):
    try:
        os.makedirs(path)
    except OSError:
        print ("Creation of the directory %s failed" % path)
    else:
        print ("Successfully created the directory %s " % path)

print('Digite o nome do Paciente:')
paciente = input()
subpasta = os.getcwd() + "\\" + "Medições" + "\\" + paciente + "\\" +
datetime.datetime.now().strftime("%y-%m-%d - %H%M%S") + "\\"
pastaOriginal = "Sinal Original\\"
pastaIIR = "Sinal Filtrado IIR\\"
pastaFIR = "Sinal Filtrado FIR\\"
print(subpasta)
if not os.path.exists(subpasta):
    CriarNovoDiretorio(subpasta)
    CriarNovoDiretorio(subpasta + pastaOriginal)
    CriarNovoDiretorio(subpasta + pastaIIR)
    CriarNovoDiretorio(subpasta + pastaFIR)


#Etapa 3
gerarCsv(x,y,subpasta + pastaOriginal)

#Etapa 4
#4.1
plt.figure()
plt.plot(x,y,'ro')
plt.xlabel('Tempo (s)')
plt.ylabel('Sinal de entrada (int)')
```

```
118 plt.savefig(subpasta + pastaOriginal + 'Função original')
119
120 #4.2
121 plt.figure()
122 plt.plot(x,y,color='black')
123
124 plt.xlabel('Tempo (s)')
125 plt.ylabel('Sinal de entrada (int)')
126 plt.savefig(subpasta + pastaOriginal + 'Função original Interpolada')
127
128 #4.3
129 #derivando por equaçaõ de diferenças
130 df = GeraEquacaoDeDiferencas(x,y)
131 df2 = GeraEquacaoDeDiferencas(x,df)
132 GerarGráficosDeEquacaoDeDiferencas(x,y,df,df2,N*T,paciente,subpasta + pastaOriginal
    + 'Equações de Diferenças')
133
134 #4.4
135 yf = fft(y)
136 xf = np.linspace(0.0, 1.0/(2.0*T), N//2)
137
138 plt.figure()
139 plt.plot(xf, 2.0/N * np.abs(yf[0:N//2]))
140 plt.savefig(subpasta + pastaOriginal + 'FFT')
141
142
143 #5
144 b, a = signal.butter(3, 0.5)
145 zi = signal.lfilter_zi(b,a)
146 y_filtrado, _ = signal.lfilter(b,a, y, zi=zi*y[0])
147 gerarCsv(x,y_filtrado,subpasta + pastaIIR )
148
149
150 #5.1
151 plt.figure()
152 plt.plot(x,y_filtrado,'ro')
153 plt.xlabel('Tempo (s)')
154 plt.ylabel('Sinal de entrada (int)')
155 plt.savefig(subpasta + pastaIIR + 'Função original')
156
157 #5.2
158 plt.figure()
159 plt.plot(x,y_filtrado,color='black')
160 plt.xlabel('Tempo (s)')
161 plt.ylabel('Sinal de entrada (int)')
162 plt.savefig(subpasta + pastaIIR + 'Função original Interpolada')
163
164 #5.3
165 df = GeraEquacaoDeDiferencas(x,y_filtrado)
166 df2 = GeraEquacaoDeDiferencas(x,df)
167 GerarGráficosDeEquacaoDeDiferencas(x,y_filtrado,df,df2,N*T,paciente,subpasta +
    pastaIIR + 'Equações de Diferenças')
168
169 #5.4
170 yf2 = fft(y_filtrado)
171 plt.figure()
172 plt.plot(xf, 2.0/N * np.abs(yf2[0:N//2]), 'r')
173 plt.savefig(subpasta + pastaIIR + 'FFT')
174
175
176
```

```
177 #6
178 c = signal.firwin(N, 0.1)
179 y_filtrado2 = signal.lfilter(c,[1.0], y)
180 gerarCsv(x,y_filtrado2,subpasta + pastaFIR )
181
182 #6.1
183 plt.figure()
184 plt.plot(x,y_filtrado2,'ro')
185 plt.xlabel('Tempo (s)')
186 plt.ylabel('Sinal de entrada (int)')
187 plt.savefig(subpasta + pastaFIR + 'Função original')
188
189 #6.2
190 plt.figure()
191 plt.plot(x,y_filtrado2,color='black')
192 plt.xlabel('Tempo (s)')
193 plt.ylabel('Sinal de entrada (int)')
194 plt.savefig(subpasta + pastaFIR + 'Função original Interpolada')
195
196 #6.3
197 df = GeraEquacaoDeDiferencas(x,y_filtrado2)
198 df2 = GeraEquacaoDeDiferencas(x,df)
199 GerarGráficosDeEquacaoDeDiferencas(x,y_filtrado2,df,df2,N*T,paciente,subpasta +
    pastaFIR + 'Equações de Diferenças')
200
201 #6.4
202 yf3 = fft(y_filtrado2)
203
204 plt.figure()
205 plt.plot(xf, 2.0/N * np.abs(yf3[0:N//2]), 'r')
206 plt.savefig(subpasta + pastaFIR + 'FFT')
```