

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

DIANA MACEDO RODRIGUES

**SISTEMA DE AQUISIÇÃO DE DADOS PARA ANÁLISE DE APNEIA
DO SONO**

FLORIANÓPOLIS, 2026.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA**

DIANA MACEDO RODRIGUES

**SISTEMA DE AQUISIÇÃO DE DADOS PARA ANÁLISE DE APNEIA
DO SONO**

Trabalho de Conclusão de Curso submetido
ao Instituto Federal de Educação, Ciência e
Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Bacharel em Engenharia Eletrônica.

Orientador:
Prof. Carlos Gontarski Speranza, Dr.Eng.

Coorientador:
Prof. Pedro Giassi Junior, Dr.Eng.

FLORIANÓPOLIS, 2026.

Ficha de identificação da obra elaborada pelo autor.

Rodrigues, Diana
Sistema de Aquisição de Dados para Análise de Apneia do Sono. / Diana Rodrigues; orientação de Carlos Gontarski Speranza ; coorientação de Pedro Giassi Junior.
- Florianópolis, SC, 2026.
98 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Eletrônica. Departamento Acadêmico de Eletrônica.
Inclui Referências.

1. Apneia Obstrutiva do Sono.. 2. Aquisição de dados fisiológicos.. 3. Sitemas embarcados.. 4. Internet das Coisas.. 5. Processamento Digital de Sinais.. I. Gontarski Speranza , Carlos. II. Giassi Junior, Pedro.
III. Instituto Federal de Santa Catarina. IV. Sistema de Aquisição de Dados para Análise de Apneia do Sono..

SISTEMA DE AQUISIÇÃO DE DADOS PARA ANÁLISE DE APNEIA DO SONO

DIANA MACEDO RODRIGUES

Este trabalho foi julgado adequado para obtenção do título de Bacharel em Engenharia Eletrônica e aprovado na sua forma final pela banca examinadora do Curso Engenharia Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 06 de agosto, 2026.

Banca Examinadora:

Documento assinado digitalmente
 **CARLOS GONTARSKI SPERANZA**
Data: 12/08/2026 14:10:36-0300
Verifique em <https://validar.iti.gov.br>

Carlos Gontarski Speranza, Dr.Eng.

Documento assinado digitalmente
 **PEDRO GIASSI JUNIOR**
Data: 12/08/2026 14:42:12-0300
Verifique em <https://validar.iti.gov.br>

Pedro Giassi Junior, Dr.Eng.
Instituto Federal de Santa Catarina - IFSC

Documento assinado digitalmente
 **DANIEL LOHMANN**
Data: 12/08/2026 17:07:46-0300
Verifique em <https://validar.iti.gov.br>

Daniel Lohmann, Me.
Instituto Federal de Santa Catarina - IFSC

Documento assinado digitalmente
 **HUGO MARCONDES**
Data: 13/08/2026 14:10:05-0300
Verifique em <https://validar.iti.gov.br>

Hugo Marcondes, Me.
Instituto Federal de Santa Catarina - IFSC

Dedico este trabalho aos meus pais,
que sempre foram meu alicerce e os maiores
apoiadores da minha jornada e dos meus sonhos.

AGRADECIMENTOS

Agradeço primeiramente a Deus que me deu forças neste período, agradeço aos meus pais Vera e Edemilson que em todos os momentos estiveram ao meu lado dando apoio e me dando forças para continuar este sonho. Ao meu irmão Henrique que ouviu diversas vezes meus desabafos e me deu diversos conselhos apesar de mais novo.

Aos meus amigos Eduardo, Giovanna e Thaine que desde o início do curso me acompanham, tanto na graduação quanto na vida e que foram essenciais para eu não desistir dessa jornada. À minha melhor amiga Gabriela, que, mesmo sendo de outro curso, sempre me ouviu com paciência e esteve presente nos momentos difíceis.

Às minhas duplas que vieram ao longo da graduação e se tornaram grandes amigos, João, Rafael e Sarah vocês me apoiaram em matérias que nem eu mesma acreditava passar e juntos fomos mais fortes para chegar até aqui.

Aos meus orientadores Carlos Gontarski Speranza e Pedro Giassi Junior, por toda a dedicação, paciência e estímulo que me deram ao longo desses dois últimos semestres, vocês são pessoas das quais admiro e quero seguir como inspiração acadêmica.

Ao Instituto Federal de Santa Catarina (IFSC) e a todo o Departamento de Eletrônica (DAELN) por toda a estrutura, professores extremamente qualificados, ensino de qualidade e ambiente que ensina não somente na área técnica como no âmbito humano também.

“Renda-se, como eu me rendi.
Mergulhe no que você não
conhece como eu mergulhei.”
- Clarice Lispector

RESUMO

A apneia obstrutiva do sono (AOS) é um distúrbio respiratório caracterizado por episódios recorrentes de obstrução parcial ou completa das vias aéreas superiores durante o sono, ocasionando alterações em parâmetros fisiológicos que podem auxiliar na identificação desses eventos. Considerando a complexidade da aquisição direta dos sinais respiratórios, este trabalho apresenta o desenvolvimento de um sistema de aquisição de dados baseado em sistemas embarcados e Internet das Coisas (IoT), destinado à obtenção de parâmetros fisiológicos relacionados à AOS por meio de métodos indiretos. O sistema foi desenvolvido utilizando o microcontrolador ESP32, integrado aos sensores MAX30100, MPU6050 e INMP441 para aquisição da saturação de oxigênio, frequência cardíaca, posição corporal e sinais sonoros associados ao ronco. O gerenciamento das tarefas foi realizado por meio do sistema operacional FreeRTOS, enquanto o processamento do sinal sonoro empregou a Transformada Rápida de Fourier (FFT) para análise das frequências características do ronco. Os dados adquiridos foram processados localmente e transmitidos periodicamente para a plataforma ThingSpeak utilizando a funcionalidade Bulk Write, possibilitando seu armazenamento em nuvem. Os resultados obtidos demonstraram o funcionamento integrado do sistema, permitindo a aquisição contínua, o processamento e o armazenamento dos parâmetros fisiológicos analisados. Dessa forma, o sistema desenvolvido constitui uma plataforma para aquisição de dados relacionados à apneia obstrutiva do sono, podendo servir como base para estudos futuros voltados à análise desses sinais.

Palavras-chave: Apneia Obstrutiva do Sono. Aquisição de dados fisiológicos. Sistemas embarcados. Internet das Coisas. Processamento Digital de Sinais.

ABSTRACT

Obstructive Sleep Apnea (OSA) is a respiratory disorder characterized by recurrent episodes of partial or complete obstruction of the upper airway during sleep, resulting in changes in physiological parameters that may assist in the identification of these events. Considering the complexity of direct respiratory signal acquisition, this work presents the development of a data acquisition system based on embedded systems and the Internet of Things (IoT), designed to obtain physiological parameters related to OSA through indirect methods. The system was developed using the ESP32 microcontroller integrated with MAX30100, MPU6050, and INMP441 sensors to acquire oxygen saturation, heart rate, body position, and snoring-related sound signals. Task management was performed using the FreeRTOS real-time operating system, while sound signal processing employed the Fast Fourier Transform (FFT) to analyze the frequency range associated with snoring. The acquired data were processed locally and periodically transmitted via Wi-Fi to the ThingSpeak cloud platform using the Bulk Write functionality. The results demonstrated the integrated operation of the system, enabling continuous acquisition, processing, and cloud storage of the analyzed physiological parameters. Therefore, the proposed system provides a platform for acquiring data related to obstructive sleep apnea and may support future studies involving the analysis of these physiological signals.

Keywords: Obstructive Sleep Apnea. Physiological Data Acquisition. Embedded Systems. Internet of Things. Digital Signal Processing.

LISTA DE FIGURAS

Figura 1: Modelo mecânico clássico de um acelerômetro como sistema mola-massa-amortecedor.....	31
Figura 2: Representação vetorial do Efeito Coriolis aplicado a uma massa vibratória em um giroscópio.....	32
Figura 3 : Diagrama da interface de Comunicação I ² C.....	34
Figura 4 : Diagrama da interface de Comunicação I ² S.....	35
Figura 5: Funcionamento da plataforma.....	37
Figura 6: Diagrama geral do sistema.....	40
Figura 7: ESP-32S	42
Figura 8: Diagrama de ligação do sistema	42
Figura 9: Diagrama temporal da comunicação I ² C entre o ESP32 e o sensor MAX30100	44
Figura 10: Módulo MAX30100 utilizado.....	44
Figura 11: Diagrama temporal da comunicação I ² C entre o ESP32 e o sensor MPU6050. Fonte: A autora (2026).....	45
Figura 12: Módulo MPU6050 utilizado.....	45
Figura 13: Diagrama temporal da comunicação I ² S entre o ESP32 e o microfone INMP441	46
Figura 14: Módulo INMP441 utilizado.....	46
Figura 15 - Trecho do código referente a Ratio-Of-Ratios	51
Figura 16 - Fluxograma de aquisição e processamento MAX30100.....	52
Figura 17 - Trecho do código referente a posição.....	54
Figura 18: Fluxograma de aquisição e processamento MPU6050	55
Figura 19 - Trecho do código responsável pelas funções do microfone.....	59
Figura 20: Fluxograma de aquisição e processamento INMP441	60
Figura 21: Organização dos canais e campos da plataforma ThingSpeak	61
Figura 22 - Fluxograma da plataforma ThingSpeak.....	63
Figura 23: Protótipo físico em funcionamento	65
Figura 24: Captura do monitor serial correspondente ao instante do teste.....	65
Figura 25 - Trecho responsável pelo armazenamento das amostras no buffer temporário.....	66
Figura 26: Visualização em tempo real dos valores de saturação de oxigênio armazenados na plataforma ThingSpeak	69
Figura 27: Visualização em tempo real dos valores de frequência cardíaca armazenados na plataforma ThingSpeak	69

Figura 28: Visualização em tempo real (teste de 15min) dos valores de frequência cardíaca e saturação de oxigênio utilizando a ferramenta MATLAB Analysis do ThingSpeak.....	70
Figura 29: Visualização em tempo real (teste de 1h) dos valores de frequência cardíaca e saturação de oxigênio utilizando a ferramenta MATLAB Analysis do ThingSpeak.....	71
Figura 30: Visualização em tempo real dos valores da posição armazenados na plataforma ThingSpeak	73
Figura 31: Visualização em tempo real (teste de 15min) das posições utilizando a ferramenta MATLAB Analysis do ThingSpeak	75
Figura 32 - Posição teste barriga para cima	75
Figura 33: Visualização em tempo real (teste de 1h) das posições utilizando a ferramenta MATLAB Analysis do ThingSpeak	76
Figura 34 - Posição teste virado de lado	76
Figura 35 - Posição teste sentado.....	77
Figura 36: Ferramenta utilizada para geração de sinais senoidais durante os testes experimentais.....	78
Figura 37 - Trecho do código responsável pelo armazenamento temporário INMP441	79
Figura 38: Visualização em tempo real dos valores da energia armazenados na plataforma ThingSpeak	81
Figura 39: Visualização em tempo real dos valores do som (ronco) armazenados na plataforma ThingSpeak.....	82
Figura 40: Visualização em tempo real (teste de 15min) da energia e detecção de ronco utilizando a ferramenta MATLAB Analysis do ThingSpeak.....	83
Figura 41: Visualização em tempo real (teste de 1h) da energia e detecção de ronco utilizando a ferramenta MATLAB Analysis do ThingSpeak.....	83
Figura 42: Monitor serial durante os testes com maior quantidade de amostras	84
Figura 43: Modelo de bateria utilizada.....	85
Figura 44: Modelo do módulo BMS utilizado.....	86
Figura 45: Regulador de tensão LM2596 Step Down	86
Figura 46: Sistema de alimentação montado	87

LISTA DE QUADROS

Quadro 1 - Classificação da gravidade da AOS.....	24
---	----

LISTA DE TABELAS

Tabela 1 - Classificação adotada para as posições corporais.....	54
Tabela 2 - Comparação protótipo X oxímetro comercial	66

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i> (Conversor Analógico-Digital)
AOS	Apneia Obstrutiva do Sono
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
BCLK	<i>Bit Clock</i>
BMS	<i>Battery Management System</i> (Placa de Sistema de Gerenciamento de Bateria)
BPM	<i>Batimentos Por Minuto</i> (Frequência Cardíaca)
CC	Corrente Contínua
CoAP	<i>Constrained Application Protocol</i>
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
DAELN	Departamento Acadêmico de Eletrônica
DC	<i>Direct Current</i> (Componente Contínua)
DFT	<i>Discrete Fourier Transform</i> (Transformada Discreta de Fourier)
ESP-IDF	<i>Espressif IoT Development Framework</i>
FFT	<i>Fast Fourier Transform</i> (Transformada Rápida de Fourier)
FIFO	<i>First In, First Out</i> (Estrutura/Memória de Buffer de Periféricos)
FreeRTOS	<i>Real-Time Operating System</i> (Sistema Operacional de Tempo Real Livre)
GPIO	<i>General Purpose Input/Output</i> (Portas de Entrada e Saída Digital)
HI	Hipoxemia Intermitente
HTTP	<i>Hypertext Transfer Protocol</i>
I ² C	<i>Inter-Integrated Circuit</i>
I ² S	<i>Inter-IC Sound</i>

IAH	Índice de Apneia-Hipopneia
IFSC	Instituto Federal de Santa Catarina
IoT	<i>Internet of Things</i> (Internet das Coisas)
JSON	<i>JavaScript Object Notation</i>
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
MEMS	<i>Micro-Electro-Mechanical Systems</i> (Sistemas Microeletromecânicos)
MQTT	<i>Message Queuing Telemetry Transport</i>
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
SCK	<i>Serial Clock</i> (Interface I ² S)
SCL	<i>Serial Clock</i> (Interface I ² C)
SD	<i>Serial Data</i> (Interface I ² S)
SDA	<i>Serial Data</i> (Interface I ² C)
SoC	<i>System-on-Chip</i>
SpO ₂	Saturação Periférica de Oxigênio
TCC	Trabalho de Conclusão de Curso
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
USB	<i>Universal Serial Bus</i>
Wi-Fi	<i>Wireless Fidelity</i>
WS	<i>Word Select</i>

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Justificativa	19
1.2	Definição do Problema	20
1.3	Objetivo Geral	20
1.4	Objetivos Específicos	21
1.5	Estrutura do Trabalho	21
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Apneia Obstrutiva do sono	23
2.1.1	Definição e Caracterização da Apneia Obstrutiva do sono	23
2.1.2	Dessaturação de oxigênio	24
2.1.3	Alterações da frequência cardíaca	25
2.1.4	Ronco como indicador acústico da AOS	25
2.2	Internet das Coisas	26
2.2.1	Conceitos fundamentais de IoT	26
2.2.2	Monitoramento Remoto	28
2.3	Componentes utilizados no sistema	28
2.3.1	Microcontrolador	28
2.3.2	Sensor de oximetria de pulso e batimento cardíaco	30
2.3.3	Acelerômetro	31
2.3.4	Microfone MEMS	33
2.3.5	Protocolo de Comunicação I ² C	33
2.3.6	Protocolo de Comunicação I ² S	34
2.3.7	Sistema Operacional FreeRTOS	35
2.3.8	ThingSpeak	36
3	METODOLOGIA	40
3.1	Métodos aplicados	40
3.2	Desenvolvimento do Hardware	41
3.2.1	Microcontrolador ESP32	41
3.2.2	Sensor MAX30100	43
3.2.3	Sensor MPU6050	44
3.2.4	Sensor INMP441	45
3.3	Desenvolvimento do Software	48
3.3.1	Estrutura Geral do Software	48
3.3.2	Aquisição e Processamento dos Dados MAX30100	49
3.3.3	Aquisição e Processamento dos Dados MPU6050	52
3.3.4	Aquisição e Processamento dos Dados INMP441	56
3.3.5	Implementação da Plataforma ThingSpeak	60
4	APRESENTAÇÃO DOS RESULTADOS	64
4.1	Avaliação do Monitoramento da Frequência Cardíaca e Saturação de Oxigênio	64
4.1.1	Validação experimental	64
4.1.2	Armazenamento e envio	66
4.1.3	Monitoramento contínuo	68
4.2	Avaliação do Monitoramento da Posição Corporal	72
4.2.1	Monitoramento contínuo	73
4.3	Avaliação do Monitoramento do Ronco	77

4.3.1	Testes experimentais.....	78
4.3.2	Armazenamento e transmissão dos dados.....	79
4.3.3	Monitoramento contínuo.....	80
4.3.4	Avaliação da comunicação com a plataforma	84
4.4	Fonte de alimentação.....	85
5	CONSIDERAÇÕES FINAIS.....	89
5.1	Sugestões para trabalhos futuros.....	90
	REFERÊNCIAS	92
	APÊNDICE A - Repositório digital de códigos-fontes do sistema.....	97

1 INTRODUÇÃO

A apneia obstrutiva do sono (AOS) é um distúrbio que se caracteriza por pausas de no mínimo 10 segundos na respiração durante o período de sono, conforme estabelecido por Berry et al. (2012). No cenário global, estima-se que mundialmente cerca de 936 milhões de pessoas entre 30 e 69 anos sejam afetadas por esta condição (BENJAFIELD *et al.*, 2019). Demonstrando que este é um problema de saúde pública que está em grande crescimento. (BENJAFIELD *et al.*, 2019).

Um sono desregulado pode afetar de forma significativa todo o metabolismo humano, aumentando as chances de doenças cardíacas, diabetes, obesidade, entre outras condições sérias (WORLEY, 2018). Além dos fatores físicos, o sono instável interfere também em outros âmbitos, existem estudos que correlacionam este problema a diversas alterações hormonais e psicológicas que podem por vezes diminuir a qualidade de vida da pessoa. (CABELLINO *et al.*, 2025).

A identificação dos eventos respiratórios é considerada uma das principais etapas para o diagnóstico da apneia obstrutiva do sono. Entretanto, a aquisição e a análise direta desses sinais constituem um desafio técnico, uma vez que requerem a utilização simultânea de diferentes sensores para monitorar o fluxo aéreo e o esforço respiratório, como cânulas nasais, termistores, pneumotacógrafos e cintas torácicas e abdominais (BERRY et al., 2018; EUROPEAN RESPIRATORY SOCIETY, 2023).

Além da necessidade de múltiplos dispositivos, fatores como deslocamento dos sensores durante o sono, variações na qualidade dos sinais e limitações inerentes a cada tecnologia podem comprometer a confiabilidade das medições. Em razão dessas dificuldades, diversos estudos têm investigado métodos indiretos para aquisição de informações relacionadas aos eventos respiratórios, utilizando parâmetros fisiológicos como saturação de oxigênio, frequência cardíaca, sons respiratórios associados ao ronco e posição corporal, os quais apresentam forte correlação com a ocorrência dos episódios de apneia (PEVERNAGIE; AARTS; DE MEYER, 2010; BERRY et al., 2018).

Sabendo dos problemas advindos da AOS, exames preliminares para a identificação desta condição são altamente indicados pelos médicos especialistas (Jonas *et al.*, 2017). Dentre eles, destaca-se a polissonografia, considerada o exame

padrão-ouro para o diagnóstico dos distúrbios do sono. Esse exame consiste no monitoramento simultâneo de diversos parâmetros fisiológicos, como movimentos oculares, atividade muscular, frequência cardíaca, fluxo respiratório e saturação de oxigênio durante o período de sono (KAPUR *et al.*, 2017).

Todavia, trata-se de um exame de alto custo, tradicionalmente realizado na própria clínica, onde o paciente precisa passar a noite sob cuidados e monitoramento (KAPUR *et al.*, 2017). Atualmente, existe a possibilidade de fazer um exame em ambiente domiciliar seguindo o mesmo princípio da polissonografia tradicional, porém de uma forma mais simplificada (KAPUR *et al.*, 2017). Ainda assim este método requer um investimento alto e possui algumas limitações quanto ao armazenamento dos dados, sem integração com monitoramento remoto ou armazenamento em nuvem.

Diante desta problemática, verifica-se a necessidade do desenvolvimento de sistemas capazes de realizar a aquisição integrada de parâmetros fisiológicos relacionados à apneia obstrutiva do sono por meio de métodos indiretos. A utilização de sinais como saturação de oxigênio, frequência cardíaca, ronco e posição corporal possibilita a obtenção contínua de informações relevantes para futuras avaliações médicas, além de permitir o armazenamento dessas informações em plataformas em nuvem.

1.1 Justificativa

A Apneia Obstrutiva do Sono (AOS) é um distúrbio de elevada ocorrência na população e cujo diagnóstico ainda apresenta certa complexidade. Os exames utilizados para sua identificação possuem custo elevado e, em muitos casos, podem ser desconfortáveis devido à necessidade de permanência do paciente em ambiente clínico para sua realização. Mesmo quando realizados em ambiente domiciliar, ainda apresentam limitações que devem ser consideradas (KAPUR *et al.*, 2017).

Sob essa perspectiva e diante de um avanço geral nas tecnologias aplicadas à área da saúde, tornou-se útil o desenvolvimento de sistemas contendo sensores e microcontroladores capazes de adquirir dados e facilitar as interações com

os médicos para a identificação da apneia do sono e um acompanhamento mais preciso das condições do paciente.

Embora a polissonografia seja considerada o exame padrão-ouro para diagnóstico da AOS, sua realização pode apresentar limitações relacionadas ao custo, à infraestrutura necessária e à disponibilidade do exame. Nesse contexto, torna-se relevante o desenvolvimento de sistemas capazes de monitorar parâmetros fisiológicos que se alteram durante os episódios de AOS, permitindo a obtenção de informações de forma indireta sobre a ocorrência desses eventos.

Sistemas baseados em sensores e microcontroladores possibilitam a aquisição contínua de parâmetros como saturação de oxigênio, frequência cardíaca, sons respiratórios e posição corporal. Embora esses sinais não permitam diagnosticar diretamente a AOS, sua análise conjunta fornece informações relevantes sobre as alterações fisiológicas associadas ao distúrbio, podendo auxiliar no monitoramento e acompanhamento dos pacientes.

Quando integrados aos conceitos da Internet das Coisas (IoT), esses sistemas permitem que os dados adquiridos sejam transmitidos e armazenados em plataformas na nuvem, possibilitando o acesso remoto às informações por profissionais da saúde e favorecendo o acompanhamento contínuo do paciente.

1.2 Definição do Problema

Como um sistema embarcado baseado no ESP32 pode adquirir, processar e disponibilizar dados de saturação de oxigênio, frequência cardíaca, posição corporal e sinais acústicos relacionados à Apneia Obstrutiva do Sono?

1.3 Objetivo Geral

Implementar um sistema funcional de aquisição de dados utilizando um microcontrolador para adquirir, processar e transmitir à nuvem informações de saturação de oxigênio, frequência cardíaca, ronco e posição corporal relacionadas à apneia obstrutiva do sono.

1.4 Objetivos Específicos

- a) Implementar o sistema de aquisição de dados utilizando o microcontrolador e os sensores.
- b) Validar o funcionamento da aplicação analisando as leituras obtidas pelos sensores;
- c) Integrar os dados obtidos pelos sensores a uma plataforma de computação em nuvem por meio da tecnologia IoT;
- d) Implementar o armazenamento e a visualização das informações em uma plataforma em nuvem.

1.5 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos. No primeiro capítulo, são apresentados a contextualização do tema, a definição do problema de pesquisa, a justificativa para o desenvolvimento do trabalho, bem como o objetivo geral e objetivos específicos.

O segundo capítulo aborda a fundamentação teórica necessária para a compreensão do trabalho, apresentando conceitos relacionados à Apneia Obstrutiva do Sono, aos principais parâmetros fisiológicos associados ao distúrbio, às tecnologias de Internet das Coisas (IoT) empregadas na aquisição de dados, além das características dos sensores e demais componentes eletrônicos utilizados no desenvolvimento do sistema.

O terceiro capítulo descreve a metodologia adotada, contemplando os materiais e equipamentos utilizados, a arquitetura do sistema desenvolvido, o desenvolvimento do hardware e do software, bem como os algoritmos implementados para aquisição, processamento e transmissão dos dados.

O quarto capítulo apresenta os resultados obtidos durante os testes experimentais realizados com o protótipo desenvolvido, incluindo a validação dos dados adquiridos pelos sensores, a análise do desempenho do sistema implementado, a transmissão das informações para a plataforma de nuvem, a

avaliação da autonomia do sistema de alimentação e a discussão dos resultados obtidos.

Por fim, o quinto capítulo apresenta as considerações finais do trabalho, destacando as principais contribuições do sistema desenvolvido, suas limitações e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os fundamentos teóricos necessários para o desenvolvimento deste trabalho. Inicialmente, é apresentada uma revisão sobre a Apneia Obstrutiva do Sono (AOS), abordando sua definição, suas características fisiológicas e os principais sinais biomédicos relacionados ao distúrbio. Na sequência, são discutidas as técnicas de monitoramento utilizadas para sua identificação, destacando-se os parâmetros utilizados neste projeto. Ao final, são descritos os componentes e as tecnologias que compõem o sistema desenvolvido.

Posteriormente são apresentados os conceitos relacionados a Internet das Coisas (IoT) aplicadas a área da saúde, fazendo destaque aos sistemas para aquisição, transmissão e armazenamento dos dados adquiridos. E por fim, serão descritos os principais componentes utilizados no desenvolvimento deste sistema, seu funcionamento e suas características.

2.1 Apneia Obstrutiva do sono

2.1.1 Definição e Caracterização da Apneia Obstrutiva do sono

A Apneia Obstrutiva do sono é um distúrbio respiratório que consiste em uma obstrução parcial ou completa das vias aéreas superiores, dificultando a passagem de ar. Essa obstrução ocorre principalmente devido a um relaxamento muscular da faringe, fazendo com que o fluxo de ar para os pulmões seja reduzido ou interrompido brevemente. Este problema consiste em pausas anormais da respiração durante o período do sono. Consistindo num comprometimento da qualidade do sono e afetando a oxigenação sanguínea. (HO; BRASS, 2011; SANKRI-TARBICHI, 2012).

Esses eventos podem ser classificados em dois tipos: a apneia e a hipopneia. A apneia é considerada aquela na qual o fluxo de ar é cessado por completo enquanto na hipopneia o fluxo é comprometido apenas de forma parcial. Em ambos os casos é considerado o tempo mínimo de pausa de 10 segundos. Além disso, na hipopneia também é comum um microdespertar cerebral e uma dessaturação específica (BERRY et al., 2018; KAPUR et al., 2017).

Para avaliar a gravidade da apneia é utilizado o Índice de Apneia-Hipopneia (IAH). Que consiste na verificação da média de eventos de apneia e hipopneia que ocorrem durante cada hora de sono, permitindo determinar a frequência em que essas pausas estão presentes durante o período de repouso do paciente avaliado (BERRY et al., 2018; HUSSEIN et al., 2024).

Com base nos valores obtidos para o IAH, é possível classificar a gravidade da AOS em quatro categorias: normal, caracterizada pela ausência ou ocorrência de menos de cinco eventos respiratórios por hora de sono ($IAH < 5$), leve, moderada e grave. Este processo é importante para auxiliar os profissionais da saúde, para que posteriormente cada avaliação clínica seja executada da forma mais adequada durante o tratamento (HUSSEIN et al., 2024; KAPUR et al., 2017).

Quadro 1 - Classificação da gravidade da AOS

Classificação	IAH
Normal	Menor que 5
Leve	Entre 5 e 14
Moderada	Entre 15 e 29
Grave	Acima de 30

Fonte: Adaptado de Ho e Brass (2011, p. 66) e Hussein et al. (2024, p. 61).

No quadro 1 é possível observar os valores de referência para a classificação da gravidade da AOS.

2.1.2 Dessaturação de oxigênio

A dessaturação de oxigênio corresponde à diminuição da saturação de oxigênio no sangue arterial. Em termos fisiológicos, isso significa que uma quantidade menor de hemoglobina está transportando oxigênio, resultando na redução da oxigenação dos tecidos (SCHLOTTHAUER et al., 2018). No caso da AOS esse fenômeno está ligado ao fato de que na apneia e hipopneia ocorre uma diminuição do fluxo de ar nos pulmões o que leva a essa condição.

Devido a ocorrências recorrentes de dessaturação de oxigênio, o paciente pode desenvolver outros problemas de saúde - como a Hipoxemia intermitente (HI) que é caracterizada por ciclos de redução e recuperação da oxigenação sanguínea. A HI pode estar relacionada ao desenvolvimento de doenças cardiovasculares, como

hipertensão, arritmias e infarto, bem como distúrbios metabólicos, como resistência à insulina e diabetes (SEQUEIRA; BANDEIRA; AZEVEDO, 2019).

O monitoramento da saturação de oxigênio (SpO_2) é um importante parâmetro para a análise e identificação da apneia e hipopneia (EPSTEIN et al., 2009; SEQUEIRA; BANDEIRA; AZEVEDO, 2019). Sabendo que um dos sinais característicos desses distúrbios são episódios de redução da oxigenação sanguínea, o acompanhamento direto auxilia na identificação desses episódios e facilita a avaliação médica.

2.1.3 Alterações da frequência cardíaca

Além das alterações na oxigenação sanguínea, outro sinal frequentemente observado em pacientes com AOS corresponde às alterações na frequência cardíaca. Essas modificações decorrem das respostas fisiológicas desencadeadas durante os episódios de apneia e podem comprometer a integridade das funções neurocardíacas, bem como alterar a dinâmica do sistema nervoso autônomo (SEQUEIRA; BANDEIRA; AZEVEDO, 2019).

Durante os eventos de AOS, a diminuição da oxigenação faz com que o corpo entre em um estado de alerta. Nessa condição, ocorre a ativação do sistema nervoso simpático, que promove a elevação da frequência cardíaca como uma resposta fisiológica à redução da oxigenação sanguínea.

Tendo conhecimento de por que essas alterações ocorrem, torna-se necessário compreender o motivo pelo qual o paciente apresenta essas respostas fisiológicas. Dessa forma entende-se que ter um monitoramento preciso e contínuo para identificar problemas de saúde é essencial. Também é importante para verificar os efeitos causados por esse distúrbio.

2.1.4 Ronco como indicador acústico da AOS

O ronco é um sintoma comum da AOS e consiste na vibração das vias aéreas superiores. Isso ocorre durante a passagem de ar na faringe devido ao estreitamento dessa região (ALENCAR et al., 2013; CASALE et al., 2009). O fluxo de

ar nesse momento torna-se turbulento, e produzindo o som característico que conhecemos como ronco.

Embora esse sintoma não seja exclusivo de pacientes com AOS, é um sintoma frequentemente observado em pacientes com esse distúrbio. Dependendo do nível de obstrução, podem ocorrer alterações na intensidade e frequência dos sons, um indicativo de dificuldades respiratórias durante o sono.

Dessa forma, o ronco constitui um importante indicador da apneia obstrutiva do sono, pois modificações em suas características acústicas, como intensidade, frequência e padrão temporal, podem estar relacionadas aos episódios de obstrução das vias aéreas superiores. Nesse contexto, o monitoramento e a análise desses sinais sonoros podem contribuir para a identificação da ocorrência e da frequência desses eventos respiratórios durante o sono.

2.2 Internet das Coisas

2.2.1 Conceitos fundamentais de IoT

A Internet das Coisas (*Internet of Things* – IoT) é um conceito que descreve a conexão de objetos físicos à internet, permitindo que dispositivos colem, processem, compartilhem e recebam informações de forma autônoma. Esses dispositivos são equipados com sensores, atuadores, capacidade de processamento e interfaces de comunicação, permitindo a integração em uma rede capaz de monitorar o ambiente e executar ações sem a necessidade de interferência humana de forma contínua. Esse conceito representa uma evolução da internet tradicional, expandindo a comunicação entre pessoas para incluir também a interação entre objetos inteligentes (INTERNET OF THINGS WIKI, 2024).

A consolidação da Internet das Coisas foi possibilitada pelo avanço simultâneo das tecnologias de comunicação, dos sistemas embarcados e da computação em nuvem. A combinação dessas tecnologias permitiu o desenvolvimento de dispositivos cada vez menores, mais eficientes e mais conectados, capazes de adquirir informações do ambiente, processá-las localmente e disponibilizar remotamente para diferentes aplicações. Segundo Atzori, Iera e Morabito (2010), a IoT representa a ligação entre o mundo físico e o ambiente digital,

permitindo que objetos cotidianos passem a fazer parte de uma infraestrutura inteligente de comunicação.

Inicialmente, sensores realizam a aquisição de variáveis físicas de interesse, convertendo grandezas físicas como temperatura, aceleração ou pressão em dados digitais. Em seguida, o sistema embarcado realiza o processamento local dessas informações, podendo executar alguns algoritmos como os filtros e tomadas de decisões (GUBBI et al., 2013).

Posteriormente, os dados são transmitidos por meio de tecnologias de comunicação, como Wi-Fi, Bluetooth, ZigBee ou LoRaWan, para plataformas em nuvem responsáveis pelo armazenamento, processamento adicional e disponibilização das informações aos usuários (GUBBI et al., 2013).

A escolha da tecnologia de comunicação depende dos requisitos específicos de cada aplicação, para isso é necessário considerar fatores como alcance, taxa de transmissão, consumo energético e disponibilidade de infraestrutura. Da mesma forma, diferentes protocolos podem ser utilizados para a comunicação entre dispositivos e servidores. Destacam-se HTTP, MQTT e CoAP, que são amplamente usados em aplicações IoT devido à sua eficiência e flexibilidade.

Na área da saúde, essa tecnologia possibilita o desenvolvimento de dispositivos capazes de monitorar continuamente parâmetros fisiológicos, permitindo a aquisição de informações em tempo real e seu armazenamento em plataformas acessíveis de forma remota. Essa abordagem favorece o diagnóstico precoce, o acompanhamento de forma contínua do paciente e a implementação de soluções voltadas à telemedicina, por meio da aquisição e transmissão remota de dados fisiológicos (ISLAM et al., 2015).

Neste contexto, a Internet das Coisas constitui a base tecnológica utilizada neste trabalho, permitindo integrar sensores, um microcontrolador e uma plataforma de nuvem para realizar a aquisição e o monitoramento de parâmetros relacionados à Apneia Obstrutiva do Sono.

2.2.2 Monitoramento Remoto

O monitoramento remoto consiste na utilização de dispositivos eletrônicos conectados a redes de comunicação para acompanhar de forma constante parâmetros fisiológicos ou clínicos de um paciente, permitindo que essas informações sejam adquiridas, transmitidas e analisadas à distância.

De forma ampla, um sistema de monitoramento remoto é composto por sensores responsáveis pelas variáveis de interesse, um sistema embarcado encarregado do processamento dos dados e um módulo de comunicação que realiza a transmissão de informações para uma plataforma de nuvem. Nessa plataforma, os dados podem ser armazenados, organizados e disponibilizados por meio de interfaces gráficas, possibilitando assim o acompanhamento das informações quase que em tempo real por profissionais ou usuários que façam uso do sistema (GUBBI et al., 2013).

No caso da Apneia Obstrutiva do Sono, o monitoramento remoto torna-se especialmente importante, uma vez que diversos parâmetros relacionados a este distúrbio podem ser adquiridos por sensores de fácil acesso. Dados de saturação de oxigênio, frequência cardíaca, posição corporal e eventos de ronco podem ser monitorados de forma contínua durante o sono, permitindo acompanhar esses sinais ao longo de todo o período de realização do exame (KAPUR et al., 2017).

Nesse cenário, o sistema desenvolvido neste trabalho utiliza princípios de monitoramento remoto para adquirir constantemente parâmetros fisiológicos relacionados a AOS. Os dados coletados pelos sensores são processados pelo microcontrolador e transmitidos para a plataforma ThingSpeak, permitindo o armazenamento e a visualização de forma remota das informações monitoradas.

2.3 Componentes utilizados no sistema

2.3.1 Microcontrolador

Os microcontroladores são circuitos integrados desenvolvidos para executar funções específicas de controle e processamento em sistemas embarcados. Diferentemente dos computadores de propósito geral, eles integram em um único

circuito a unidade central de processamento (CPU), memórias de programa e de dados, além de diversos periféricos responsáveis pela comunicação com dispositivos externos (VALVANO, 2012). Essa elevada integração torna os microcontroladores componentes compactos e de custo reduzido, sendo amplamente utilizados em aplicações que envolvem monitoramento, aquisição e controle de sinais (VALVANO, 2012).

Em geral, um microcontrolador é composto por uma unidade de processamento responsável pela execução das instruções programadas, memória não volátil que é destinada ao armazenamento permanente do firmware e memória RAM utilizada para o armazenamento temporário de variáveis durante a execução do software (VALVANO, 2012). Além desses elementos, esses dispositivos incluem periféricos que possibilitam a interação com ambiente externo, como as portas de entradas e saídas digitais (GPIO), conversores analógicos-digitais (ADC), temporizadores (timers), módulos PWM e outras interfaces de comunicação serial.

Dentre as interfaces de comunicação mais utilizadas destacam-se UART (Universal Asynchronous Receiver-Transmitter), SPI (*Serial Peripheral Interface*), I²C (*Inter-Integrated Circuit*) e I²S (Inter-IC Sound). Essas interfaces de comunicação permitem a interação com sensores, módulos de comunicação, memórias externas e outros dispositivos eletrônicos (VALVANO, 2017). A disponibilidade de diversas interfaces em um mesmo microcontrolador proporciona elevada flexibilidade durante o desenvolvimento de sistemas embarcados, permitindo assim a integração simultânea de diferentes sensores e periféricos (VALVANO, 2017).

Atualmente, esses microcontroladores são capazes de realizar processamento digital de sinais, aquisição e tratamento de dados adquiridos dos sensores, execução de algoritmos de controle, comunicação com redes sem fio, gerenciamento de tarefas tornando o sistema multitarefa, o que torna este componente essencial em aplicações de Internet das Coisas (IoT), automação industrial, instrumentação eletrônica e monitoramento remoto (ESPRESSIF SYSTEMS, 2024).

Dentre os diversos microcontroladores disponíveis atualmente, evidenciam-se dispositivos que integram recursos de comunicação sem fio, grande capacidade de processamento e com diversas interfaces de comunicação,

características estas que os tornam especialmente adequados para aplicações baseadas em IoT (ESPRESSIF SYSTEMS, 2024).

2.3.2 Sensor de oximetria de pulso e batimento cardíaco

O sensor de oximetria de pulso e batimento cardíaco é um sensor amplamente utilizado em aplicações e projetos biomédicos devido à sua capacidade de estimar a frequência cardíaca e a saturação do oxigênio no sangue (SpO_2) por meio da aquisição e do processamento de sinais ópticos. Por integrar a aquisição desses dois sinais fisiológicos em um único módulo, o sensor tornou-se uma alternativa adequada para sistemas de monitoramento da saúde.

O princípio de funcionamento deste sensor baseia-se na utilização de dois LEDs integrados, sendo um vermelho e um infravermelho, que operam com comprimentos de onda específicos de aproximadamente 660 nm e 880 nm, respectivamente (MAXIM INTEGRATED, 2014). Além disso possui um fotodetector e um circuito responsável pelo processamento dos sinais adquiridos.

Durante a medição, os LEDs vermelho e infravermelho emitem luz que atravessa os tecidos da região analisada. Parte dessa radiação é absorvida pela hemoglobina presente no sangue, sendo que a quantidade de luz absorvida depende do grau de oxigenação da hemoglobina. A parcela remanescente é transmitida através do tecido e detectada pelo fotodiodo localizado no sensor. A partir da diferença entre a intensidade da luz emitida e da luz detectada, o sensor realiza o processamento dos sinais ópticos para estimar a saturação de oxigênio e a frequência cardíaca (MAXIM INTEGRATED, 2014).

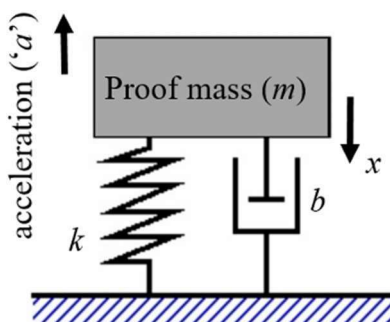
Após a aquisição dos sinais ópticos, o circuito do sensor processa as variações detectadas pelo fotodiodo para estimar a saturação de oxigênio e calcular a frequência cardíaca. A saturação de oxigênio representa a porcentagem de hemoglobina oxigenada em relação à hemoglobina total presente no sangue, enquanto a frequência cardíaca corresponde ao número de batimentos por minuto. Ambos os parâmetros são importantes para a identificação da AOS, uma vez que episódios de apneia frequentemente provocam dessaturação de oxigênio e alterações na frequência cardíaca.

2.3.3 Acelerômetro

O acelerômetro é um dispositivo capaz de medir aceleração linear e velocidade angular em três eixos ortogonais (X, Y e Z). Esses sensores integram um acelerômetro triaxial e um giroscópio triaxial em um único circuito, possibilitando a aquisição simultânea dessas grandezas físicas. Devido à elevada integração, apresentam dimensões reduzidas, baixo consumo de energia e ampla utilização em aplicações de navegação, robótica, sistemas embarcados e monitoramento de movimento (INVENSENSE, 2013).

O acelerômetro realiza as medições da seguinte forma: mede a aceleração linear usando como base a variação da capacitância resultante do deslocamento de uma massa microscópica. Quando o sensor sofre alguma interação a inércia faz com que a massa de prova (presente no interior do sensor) se desloque em direção oposta ao movimento realizado pelo corpo. Na Figura 1 pode-se verificar este modelo de medição.

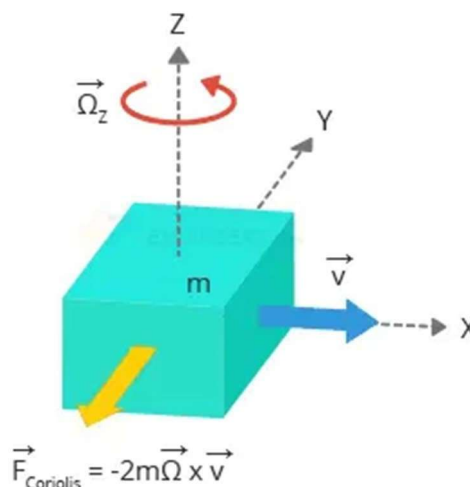
Figura 1: Modelo mecânico clássico de um acelerômetro como sistema mola-massa-amortecedor.



Fonte: Mohammed, Elfadel e Rasras (2018, p. 3).

Já o giroscópio realiza as medições utilizando o Efeito Coriolis, quando o objeto sofre uma rotação angular, o movimento de rotação gera uma força física perpendicular ao plano original, este efeito também utiliza de variação capacitiva para realizar as medições. Como pode-se observar na Figura 2 este efeito.

Figura 2: Representação vetorial do Efeito Coriolis aplicado a uma massa vibratória em um giroscópio.



Fonte: Arduino Kit Project (2026).

Para aplicações de monitoramento da posição corporal, algumas características do acelerômetro tornam-se particularmente importantes. O dispositivo deve possibilitar a aquisição da aceleração em três eixos, permitindo identificar alterações na orientação do sensor em relação ao campo gravitacional terrestre. Em situações aproximadamente estáticas, como ocorre durante grande parte do período de sono, a distribuição da aceleração gravitacional entre os eixos pode ser utilizada para determinar a orientação do corpo. Dessa forma, acelerômetros triaxiais posicionados no tronco podem ser empregados para diferenciar posições como supina, prona e laterais (TATE et al., 2020).

Para esse tipo de aplicação também é importante que o acelerômetro apresente faixa de medição e sensibilidade adequadas para detectar a componente gravitacional e as alterações de orientação decorrentes dos movimentos corporais. Como referência, dispositivos destinados à aquisição de movimento podem disponibilizar diferentes faixas de medição, permitindo adequar a sensibilidade às características da aplicação. O MPU6050, por exemplo, disponibiliza faixas configuráveis de ± 2 g, ± 4 g, ± 8 g e ± 16 g, além de conversores de 16 bits para os eixos do acelerômetro (INVENSENSE, 2013).

A inclusão e utilização deste dispositivo está relacionada à influência de algumas posições sobre a ocorrência e intensidade de eventos de AOS, principalmente a posição supina. Dessa forma, o monitoramento postural possibilita a

obtenção de informações complementares para a avaliação dos dados coletados pelos demais sensores.

2.3.4 Microfone MEMS

O microfone digital baseado na tecnologia MEMS (*Micro-Electro-Mechanical Systems*) são dispositivos que integram estruturas microscópicas e circuitos eletrônicos em um único componente semicondutor (GARDNER; VARADAN; AWADELKARIM, 2001). Essa tecnologia possibilita o desenvolvimento de sensores compactos, de baixo consumo de energia, elevada confiabilidade e boa estabilidade, características que tornam esses dispositivos amplamente utilizados em aplicações de aquisição de áudio. Além disso, apresenta baixo consumo de energia e captação omnidirecional, ou seja, capta sons oriundos de todas as direções ao seu redor (INVENSENSE, 2014).

As ondas sonoras correspondem a variações de pressão no ar que atingem a estrutura do microfone e fazem vibrar uma membrana flexível localizada em seu interior. Essas vibrações modificam a capacitância formada entre a membrana móvel e uma placa fixa perfurada existente no dispositivo. As variações de capacitância são continuamente medidas pelo circuito eletrônico e convertidas em um sinal digital que representa o áudio captado (INVENSENSE, 2014). Dessa forma, os microfones digitais MEMS constituem uma alternativa adequada para a captação de sons durante o sono, fornecendo sinais que podem ser posteriormente processados para a identificação de eventos associados à AOS.

2.3.5 Interface de Comunicação I²C

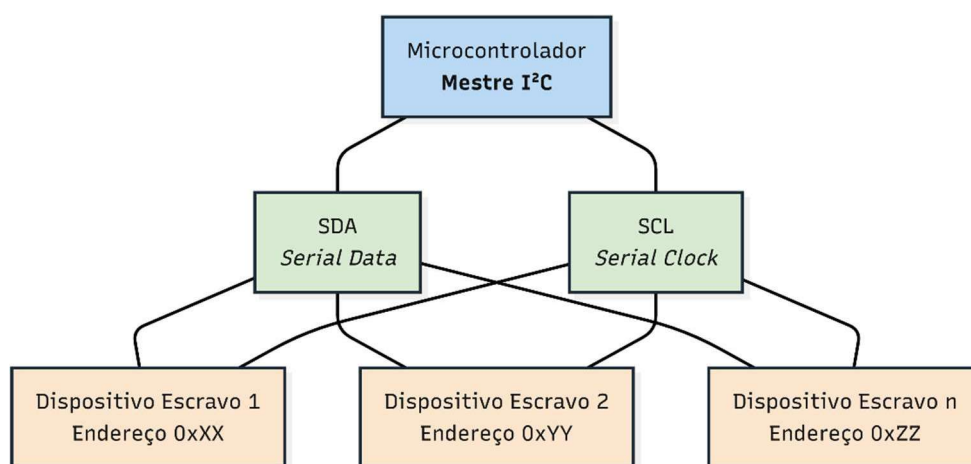
É uma interface de comunicação que consiste em um barramento de comunicação bidirecional (tecnologia desenvolvida pela Philips). É amplamente utilizada para a troca de informações entre circuitos que possam estar localizados na mesma placa. É um barramento síncrono, ou seja, necessita que ambos os circuitos estejam coordenados pelo mesmo sinal de clock, criado também para simplificar o design, pois todos os dispositivos conectados ao barramento compartilham as mesmas linhas SDA e SCL, dispensando a necessidade de linhas de comunicação

dedicadas para cada periférico, reduzindo assim a quantidade de trilhas na placa (NXP SEMICONDUCTORS, 2021).

Essa interface faz uso de duas linhas de comunicação: SDA (Serial Data) que é a parte responsável pelo fluxo dos dados bidirecionais, e SCL (Serial Clock) que é o responsável pela coordenação síncrona entre os dispositivos conectados no barramento. O método utilizado é a arquitetura mestre-escravo, na qual o microcontrolador é o mestre e detém controle do sinal de SCL, enquanto os periféricos adicionados ao barramento são os escravos, que respondem às solicitações recebidas do mestre.

Além disso, a interface faz uso de endereçamento exclusivo, o que facilita a comunicação individual com cada sensor de maneira isolada, uma das maiores vantagens de utilizar esta interface é a possibilidade de incluir vários outros sensores no mesmo barramento sem necessitar de pinos extras. Pode-se observar na Figura 3 o fluxograma dessa interface.

Figura 3 : Diagrama da interface de Comunicação I²C.



Fonte: A autora (2026).

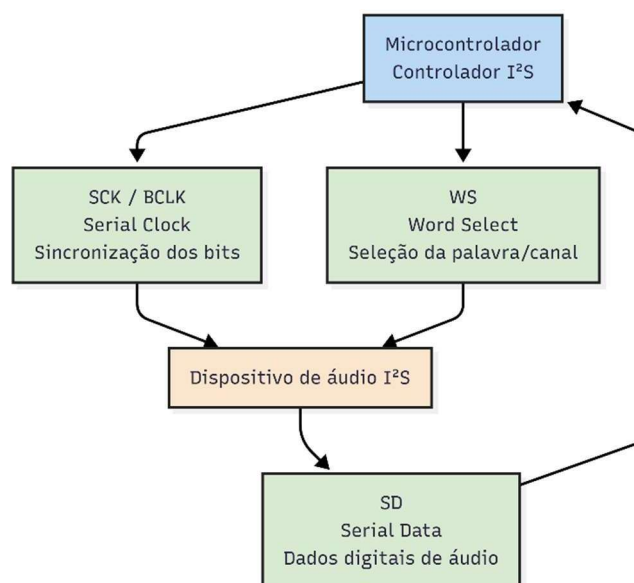
2.3.6 Interface de Comunicação I²S

A interface de comunicação I²S constitui-se de um barramento síncrono que foi também projetado pela Philips para a transferência de dados de áudio de forma digital entre sistemas. Foi desenvolvido com o objetivo de reduzir a necessidade de

utilizar conversores analógicos digitais entre os circuitos bem como diminuir ruídos do sistema (PHILIPS SEMICONDUCTORS, 1986).

Esta interface baseia-se em três sinais principais sendo elas: BCLK(*Bit Clock*), que é responsável pela sincronização temporal de cada bit individualmente do sinal, WS (*Word Select*) que é responsável pela identificação do canal de áudio de onde vem o dado (se esquerdo ou direito) e também define o início de cada palavra. Por fim, o SD (*Serial Data*) é responsável pela transmissão serial dos dados de áudio. Os bits são enviados sequencialmente, em formato de complemento de dois, sendo o bit mais significativo transmitido primeiro. Essa organização permite uma transmissão contínua e sincronizada dos sinais de áudio, tornando a interface amplamente utilizado em aplicações de aquisição e processamento digital de áudio. Na Figura 4 está o fluxograma dessa comunicação.

Figura 4 : Diagrama da interface de Comunicação I²S.



Fonte: A autora (2026).

2.3.7 Sistema Operacional FreeRTOS

O FreeRTOS é um tipo de Sistema Operacional de tempo real, muito utilizado para aplicações que possuem alto rigor de tempo devido à complexidade dessas aplicações, usado principalmente em sistemas embarcados devido ao seu baixo consumo de recursos e sua simplicidade (FREERTOS, 2026).

O FreeRTOS pode ser dividido em diversas tarefas independentes, das quais possuem o nome de *tasks*. Cada uma dessas tarefas é encarregada de ser responsável por uma função específica da aplicação, permitindo que múltiplas tarefas sejam realizadas, caracterizando o sistema operacional como multitarefa (FREERTOS, 2026).

O kernel é a parte responsável por gerenciar e coordenar as tarefas do FreeRTOS. Essa parte é responsável por decidir qual tarefa será realizada em determinado momento, utilizando como base as prioridades que já estavam definidas. Além disso, também salva o estado da tarefa atual e restaura o da próxima, fazendo todo o gerenciamento mínimo como: criar, excluir, suspender e retomar. Gerencia as filas e os grupos de eventos e também comanda a temporização das atividades.

Portanto, a utilização do sistema FreeRTOS permite um melhor aproveitamento dos recursos do sistema, além de simplificar o desenvolvimento utilizando multitarefas. Dessa forma, é possível executar diversos processos de maneira organizada e eficiente, o que garante um maior controle sobre as tarefas e atividades do sistema.

2.3.8 ThingSpeak

A plataforma ThingSpeak é um serviço de Internet das Coisas (IoT) desenvolvido para aquisição, armazenamento, visualização e análise de dados provenientes de dispositivos conectados à internet. A plataforma permite que sensores, microcontroladores e sistemas embarcados enviem informações utilizando protocolos baseados em HTTP ou MQTT, armazenando automaticamente os dados em servidores na nuvem e disponibilizando ferramentas para visualização gráfica e exportação das informações (MATHWORKS, 2024).

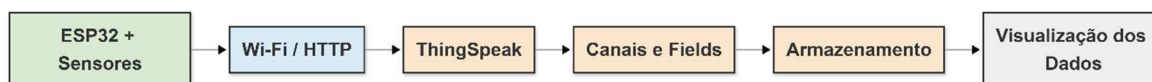
Uma das principais características da plataforma é sua arquitetura baseada em canais (*Channels*). Cada canal representa um conjunto independente de dados e pode armazenar até oito variáveis distintas, denominadas Fields. Cada registro recebido é automaticamente associado a um identificador e a um registro temporal (*timestamp*), permitindo o acompanhamento cronológico das informações adquiridas.

Além disso, cada canal possui chaves de acesso (*API Keys*), utilizadas para controlar a autenticação das operações de leitura e escrita dos dados.

Além do armazenamento das informações, o ThingSpeak disponibiliza recursos para processamento e visualização dos dados, permitindo a geração automática de gráficos, exportação em diferentes formatos e integração com ferramentas como o MATLAB, possibilitando análises estatísticas e processamento adicional diretamente na plataforma (MATHWORKS, 2024).

Outro recurso disponibilizado pela plataforma é a funcionalidade Bulk Write, responsável pelo envio de múltiplas amostras em uma única requisição HTTP. Diferentemente do envio convencional, em que cada amostra é transmitida individualmente, o Bulk Write permite agrupar diversas leituras antes da transmissão, reduzindo o número de conexões estabelecidas com o servidor. Essa abordagem diminui o *overhead* da comunicação, melhora a eficiência da utilização da rede e reduz a probabilidade de perda de dados em aplicações que realizam aquisição contínua de informações. A Figura 5 representa o funcionamento da plataforma.

Figura 5: Funcionamento da plataforma.



Fonte: A autora (2026).

2.3.9 Transformada Rápida de Fourier

A análise de sinais pode ser realizada tanto no domínio do tempo quanto no domínio da frequência. No domínio do tempo, observa-se a variação da amplitude do sinal ao longo do tempo, enquanto a representação no domínio da frequência permite identificar as diferentes componentes de frequência presentes no sinal. Para sinais amostrados digitalmente, essa representação pode ser obtida por meio da Transformada Discreta de Fourier (*Discrete Fourier Transform – DFT*), amplamente utilizada na análise espectral de séries temporais e sinais adquiridos experimentalmente (DONNELLY; RUST, 2005).

A DFT realiza a transformação de uma sequência discreta contendo N amostras do domínio do tempo para uma representação no domínio da frequência. Entretanto, o cálculo direto dessa transformação pode exigir uma quantidade elevada

de operações computacionais. A Transformada Rápida de Fourier (*Fast Fourier Transform* – FFT) consiste em uma classe de algoritmos capazes de calcular a DFT de forma mais eficiente, reduzindo significativamente o esforço computacional necessário (COOLEY; TUKEY, 1965; DONNELLY; RUST, 2005).

A representação no domínio da frequência permite observar a distribuição das componentes espectrais presentes em um sinal. Dessa maneira, a aplicação da FFT possibilita obter informações relacionadas às magnitudes ou potências associadas às diferentes frequências, permitindo a análise de regiões específicas do espectro. Essa característica torna a técnica adequada para a análise de sinais adquiridos experimentalmente, incluindo sinais sonoros (DONNELLY; RUST, 2005).

Na aplicação da DFT ou FFT a blocos finitos de amostras, podem ocorrer efeitos de espalhamento da energia entre componentes do espectro. Uma técnica utilizada para reduzir esses efeitos consiste na aplicação de funções de janelamento ao sinal antes da transformação. Entre as diferentes funções existentes encontra-se a janela de Hamming, utilizada para modificar gradualmente as amplitudes das amostras nas extremidades do bloco analisado e reduzir efeitos associados à descontinuidade dessas extremidades (HARRIS, 1978; DONNELLY; RUST, 2005).

A frequência de amostragem é um parâmetro importante para a análise espectral de sinais. De acordo com o critério de Nyquist, a frequência de amostragem deve ser superior ao dobro da maior frequência presente no sinal limitado em banda, evitando a ocorrência de *aliasing*, fenômeno no qual componentes de frequências mais elevadas podem ser representadas incorretamente em frequências inferiores após a amostragem (DONNELLY; RUST, 2005). Além disso, na aplicação da FFT, a frequência de amostragem e o número de amostras utilizadas determinam a resolução espectral obtida, influenciando a capacidade de distinguir as diferentes componentes de frequência do sinal.

2.3.10 Sistema de alimentação externo

Em sistemas embarcados alimentados por baterias, a autonomia está diretamente relacionada à energia disponível no sistema de alimentação e à potência consumida pelos componentes durante a operação. Em associações de células em

série, as tensões individuais são somadas, enquanto a capacidade, expressa em ampère-hora (Ah), permanece equivalente à capacidade de uma célula. A energia nominal disponível pode ser estimada pelo produto entre a tensão nominal do conjunto e sua capacidade, sendo normalmente expressa em watt-hora (Wh).

O consumo energético pode variar de acordo com as condições de funcionamento dos componentes. No ESP32, por exemplo, a utilização da comunicação Wi-Fi representa uma parcela importante desse consumo, podendo atingir valores típicos de aproximadamente 240 mA durante determinadas condições de transmissão (ESPRESSIF SYSTEMS, 2025). Além do microcontrolador, os sensores e o circuito de conversão de tensão também contribuem para o consumo total do sistema. Dessa forma, a autonomia efetivamente obtida pode diferir da estimativa teórica devido às condições de operação, às perdas do circuito de alimentação e à capacidade efetivamente disponível nas baterias.

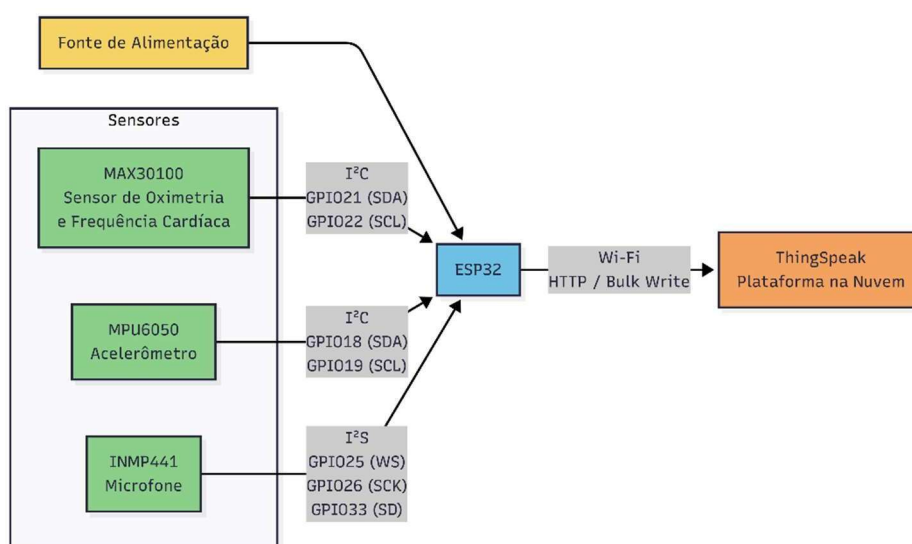
3 METODOLOGIA

3.1 Métodos aplicados

A metodologia aplicada neste trabalho constituiu-se no desenvolvimento de um sistema designado a adquirir e disponibilizar dados fisiológicos remotamente para monitoramento de Apneia do Sono (AOS). Para isso, foram utilizados sensores capazes de realizar a aquisição de sinais fisiológicos e acústicos, com base nos sintomas mais comuns desse distúrbio.

O sistema foi desenvolvido usando o microcontrolador ESP32, responsável por estabelecer a comunicação com os sensores, processar os dados adquiridos e enviá-los para a plataforma ThingSpeak. Já para a aquisição dos dados fisiológicos foram utilizados os seguintes sensores: MAX30100 para adquirir a saturação de oxigênio e frequência cardíaca, o sensor MPU6050 foi empregado para medir as acelerações nos eixos X, Y e Z, utilizadas posteriormente para a identificação da posição corporal do paciente. O INMP441 foi utilizado para captar os sinais sonoros que estão diretamente ligados ao sintoma ronco. A Figura 6 apresenta o diagrama geral do sistema desenvolvido.

Figura 6: Diagrama geral do sistema



Fonte: A autora (2026).

Embora o protocolo I²C permita a conexão de múltiplos dispositivos em um mesmo barramento por meio de endereços distintos, durante os testes iniciais foram

observadas interrupções na aquisição dos dados do MAX30100 durante a operação simultânea dos sensores. Como parte do processo de investigação do problema, o MAX30100 e o MPU6050 foram configurados em barramentos I²C independentes, utilizando diferentes pares de pinos SDA e SCL disponíveis no ESP32.

Posteriormente, verificou-se que a interrupção das leituras não estava relacionada ao endereçamento ou ao compartilhamento do barramento I²C, mas à forma como as diferentes rotinas de aquisição estavam sendo executadas, principalmente durante a aquisição e o processamento dos dados provenientes do microfone INMP441. O problema foi solucionado por meio da utilização do FreeRTOS, organizando as operações de aquisição e processamento em tarefas independentes. Apesar da identificação da causa, optou-se por manter os dois barramentos I²C separados, uma vez que essa configuração já havia sido adotada e validada nos testes do sistema.

Para o desenvolvimento do software foi utilizado o ESP-IDF (*Espressif IoT Development Framework*), por meio do ESP-IDF *Command Prompt*, em conjunto com o sistema operacional FreeRTOS. O Visual Studio Code foi empregado apenas para edição dos arquivos-fonte. Essa configuração permitiu a implementação das tarefas de aquisição, processamento e transmissão dos dados. Após o processamento, as informações foram enviadas para a plataforma ThingSpeak por meio da interface Wi-Fi integrada ao ESP32, possibilitando o armazenamento e a análise remota dos dados.

3.2 Desenvolvimento do Hardware

3.2.1 Microcontrolador ESP32

O microcontrolador ESP32 foi escolhido como unidade principal de processamento devido à sua capacidade de processamento, disponibilidade de interfaces de comunicação I²C, I²S, disponibilidade de pinos GPIO e conectividade Wi-Fi integrada. Além disso, a utilização prévia deste microcontrolador em outros projetos pela autora contribuiu para a sua seleção, facilitando o desenvolvimento e integração dos módulos do sistema. Na Figura 7 pode-se verificar o microcontrolador ESP-32S utilizado.

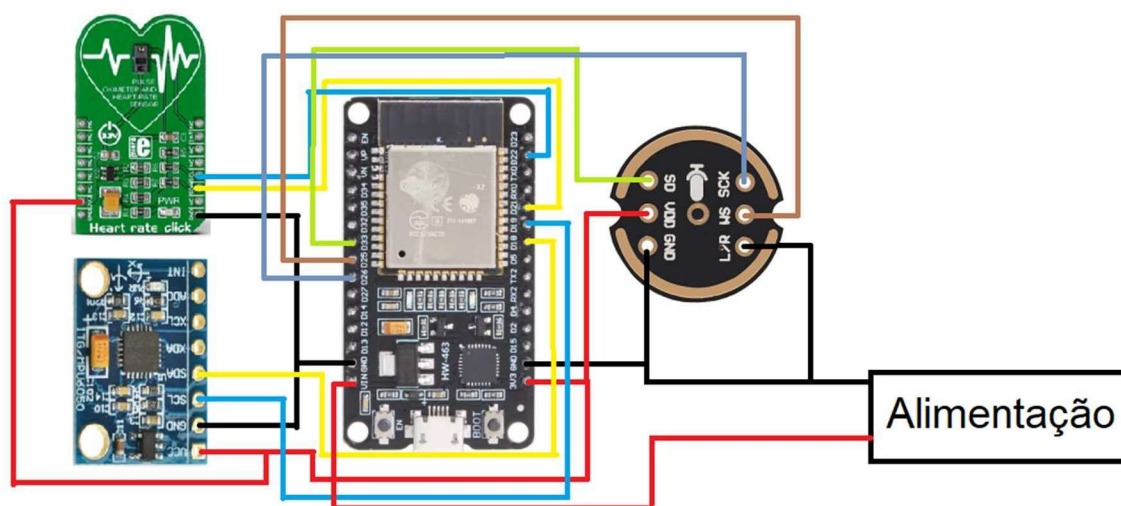
Figura 7: ESP-32S



Fonte: MakerHero (2026).

Para aquisição de dados provenientes dos sensores foram usadas diferentes interfaces de comunicação disponibilizadas pelo ESP32. Na Figura 8 é possível observar o diagrama de ligação do sistema.

Figura 8: Diagrama de ligação do sistema.



Fonte: A autora (2026).

Além do gerenciamento das interfaces de comunicação, o ESP32 também é responsável pela execução dos algoritmos de processamento de sinais adquiridos. Os dados oriundos dos sensores MAX30100 foram inicialmente processados pela biblioteca do dispositivo para estimar a frequência cardíaca e a saturação de oxigênio, sendo posteriormente submetidos a uma filtragem exponencial para suavização das leituras. Os dados adquiridos pelo MPU6050 correspondem às acelerações nos eixos

X, Y e Z, utilizadas pelo algoritmo para identificar a posição corporal do paciente. Já para o INMP441, as amostras digitais de áudio passaram por uma etapa de processamento composta pela remoção da componente contínua (*DC Offset*), aplicação da janela de Hamming e FFT, permitindo a análise das frequências associadas ao ronco e a identificação dos eventos de interesse.

Após o processamento local, os dados foram transmitidos via interface Wi-Fi integrada ao ESP32 para a plataforma ThingSpeak, possibilitando o armazenamento e monitoramento remoto das informações coletadas durante o período de análise.

3.2.2 Sensor MAX30100

O sensor MAX30100 foi selecionado para realizar a aquisição dos sinais utilizados no monitoramento da saturação de oxigênio e da frequência cardíaca do paciente. O dispositivo integra LEDs vermelho e infravermelho, fotodetector e circuito de condicionamento e conversão dos sinais ópticos.

No sistema desenvolvido, o sensor é responsável pela aquisição e digitalização desses sinais, disponibilizando amostras dos canais infravermelho (IR) e vermelho (RED), enquanto o processamento dessas amostras para obtenção da frequência cardíaca e estimativa da saturação de oxigênio (SpO_2) é realizado pelo ESP32. A utilização deste sensor é justificada pela relação desses parâmetros com os eventos de apneia, que podem provocar dessaturação sanguínea e alterações cardíacas durante o sono, conforme citado no capítulo anterior.

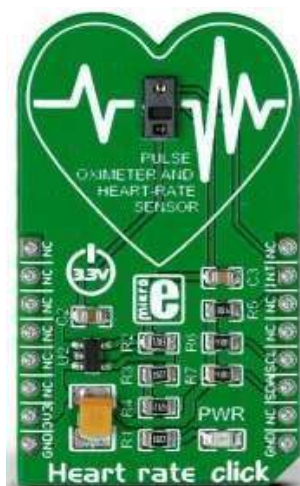
A comunicação entre o MAX30100 e o ESP32 foi estabelecida usando a interface I²C, responsável pela transferência dos dados referentes aos sinais fisiológicos do paciente. Foi utilizado os pinos GPIO 21 e GPIO 22 para os dados dos sinais SDA e SCL, respectivamente. A Figura 9 apresenta o diagrama temporal simplificado da comunicação I²C empregada na transação de leitura dos registradores do sensor. O sensor foi alimentado com tensão de 3.3 V, compatível com a operação do microcontrolador. A Figura 10 apresenta o módulo MAX30100 utilizado no sistema.

Figura 9: Diagrama temporal da comunicação I²C entre o ESP32 e o sensor MAX30100.



Fonte: A autora (2026).

Figura 10: Módulo MAX30100 utilizado.



Fonte: MikroElektronika (2026).

3.2.3 Sensor MPU6050

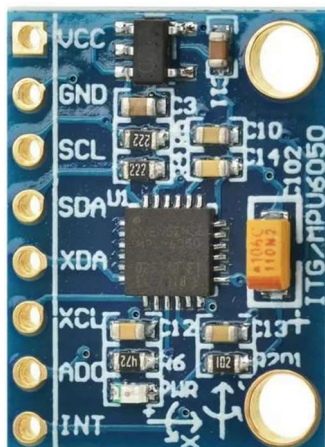
O sensor MPU6050 foi empregado e escolhido com a finalidade de monitorar as posições corporais do paciente durante o período de análise. Para integração com o sistema o sensor MPU6050 faz uso da comunicação I²C, configurada em barramento dedicado e independente dos demais dispositivos do microcontrolador. A comunicação foi realizada usando os pinos GPIO 18 para SDA e GPIO 19 para SCL e possui alimentação de 3.3 V que segue os níveis lógicos do microcontrolador. A Figura 11 apresenta o diagrama temporal simplificado da comunicação I²C empregada na transação de leitura dos registradores do sensor. Na Figura 12 apresenta o módulo MPU6050 utilizado no sistema.

Figura 11: Diagrama temporal da comunicação I²C entre o ESP32 e o sensor MPU6050.



Fonte: A autora (2026).

Figura 12: Módulo MPU6050 utilizado.



Fonte: EasyTronics (2026).

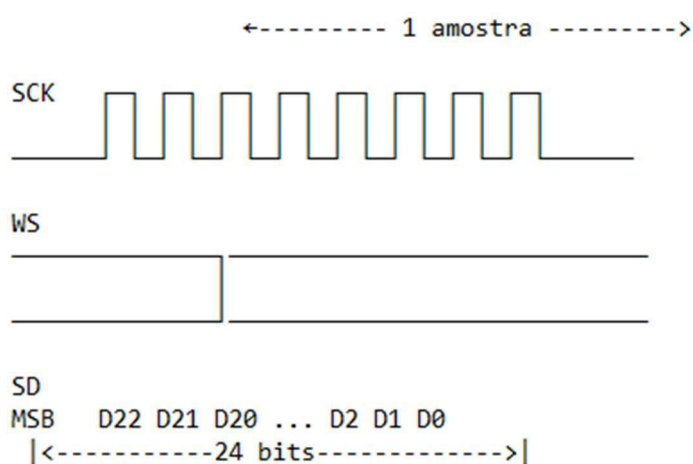
Os dados adquiridos pelo sensor foram utilizados para determinar a posição corporal do paciente durante o monitoramento. A partir das informações obtidas nos eixos de medição foi possível identificar e registrar a posição predominante do indivíduo ao longo do período de aquisição.

3.2.4 Sensor INMP441

O microfone INMP441 foi utilizado para aquisição dos sinais acústicos associados ao ronco, um dos principais sintomas perceptíveis da AOS. A escolha desse sensor ocorreu devido à sua capacidade de realizar capturas digitais de áudio, baixo consumo energético e sensibilidade de $-26 \text{ dBFS} \pm 3 \text{ dB}$, relação sinal-ruído (SNR) de 61 dBA e resposta em frequência de 60 Hz a 15 kHz, características importantes para o sistema. Além disso, o INMP441 possui conversor analógico-digital (ADC) integrado e interface de comunicação I²S, eliminando a necessidade de circuitos externos para a digitalização do sinal. O microfone opera com frequência de amostragem de 16 kHz a 48 kHz, resolução de 24 bits e tensão de alimentação entre 1,8 V e 3,3 V, características que o tornam adequado para aquisição de sinais acústicos.

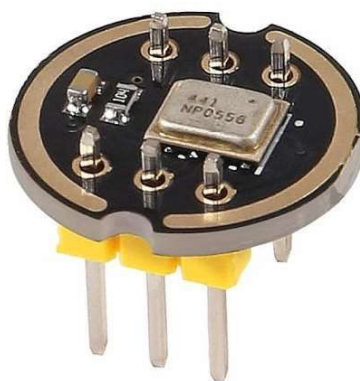
O sensor INMP441 foi integrado no sistema utilizando a interface I²S, foram utilizados os pinos GPIO 25 para WS, GPIO 26 para SCK e GPIO 33 para SD, e na Figura 13 pode ser observado o diagrama temporal da comunicação I²S entre o ESP32 e o microfone INMP441. A Figura 14 mostra o módulo utilizado no sistema.

Figura 13: Diagrama temporal da comunicação I²S entre o ESP32 e o microfone INMP441.



Fonte: A autora (2026).

Figura 14: Módulo INMP441 utilizado.



Fonte: Electropeak (2026).

Os sinais de áudio captados pelo microfone foram transmitidos pelo sensor para o microcontrolador ESP32 para processamento. As amostras de áudio foram adquiridas com uma frequência de amostragem de 4 kHz, considerado suficiente para este trabalho, uma vez que atende ao critério de Nyquist, que estabelece que a frequência de amostragem deve ser, no mínimo, o dobro da maior frequência de interesse presente no sinal. Como a análise foi feita utilizando o intervalo de 300 Hz a 1500 Hz, uma frequência de 4 kHz permite representar adequadamente os sons

respiratórios e do ronco sem aumentar desnecessariamente o volume de dados processados.

3.2.5 Sistema de alimentação

Para a alimentação do sistema foram utilizadas duas células de íons de lítio do tipo 18650, cada uma com tensão nominal de 3,6 V e capacidade nominal de 2600 mAh. As células foram conectadas em série, de modo que suas tensões fossem somadas, resultando em uma tensão nominal de 7,2 V, enquanto a capacidade do conjunto permaneceu em 2600 mAh. Em associações em série, ocorre o aumento da tensão do conjunto sem o aumento da capacidade expressa em ampère-hora.

A partir desses valores, foi determinada a energia nominal teórica disponível no conjunto de baterias. Para isso, a capacidade de 2600 mAh foi convertida para 2,6 Ah e aplicada à relação entre tensão e capacidade:

$$E = V \cdot C \quad (1)$$

Dessa forma, o conjunto apresenta uma energia nominal teórica de aproximadamente 18,72 Wh. Esse valor corresponde à energia calculada a partir das especificações nominais das células, não considerando possíveis perdas no circuito de conversão ou redução da capacidade efetiva das baterias decorrente de seu estado de conservação. A relação entre tensão, capacidade em Ah e energia em Wh é utilizada para caracterizar a energia disponível em baterias.

Para avaliar a autonomia do sistema, foi realizado um teste mantendo o protótipo em funcionamento contínuo nas mesmas condições utilizadas durante o monitoramento, com os sensores realizando as aquisições, o ESP32 executando o processamento dos sinais e a comunicação Wi-Fi permanecendo ativa para o envio dos dados à plataforma ThingSpeak. O tempo de funcionamento foi acompanhado até que o nível de alimentação não permitisse a continuidade adequada da operação do sistema.

3.3 Desenvolvimento do Software

O software do sistema foi desenvolvido utilizando ESP-IDF em conjunto com o sistema operacional FreeRTOS. Nele foi implementado um sistema de multitarefas independentes, permitindo assim que os processos de aquisição, processamento e transmissão dos dados fossem executados simultaneamente. Essa abordagem foi escolhida para reduzir interferências entre as diferentes funcionalidades do sistema.

3.3.1 Estrutura Geral do Software

Foram desenvolvidas tarefas específicas para cada módulo do sistema, uma tarefa foi responsável pela aquisição dos dados do sensor de oximetria e frequência cardíaca MAX30100, uma segunda tarefa realizou a aquisição dos dados do acelerômetro MPU6050 e uma terceira tarefa adquiriu os dados do microfone INMP441. Além dessas três tarefas principais, foram implementadas outras duas tarefas, uma responsável pelo Wi-Fi e uma pela comunicação com o Thingspeak.

A tarefa responsável pelo MAX30100 recebeu prioridade 3, a maior entre as tarefas implementadas, devido à necessidade de aquisição frequente dos sinais ópticos utilizados nos cálculos de frequência cardíaca e saturação de oxigênio. As tarefas do MPU6050 e do INMP441 receberam prioridade 2, enquanto as tarefas responsáveis pela comunicação Wi-Fi e pelo envio dos pacotes *Bulk Write* receberam prioridade 1. Dessa forma, as operações diretamente relacionadas à aquisição dos sinais receberam prioridade superior às operações de comunicação e transmissão dos dados.

Além das prioridades, foram definidos diferentes intervalos de execução. A tarefa do MAX30100 utiliza um intervalo de 20 ms entre os ciclos, enquanto as tarefas do MPU6050 e do INMP441 utilizam intervalos de 2 s. A tarefa responsável pelo *Bulk Write* realiza o envio a cada 15 s. Durante os períodos definidos por `vTaskDelay()`, as respectivas tarefas permanecem bloqueadas, permitindo ao escalonador disponibilizar o processamento para outras tarefas prontas para execução.

As trocas de informações entre essas diferentes tarefas foram realizadas utilizando filas (*queues*) que são disponibilizadas pelo FreeRTOS. Esse mecanismo

permite a comunicação entre tarefas de forma sincronizada e segura, evitando conflitos durante o acesso aos dados compartilhados (ESPRESSIF SYSTEMS, 2025). No sistema desenvolvido, as filas foram empregadas para transferir os dados adquiridos pelos sensores até a tarefa responsável pela transmissão das informações para a plataforma ThingSpeak.

3.3.2 Aquisição e Processamento dos Dados MAX30100

Como descrito no capítulo anterior a aquisição dos sinais fisiológicos foi realizada através do sensor MAX30100, responsável pela obtenção da frequência cardíaca e da saturação de oxigênio (SpO_2). Para isso, foi configurada uma tarefa própria no sistema operacional FreeRTOS, executada constantemente durante o funcionamento do sistema. Essa tarefa é responsável pela leitura das amostras digitais dos canais infravermelho (IR) e vermelho (RED) fornecidas pelo sensor, pelo processamento dos dados e armazenamento temporário das informações para posterior envio à plataforma ThingSpeak.

Após a inicialização do sensor, a tarefa realiza a leitura contínua dos valores correspondentes aos LEDs vermelho e infravermelho por meio da memória FIFO do dispositivo. Essas leituras são utilizadas como base para o processamento dos parâmetros monitorados.

A função `max30100_read_fifo()`, como pode ser visto na Figura 15 e linha 353 do código, realiza a leitura das amostras disponibilizadas pelo sensor e as armazena em buffers circulares utilizados nas etapas posteriores de processamento.

De início é realizada uma etapa de validação da qualidade do sinal adquirido. Para isso, utiliza-se a intensidade do sinal infravermelho (IR), verificando se o dedo do paciente se encontra corretamente posicionado sobre o sensor. Leituras onde a intensidade do sinal é inferior ao limiar previamente estabelecido são consideradas inválidas e descartadas, evitando que ruídos ou a ausência do dedo produzam resultados incorretos.

O limite definido pela constante `IR_VALID_THRESHOLD`, conforme pode ser visto na Figura 15 e linha 362 do código, foi estabelecido experimentalmente durante a etapa de desenvolvimento do sistema, permitindo diferenciar leituras válidas

daquelas obtidas quando o dedo não estava corretamente posicionado sobre o sensor. Quando o valor do sinal infravermelho permanece abaixo desse limite, o sistema interrompe o processamento da frequência cardíaca e reinicializa as variáveis utilizadas na detecção dos batimentos, impedindo que ruídos ou a ausência do dedo produzam valores incorretos.

Após essa etapa, o sinal infravermelho passa por um processo de filtragem exponencial, reduzindo as pequenas oscilações e variações presentes durante a aquisição dos dados. Em seguida, o sistema calcula um valor de referência utilizando a amplitude do sinal infravermelho obtido durante as últimas amostras. Sempre que o sinal ultrapassa esse valor de referência, é identificado um novo batimento cardíaco, possibilitando o cálculo do intervalo entre batimentos consecutivos.

Foi empregada uma filtragem exponencial de primeira ordem utilizando um fator de suavização (α) igual a 0,3, esse detalhe pode ser observado na Figura 15 e linha 363 do código. Essa técnica combina a amostra atual com a leitura filtrada anterior, reduzindo oscilações provocadas por ruídos e pequenas movimentações durante a aquisição do sinal.

Figura 15 - Trecho do código referente a Ratio-Of-Ratios

```

348     ESP_LOGI(TAG, "MAX30100 Task iniciada - RATIO-OF-RATIOS + Bulk 2s");
349
350     while (1) {
351         unsigned long now = xTaskGetTickCount() * portTICK_PERIOD_MS;
352
353         max30100_read_fifo(&max30100, &ir, &red);
354
355         ir_buffer[buffer_idx] = ir;
356         red_buffer[buffer_idx] = red;
357         buffer_idx = (buffer_idx + 1) % BUFFER_SIZE;
358
359         if (buffer_filled < BUFFER_SIZE)
360             buffer_filled++;
361
362         if (ir > IR_VALID_THRESHOLD) {
363             float alpha = 0.3f;
364             float filtered_ir = alpha * ir + (1 - alpha) * last_filtered;
365             last_filtered = filtered_ir;
366
367             if (filtered_ir < min_val) min_val = filtered_ir;
368             if (filtered_ir > max_val) max_val = filtered_ir;
369
370             uint16_t threshold = min_val + (max_val - min_val) * 0.6;

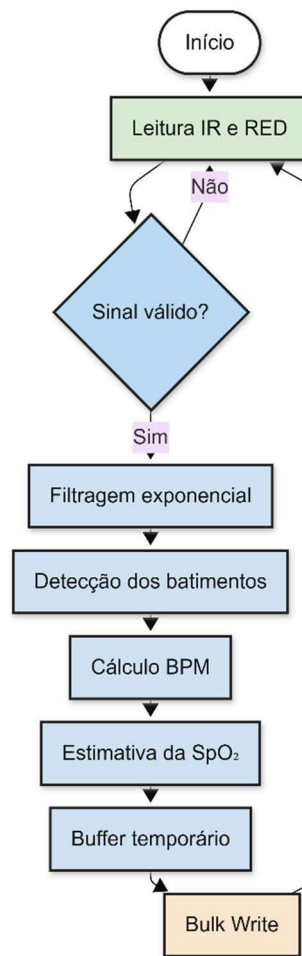
```

Fonte: A autora (2026).

Em paralelo, é realizada a estimativa da saturação periférica de oxigênio utilizando os sinais ópticos obtidos pelo sensor. Após o processamento das amostras, os valores adquiridos também são submetidos a um processo de suavização, reduzindo as oscilações e aumentando a estabilidade das medições apresentadas pelo sistema.

Após a conclusão do processamento, os valores de frequência cardíaca e saturação do oxigênio são armazenados em buffers temporários. Diferentemente do envio individual de cada leitura, o sistema faz uso do recurso *Bulk Write* da plataforma ThingSpeak, agrupando diversas amostras antes da transmissão. Essa estratégia reduz o número de requisições HTTP realizadas pelo ESP32, diminui o consumo energético associado à comunicação Wi-Fi e permite o armazenamento constante das informações monitoradas. A Figura 16 apresenta o fluxograma implementado para a aquisição e processamento provenientes do sensor MAX30100.

Figura 16 - Fluxograma de aquisição e processamento MAX30100.



Fonte: A autora (2026).

3.3.3 Aquisição e Processamento dos Dados MPU6050

O monitoramento da posição corporal foi realizado utilizando o sensor MPU6050, este é um sensor que integra um acelerômetro triaxial com capacidade de medir nos eixos X, Y e Z. No sistema desenvolvido, foi implementada uma tarefa específica para realizar a aquisição contínua desses dados e determinar automaticamente a posição do paciente durante o período de monitoramento.

Durante a execução da tarefa, o ESP32 realiza continuamente a leitura das componentes de aceleração dos eixos X, Y e Z por meio da interface I²C. A partir desses valores, o algoritmo identifica qual eixo apresenta a maior componente da aceleração gravitacional, permitindo determinar a orientação do sensor e, consequentemente, a posição corporal do paciente.

Conforme apresentado na Figura 17 e linha de código 584, a função `mpu6050_read_accel ()` realiza a leitura das acelerações correspondentes aos eixos X, Y e Z do sensor MPU6050. Durante o repouso a aceleração predominante corresponde à gravidade terrestre, tornando possível determinar automaticamente a posição do corpo apenas comparando as componentes medidas em cada eixo.

Após a aquisição, os valores de aceleração são comparados com limites previamente definidos para cada eixo, conforme observado na Figura 17 a partir da linha de código 586. Considerando que, durante o repouso, a aceleração gravitacional atua predominantemente sobre um dos eixos do acelerômetro, torna-se possível identificar a orientação do sensor em relação a gravidade. Dessa forma, torna-se possível utilizar a direção predominante desse vetor para estimar a posição do paciente, sem a necessidade de integração dos sinais ou utilização dos dados provenientes do giroscópio.

A posição corporal é identificada depois da determinação de qual eixo apresenta a maior componente gravitacional e o sentido dessa aceleração. Dessa forma, o sistema classifica automaticamente a orientação do paciente em diferentes posições previamente estabelecidas.

Os limites utilizados para classificação foram definidos durante a etapa de desenvolvimento do sistema, levando em consideração a orientação do sensor quando submetido às diferentes posições corporais. Sendo assim, o algoritmo associa cada faixa de aceleração a uma posição previamente estabelecida, conforme mostrado na Tabela 1.

Tabela 1 - Classificação adotada para as posições corporais.

Identificador	Posição corporal
1	Barriga para cima
2	Barriga para baixo
3	Lado direito
4	Lado esquerdo
5	Sentado ou em pé

Fonte: A autora (2026).

Após a classificação da posição corporal, os valores identificados são armazenados de forma temporária na memória do ESP32 para depois ser transmitido à plataforma por meio do mecanismo Bulk Write. Assim como nos demais módulos do sistema, essa estratégia foi adotada para reduzir o número de requisições HTTP, diminuir o consumo energético associado à comunicação Wi-Fi e atender as limitações da plataforma.

Figura 17 - Trecho do código referente a posição

```

583     while (1) {
584         mpu6050_read_accel(&mpu6050, &ax, &ay, &az);
585
586         int pos = 0;
587         if (az > 12000) pos = 1;
588         else if (az < -12000) pos = 2;
589         else if (ay > 12000) pos = 3;
590         else if (ay < -12000) pos = 4;
591         else if (ax < -12000) pos = 5;
592
593         if (mpu_bulk_idx < MPU_BULK_SIZE) {
594             mpu_bulk_buffer[mpu_bulk_idx] = pos;
595             mpu_bulk_idx++;
596         }
597
598         ESP_LOGI(TAG, "[MPU] Pos=%d (BULK[%d])", pos, mpu_bulk_idx);
599
600         vTaskDelay(pdMS_TO_TICKS(2000));
601     }

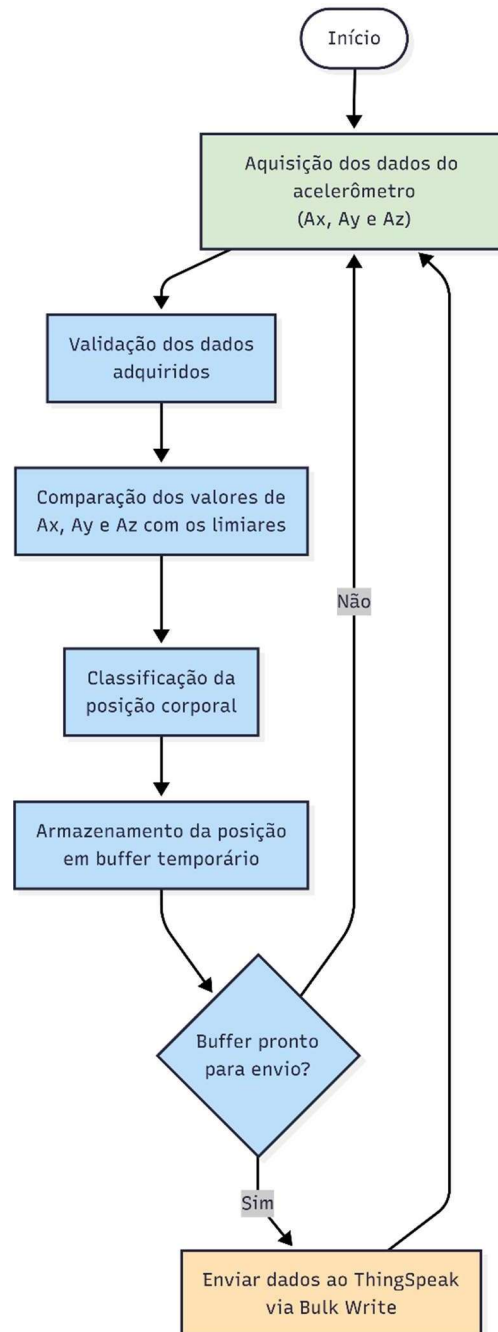
```

Fonte: A autora (2026).

O algoritmo foi programado para armazenar uma nova posição corporal aproximadamente a cada dois segundos. Depois as posições armazenadas são

agrupadas e transmitidas em uma única requisição HTTP a cada quinze segundos. Na Figura 18 é possível observar o fluxograma deste sistema.

Figura 18: Fluxograma de aquisição e processamento MPU6050.



Fonte: A autora (2026).

3.3.4 Aquisição e Processamento dos Dados INMP441

O monitoramento dos sinais acústicos foi realizado utilizando o microfone INMP441, o mesmo é responsável pela aquisição dos sons produzidos durante o período de sono do paciente. O principal objetivo dessa etapa foi identificar características associadas ao ronco, um dos principais sintomas observáveis da Apneia Obstrutiva do Sono (AOS). Para isso, foi desenvolvida uma tarefa específica para esta função, ela é executada continuamente durante o funcionamento do sistema responsável pela aquisição, processamento e armazenamento dos sinais sonoros.

Durante o período de execução da tarefa, o ESP32 realiza periodicamente a aquisição de blocos de 256 amostras do sinal, utilizando uma frequência de amostragem de 4 kHz. Foi definida essa configuração considerando que as componentes espectrais do ronco se encontram abaixo de 2 kHz, permitindo assim a representação mais adequada do sinal adquirido. A aquisição é feita utilizando a interface I²S, sendo os dados armazenados em um vetor de amostras para serem processadas posteriormente. Na Figura 19 e na linha de código 502 observa-se a função descrita anteriormente.

Conforme apresentado no trecho da Figura 19 e na linha de código 502, a função `i2s_manual_read ()` realiza a leitura das amostras captadas pelo microfone digital. As amostras adquiridas são armazenadas no vetor `audio_samples`, que serve como entrada para as etapas seguintes.

Após a aquisição das amostras, inicia-se uma etapa de pré-processamento do sinal. Os dados recebidos pela interface I²S são armazenados em variáveis inteiras de 32 bits. Antes do processamento, é realizado um deslocamento de 14 bits para a direita (`>> 14`), reduzindo a magnitude numérica das amostras e adequando sua escala para as operações subsequentes. Esse deslocamento equivale, numericamente, a uma divisão por 2^{14} , ou 16.384, reduzindo a amplitude dos valores sem modificar a forma do sinal.

Para cada novo bloco de amostras adquirido, é calculada a média de todas as amostras adquiridas e esse valor é subtraído de cada ponto do sinal, removendo a componente contínua (*DC Offset*). Essa etapa é necessária porque a presença de uma componente contínua acaba deslocando o sinal em relação ao eixo de referência,

concentrando energia na frequência zero e podendo comprometer a representação espectral obtida na etapa seguinte pela FFT. Na figura 19 e linha de código 505 é possível verificar o trecho do código responsável pelo ajuste da escala das amostras e pela remoção da componente contínua do sinal.

A remoção dessa componente contínua é importante porque evita que deslocamentos no nível médio do sinal interfiram na análise, centralizando o sinal em torno de zero, acabamos por reduzir a influência de componentes não relacionados ao conteúdo sonoro de interesse, melhorando a representação do sinal no domínio da frequência.

Em seguida, foi aplicada uma janela de Hamming sobre todas as amostras do sinal. Como o processamento é realizado sobre blocos finitos de dados, passível de ocorrer descontinuidades entre o início e o fim de cada janela de aquisição, provocando o espalhamento da energia entre frequências adjacentes, fenômeno este conhecido como vazamento espectral. A utilização da janela de Hamming reduz esse efeito, proporcionando uma representação mais precisa do sinal aumentando a confiabilidade da análise do sistema. Na Figura 19 e na linha de código 517 pode-se observar o trecho do código responsável pela aplicação da janela de Hamming.

A utilização da janela de Hamming contribui para tornar a análise espectral mais estável, principalmente em sinais de áudio captados em ambiente real, nos quais podem ocorrer variações bruscas e ruídos externos.

Após concluída a etapa de pré-processamento, o ESP32 executa uma implementação da Transformada Rápida de Fourier (FFT). A partir dos coeficientes complexos obtidos pela FFT, é calculada a magnitude espectral, que representa a contribuição de cada frequência para a composição do sinal. Na Figura 19 e na linha de código 522 observa-se o trecho do código responsável pela aplicação da FFT.

A FFT converte o sinal de áudio do domínio do tempo para o domínio da frequência, permitindo identificar quais componentes espectrais apresentam maior intensidade durante o intervalo analisado. Essa etapa é essencial para a detecção de ronco, pois o sistema não considera apenas a amplitude total do som captado, mas sim a energia que fica concentrada em uma faixa específica de frequências

Apesar de a FFT ser calculada para todo o espectro de frequências do sinal adquirido, apenas as componentes compreendidas entre 300 Hz e 1500 Hz são

consideradas na etapa de análise. Essa escolha foi baseada na literatura, que demonstra que a maior parte das componentes espectrais do ronco está concentrada predominantemente abaixo de 2 kHz, sendo observadas regiões de maior interesse entre aproximadamente 300 Hz e 1500 Hz, dependendo do local de obstrução das vias aéreas e das características acústicas do ronco (LEE et al.,2016). Dessa forma, restringir a análise a esses valores reduz a influência de ruídos externos e concentra o processamento nas frequências mais relevantes para a identificação do ronco. Na Figura 19 e na linha de código 533 pode-se verificar o trecho do código responsável pela seleção da faixa de frequência utilizada para cálculo da energia sonora.

A seleção dessa faixa permite que o sistema concentre e analise as componentes mais relevantes para os sons associados ao ronco, reduzindo a influência de ruídos de baixa frequência e de componentes sonoras fora da região de interesse. Logo após o cálculo da energia média nessa faixa, o valor obtido é normalizado e comparado com um limiar que foi pré definido experimentalmente

Em seguida, o valor de energia obtido é normalizado e comparado com um limite previamente estabelecido. Esse limite atua como um critério de decisão, o que permite distinguir sons de baixa intensidade e ruídos do ambiente, quando a energia ultrapassa o valor de referência o sistema identifica e classifica o período analisado como ronco. Do contrário, é considerado como um período do qual não houve ocorrência de ronco. Na Figura 19 e na linha de código 540 consta o trecho do código responsável pela classificação dos eventos de ronco.

Figura 19 - Trecho do código responsável pelas funções do microfone

```

501 while (1) {
502     int n = i2s_manual_read(&i2s_mic, audio_samples, SAMPLES, 50);
503
504     if (n >= 50) {
505         int32_t soma = 0;
506         for (int i = 0; i < n; i++) {
507             soma += (audio_samples[i] >> 14);
508         }
509         float media = (float)soma / n;
510
511         for (int i = 0; i < SAMPLES; i++) {
512             if (i < n) fft_real[i] = (audio_samples[i] >> 14) - media;
513             else fft_real[i] = 0;
514             fft_imag[i] = 0;
515         }
516
517         for (int i = 0; i < SAMPLES; i++) {
518             float window = 0.54f - 0.46f * cosf(2 * 3.14159265f * i / (SAMPLES - 1));
519             fft_real[i] *= window;
520         }
521
522         simple_fft(fft_real, fft_imag, SAMPLES);
523
524         for (int i = 0; i < SAMPLES / 2; i++) {
525             magnitudes[i] = sqrtf(fft_real[i] * fft_real[i] + fft_imag[i] * fft_imag[i]);
526         }
527
528         double energia_ronco = 0;
529         int count = 0;
530
531         for (int i = 0; i < SAMPLES / 2; i++) {
532             float freq = (i * 1.0f * SAMPLING_FREQ) / SAMPLES;
533             if (freq >= 300.0f && freq <= 1500.0f) {
534                 energia_ronco += magnitudes[i];
535                 count++;
536             }
537         }
538         energia_ronco /= (count > 0 ? count : 1);
539
540         float energia_norm = energia_ronco / 100.0f;
541         if (energia_norm > 100) energia_norm = 100;
542         bool ronco = (energia_norm > 2.0f);
543     }

```

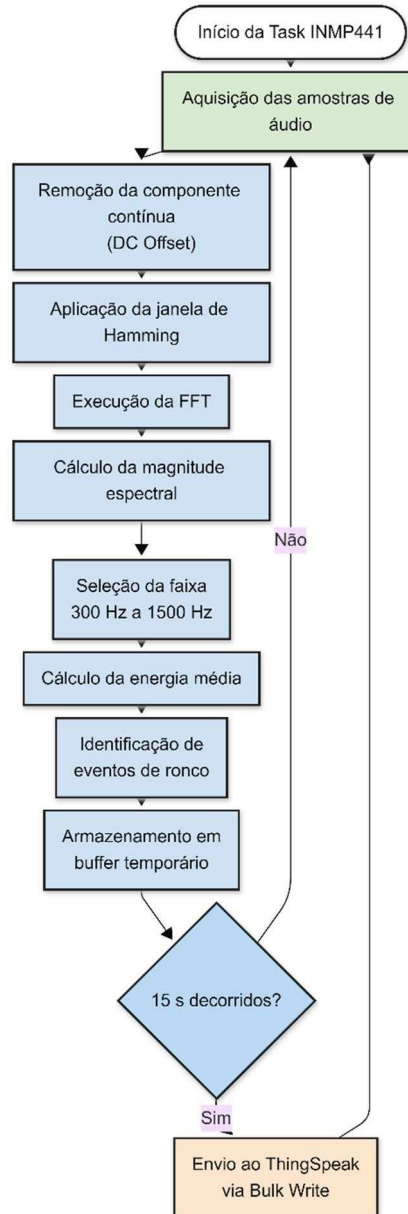
Fonte: A autora (2026).

Quando a energia normalizada ultrapassa o limiar estabelecido, o sistema classifica o intervalo analisado como um evento de ronco, armazenando o valor 1 na variável “ronco”, caso contrário o intervalo é classificado como uma ausência de ronco, recebendo o valor 0. Sendo assim, o sistema não apenas registra a intensidade sonora, mas também produz uma variável binária que indica a ocorrência ou não de ronco durante o monitoramento.

Por fim, os valores calculados são armazenados em um buffer temporário localizado na memória do ESP32. Os dados permanecem armazenados até que seja

realizado o envio periódico para a plataforma ThingSpeak utilizando o mecanismo Bulk Write. Na Figura 20 verifica-se o fluxograma deste sistema.

Figura 20: Fluxograma de aquisição e processamento INMP441.



Fonte: A autora (2026).

3.3.5 Implementação da Plataforma ThingSpeak

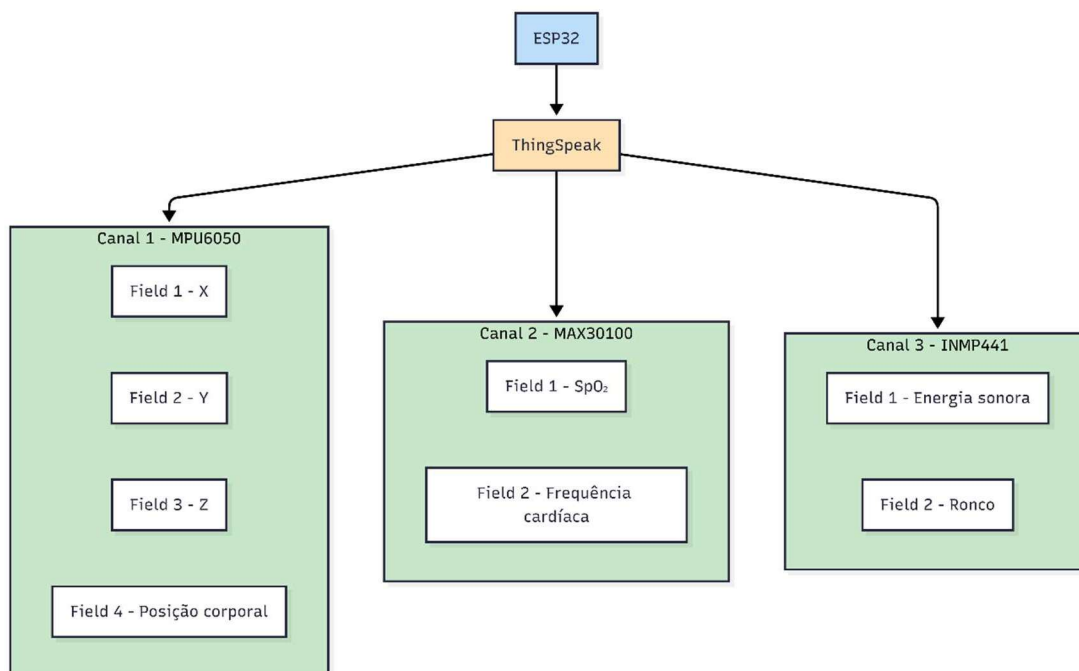
Após a aquisição e o processamento dos sinais advindos dos sensores, os dados são armazenados em *buffers* localizados na memória do ESP32. Esse procedimento permite separar a etapa de processamento da etapa de comunicação

com a plataforma, evitando que o tempo necessário de transmissão interfira na execução das demais tarefas.

Uma tarefa específica do FreeRTOS é responsável pelo gerenciamento da comunicação com a plataforma ThingSpeak, essa tarefa é executada periodicamente e realiza o envio de dados utilizando o mecanismo Bulk Write.

Considerando que os diferentes sensores empregados no sistema apresentam frequências distintas de aquisição, optou-se pela utilização de três canais independentes na plataforma, permitindo a organização dos dados de acordo com suas características e facilitando posteriormente sua análise. Na Figura 18 pode-se observar a organização dos canais e campos da plataforma.

Figura 21: Organização dos canais e campos da plataforma ThingSpeak.



Fonte: A autora (2026).

O primeiro canal foi utilizado para armazenamento dos dados provenientes do acelerômetro MPU6050. Nesse caso foram utilizados quatro campos, correspondentes às componentes dos eixos X, Y e Z do acelerômetro e à posição corporal identificada pelo algoritmo do sistema.

O segundo canal foi destinado ao armazenamento dos dados provenientes do sensor MAX30100, sendo utilizados dois campos (*Fields*): o primeiro para registro

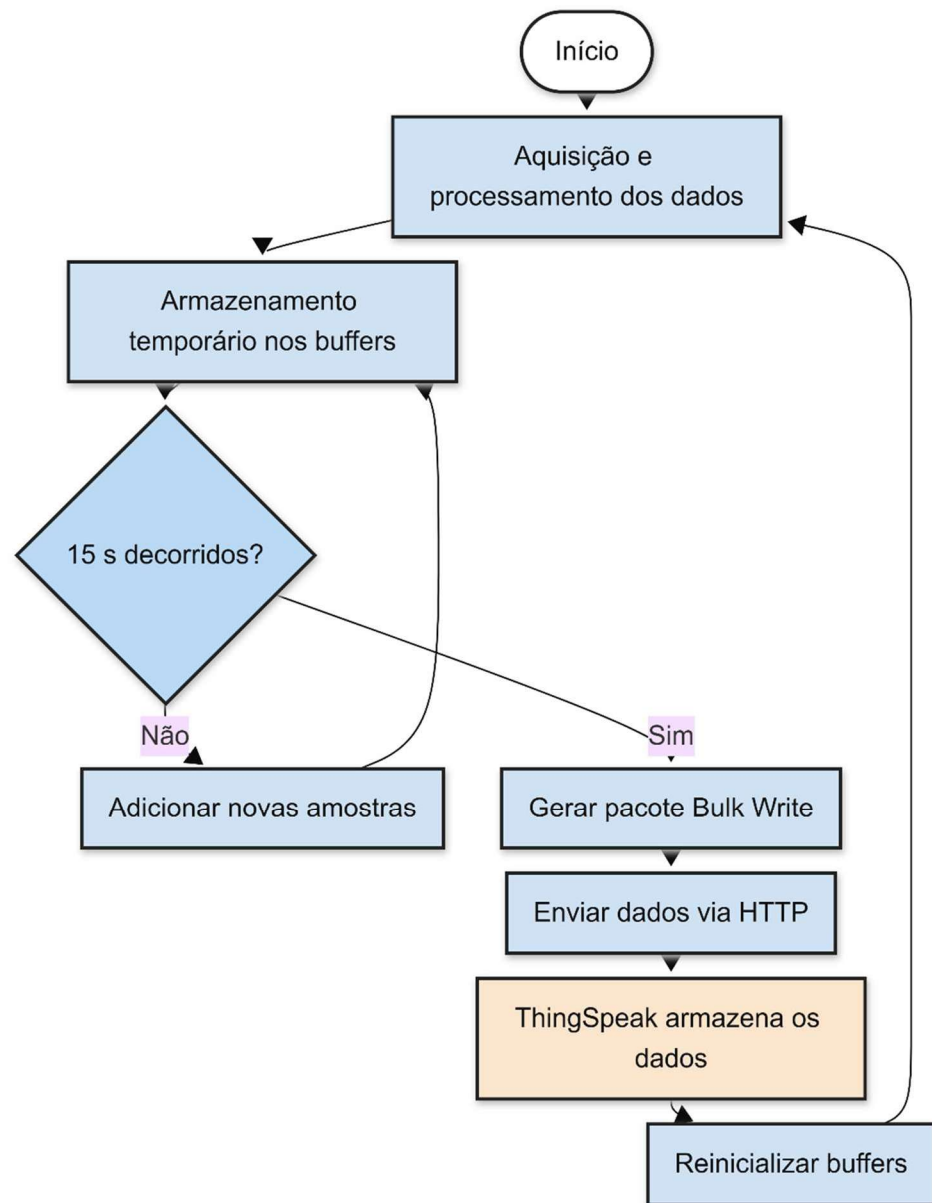
da saturação de oxigênio (SpO_2) e o segundo para armazenamento da frequência cardíaca (BPM).

Por fim, o terceiro canal foi destinado ao armazenamento das informações obtidas pelo microfone INMP441. Neste canal foram utilizados dois campos, sendo o primeiro responsável pelo armazenamento da energia sonora obtida a partir da análise espectral do sinal de áudio e o segundo utilizado para registrar o resultado da classificação referente à presença ou ausência de ronco.

Inicialmente, foram realizados testes utilizando transmissões individuais por meio de requisições HTTP. Entretanto, verificou-se que essa estratégia ocasionava aumento significativo do número de conexões com o servidor, reduzindo a eficiência da comunicação e ocasionando falhas durante longos períodos de aquisição. Em função desse comportamento, adotou-se a funcionalidade Bulk Write, permitindo que diversas amostras fossem armazenadas temporariamente em buffers no ESP32 e posteriormente transmitidas em uma única requisição HTTP.

Dessa forma, os dados permanecem organizados em memória até o intervalo de transmissão estabelecido pelo sistema. Após esse período, todas as amostras eram agrupadas em um único pacote e enviadas simultaneamente para o ThingSpeak, reduzindo o número de requisições realizadas, melhorando a eficiência da comunicação e aumentando a confiabilidade do armazenamento das informações adquiridas. Na Figura 22 verifica-se o fluxograma da plataforma ThingSpeak.

Figura 22 - Fluxograma da plataforma ThingSpeak.



Fonte: A autora (2026).

4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo serão apresentados os resultados encontrados durante a etapa experimental do sistema desenvolvido para aquisição dos parâmetros relacionados à Apneia Obstrutiva do Sono. Após a conclusão do desenvolvimento do hardware e da implementação do software, foram realizados testes em condições controladas, com o objetivo de verificar o funcionamento dos sensores, dos algoritmos e da comunicação com a plataforma ThingSpeak. Durante os testes, a protoboard contendo o microcontrolador, o acelerômetro e o microfone foi fixada na região torácica do voluntário, enquanto o sensor MAX30100 foi posicionado no dedo indicador para a aquisição dos sinais.

Primeiramente serão descritos os procedimentos adotados durante a realização dos testes e após isso serão apresentados os resultados obtidos para cada módulo do sistema, incluindo o monitoramento da frequência cardíaca e da saturação de oxigênio, da posição corporal e da detecção de ronco. Além disso, serão discutidos o desempenho do sistema de transmissão utilizando o recurso Bulk Write, o armazenamento das informações na plataforma ThingSpeak, a autonomia do sistema de alimentação do projeto e as limitações observadas.

4.1 Avaliação do Monitoramento da Frequência Cardíaca e Saturação de Oxigênio

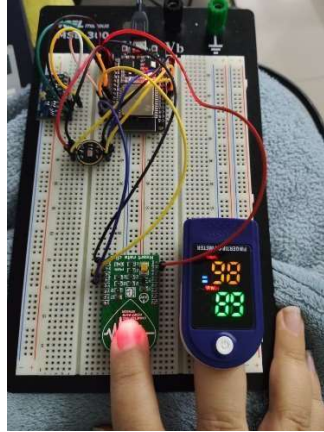
4.1.1 Validação experimental

Com o algoritmo já implementado foi realizada uma etapa de validação experimental, com o objetivo de verificar a coerência dos valores obtidos pelo sistema desenvolvido, para isso, foi utilizado um oxímetro de pulso comercial como equipamento de referência para comparação qualitativa das medições de frequência cardíaca e saturação de oxigênio.

O teste experimental foi realizado mantendo simultaneamente o sensor MAX30100 e o oxímetro comercial em funcionamento, conforme na Figura 23. Durante o teste o sistema desenvolvido processava de forma contínua os sinais do sensor e apresentava os valores calculados por meio do monitor serial do ESP32,

enquanto o oxímetro exibia as informações diretamente em seu display. A Figura 24 apresenta a captura do monitor serial correspondente ao instante do teste.

Figura 23: Protótipo físico em funcionamento.



Fonte: A autora (2026).

Figura 24: Captura do monitor serial correspondente ao instante do teste.

```

ESP-IDF 5.2 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-f42326bf65dccc3372af4dce79ef4e01

I (330368) MAIN: [MAX] BPM=88.6 | SpO2=96.0% | IR=21824
I (330448) THINGSPEAK: Enviando: 96.0, 88.6
I (330628) MAIN: [MPU] X=-1836 Y=-932 Z=15608 | Pos=1
I (330648) THINGSPEAK: Enviando: -1836, -932
I (331248) THINGSPEAK: Enviando: 1.90, 0
I (331368) MAIN: [MAX] BPM=90.2 | SpO2=96.0% | IR=22080
I (331448) THINGSPEAK: Enviando: 96.0, 90.2
I (332368) MAIN: [MAX] BPM=91.2 | SpO2=96.0% | IR=21984
I (332448) THINGSPEAK: Enviando: 96.0, 91.2
I (332628) MAIN: [MPU] X=-1788 Y=-904 Z=15664 | Pos=1
I (332648) THINGSPEAK: Enviando: -1788, -904
I (333368) MAIN: [MAX] BPM=92.0 | SpO2=96.0% | IR=21920
I (333448) THINGSPEAK: Enviando: 96.0, 92.0
I (334368) MAIN: [MPU] BPM=92.5 | SpO2=96.0% | IR=21888
I (334448) THINGSPEAK: Enviando: 96.0, 92.5
I (334628) MAIN: [MPU] X=-1800 Y=-892 Z=15544 | Pos=1
I (334648) THINGSPEAK: Enviando: -1800, -892
I (335368) MAIN: [MAX] BPM=92.9 | SpO2=96.0% | IR=21984
I (335448) THINGSPEAK: Enviando: 96.0, 92.9
I (336248) THINGSPEAK: Enviando: 1.11, 0
I (336368) MAIN: [MAX] BPM=89.7 | SpO2=96.0% | IR=21888
I (336448) THINGSPEAK: Enviando: 96.0, 89.7
I (336628) MAIN: [MPU] X=-1780 Y=-856 Z=15776 | Pos=1
I (336648) THINGSPEAK: Enviando: -1780, -856
I (337368) MAIN: [MAX] BPM=87.5 | SpO2=96.0% | IR=21984
I (337448) THINGSPEAK: Enviando: 96.0, 87.5

```

Fonte: A autora (2026).

A comparação entre os valores apresentados pelos dispositivos é apresentada na Tabela 2.

Tabela 2 - Comparação protótipo X oxímetro comercial.

Parâmetro	Sistema desenvolvido	Oxímetro comercial	Diferença
Freq. Cardíaca	88,6 bpm	90 bpm	1,4 bpm
SpO ₂	96%	98%	2%

Fonte: A autora (2026).

Observa-se que os valores obtidos pelo sistema desenvolvido permanecem próximos daqueles apresentados pelo equipamento comercial durante o período de teste. As pequenas diferenças observadas eram esperadas, uma vez que as medições foram realizadas em dedos diferentes. Além disso, o equipamento comercial emprega processamento interno proprietário, cujos algoritmos e critérios de filtragem não são divulgados pelo fabricante. Dessa forma, essa comparação teve como objetivo validar qualitativamente a coerência das medições obtidas pelo protótipo, não sendo caracterizado como uma calibração.

4.1.2 Armazenamento e envio

Após o cálculo da frequência cardíaca e da saturação de oxigênio, os valores não são enviados imediatamente à plataforma ThingSpeak, devido a limitação de atualização da plataforma ocorreu a implementação do mecanismo Bulk Write, no qual diversas amostras são armazenadas na memória do ESP32 e posteriormente transmitidas em uma única requisição de HTTP, na Figura 25 e início do código na linha 468, observa-se o trecho responsável pelo armazenamento das amostras no buffer temporário.

Figura 25 - Trecho responsável pelo armazenamento das amostras no buffer temporário

```

468 if (now - last_max_bulk_time >= 2000) {
469     last_max_bulk_time = now;
470
471     if (max_bulk_idx < MAX_BULK_SIZE) {
472         max_bpm_buffer[max_bulk_idx] = bpm_filtrado;
473         max_spo2_buffer[max_bulk_idx] = spo2_filtrado;
474         max_bulk_idx++;

```

Fonte: A autora (2026).

O sistema foi programado para armazenar uma nova amostra considerada válida aproximadamente a cada 2 segundos, enquanto a transmissão para o THingSpeak ocorre a cada 15 segundos, considerando essas informações seria possível armazenar aproximadamente 7,5 amostras:

$$N = T_{trans} \div T_{amostra} \quad (2)$$

Entretanto, durante os testes experimentais foi possível observar que a quantidade armazenada por ciclo foi inferior ao valor teórico. Isso ocorre porque apenas leituras que foram consideradas válidas foram adicionadas ao buffer, sendo descartadas aquelas obtidas durante instabilidades do sinal, ausência do dedo ou com intervalos insuficientes entre batimentos consecutivos.

Para verificar a taxa real de armazenamento, foi analisado o arquivo JSON retornado pela API do ThingSpeak. Durante o intervalo escolhido para realizar o cálculo da taxa foram analisados aproximadamente 332 segundos e 104 amostras, o que nos permitiu os seguintes valores:

A taxa média de aquisição é de 0,31 amostras por segundo:

$$T_{média} = N_{total} \div \Delta t \quad (3)$$

Intervalo médio entre amostras armazenadas foi de 3,19 segundos:

$$T_{médio} = \Delta t \div N_{total} \quad (4)$$

Considerando que o envio ocorre a cada 15 segundos, a quantidade média de amostras transmitidas por ciclo é de 4,68 amostras, utilizando a seguinte equação:

$$N_{ciclo} = (N_{tot} / \Delta t) \cdot T_{trans} \quad (5)$$

Esse resultado demonstra que, apesar de o buffer ter sido dimensionado para armazenar até sete amostras por ciclo de transmissão, durante o teste analisado foram enviadas, em média, aproximadamente cinco amostras válidas por pacote. Essa diferença está diretamente relacionada ao processo de validação do sinal

implementado, que prioriza a qualidade das informações armazenadas ao invés da quantidade de amostras transmitidas.

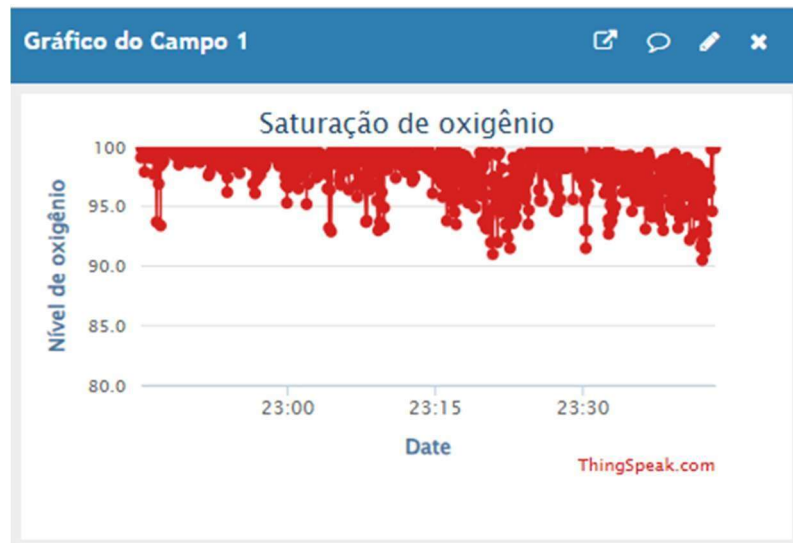
4.1.3 Monitoramento contínuo

Após a validação do algoritmo e da transmissão das informações para a plataforma ThingSpeak, foram realizados testes experimentais com o objetivo de avaliar o comportamento do sistema durante o monitoramento contínuo dos parâmetros. Durante a etapa de desenvolvimento, foram executados diversos testes com duração aproximada de 15 minutos e 1 hora, permitindo verificar a estabilidade das leituras, o funcionamento do algoritmo e a transmissão contínua das informações. Além desses testes, foram realizados testes longos o de maior duração teve 5 horas e 20 minutos, buscando avaliar o desempenho do sistema durante um período prolongado de monitoramento.

Os resultados apresentados neste trabalho correspondem aos testes de 15 minutos e de 1 hora, por representarem medições válidas e consistentes para a avaliação do sistema. Durante o teste de 5 horas e 20 minutos, realizado durante o período de sono, o sensor MAX30100 perdeu o contato com o dedo do voluntário ao longo da noite. Essa condição foi identificada apenas durante a análise dos dados registrados, impossibilitando a utilização desse ensaio na validação das medições. Essa ocorrência evidenciou uma limitação do protótipo, relacionada à dificuldade de manter o sensor adequadamente fixado ao dedo durante longos períodos de monitoramento.

Conforme observado na Figura 26, a plataforma permitiu visualizar continuamente os valores enviados pelo sistema desenvolvido. Os dados de saturação de oxigênio permaneceram predominantemente entre 95% e 100%, valores compatíveis com indivíduos saudáveis em repouso.

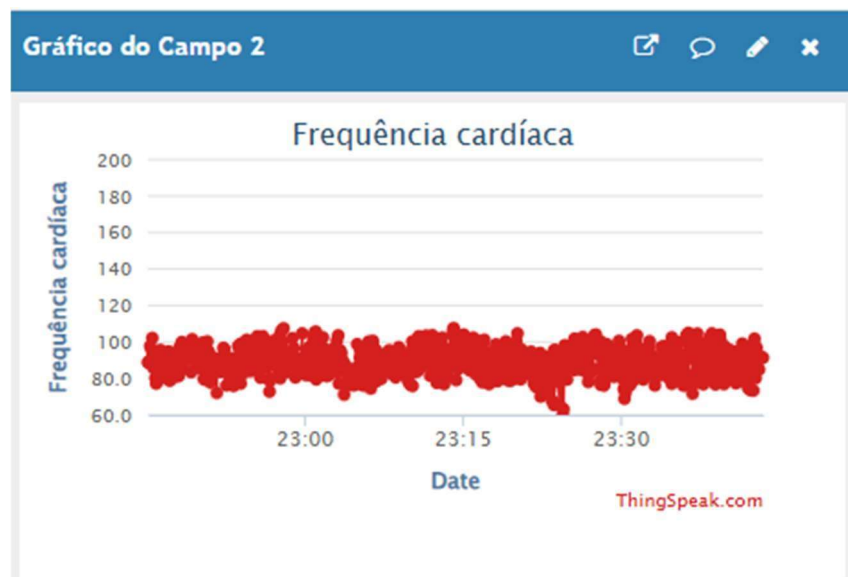
Figura 26: Visualização em tempo real dos valores de saturação de oxigênio armazenados na plataforma ThingSpeak.



Fonte: A autora (2026).

Da mesma forma, na Figura 27 a frequência cardíaca apresentou pequenas variações ao longo do período monitorado, comportamento esperado em função das variáveis fisiológicas naturais e de pequenas movimentações realizadas durante o teste.

Figura 27: Visualização em tempo real dos valores de frequência cardíaca armazenados na plataforma ThingSpeak.

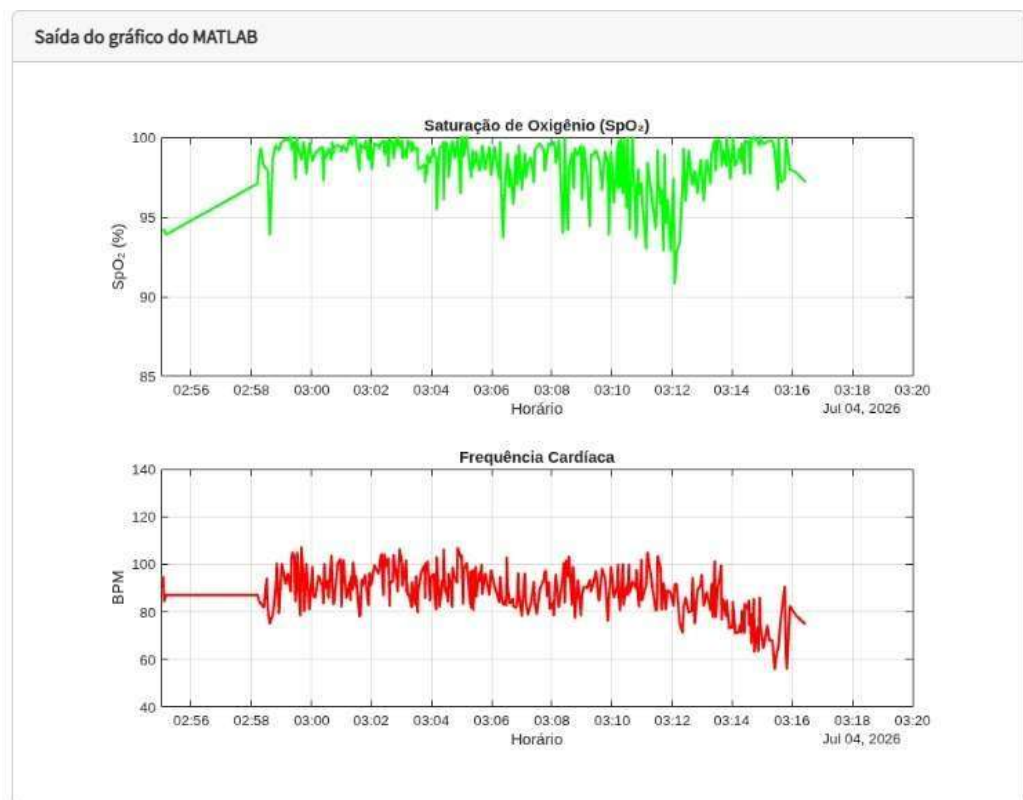


Fonte: A autora (2026).

Embora a plataforma ThingSpeak apresente os dados individualmente em tempo real, a utilização da ferramenta MATLAB integrada possibilitou uma análise

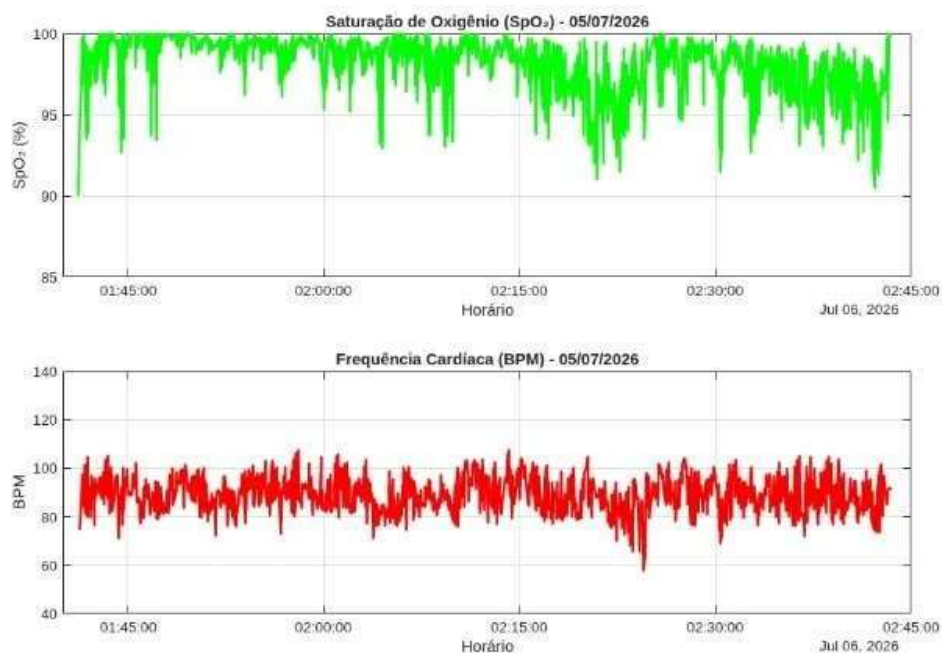
mais detalhada do comportamento temporal dos parâmetros monitorados. De acordo com a Figura 28 e com a Figura 29, é possível observar que a saturação de oxigênio permaneceu relativamente estável durante praticamente todo o teste, apresentando pequenas oscilações que podem ser atribuídas ao próprio sensor óptico, pequenas alterações na pressão exercida pelo dedo sobre o sensor e ao movimento natural da mão durante o monitoramento.

Figura 28: Visualização em tempo real (teste de 15min) dos valores de frequência cardíaca e saturação de oxigênio utilizando a ferramenta MATLAB Analysis do ThingSpeak.



Fonte: A autora (2026).

Figura 29: Visualização em tempo real (teste de 1h) dos valores de frequência cardíaca e saturação de oxigênio utilizando a ferramenta MATLAB Analysis do ThingSpeak.



Fonte: A autora (2026).

A frequência cardíaca apresentou maior variabilidade quando comparada à saturação de oxigênio, comportamento esperado devido à própria dinâmica fisiológica do organismo e às pequenas variações existentes entre batimentos consecutivos. Apesar dessas oscilações, observa-se que os valores permaneceram dentro de uma faixa fisiologicamente compatível durante todo o período de teste.

A estabilidade observada nos gráficos confirma a eficiência das etapas de processamento implementadas no sistema desenvolvido, a utilização da validação da intensidade do sinal infravermelho, da filtragem exponencial, do limiar adaptativo para detectar os batimentos e da suavização aplicada aos parâmetros fisiológicos contribuiu para reduzir oscilações provocadas por ruídos presentes no sinal. Dessa maneira, somente leituras consideradas válidas foram armazenadas e transmitidas para a plataforma.

Os resultados obtidos demonstram que o sistema desenvolvido foi capaz de realizar as aquisições contínuas da frequência cardíaca e da saturação de oxigênio durante o período de aquisição, mantendo estabilidade nas medições, coerência entre os valores obtidos e funcionamento adequado do processamento e da transmissão proposta.

4.2 Avaliação do Monitoramento da Posição Corporal

A avaliação do monitoramento da posição corporal teve como objetivo verificar o funcionamento do algoritmo de aquisição, processamento e classificação da orientação do paciente durante o período desse tipo de monitoramento. Conforme foi descrito na Seção 3.3.3, esse módulo foi implementado utilizando uma tarefa específica executada continuamente pelo sistema FreeRTOS, responsável, pela leitura periódica dos valores de aceleração fornecidos pelo sensor MPU6050, processamento das informações e armazenamento temporário das posições identificadas para depois realizar o envio à plataforma ThingSpeak.

Para verificar a taxa real de aquisição durante os testes experimentais, foi analisado o arquivo JSON disponibilizado pela API do ThingSpeak, utilizando um intervalo selecionado de aproximadamente 271 segundos de monitoramento e 99 amostras válidas registradas. É importante salientar que esses valores do JSON são disponibilizados como dados recentes, então não é possível selecionar um número maior de tempo ou de amostras.

Taxa média de aquisição foi de 0,36 amostras por segundo:

$$T_{média} = N_{total} \div \Delta t \quad (6)$$

Intervalo médio entre amostras armazenadas foi de 2,73 segundos:

$$T_{médio} = \Delta t \div N_{total} \quad (7)$$

Considerando que o envio ocorre a cada 15 segundos, a quantidade média de amostras transmitidas por ciclo é 5,47 amostras:

$$N_{ciclo} = (N_{tot} / \Delta t) \cdot T_{trans} \quad (8)$$

Considerando que o envio das informações ocorre a cada 15 segundos, foram transmitidas, em média, aproximadamente cinco a seis posições corporais por requisição, valor bastante próximo do esperado durante o desenvolvimento do sistema. A pequena diferença em relação ao valor teórico está relacionada ao tempo

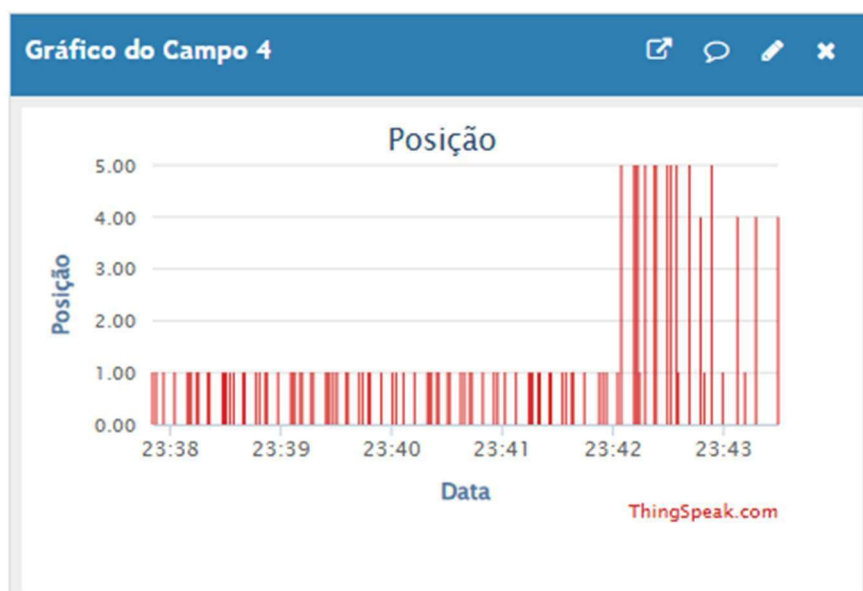
de processamento das tarefas executadas pelo FreeRTOS e ao tempo necessário para o envio das informações via rede Wi-Fi.

4.2.1 Monitoramento contínuo

Após a implementação do algoritmo para classificar a posição, foram realizados testes experimentais com o objetivo de verificar o comportamento do sistema durante o monitoramento contínuo da posição corporal. Os testes foram realizados simultaneamente aos testes dos demais sensores do sistema, permitindo avaliar o funcionamento integrado do protótipo. Durante os testes, o módulo contendo o sensor MPU650 foi fixado na região do tórax do voluntário, como pode-se observar na Figura 32, permanecendo preso ao corpo durante todo o período de aquisição dos dados. Enquanto isso, o ESP32 executava continuamente as tarefas de aquisição, classificação e transmissão das informações para a plataforma ThingSpeak.

Inicialmente, os dados foram acompanhados diretamente pela plataforma, conforme apresentado na Figura 30.

Figura 30: Visualização em tempo real dos valores da posição armazenados na plataforma ThingSpeak.



Fonte: A autora (2026).

A Figura 30 apresenta o histórico das posições corporais identificadas pelo algoritmo e armazenadas na plataforma. Observa-se que, durante a maior parte do período analisado, o usuário permaneceu classificado na posição 1, correspondente

à barriga para cima. Nos instantes finais da aquisição, são observadas transições para outros identificadores de posição, indicando mudanças na postura do usuário durante o teste.

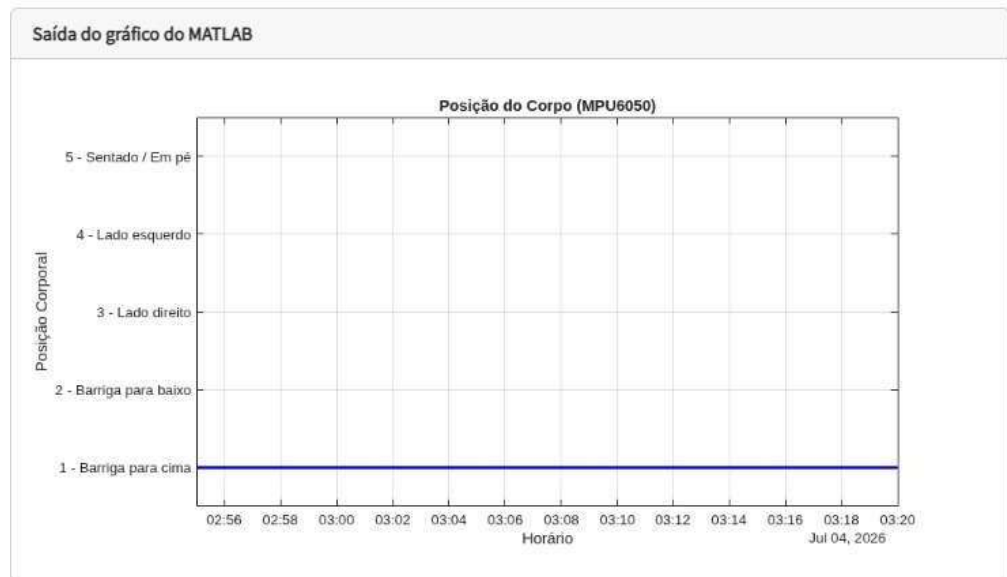
Depois, os mesmos dados foram analisados utilizando a ferramenta MATLAB que é integrada ao ThingSpeak, permitindo uma visualização temporal mais detalhada da classificação realizada pelo algoritmo. O resultado é apresentado na Figura 31.

Observa-se na Figura 31 que todas as amostras permaneceram classificadas como posição 1 (barriga para cima), durante todo o período de monitoramento, evidenciando a estabilidade do algoritmo de classificação desenvolvido. A Figura 32 exemplifica como o teste foi realizado em um voluntário.

A ausência de mudanças de posição ao longo do teste confirma que o sistema foi capaz de identificar corretamente a orientação do sensor e manter a classificação estável durante o período de monitoramento contínuo. Já na Figura 33 foi possível observar as mudanças de posições que ocorreram durante o período analisado, corroborando com a confirmação anterior, seguidas das Figuras 34 e 35 que representam as posições deitado de lado e sentado ao qual o voluntário foi submetido.

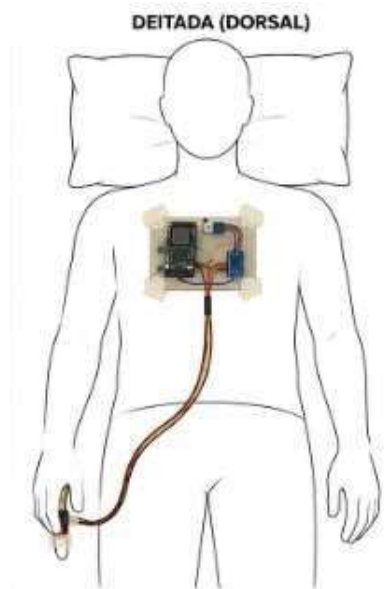
Esses resultados permitem verificar que o algoritmo respondeu às alterações de orientação detectadas pelo acelerômetro e manteve classificações estáveis durante períodos sem alterações significativas dos sinais. Entretanto, como não foi realizado um registro temporal independente da posição corporal do voluntário durante todo o período de aquisição, os resultados apresentados não permitem determinar quantitativamente a acurácia da classificação.

Figura 31: Visualização em tempo real (teste de 15min) das posições utilizando a ferramenta MATLAB Analysis do ThingSpeak.



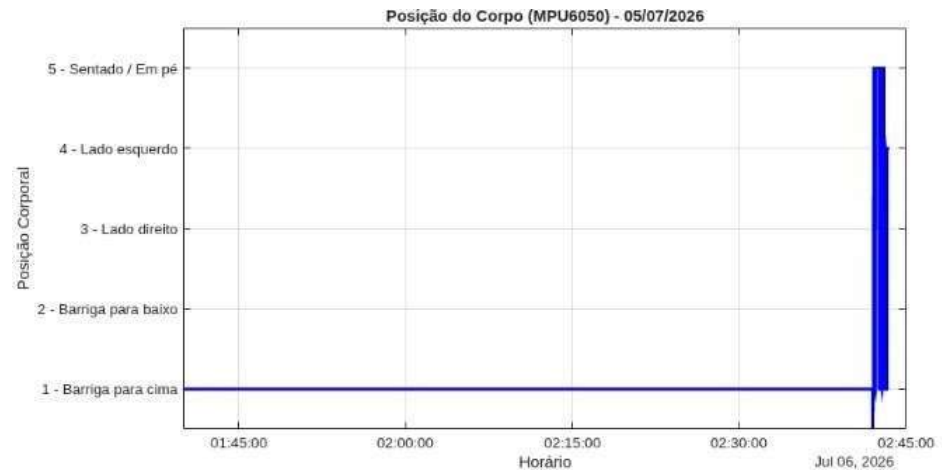
Fonte: A autora (2026).

Figura 32 - Posição teste barriga para cima



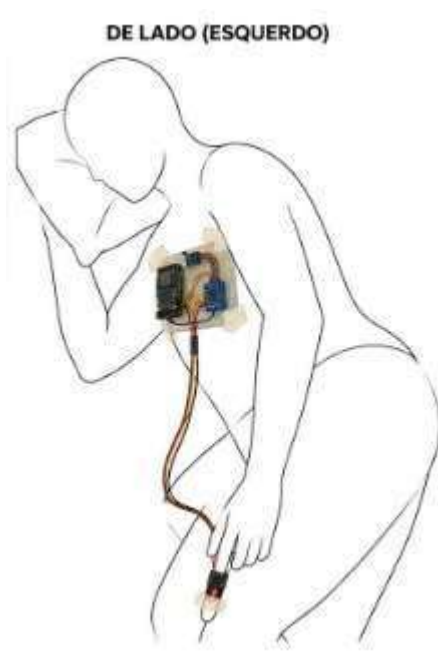
Fonte: A autora com auxílio de IA (2026).

Figura 33: Visualização em tempo real (teste de 1h) das posições utilizando a ferramenta MATLAB Analysis do ThingSpeak.



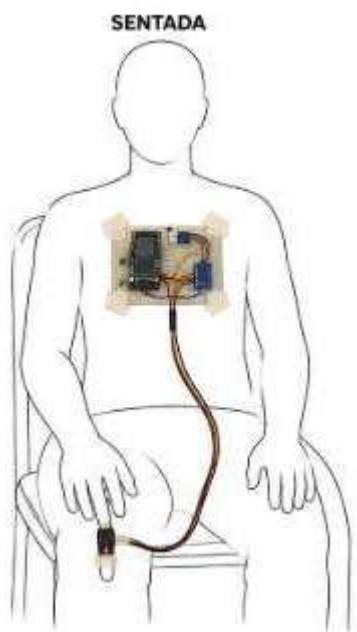
Fonte: A autora (2026).

Figura 34 - Posição teste virado de lado



Fonte: A autora com auxílio de IA (2026).

Figura 35 - Posição teste sentado



Fonte: A autora com auxílio de IA (2026).

Além disso, a inexistência de oscilações entre posições consecutivas demonstra que os limites adotados para classificação apresentam bons resultados durante as condições experimentais avaliadas. Caso esses limites fossem inadequados, pequenas variações naturais das leituras do acelerômetro poderiam provocar alternâncias frequentes entre posições, comportamento que não foi observado durante os testes.

Por fim, os resultados obtidos demonstram que o algoritmo implementado para o MPU6050 foi capaz de realizar a classificação da posição corporal de forma consistente durante todo o período de aquisição. A integração entre aquisição periódica, processamento no ESP32 e transmissão para a plataforma mostrou-se adequada para aplicações de monitoramento de forma remota da posição corporal.

4.3 Avaliação do Monitoramento do Ronco

A avaliação do sistema responsável pelo monitoramento do som teve como principal objetivo verificar o funcionamento do sistema incluindo aquisição, processamento e classificação dos sinais captados pelo microfone INMP441.

Conforme visto na Seção 3.3.4, essa etapa foi implementada por meio de uma tarefa dedicada executada de forma contínua pelo sistema FreeRTOS, permitindo assim que o ESP32 realizasse a aquisição dos sinais de áudio, processamento por meio da Transformada Rápida de Fourier (FFT), cálculo da energia sonora e a identificação de eventos associados ao ronco.

4.3.1 Testes experimentais

Para a realização dos testes experimentais e a comprovação de que o sistema cumpre com as frequências desejadas foi utilizada a ferramenta Online Tone Generator, apresentada na Figura 36, capaz de gerar sinais senoidais em frequências previamente definidas. Durante os testes, diferentes frequências foram reproduzidas próximas ao microfone enquanto o ESP32 executava continuamente a aquisição dos sinais e o processamento.

Figura 36: Ferramenta utilizada para geração de sinais senoidais durante os testes experimentais.



Fonte: Adaptado de Online Tone Generator (2026).

Embora a faixa de frequência entre 300 Hz e 1500 Hz tenha sido definida com base na literatura sobre ronco, a ferramenta Online Tone Generator foi empregada para validar experimentalmente a resposta do sistema implementado, verificando se as componentes espectrais geradas eram corretamente identificadas.

4.3.2 Armazenamento e transmissão dos dados

Após o processamento dos sinais sonoros e a classificação da presença ou ausência de ronco, os resultados obtidos não são enviados imediatamente para a plataforma ThingSpeak. Assim como ocorreu nos demais módulos do sistema, foi utilizado o mecanismo de armazenamento temporário, permitindo assim agrupar diversas amostras antes da transmissão. Na Figura 37 e com início de código na linha 545 pode-se verificar a parte do código responsável pelo armazenamento temporário das informações do INMP441.

Figura 37 - Trecho do código responsável pelo armazenamento temporário INMP441

```

545 |         if (mic_bulk_idx < MIC_BULK_SIZE) {
546 |             mic_bulk_buffer[mic_bulk_idx] = energia_norm;
547 |             mic_ronco_buffer[mic_bulk_idx] = ronco ? 1 : 0;
548 |             mic_bulk_idx++;
549 |         }

```

Fonte: A autora (2026).

Conforme o trecho verificado, para cada intervalo analisado são armazenados simultaneamente o valor correspondente à energia sonora calculada e o resultado da classificação do algoritmo, indicando presença ou ausência de ronco. Esses valores permanecem armazenados na memória do ESP32 até que seja atingido o instante programado para a transmissão via Bulk Write.

Durante o desenvolvimento do sistema, o algoritmo foi configurado para realizar uma nova aquisição aproximadamente a cada 2 segundos, enquanto a transmissão dos dados para a plataforma ocorre em intervalos de aproximadamente 15 segundos.

Para verificar a taxa real de armazenamento dos dados durante os testes experimentais, foi analisado o arquivo JSON disponibilizado pela API do ThingSpeak, foi utilizado o intervalo de 274 segundos e aproximadamente 105 amostras válidas, ressaltando que o JSON disponibilizado pela plataforma contém apenas dados recentes não sendo possível a escolha de tempo ou de amostras.

Taxa média de aquisição foi de 0,38 amostras por segundo:

$$T_{média} = N_{total} \div \Delta t \quad (9)$$

Intervalo médio entre amostras armazenadas foi de 2,69 segundos:

$$T_{\text{médio}} = \Delta t \div N_{\text{total}} \quad (10)$$

Considerando que o envio ocorre a cada 15 segundos, a quantidade média de amostras transmitidas por ciclo é 5,74 amostras:

$$N_{\text{ciclo}} = (N_{\text{tot}} / \Delta t) \cdot T_{\text{trans}} \quad (11)$$

Assim, durante cada transmissão foram enviados, em média, aproximadamente cinco a seis registros, valor bem próximo ao previsto durante o desenvolvimento do sistema. A pequena diferença pode estar relacionada ao tempo necessário para aquisição do sinal, processamento, escalonamento das tarefas FreeRTOS e transmissão das informações via rede Wi-Fi.

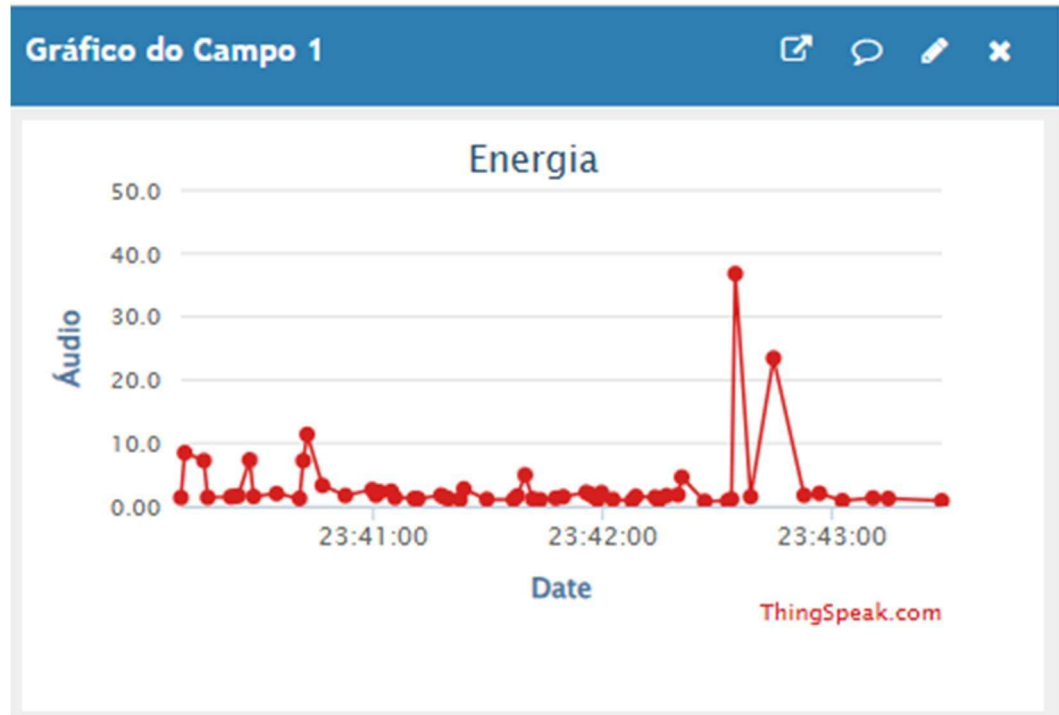
4.3.3 Monitoramento contínuo

Após a validação do sistema e da transmissão das informações para a plataforma, foram realizados ensaios experimentais simultaneamente com os outros sensores, com o objetivo de verificar o comportamento do sistema durante o monitoramento contínuo dos sinais.

Durante todo o período de aquisição, o ESP32 permaneceu executando continuamente as tarefas responsáveis pela leitura do microfone, processamento, classificação dos eventos de ronco e armazenamento e transmissão para a plataforma.

Inicialmente, os resultados foram acompanhados diretamente pela plataforma ThingSpeak, conforme apresentado na Figura 38.

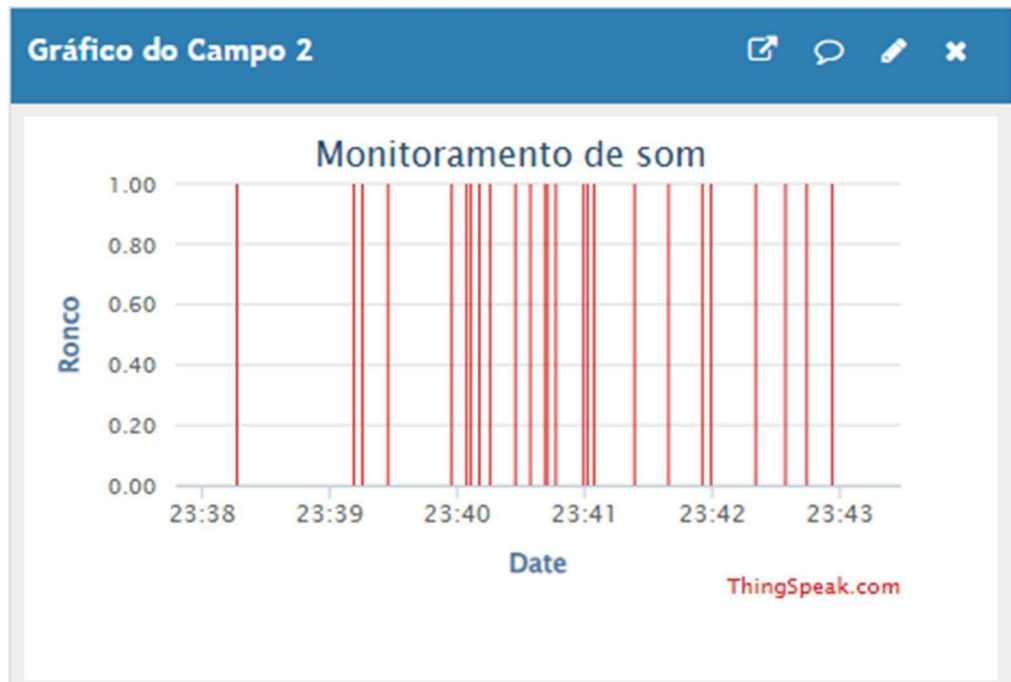
Figura 38: Visualização em tempo real dos valores da energia armazenados na plataforma ThingSpeak.



Fonte: A autora (2026).

A Figura 38 apresenta o histórico das amostras de energia sonora registradas na plataforma. As variações observadas ao longo do tempo evidenciam a presença de diferentes níveis de energia do sinal adquirido, permitindo identificar os instantes em que ocorreram eventos sonoros de maior intensidade. Os resultados demonstram o funcionamento adequado da cadeia de aquisição, processamento e transmissão dos dados, confirmando que as informações provenientes do sensor INMP441 foram corretamente registradas na plataforma em nuvem. Na Figura 39 pode-se analisar a classificação dos eventos de ronco utilizando a plataforma ThingSpeak.

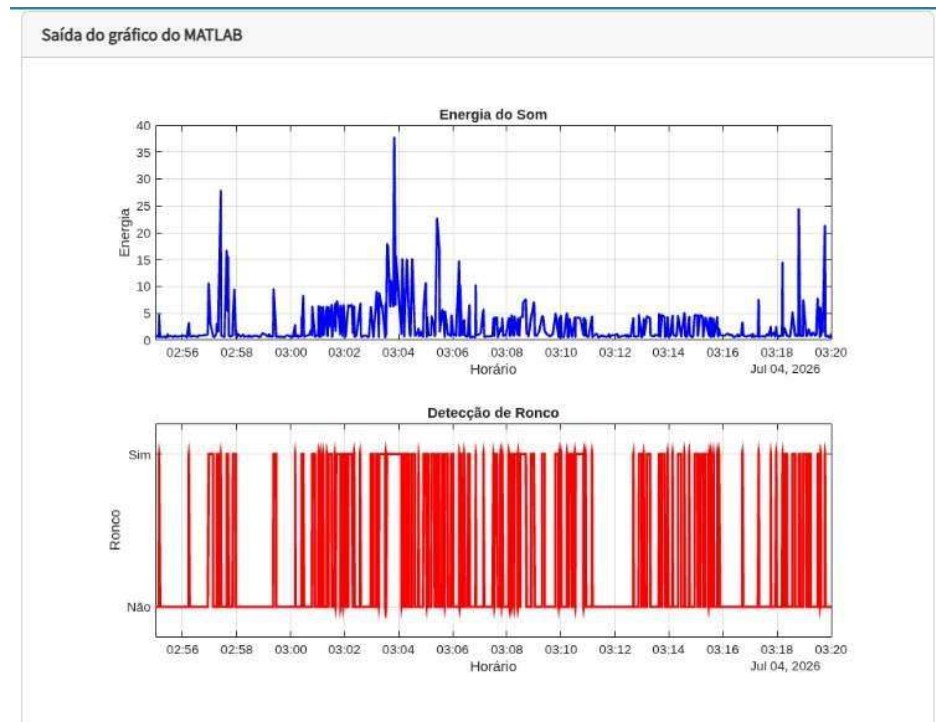
Figura 39: Visualização em tempo real dos valores do som (ronco) armazenados na plataforma ThingSpeak.



Fonte: A autora (2026).

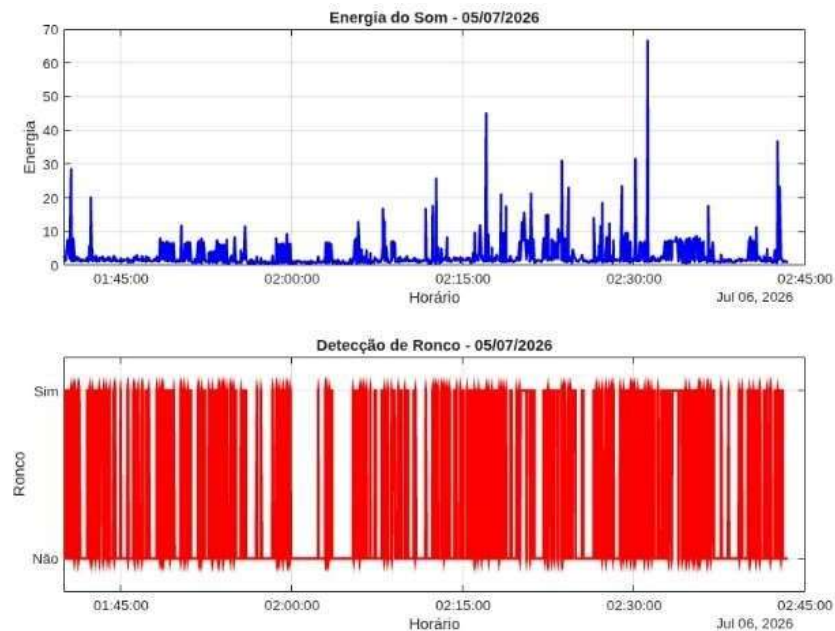
Verifica-se que o algoritmo alterna entre os estados 0 e 1, indicando respectivamente ausência ou presença de ronco durante os intervalos analisados. Posteriormente, os mesmos dados foram analisados utilizando a ferramenta MATLAB integrada ao ThingSpeak, permitindo uma visualização mais detalhada dos sinais adquiridos. Na Figura 40 e na Figura 41 verifica-se a visualização em tempo real da energia e detecção de roncos utilizando a ferramenta MATLAB Analysis do ThingSpeak nos períodos de 15 minutos e 1 hora.

Figura 40: Visualização em tempo real (teste de 15min) da energia e detecção de ronco utilizando a ferramenta MATLAB Analysis do ThingSpeak.



Fonte: A autora (2026).

Figura 41: Visualização em tempo real (teste de 1h) da energia e detecção de ronco utilizando a ferramenta MATLAB Analysis do ThingSpeak.



Fonte: A autora (2026).

A análise apresentada na Figura 40 e na Figura 41 evidencia uma relação direta entre o aumento da energia sonora e a ocorrência dos eventos classificados

como ronco, sempre que a energia calculada ultrapassou o limiar estabelecido, o algoritmo classificou como um intervalo de ronco.

Esse comportamento demonstra que a FFT utilizada foi capaz de destacar as componentes espectrais presentes na faixa de frequências analisadas durante os testes, permitindo distinguir períodos com maior energia espectral daqueles em que apenas sinais de baixa intensidade ou ruídos estavam presentes.

4.3.4 Avaliação da comunicação com a plataforma

Durante os testes iniciais do sistema foram realizados testes com diferentes configurações do mecanismo Bulk Write, buscando aumentar a quantidade de amostras transmitidas em cada requisição HTTP para a plataforma ThingSpeak. O objetivo dessa modificação era aumentar o número de amostras recebidas pela plataforma, aumentando a eficiência da transmissão dos dados.

Entretanto, durante os testes verificou-se que o aumento da quantidade de informações enviadas em cada pacote ocasionava falhas de comunicação com a plataforma. A Figura 42 apresenta um trecho do monitor serial obtido durante esses testes.

Figura 42: Monitor serial durante os testes com maior quantidade de amostras.

```
I (247602) MAIN: [MAX] BPM=74.2 | SpO2=91.9% (BULK[8])
I (248092) MAIN: -----
I (248092) MAIN: [BULK] Enviando pacotes para ThingSpeak (HTTP)
I (248382) MAIN: [MPU] X=-2752 Y=-84 Z=15356 | Pos=1 (BULK[10])
I (249602) MAIN: [MAX] BPM=77.3 | SpO2=88.4% (BULK[9])
I (249882) MAIN: [MPU] X=-2832 Y=-36 Z=15448 | Pos=1 (BULK[10])
I (251382) MAIN: [MPU] X=-2848 Y=-48 Z=15600 | Pos=1 (BULK[10])
I (251602) MAIN: [MAX] BPM=74.8 | SpO2=89.0% (BULK[10])
I (251882) MAIN: [MIC] Energia=11.44 | Freq=156Hz | Ronco=SIM (BULK[15])
I (252882) MAIN: [MPU] X=-2928 Y=-76 Z=15484 | Pos=1 (BULK[10])
I (254382) MAIN: [MPU] X=-2828 Y=-150 Z=15428 | Pos=1 (BULK[10])
W (254802) HTTP_CLIENT: Connection timed out before data was ready!
E (254802) MAIN: [BULK] MIC: Falha no envio! (ESP_ERR_HTTP_EAGAIN)
I (255882) MAIN: [MPU] X=-3076 Y=-8 Z=15644 | Pos=1 (BULK[10])
I (256922) MAIN: [MIC] Energia=24.58 | Freq=375Hz | Ronco=SIM (BULK[3])
I (257382) MAIN: [MPU] X=-2812 Y=-24 Z=15504 | Pos=1 (BULK[10])
I (258882) MAIN: [MPU] X=-2812 Y=-44 Z=15376 | Pos=1 (BULK[10])
I (260382) MAIN: [MPU] X=-2792 Y=-148 Z=15528 | Pos=1 (BULK[10])
I (261882) MAIN: [MPU] X=-2792 Y=-64 Z=15512 | Pos=1 (BULK[10])
I (261962) MAIN: [MIC] Energia=3.09 | Freq=16Hz | Ronco=NAO (BULK[8])
```

Fonte: A autora (2026).

Conforme apresentado na Figura 42, observa-se a ocorrência da mensagem *Connection timed out before data was ready*, indicando que a requisição HTTP não foi concluída dentro do tempo limite estabelecido pela biblioteca utilizada no ESP-IDF.

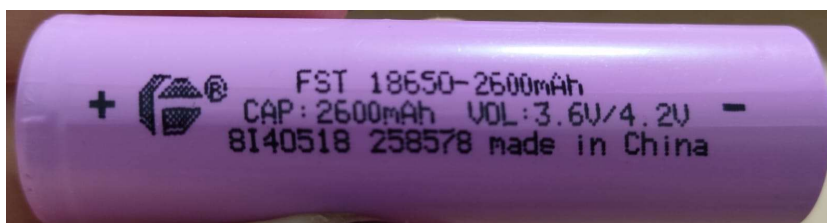
Após essa etapa, foi realizada uma nova configuração do mecanismo Bulk Write, reduzindo a quantidade de amostras enviadas por tempo de transmissões, essa configuração apresentou funcionamento estável durante os outros testes experimentais, eliminando os erros de comunicação observados anteriormente e permitindo o armazenamento contínuo na plataforma.

Com base nos resultados obtidos, optou-se por manter uma configuração que priorizasse a confiabilidade da transmissão em detrimento da quantidade máxima de amostras por pacote. Essa decisão garantiu o funcionamento contínuo do sistema durante todo o período de monitoramento, evitando perdas de dados causadas por falhas na comunicação com a plataforma.

4.4 Fonte de alimentação

Para possibilitar o funcionamento do sistema de forma independente da alimentação fornecida pela porta USB do computador, foi desenvolvido um circuito de alimentação usando baterias 18650, como mostrado na Figura 35, um módulo de gerenciamento de carga BMS e um regulador de tensão ajustável LM2596 Step Down.

Figura 43: Modelo de bateria utilizada.



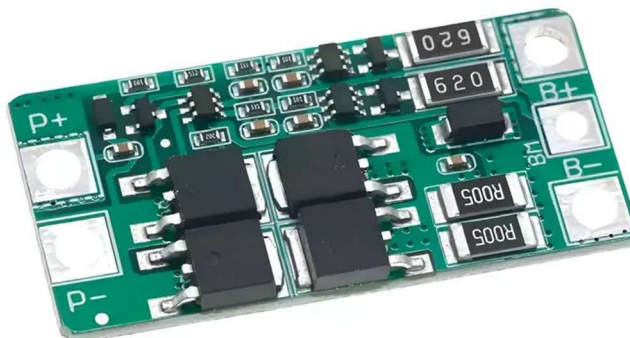
Fonte: A autora (2026).

As baterias utilizadas possuem tensão nominal de 3.6 V e capacidade de 2600 mAh, sendo empregadas duas unidades conectadas em série para fornecer tensão suficiente ao regulador de alimentação.

O módulo BMS foi responsável pela proteção das baterias durante os processos de carga e descarga, realizando proteção contra sobretensão, subtensão, sobrecorrente e curto-circuito. Além disso, ao interromper a alimentação em condições anormais de operação, o BMS também contribui para a proteção dos circuitos eletrônicos conectados ao sistema de alimentação. Na Figura 36 pode-se verificar o

modelo do módulo BMS utilizado no sistema e na Figura 37 o modelo do regulador de tensão LM2596 Step Down.

Figura 44: Modelo do módulo BMS utilizado.



Fonte: Mercado Livre (2026).

Figura 45: Regulador de tensão LM2596 Step Down.

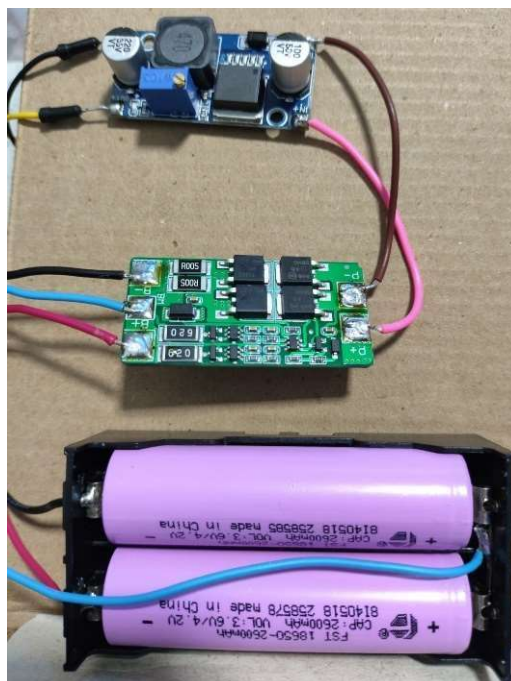


Fonte: MakerHero (2026).

Após o módulo BMS, a tensão é aplicada ao regulador LM2596, responsável pela redução e estabilização da tensão utilizada para alimentar o ESP32 e os sensores empregados no sistema.

Logo após a seleção dos componentes, foi realizada a montagem do circuito de alimentação. As duas baterias foram conectadas em série e após conectadas ao módulo BMS, cuja saída foi ligada ao regulador LM2596, a saída regulada alimentou diretamente o ESP32 e os demais módulos eletrônicos do sistema. Foram medidos os valores de tensão das partes, na saída das baterias foi encontrada uma tensão de aproximadamente 7.4 V, na saída do módulo BMS foi medido o mesmo valor de 7.4 V, enquanto que na saída do regulador (e entrada do ESP) foi encontrada uma tensão de 5.3 V, pode-se observar este circuito montado na Figura 38.

Figura 46: Sistema de alimentação montado.



Fonte: A autora (2026).

Após a validação do circuito de alimentação, o conjunto foi integrado ao restante do protótipo, responsável pela aquisição dos sinais fisiológicos. Durante os testes experimentais permaneceram ativos simultaneamente o ESP32, o sensor MAX30100, o acelerômetro MPU6050, o microfone INMP441, além do módulo Wi-Fi utilizado para a comunicação com a plataforma.

Com o circuito de alimentação completamente montado e integrado ao protótipo, foi realizado um teste de forma experimental com o objetivo de determinar a autonomia do sistema em operação contínua. Inicialmente as duas baterias foram completamente carregadas antes de ligadas ao sistema.

Em seguida, ocorreu a energização do sistema e iniciou-se assim o teste de funcionamento contínuo, durante este período todas as tarefas do sistema continuaram rodando simultaneamente como o programado.

O sistema permaneceu operando continuamente até que a tensão das baterias atingisse o limite mínimo de operação, ocasionando o desligamento da alimentação. Durante esse teste foi observada uma autonomia de aproximadamente 5 horas e 20 minutos de funcionamento ininterrupto.

O valor de autonomia menor do que o esperado se calculado teoricamente que seria de mais de 6 horas, pode ser justificada principalmente pelo estado de conservação das baterias utilizadas. As células empregadas não eram novas e já apresentavam sinais de desgaste decorrentes de usos anteriores, reduzindo sua capacidade efetiva de armazenamento de energia em relação ao valor nominal especificado pelo fabricante, para além do fato de que provavelmente as células não eram originais.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo desenvolver um sistema para aquisição de dados fisiológicos relacionados à Apneia Obstrutiva do Sono (AOS), utilizando o microcontrolador ESP32, e os sensores MA30100, MPU6050 e o microfone INMP441, responsáveis pela aquisição da frequência cardíaca, saturação de oxigênio, posição corporal e sinais sonoros respectivamente.

Ao longo do desenvolvimento foram utilizados diversos módulos de hardware e desenvolvido o software do protótipo. O sistema foi implementado utilizando o sistema operacional FreeRTOS para execução paralela das tarefas de aquisição, processamento e transmissão dos dados, permitindo que os diferentes sensores operassem de forma independente, mas sincronizada. Além disso, foi desenvolvido um sistema de comunicação utilizando a plataforma ThingSpeak, possibilitando o armazenamento em nuvem e visualização das informações adquiridas em tempo real.

Os resultados dos testes experimentais demonstraram que o sistema desenvolvido foi capaz de realizar continuamente a aquisição dos sinais provenientes dos três sensores utilizados. Para o sensor MAX30100, os valores obtidos de frequência cardíaca e saturação de oxigênio apresentaram comportamento coerente quando comparados a um oxímetro de pulso comercial, evidenciando o funcionamento adequado do algoritmo implementado. O acelerômetro MPU6050 permitiu identificar corretamente diferentes posições corporais adotadas, enquanto o microfone INMP441 possibilitou a identificação de momentos associados ao ronco por meio da análise espectral do sinal.

Também foram avaliadas as estratégias de armazenamento e transmissão dos dados, a implementação do mecanismo *Bulk Write* mostrou-se adequada para atender às limitações da plataforma ThingSpeak, permitindo o agrupamento de diversas amostras em uma única requisição HTTP, reduzindo a quantidade de transmissões realizadas pelo ESP32 e aumentando a eficiência da comunicação. Os testes realizados demonstraram ainda que a plataforma MATLAB Analysis integrada ao ThingSpeak constitui uma ferramenta útil para análise gráfica dos sinais adquiridos, permitindo melhor visualização da evolução temporal dos parâmetros registrados.

Em relação ao sistema de alimentação, foi desenvolvido um circuito composto por duas baterias de íons de lítio do tipo 18650, módulo BMS e regulador de tensão ajustável LM2596 Step Down. Os testes realizados indicaram uma autonomia aproximada de cinco horas de funcionamento contínuo. Embora esse valor tenha sido inferior à estimativa teórica, o resultado é compatível com as condições em que os testes foram conduzidos, cabe destacar que os testes foram realizados utilizando baterias previamente utilizadas, condição que influenciou diretamente a autonomia obtida. Dessa forma, espera-se que a utilização de baterias novas, associada à otimização do consumo energético do sistema, possibilite ampliar o tempo de funcionamento para valores mais próximos de um período completo de sono.

De maneira geral, os objetivos propostos neste trabalho foram alcançados, sendo desenvolvido um protótipo funcional capaz de adquirir, processar, armazenar e transmitir continuamente dados fisiológicos relacionados à AOS. Os resultados obtidos demonstraram o funcionamento de forma integrada do hardware e do software do sistema, evidenciando a viabilidade técnica da solução proposta para aquisição de informações fisiológicas e disponibilização desses dados em uma plataforma de armazenamento em nuvem.

5.1 Sugestões para trabalhos futuros

Como continuidade deste trabalho, uma possibilidade de evolução consiste na utilização de algoritmos mais avançados para análise dos sinais sonoros, empregando técnicas de aprendizado de máquina ou inteligência artificial para classificação automática de eventos relacionados ao rondo e à apneia, reduzindo a ocorrência de falsos positivos provocados por ruídos ambientais.

Também se recomenda a realização de estudos envolvendo um número maior de participantes, que neste projeto foram de 4 pessoas, permitindo comparar os dados adquiridos pelo sistema com exames clínicos de referência, como a polissonografia, contribuindo para avaliar quantitativamente o desempenho da solução desenvolvida.

Como evolução da plataforma de aquisição, pode-se desenvolver uma aplicação na web dedicada para visualização das informações, permitindo organizar os dados de forma mais intuitiva, gerar relatórios automáticos e facilitar a análise dos registros obtidos durante o monitoramento.

Por fim, recomenda-se a integração de novos sensores fisiológicos, como a cinta piezoelétrica, permitindo ampliar a quantidade de parâmetros adquiridos e tornando o sistema uma plataforma mais completa para aquisição de dados relacionados aos distúrbios respiratórios do sono.

REFERÊNCIAS

- ARDUINO KIT PROJECT. **MPU6050 Accelerometer and Gyroscope Sensor Arduino Tutorial**. Disponível em: <https://arduinokitproject.com/mpu6050-accel-gyro-arduino-tutorial/>. Acesso em: 07 jul. 2026.
- ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A Survey. **Computer Networks**, [S. l.], v. 54, n. 15, p. 2787-2805, 2010.
- BENJAFIELD, A. V. et al. Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis. **The Lancet Respiratory Medicine**, [S. l.], v. 7, n. 8, p. 687-698, ago. 2019.
- BERRY, R. B. et al. Rules for Scoring Respiratory Events in Sleep: Update of the 2007 AASM Manual for the Scoring of Sleep and Associated Events. **Journal of Clinical Sleep Medicine**, [S. l.], v. 8, n. 5, p. 597-619, 2012. DOI: <https://doi.org/10.5664/jcsm.2172>.
- CABELLINO, L. F. et al. Privação do Sono e Saúde Mental: Impactos Neurobiológicos e Comportamentais. **Brazilian Journal of Implantology and Health Sciences**, [S. l.], v. 7, n. 4, p. 660-672, 2025. Disponível em: <https://doi.org/10.36557/2674-8169.2025v7n4p660-672>. Acesso em: 22 mar. 2026.
- COOLEY, James W.; TUKEY, John W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of Computation**, v. 19, n. 90, p. 297-301, abr. 1965. DOI: 10.1090/S0025-5718-1965-0178586-1.
- DONNELLY, Denis; RUST, Bert W. The Fast Fourier Transform for Experimentalists, Part I: Concepts. **Computing in Science & Engineering**, v. 7, n. 2, p. 80-88, mar./abr. 2005. DOI: 10.1109/MCSE.2005.42.
- EASYTRONICS. **MPU6050 Acelerômetro e Giroscópio de 3 Eixos**. Disponível em: <https://www.easytronics.com.br/mpu6050-acelerometro-e-giroscopio-de-3-eixos>. Acesso em: 07 jul. 2026.
- ELECTROPEAK. **INMP441 MEMS Microphone Module I2S Interface**. Disponível em: <https://electropeak.com/inmp441-mems-microphone-module-i2s-interface>. Acesso em: 06 jul. 2026.

EPSTEIN, L. J. et al. Clinical Guideline for the Evaluation, Management and Long-term Care of Obstructive Sleep Apnea in Adults. **Journal Of Clinical Sleep Medicine**, [S. l.], v. 5, n. 3, p. 263-276, jun. 2009.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet**: 2.4 GHz Wi-Fi + Bluetooth + Bluetooth LE SoC. v5.2. Shanghai: Espressif Systems, 2024. Disponível em: <https://www.espressif.com/en/support/documents/technical-documents>. Acesso em: 15 jun. 2026.

ESPRESSIF SYSTEMS. **ESP32 Technical Reference Manual**. v5.3. Shanghai: Espressif Systems, 2024. Disponível em: <https://www.espressif.com/en/support/documents/technical-documents>. Acesso em: 15 jun. 2026.

EUROPEAN RESPIRATORY SOCIETY. ERS technical standards for respiratory polygraphy. **European Respiratory Journal**, [S. l.], v. 61, n. 1, 2023. DOI: <https://doi.org/10.1183/13993003.00422-2022>.

FREERTOS. **RTOS Fundamentals**. [S. l.], 2026. Disponível em: <https://www.freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals>. Acesso em: 15 jun. 2026.

GARDNER, J. W.; VARADAN, V. K.; AWADELKARIM, O. O. **Microsensors, MEMS, and Smart Devices**. Chichester: John Wiley & Sons, 2001.

GUBBI, J. et al. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, [S. l.], v. 29, n. 7, p. 1645-1660, 2013.

HARRIS, Fredric J. On the use of windows for harmonic analysis with the discrete Fourier transform. **Proceedings of the IEEE**, v. 66, n. 1, p. 51-83, jan. 1978. DOI: 10.1109/PROC.1978.10837.

HO, M. L.; BRASS, S. D. Obstructive sleep apnea. **Neurology International**, Basel, v. 3, n. 3, e15, 2011. DOI: <https://doi.org/10.4081/ni.2011.e15>. Disponível em: <https://www.pagepressjournals.org/ni/article/view/ni.2011.e15>. Acesso em: 29 jun. 2026.

INTERNET OF THINGS WIKI. **Internet of Things Definition**. [S. l.], 2024. Disponível em: <https://internetofthingswiki.com/internet-of-things-definition/>. Acesso em: 29 jun. 2026.

INVENSENSE, INC. **INMP441**: Omnidirectional Microphone with Bottom Port and I2S Digital Output. San Jose, CA: InvenSense, Inc., 2014. Data Sheet, Rev. 1.0, p. 1-21, fev. 2014.

INVENSENSE, INC. **MPU-6000 and MPU-6050 Register Map and Descriptions**. Sunnyvale, CA: InvenSense, Inc., 2013. Data Sheet, Rev. 4.2, p. 1-46, ago. 2013.

ISLAM, S. M. R. et al. The Internet of Things for Health Care: A Comprehensive Survey. **IEEE Access**, [S. l.], v. 3, p. 678-708, 2015.

JONAS, D. E. et al. Screening for Obstructive Sleep Apnea in Adults: Evidence Report and Systematic Review for the US Preventive Services Task Force. **JAMA**, [S. l.], v. 317, n. 4, p. 415-433, 2017. DOI: <https://doi.org/10.1001/jama.2016.19635>.

KAPUR, V. K. et al. Clinical Practice Guideline for Diagnostic Testing for Adult Obstructive Sleep Apnea: An American Academy of Sleep Medicine Clinical Practice Guideline. **Journal of Clinical Sleep Medicine**, [S. l.], v. 13, n. 3, p. 479-504, 2017. DOI: <https://doi.org/10.5664/jcsm.6506>.

LEE, L.-A. et al. Snoring Sounds Predict Obstruction Sites and Surgical Response in Patients with Obstructive Sleep Apnea Hypopnea Syndrome. **Scientific Reports**, [S. l.], v. 6, n. 30629, p. 1-11, 2016. Available from: <https://doi.org/10.1038/srep30629>.

MAKERHERO. **Placa ESP32 DevKitC V4**. Disponível em: <https://www.makerhero.com/produto/placa-esp32-devkitc-v4/>. Acesso em: 06 jul. 2026.

MAKERHERO. **Regulador de Tensão LM2596 Conversor DC-DC Step-Down**. Disponível em: <https://www.makerhero.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>. Acesso em: 06 jul. 2026.

MATHWORKS. **Configure Accounts and Channels**. Natick, MA: MathWorks, 2024. Disponível em: <https://www.mathworks.com/help/thingspeak/users.html>. Acesso em: 11 jul. 2026.

MATHWORKS. **ThingSpeak Documentation**. Natick, MA: MathWorks, 2024. Disponível em: <https://www.mathworks.com/help/thingspeak/>. Acesso em: 11 jul. 2026.

MATHWORKS. **Write Data to Channel**. Natick, MA: MathWorks, 2024. Disponível em: <https://www.mathworks.com/help/thingspeak/write-data.html>. Acesso em: 11 jul. 2026.

MAXIM INTEGRATED. **MAX30100**: Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health. San Jose, CA: Maxim Integrated, 2014. Data Sheet, Rev. 0, p. 1-29, set. 2014.

MERCADO LIVRE. **Placa de Proteção BMS 2S 10A com Balanceamento LiFePO4**. Disponível em: <https://www.mercadolivre.com.br/placa-de-protecao-bms-2s-10a-c-balanceamento-lifepo4-72v/up/MLBU3412857637>. Acesso em: 07 jul. 2026.

MIKROELEKTRONIKA. **Heart rate click**. [S. l.], 2026. Disponível em: <https://www.mikroe.com/heart-rate-click>. Acesso em: 20 jun. 2026.

MOHAMMED, Zakriya; ELFADEL, Ibrahim M.; RASRAS, Mahmoud. A Review of Monolithic Multi-Axis Capacitive MEMS Accelerometers. **Micromachines**, v. 9, n. 11, p. 602, 2018.

NXP SEMICONDUCTORS. **UM10204**: I2C-bus specification and user manual. Eindhoven: NXP Semiconductors, 2021. Rev. 7.0, p. 1-64, out. 2021.

ONLINE TONE GENERATOR. **Free Online Tone Generator**. Disponível em: <https://onlinetonegenerator.com/>. Acesso em: 10 jun. 2026.

PEVERNAGIE, D.; AARTS, R. M.; DE MEYER, M. The Acoustics of Snoring. **Sleep Medicine Reviews**, [S. l.], v. 14, n. 2, p. 131-144, 2010. DOI: <https://doi.org/10.1016/j.smrv.2009.06.002>.

PHILIPS SEMICONDUCTORS. **I2S bus specification**. Eindhoven: Philips Semiconductors, 1986. p. 1-7, fev. 1986.

SANKRI-TARBICHI, A. G. Obstructive Sleep Apnea-Hypopnea Syndrome: Etiology and Diagnosis. **Avicenna Journal of Medicine**, [S. l.], v. 2, n. 1, p. 3-8, 2012. DOI: <https://doi.org/10.4103/2231-0770.94806>.

SEQUEIRA, V. C. C.; BANDEIRA, P. M.; AZEVEDO, J. C. M. Heart rate variability in adults with obstructive sleep apnea: a systematic review. **Sleep Science**, Rio de Janeiro, v. 12, n. 3, p. 214-221, set. 2019.

SURANI, S.; TAWEESEDT, P. Obstructive Sleep Apnea: New Perspective. **Medicina**, Kaunas, v. 59, n. 1, p. 75, dez. 2022. DOI: <https://doi.org/10.3390/medicina59010075>. Disponível em: <https://www.mdpi.com/1648-9144/59/1/75>. Acesso em: 20 mar. 2026.

VALVANO, J. W. **Embedded Systems**: Introduction to ARM® Cortex™-M Microcontrollers. Volume 1. Austin: Self-published, 2012. Disponível em: <https://users.ece.utexas.edu/~valvano/Volume1/E-Book/>.

VALVANO, J. W. **Embedded Systems**: Real-Time Operating Systems for ARM® Cortex™-M Microcontrollers. Volume 3. 4. ed. Austin: Self-published, 2017.

WORLEY, S. L. The Extraordinary Importance of Sleep: The Detrimental Effects of Inadequate Sleep on Health and Public Safety Drive an Explosion of Sleep Research. **Pharmacy and Therapeutics**, [S. l.], v. 43, n. 12, p. 758-763, dez. 2018. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6281147/>. Acesso em: 22 mar. 2026.

APÊNDICE A - Repositório digital de códigos-fontes do sistema

Link de acesso ao repositório:

https://drive.google.com/drive/folders/1m6MC9KHPkuoxVXZDg2t4GqHCO2MT-3XB?usp=drive_link