

Luana Ribeiro Gomes

Sistema de Monitoramento de Sinais com Ênfase na Análise da Ausculta Pulmonar

Brasil

2024

Luana Ribeiro Gomes

Sistema de Monitoramento de Sinais com Ênfase na Análise da Ausculta Pulmonar

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Universidade do Estado do Amazonas – UEA

Escola Superior de Tecnologia – EST

Engenharia de Controle e Automação

Orientador: José Ruben Sicchar Vilchez

Brasil

2024

Luana Ribeiro Gomes

Sistema de Monitoramento de Sinais com Ênfase na Análise da Ausculta Pulmonar

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Trabalho aprovado. Brasil, 20 de dezembro de 2024:



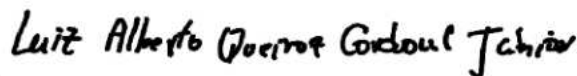
Prof. Dr. José Ruben Sicchar Vilchez
Universidade do Estado do Amazonas (UEA)
- Orientador



Prof. Dr. Florindo Antônio Ayres Jr
Universidade Federal do Amazonas (UFAM)



**Prof. Dr. Victor Enrique Vermehren
Valenzuela**
Universidade do Estado do Amazonas (UEA)



**Prof. Dr. Luiz Alberto Queiroz
Cordovil Junior**
Universidade do Estado do Amazonas (UEA)

Brasil
2024

Agradecimentos

Primeiramente, agradeço a Deus por me proporcionar todas as experiências que moldaram minha vida.

À minha mãe, pelo constante incentivo e apoio incondicional ao longo desta jornada. Agradeço por ser meu exemplo de força, determinação e generosidade, e por sempre acreditar em mim, mesmo nos momentos em que eu duvidava de mim mesma. Sem a sua dedicação, paciência, confiança e investimento eu não teria chegado até aqui.

Ao meu orientador, Prof. Dr. José Ruben Sicchar Vilchez, pelo suporte durante todo o processo de elaboração deste trabalho e por sua orientação ao longo da minha trajetória acadêmica.

Ao meu parceiro de pesquisa, João Victor Bentes Soares, que esteve ao meu lado em todos os momentos desta jornada acadêmica. Agradeço por sua amizade, apoio e companheirismo, que foram fundamentais para que eu pudesse enfrentar os desafios da faculdade. Juntos, passamos por momentos de dificuldades, conquistas e aprendizados, e foi a sua presença constante que fez tudo parecer mais leve.

Ao meu namorado, Emerson Neto, pela disposição em sempre me ajudar, mesmo nos momentos mais desafiadores. Sua generosidade e paciência foram fundamentais em várias etapas desse processo.

Aos meus amigos, José Ricardo e Geanne Olímpio, por suas colaborações em etapas importantes no desenvolvimento deste trabalho.

A todos os amigos e professores do curso de Engenharia de Controle e Automação, pela troca constante de conhecimentos e apoio.

À Universidade do Estado do Amazonas, por me proporcionar diversas oportunidades de aprendizado e crescimento ao longo do curso.

Por fim, agradeço a todos que, de maneira direta ou indireta, contribuíram para a realização deste trabalho.

“Se você controlar sua respiração, você controlará todo o resto.”
(Fidelis, Vinícius Silva)

Resumo

A ausculta pulmonar é uma técnica milenar que consiste na análise dos sons emitidos pelos pulmões durante a respiração, desempenhando um papel fundamental na avaliação clínica e no diagnóstico de condições respiratórias. Apesar de sua simplicidade e eficácia, a prática convencional da ausculta pode ser limitada pela ausência de ferramentas que permitam o monitoramento contínuo, especialmente em pacientes com doenças respiratórias crônicas. Diante disso, o acompanhamento regular desses sinais torna-se crucial para o diagnóstico precoce e o gerenciamento eficaz de condições clínicas, contribuindo para uma melhora na qualidade de vida dos pacientes. Este trabalho apresenta o desenvolvimento de um sistema integrado de *hardware* e *software* que incorpora princípios de instrumentação biomédica com automação, oferecendo uma solução tecnológica para a supervisão de pacientes com possíveis distúrbios respiratórios. O protótipo desenvolvido utiliza sensores para captar sinais pulmonares, realiza o processamento e filtragem desses dados, e os transmite a um aplicativo, facilitando a interação entre pacientes e profissionais de saúde. Além disso, o sistema permite a criação de históricos e o monitoramento remoto, tornando-se uma ferramenta promissora para uso em ambientes clínicos e domiciliares.

Palavras-chave: Ausculta Pulmonar, Doenças Respiratórias, Instrumentação Biomédica, Automação, Monitoramento Remoto.

Abstract

Pulmonary auscultation is an ancient technique that involves analyzing the sounds emitted by the lungs during respiration, playing a fundamental role in clinical evaluation and the diagnosis of respiratory conditions. Despite its simplicity and effectiveness, conventional auscultation practice can be limited by the lack of tools that enable continuous monitoring, especially in patients with chronic respiratory diseases. Therefore, regular monitoring of these signals becomes crucial for early diagnosis and effective management of clinical conditions, contributing to an improved quality of life for patients. This work presents the development of an integrated hardware and software system that incorporates biomedical instrumentation principles with automation, providing a technological solution for the supervision of patients with potential respiratory disorders. The developed prototype uses sensors to capture pulmonary signals, processes and filters the data, and transmits it to an application, facilitating the interaction between patients and healthcare professionals. Additionally, the system allows for the creation of patient histories and remote monitoring, making it a promising tool for use in both clinical and home environments.

Keywords: Pulmonary Auscultation, Respiratory Diseases, Biomedical Instrumentation, Automation, Remote Monitoring.

Lista de ilustrações

Figura 1 – Sistema respiratório.	16
Figura 2 – Pontos de ausculta pulmonar sobre o tórax. (a) Face anterior e (b) Face posterior	17
Figura 3 – Estetoscópio mecânico atual.	18
Figura 4 – Estetoscópio eletrônico.	19
Figura 5 – Sensibilidade limiar (medida em decibéis) do ouvido humano.	20
Figura 6 – Módulo INMP441.	24
Figura 7 – Arquitetura Harvard de um microcontrolador.	24
Figura 8 – Módulo de desenvolvimento Cortex M3 com microcontrolador STM32.	26
Figura 9 – ESP32-WROOM-32.	27
Figura 10 – Fluxograma de desenvolvimento do trabalho.	32
Figura 11 – Arquitetura do sistema.	33
Figura 12 – Arquitetura módulo STM32-WROOM.	33
Figura 13 – Conexão entre o sensor de som INMP441 e o módulo de desenvolvimento STM32.	34
Figura 14 – Arquitetura e dos pinos do ESP32.	35
Figura 15 – Conexão entre o nó sensor e o ESP32-WROOM.	35
Figura 16 – Protótipo inicial.	36
Figura 17 – <i>Design</i> da estrutura do protótipo.	36
Figura 18 – Arquitetura de desenvolvimento do aplicativo.	39
Figura 19 – Fluxograma das principais telas de acesso do usuário.	40
Figura 20 – Estrutura dos níveis de acesso médico e paciente	41
Figura 21 – Estrutura de <i>back-end</i> e <i>front-end</i>	42
Figura 22 – Diagrama de integração do <i>hardware</i> e <i>software</i>	42
Figura 23 – Protótipo final.	44
Figura 24 – Sinais captados pelo sensor INMP441.	45
Figura 25 – Sinal filtrado captado pelo sensor INMP441.	46
Figura 26 – Diagrama de magnitude, destacando a faixa de 100 a 500 Hz.	47
Figura 27 – Diagrama de fase, destacando a faixa de 100 a 500 Hz.	48
Figura 28 – Diagrama de polos e zeros.	49
Figura 29 – Sinais captados pelo sensor INMP441 e ICBHI.	49
Figura 30 – Sinais captados pelo inicialmente pelo sensor INMP441 e STM32.	50
Figura 31 – Sinais captados pelo sensor INMP441 e STM32 filtrados.	51
Figura 32 – Comunicação entre o microcontrolador STM32 e o módulo ESP32.	52
Figura 33 – Comunicação entre o ESP32 e o aplicativo.	52
Figura 34 – Telas de <i>login</i> - Médico.	53

Figura 35 – Telas de acompanhamento de pacientes - Médico.	54
Figura 36 – Telas <i>login</i> - Pacientes.	54
Figura 37 – Telas principais - Pacientes.	55
Figura 38 – Tela principal de monitoramento.	55
Figura 39 – Realização dos testes.	56
Figura 40 – Dados captados nos testes.	57
Figura 41 – Realização dos testes por profissionais da saúde.	57
Figura 42 – Aplicação do <i>quiz</i> para coleta de <i>feedback</i>	58
Figura 43 – Distribuição das Respostas do Quiz de Avaliação.	59
Figura 44 – Página de <i>login</i> usando o <i>authenticate</i> do <i>Firebase</i>	66
Figura 45 – Página de login usando o <i>authenticate</i> do <i>Firebase 2</i>	67
Figura 46 – Página de <i>Home</i> com opções de <i>login e register</i>	67
Figura 47 – Página <i>register</i>	68
Figura 48 – Página de registrar usuário no <i>Firebase</i>	68
Figura 49 – Página de registrar usuário no <i>Firebase 2</i>	69
Figura 50 – Página de Ausculta pulmonar 1.	70
Figura 51 – Página de Ausculta pulmonar 2.	70
Figura 52 – Página de Ausculta pulmonar 3.	71
Figura 53 – Código utilizado para filtragem do sinal 1.	72
Figura 54 – Código utilizado para filtragem do sinal 2.	72
Figura 55 – Código utilizado para filtragem do sinal 3.	73

Lista de abreviaturas e siglas

BLE	<i>Bluetooth Low Energy</i>
CPU	Unidade Central de Processamento
COVID-19	<i>Coronavirus Disease 2019</i>
CSS	<i>Cascading Style Sheets</i>
DPOC	Doença Pulmonar Obstrutiva Crônica
FFT	Transformada Rápida de Fourier
FR	Fisioterapia Respiratória
GPIO	<i>General Purpose Input/Output</i>
HTML	<i>HyperText Markup Language</i>
HTTP	Protocolo de Transferência de Hipertexto
I2C	<i>Inter-Integrated Circuit</i>
ICBHI	<i>International Conference on Biomedical and Health Informatics</i>
IIR	<i>Infinite Impulse Response</i>
INMETRO	Instituto Nacional de Metrologia, Qualidade e Tecnologia
IOT	<i>Internet of Things</i>
ISLA	<i>International Society for Lung Sounds</i>
JSON	<i>JavaScript Object Notation</i>
MCUs	<i>Microcontroller Units</i>
PROM	<i>Programmable Read-Only Memory</i>
RAM	<i>Random Access Memory</i>
SD	<i>Data Out</i>
TFD	Transformada Discreta de Fourier
ULA	Unidade Lógica Aritmética
USB	Universal Serial Bus
UART	<i>Universal Asynchronous Receiver-Transmitter</i>

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	Estrutura do Trabalho	14
2	REFERENCIAL TEÓRICO	15
2.1	Fisioterapia respiratória	15
2.2	Sistema respiratório e a Ausculta pulmonar	16
2.3	Estetoscópio	17
2.3.1	Convencionais (Mecânicos)	18
2.3.2	Eletrônicos	18
2.4	Limitações da Ausculta pulmonar	19
2.5	Classificação dos sons respiratórios	20
2.5.1	Normais	21
2.5.2	Adventícios	21
2.6	Telemedicina	22
2.7	Sensores	23
2.8	Microcontroladores	24
2.8.1	STM32	25
2.8.2	ESP32	27
2.9	Processamento de sinal	28
2.10	Transformada de Fourier	29
2.11	Tecnologia de desenvolvimento	30
2.11.1	<i>React.js</i>	30
2.11.1.1	Características	30
2.11.2	<i>Firebase</i>	31
2.11.3	<i>Figma</i>	31
3	MATERIAS E MÉTODOS	32
3.1	Hardware	32
3.1.1	Montagem e integração dos dispositivos	33
3.1.1.1	Conexão do Sensor de Microfone INMP441 ao STM32	33
3.1.1.2	Conexão entre o STM32 e o ESP32	34
3.1.2	<i>Design</i> da estrutura do protótipo	36
3.1.3	Filtragem do sinal	37

3.1.4	Validação por meio de comparação com o banco de dados	38
3.2	Software	38
3.2.1	Modelagem do fluxograma das telas	39
3.2.2	Implementação do <i>back-end</i> e <i>front-end</i> do sistema	41
3.3	Comunicação entre <i>software</i> e <i>hardware</i>	42
3.4	Validação	43
4	RESULTADOS	44
4.1	Protótipo	44
4.2	Processamento de sinais	45
4.2.1	Análise do sinal captado pelo sensor INMP441	45
4.2.2	Filtragem	50
4.3	Comunicação do sistema	51
4.4	Aplicativo	53
4.4.1	Interface do <i>front-end</i>	53
4.4.1.1	Acesso médico	53
4.4.1.2	Acesso pacientes	54
4.4.2	Implementação do <i>back-end</i> e <i>front-end</i>	55
4.5	Procedimento experimental	56
5	CONCLUSÕES	60
5.1	Trabalhos Futuros	61
	REFERÊNCIAS	62
	APÊNDICES	65
.1	Códigos de Implementação e Configuração do Sistema	66
.1.1	Códigos de cadastro e autenticação no <i>Firebase</i>	66
.1.2	Código das telas de ausculta pulmonar	69
.1.3	Código utilizado para filtragem do sinal.	71

1 Introdução

As doenças respiratórias crônicas, como asma, DPOC (Doença Pulmonar Obstrutiva Crônica), fibrose cística e fibrose pulmonar, têm se destacado como uma preocupação crescente na sociedade contemporânea. Essas condições, frequentemente acompanhadas por comorbidades, refletem a necessidade de métodos inovadores e eficazes para o monitoramento contínuo da saúde respiratória. A ausculta pulmonar, tradicionalmente realizada com estetoscópios, ocupa um papel central na detecção e acompanhamento dessas condições, sendo amplamente reconhecida como uma técnica essencial no diagnóstico clínico. No entanto, a ausência de monitoramento constante, especialmente em pacientes com condições crônicas, pode resultar em diagnósticos tardios, agravamento das doenças e redução significativa da qualidade de vida.

A pandemia de COVID-19 exacerbou essa situação, aumentando a incidência de complicações pulmonares, como a Síndrome do Desconforto Respiratório Agudo (SDRA) e pneumonia viral, em um grande número de indivíduos. Neste contexto, a adoção de tecnologias mais avançadas tem se mostrado promissora. Avanços na compreensão acústica dos sons pulmonares, viabilizados por estudos como os de [Pasterkamp, Kraman e Wodicka \(1997\)](#), demonstram como novas abordagens tecnológicas podem superar as limitações dos estetoscópios tradicionais. Estes instrumentos, conforme [Abella e Penney \(1992\)](#), tendem a distorcer certas frequências de relevância clínica, destacando a necessidade de soluções mais precisas e abrangentes.

Segundo [Campanholi \(2018\)](#) ademais, complicações pulmonares figuram entre as principais causas de morbimortalidade, principalmente em contextos hospitalares, como em Unidades de Terapia Intensiva (UTI). Pacientes em recuperação pós-operatória, sob ventilação mecânica ou com restrições de movimento enfrentam desafios significativos que demandam monitoramento próximo e constante. Nesse cenário, a ausculta pulmonar ganha ainda mais relevância, dado seu caráter não invasivo e sua capacidade de identificar rapidamente alterações respiratórias que podem indicar quadros críticos ou deterioração clínica.

Os avanços tecnológicos possibilitam não apenas uma análise mais precisa dos sons pulmonares, mas também a integração de ferramentas digitais que favorecem a interação médico-paciente. Sensores acústicos, aliados a sistemas computacionais, oferecem uma nova perspectiva para o diagnóstico e acompanhamento das condições respiratórias. Por meio da digitalização da ausculta pulmonar, torna-se viável monitorar pacientes remotamente, permitindo que intervenções sejam realizadas de forma mais ágil e personalizada, principalmente em populações vulneráveis ou em locais de difícil acesso.

Este trabalho propõe o desenvolvimento de um sistema capaz de captar sons pulmonares utilizando sensores integrados a microcontroladores, aplicando filtragem para melhorar a qualidade do som na faixa de frequência entre 100 Hz e 500 Hz, onde se concentram os sons pulmonares mais relevantes para a análise clínica. O sistema transmite esses dados para um aplicativo, visando futuramente possibilitar que profissionais de saúde utilizem a ferramenta para acompanhar e analisar os sinais captados de forma prática e eficiente. A proposta busca oferecer uma base tecnológica para o monitoramento de condições respiratórias, garantindo que os sinais sejam apresentados de forma clara e confiável.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um sistema integrado de *hardware* e *software* para captar, filtrar e transmitir sons pulmonares, garantindo que os sinais sejam apresentados de forma clara e confiável, visando promover suporte no monitoramento e análise de condições respiratórias.

1.1.2 Objetivos específicos

1. Definir as principais faixas de sons pulmonares a serem monitorados;
2. Selecionar os dispositivos necessários para a implementação do sistema de monitoramento;
3. Integrar os componentes do sistema, garantindo sua compatibilidade e funcionamento adequado;
4. Aplicar técnicas de filtragem nos sinais adquiridos, visando aprimorar a qualidade e precisão das medições;
5. Realizar os testes preliminares do sistema desenvolvido;
6. Validar os resultados obtidos por meio de análises comparativas;
7. Modelar o fluxo de telas do aplicativo para o monitoramento dos sinais captados;
8. Desenvolver aplicativo para amostragem e acompanhamento clínico remoto, proporcionando uma interface intuitiva para a coleta de dados e facilitando a gestão eficiente de condições médicas à distância;
9. Validar a comunicação entre *hardware* e *software*;
10. Realizar a validação clínica dos resultados obtidos.

1.2 Estrutura do Trabalho

Este estudo está organizado em capítulos, conforme descrito a seguir. O [Capítulo 2](#) apresenta o referencial teórico, abordando os temas fundamentais para o desenvolvimento deste trabalho. No [Capítulo 3](#), são detalhados os materiais e métodos empregados no projeto. O [Capítulo 4](#) traz os resultados obtidos nos testes realizados. Por fim, o [Capítulo 5](#) reúne as considerações finais e trabalhos futuros.

2 Referencial Teórico

Neste capítulo, destacam-se as principais características das tecnologias aplicadas à ausculta pulmonar, incluindo a fisioterapia respiratória, sensores, microcontroladores e as ferramentas de desenvolvimento utilizadas na solução proposta. Além disso, são apresentados os fundamentos teóricos identificados em relação à implementação do sistema de monitoramento.

2.1 Fisioterapia respiratória

A Fisioterapia Respiratória (FR) compreende um conjunto de técnicas manuais, podendo envolver o uso de equipamentos em determinados momentos, destinados a intervir nos processos obstrutivos, tanto de maneira preventiva quanto curativa. De acordo com [Campos et al. \(2007\)](#), a FR desempenha um papel importante no tratamento de diversas condições pulmonares. Seus principais objetivos são a prevenção do acúmulo de secreções nas vias aéreas, otimização da eficácia da ventilação, promoção da limpeza e drenagem de secreções, aprimoramento do mecanismo respiratório, controle da respiração e a melhoria da efetividade da tosse.

As técnicas, conhecidas como recursos manuais, referem-se a manobras cinesioterápicas respiratórias ou manobras manuais da fisioterapia respiratória [Verri et al. \(2023\)](#). Dividida em quatro categorias específicas de classificação: ambulatorial, hospitalar, pediátrica e domiciliar, a FR demanda a aplicação por profissionais experientes. No entanto, mesmo na ausência de diagnóstico de distúrbios respiratórios, os pacientes podem realizar exercícios para aprimorar sua capacidade pulmonar.

Com a disseminação da COVID-19 e as medidas de isolamento, muitas pessoas deixaram de praticar atividades físicas, levando, frequentemente, ao sedentarismo. Isso pode resultar na diminuição da capacidade pulmonar e aumentar as chances de acúmulo de secreções nos pulmões [Leon, Amaral e Júnio \(2022\)](#). Além disso, em casos mais graves de COVID-19, muitos pacientes demandam intervenções de Fisioterapia Respiratória (FR). Isso se torna importante, pois, dependendo da gravidade da doença, a recuperação pulmonar pode estender-se por um período de três a seis meses [Murphy \(2008\)](#).

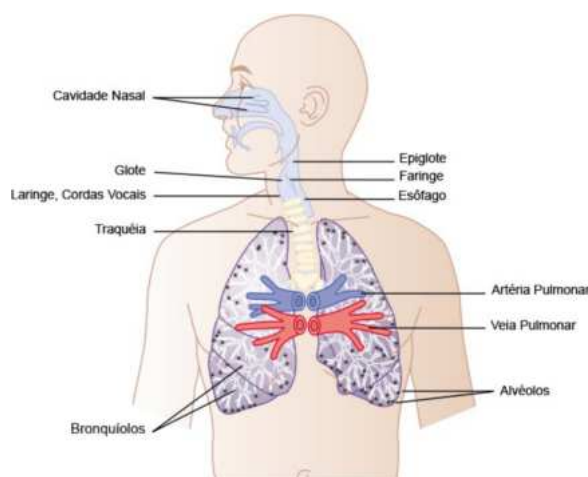
A Fisioterapia Respiratória (FR) desempenha uma significativa função no programa de reabilitação, visando promover e maximizar a independência funcional do paciente em suas atividades diárias. Isso resulta em uma melhoria expressiva na qualidade de vida, refletida no aumento da tolerância ao exercício e na redução da dispneia [Vettorazzi \(2006\)](#). Adicionalmente, a FR contribui para aprimorar a ausculta pulmonar, destacando seus

benefícios abrangentes no bem-estar e na saúde respiratória do indivíduo.

2.2 Sistema respiratório e a Ausculta pulmonar

De acordo com as afirmações de [Bott \(2014\)](#), corroboradas por [Naves \(2015\)](#), o sistema respiratório desempenha um papel fundamental na oxigenação dos tecidos e na eliminação do dióxido de carbono no organismo humano. Esse sistema é composto por uma estrutura complexa que engloba as vias aéreas e os pulmões, incumbidos de conduzir o oxigênio desde as cavidades nasais até as regiões pulmonares, onde ocorrem as trocas gasosas [Hirata \(2004\)](#). A [Figura 1](#) proporciona uma visão abrangente das principais componentes desse sistema, que inclui nariz, cavidade nasal, faringe, laringe, glote, traqueia, brônquios, bronquíolos, alvéolos dentro do pulmão, entre outras estruturas.

Figura 1 – Sistema respiratório.



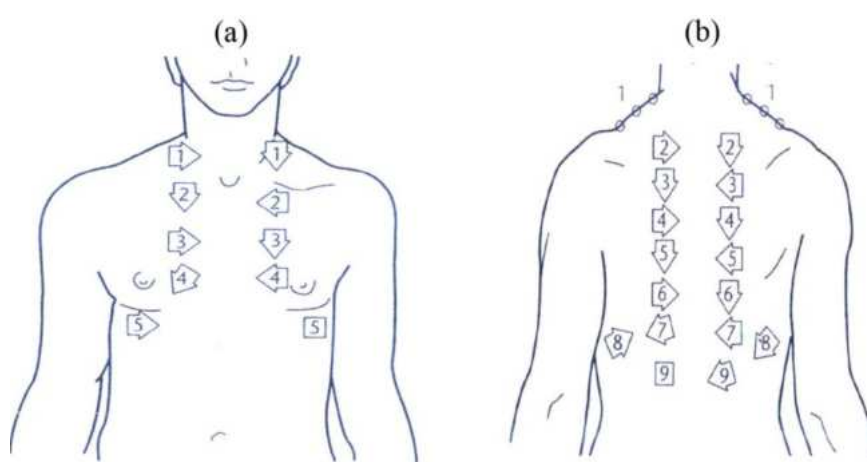
Fonte: Adaptação de [Bott \(2014, p. 288\)](#).

Conforme descrito por [Bott \(2014\)](#), mencionado por [Naves \(2015\)](#), durante a inspiração, o ar percorre as vias aéreas até alcançar os pulmões, resultando na expansão dos alvéolos. Na expiração, o ar é expelido dos alvéolos para fora do corpo. Contudo, a parede envolvente é relativamente rígida e heterogênea devido à presença de músculos e ossos, prejudicando a transmissão dos sons [Carvalho e Souza \(2007\)](#). A análise desses sons requer o uso da técnica de ausculta pulmonar, uma ferramenta essencial na avaliação de doenças respiratórias [Murphy \(2008\)](#). Destaca-se que essa técnica é reconhecida como o método não invasivo mais fundamental para diagnosticar condições como asma, bronquite, pneumonia, doença pulmonar obstrutiva crônica (DPOC), entre outras [Gross et al. \(2000\)](#).

2.3 Estetoscópio

Segundo [Lehrer \(2002\)](#), a técnica de ausculta requer o uso do estetoscópio, que deve ser deslocado para frente e para trás entre as regiões dos segmentos pulmonares comparáveis em ambos os lados do tórax. É necessário realizar esse procedimento em um ambiente silencioso, preferencialmente com o tórax desnudo, seguindo uma sequência específica conforme representado na [Figura 2](#). O processo começa na parede torácica superior, descendo bilateralmente para a comparação, alcançando o sexto espaço intercostal anteriormente, prosseguindo até o décimo espaço intercostal, e nas laterais do tórax correspondentes à área dos pulmões ([BAID, 2006](#)).

Figura 2 – Pontos de ausculta pulmonar sobre o tórax. (a) Face anterior e (b) Face posterior



Fonte: [Lehrer \(2002, p. 75\)](#).

A técnica de ausculta respiratória requer uma respiração profunda pela boca, com o examinador concentrando-se inicialmente na fase de inspiração, analisando seus componentes normais e adventícios. Em seguida, a expiração deve ser analisada da mesma forma ([LEHRER, 2002](#)).

Durante a ausculta do aparelho respiratório, são identificados dois sons distintos dependendo da região examinada. Sobre a laringe e a traqueia, há um ruído de alta intensidade, predominando na fase expiratória e denominado ruído laringotraqueal. Conforme [Lehrer \(2002\)](#), na superfície do tórax, encontra-se um ruído mais suave, predominando na fase inspiratória, conhecido como murmúrio vesicular.

A utilização do estetoscópio representa um dos métodos diagnósticos mais importantes e acessíveis para obter informações sobre a estrutura e função pulmonar de maneira não invasiva ([PASTERKAMP; KRAMAN; WODICKA, 1997](#)). Eles funcionam como simples condutores de sons entre as superfícies corporais do paciente e os ouvidos do examinador, sendo capazes de amplificar ou atenuar seletivamente em determinado espectro de interesse

clínico (PASTERKAMP; KRAMAN; WODICKA, 1997). Atualmente, os estetoscópios se dividem em duas categorias: os convencionais (mecânicos) e os eletrônicos.

2.3.1 Convencionais (Mecânicos)

O estetoscópio mecânico, visto na Figura 3 é um dispositivo simples e direto, projetado para amplificar os sons internos do corpo humano. Ele é composto por uma campânula (parte que entra em contato com o corpo) conectada a tubos de metal que se dividem e vão até os ouvidos do profissional de saúde. Quando a campânula é colocada sobre a área de interesse, como o peito para auscultar os pulmões ou o coração, os sons são transmitidos através dos tubos e alcançam os ouvidos do médico ou enfermeiro.

Figura 3 – Estetoscópio mecânico atual.



Fonte: Naves (2015, p. 20).

A interpretação dos sons auscultados no estetoscópio mecânico requer habilidade e experiência clínicas. O profissional de saúde deve ser capaz de distinguir entre diferentes tipos de ruídos, como batimentos cardíacos normais, sopros cardíacos, crepitações pulmonares, entre outros.

2.3.2 Eletrônicos

O estetoscópio eletrônico, visto na Figura 4, incorpora tecnologia avançada para melhorar a auscultação. Ele é equipado com microfones na campânula que convertem os sons em sinais elétricos. Esses sinais são então processados eletronicamente, permitindo amplificação, filtragem e ajustes de frequência. O som processado é então transmitido aos ouvidos do profissional por meio de fones de ouvido.

Figura 4 – Estetoscópio eletrônico.



Fonte: [Leng et al. \(2015, p. 5\)](#).

Essa versão permite uma auscultação mais clara e pode incluir recursos adicionais, como conectividade *Bluetooth* para gravar ou compartilhar os sons auscultados. O estetoscópio eletrônico é especialmente útil em ambientes ruidosos ou para profissionais com dificuldades auditivas. Porém apresenta alto custo em relação aos mecânicos.

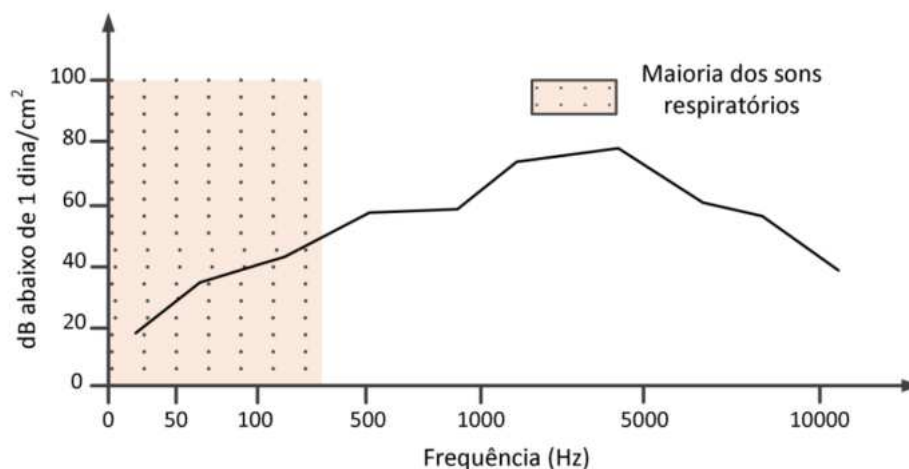
2.4 Limitações da Ausculta pulmonar

Segundo [Naves \(2015\)](#), a maior parte dos sons respiratórios está em uma faixa de frequência que escapa à detecção pelo ouvido humano, uma vez que o ouvido médio de uma pessoa jovem pode perceber vibrações sonoras com frequências entre 20 e 20000 Hz, apresentando variações significativas na sensibilidade nessa faixa. A Figura 5 ilustra a sensibilidade limiar (medida em decibéis) do ouvido humano. De acordo com [Naves \(2015\)](#) e [Lehrer \(2002\)](#), dado que a maioria dos sons respiratórios está abaixo de 500 Hz, o ouvido humano demonstra pouca sensibilidade a esses sons, resultando na sua não detecção.

Assim, a capacidade auditiva humana não é capaz de abranger plenamente o fenômeno sonoro, uma vez que a percepção de um som não é exclusivamente determinada por sua frequência, mas também pela intensidade com que é gerado, visto que sons mais graves necessitam de uma intensidade maior para se tornarem audíveis [Naves \(2015\)](#). O estetoscópio tornou-se uma ferramenta essencial na prática médica, sendo frequentemente o primeiro método utilizado para a avaliação de doenças respiratórias em consultórios médicos ([LEHRER, 2002](#)).

Embora a ausculta seja um exame rápido, de baixo custo financeiro e considerado

Figura 5 – Sensibilidade limiar (medida em decibéis) do ouvido humano.



Fonte: Naves (2015, p. 25).

não invasivo, possui diversas limitações. Uma dessas limitações é a imprecisão, uma vez que depende da habilidade auditiva do profissional que a utiliza. O avanço tecnológico tem contribuído para mitigar essas limitações, possibilitando análises mais precisas e digitais desses sons.

2.5 Classificação dos sons respiratórios

Conforme apontado por Hirata (2004), em 1819, Laënnec publicou um tratado sobre os sons pulmonares auscultados, correlacionando diagnósticos com observações de necropsias para estabelecer relações entre enfermidades e sons auscultados. No entanto, a terminologia desses termos varia ao ser traduzida para diferentes idiomas, levando à necessidade de padronização. A ISLA (*International Society for Lung Sounds*) desempenhou um importante papel ao padronizar os sons respiratórios adventícios em duas categorias principais: contínuos e descontínuos. Os sons adventícios contínuos incluem sibilos e roncosp, enquanto os descontínuos são subdivididos em estertores finos e grossos. Essa padronização proporcionou uma base consistente para a compreensão e comunicação dos sons respiratórios em contextos clínicos e acadêmicos.

Quanto aos sons respiratórios normais, é possível discernir três tipos distintos que podem ser auscultados no tórax: murmúrio vesicular, respiração brônquica e respiração broncovesicular (VISICH, 1981). A seguir, os conceitos referentes a esses diferentes sons respiratórios normais, serão apresentados.

2.5.1 Normais

Riella (2008) menciona que em indivíduos sem problemas respiratórios, as frequências audíveis resultam das vibrações mecânicas no aparelho respiratório e do fluxo gasoso que percorre seus órgãos. Essas vibrações, denominadas sons respiratórios normais, variam conforme os pontos de origem no sistema respiratório e os locais de ausculta. Os sons respiratórios normais incluem murmúrio vesicular, respiração brônquica e respiração broncovesicular.

1. Murmúrio Vesicular: Observa-se a presença de murmúrio vesicular em praticamente toda a região do tórax. Esse murmúrio é caracterizado por ser suave, assemelhando-se a um suspiro, e apresenta frequência baixa.
2. Respiração Brônquica: Identifica-se a respiração brônquica ao auscultar a face anterior do tórax, nas proximidades do esterno. Essa forma de respiração assemelha-se ao murmúrio vesicular, distinguindo-se principalmente pela intensidade menor do componente expiratório.
3. Respiração Broncovesicular: A respiração broncovesicular é percebida, em condições normais, na região esternal superior. No entanto, sua detecção em outras áreas sugere a presença de condensação pulmonar. É importante destacar que nesse tipo de respiração, combinam-se características da respiração brônquica com o murmúrio vesicular

2.5.2 Adventícios

Frequentemente, a ausculta de sons respiratórios adventícios está associada à existência de irregularidades no sistema respiratório. Tais anormalidades geram sons distintos, os quais são passíveis de identificação durante a aplicação da técnica de ausculta.

Os sons adventícios descontínuos são identificados por sua tonalidade, amplitude e duração, sendo denominados estertores. Esses ruídos exibem formas de onda que indicam um impulso breve seguido por amortecimento. Conforme Riella (2008), a distribuição de frequências durante esse impulso reflete as propriedades do som auscultado, resultando na classificação de estertores finos e estertores grossos. Ambos podem ser percebidos tanto na fase inspiratória quanto na expiratória.

Os estertores finos são caracterizados por uma frequência relativamente alta, baixa amplitude e duração mais curta, sendo audíveis apenas na segunda metade da inspiração. Sua presença não é afetada pela tosse, e sua modificação ocorre principalmente devido à posição e à influência da gravidade. Por outro lado, os estertores grossos apresentam frequência mais baixa, maior amplitude e duração comparativamente mais longa em relação

aos estertores finos. São predominantemente ouvidos no início da inspiração e na maioria das expirações, sendo suscetíveis a modificações induzidas pela tosse.

Os sons adventícios contínuos manifestam-se por ruídos que persistem de forma constante. Eles podem se apresentar com intensidade significativa e surgir tanto durante a inspiração quanto na expiração. Além disso, podem ser localizados ou disseminados, ocorrendo em quantidades variadas. Esses sons contínuos são categorizados em sibilos e roncos.

Os roncos são identificados por sons profundos e ruidosos, com uma frequência predominante de 200 Hz ou menos, assemelhando-se ao ruído associado ao ronco. Esses sons surgem quando as vias respiratórias mais amplas estão obstruídas por líquidos e podem ser percebidos tanto na inspiração quanto na expiração. Por outro lado, os sibilos são os sons mais frequentes nos registros pulmonares, caracterizados por uma tonalidade aguda, assemelhando-se a um assobio ou chiado. Um resumo de todos os sons pulmonares adventícios mencionados está disponível na Tabela 1.

Tabela 1 – Nomenclatura proposta pela ISLA para os diferentes tipos de sons adventícios.

Tipo	Subdivisão	Descrição
Descontínuo	Estertor Fino	Alta tonalidade, baixa amplitude e curta duração
Descontínuo	Estertor Grosso	Baixa tonalidade, alta amplitude e longa duração
Contínuo	Sibilo	Alta tonalidade
Contínuo	Ronco	Baixa tonalidade

Fonte: Adaptado por [Naves \(2015, p.26\)](#)

2.6 Telemedicina

Um dos principais objetivos dos profissionais de saúde é assegurar um acompanhamento constante dos pacientes, buscando otimizar o monitoramento e os cuidados prestados. Nesse contexto, observa-se que empresas de saúde, hospitais e clínicas estão empenhados em encontrar métodos que facilitem a detecção de anormalidades em monitoramentos remotos. A telemedicina, definida como uma tecnologia de informação e comunicação, desempenha um papel fundamental nesse contexto, especialmente quando a distância é um fator crítico ([BAKER; STANLEY, 2018](#)).

Atualmente, é evidente que o acesso, a equidade, a qualidade e os custos representam quatro das principais questões enfrentadas pelos sistemas de saúde universais, em meio ao crescimento populacional e às transformações nas características de saúde e doença ([WALLER; STOTLER, 2018](#)). Diante desse cenário, [Taylor, Capling e Portnoy \(2018\)](#) destaca a relevância da telemedicina, pois ela contribui para a redução do tempo de diagnóstico e tratamento de problemas de saúde, aumentando assim a eficácia dos serviços médicos. Isso justifica o investimento nas tecnologias necessárias para sua implementação.

A progressão acelerada no uso de tecnologias traz consigo uma série de soluções inovadoras para atender às necessidades humanas. A interconexão entre o mundo físico e virtual dessas novas tecnologias é atualmente referida como *Internet* das Coisas (IoT, do inglês *Internet of Things*) (ALBAHRI et al., 2021). As aplicações da IoT na área da saúde têm o potencial de proporcionar cuidados abrangentes aos pacientes, abrangendo desde tratamentos emergenciais hospitalares até cuidados de longo prazo e comunitários (OLIVEIRA; SILVA, 2017). Nesse contexto, Zhong, Xu e Cao (2023) destacam a aplicação da IoT na medicina, particularmente na medicina preventiva, promovendo uma melhor qualidade de vida e bem-estar aos pacientes com custos mais acessíveis para medicamentos, consultas e outros procedimentos médicos.

2.7 Sensores

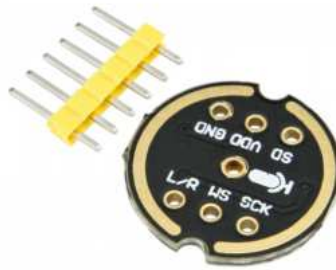
Os sensores são dispositivos eletrônicos projetados para identificar estímulos, como movimento, som, luminosidade, entre outros, e responder a esses estímulos com outra saída. A função primordial de um sensor consiste em detectar um estímulo e transmiti-lo, mediante programação adequada e conexão a um microcontrolador.

Os sensores podem ser categorizados como digitais ou analógicos. Conforme Martinazzo e Orlando (2016) explicam, os sensores digitais operam com níveis de tensão específicos, identificados como estados *LOW* e *HIGH*, ou alternativamente como "0" e "1". Por outro lado, os sensores analógicos têm a capacidade de gerar um sinal contínuo que representa variações de tensão, capacitância e corrente.

Os sensores de som são dispositivos eletrônicos projetados para mensurar a intensidade sonora no ambiente por meio de um transdutor acústico. Um exemplo representativo é o módulo INMP441, ilustrado na Figura 6. Esse dispositivo é um transdutor acústico omnidirecional recomendado para plataformas com barramento I2S, o qual separa os sinais de *clock* e serial. O microfone incorpora um conversor analógico para digital e um filtro *antialiasing*, um filtro de passa-baixa que seleciona as frequências desejadas, eliminando as indesejáveis. Entre as suas especificações notáveis, podem ser mencionadas:

- Modelo: INMP441;
- Adequação: placas com barramento I2S;
- Voltagem de Operação: 1,8V a 3,3V VCC;
- Amplitude de Frequência: 60 Hz a 15 kHz;
- Sensibilidade: -26 dBFS.

Figura 6 – Módulo INMP441.

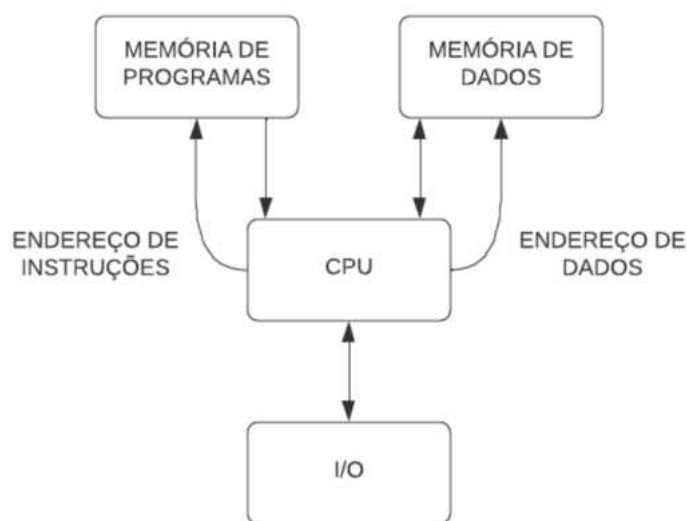


Fonte: Leon, Amaral e Júnio (2022, p.6).

2.8 Microcontroladores

Conforme mencionado por MiyamotoYanagiya et al. (2015), um microcontrolador é um circuito integrado programável que incorpora toda a estrutura, incluindo a arquitetura de um microcomputador. Em outras palavras, o microcontrolador compreende componentes como processador (Unidade Lógica e Aritmética - ULA), memória para armazenamento de programas, interfaces de entrada e saída, conversores analógico-digitais, Unidade Central de Processamento (CPU) e outros elementos. Em termos gerais, os microcontroladores são caracterizados como a integração de *hardware* e *software* em um único chip. A Figura 7 exemplifica os blocos internos presentes no microcontrolador.

Figura 7 – Arquitetura Harvard de um microcontrolador.



Fonte: Valdés-Pérez e Pallàs-Areny (2007, p.23)

- Unidade Central de Processamento (CPU), responsável por interpretar as instruções

do programa.

- Memória Programável Somente Leitura (PROM), destinada a armazenar os dados do programa.
- Memória de Acesso Aleatório (RAM), empregada para armazenar as variáveis utilizadas pelo programa.
- Componentes de Entrada/Saída (I/O) para comunicação com dispositivos externos ou recebimento de dados.

De acordo com [Batista e Garcia \(2015\)](#), os microcontroladores operam por meio do desempenho de cada componente integrado. Cada elemento desempenha uma função específica da seguinte maneira:

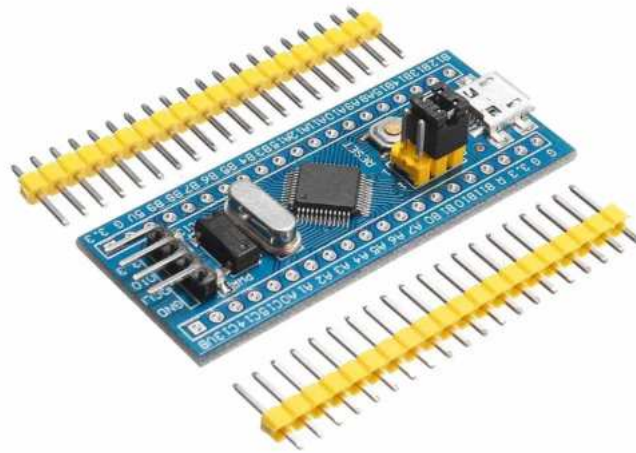
- A Unidade Lógica e Aritmética (ULA) executa cálculos e comparações envolvendo registradores e/ou lógicas para tomada de decisões, sendo o centro do microprocessador.
- A memória, dividida em memória de programa, memória de dados, contador de programa, entre outros, é responsável pelo armazenamento de dados.
- Linhas de I/O são encarregadas das entradas e saídas do microcontrolador.

No que se refere à organização, a arquitetura de Harvard se destaca por possuir duas memórias distintas e independentes, em termos de barramento e ligação ao processador. Essa arquitetura se caracteriza pelo acesso separado à memória de dados em relação à memória do programa. Os microcontroladores que adotam a arquitetura de Harvard são versáteis, sendo aplicáveis em diversas situações, inclusive em projetos de baixo custo. A utilização dessa arquitetura em microcontroladores é essencial para otimizar o desempenho com baixa frequência de *clock*. Devido a suas características, é possível construir microcontroladores de maneira econômica, proporcionando um desempenho satisfatório para dispositivos embarcados ([BATISTA; GARCIA, 2015](#)).

2.8.1 STM32

O STM32, Figura 8, é um microcontrolador flash de 32 bits baseado no núcleo ARM® Cortex®-M, oferecendo aos desenvolvedores de MCUs um ambiente de desenvolvimento aberto e flexível, com uma ampla gama de ferramentas de *hardware* e *software* para facilitar o início dos projetos. Esse microcontrolador integra características de alto desempenho, processamento em tempo real, recursos de sinal digital, baixo consumo de energia e operação em baixa voltagem, mantendo alto nível de integração e facilidade de desenvolvimento ([DENG et al., 2020](#)).

Figura 8 – Módulo de desenvolvimento Cortex M3 com microcontrolador STM32.



Fonte: [Maker Portal](#) (2019).

A série STM32 com interconexão aprimorada também oferece desempenho avançado para aplicações de áudio, incorporando um mecanismo de *loop* de fase bloqueada (PLL) que possibilita comunicação I2S com qualidade de nível de áudio. Suas principais características incluem:

- Microcontrolador: STM32;
- Núcleo: ARM 32 Cortex-M3;
- Velocidade Máxima da CPU: 72 MHz;
- Memória RAM: 20 kB;
- Memória Flash: 64 kB;
- Faixa de Tensão Operacional: 1.62V a 3.63V;
- Comunicação: SPI, I2C, UART, CAN
- Entradas ADC: 10;
- Pinos de Entrada/Saída: 44.

2.8.2 ESP32

Segundo [Kolban \(2018\)](#), o ESP32, visto na Figura 9 foi desenvolvido pela *Espressif Systems* e entrou no mercado em 2016. Este módulo destaca-se pela sua elevada velocidade de processamento de dados e conectividade *Wi-Fi*, sendo atualmente um dos módulos mais populares, inclusive devido à sua capacidade de armazenamento superior a outras plataformas de desenvolvimento com microcontrolador integrado, como o Arduíno, por exemplo ([IBRAHIM, 2019](#)).

Figura 9 – ESP32-WROOM-32.



Fonte: [Savarati \(2021\)](#).

O ESP32 é amplamente empregado em projetos IoT devido às diversas bibliotecas disponíveis para uma variedade de sensores, sendo geralmente programado na linguagem C++. O microcontrolador incorporado oferece um desempenho eficiente, além de contar com um amplificador de baixo ruído. Abaixo, estão listadas algumas das principais características do modelo ESP32-WROOM-32:

- Microcontrolador: ESP32;
- CPU: Xtensa Dual-Core 32-bit LX6;
- ROM: 448 KBytes;
- RAM: 520 KBytes;
- *Flash*: 4 MB;
- *Clock*: 240 MHz;
- *Bluetooth*: BLE 4.2;
- Padrão *Wireless*: 802.11 b/g/n;
- 18 Pinos ADC;

- 3 Pinos de interfaces SPI;
- 3 Pinos de interfaces UART;
- 2 Pinos de interfaces I2C;
- 2 Pinos de interfaces I2S.

Dessa forma, o módulo ESP32 oferece uma capacidade superior de SRAM, permitindo uma maior corrente elétrica no circuito. Além disso, destaca-se por um *clock* mais elevado em comparação a outros microcontroladores e integração de comunicação *Bluetooth*. No entanto, é importante notar que nem todas as bibliotecas desenvolvidas são compatíveis com esse microcontrolador.

2.9 Processamento de sinal

A filtragem digital é essencial no processamento de sinais, principalmente para a análise de sons respiratórios captados por sensores como o INMP441. O filtro passa-banda se destaca como uma técnica eficaz para isolar a faixa de frequência de interesse, permitindo que componentes acústicos relevantes sejam preservados enquanto ruídos fora dessa faixa são atenuados. Para sons respiratórios, a faixa entre 100 Hz e 500 Hz concentra as informações mais úteis para análise clínica, como padrões de respiração e detecção de anomalias, enquanto frequências abaixo e acima desse intervalo geralmente representam ruídos indesejados, como batimentos cardíacos e interferências ambientais (SOUSA, 2004).

A implementação de filtros passa-banda em sistemas embarcados, como o STM32, é uma abordagem prática e eficiente. Esses microcontroladores permitem que a filtragem seja realizada em tempo real, sem depender de processamento externo. Além de melhorar a relação sinal-ruído, o filtro passa-banda permite que o sistema se concentre exclusivamente nas frequências de interesse, otimizando o uso de recursos computacionais e garantindo maior precisão nos resultados (COLLIMATOR, 2023). No caso do INMP441, a filtragem passa-banda é crucial para transformar os sinais brutos captados em dados clínicos relevantes, facilitando o monitoramento e a análise contínua de condições respiratórias.

O MATLAB é amplamente utilizado para a análises de sinais, devido à sua robustez e flexibilidade na manipulação de dados. Nesse contexto de sinais pulmonares, o *software* oferece um conjunto de ferramentas avançadas para projetar e aplicar filtros digitais, como o filtro *Butterworth*, que é ideal para isolar frequências específicas enquanto preserva a integridade do sinal (BRITO; OLIVEIRA, 2017). Com comandos como *butter* para calcular os coeficientes do filtro e *filtfilt* para aplicar a filtragem sem introduzir atraso de fase, o MATLAB permite a remoção de ruídos indesejados, como interferências ambientais e eletrônicas. Além disso, sua capacidade de realizar a Transformada Rápida de Fourier

(FFT) possibilita a visualização e análise do espectro de frequência antes e após a aplicação do filtro, garantindo que as frequências de interesse sejam preservadas ([The MathWorks, Inc, 2015](#)).

2.10 Transformada de Fourier

Cada sinal periódico tem a capacidade de ser expresso como a soma das séries de Fourier. Realizando uma análise apropriada, é viável obter uma representação de Fourier para sinais de duração finita. Essa representação é conhecida como Transformada Discreta de Fourier (TFD) ([SCHMIDT, 2013](#)). A TFD pode ser descrita da seguinte maneira:

$$x[k] = \sum_{n=0}^{N-1} x[n] \cdot W_{kn}^N \quad k = 0, 1, \dots, N - 1 \quad (2.1)$$

$$W^N = e^{-j\frac{2\pi}{N}} \quad (2.2)$$

Conforme apontado por [Schmidt \(2013\)](#), ao examinar a resolução direta, nota-se que implica em N multiplicações complexas e $N - 1$ adições complexas para cada k . Logo, o cálculo direto de uma Transformada Discreta de Fourier (TFD) é da ordem $O(N^2)$. Em situações em que N é pequeno, a resolução em si não consome considerável tempo ou recursos. No entanto, para valores significativamente grandes de N , o cálculo direto torna-se ineficiente. Isso ocorre não apenas devido ao tempo prolongado necessário, mas também pela acumulação de recursos essenciais. Por exemplo, considerando que cada operação leva aproximadamente 1ns, o cálculo direto da TFD para N grande levaria cerca de 13.343 dias.

Observa-se, portanto, que a ordem do cálculo direto estabelece uma limitação para aplicações que fazem uso da Transformada Discreta de Fourier (TFD), especialmente em tempo real. Isso ocorre porque, para valores de N acima de um determinado limiar, o cálculo pode tornar-se excessivamente lento, e os recursos disponíveis podem se revelar insuficientes. É nesse contexto que surge a Transformada Rápida de Fourier (FFT), um algoritmo concebido para o eficiente cálculo da TFD. Sua relevância reside na capacidade de eliminar grande parte dos cálculos repetitivos aos quais a TFD está sujeita, resultando em um cálculo mais rápido a um custo menor ([SCHMIDT, 2013](#)).

De acordo com [Schmidt \(2013\)](#), ao realizar inicialmente uma Transformada Discreta de Fourier (TFD) com N amostras, o processo pode ser conduzido por $p = \log_2 N$ divisões. Ao calcularmos o custo das operações envolvidas nesse método, constatamos que o algoritmo possui complexidade $O(N \cdot \log_2 N)$.

O algoritmo FFT foi concebido originalmente por Carl Friedrich Gauss em 1805. Apesar de diferentes versões do algoritmo terem sido descobertas ao longo dos anos, a FFT só ganhou popularidade a partir de 1965, com a publicação de James Cooley e John Tukey. Foi por meio deles que o algoritmo foi reinventado e eficientemente descrito para execução em computadores (SCHMIDT, 2013).

2.11 Tecnologia de desenvolvimento

A crescente tendência dos *frameworks JavaScript* na comunidade de desenvolvimento gerou o surgimento de múltiplas opções destinadas a auxiliar os programadores em suas tarefas. No entanto, o número desses *frameworks* cresceu exponencialmente em comparação com uma década atrás. Isso torna mais desafiador para os profissionais identificar quais deles realmente podem agregar valor às suas necessidades (SOUSA; GONCALVES, 2020).

2.11.1 *React.js*

O *React* (também conhecido como *React.js* ou *ReactJS*) é uma biblioteca de código aberto em *JavaScript*, focada na criação de interfaces de usuário (*frontend*) para páginas da *web*. Ele é mantido pelo *Facebook*, *Instagram*, outras empresas, e por uma comunidade de desenvolvedores individuais. Em 2015, o *Facebook* introduziu o módulo *React Native*, que em conjunto com o *React.js*, permite o desenvolvimento de aplicativos para *Android* e *iOS* utilizando componentes de interface nativos de ambas as plataformas, sem a necessidade de recorrer ao HTML (REACT.DEV, 2023).

2.11.1.1 Características

As principais características do *React.js* incluem:

- Simplicidade: Sua abordagem baseada em componentes, ciclo de vida bem definido e uso simplificado de *JavaScript* tornam o *React.js* fácil e prático de usar.
- Fácil aprendizado: Requer apenas conhecimentos básicos de HTML (*HyperText Markup Language*) e CSS (*Cascading Style Sheets*) para compreender e aprender o *React.js* facilmente. O HTML é a linguagem de marcação usada para estruturar o conteúdo de uma página *web*, definindo elementos como títulos, parágrafos, listas, imagens e *links*. Já o CSS é a linguagem responsável por definir o estilo e a aparência visual desses elementos, controlando aspectos como cores, fontes, tamanhos e posicionamento. Com uma base simples de HTML e CSS, é possível entender os fundamentos do *React.js* e criar interfaces interativas.

- Abordagem nativa: O *React.js* pode ser usado no desenvolvimento de aplicativos móveis (com *React Native*) e suporta a reutilização de código, permitindo a implementação de aplicativos para *iOS*, *Android* e *web* simultaneamente.
- *Data Binding*: Utiliza ligação de dados unidirecional e uma arquitetura de aplicativo chamada Flux, que controla o fluxo de dados para os componentes por meio de um ponto de controle.
- Testabilidade: As aplicações desenvolvidas em *React.js* são mais fáceis de serem testadas.

2.11.2 *Firebase*

O *Firebase* é uma plataforma de aplicativos *web* que oferece uma variedade de serviços para desenvolvedores. Isso inclui armazenamento de dados em formato JSON, análise de aplicativos, mensagens em nuvem, autenticação de usuários, banco de dados em tempo real, armazenamento de arquivos, teste de aplicativos para *Android*, relatórios de falhas e notificações direcionadas. Esses serviços permitem aos desenvolvedores criar aplicativos de alta qualidade, analisar o uso do aplicativo, se comunicar com os usuários, autenticar usuários de forma segura, armazenar e sincronizar dados em tempo real, armazenar e gerenciar arquivos, testar aplicativos em vários dispositivos e receber relatórios detalhados sobre falhas (KHAWAS; SHAH, 2018).

2.11.3 *Figma*

O *Figma* é uma ferramenta de design colaborativa baseada na nuvem, amplamente utilizada para a criação de interfaces de usuário e experiências de usuário. Reconhecido por sua acessibilidade e facilidade de uso, o *Figma* permite o desenvolvimento de projetos de forma integrada e em tempo real. Além de oferecer funcionalidades robustas para prototipagem, criação de *layouts* e design responsivo, o *Figma* opera diretamente em navegadores da *web* e é compatível com múltiplos sistemas operacionais, como *Windows*, *Mac* e *Linux*, sem a necessidade de instalação complexa. Sua abordagem baseada na nuvem garante o armazenamento seguro de arquivos, sincronização automática e acesso remoto, destacando-se como uma solução eficiente e moderna para o desenvolvimento de aplicativos e projetos digitais (SANTOSO, 2024).

3 Materias e Métodos

O objetivo deste trabalho é o desenvolvimento de um sistema de baixo custo, combinando *hardware* e *software*, para realizar a ausculta pulmonar e permitir o monitoramento remoto de pessoas que necessitam de acompanhamento. O sistema a ser desenvolvido captura os sons pulmonares com o objetivo de auxiliar futuramente profissionais de saúde no acompanhamento de pacientes. O fluxo geral do desenvolvimento pode ser visto na Figura 10.

Figura 10 – Fluxograma de desenvolvimento do trabalho.



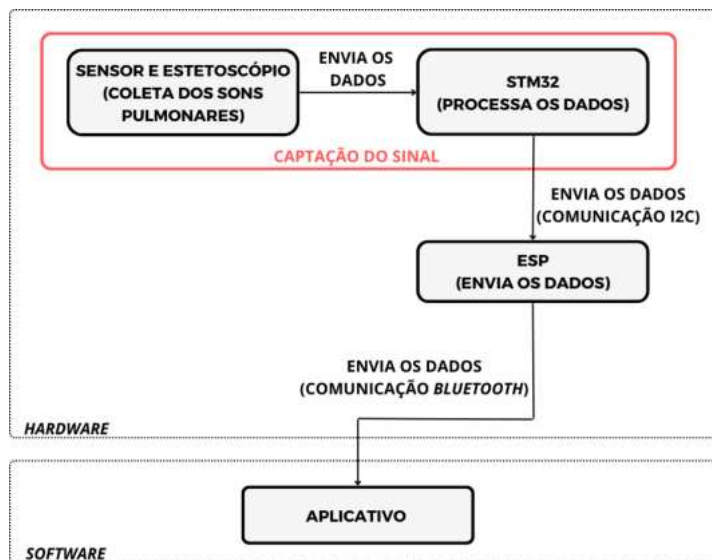
Fonte: Autor.

3.1 Hardware

Para desenvolver a parte de *hardware* da proposta, o processamento foi dividido em duas etapas como mostra Figura 11. Na primeira etapa, foi construída uma configuração de nó sensor com o módulo de desenvolvimento STM32 ARM Cortex M3, que processa os dados captados pelo sensor de microfone digital INMP441. Na segunda etapa, o STM32 é conectado ao módulo ESP32, utilizando comunicação I2C para enviar os dados processados

A finalidade é captar sinais de áudio do ambiente com o microfone digital INMP441, processá-los inicialmente com o microcontrolador STM32 ARM Cortex M3 e, em seguida,

Figura 11 – Arquitetura do sistema.



Fonte: Autor.

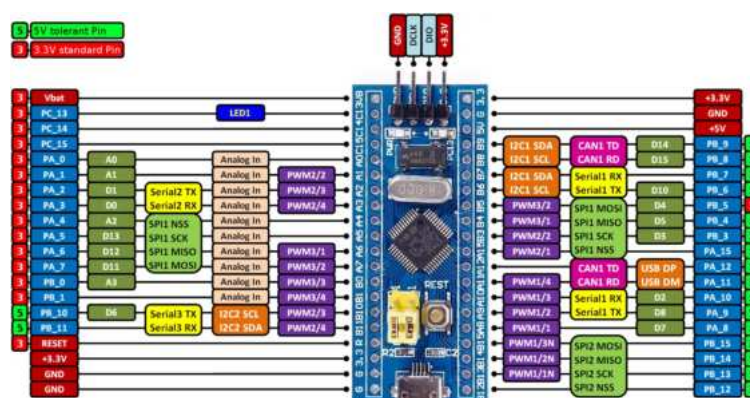
transmitir os dados ao ESP32 WROOM, que faz a comunicação com o aplicativo via *Bluetooth*.

3.1.1 Montagem e integração dos dispositivos

3.1.1.1 Conexão do Sensor de Microfone INMP441 ao STM32

Primeiro, conectamos o INMP441 ao STM32 que recebe os dados digitais do microfone pelo protocolo I2S, possibilitando o processamento inicial dos dados de áudio. A Figura 12 apresenta a arquitetura e dos pinos do STM32.

Figura 12 – Arquitetura módulo STM32-WROOM.

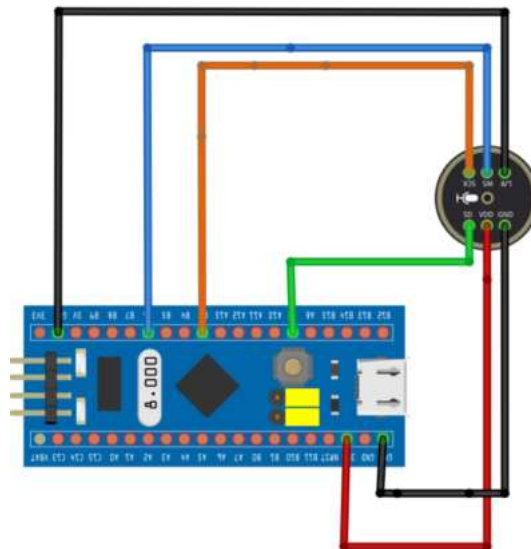


Fonte: Maker Portal (2019).

- VCC do INMP441 é conectado ao 3,3V do STM32, fornecendo a alimentação necessária para o microfone (ligação vermelha).
- GND do INMP441 é conectado ao GND do STM32, estabelecendo uma referência de terra comum (ligação preta).
- Conexões I2S:
 - SD (*Data Out*) do INMP441, que transmite o áudio digital, é conectado ao GPIO (General Purpose Input/Output) do STM32 configurado para entrada de dados I2S (ligação verde).
 - SCK (*Bit Clock*) do INMP441 é conectado ao GPIO do STM32 configurado para clock I2S (ligação laranja).
 - WS (*Word Select*) do INMP441, que define o canal de áudio, é ligado a um GPIO do STM32 configurado para seleção de canal (ligação azul).

A Figura 13, apresenta o esquemático das conexões

Figura 13 – Conexão entre o sensor de som INMP441 e o módulo de desenvolvimento STM32.



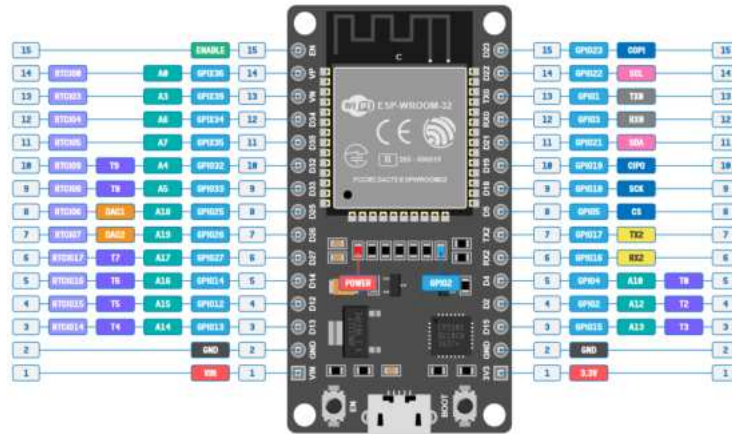
Fonte: Autor.

3.1.1.2 Conexão entre o STM32 e o ESP32

O STM32 envia os dados para o ESP32 utilizando comunicação I2C, permitindo uma troca eficiente e confiável entre os dois dispositivos. Essa conexão possibilita que o ESP32 receba os dados de áudio processados pelo STM32 e os transmita ao aplicativo

de monitoramento via *Bluetooth*. A Figura 14 apresenta os detalhes da arquitetura e dos pinos do ESP32. A Figura 15 mostra como fica o esquemático completo.

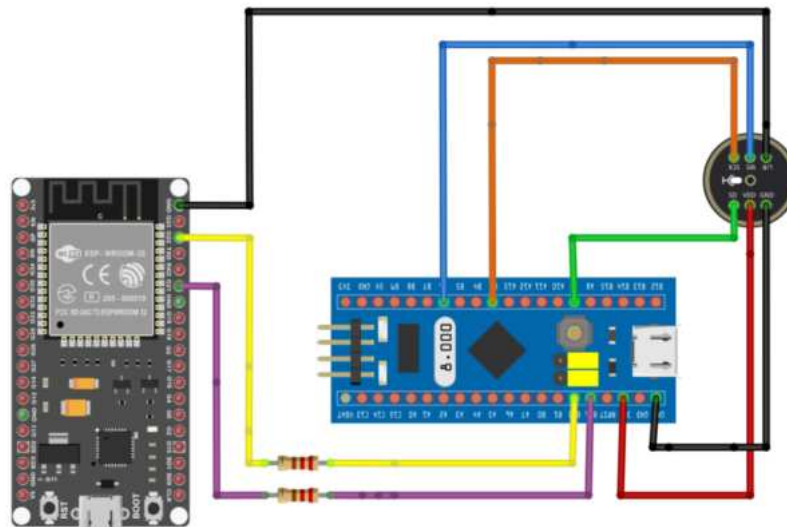
Figura 14 – Arquitetura e dos pinos do ESP32.



Fonte: Savarati (2021).

- O pino SDA do STM32 conecta-se ao SDA do ESP32 (ligação amarela), enquanto o SCL conecta-se ao SCL, garantindo a comunicação I2C e a sincronização entre os dispositivos (ligação magenta). Dois resistores de 4.7 kΩ são usados nos pinos para o correto funcionamento do barramento.

Figura 15 – Conexão entre o nó sensor e o ESP32-WROOM.

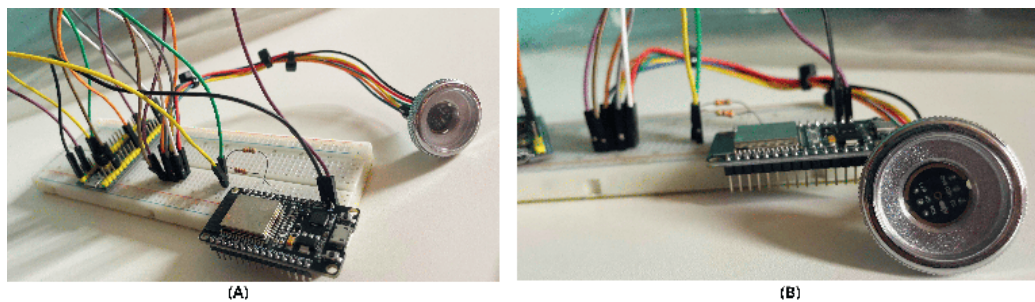


Fonte: Autor.

3.1.2 *Design* da estrutura do protótipo

Inicialmente, os componentes foram montados em uma *protoboard*, garantindo organização e facilitando ajustes iniciais no circuito. O protótipo, ilustrado na Figura 16 (A), foi construído utilizando o sensor de microfone INMP441, integrado à campânula de um estetoscópio, conforme mostrado na Figura 16 (B). Posteriormente, o circuito foi montado em uma placa perfurada para viabilizar a modelagem e impressão da estrutura final.

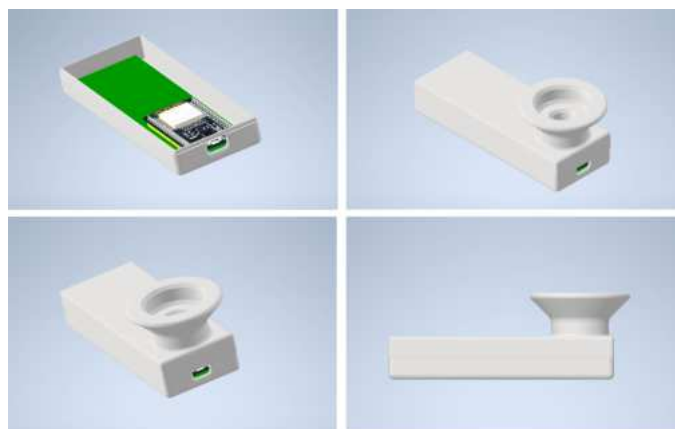
Figura 16 – Protótipo inicial.



Fonte: Autor.

A estrutura do protótipo foi desenvolvida no *software Autodesk Inventor Professional 2024*, conforme apresentado na Figura 17. O *design* foi projetado para acomodar a placa com os circuitos no interior, garantindo proteção aos componentes eletrônicos, enquanto a campânula, contendo o sensor de microfone, é posicionada na parte externa para capturar os sinais de áudio. A fabricação foi realizada por impressão 3D, proporcionando precisão e adaptabilidade às necessidades do projeto.

Figura 17 – *Design* da estrutura do protótipo.



Fonte: Autor.

3.1.3 Filtragem do sinal

No protótipo desenvolvido, a filtragem dos sinais captados pelo sensor INMP441 foi realizada diretamente no microcontrolador STM32, utilizando um filtro IIR (*Infinite Impulse Response*) *Butterworth* passa-banda de segunda ordem, como mostra a função de transferência na equação 3.1. O sensor INMP441 transmite os sinais de áudio em formato digital via I2S, permitindo que o microcontrolador STM32 processe os dados diretamente, sem necessidade de conversão adicional. O filtro foi configurado para isolar os sinais respiratórios dentro da faixa de frequência de 100 Hz a 500 Hz e com uma taxa de amostragem de 44100 Hz, que é onde estão os componentes mais relevantes para a análise dos sons pulmonares. Essa faixa foi escolhida para eliminar ruídos de baixa frequência, como batimentos cardíacos, e de alta frequência, como interferências ambientais.

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \quad (3.1)$$

- b_0, b_1, b_2 são os coeficientes de numerador.
- a_1, a_2 são os coeficientes de denominador.

A implementação do filtro IIR passa-banda foi realizada utilizando a biblioteca CMSIS-DSP da ARM, que oferece funções otimizadas para processadores ARM Cortex-M. A filtragem foi realizada diretamente sobre os dados digitais capturados em tempo real, assegurando um processamento rápido e preciso dos sinais. Os coeficientes do filtro (equação 3.1) foram calculados previamente no MATLAB, com base nas frequências de corte, como mostra a equação 3.2, e carregados diretamente no código do microcontrolador, apresentado no Apêndice .1.3. A ordem do filtro é $N = 2$, mas devido à sua natureza passa-banda, o grau do polinômio no numerador e no denominador é 4.

Coeficientes de Numerador (b):

$$b_0 = 0.0008, \quad b_1 = 0, \quad b_2 = -0.0016, \quad b_3 = 0, \quad b_4 = 0.0008$$

Coeficientes de Denominador (a):

$$a_1 = 1.0000, \quad a_2 = -3.9175, \quad a_3 = 5.7576, \quad a_4 = -3.7627$$

Assim temos:

$$H(z) = \frac{0.0008 + 0z^{-1} - 0.0016z^{-2} + 0z^{-3} + 0.0008z^{-4}}{1 + (-3.9175)z^{-1} + 5.7576z^{-2} - 3.7627z^{-3} + 0.9226z^{-4}} \quad (3.2)$$

Essa abordagem permitiu que o STM32 realizasse o processamento do sinal garantindo que apenas os dados relevantes, ou seja, os sinais pulmonares dentro da faixa de frequência escolhida, fossem transmitidos ao ESP32 para posterior envio ao aplicativo. Com isso, foi possível garantir a clareza dos sons respiratórios e um monitoramento eficaz e contínuo das condições pulmonares.

3.1.4 Validação por meio de comparação com o banco de dados

Foi realizada a validação do protótipo por meio de uma análise espectral no MATLAB. Nessa análise, foram comparados os sinais captados pelo dispositivo com os dados do banco de sons pulmonares da ICBHI (*International Conference on Biomedical and Health Informatics*)¹, amplamente utilizado para pesquisa e desenvolvimento na área de análise de sons respiratórios. Este banco de dados foi criado para fornecer uma base padronizada, contendo gravações de sons pulmonares e informações associadas, com o objetivo de apoiar o desenvolvimento de sistemas automatizados de análise e diagnóstico médico. A análise espectral foi realizada aplicando-se a Transformada de Fourier, que permitiu a conversão dos sinais do domínio do tempo para o domínio da frequência, facilitando a identificação das componentes espectrais mais relevantes de cada som pulmonar.

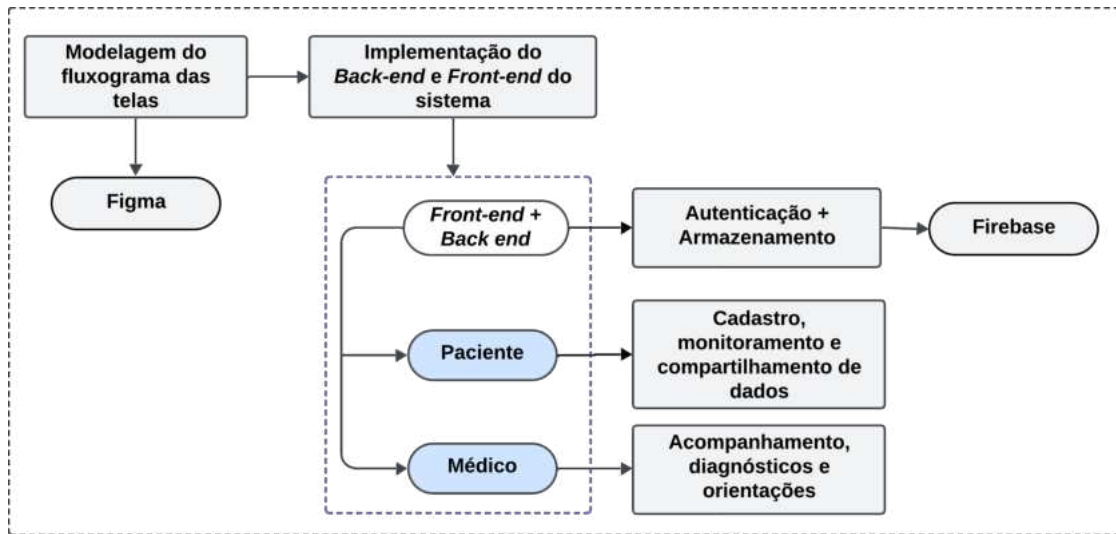
Os sinais captados pelo protótipo foram processados para realçar as frequências típicas de sons pulmonares clínicos, como sibilos e estertores, que se concentram na faixa de 100 Hz a 500 Hz. Em seguida, esses espectros foram comparados com os registros do banco de dados da ICBHI, que contém sons classificados e padronizados por especialistas, abrangendo diversas condições respiratórias. A semelhança entre os espectros foi avaliada com base na localização e intensidade das frequências dominantes, identificando padrões acústicos compatíveis com os dados de referência.

3.2 *Software*

O *software*, desenvolvido em conjunto com profissionais de saúde, registra as sessões em um banco de dados, permitindo a criação de históricos, servindo como suporte ao diagnóstico clínico. Embora facilite o monitoramento remoto, o sistema complementa, e não substitui, a atuação dos profissionais, promovendo uma abordagem colaborativa que une tecnologia e conhecimento clínico para oferecer diagnósticos mais precisos e melhor atendimento ao paciente. A Figura 18 apresenta o fluxo do desenvolvimento do aplicativo.

¹ <<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HT6PKI>>

Figura 18 – Arquitetura de desenvolvimento do aplicativo.

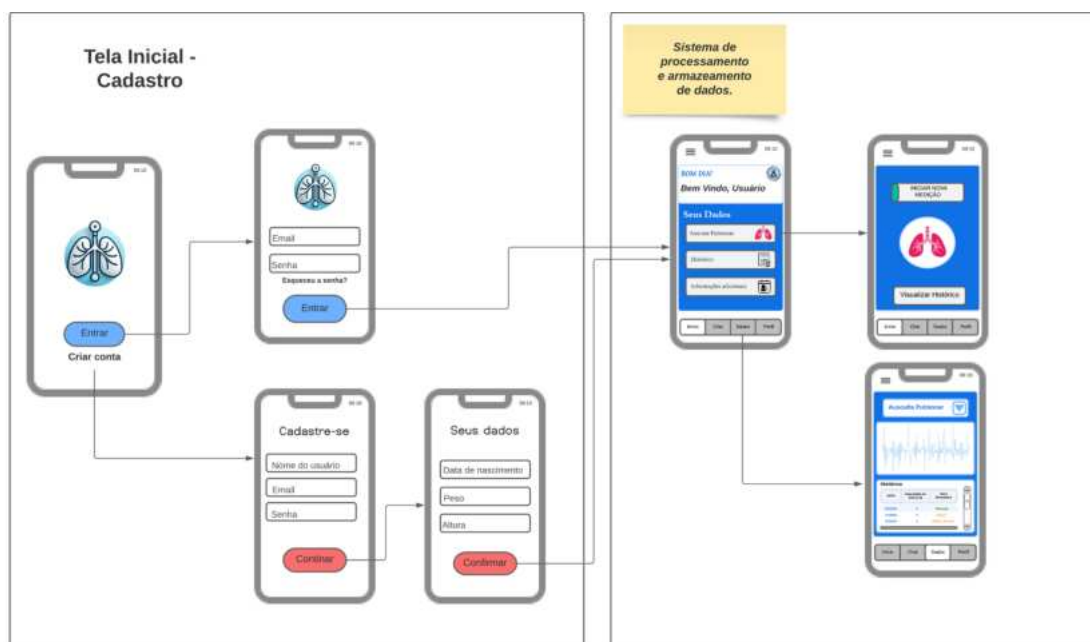


Fonte: Autor.

3.2.1 Modelagem do fluxograma das telas

Assim, foi desenvolvida uma prototipagem das telas da solução proposta com o intuito de exemplificar como se daria o desenvolvimento do *software*. Em linhas gerais, protótipos de tela são modelos funcionais que representam a interface e simulações básicas de um sistema a partir de especificações iniciais, ajudando a visualizar a aparência e o funcionamento do *software* Leon, Amaral e Júnior (2023). Esses protótipos permitem que tanto futuros usuários quanto desenvolvedores avaliem e ajustem funcionalidades da aplicação. Libera e Oliveira (2021) destaca que não é necessário que o protótipo inclua todas as funcionalidades do produto final; apenas aquelas que atendam aos objetivos do teste ou da análise em questão. A Figura 19, mostra o fluxo inicial das principais telas de acesso do usuário, antes do desenvolvimento final.

Figura 19 – Fluxograma das principais telas de acesso do usuário.

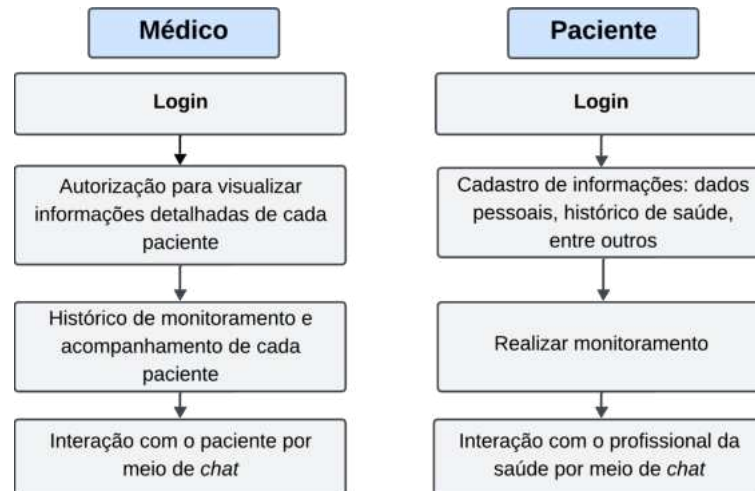


Fonte: Autor.

Para o desenvolvimento das telas, foi utilizada a ferramenta Figma, que facilita a criação de interfaces digitais de forma colaborativa e eficiente, permitindo edições em tempo real no navegador e oferecendo suporte para a criação de *layouts* para dispositivos móveis e telas *web*. Com a prototipação de telas concluída, foi possível desenvolver as telas da aplicação. O sistema em questão apresenta uma estrutura com dois níveis de acesso distintos, destinados aos usuários médico e paciente como apresentado na Figura 20.

- Médico: É essencial que médicos tenham acesso a uma plataforma digital dedicada para otimizar o gerenciamento das informações médicas e garantir diagnósticos precisos. Após criar uma conta e fazer login, eles podem visualizar detalhes completos dos pacientes, incluindo históricos e registros de medições, proporcionando uma compreensão abrangente da condição de saúde ao longo do tempo.
- Paciente: O aplicativo oferece facilidades para o gerenciamento da saúde do paciente, permitindo que usuários insiram informações pessoais e histórico médico, facilitando o acesso dos profissionais e o armazenamento seguro de medições. A comunicação direta com médicos especialistas e o acompanhamento promovem uma abordagem participativa no cuidado com a saúde.

Figura 20 – Estrutura dos níveis de acesso médico e paciente

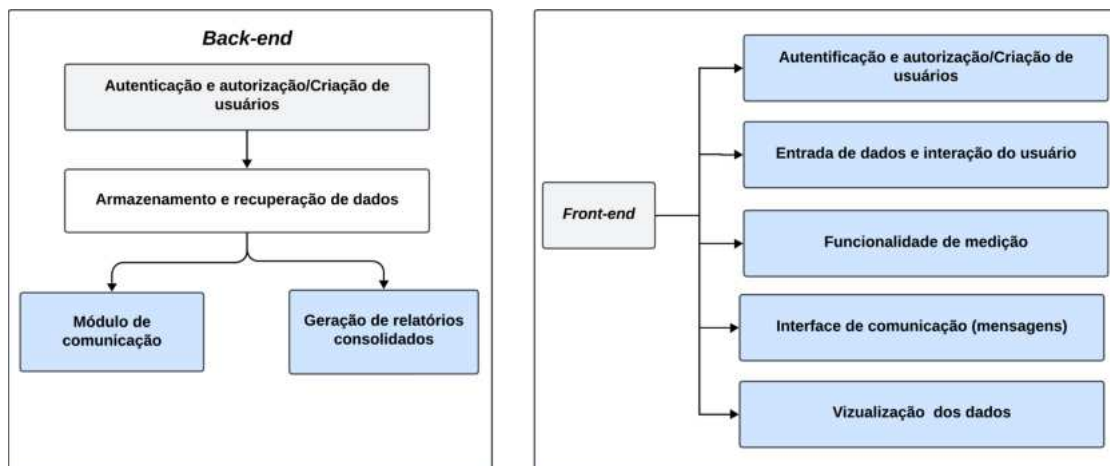


Fonte: Autor.

3.2.2 Implementação do *back-end* e *front-end* do sistema

A Figura 21 apresenta como foi organizada a estrutura de um *back-end* robusto com um *front-end* interativo e intuitivo, utilizando tecnologias modernas para garantir eficiência e segurança, apresentado no Apêndice .1.2. O *front-end* da aplicação foi desenvolvido com *React.js*, que, combinado com *JavaScript*, *TypeScript*, HTML e CSS, oferece uma interface amigável e responsiva para os usuários. Essa camada permite a autenticação de usuários, entrada de dados, interação com as funcionalidades do sistema e visualização dos dados processados. Para a comunicação com o *back-end*, o sistema utiliza o *Firebase*, que também é responsável pelo armazenamento seguro e eficiente dos dados capturados, como informações provenientes da ausculta pulmonar e registros feitos por profissionais de saúde.

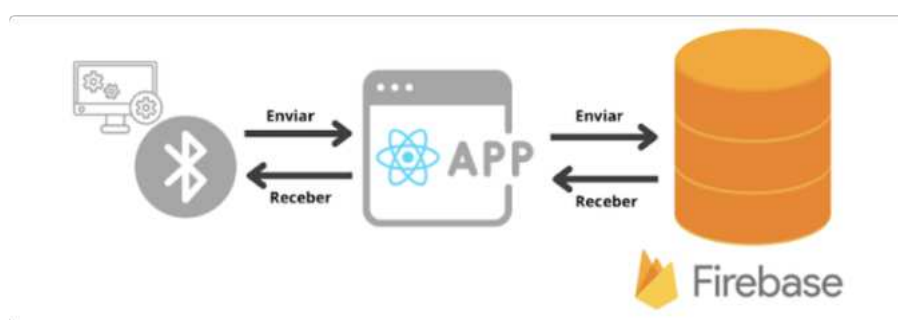
O *Firebase* não só armazena os dados, mas também gerencia a sincronização em tempo real entre o *front-end* e o *back-end*, proporcionando um ambiente robusto e simplificado para o desenvolvimento de aplicações *web* e *mobile*. Ele facilita a autenticação de usuários, oferecendo APIs prontas para gerenciar permissões e acessos. Além disso, o *Firebase* assegura a comunicação entre as camadas do sistema, permitindo que os dados armazenados sejam utilizados para gerar relatórios consolidados no *front-end*. Essa combinação de tecnologias garante um fluxo contínuo e seguro de informações, resultando em uma solução eficiente, moderna e escalável, que atende às necessidades tanto dos usuários finais quanto dos profissionais que operam o sistema.

Figura 21 – Estrutura de *back-end* e *front-end*.

Fonte: Autor.

3.3 Comunicação entre *software* e *hardware*

O protótipo utiliza um sistema de comunicação em que o *hardware* responsável pela captação dos sons pulmonares transmite os dados para um aplicativo via *Bluetooth*. Esses dados são então armazenados no *Firebase*, que serve como banco de dados central, permitindo o acesso posterior e a construção de um histórico de medições. O diagrama apresentado na Figura 22 ilustra o fluxo de informações nesse sistema, destacando os três principais componentes: o *hardware* (sensor, microcontrolador e módulo *Bluetooth*), o aplicativo e o *Firebase*.

Figura 22 – Diagrama de integração do *hardware* e *software*.

Fonte: Autor.

Esse fluxo de informações ocorre de forma bidirecional. O *hardware* envia os dados capturados para o aplicativo, que, por sua vez, encaminha essas informações ao *Firebase*. Em sentido inverso, o *Firebase* pode retornar informações relevantes para o aplicativo, que então as envia ao *hardware*, caso necessário. Essa arquitetura garante uma comunicação

eficiente entre os elementos do sistema, permitindo tanto o armazenamento seguro dos dados quanto a interação em tempo real entre os dispositivos.

3.4 Validação

Com o sistema devidamente integrado, uma série de testes foi conduzida tanto no *hardware* quanto no *software*, buscando ajustes precisos até que os resultados atingissem níveis satisfatórios. Garantida a conformidade técnica, o projeto foi submetido à Plataforma Brasil, encaminhado ao Comitê de Ética da Universidade do Estado do Amazonas - UEA - 5016.

4 Resultados

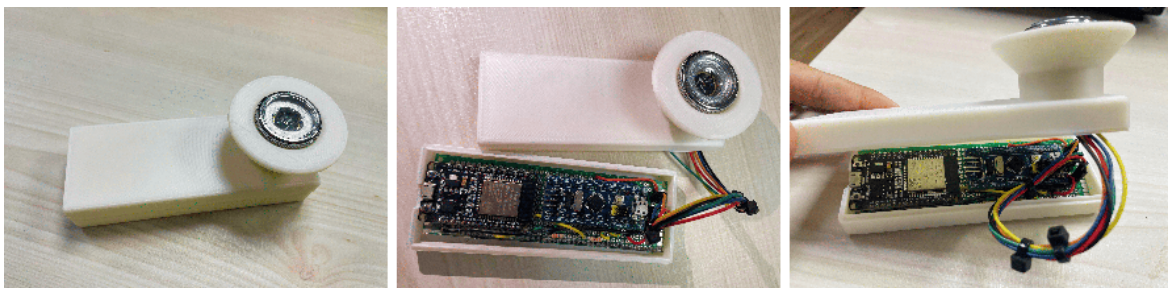
Neste capítulo, serão apresentados em detalhes os principais resultados do trabalho, incluindo o protótipo desenvolvido, responsável pela captação dos sons respiratórios, e o processo de análise aplicado para aprimorar sua qualidade. Além disso, serão exibidas as telas do aplicativo, projetadas para facilitar a interação com o sistema, demonstrando a integração entre o *hardware* e o *software* para o monitoramento eficiente dos dados coletados.

4.1 Protótipo

O protótipo foi construído utilizando o sensor de microfone INMP441, acoplado dentro da campânula de um estetoscópio, para otimizar a captação dos sons respiratórios e reduzir os ruídos externos. O sensor está conectado à placa STM32, responsável pelo processamento inicial do sinal, enquanto a ESP32 transmite os dados para o aplicativo.

Após a finalização da montagem, o protótipo impresso apresentou uma estrutura compacta e funcional, com todos os componentes devidamente encaixados e conectados como mostra a Figura 23. A impressão 3D permitiu uma montagem precisa e alinhada, garantindo que os sensores e microcontroladores estivessem posicionados corretamente para capturar os sinais pulmonares. A parte externa foi projetada para ser ergonômica, facilitando a manipulação e utilização do dispositivo durante os testes e futuras implementações. O resultado final atendeu às expectativas em termos de funcionalidade e praticidade.

Figura 23 – Protótipo final.



Fonte: Autor.

4.2 Processamento de sinais

4.2.1 Análise do sinal captado pelo sensor INMP441

Para validar a capacidade do sensor INMP441 em captar sinais respiratórios de alta qualidade, foram realizados testes iniciais utilizando o MATLAB como ferramenta de análise. Os dados captados pelo sensor foram salvos em formato de áudio, incluindo tanto registros de sons reais quanto sons pulmonares *online*¹. Esses áudios foram analisados e comparados, no MATLAB, com os dados do banco de dados ICBHI, que fornece registros em formato de áudio e texto. O objetivo dessa comparação foi avaliar a qualidade dos sinais captados e a capacidade do sistema em representar fielmente os padrões pulmonares, permitindo a identificação de diferenças entre sinais normais e anormais.

A Figura 24 apresenta o sinal pulmonar bruto captado e exibido no domínio do tempo, onde apresenta oscilações de difícil interpretação devido à presença de ruídos ambientais e interferências de baixa frequência, como batimentos cardíacos ou movimentos musculares. Esses elementos mascaram os sons pulmonares de interesse, dificultando a análise clínica e a identificação de padrões respiratórios ou anormalidades específicas. No domínio da frequência, observa-se uma concentração de energia em baixas frequências e uma redução significativa nas altas, o que reforça a necessidade de um processo de filtragem adequado.

Figura 24 – Sinais captados pelo sensor INMP441.



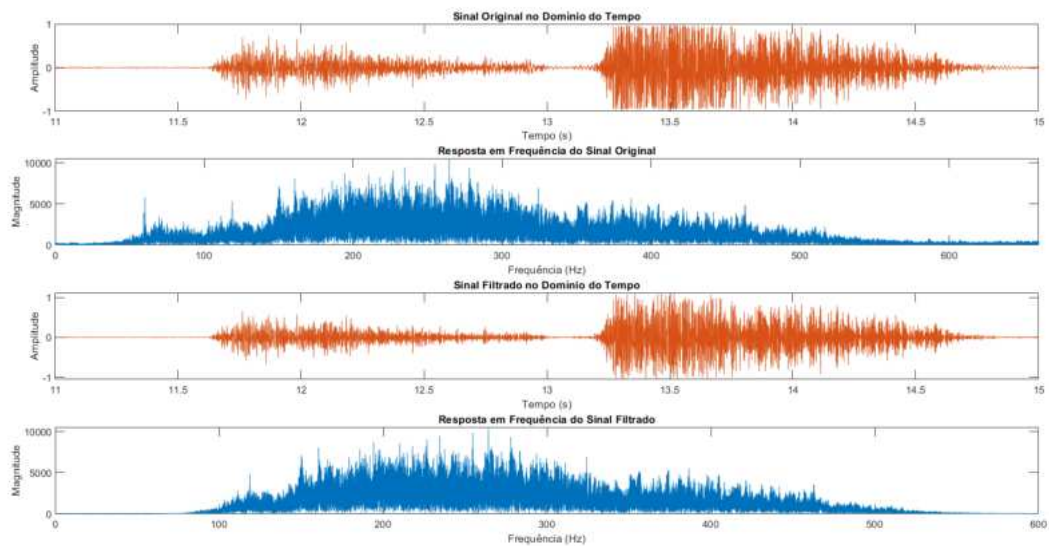
Fonte: Autor.

Com base nessa análise inicial, foi selecionada a faixa de frequência de 100 Hz a 500 Hz como a região de maior relevância clínica, considerando que nessa faixa se concentram os sons pulmonares mais significativos. Para isolar essas frequências, foi implementada uma filtragem digital utilizando o filtro passa-banda *Butterworth*, devido à sua capacidade de fornecer uma resposta suave e eficaz dentro da banda de interesse. Adicionalmente, a técnica *filtfilt* foi aplicada para garantir a filtragem sem atraso de fase, preservando a

¹ Disponível em: <<<https://www.youtube.com/c/Thinklabs1>>>

integridade temporal do sinal. Essa abordagem resultou em um sinal mais limpo e com qualidade aprimorada, adequado para análises clínicas mais precisas, conforme ilustrado na Figura 25.

Figura 25 – Sinal filtrado captado pelo sensor INMP441.



Fonte: Autor.

A frequência normalizada é uma forma de representar a frequência relativa à frequência de *Nyquist*, que é metade da taxa de amostragem do sistema. No caso, a taxa de amostragem é 44100 Hz, logo a frequência de *Nyquist* é 22050 Hz. Quando normalizamos frequências, dividimos a frequência de interesse pela frequência de *Nyquist*.

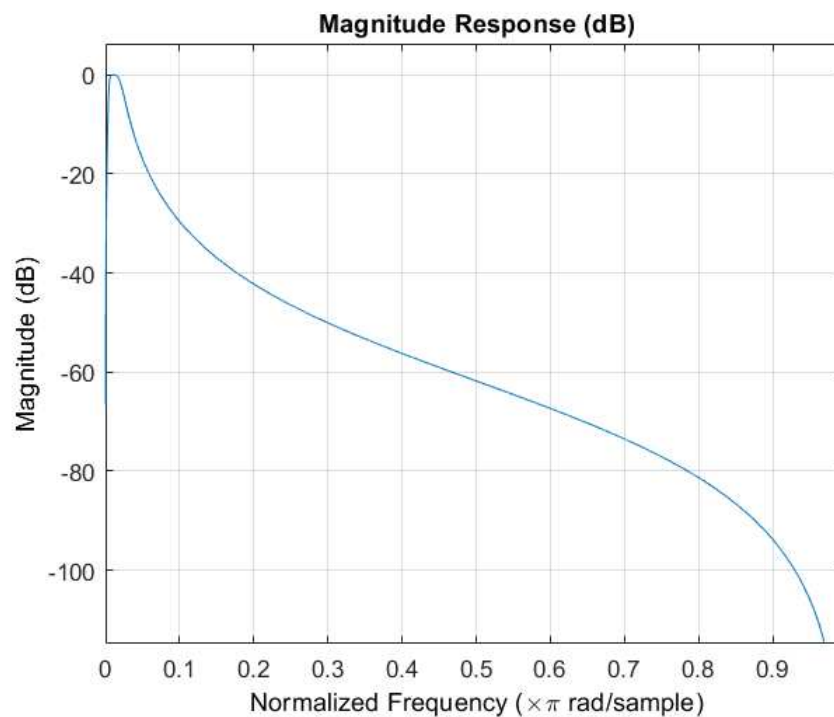
Para um filtro passa-banda com frequências de corte em 100 Hz e 500 Hz:

- 100 Hz: Quando dividido por 22050 Hz, o resultado é aproximadamente 0.0045.
- 500 Hz: Quando dividido por 22050 Hz, o resultado é aproximadamente 0.0227.

No diagrama de magnitude da Figura 26, o eixo X mostra essas frequências normalizadas, onde 0.0045 corresponde a 100 Hz e 0.0227 a 500 Hz. O eixo Y, por sua vez, mede a magnitude em decibéis (dB). Valores próximos de 0 dB indicam que a frequência passa pelo filtro sem atenuação, enquanto valores negativos indicam atenuação.

Dentro da faixa de 100 Hz a 500 Hz, a magnitude é próxima de 0 dB, o que significa que essas frequências são transmitidas quase sem alteração, preservando os sons pulmonares importantes para a análise. Fora dessa faixa, a magnitude cai significativamente, indicando que essas frequências são atenuadas, ajudando a eliminar ruídos e interferências indesejadas.

Figura 26 – Diagrama de magnitude, destacando a faixa de 100 a 500 Hz.

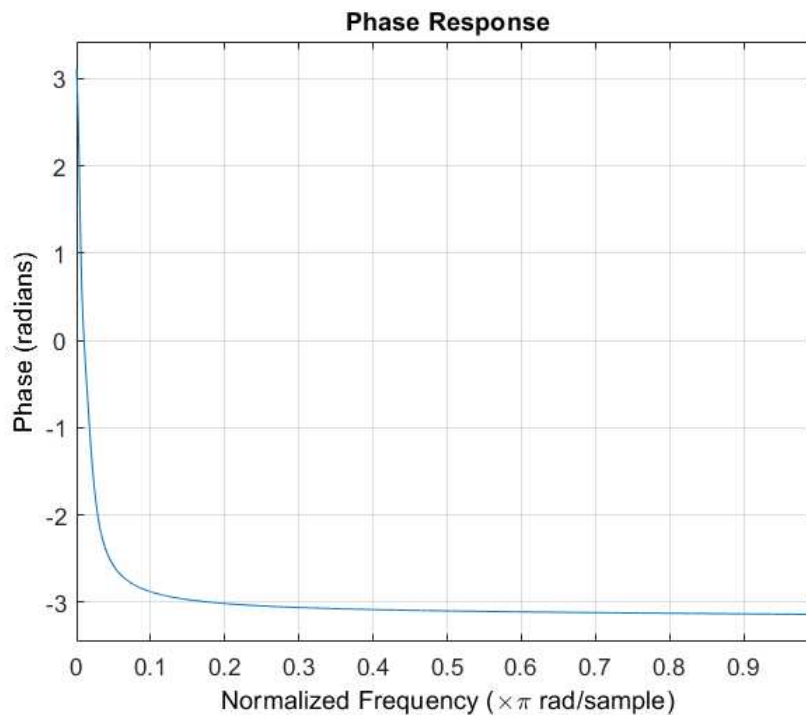


Fonte: Autor.

Este comportamento do filtro *Butterworth* passa-banda é ideal para garantir que apenas as frequências relevantes sejam mantidas.

O diagrama de fase, Figura 27, mostra uma resposta de fase linear dentro da faixa de passagem. Essa linearidade é essencial para garantir que todas as componentes de frequência do sinal sejam deslocadas uniformemente no tempo, preservando a coerência temporal e evitando distorções de forma de onda. No eixo X, as frequências são normalizadas em relação à frequência de *Nyquist* (22050 Hz), enquanto o eixo Y mostra a fase em radianos. As frequências dentro da faixa de 100 Hz a 500 Hz experimentam um deslocamento de fase relativamente constante, enquanto as frequências fora dessa faixa são severamente atenuadas.

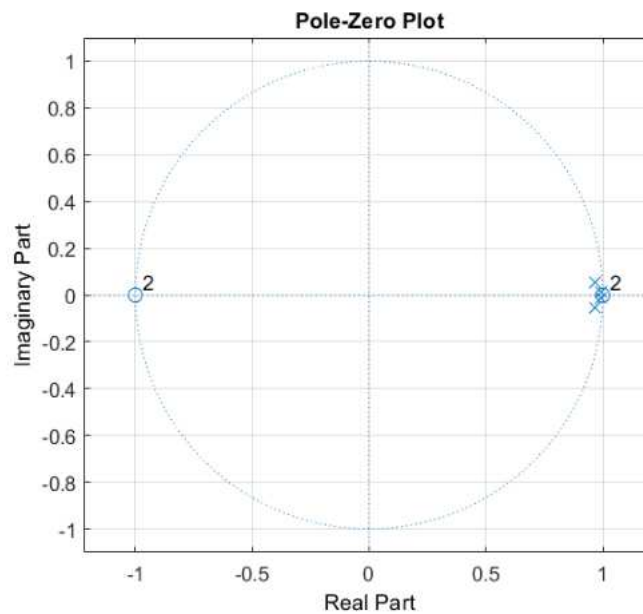
Figura 27 – Diagrama de fase, destacando a faixa de 100 a 500 Hz.



Fonte: Autor.

No diagrama de polos e zeros, Figura 28, podemos observar que existem dois zeros (representados por "o") e polos (representados por "x") localizados dentro do círculo unitário. A presença de zeros e polos no plano complexo é essencial para determinar o comportamento do filtro. No caso do filtro passa-banda *Butterworth*, os polos estão distribuídos de forma que garantem uma resposta estável e suave, sem oscilações, dentro da faixa de passagem (100 a 500 Hz). A localização dos polos dentro do círculo unitário assegura que o filtro seja estável, o que é uma característica fundamental para garantir a fidelidade do sinal. Já os zeros influenciam a forma da transição entre a faixa de passagem e a faixa de rejeição, ajudando a controlar a atenuação das frequências fora da faixa de interesse, mas não afetam a estabilidade do sistema. A distribuição dos polos e zeros neste diagrama garante que o filtro seja eficaz em preservar as frequências dentro da faixa de 100 a 500 Hz, ao mesmo tempo em que atenua as frequências fora dessa faixa, sem introduzir distorções ou instabilidade no sinal. Assim, o filtro apresenta um comportamento esperado para a análise dos sinais, dentro da faixa de interesse.

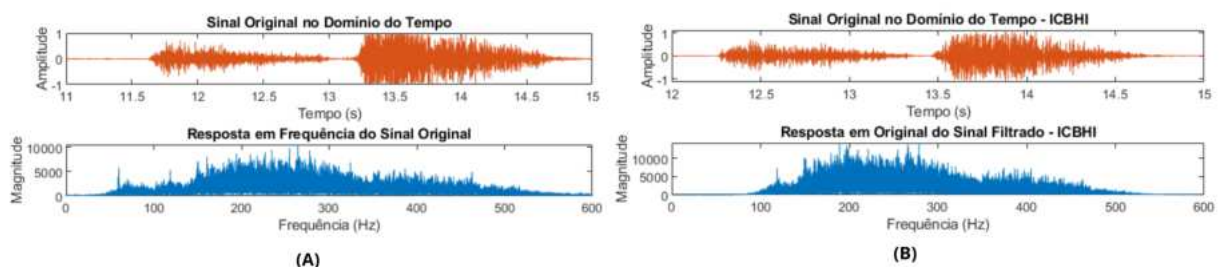
Figura 28 – Diagrama de polos e zeros.



Fonte: Autor.

Assim os sinais captados pelo sensor foram processados no MATLAB, onde foram aplicadas técnicas como a Transformada Rápida de Fourier (FFT) para identificar as principais frequências presentes no sinal respiratório. Essa abordagem permitiu visualizar os padrões espectrais dos sons pulmonares, visto na Figura 29 (A), e compará-los com os espectros extraídos do banco de dados ICBHI, visto na Figura 29 (B) garantindo que o sensor fosse capaz de captar sinais com características acústicas semelhantes às registradas em um ambiente clínico.

Figura 29 – Sinais captados pelo sensor INMP441 e ICBHI.



Fonte: Autor.

Embora o MATLAB tenha sido eficiente para validar o sensor e realizar a análise dos sinais captados, identificou-se uma limitação no uso da ferramenta para a filtragem em tempo real. A dependência de um computador externo para processar os dados e a latência

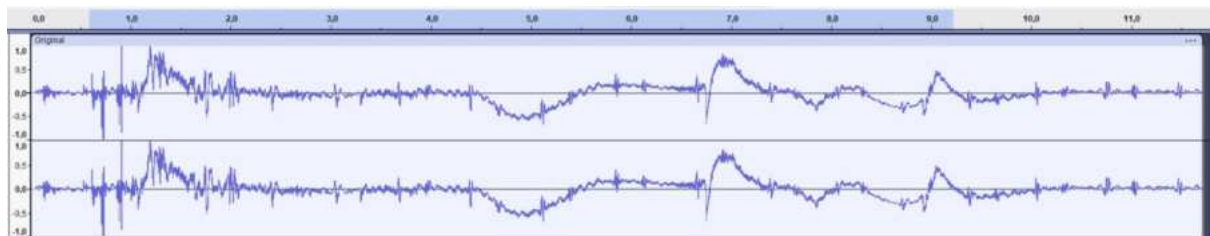
introduzida durante a transferência dos sinais tornam o uso do MATLAB inviável nas aplicações práticas desejadas para monitoramento respiratório nesse estudo. Além disso, a necessidade de transferir grandes quantidades de dados brutos para o MATLAB exige recursos de comunicação e armazenamento adicionais, o que não é ideal para sistemas embarcados. Com base nos resultados obtidos, o filtro foi projetado e implementado no STM32, garantindo o processamento dos sinais em tempo real e a transmissão eficiente dos dados ao aplicativo.

4.2.2 Filtragem

Os gráficos apresentados fornecem uma visão do processamento de sinais captados pelo sensor INMP441, destacando as etapas de filtragem realizadas no STM32 para melhorar a qualidade dos dados adquiridos.

Na Figura 30, observa-se o sinal bruto captado, que apresenta um alto nível de ruído. Esses ruídos incluem interferências de baixa frequência, como batimentos cardíacos, e de alta frequência, como ruídos ambientais, tornando a análise médica do som inicial significativamente difícil. Além disso, a presença de artefatos relacionados à manipulação do protótipo também é evidente.

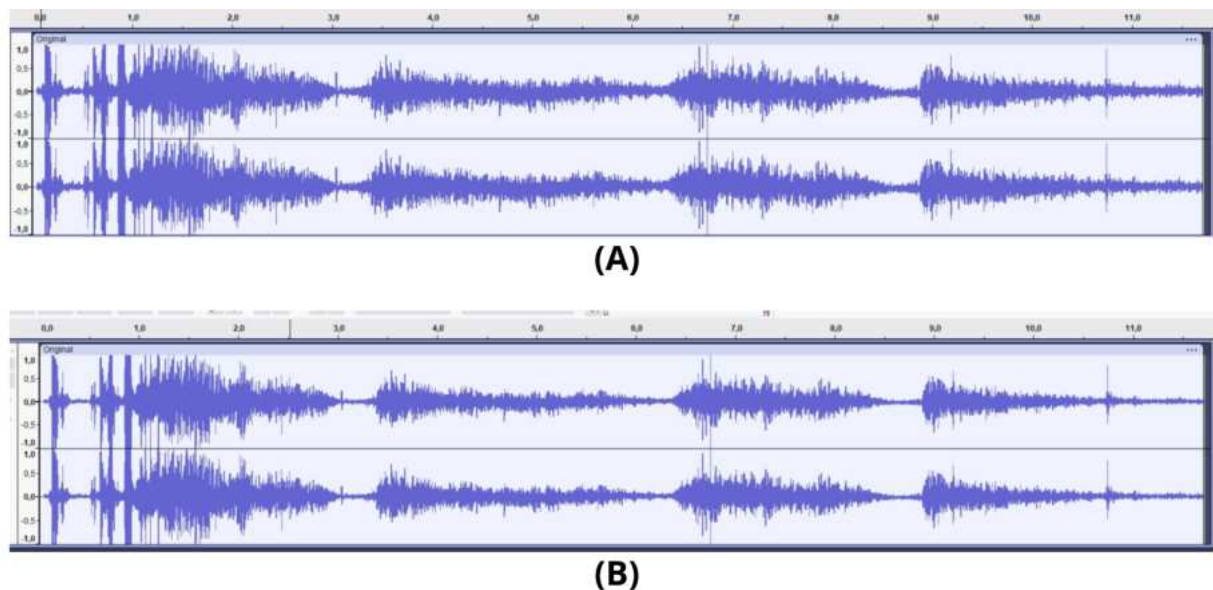
Figura 30 – Sinais captados pelo inicialmente pelo sensor INMP441 e STM32.



Fonte: Autor.

Já na Figura 31 (A), foi aplicado um filtro passa-banda, projetado para priorizar a faixa de frequência de interesse (100 Hz a 500 Hz). O filtro eliminou ruídos de baixa frequência e interferências de alta frequência. A Figura 31 (B), em particular, demonstra uma redução significativa dos ruídos, resultando em um sinal mais limpo e adequado para análise médica detalhada. Essa etapa final de filtragem assegura a qualidade do sinal processado, facilitando a identificação de possíveis anormalidades respiratórias.

Figura 31 – Sinais captados pelo sensor INMP441 e STM32 filtrados.



Fonte: Autor.

4.3 Comunicação do sistema

Após a validação da captação do sinal, o próximo passo foi testar a comunicação entre o STM32 e o ESP32, utilizando o protocolo I2C. Nesse processo, o microcontrolador foi responsável por transmitir os dados captados e processados para o módulo de comunicação, que os encaminhou ao aplicativo. Na Figura 32, são apresentados os valores enviados entre o STM32 e o ESP32 durante a comunicação, onde verificou-se que as informações são recebidas sem atrasos, permitindo a apresentação dos resultados em tempo real.

Para a comunicação entre o ESP32 e o aplicativo, foi implementado o protocolo *Bluetooth Low Energy* (BLE), permitindo a transmissão eficiente dos dados processados pelo microcontrolador. O ESP32, configurado como um dispositivo BLE, envia os dados capturados diretamente para o aplicativo em tempo real, sem a necessidade de conexões físicas. Dessa forma, para iniciar a sessão de ausculta pulmonar, o usuário deve conectar o *smartphone* ao ESP32 via *Bluetooth*, como ilustrado na Figura 33, possibilitando a troca de informações e a transmissão dos dados captados de forma confiável para o aplicativo.

Figura 32 – Comunicação entre o microcontrolador STM32 e o módulo ESP32.

```

COM4
-- Sistema de Análise Pulmonar --
- Iniciando Comunicação com ESP32
#OK
211
236
158
156
105
143
42
152
178
82
0
223
6
58
248
275
171
281
44
155

COM5
-- Sistema de Análise Pulmonar --
- Aguardando Sinal do STM 32
#OK - Recebido
211
236
158
156
105
143
42
152
178
82
0
223
6
58
248
275

```

Fonte: Autor.

Figura 33 – Comunicação entre o ESP32 e o aplicativo.



Fonte: Autor.

4.4 Aplicativo

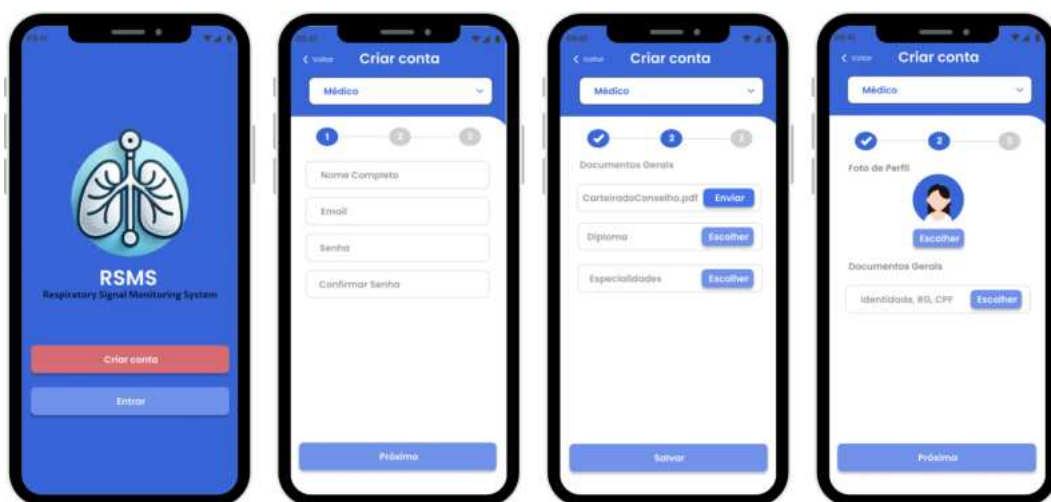
4.4.1 Interface do *front-end*

Os resultados das telas desenvolvidas, no ambiente Figma², demonstram a interação entre o usuário e o sistema, destacando a importância da interface gráfica para facilitar o acesso e a compreensão das informações. As telas foram configuradas para atender dois perfis distintos de acesso: médicos e pacientes.

4.4.1.1 Acesso médico

No caso do médico, a Figura 34 apresenta as telas de cadastro, onde o profissional deve realizar seu registro e anexar um documento que comprove sua profissão, garantindo maior segurança e confiabilidade para o paciente.

Figura 34 – Telas de *login* - Médico.



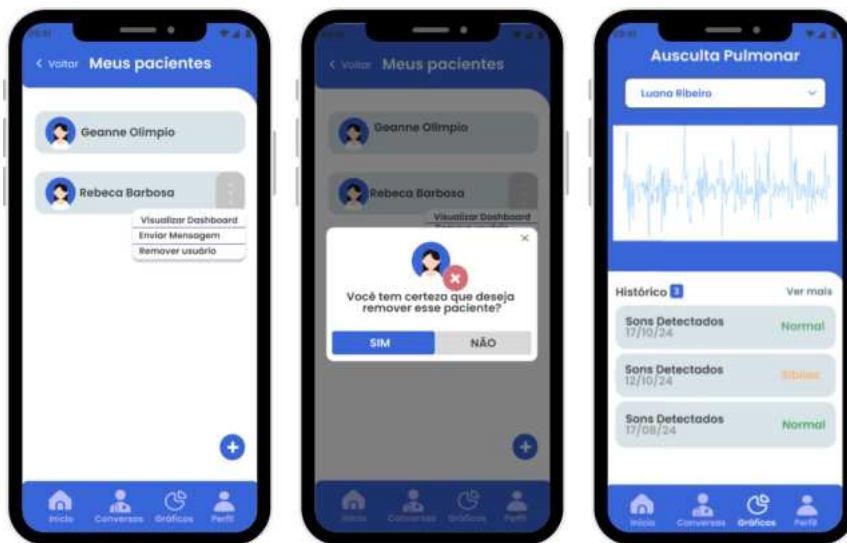
Fonte: Autor.

Já as telas da Figura 35 mostram a seleção de pacientes, permitindo ao médico buscar os pacientes cadastrados em seu banco de dados. Além disso, ele pode enviar mensagens, monitorar os dados dos pacientes e até removê-los, se necessário.

A interface foi projetada para ser limpa e organizada, permitindo que o médico visualize os sinais captados de forma clara e acompanhe o histórico de monitoramento dos pacientes. Essas funcionalidades visam facilitar o uso e garantir que as informações sejam apresentadas de maneira acessível e intuitiva.

² <<https://goo.su/mdOLS>>

Figura 35 – Telas de acompanhamento de pacientes - Médico.

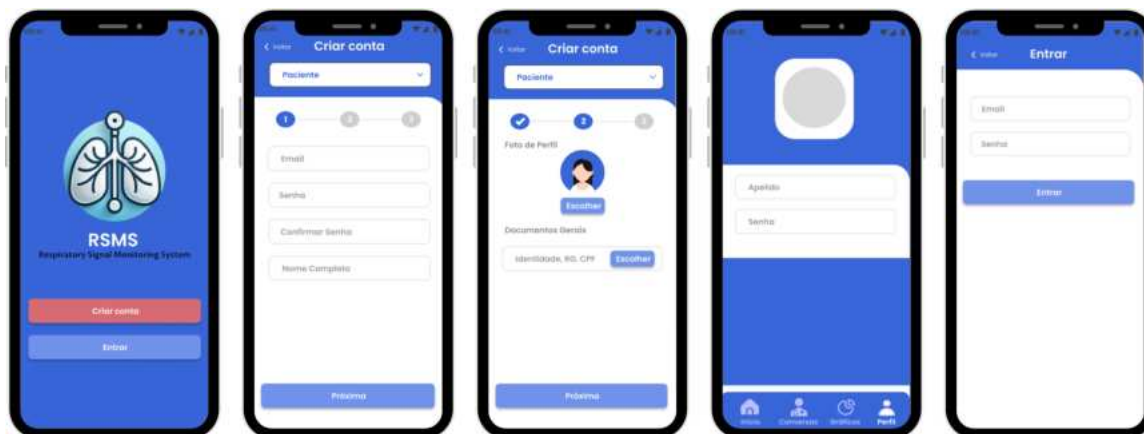


Fonte: Autor.

4.4.1.2 Acesso pacientes

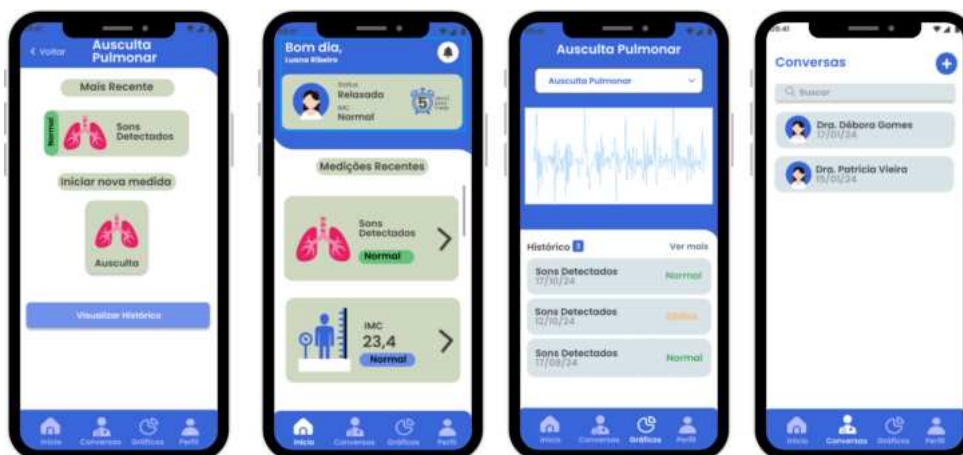
Para os pacientes, a Figura 36 apresenta as telas de *login*, projetadas de forma simples e intuitiva. Já a Figura 37 exhibe as telas com foco na apresentação clara dos dados essenciais para o acompanhamento da saúde do paciente. Nessa interface, os pacientes podem iniciar um novo monitoramento e visualizar as informações sobre suas condições respiratórias de maneira acessível e também se comunicar com o profissional de saúde.

Figura 36 – Telas *login* - Pacientes.



Fonte: Autor.

Figura 37 – Telas principais - Pacientes.



Fonte: Autor.

4.4.2 Implementação do *back-end* e *front-end*

A Figura 38 mostra a tela principal do aplicativo, desenvolvida como parte da interface *front-end* projetada com foco em clareza e acessibilidade, que exibe o monitoramento dos sinais pulmonares em tempo real e um histórico das classificações realizadas pelo profissional de saúde responsável.

Figura 38 – Tela principal de monitoramento.



Fonte: Autor.

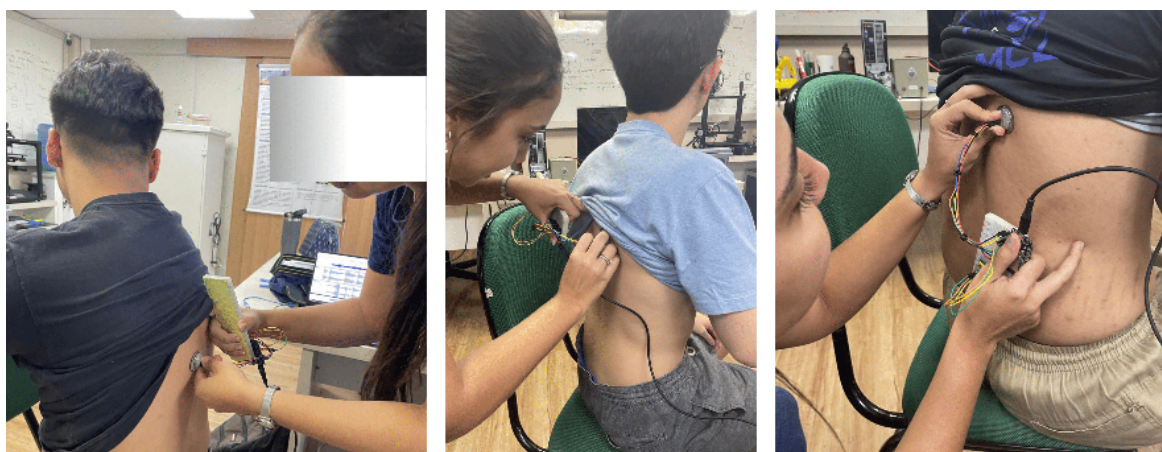
No histórico exibido, cada registro contém a data e o diagnóstico, que pode ser inserido por um profissional como "Normal" ou "Sibilos", fornecendo um acompanhamento organizado dos sinais analisados. Essa funcionalidade é alimentada pelo *back-end*, que processa e armazena os dados recebidos do protótipo no banco de dados, integrando-os com a interface gráfica.

A comunicação entre *back-end* e *front-end* é feita via *Bluetooth*, com os dados sendo transmitidos pela ESP32 diretamente para o aplicativo. Essa integração garante a transmissão confiável dos dados processados, permitindo ao usuário uma experiência contínua e funcional durante a análise dos sinais pulmonares. O resultado reflete uma implementação eficiente da interação entre *hardware* e *software*, traduzindo os dados técnicos em informações acessíveis e úteis.

4.5 Procedimento experimental

Durante a experimentação com o protótipo, foi essencial garantir uma posição adequada do sensor para a captura de dados precisos e confiáveis. O estetoscópio, com o sensor INMP441 acoplado, foi posicionado nas áreas específicas de ausculta pulmonar, conforme ilustrado na Figura 39, enquanto o paciente realizava movimentos controlados de inspiração e expiração.

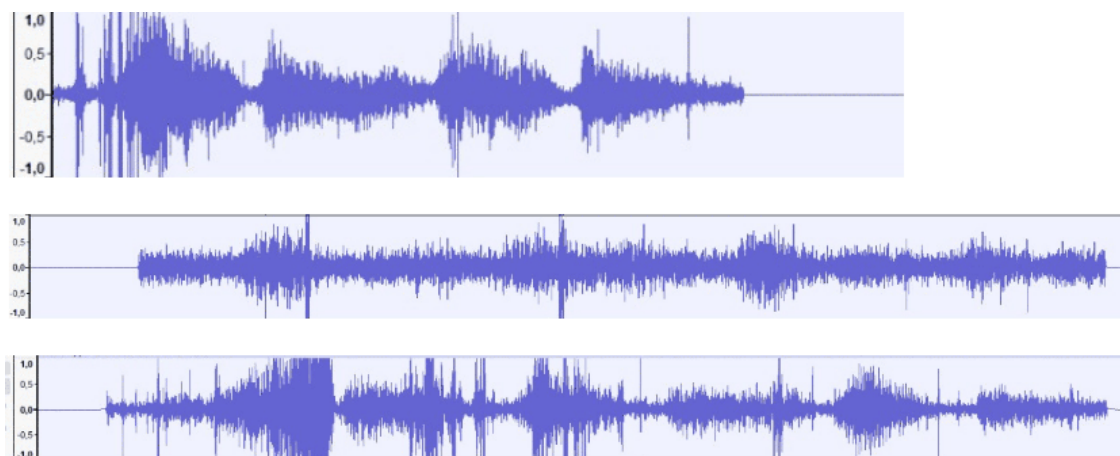
Figura 39 – Realização dos testes.



Fonte: Autor.

Os sons captados apresentaram uma qualidade satisfatória, proporcionando uma audição clara, com baixos ruídos indesejados e fidelidade na reprodução das características acústicas esperadas, conforme ilustrado na Figura 55. Podendo assim serem enviados para o aplicativo, que permite uma avaliação clínica da classificação destes sons.

Figura 40 – Dados captados nos testes.



Fonte: Autor.

O sistema desenvolvido foi submetido a testes com profissionais da área de saúde para validação em relação à sua aplicabilidade e aceitação como ferramenta de auxílio na realização da ausculta pulmonar, como mostra a Figura 41. Durante os testes, o dispositivo foi utilizado em simulações de cenários clínicos, permitindo aos profissionais avaliar a precisão e a praticidade do sistema. Após a utilização, foi aplicado um questionário (*quiz*), Figura 42, para coletar *feedback* sobre aspectos como facilidade de uso, qualidade dos sinais captados e potencial de aplicação no diagnóstico e monitoramento pulmonar.

Figura 41 – Realização dos testes por profissionais da saúde.



Fonte: Autor.

Figura 42 – Aplicação do *quiz* para coleta de *feedback*.

Fonte: Autor.

Os resultados obtidos com o *quiz* de avaliação, visto na Tabela 2 embora baseados em uma amostra pequena de profissionais, foram de grande importância para validar a proposta do sistema. Dois participantes classificaram o sistema como "ótimo", enquanto um o avaliou como "bom", destacando aspectos como a facilidade de uso e a intuitividade da interface. O potencial do sistema como ferramenta auxiliar na ausculta pulmonar foi considerado "muito alto" por dois profissionais e "alto" por outro, refletindo sua relevância prática. Além disso, todos os entrevistados recomendaram o uso do sistema em aplicações clínicas. Apesar da limitação da amostra, os *feedbacks* recebidos forneceram *insights* valiosos para o aprimoramento do sistema, reforçando sua viabilidade e apontando direções para futuros ajustes.

Tabela 2 – Respostas ao *quiz* de Avaliação do Sistema.

Perguntas	Profissional 1	Profissional 2	Profissional 3
Como você avalia a qualidade dos sinais captados pelo sistema?	Ótima	Ótima	Boa
O sistema é fácil de usar e intuitivo?	Totalmente intuitivo	Totalmente intuitivo	Totalmente intuitivo
Qual é o potencial do sistema como ferramenta auxiliar na ausculta pulmonar?	Muito alto	Muito alto	Alto
Você recomendaria o uso deste sistema em aplicações clínicas?	Sim, com certeza	Sim, com certeza	Sim, com certeza
Como você classificaria o sistema de forma geral?	Ótimo	Ótimo	Bom

Fonte: Autor.

O gráfico apresentado na Figura 43 ilustra de forma clara a distribuição das respostas

obtidas no *quiz*, destacando a predominância de avaliações positivas, especialmente nas categorias de qualidade do sinal, facilidade de uso e recomendação. Essa representação visual reforça a percepção de que o sistema atende às expectativas iniciais de funcionalidade e aplicabilidade clínica.

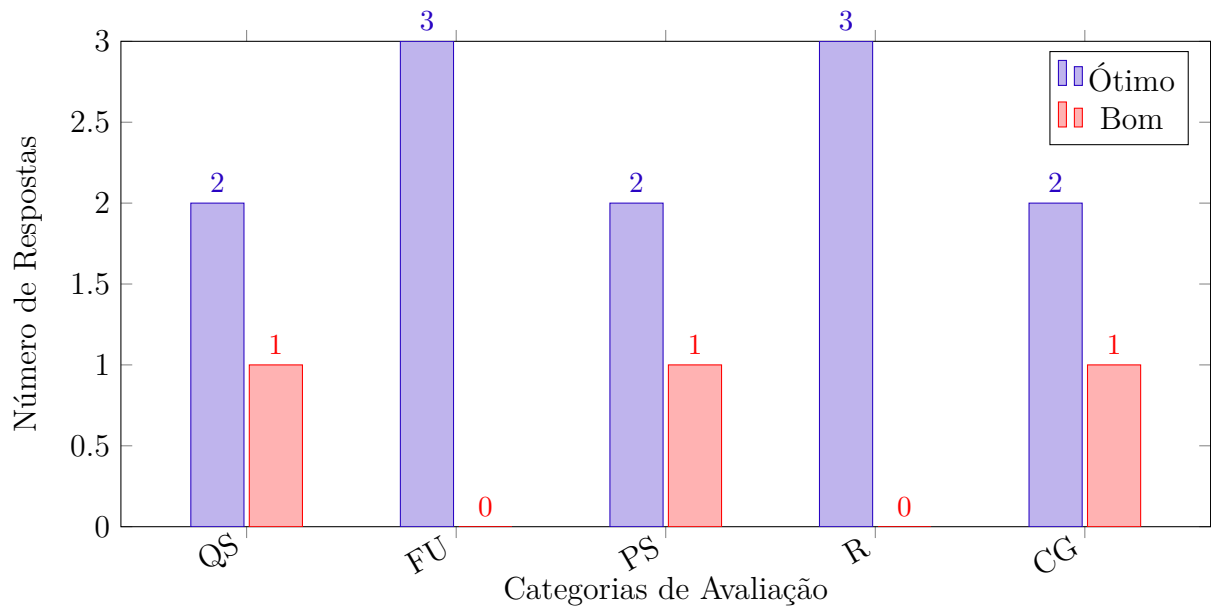


Figura 43 – Distribuição das Respostas do Quiz de Avaliação.

Legenda

- QS - Qualidade do Sinal
- FU - Facilidade de Uso
- PS - Potencial do sistema
- R - Recomendação
- CG - Classificação Geral

5 Conclusões

O presente trabalho teve como objetivo o desenvolvimento de um sistema integrado de *hardware* e *software* para o monitoramento de sons pulmonares, combinando tecnologias para a captação, filtragem e transmissão de sinais acústicos. O *hardware* foi composto pelo sensor de microfone INMP441, responsável pela captação dos sons respiratórios, e pelos microcontroladores STM32 e ESP32, que realizaram o processamento e a transmissão dos dados. Para garantir a qualidade do sinal captado, foi aplicada uma filtragem diretamente no STM32, utilizando um filtro passa-banda com frequência de corte entre 100 Hz e 500 Hz, faixa escolhida por ser relevante para a análise de sons pulmonares. Essas frequências são adequadas para eliminar ruídos de baixa frequência, como batimentos cardíacos, e de alta frequência, como interferências ambientais, garantindo a clareza do sinal respiratório.

A comunicação entre os dispositivos foi estabelecida por meio do protocolo I2C para a troca de dados entre o STM32 e o ESP32. Em seguida, os dados foram transmitidos ao aplicativo utilizando *Bluetooth Low Energy* (BLE). Essa configuração possibilitou uma comunicação eficiente, com dados sendo enviados em tempo real, atendendo a um dos principais requisitos do sistema. O aplicativo desenvolvido mostrou-se uma ferramenta útil para profissionais de saúde, permitindo o monitoramento remoto de pacientes de forma prática e acessível.

Nos testes experimentais, foram utilizados sons reais, captados com o sensor INMP441, além de sons respiratórios disponibilizados em sites especializados, para realizar comparações com os dados do banco de dados ICBHI. O ICBHI é um banco de dados amplamente utilizado, oferecendo uma ampla gama de sinais respiratórios, tanto em formato de áudio quanto em texto, permitindo uma análise detalhada da sensibilidade do sensor. A comparação entre os dados captados e os fornecidos pelo banco de dados confirmou que o sensor apresentou boa sensibilidade, com os sinais capturados sendo de boa qualidade e confiabilidade para análise.

A análise espectral dos sinais no MATLAB confirmou o bom funcionamento do protótipo, e os testes de comunicação demonstraram que os dispositivos se conectaram de forma estável, atendendo aos requisitos do sistema. A filtragem aplicada no STM32 contribuiu significativamente para a remoção de ruídos, facilitando a interpretação dos sinais respiratórios. Esses resultados validaram o sistema como uma solução confiável e eficaz para o monitoramento de sons pulmonares em tempo real.

Além disso, o sistema foi testado com profissionais da área de saúde para validar sua aplicabilidade. Durante os testes, os profissionais utilizaram o protótipo em cenários simulados de ausculta pulmonar e forneceram *feedback* sobre a eficácia do sistema. Embora

a amostra de profissionais tenha sido pequena, as opiniões coletadas foram importantes para a avaliação da aceitação do sistema, destacando a facilidade de uso, a qualidade dos sinais captados e a viabilidade do sistema para ser utilizado em ambientes clínicos. Os resultados foram positivos, confirmando a efetividade do dispositivo como ferramenta de auxílio no monitoramento de sinais respiratórios.

5.1 Trabalhos Futuros

Nos trabalhos futuros, pretende-se integrar o sistema de monitoramento de sons pulmonares à solução existente que já inclui a análise de sinais de ECG (eletrocardiograma) e SpO₂ (saturação) como mostra [Gomes, Soares e Vilchez \(2023\)](#). Serão realizados testes adicionais para aprimorar a captação e a filtragem dos sinais respiratórios, com o intuito de melhorar a precisão na análise e interpretação desses dados. A ideia é otimizar os algoritmos de processamento de sinais pulmonares, garantindo uma avaliação mais detalhada e precisa, alinhando os dados respiratórios com os sinais cardíacos e de oxigenação.

Além disso, pretende-se implementar esses aprimoramentos na plataforma já estabelecida, permitindo uma análise integrada e em tempo real dos sinais respiratórios, cardíacos e de oxigenação. Isso contribuirá para um monitoramento mais completo e eficiente, oferecendo aos profissionais de saúde uma ferramenta avançada para acompanhamento remoto e avaliação contínua de pacientes com condições respiratórias e cardiovasculares.

Referências

- ABELLA, M.; PENNEY, D. G. Comparison of the acoustic properties of six popular stethoscopes. *Journal of the Acoustical Society of America*, v. 91, 1992. ISSN NA. Citado na página 12.
- ALBAHRI, A. S. et al. *IoT-based telemedicine for disease prevention and health promotion: State-of-the-Art*. 2021. Citado na página 23.
- BAID, H. *The process of conducting a physical assessment: a nursing perspective*. 2006. Citado na página 17.
- BAKER, J.; STANLEY, A. *Telemedicine Technology: a Review of Services, Equipment, and Other Aspects*. 2018. Citado na página 22.
- BATISTA, O. F. H.; GARCIA, M. V. R. Automatização do sistema de proteção térmica para sala de servidores de redes utilizando arduino. *Caleidoscópio*, v. 1, 2015. Citado na página 25.
- BOTT, R. *Guyton and Hall Textbook of Medical Physiology 13ed.* [S.l.: s.n.], 2014. ISSN 0717-6163. Citado na página 16.
- BRITO, H. M. B. F.; OLIVEIRA, J. Bezerra de. Procedimento de medida da vibração de corpo inteiro via programação matlab de acordo com a iso 2631. *Interscientia*, v. 5, n. 2, p. 136–146, 2017. Disponível em: <<https://periodicos.unipe.br/index.php/interscientia/article/view/476>>. Citado na página 28.
- CAMPANHOLI, L. L. *Fundamentos e práticas da fisioterapia*. [S.l.: s.n.], 2018. Citado na página 12.
- CAMPOS, R. da S. et al. Efeito do aumento do fluxo expiratório ativo-assistido em crianças com pneumonia. *Arquivos Médicos do ABC*, v. 32, 2007. ISSN 2358-0739. Citado na página 15.
- CARVALHO, V. O.; SOUZA, G. E. C. O estetoscópio e os sons pulmonares: uma revisão da literatura. *Revista de Medicina*, v. 86, 2007. ISSN 0034-8554. Citado na página 16.
- COLLIMATOR. *What is a second order filter?* 2023. Acesso em: 28 out. 2023. Disponível em: <<https://www.collimator.ai/reference-guides/what-is-a-second-order-filter>>. Citado na página 28.
- DENG, B. et al. Research on stm32 development board based on arm cortex-m3. In: *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. [S.l.]: IEEE, 2020. Citado na página 25.
- GOMES, L. R.; SOARES, J. V. B.; VILCHEZ, J. R. S. Aplicativo móvel de monitoramento e registro de sinais biomédicos. In: _____. [S.l.: s.n.], 2023. Citado na página 61.
- GROSS, V. et al. The relationship between normal lung sounds, age, and gender. *American Journal of Respiratory and Critical Care Medicine*, v. 162, 2000. ISSN 1073449X. Citado na página 16.

- HIRATA, M. H. Biofísica: fundamentos e aplicações. *Revista Brasileira de Ciências Farmacêuticas*, v. 40, 2004. ISSN 1516-9332. Citado 2 vezes nas páginas 16 e 20.
- IBRAHIM, D. *The Complete ESP32 Projects Guide*. [S.l.]: Elektor International Media BV, 2019. Citado na página 27.
- KHAWAS, C.; SHAH, P. Application of firebase in android app development-a study. *International Journal of Computer Applications*, v. 179, 2018. Citado na página 31.
- KOLBAN, N. *Kolban's Book on ESP32*. [S.l.]: Elektor International Media BV, 2018. Citado na página 27.
- LEHRER, S. *Understanding Lung Sounds: Third Edition*. [S.l.]: W.B. Saunders Company, 2002. Citado 2 vezes nas páginas 17 e 19.
- LENG, S. et al. *The electronic stethoscope*. 2015. Citado na página 19.
- LEON, G. O. de M.; AMARAL Érico Marcelo Hoff do; JÚNIO, J. S. D. Sistema para auxílio na aplicação e análise da ausculta pulmonar. *XIX Congresso Brasileiro de Informática em Saúde*, 2022. Citado 2 vezes nas páginas 15 e 24.
- LEON, G. O. de M.; AMARAL Érico Marcelo Hoff do; JÚNIOR, J. S. D. Sistema para auxílio na aplicação e análise da ausculta pulmonar. *Journal of Health Informatics*, v. 15, 2023. Citado na página 39.
- LIBERA, F. B. D.; OLIVEIRA, A. A. M. de. Monitoramento de sinais vitais no ambiente da telemedicina. 2021. Citado na página 39.
- MAKER PORTAL. *SAMD21 M0 Mini Arduino Board*. 2019. 15 jan. 2024. Disponível em: <<https://makersportal.com/blog/2019/6/14/samd21-m0-mini-arduino-board>>. Citado 2 vezes nas páginas 26 e 33.
- MARTINAZZO, C. A.; ORLANDO, T. Comparação entre três tipos de sensores de temperatura em associação com arduino. *PERSPECTIVA, Erechim*. v. 40, n.151, p. 93-10, 2016. Citado na página 23.
- MIYAMOTOYANAGIYA, E. F. et al. Desenvolvimento de manipuladores robóticos visando sua utilização no ensino médio. *8º Congresso de extensão universitária da UNESP*, p. 1-4, 2015. Citado na página 24.
- MURPHY, R. L. *In defense of the stethoscope*. 2008. Citado 2 vezes nas páginas 15 e 16.
- NAVES, R. *UM ESTUDO DE RECONHECIMENTO DE SONS PULMONARES BASEADO EM TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL*. 2015. Citado 5 vezes nas páginas 16, 18, 19, 20 e 22.
- OLIVEIRA, J. L.; SILVA, R. O. D. A internet das coisas (iot) com enfoque na saúde. *Tecnologia em Projeção*, v. 8, 2017. Citado na página 23.
- PASTERKAMP, H.; KRAMAN, S. S.; WODICKA, G. R. Respiratory sounds. *American Journal of Respiratory and Critical Care Medicine*, v. 156, 1997. ISSN 1073-449X. Citado 3 vezes nas páginas 12, 17 e 18.
- REACT.DEV. *Learn React*. 2023. Citado na página 30.

- RIELLA, R. J. *PROPOSTAS DE TÉCNICAS PARA CARACTERIZAÇÃO E CLASSIFICAÇÃO AUTOMÁTICA DE SONS PULMONARES ADVENTÍCIOS*. 2008. Citado na página 21.
- SANTOSO, M. Implementation of ui/ux concepts and techniques in web layout design with figma. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, v. 6, n. 2, p. 279–285, Apr. 2024. Disponível em: <<https://jurnal.unidha.ac.id/index.php/jteksis/article/view/1223>>. Citado na página 31.
- SAVARATI. *ESP32 DevKitC WROOM-32D*. 2021. Acesso em: 2 fev. 2024. Disponível em: <<https://www.saravati.com.br/esp32-devkitc-wroom-32d.html>>. Citado 2 vezes nas páginas 27 e 35.
- SCHMIDT, A. L. *FFT: Transformada Rápida de Fourier*. Bahía Blanca, Argentina: [s.n.], 2013. Citado 2 vezes nas páginas 29 e 30.
- SOUSA, F. W. J. *Equalização de Sinais de Áudio*. 2004. Acesso em: 5 nov. 2023. Disponível em: <https://cursosdeaudiobh.com/wpcontent/uploads/2020/04/Equalizacao_de_Sinais_de_audio_-_Pesquisa_de_est%C3%A1gio.pdf>. Citado na página 28.
- SOUSA, M. D.; GONCALVES, A. Humanportal - a react.js case study. In: . [S.l.: s.n.], 2020. v. 2020-June. ISSN 21660735. Citado na página 30.
- TAYLOR, L.; CAPLING, H.; PORTNOY, J. M. *Administering a Telemedicine Program*. 2018. Citado na página 22.
- The MathWorks, Inc. *Documentação MATLAB. R2013a*. [S.l.], 2015. Disponível em: <<https://www.mathworks.com/help/matlab/>>. Acesso em: 23 nov. 2024. Citado na página 29.
- VALDÉS-PÉREZ, F. E.; PALLÀS-ARENY, R. *Microcontroladores: fundamentos y aplicaciones con PIC*. [S.l.: s.n.], 2007. Citado na página 24.
- VERRI, L. D. F. et al. Atuação fisioterapêutica no pós-operatório de pacientes submetidos a cirurgia bariátrica. *REVISTA SAÚDE DOS VALES*, 2023. Citado na página 15.
- VETTORAZZI, S. de F. *Implantação e resultados de um programa de reabilitação pulmonar em uma instituição de ensino superior*. 2006. Citado na página 15.
- VISICH, M. A. Knowing what you hear: A guide to assessing: Breath and heart sounds. *Nursing*, v. 11, 1981. ISSN 15388689. Citado na página 20.
- WALLER, M.; STOTLER, C. *Telemedicine: a Primer*. 2018. Citado na página 22.
- ZHONG, Y.; XU, Z. H.; CAO, L. Intelligent iot-based telemedicine systems implement for smart medical treatment. *Personal and Ubiquitous Computing*, v. 27, 2023. ISSN 16174917. Citado na página 23.

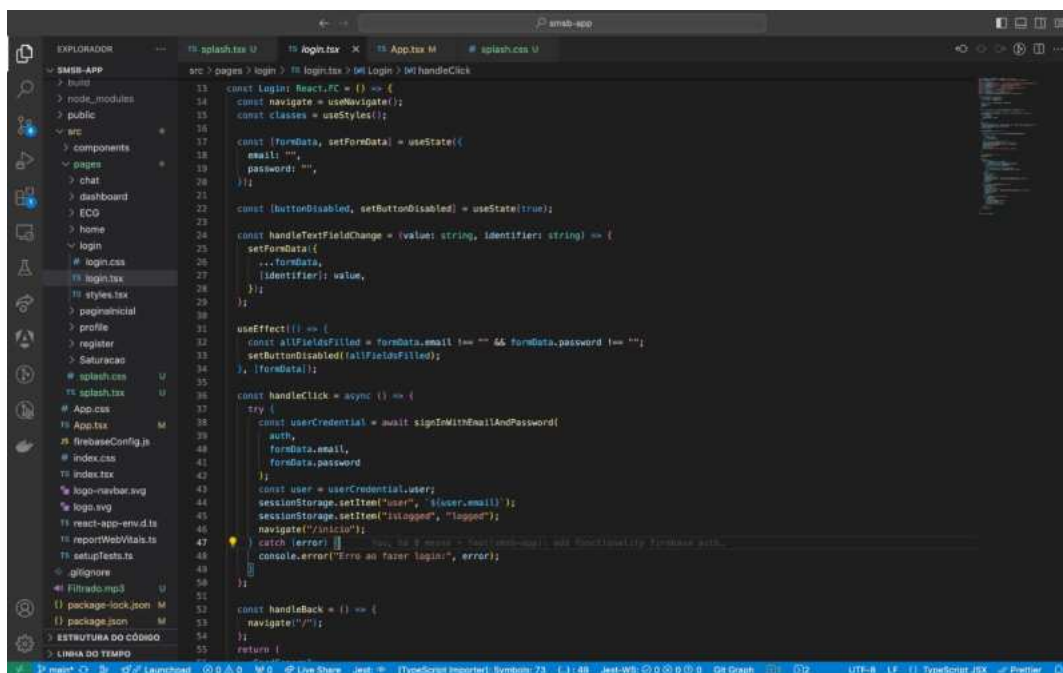
Apêndices

.1 Códigos de Implementação e Configuração do Sistema

.1.1 Códigos de cadastro e autenticação no *Firebase*

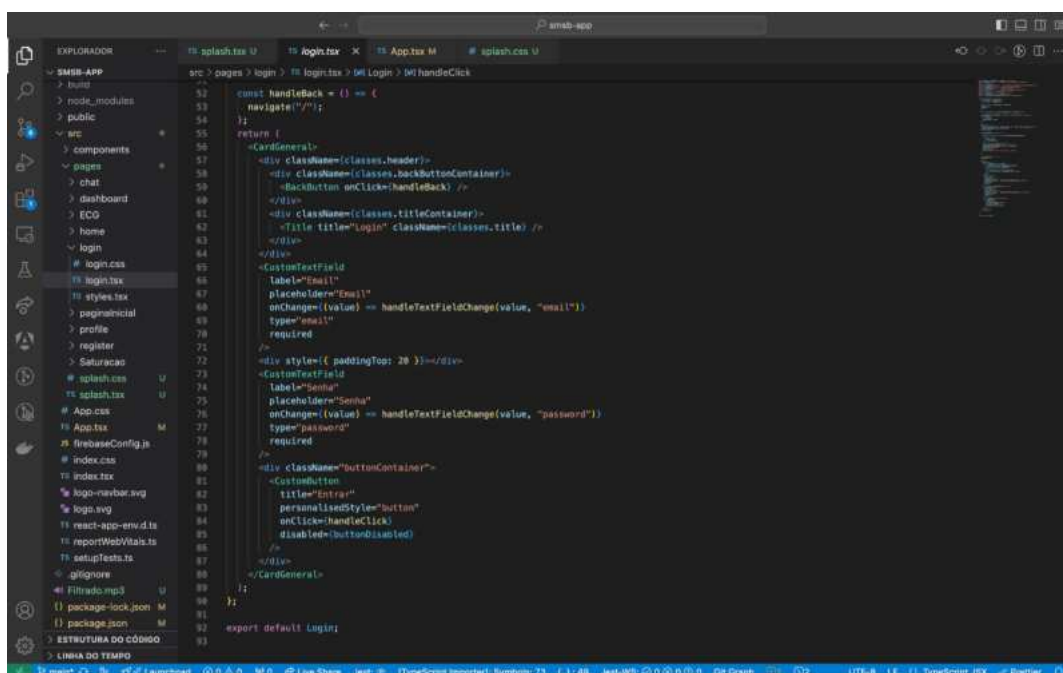
As figuras abaixo apresentam os códigos para *login* e autenticação do banco de dados.

Figura 44 – Página de *login* usando o *authenticate* do *Firebase*.



```
src > pages > login > login.tsx > login > login.handleClick
13  const login: React.FC = () => {
14  const navigate = useNavigate();
15  const classes = useStyles();
16
17  const [formData, setFormData] = useState({
18    email: "",
19    password: "",
20  });
21
22  const [buttonDisabled, setButtonDisabled] = useState(true);
23
24  const handleTextFieldChange = (value: string, identifier: string) => {
25    setFormData({
26      ...formData,
27      [identifier]: value,
28    });
29  };
30
31  useEffect(() => {
32    const allFieldsFilled = formData.email !== "" && formData.password !== "";
33    setButtonDisabled(!allFieldsFilled);
34  }, [formData]);
35
36  const handleClick = async () => {
37    try {
38      const userCredentials = await signInWithEmailAndPassword(
39        auth,
40        formData.email,
41        formData.password
42      );
43      const user = userCredentials.user;
44      sessionStorage.setItem("user", JSON.stringify(user));
45      sessionStorage.setItem("logged", "logged");
46      navigate("/inicio");
47    } catch (error) {
48      console.error("Erro ao fazer login:", error);
49    }
50  };
51
52  const handleBack = () => {
53    navigate("/");
54  };
55  return (
```

Fonte: Autor.

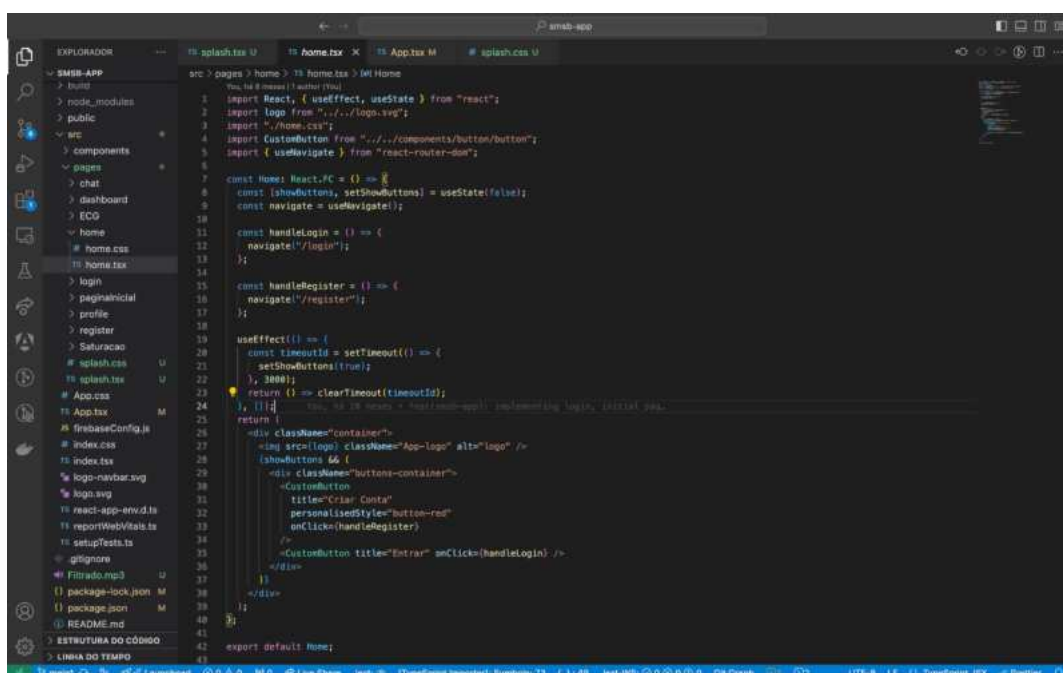
Figura 45 – Página de login usando o *authenticate* do *Firebase 2*.

```
src > pages > login > TS Login > M! handleClick
const handleBack = () => {
  navigate("/");
};

return (
  <CardGeneral>
    <div className={classes.header}>
      <div className={classes.backButtonContainer}>
        <BackButton onClick={handleBack} />
      </div>
      <div className={classes.titleContainer}>
        <Title title="Login" className={classes.title} />
      </div>
      <div>
        <CustomTextField
          label="Email"
          placeholder="Email"
          onChange={value => handleTextFieldChange(value, "email")}
          type="email"
          required
        />
        <div style={{ padding: 20 }}>
          <CustomTextField
            label="Senha"
            placeholder="Senha"
            onChange={value => handleTextFieldChange(value, "password")}
            type="password"
            required
          />
        </div>
        <div className="buttonContainer">
          <CustomButton
            title="Entrar"
            personalisedStyle="button"
            onClick={handleClick}
            disabled={buttonDisabled}
          />
        </div>
      </div>
    </CardGeneral>
  );
};

export default Login;
```

Fonte: Autor.

Figura 46 – Página de *Home* com opções de *login* e *register*.

```
src > pages > home > TS Home > M! Home
import React, { useEffect, useState } from "react";
import logo from "../logo.svg";
import "./home.css";
import CustomButton from "../components/button/button";
import { useNavigate } from "react-router-dom";

const Home: React.FC = () => {
  const [showButtons, setShowButtons] = useState(false);
  const navigate = useNavigate();

  const handleLogin = () => {
    navigate("/login");
  };

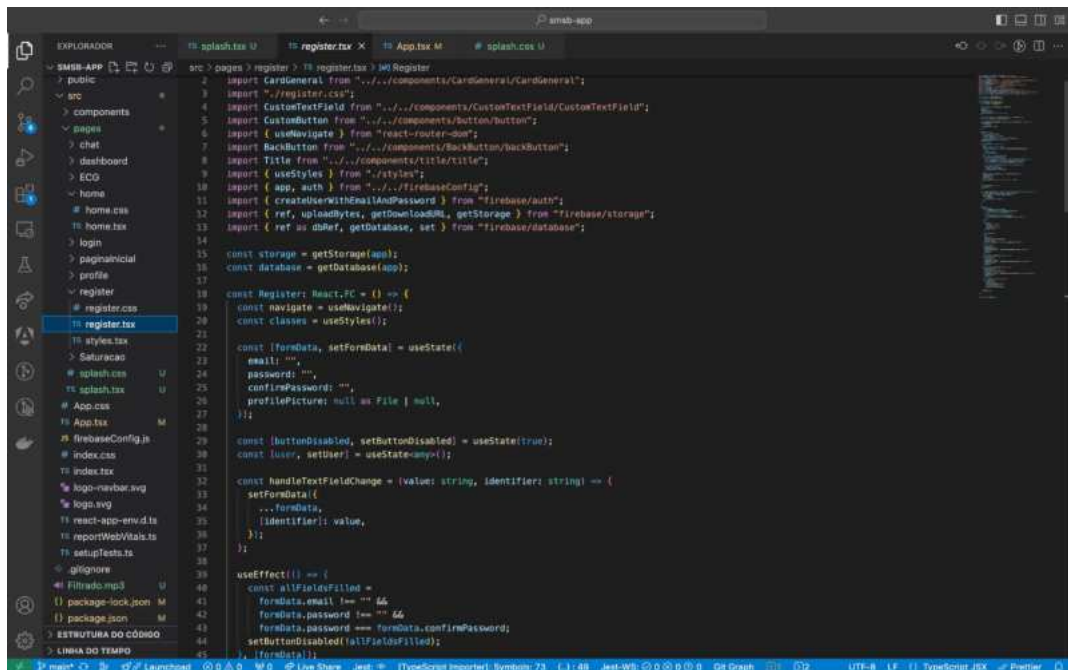
  const handleRegister = () => {
    navigate("/register");
  };

  useEffect(() => {
    const timeoutId = setTimeout(() => {
      setShowButtons(true);
    }, 3000);
    return () => clearTimeout(timeoutId);
  }, []);

  return (
    <div className="container">
      <img src={logo} className="App-logo" alt="logo" />
      {showButtons && (
        <div className="buttons-container">
          <CustomButton
            title="Criar Conta"
            personalisedStyle="button-red"
            onClick={handleRegister}
          />
          <CustomButton title="Entrar" onClick={handleLogin} />
        </div>
      )}
    </div>
  );
};

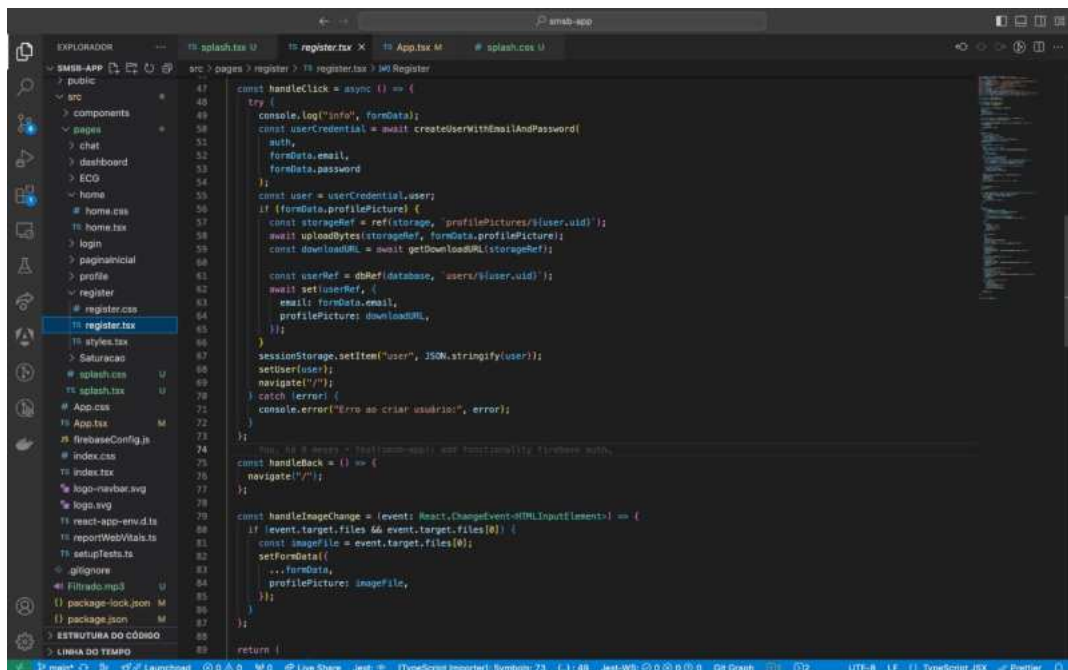
export default Home;
```

Fonte: Autor.

Figura 47 – Página *register*.

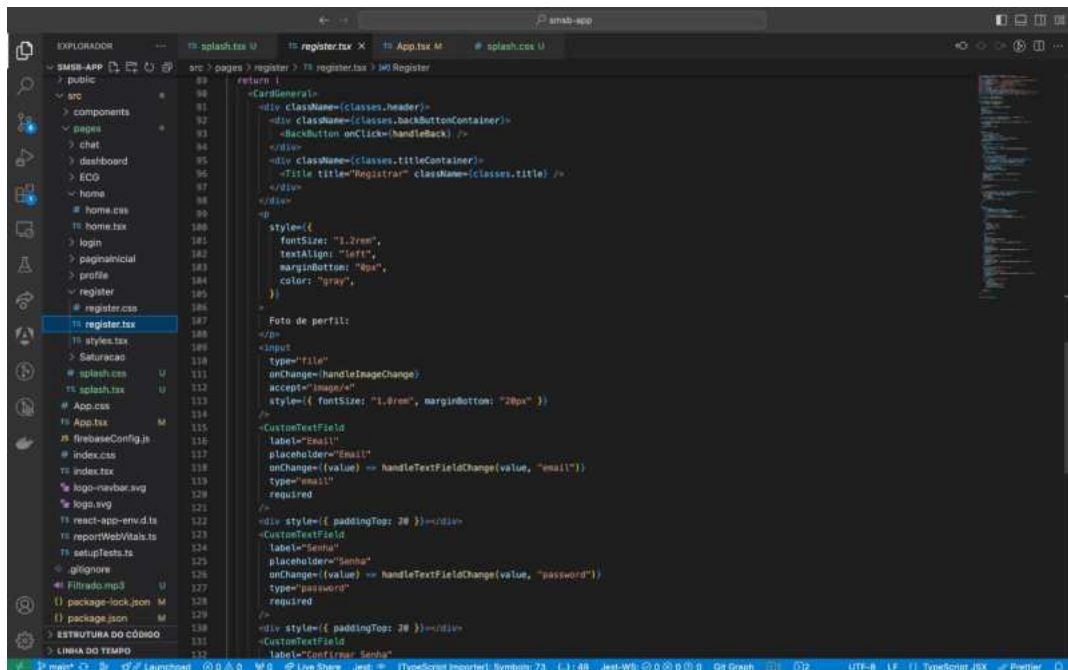
```
src > pages > register > register.tsx | 10 Register
 2 import CardGeneral from "../../components/CardGeneral/CardGeneral";
 3 import "../register.css";
 4 import CustomTextField from "../../components/CustomTextField/CustomTextField";
 5 import CustomButton from "../../components/button/button";
 6 import { useNavigate } from "react-router-dom";
 7 import BackButton from "../../components/BackButton/backButton";
 8 import Title from "../../components/title/title";
 9 import { useStyle } from "../style";
10 import { app_auth } from "../../firebaseConfig";
11 import { createUserWithEmailAndPassword } from "firebase/auth";
12 import { ref, uploadBytes, getDownloadURL, getStorage } from "firebase/storage";
13 import { ref as dbRef, getDatabase, set } from "firebase/database";
14
15 const storage = getStorage(app);
16 const database = getDatabase(app);
17
18 const Register: React.FC = () => {
19   const navigate = useNavigate();
20   const classes = useStyle();
21
22   const [formData, setFormData] = useState({
23     email: "",
24     password: "",
25     confirmPassword: "",
26     profilePicture: null as File | null,
27   });
28
29   const [buttonDisabled, setButtonDisabled] = useState(true);
30   const [user, setUser] = useState<any>();
31
32   const handleTextFieldChange = (value: string, identifier: string) => {
33     setFormData({
34       ...formData,
35       [identifier]: value,
36     });
37   };
38
39   useEffect(() => {
40     const allFieldsFilled =
41       formData.email !== "" &&
42       formData.password !== "" &&
43       formData.password === formData.confirmPassword;
44     setButtonDisabled(!allFieldsFilled);
45   }, [formData]);
```

Fonte: Autor.

Figura 48 – Página de registrar usuário no *Firebase*.

```
src > pages > register > register.tsx | 10 Register
47 const handleClick = async () => {
48   try {
49     console.log("info", formData);
50     const userCredential = await createUserWithEmailAndPassword(
51       auth,
52       formData.email,
53       formData.password
54     );
55     const user = userCredential.user;
56     if (formData.profilePicture) {
57       const storageRef = ref(storage, `profilePictures/${user.uid}`);
58       await uploadBytes(storageRef, formData.profilePicture);
59       const downloadURL = await getDownloadURL(storageRef);
60
61       const userRef = dbRef(database, `users/${user.uid}`);
62       await setUserRef, {
63         email: formData.email,
64         profilePicture: downloadURL,
65       });
66     }
67     sessionStorage.setItem("user", JSON.stringify(user));
68     setUser(user);
69     navigate("/");
70   } catch (error) {
71     console.error("Erro ao criar usuário:", error);
72   }
73 };
74
75 // 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
80 const handleImageChange = (event: React.ChangeEvent<HTMLInputElement>) => {
81   if (event.target.files && event.target.files[0]) {
82     const imageFile = event.target.files[0];
83     setFormData({
84       ...formData,
85       profilePicture: imageFile,
86     });
87   }
88 };
89
90 return;
```

Fonte: Autor.

Figura 49 – Página de registrar usuário no *Firebase 2*.

```
return (
  <CardHeader>
    <div className={classes.header}>
      <div className={classes.backButtonContainer}>
        <BackButton onClick={handleBack} />
      </div>
      <div className={classes.titleContainer}>
        <Title title="Registrar" className={classes.title} />
      </div>
    </div>
    <div>
      <div style={
        {
          fontSize: "1.2rem",
          textAlign: "left",
          marginBottom: "8px",
          color: "gray",
        }
      }>
        Foto de perfil:
      </div>
      <input
        type="file"
        onChange={handleImageChange}
        accept="image/*"
        style={{ fontSize: "1.6rem", marginBottom: "20px" }}
      />
      <CustomTextField
        label="Email"
        placeholder="Email"
        onChange={value => handleTextFieldChange(value, "email")}
        type="email"
        required
      />
      <div style={{ paddingTop: 20 }}></div>
      <CustomTextField
        label="Senha"
        placeholder="Senha"
        onChange={value => handleTextFieldChange(value, "password")}
        type="password"
        required
      />
      <div style={{ paddingTop: 20 }}></div>
      <CustomTextField
        label="Confirme Senha"

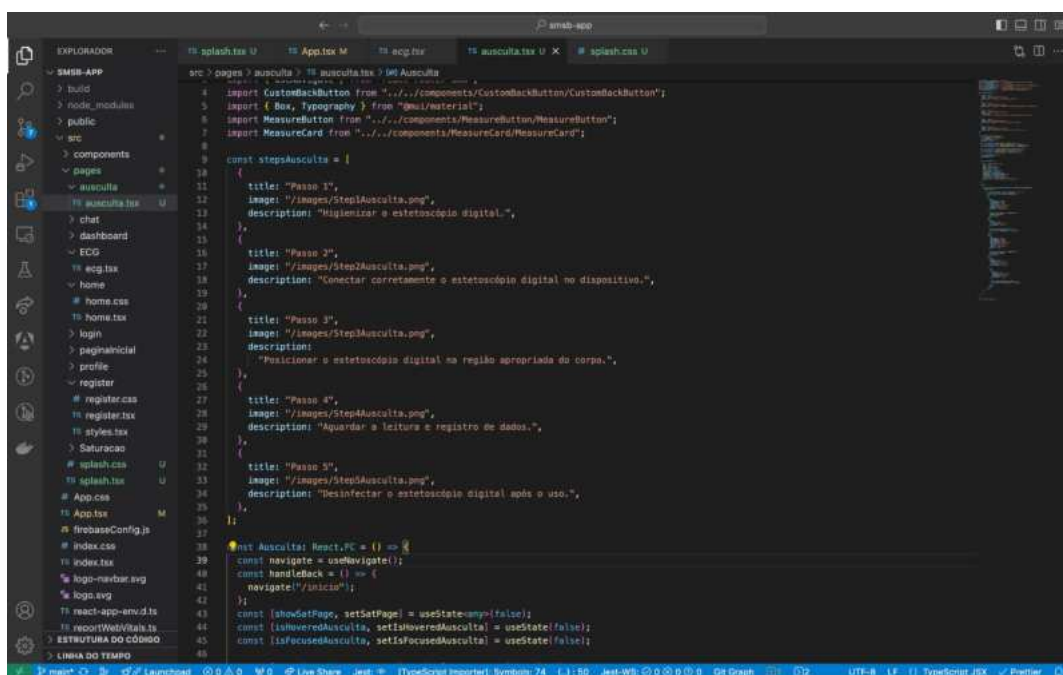
```

Fonte: Autor.

.1.2 Código das telas de auscultação pulmonar

As figuras abaixo apresentam as configurações das telas para o monitoramento da auscultação pulmonar.

Figura 50 – Página de Ausculta pulmonar 1.



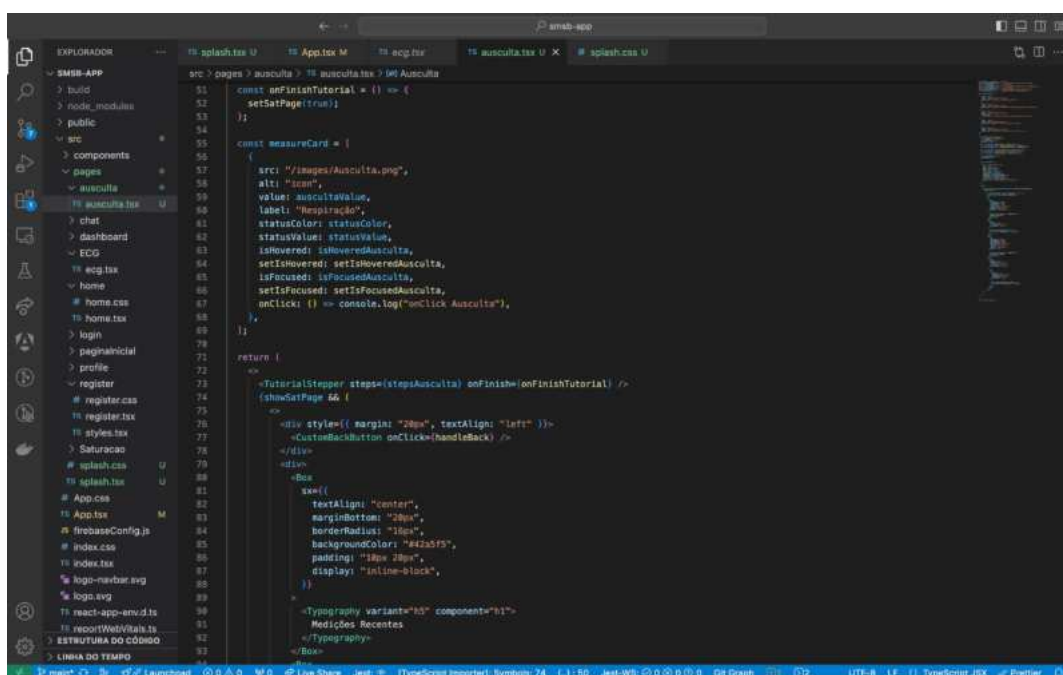
```
src > pages > ausculta > ausculta.tsx | @Ausculta
import CustomBackButton from "../../components/CustomBackButton/CustomBackButton";
import { Box, Typography } from "@mui/material";
import MeasureButton from "../../components/MeasureButton/MeasureButton";
import MeasureCard from "../../components/MeasureCard/MeasureCard";

const stepsAusculta = [
  {
    title: "Passo 1",
    image: "/images/Step1Ausculta.png",
    description: "Higienizar o estetoscópio digital.",
  },
  {
    title: "Passo 2",
    image: "/images/Step2Ausculta.png",
    description: "Conectar corretamente o estetoscópio digital no dispositivo.",
  },
  {
    title: "Passo 3",
    image: "/images/Step3Ausculta.png",
    description: "Posicionar o estetoscópio digital na região apropriada do corpo.",
  },
  {
    title: "Passo 4",
    image: "/images/Step4Ausculta.png",
    description: "Apurar a leitura e registro de dados.",
  },
  {
    title: "Passo 5",
    image: "/images/Step5Ausculta.png",
    description: "Desinfectar o estetoscópio digital após o uso.",
  }
];

const Ausculta: React.FC = () => {
  const navigate = useNavigate();
  const handleBack = () => {
    navigate("/inicio");
  };
  const [showSatPage, setShowSatPage] = useState<any>(false);
  const [isOveredAusculta, setIsOveredAusculta] = useState<false>;
  const [isFocusedAusculta, setIsFocusedAusculta] = useState<false>;
};
```

Fonte: Autor.

Figura 51 – Página de Ausculta pulmonar 2.



```
src > pages > ausculta > ausculta.tsx | @Ausculta
const onFinishTutorial = () => {
  setShowSatPage(true);
};

const measureCard = [
  {
    src: "/images/Ausculta.png",
    alt: "Ausculta",
    value: auscultavalue,
    label: "Respiração",
    statusColor: statusColor,
    statusValue: statusValue,
    isOvered: isOveredAusculta,
    setIsOvered: setIsOveredAusculta,
    isFocused: isFocusedAusculta,
    setIsFocused: setIsFocusedAusculta,
    onClick: () => console.log("onClick Ausculta"),
  },
];

return (
  <TutorialStepper steps={stepsAusculta} onFinish={onFinishTutorial} />
  <showSatPage {showSatPage} {
    <div style={{ margin: "20px", textAlign: "left" }}>
      <CustomBackButton onClick={handleBack} />
    </div>
    <div>
      <Box>
        <TextAlign="center",
        marginBottom: "20px",
        borderRadius: "15px",
        backgroundColor: "#42a5f5",
        padding: "10px 20px",
        display: "inline-block",
      />
      <Typography variant="h3" component="h1">
        Medições Recentes
      </Typography>
    </Box>
  } />
);
```

Fonte: Autor.

Figura 53 – Código utilizado para filtragem do sinal 1.

```

1 #include "stm32f4xx_hal.h"
2 #include "arm_math.h" // Biblioteca CMSIS-DSP para processamento de sinais
3
4 // Configurações do filtro passa-banda IIR
5 #define NUM_TAPS 5 // Número de coeficientes do filtro
6 float32_t iirCoeffs[NUM_TAPS] = {0.00078033, 0.0, -0.00156065, 0.0, 0.00078033}; // Numerador (b)
7 float32_t iirState[NUM_TAPS]; // Estado do filtro IIR
8 arm_biquad_casd_df1_inst_f32 S; // Estrutura do filtro
9
10 // Buffers para captura e processamento dos dados
11 #define BUFFER_SIZE 256
12 int16_t audioBuffer[BUFFER_SIZE]; // Buffer para capturar os dados via I2S
13 float32_t inputBuffer[BUFFER_SIZE]; // Buffer de entrada (convertido para float)
14 float32_t outputBuffer[BUFFER_SIZE]; // Buffer de saída (sinais filtrados)
15
16 // Inicialização do I2S para o sensor INMP441
17 void I2S_Init(void) {
18     hi2s1.Init.AudioFreq = I2S_AUDIOFREQ_44100; // Frequência de amostragem de 44.1 kHz
19     hi2s1.Init.DataFormat = I2S_DATAFORMAT_16B;
20     hi2s1.Init.CPOL = I2S_CPOL_LOW;
21     hi2s1.Init.Mode = I2S_MODE_MASTER_RX; // Modo mestre, recebendo dados
22     HAL_I2S_Init(&hi2s1);
23 }
24
25 // Configuração do I2C para comunicação com ESP32
26 void I2C_Init(void) {
27     hi2c1.Init.ClockSpeed = 100000; // Velocidade de 100 kHz
28     hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
29     hi2c1.Init.OwnAddress1 = 0x52; // Endereço do STM32
30     hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
31     HAL_I2C_Init(&hi2c1);
32 }
33
34 // Função para capturar dados via I2S
35 void I2S_Capture(void) {
36     HAL_I2S_Receive(&hi2s1, (uint16_t*)audioBuffer, BUFFER_SIZE, HAL_MAX_DELAY);

```

Fonte: Autor.

Figura 54 – Código utilizado para filtragem do sinal 2.

```

39 // Função para converter os dados do buffer de int16_t para float32_t
40 void ConvertToFloat(int16_t *src, float32_t *dest, uint16_t size) {
41     for (uint16_t i = 0; i < size; i++) {
42         dest[i] = (float32_t)src[i] / 32768.0f; // Normaliza os valores entre -1 e 1
43     }
44 }
45
46 // Função para aplicar o filtro IIR nos dados capturados
47 void ApplyIIRFilter(float32_t *input, float32_t *output, uint16_t size) {
48     arm_biquad_cascade_df1_f32(&S, input, output, size); // Aplica o filtro nos dados
49 }
50
51 // Função para inicializar o filtro IIR
52 void InitIIRFilter(void) {
53     arm_biquad_cascade_df1_init_f32(&S, 1, iirCoeffs, iirState); // Inicializa o filtro
54 }
55
56 // Função para enviar dados filtrados via I2C para o ESP32
57 void SendDataI2C(float32_t *data, uint16_t size) {
58     uint8_t buffer[BUFFER_SIZE * sizeof(float32_t)];
59     for (uint16_t i = 0; i < size; i++) {
60         memcpy(&buffer[i * sizeof(float32_t)], &data[i], sizeof(float32_t));
61     }
62     HAL_I2C_Master_Transmit(&hi2c1, (ESP32_I2C_ADDRESS << 1), buffer, size * sizeof(float32_t), HAL_MAX_DELAY);
63 }
64
65 int main(void) {
66     HAL_Init(); // Inicializa a HAL
67     SystemClock_Config(); // Configuração do clock do sistema
68     I2S_Init(); // Inicializa o I2S
69     I2C_Init(); // Inicializa o I2C
70     InitIIRFilter(); // Inicializa o filtro IIR
71
72     while (1) {
73         // Captura os dados de áudio do sensor via I2S

```

Fonte: Autor.

Figura 55 – Código utilizado para filtragem do sinal 3.

```
64
65 int main(void) {
66     HAL_Init(); // Inicializa a HAL
67     SystemClock_Config(); // Configuração do clock do sistema
68     I2S_Init(); // Inicializa o I2S
69     I2C_Init(); // Inicializa o I2C
70     InitIIRFilter(); // Inicializa o filtro IIR
71
72     while (1) {
73         // Captura os dados de áudio do sensor via I2S
74         I2S_Capture();
75
76         // Converte os dados capturados para float
77         ConvertToFloat(audioBuffer, inputBuffer, BUFFER_SIZE);
78
79         // Aplica o filtro IIR nos dados capturados
80         ApplyIIRFilter(inputBuffer, outputBuffer, BUFFER_SIZE);
81
82         // Envia os dados filtrados via I2C para o ESP32
83         SendDataI2C(outputBuffer, BUFFER_SIZE);
84     }
85 }
```

Fonte: Autor.