

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* BETIM  
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Rodrigo de Souza Gonçalves

**IMPLEMENTAÇÃO DE UMA CENTRAL DE ALARME HOSPITALAR  
COM ESP32 LORA E RASPBERRY PI**

Betim  
2025

RODRIGO DE SOUZA GONÇALVES

**IMPLEMENTAÇÃO DE UMA CENTRAL DE ALARME HOSPITALAR  
COM ESP32 LORA E RASPBERRY PI**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

**Orientador:** Prof. Me. Virgil Del Duca Almeida

Betim  
2025

## FICHA CATALOGRÁFICA

G635i Gonçalves, Rodrigo de Souza

Implementação de uma central de alarme hospitalar com ESP32  
LoRa e Raspberry Pi / Rodrigo de Souza Gonçalves. – 2025.

84 f.: il.

Trabalho de conclusão de curso (Bacharelado em  
Engenharia de Controle e Automação) - Instituto Federal de Educação,  
Ciência e Tecnologia de Minas Gerais, Campus Betim, 2025.

Orientação: Prof. Me. Virgil Del Duca Almeida

1. Sistemas de alarme eletrônico. 2. Sensores. 3. Microcontroladores.  
4. Engenharia de Controle e Automação. I. Gonçalves, Rodrigo de  
Souza. II. Título.

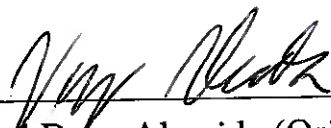
CDU: 681.51

Rodrigo de Souza Gonçalves

**IMPLEMENTAÇÃO DE UMA CENTRAL DE ALARME HOSPITALAR  
COM ESP32 LORA E RASPBERRY PI**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Betim, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 29 / 01 / 2025 pela banca examinadora:



---

Prof. Me. Virgil Del Duca Almeida (Orientador) - IFMG - Campus Betim



---

Prof. Dr. Leandro Freitas de Abreu - IFMG - Campus Betim



---

Prof. Dr. Arthur Hermano Resendo Rosa - IFMG - Campus Betim

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus, por me conceder força, perseverança e saúde para superar cada etapa desta jornada.

À minha esposa, pelo apoio incondicional, pelas palavras de encorajamento e pelo amor que me sustentou nos momentos mais desafiadores. Sua presença foi essencial para que eu pudesse chegar até aqui.

À minha família, que sempre acreditou em mim, oferecendo suporte e motivação.

Ao meu orientador, Virgil, expresso gratidão pela paciência e pelas valiosas contribuições que enriqueceram o desenvolvimento deste trabalho. Sua orientação foi fundamental para a realização deste projeto.

Agradeço também a todos os professores que, ao longo do curso, transmitiram conhecimento e despertaram em mim a vontade de aprender continuamente.

## RESUMO

Este trabalho tem como objetivo desenvolver uma central de alarmes para monitoramento em tempo real de parâmetros críticos de gases medicinais, como pressão de vácuo, ar comprimido e oxigênio. A proposta visa fornecer uma solução acessível e eficiente, utilizando tecnologias de hardware e software. O sistema foi desenvolvido com duas placas ESP32 LoRa, uma configurada para transmissão e a outra para recepção, junto com um Raspberry Pi 4, que atua como concentrador dos dados recebidos via ESP32 LoRa. Os dados são processados através do Node-RED, uma ferramenta de desenvolvimento visual que permite a integração de dispositivos e sensores. Fluxos foram programados no Node-RED para capturar, armazenar e analisar os valores de entrada, determinando situações de alerta quando os limites predefinidos são excedidos. Os métodos empregados incluem a configuração de fluxos no Node-RED para o processamento das informações recebidas e a utilização de lógica condicional para verificar os estados dos sensores e a definição de critérios de acionamento de alarmes. Os dados foram armazenados no Google Sheets e o sistema automaticamente envia notificações via Telegram quando os valores dos gases medicinais estão fora dos parâmetros de controle. Os resultados obtidos indicam que o sistema é capaz de monitorar continuamente os parâmetros e emitir alertas em situações críticas, como ausência total de fornecimento de gás ou pressão inadequada. O sistema mostrou-se confiável em cenários simulados, evidenciando sua precisão e eficiência na ativação dos alarmes. Como consideração final, o sistema apresenta-se como uma alternativa viável para o monitoramento de sistemas de gases hospitalares, contribuindo para a segurança e a operação eficiente de ambientes críticos. Estudos futuros podem abranger a ampliação do sistema para acompanhar outras variáveis, aumentando sua aplicabilidade e funcionalidade.

**Palavras-chave:** Monitoramento; Gases Medicinais; ESP32 LoRa; Raspberry Pi; Alarmes; Central.

## ABSTRACT

This work aims to develop an alarm center for real-time monitoring of critical parameters of medical gases, such as vacuum pressure, compressed air and oxygen. The proposal aims to provide an affordable and efficient solution, using hardware and software technologies. The system was developed with two ESP32 LoRa boards, one configured for transmission and the other for reception, together with a Raspberry Pi 4, which acts as a hub for data received via ESP32 LoRa. The data is processed through Node-RED, a visual development tool that allows the integration of devices and sensors. Flows were programmed in Node-RED to capture, store and analyze the input values, determining alert situations when predefined limits are exceeded. The methods employed include the configuration of flows in Node-RED to process the information received and the use of conditional logic to check the states of the sensors and define alarm triggering criteria. The data was stored in Google Sheets and the system automatically sends notifications via Telegram when the values of the medical gases are outside the control parameters. The results obtained indicate that the system is capable of continuously monitoring parameters and issuing alerts in critical situations, such as a total lack of gas supply or inadequate pressure. The system proved to be reliable in simulated scenarios, demonstrating its accuracy and efficiency in activating alarms. As a final consideration, the system presents itself as a viable alternative for monitoring hospital gas systems, contributing to the safety and efficient operation of critical environments. Future studies may include expanding the system to monitor other variables, increasing its applicability and functionality.

**Keywords:** Monitoring; Medical Gases; ESP32 LoRa; Raspberry Pi; Alarms; Central.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Tubulação de gases medicinais . . . . .	17
Figura 2 – Comunicação I2C entre mestre e escravo . . . . .	23
Figura 3 – Sensor de pressão XGZP6847A . . . . .	25
Figura 4 – Arquitetura central alarmes com Esp32 LoRa e Raspberry Pi . . . . .	28
Figura 5 – Mapa pinos do Esp32 LoRa V3 . . . . .	30
Figura 6 – Antenas LoRa 915 MHz . . . . .	31
Figura 7 – Curva característica ADC ESP2 . . . . .	34
Figura 8 – Teste experimental ADC ESP32 LoRa . . . . .	35
Figura 9 – Curva experimental ESP32 LoRa . . . . .	36
Figura 10 – Erro percentual entre curva teórica e curva real . . . . .	37
Figura 11 – ADC descalibrado limitado 200mV a 3000mV . . . . .	38
Figura 12 – ADC calibrado método regressão polinomial limitado 200mV a 3000mV . . . . .	39
Figura 13 – ADC calibrado método regressão linear limitado 200mV a 3000mV . . . . .	40
Figura 14 – ADC calibrado método biblioteca de calibração da Espressif limitado 200mV a 3000mV . . . . .	41
Figura 15 – Curva de pressão XGZP6847A . . . . .	43
Figura 16 – Teste funcional sensor de pressão XGZP6847A . . . . .	44
Figura 17 – Circuito divisor de tensão . . . . .	45
Figura 18 – Placa PCB em desenvolvimento . . . . .	50
Figura 19 – Placa de circuito impresso final . . . . .	52
Figura 20 – Montagem final central de alarmes . . . . .	53
Figura 21 – Central de alarmes módulo transmissor . . . . .	54
Figura 22 – Central de alarmes módulo receptor . . . . .	55
Figura 23 – Central de alarmes com módulos transmissor e receptor . . . . .	56
Figura 24 – Transmissor Esp32 LoRa . . . . .	58
Figura 25 – Ambiente de programação no Node-Red . . . . .	59
Figura 26 – Dashboards central alarmes . . . . .	61
Figura 27 – Dashboard final central de alarmes . . . . .	65
Figura 28 – Notificações recebidas via Telegram . . . . .	66
Figura 29 – Base dados central de alarmes . . . . .	67
Figura 30 – Posto de enfermagem módulo transmissor . . . . .	68
Figura 31 – Manutenção módulo receptor . . . . .	69
Figura 32 – Alcance da comunicação com ESP32 LoRa . . . . .	70
Figura 33 – Esp 32 LoRa transmissor e receptor . . . . .	72

## **LISTA DE QUADROS**

Quadro 1 – Especificações gerais do ESP32 LoRa V3 . . . . .	29
Quadro 2 – Parâmetros do sensor de pressão e ADC . . . . .	49
Quadro 3 – Especificações do Raspberry Pi 4 . . . . .	62
Quadro 4 – Lista de materiais utilizados no projeto central de alarmes . . . . .	73
Quadro 5 – Softwares utilizados no projeto . . . . .	74

## LISTA DE TABELAS

Tabela 1 – Faixas de frequências permitidas para LoRa no Brasil . . . . .	20
Tabela 2 – Comparativo entre antena 1 (integrada ao módulo ESP32) e antena 2 (TX915-JKS-IPX20) . . . . .	32
Tabela 3 – Teste funcional do sensor de pressão XGZP6847A . . . . .	46

## **LISTA DE ABREVIATURAS E SIGLAS**

CSS	Chirp Spread Spectrum
IoT	Internet das Coisas
ISM	Industrial Scientific and Medical
ANATEL	Agência Nacional de Telecomunicações
I2C	Inter-Integrated Circuit
TI	Texas Instruments
TCP	Transmission Control Protocol
IP	Internet Protocol
IFMG	Instituto Federal de Minas Gerais
WDT	Watchdog Timer
UART	Universal Asynchronous Receiver-Transmitter
GSheet	Google Sheets
IBM	International Business Machines Corporation
RSSI	Received Signal Strength Indicator
ADC	Analog to Digital Converter

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Justificativa</b>	<b>14</b>
<b>1.2</b>	<b>Objetivos</b>	<b>15</b>
<i>1.2.1</i>	<i>Objetivo geral</i>	<i>15</i>
<i>1.2.2</i>	<i>Objetivos específicos</i>	<i>15</i>
<b>1.3</b>	<b>Organização do trabalho</b>	<b>15</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>17</b>
<b>2.1</b>	<b>Gases medicinais</b>	<b>17</b>
<b>2.2</b>	<b>Fadiga de alarmes</b>	<b>18</b>
<b>2.3</b>	<b>Internet das coisas</b>	<b>18</b>
<b>2.4</b>	<b>Introdução à tecnologia LoRa</b>	<b>19</b>
<i>2.4.1</i>	<i>Configuração dos parâmetros LoRa</i>	<i>20</i>
<b>2.5</b>	<b>Protocolo de comunicação I2C</b>	<b>22</b>
<b>2.6</b>	<b>Protocolo de rede</b>	<b>23</b>
<i>2.6.1</i>	<i>TCP-IP</i>	<i>24</i>
<b>2.7</b>	<b>Sensor de pressão XGZP6847A</b>	<b>25</b>
<b>2.8</b>	<b>Monitoramento de alarmes hospitalares</b>	<b>26</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>27</b>
<b>3.1</b>	<b>Arquitetura geral do sistema proposto</b>	<b>27</b>
<b>3.2</b>	<b>ESP32 LoRa</b>	<b>28</b>
<i>3.2.1</i>	<i>Resultado do teste de alcance da comunicação ESP32 LoRa</i>	<i>32</i>
<b>3.3</b>	<b>Linearização ADC ESP32 LoRa</b>	<b>33</b>
<i>3.3.1</i>	<i>Resultados da linearização do ADC ESP32 LoRa</i>	<i>37</i>
<b>3.4</b>	<b>Condicionamento de sinal</b>	<b>41</b>
<b>3.5</b>	<b>Sensor de pressão XGZP6847A</b>	<b>42</b>
<i>3.5.1</i>	<i>Resultado teste funcional sensor de pressão XGZP6847A</i>	<i>43</i>
<b>3.6</b>	<b>Placa de circuito impresso</b>	<b>50</b>
<i>3.6.1</i>	<i>Projeto e fabricação da PCB</i>	<i>51</i>
<b>3.7</b>	<b>Montagem do protótipo</b>	<b>53</b>

<b>3.8</b>	<b>Programação ESP32 LoRa</b> . . . . .	<b>56</b>
<b>3.9</b>	<b>Node-RED</b> . . . . .	<b>59</b>
<b>3.10</b>	<b>Raspberry Pi</b> . . . . .	<b>61</b>
<b>3.11</b>	<b>Implementação do Node-RED no Raspberry Pi</b> . . . . .	<b>63</b>
<b>3.12</b>	<b>Instalação dos módulos transmissor e receptor</b> . . . . .	<b>68</b>
<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>72</b>
<b>4.1</b>	<b>Materiais usados no projeto</b> . . . . .	<b>72</b>
<b>4.2</b>	<b>Lista de <i>software</i></b> . . . . .	<b>74</b>
<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> . . . . .	<b>75</b>
<b>5.1</b>	<b>Trabalhos futuros</b> . . . . .	<b>75</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>77</b>
	<b>APÊNDICE A – MANUAL TÉCNICO DE OPERAÇÃO E MANUTENÇÃO DO PROTÓTIPO DA CENTRAL DE ALARMES</b> . . . . .	<b>79</b>
	<b>APÊNDICE B – TERMO DE ANUÊNCIA</b> . . . . .	<b>83</b>

# 1 INTRODUÇÃO

Nos últimos anos, os avanços da tecnologia têm desempenhado um papel importante na melhoria dos sistemas de saúde, especialmente no que diz respeito à segurança e monitoramento em ambientes hospitalares. A rápida detecção e resposta as emergências são essenciais para proteger pacientes e profissionais da saúde.

Dentro do complexo do Hospital Público de Betim temos o setor de manutenção, este se localiza em um prédio anexo no mesmo quarteirão. Quando ocorre uma intercorrência nos gases medicinais, a equipe médica aciona o setor de manutenção para verificar o problema. Porém, sempre que isso ocorre, a situação já está crítica, pois os ventiladores mecânicos já estão alarmando. A pressão sobre o técnico de manutenção, nessas circunstâncias, pode o deixar paralisado, sem clareza sobre os procedimentos a serem adotados. A ausência de um monitoramento eficaz e confiável dos gases medicinais no setor de manutenção impede uma resposta rápida em situações emergenciais.

Neste cenário, este estudo propõe a implementação de uma central de alarme hospitalar com o dispositivo ESP32 LoRa que seja capaz de transferir dados por rádio frequência. A central de alarme é indispensável para assegurar a comunicação precisa de eventos críticos, como emergências médicas, incêndios ou falhas no sistema, em um ambiente hospitalar. Com a central de alarme, é possível monitorar em tempo real variações na pressão dos gases medicinais. Ao receber um alarme, a equipe responsável pode agir de forma imediata, minimizando os riscos e garantindo uma resposta rápida e eficaz quando surgirem emergências. Isso é crucial para salvar vidas e prevenir danos maiores ao patrimônio e à infraestrutura hospitalar.

## 1.1 Justificativa

Implementar uma central de alarme hospitalar utilizando microcontrolador ESP32 com a tecnologia LoRa é uma iniciativa importante, considerando a crescente demanda por sistemas eficazes de segurança e monitoramento em ambientes hospitalares.

Os sinais sonoros disparados por alarmes de dispositivos médicos têm causado um problema que se tem discutido de forma internacional há décadas, conhecido como fadiga de alarmes. Este fenômeno ocorre quando um excesso de alarmes mascara alertas clinicamente importantes, levando os médicos a ignorá-los, silenciá-los ou desativá-los. A ocorrência de muitos alarmes pode resultar em uma sobrecarga sensorial para a equipe médica. Isso pode diminuir a sensibilidade deles aos sinais de alerta e reduzir a confiança na urgência desses alarmes, o que pode levar a uma resposta inadequada. Ignorar sinais de alarme críticos pode representar um risco significativo à segurança dos pacientes, comprometendo diretamente sua integridade. (BRIDI *et al.*, 2014).

Diante desse contexto, esse trabalho se justifica pelo propósito de implementar uma central de alarme que seja economicamente viável e que possa garantir a detecção precoce de

emergências, possibilitando uma resposta ágil por parte da equipe de manutenção. Evitando interrupções nos processos de atendimento e minimizando possíveis impactos negativos na prestação dos cuidados de saúde, além de garantir que a fadiga de alarmes não cause nenhum dano ao paciente.

## 1.2 Objetivos

### 1.2.1 *Objetivo geral*

Projetar e implementar uma central de alarme hospitalar para monitorar os gases medicinais, que seja capaz de trazer maior segurança para o ambiente hospitalar.

### 1.2.2 *Objetivos específicos*

- Projetar uma Central de Alarme com dispositivo ESP32 LoRa;
- Implementar o projeto da Central de Alarme;
- Validar o desempenho do sistema;
- Estimar viabilidade econômica do Projeto;
- Projetar placa de circuito impresso;
- Calibração do ADC do ESP32 LoRa.

## 1.3 Organização do trabalho

No Capítulo 2 deste trabalho é apresentada uma revisão bibliográfica sobre aspectos fundamentais relacionados ao desenvolvimento da central de alarmes hospitalares. São discutidos temas como os gases medicinais e sua importância, a fadiga de alarmes, a Internet das Coisas e suas aplicações em monitoramento, além da tecnologia de comunicação sem fio LoRa e do protocolo de comunicação I2C. O texto também discute os sensores específicos que serão utilizados neste projeto e as práticas recomendadas para a vigilância de alarmes hospitalares.

O Capítulo 3 detalha a metodologia empregada no desenvolvimento do sistema, descrevendo a integração dos sensores, a configuração dos módulos de comunicação e a implementação do Node-RED para o processamento e gestão dos dados. São apresentados os resultados obtidos a partir da validação e análise do sistema, incluindo testes de funcionalidade.

O Capítulo 4 e o Capítulo 5 encerram o trabalho apresentando uma síntese das principais conclusões obtidas ao longo da pesquisa, além de propor sugestões para melhorias e desenvolvimentos futuros na central de alarmes hospitalares. Essa abordagem não apenas consolida os resultados alcançados, mas também abre caminho para avanços que podem aprimorar a eficiência, a confiabilidade e a segurança do sistema, contribuindo para a evolução contínua de soluções voltadas ao ambiente hospitalar.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Gases medicinais

Para que as instalações hospitalares funcionem de forma adequada, é preciso que se tenha à disposição insumos básicos, que visam garantir a vida dos pacientes. Neste contexto, podemos citar os gases medicinais.

Os gases medicinais devem ser disponibilizados em todos os setores onde são necessários. Normalmente, esses insumos são distribuídos por meio de extensas redes de tubulação. Equipamentos específicos são empregados para assegurar que esses insumos fluam e mantenham pressões dentro de uma faixa de segurança exigida. Além disso, é fundamental preservar as propriedades químicas dos gases de modo a evitar danos, tanto aos equipamentos eletromédicos quanto ao paciente (LOPEZ; ABREU, 2013).

Agência Nacional de Vigilância Sanitária - ANVISA definiu gases medicinais como:

São medicamentos na forma de gás, gás liquefeito ou líquido criogênico isolados ou associados entre si e administrados em humanos para fins de diagnóstico médico, tratamento ou prevenção de doenças e para restauração, correção ou modificação de funções fisiológicas. Os gases medicinais são utilizados em hospitais, clínicas de saúde ou outros locais de interesse à saúde, bem como em tratamentos domiciliares de pacientes. São exemplos de gases medicinais: oxigênio medicinal; ar sintético medicinal; óxido nitroso medicinal e dióxido de carbono medicinal ((ANVISA), 2020, p.1).

Figura 1 – Tubulação de gases medicinais



Fonte: (Gás, 2023).

## 2.2 Fadiga de alarmes

Como definido na introdução do Capítulo 1, fadiga de alarmes ocorre quando um grande número de alarmes é disparado simultaneamente, fazendo com que alguns alarmes relevantes sejam silenciados, desativados ou ignorados pelos profissionais de saúde. A situação se agrava quando outros alarmes clinicamente relevantes são ignorados. A ausência de resposta aos sinais sonoros adequados não apenas compromete a segurança, mas também pode resultar na deterioração clínica de pacientes gravemente enfermos (OLIVEIRA *et al.*, 2018).

O Professor Roberto Carlos Lyra da Silva, do Departamento de Enfermagem Fundamental da Universidade Federal do Estado do Rio de Janeiro (UNIRIO), ressalta a importância de ampliar o conhecimento e a pesquisa sobre a fadiga de alarmes no Brasil. Como profissional de saúde e pesquisador, ele enfatiza a necessidade de intensificar esforços para entender melhor este fenômeno. Ele considera imperativo o desenvolvimento e a implementação de procedimentos eficazes que possam atenuar os problemas associados à fadiga de alarmes (SILVA, 2014).

## 2.3 Internet das coisas

A Internet das Coisas (IoT) é o resultado da conexão de dispositivos físicos à Internet. Essa conexão possibilita a coleta e transmissão de dados entre eles, pois os mesmos estão equipados com sensores, software e outras tecnologias. Permite que esses dispositivos possam comunicar-se e interagir entre si e com outros sistemas externos.

IoT tem como objetivo automatizar processos, aumentar a eficiência, produzir novos serviços e melhorar a qualidade de vida das pessoas. Por exemplo, os dispositivos compatíveis com IoT podem monitorar a saúde em tempo real, otimizar o uso de energia em casas e negócios. Portanto, podemos definir IoT, como um conjunto de protocolos e tecnologias, o qual permite que objetos sejam conectados a uma rede de comunicação e controlados por meio desta rede (MAGRANI, 2022).

A IoT possibilita o acesso a dados em tempo real provenientes de diversos dispositivos conectados, abrangendo desde a infraestrutura predial até os equipamentos de apoio e sistemas de manutenção. Esse monitoramento contínuo e instantâneo não só aumenta a eficiência operacional, mas também permite a identificação precoce de possíveis falhas, promovendo um ambiente hospitalar mais seguro, confiável e preparado para responder a eventuais problemas de forma ágil e eficaz. (AMARAL, 2025).

## 2.4 Introdução à tecnologia LoRa

O termo LoRa é uma abreviação de Long Range (longo alcance) e refere-se a uma tecnologia de comunicação sem fio projetada para transmitir dados a longas distâncias com baixo consumo de energia. Essa tecnologia de comunicação sem fio facilita a transmissão de dados a longas distâncias, possibilitando sua operação em frequências abaixo de 1GHz. Isso lhe confere a capacidade de superar obstáculos e realizar transmissões eficazmente, mesmo em áreas urbanas (PROTOTYPE, 2016).

Desenvolvida pela Semtech, LoRa utiliza uma técnica de modulação chamada *CSS (Chirp Spread Spectrum)*, na qual os sinais são codificados usando variações lineares de frequência ao longo do tempo, conhecidas como *chirps*. Esses *chirps* podem ser comparados a "apitos" que mudam de tom de maneira previsível, permitindo que o receptor entenda a mensagem, mesmo que o sinal seja fraco ou esteja sujeito a interferências (AUGUSTIN *et al.*, 2016).

Uma característica importante do LoRa é sua capacidade de corrigir automaticamente desvios de frequência entre transmissor e receptor, tornando-o imune ao efeito *Doppler*, que ocorre quando há movimento entre os dispositivos e pode alterar a frequência do sinal. É como se, mesmo ouvindo um apito distante enquanto anda de bicicleta, o receptor conseguisse ajustar e entender perfeitamente a mensagem. Essa robustez permite o uso de transmissores com componentes de menor precisão, reduzindo custos (AUGUSTIN *et al.*, 2016).

Além disso, os receptores LoRa possuem alta sensibilidade, captando sinais com potência de até  $-130\text{dBm}$ . Esse valor, medido em *decibéis-milivatts* (dBm), representa a intensidade mínima do sinal que o receptor consegue detectar. Para fins de comparação:

- a) Um sinal de  $-130\text{dBm}$  é extremamente fraco, como tentar ouvir um sussurro vindo de quilômetros de distância;
- b) Já um sinal de  $-30\text{dBm}$  é consideravelmente mais forte, semelhante a um som normal que você ouve a poucos metros.

Essa diferença de potência tem implicações importantes. Sinais mais fracos, como  $-130\text{dBm}$ , são ótimos para comunicação de longo alcance, mas possuem limitações em termos de taxa de dados, já que a qualidade do sinal diminui à medida que a potência enfraquece. Por outro lado, sinais mais fortes, como  $-30\text{dBm}$ , permitem maior estabilidade e taxas de transmissão mais altas (AUGUSTIN *et al.*, 2016).

A tecnologia LoRa opera principalmente em cenários onde a prioridade é alcançar grandes distâncias com baixo consumo de energia, mesmo que isso implique taxas de dados mais modestas. Ela utiliza bandas ISM (*Industrial, Scientific, and Medical*), como 433 MHz, 868 MHz ou 915 MHz, dependendo da região, suportando taxas de transmissão de até 50 kbps e envio de mensagens de 2 a 255 bytes (AUGUSTIN *et al.*, 2016).

Em resumo, LoRa é uma tecnologia eficiente e econômica, ideal para aplicações de IoT, onde é necessário transmitir pequenas quantidades de dados em longas distâncias com baixo consumo de energia. Sua simplicidade e adaptabilidade a tornam uma solução robusta para diversos cenários, incluindo monitoramento remoto (AUGUSTIN *et al.*, 2016).

A Resolução nº 680, de 27 de junho de 2017, aprovada pela ANATEL (Agência Nacional de Telecomunicações) define os requisitos técnicos e procedimentos para a certificação de equipamentos de radiocomunicação de radiação restrita. Ela estabelece os padrões e critérios que os dispositivos devem cumprir para garantir a conformidade com as normas regulatórias, visando à operação segura e eficiente desses equipamentos. As principais faixas de frequência que se destacam para a utilização dessa tecnologia no Brasil são 902-907,5 MHz e 915-928 MHz (ANATEL, 2017).

Tabela 1 – Faixas de frequências permitidas para LoRa no Brasil

Frequência Inicial	Frequência Final	Unidade
156,9	242,95	MHz
243	322	MHz
335,4	399,9	MHz
410	608	MHz
614	940	MHz
944	960	MHz
1.710	1.785	MHz
1.805	1.880	MHz
1.885	1.900	MHz
1.910	1.980	MHz
2.110	2.170	MHz
2.300	2.483,50	MHz

Fonte: (ANATEL, 2017).

Para configurar adequadamente o dispositivo, tanto para transmissão quanto para recepção, é necessário entender e ajustar alguns parâmetros fundamentais. Esses parâmetros são necessários para garantir eficiência, alcance e confiabilidade na comunicação.

### 2.4.1 Configuração dos parâmetros LoRa

A comunicação LoRa baseia-se em parâmetros que determinam a qualidade do sinal, a distância alcançada e o consumo de energia. Abaixo, destacamos os principais:

- a) Frequência de operação determina o canal pelo qual os dispositivos se comunicam. A seleção da frequência deve estar de acordo com as normas locais para garantir conformidade legal e evitar interferências com outros sistemas. Por exemplo, as bandas de 868 MHz são destinadas à Europa, enquanto a faixa de 915 MHz é utilizada na América do Norte e no Brasil (CORPORATION, 2024);

- b) Spreading factor (SF) determina o tempo necessário para transmitir um símbolo e é um parâmetro essencial para ajustar o desempenho de sistemas de comunicação. Valores comuns variam de SF7 a SF12, com impactos significativos na operação: um SF baixo, como SF7, resulta em transmissões mais rápidas, menor alcance e menor consumo de energia, enquanto um SF alto, como SF12, proporciona maior alcance, porém com transmissões mais lentas e maior consumo energético. A configuração adequada do SF permite equilibrar alcance e eficiência energética, adaptando o sistema às necessidades específicas do ambiente e do cenário de uso (CORPORATION, 2024);
- c) Bandwidth (largura de banda) pode ser comparada a uma estrada para o tráfego de dados. Estradas mais largas (largura de banda maior, como 500 kHz) permitem maior velocidade de transmissão, mas reduzem a sensibilidade do receptor, limitando o alcance. Já estradas mais estreitas (largura de banda menor, como 125 kHz) suportam menos tráfego, mas aumentam a sensibilidade e o alcance. Por isso, é importante equilibrar a velocidade desejada com a distância necessária, adaptando-se ao cenário de uso (CORPORATION, 2024);
- d) Power output (potência de transmissão) indica a força do sinal enviado, medida em dBm, que é uma unidade que expressa a potência em decibéis em relação a 1 mW (miliwatt). Valores comuns variam de 2 dBm a 20 dBm. Potências maiores aumentam o alcance do sinal, permitindo cobrir distâncias maiores, mas também elevam o consumo de energia. Isso é especialmente relevante em dispositivos alimentados por bateria. Além disso, é importante ajustar a potência para cumprir as regulamentações locais e evitar interferências. A escolha da potência deve equilibrar alcance, consumo energético e conformidade com as normas (CORPORATION, 2024);
- e) Coding rate (CR) define a proporção entre os bits de dados úteis e os bits de correção de erro em uma transmissão. Por exemplo, um CR de  $4/5$  significa que, para cada 5 bits transmitidos, 4 são dados úteis e 1 é usado para correção de erros. Valores típicos de CR variam entre  $4/5$  e  $4/8$ . Um CR mais alto (como  $4/8$ ) aumenta a resistência a erros, tornando a comunicação mais confiável em ambientes com interferência ou em longas distâncias. No entanto, isso reduz a taxa de dados, já que mais bits são dedicados à correção de erros. A escolha do CR deve equilibrar a necessidade de confiabilidade com a velocidade de transmissão, dependendo do cenário de uso (CORPORATION, 2024).

A configuração correta dos parâmetros do ESP32 LoRa é essencial para garantir uma comunicação eficiente, equilibrando alcance, consumo de energia e taxa de dados. A escolha dos valores ideais depende do cenário de aplicação, como distância entre os dispositivos, ambiente (urbano ou rural) e limitações energéticas. Uma configuração bem ajustada maximiza o desempenho e a confiabilidade do sistema (CORPORATION, 2024).

## 2.5 Protocolo de comunicação I2C

O protocolo de comunicação I2C (*Inter-Integrated Circuit*) é amplamente utilizado na eletrônica para garantir a interconexão eficiente entre dispositivos. Neste projeto, o protocolo I2C será aplicado na leitura de dados em um display conectado ao ESP32 LoRa.

Segundo (INSTRUMENTS, 2013), o protocolo I2C foi desenvolvido em 1982 pela Philips Semiconductor, atualmente NXP Semiconductor. Inicialmente, foi concebido como um protocolo de comunicação de baixa velocidade para conectar microcontroladores, processadores e outros dispositivos controladores a periféricos, como conversores de dados. Desde 2006, sua implementação não exige mais licenciamento, o que levou à sua ampla adoção por diversas empresas de semicondutores, incluindo a Texas Instruments (TI), que desenvolveu dispositivos compatíveis com esse padrão.

O I2C se destaca por sua simplicidade e economia de recursos, uma vez que requer apenas duas linhas para comunicação, a linha de dados serial e a linha de clock serial. Isso permite a conexão de múltiplos dispositivos em um único barramento usando apenas essas duas linhas. Além disso, o dispositivo controlador pode se comunicar com qualquer dispositivo alvo, por meio de um endereço I2C exclusivo transmitido pela linha de dados serial. Essas características tornam o protocolo I2C uma escolha popular para implementação por fabricantes de dispositivos (INSTRUMENTS, 2013).

O protocolo I2C é um sistema multimestre que utiliza duas linhas de sinal: SDA (dados seriais) e SCL (relógio serial). Diferente de outros protocolos, não exige seleção de escravo nem lógica de arbitragem dedicada, permitindo que múltiplos mestres e escravos compartilhem o mesmo barramento para comunicação. Cada dispositivo no barramento é identificado por um endereço de 7 bits, e os dados são organizados em bytes de 8 bits, acompanhados por bits de controle que regulam a inicialização, encerramento, direcionamento e reconhecimento da comunicação. As velocidades de transmissão padrão incluem 100 kbps, 400 kbps e 3,4 Mbps, com variantes de 10 kbps e 1 Mbps. Fisicamente, o barramento I2C consiste em duas linhas ativas, SDA e SCL, além de uma conexão de aterramento. Ambas as linhas são bidirecionais, permitindo a comunicação eficiente entre dispositivos. Conforme a especificação, o dispositivo que inicia a transferência de dados é denominado Mestre, enquanto os demais atuam como Escravos durante essa operação (LEENS, 2009).

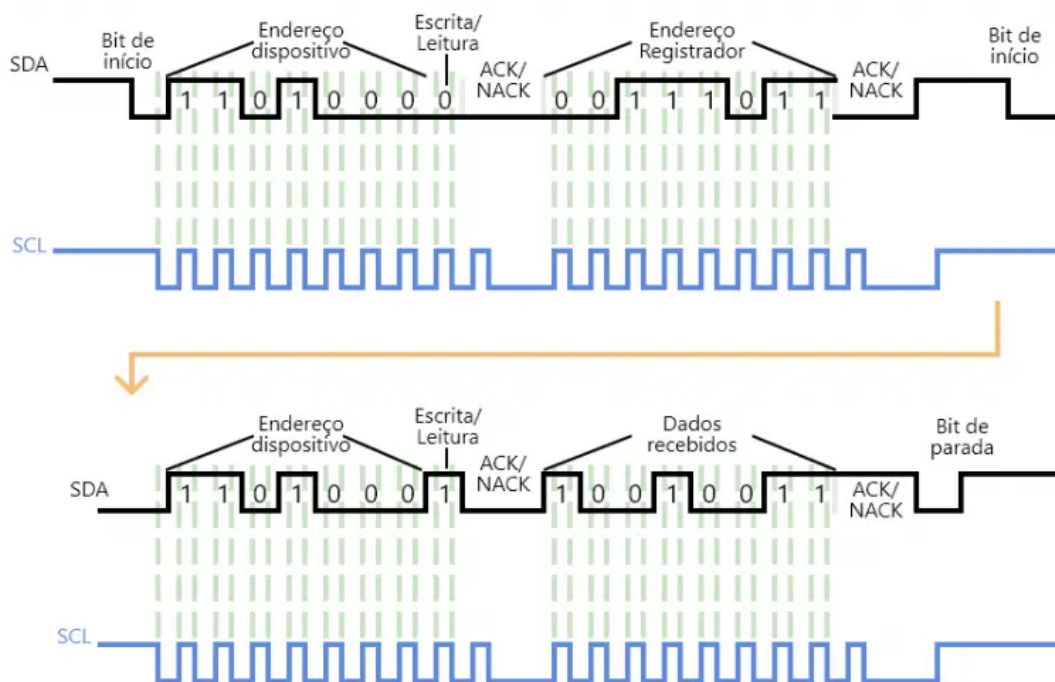
Segundo (LEENS, 2009), o funcionamento do barramento I2C começa com o envio de uma condição de *START* pelo mestre, que atua como um alerta para todos os dispositivos conectados. Nesse momento, todos os dispositivos ficam atentos, aguardando os próximos dados. O mestre então envia o endereço do dispositivo alvo, acompanhado de um bit que indica se a operação será de leitura ou escrita. Após receber o endereço, cada dispositivo conectado compara o com o seu próprio. Caso o endereço não corresponda, o dispositivo entra em estado de espera até que a condição de *STOP* seja emitida pelo mestre, liberando o barramento. No entanto, se

o endereço corresponder, o dispositivo envia um sinal de *ACKNOWLEDGE* como resposta, confirmando o reconhecimento. A partir desse ponto, o mestre pode iniciar a transmissão ou recepção de dados.

Quando a transmissão é concluída, o mestre emite a condição de *STOP*, sinalizando que o barramento está livre para novas transmissões. No caso de o mestre desejar receber dados de um escravo, o procedimento inicial é semelhante, mas o bit R/W é configurado para leitura. Após reconhecer o endereço, o escravo começa a enviar os dados solicitados, transmitindo-os *byte a byte*. Após cada *byte* recebido, o mestre deve confirmar o recebimento dos dados. Os dispositivos escravos que não forem endereçados permanecem em estado de espera até que uma condição de *STOP* seja emitida, momento em que retomam a escuta do barramento para futuras transmissões (LEENS, 2009).

A figura abaixo mostra o exemplo de uma leitura de um dispositivo I2C. O endereço do escravo é 0x68 (1101000), o registrador é 0x3B (00111011) e os dados enviados pelo escravo são: 10010011 (PROJETADO, 2024).

Figura 2 – Comunicação I2C entre mestre e escravo



Fonte: (PROJETADO, 2024).

## 2.6 Protocolo de rede

A implementação de uma central de alarmes eficiente em ambientes críticos, como hospitais, depende diretamente da escolha de protocolos de comunicação confiáveis e robustos. Nesse cenário, o protocolo TCP/IP emerge como uma solução fundamental, garantindo a transmissão

segura e em tempo real de dados essenciais para o monitoramento contínuo. Sua capacidade de integrar diversos dispositivos e sistemas em uma rede unificada permite uma comunicação ágil e estável, essencial para a detecção precoce de falhas e a resposta imediata a emergências. Além disso, o TCP/IP oferece escalabilidade e flexibilidade, adaptando-se às necessidades específicas de cada ambiente. Assim, sua adoção não apenas otimiza a eficiência operacional, mas também reforça a segurança e a confiabilidade da central de alarmes, contribuindo para a proteção de vidas e infraestruturas.

### 2.6.1 TCP-IP

Protocolo TCP/IP, é composto pelos protocolos *Transmission Control Protocol* (TCP) e *Internet Protocol* (IP), é amplamente usado como meio de comunicação de rede global. Ele fornece uma estrutura robusta e confiável para o envio e recebimento de dados em redes, garantindo que as informações cheguem ao destino corretamente e em tempo hábil.

O protocolo TCP/IP é organizado em uma estrutura de camadas, cada uma delas desempenhando funções específicas na transmissão e recebimento de dados:

- a) camada de aplicação os usuários acionam programas de aplicativos para acessar serviços disponíveis em uma internet TCP/IP. Uma aplicação se comunica com uma das camadas de protocolos de transporte para transmitir ou receber informações. Cada aplicativo seleciona o modelo de transporte adequado, que pode ser uma série de mensagens individuais ou um fluxo ininterrupto de bytes. O aplicativo fornece os dados no formato apropriado para a camada de transporte para garantir a entrega correta (COMER, 2016);
- b) camada de transporte tem como principal função proporcionar a comunicação entre um programa de aplicação e outro. Esta comunicação é conhecida como fim a fim, pois envolve aplicações em dois terminais, ao invés de aplicações entre roteadores intermediários. Uma camada de transporte tem a capacidade de não apenas regular o fluxo de informações, mas também de assegurar um transporte confiável, garantindo que os dados sejam entregues sem erros e em ordem. Para isso, o software do protocolo de transporte se estrutura para obter uma confirmação de recebimento do lado receptor e a retransmissão de pacotes perdidos do lado emissor. O software de transporte segmenta o fluxo de dados a serem transmitidos em pequenas partes (frequentemente chamadas de pacotes) e encaminha cada pacote com um endereço de destino para a próxima camada de transmissão (COMER, 2016);
- c) camada de internet é responsável pela comunicação entre computadores. Ela recebe uma solicitação para enviar um pacote da camada de transporte, juntamente com a identificação do computador destinatário. O software de Internet encapsula o pacote de transporte em um pacote IP, preenche o cabeçalho e envia o pacote IP diretamente para o destino (se o

destino está na mesma rede) ou para um roteador para ser encaminhado pela internet (se o destino é em outra rede). O software da camada de Internet também processa pacotes IP recebidos, verificando a sua validade e usando o algoritmo de encaminhamento para decidir se o pacote deve ser processado localmente ou encaminhado. Para os pacotes destinados ao próprio computador, o software na camada de internet seleciona o protocolo de transporte que irá processar o pacote (COMER, 2016);

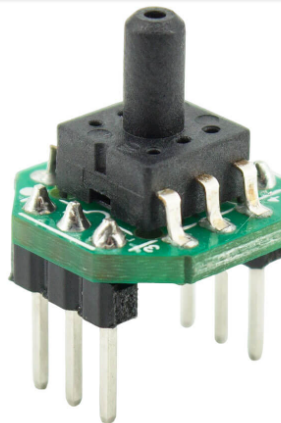
- d) camada de interface de rede e a menor camada do protocolo TCP/IP é composta pela interface de rede, encarregada de receber pacotes IP e enviá-los através de uma rede específica (COMER, 2016).

O protocolo TCP/IP serve como a base essencial para a comunicação e transferência de dados entre o Raspberry Pi e diversos serviços fundamentais para a integração do trabalho proposto, como o banco de dados, o painel de controle (*dashboard*) e o Telegram. Esse protocolo garante uma interação confiável e eficiente entre todos os componentes do sistema, desempenhando um papel vital no funcionamento e eficácia da central de alarme hospitalar.

## 2.7 Sensor de pressão XGZP6847A

O sensor de pressão XGZP6847A é essencial para detectar qualquer anomalia ou variação inesperada na pressão dos gases medicinais. Através dele podemos indicar vazamentos no sistema, obstruções nas linhas de fornecimento ou falhas em equipamentos, o que poderia representar um risco para a segurança dos pacientes. A detecção precoce de tais problemas permite uma intervenção rápida para evitar situações perigosas.

Figura 3 – Sensor de pressão XGZP6847A



Fonte: (CFsensor, 2024).

Neste projeto, o sensor XGZP6847A desempenha um papel fundamental no monitoramento dos gases medicinais. Através da sua utilização, podemos garantir que os níveis de gases estejam dentro dos limites seguros e eficazes para uso médico. Além disso, o sensor XGZP6847A é compacto e fácil de integrar, tornando-o ideal para este projeto. Para mais detalhes técnicos, consulte o datasheet do fabricante (CFsensor, 2024).

O sensor de pressão de XGZP6847A é um dispositivo de silício, oferecendo uma interface analógica ratiométrica para medição de pressão em um intervalo especificado. Ele combina um sensor piezoresistivo de silício com um circuito integrado dedicado em um pacote DIP6. Totalmente calibrado e compensado para temperatura, o XGZP6847A oferece precisão e é adequado para uma ampla gama de aplicações, como cuidados médicos, eletrodomésticos, eletrônicos de consumo, indústria automotiva, IoT e dispositivos pneumáticos. Sua integração é simples, podendo ser conectado a microcontroladores ou microprocessadores com entradas A/D (CFsensor, 2024).

Para este projeto, a escolha recaiu sobre o sensor XGZP6847A, caracterizado por uma saída analógica linear operando 0,5V a 4,5V. Embora o ESP32 LoRa tenha uma tolerância nominal para sinais analógicos de 3,3V, a decisão foi tomada a favor de utilizar um sensor com saída de 5V devido à limitação na disponibilidade de sensores compatíveis com a faixa de 3,3V.

Devido à diferença de tensão entre o sensor e o microcontrolador, é fundamental realizar o condicionamento do sinal. Esse processo consiste em adaptar o sinal analógico de saída do sensor, que opera em 5V, para a faixa de 3,3V compatível com o ESP32 LoRa. A adaptação deve ser feita de modo a preservar a linearidade do sinal, assegurando que os dados coletados pelo sensor sejam transmitidos ao microcontrolador.

## **2.8 Monitoramento de alarmes hospitalares**

Em um ambiente hospitalar, é imprescindível manter um monitoramento e avaliação constantes para identificar oportunidades de aprimoramento, assegurando a qualidade e a eficácia dos serviços prestados. Dentro da estrutura hospitalar, diferentes setores e equipamentos possuem níveis variados de criticidade, cada um exigindo monitoramento específico de parâmetros. Os profissionais encarregados pela manutenção da infraestrutura e da Engenharia Clínica desempenham um papel crucial na gestão da manutenção hospitalar, visando minimizar riscos de danos, com o objetivo de garantir a segurança tanto dos pacientes quanto de todos os colaboradores do hospital. Nesse contexto, a automação surge como uma ferramenta fundamental para aumentar a eficiência e a produtividade do estabelecimento, ao mesmo tempo em que reduz custos e minimiza possíveis falhas humanas (AMARAL, 2025).

### 3 DESENVOLVIMENTO

Este capítulo oferece uma análise detalhada dos materiais, métodos e testes utilizados no desenvolvimento do projeto de uma Central de Alarmes com ESP32 LoRa. Para atingir esse objetivo, é essencial compreender profundamente os instrumentos empregados, os procedimentos seguidos e os testes realizados ao longo do projeto. Neste capítulo, também são apresentados os resultados alcançados durante todo o trabalho.

Primeiramente, examinaremos os dispositivos utilizados para a coleta de dados e as metodologias adotadas. Cada componente do processo será abordado minuciosamente, desde a seleção dos materiais até a execução dos testes. Além disso, cada etapa será analisada detalhadamente para garantir uma compreensão abrangente das decisões tomadas e das estratégias implementadas.

O objetivo é proporcionar uma visão completa e coesa do desenvolvimento do projeto, destacando a lógica por trás de cada escolha e a eficácia dos métodos empregados para alcançar os objetivos propostos.

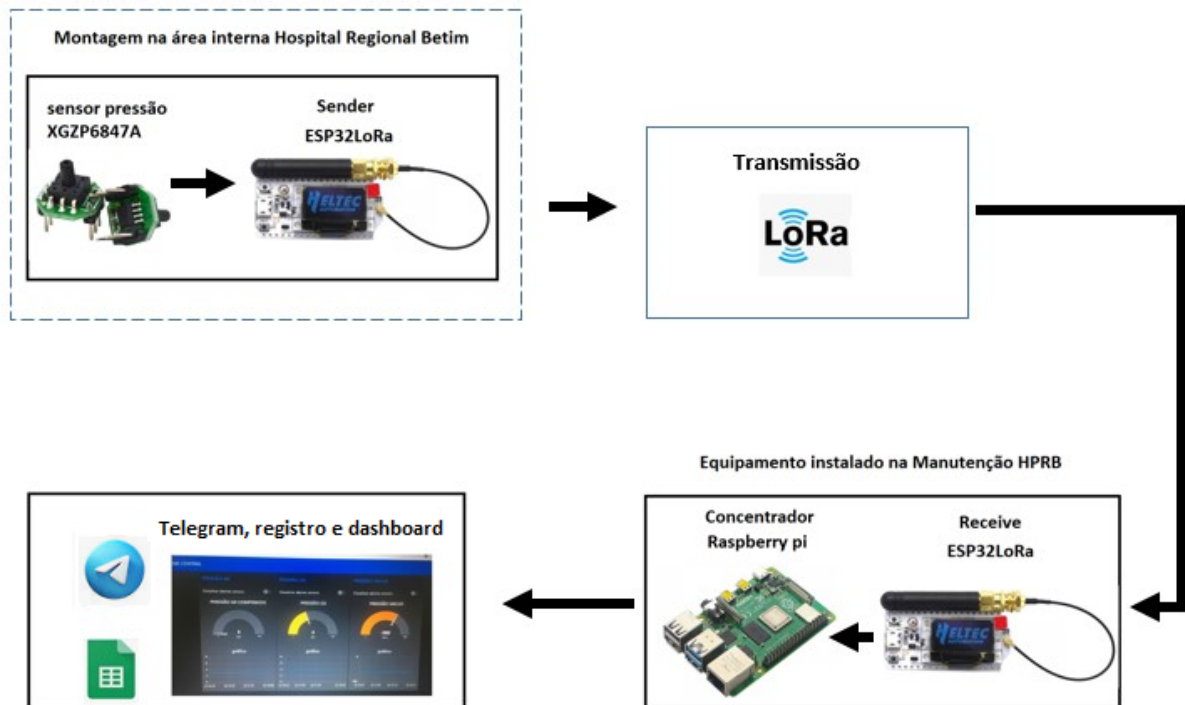
#### 3.1 Arquitetura geral do sistema proposto

O projeto consiste na implementação de *hardware* e *software* para uma central de alarmes hospitalar, fazendo a integração do ESP32 LoRa e o Raspberry Pi.

A partir da instalação de um ESP32 LoRa na área interna do Hospital Público de Betim, será possível monitorar as pressões dos gases medicinais por meio do sensor analógico XGZP6847A. Esse sensor oferece leituras contínuas e em tempo real, que, com o suporte do ESP32 LoRa, podem ser transmitidas de forma eficiente para outro dispositivo ESP32 LoRa localizado no setor de manutenção. A comunicação de longo alcance e o baixo consumo de energia do ESP32 LoRa garantem que as informações sejam enviadas de maneira confiável, permitindo um monitoramento remoto e em tempo real das condições dos gases medicinais.

Os dados coletados são transmitidos ao Raspberry Pi, que processa as informações e atualiza a interface do usuário, permitindo a visualização de um *dashboard* com as pressões dos gases medicinais em um monitor. Caso seja detectada alguma anomalia, um alarme será acionado, ativando um *buzzer* conforme necessário. Além disso, o sistema é configurado para gerar automaticamente um histórico de registros das pressões dos gases medicinais e enviar notificações instantâneas via Telegram aos responsáveis em caso de falhas no sistema, assegurando uma resposta rápida a situações críticas. A Figura 4 ilustra a arquitetura da central de alarmes, que integra o ESP32 LoRa e o Raspberry Pi no projeto proposto.

Figura 4 – Arquitetura central alarmes com Esp32 LoRa e Raspberry Pi



Fonte: Elaborada pelo autor, 2024.

Inicialmente, serão conduzidos testes de comunicação do módulo ESP32 LoRa. Esta fase envolve a implementação do *software* Meshstastic, uma ferramenta que facilita a análise do alcance do sinal no envio de pacotes (mensagens).

Em seguida, avançaremos para a linearização da entrada analógica do módulo ESP32 LoRa. Este procedimento é essencial para garantir a precisão dos dados recebidos. A linearização é um processo que converte uma curva não-linear em uma linha reta, removendo assim as distorções não lineares.

Posteriormente, será realizada uma avaliação do desempenho dos sensores de pressão XGZP6847A. Esta etapa visa analisar a consistência das leituras de pressão obtidas. Serão conduzidos testes operacionais para verificar a resposta do sensor a diferentes níveis de pressão.

Essas etapas são fundamentais para garantir o desempenho do sistema e para assegurar que os dados coletados sejam confiáveis.

### 3.2 ESP32 LoRa

Neste projeto, utilizamos a placa ESP32 LoRa V3 da Heltec, uma placa de desenvolvimento compacta que combina o microcontrolador ESP32 com um módulo LoRa. Esta placa é equipada com o ESP32S3, um microcontrolador de 32 bits com dois núcleos que opera a uma frequência de até 240 MHz.

A placa ESP32 LoRa V3 se destaca por sua conectividade, integrando Wi-Fi, LoRa e *Bluetooth*. O Wi-Fi suporta o padrão 802.11 b/g/n, permitindo velocidades de até 150 Mbps. O módulo LoRa utiliza o *chipset* SX1262, operando em frequências de 470~510 MHz, 863~928 MHz. A potência máxima de transmissão do LoRa é de  $21 \pm 1$  dBm, enquanto a sensibilidade máxima de recepção é de -134 dBm, garantindo uma comunicação em longas distâncias.

Além disso, a placa possui um *display* OLED de matriz de pontos 128x64 integrado, que pode ser usado para exibir informações de depuração. Esse *display* facilita a interface do usuário e o desenvolvimento do projeto, permitindo um monitoramento visual direto.

Em termos de interface, a placa possui 36 portas GPIO que suportam ADC, DAC, UART, SPI e I2C, oferecendo grande flexibilidade para conectar sensores e outros dispositivos. A alimentação e programação da placa são feitas através de uma porta Usb Tipo C.

A placa ESP32 LoRa V3 opera com uma tensão de 3,3V e consome menos de 10  $\mu$ A em modo de espera, tornando-a eficiente em termos de consumo energético. As dimensões compactas da placa são de  $50,2 \times 25,5 \times 10,2$  mm. Outro aspecto importante, a placa é capaz de operar em uma faixa de temperatura de  $-20 \sim 70$  °C, o Quadro 1 detalha as especificações técnicas do ESP32 LoRa V3.

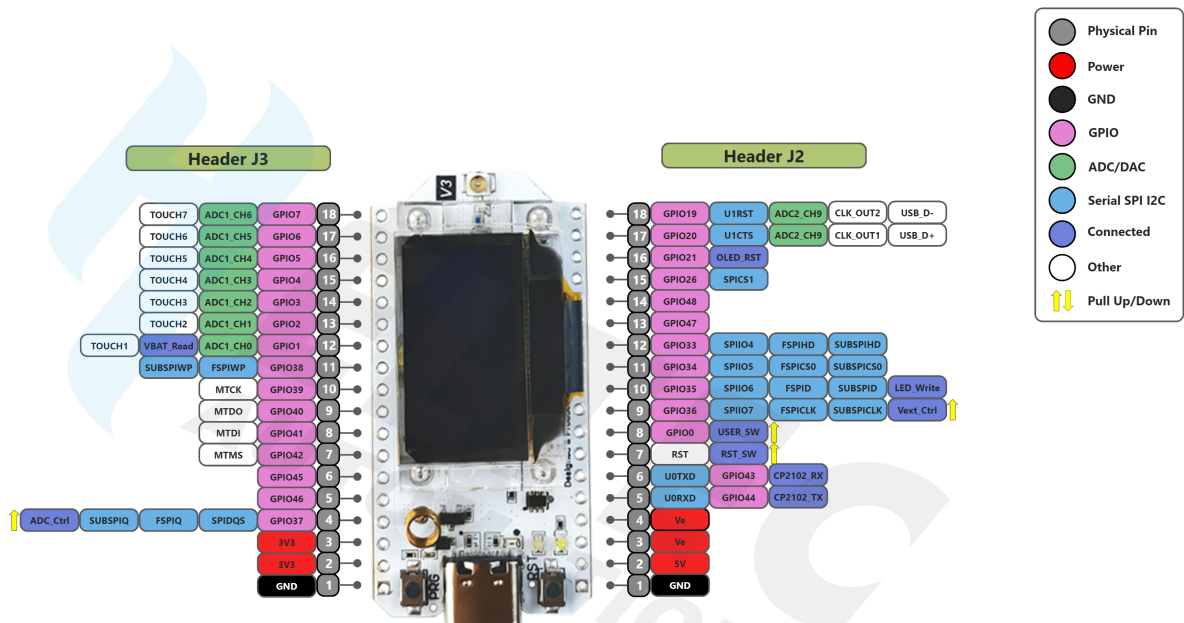
Quadro 1 – Especificações gerais do ESP32 LoRa V3

Parâmetros	Descrição
Chip Principal	ESP32-S3FN8 (Processador Xtensa® 32-bit lx7 dual core)
Chipset LoRa	SX1262
Chip USB para Serial	CP2102
Frequência	470~510 MHz, 863~928 MHz
Potência Máxima de Transmissão	$21 \pm 1$ dBm
Sensibilidade Máxima de Recepção	-134 dBm
Wi-Fi	802.11 b/g/n, até 150Mbps
Bluetooth	Bluetooth LE: Bluetooth 5, Bluetooth mesh
Recursos de Hardware	$7 \times ADC1 + 2 \times ADC2; 7 \times Touch; 3 \times UART; 2 \times I2C; 2 \times SPI;$ etc.
Memória	384KB ROM; 512KB SRAM; 16KB RTC SRAM; 8MB SiP Flash
Interface	USB Type-C; $2 \times 1.25$ interface para bateria de lítio; LoRa ANT(IPEX1.0); $2 \times 18 \times 2.54$ Header Pin
Bateria	Alimentação e carregamento com bateria de lítio de 3.7V
Temperatura de Operação	$-20 \sim 70$ °C
Dimensões	$50,2 \times 25,5 \times 10,2$ mm

Fonte: (AUTOMATION, 2025).

A Figura 5 apresenta o mapa de pinos da placa ESP32 LoRa V3, detalhando as funções e a disposição dos pinos. Este diagrama é essencial para a correta conexão de sensores, atuadores e outros periféricos, permitindo que os desenvolvedores aproveitem ao máximo a capacidade da placa.

Figura 5 – Mapa pinos do Esp32 LoRa V3



HTIT-WB32LA(F)\_V3 Pin map



Fonte: (AUTOMATION, 2025).

Para garantir o desempenho ideal do projeto, é essencial avaliar a capacidade de alcance de transmissão de dados da placa ESP32 LoRa V3. Com esse objetivo, foi adquirida a antena TX915-JKS-IPX20 (para mais detalhes técnicos, consulte o *datasheet* (EBYTE, 2023)), a fim de realizar uma comparação técnica de sua performance em relação à antena integrada ao módulo ESP32 LoRa V3. Essa análise permitirá identificar qual configuração oferece maior eficiência e confiabilidade, aspectos fundamentais para assegurar uma comunicação robusta em longas distâncias, alinhada às demandas do projeto.

Essa análise será conduzida em duas etapas distintas: a primeira utilizando a antena integrada ao módulo ESP32 e a segunda com a antena TX915-JKS-IPX20.

Nessa etapa, será medido apenas o alcance da transmissão. Com esses dados, será possível determinar qual antena oferece melhor desempenho em termos de eficiência e confiabilidade, aspectos essenciais para a comunicação em longas distâncias dentro do contexto do projeto.

Figura 6 – Antenas LoRa 915 MHz



Fonte: Elaborada pelo autor, 2025.

O *software* Meshtastic foi utilizado para verificar o alcance da comunicação LoRa no Hospital Público Betim. Esta avaliação nos permitirá determinar se a tecnologia é viável em um ambiente hospitalar onde há diversas barreiras físicas.

Meshtastic é um *software* que permite a criação e gerenciamento de redes *mesh*, ele faz parte de um projeto open-source para a criação de um sistema de comunicação descentralizado. Ele utiliza a modulação LoRa, permitindo a criação de uma rede de comunicação que não depende de nenhuma infraestrutura pré-existente. As redes *mesh* são caracterizadas por sua flexibilidade e robustez, pois cada nó está conectado a vários outros nós, permitindo que os dados encontrem diversos caminhos para chegar ao destino. Isso é especialmente útil em ambientes onde a conexão direta entre dois pontos pode não ser possível.

No contexto do nosso teste, o Meshtastic será usado para estabelecer uma rede *mesh* utilizando dois módulos ESP32 LoRa V3. Através desta rede, enviaremos e receberemos pacotes de dados para avaliar o alcance efetivo da placa ESP32 LoRa V3. O teste será realizado em diferentes pontos dentro e fora do Hospital Público Betim para garantir uma avaliação abrangente

do desempenho do dispositivo.

Inicialmente, foi realizada a instalação do *software* Meshtastic em duas placas ESP32 LoRa V3, seguindo o tutorial disponível no site oficial da Meshtastic. A configuração foi facilitada pelo aplicativo Meshtastic, disponível para download tanto para IOS quanto para Android.

Após a instalação do *software* Meshtastic nas placas ESP32 LoRa V3, utilizamos o aplicativo Meshtastic para configurá-las. Através do aplicativo, foi possível definir a região da placa, a frequência e a largura de banda.

Com a configuração concluída, iniciamos os testes para verificar a eficácia da comunicação. Uma placa foi posicionada de forma fixa no setor de manutenção do Hospital Público Betim e a outra em locais diferentes e mensagens foram enviadas de uma para outra, testando tanto a distância quanto a qualidade do sinal.

### 3.2.1 Resultado do teste de alcance da comunicação ESP32 LoRa

Os experimentos conduzidos para testar a comunicação entre os módulos ESP32 LoRa, com o uso de diferentes antenas, evidenciam a superioridade da Antena 2 (TX915-JKS-IPX20) em comparação com a antena integrada ao módulo ESP32 (Antena 1). Essa superioridade é destacada pelos resultados de intensidade do sinal, obtidos em diversos locais, tanto dentro como fora do hospital, conforme mostrado na Tabela 2.

Tabela 2 – Comparativo entre antena 1 (integrada ao módulo ESP32) e antena 2 (TX915-JKS-IPX20)

Localização	Distância (m)	Antena 1 (RSSI)	Antena 2 (RSSI)	Diferença de RSSI (dBm)	Ganho %
<b>Externa</b>	190	-126	-103	-23	18,25
	170	-119	-94	-25	21,01
	140	-114	-85	-29	25,44
	90	-104	-83	-21	20,19
	70	-103	-75	-28	27,18
	50	-94	-68	-26	27,66
	20	-63	-55	-8	12,70
<b>Interna</b>	Térreo (150 m)	-128	-114	-14	10,94
	Térreo (120 m)	-120	-108	-12	10,00
	Térreo (90 m)	-113	-100	-13	11,50
	5º Andar (92 m)	-126	-103	-23	18,25
	2º Andar (90 m)	-105	-78	-27	25,71
	Térreo (50 m)	-97	-86	-11	11,34
	Térreo (30 m)	-88	-74	-14	15,91

Fonte: Elaborada pelo autor, 2024.

A análise dos dados apresentados na Tabela 2 revela que a Antena 2 (TX915-JKS-IPX20) proporciona um desempenho superior em termos de RSSI (*Received Signal Strength Indicator*) quando comparada à Antena 1 (integrada ao módulo ESP32). Este comportamento é evidenciado pela comparação dos valores de RSSI registrados em várias distâncias e localizações.

Os dados experimentais revelam uma diferença significativa no desempenho das duas antenas, particularmente em relação à integridade dos pacotes recebidos em distâncias maiores. A Antena 1 (integrada ao módulo ESP32) apresentou uma perda de pacotes em distâncias mais longas, o que comprometeu a confiabilidade do recebimento de dados. Em contraste, a Antena 2 (TX915-JKS-IPX20) demonstrou uma taxa de recepção de 100% dos pacotes, mesmo nas distâncias mais extensas.

Devido à sua taxa elevada de recepção de pacotes, a Antena 2 (TX915-JKS-IPX20) é a opção mais adequada para aplicações que exigem segurança e confiabilidade na transmissão de dados.

Com base nos testes experimentais realizados, pode-se concluir que o módulo ESP32 LoRa, quando acoplado à Antena 2 (TX915-JKS-IPX20), demonstrou uma capacidade adequada para o ambiente. A combinação do módulo com a Antena 2 mostrou-se altamente confiável tanto no envio quanto no recebimento de pacotes, garantindo uma transmissão de dados consistente e sem perda. Essa performance evidencia um desempenho satisfatório e robusto para as necessidades do projeto, assegurando a integridade e a eficácia na comunicação ao longo das distâncias testadas.

### 3.3 Linearização ADC ESP32 LoRa

A linearidade em um sistema de conversão analógica-digital, como o ADC (Conversor Analógico-Digital), diz respeito à habilidade do sistema em transformar um sinal analógico em um sinal digital de maneira proporcional e exata. Ou seja, um ADC é considerado linear quando a saída digital varia proporcionalmente ao sinal analógico de entrada.

A falta de linearidade do ADC no ESP32 é uma questão amplamente discutida em diversos fóruns especializados. A própria Espressif, fabricante do ESP32, reconheceu essa limitação e implementou medidas para mitigar o problema.

Dentre as medidas adotadas pela Espressif, podemos destacar a comparação entre a tensão analógica de entrada e a tensão de referência para determinar o valor de cada bit no resultado digital de saída. A tensão de referência do ADC no ESP32 é projetada para ser de 1100 mV. No entanto, em diferentes chips, essa tensão pode variar entre 1000 mV e 1200 mV, tornando essencial a calibração por software, que estabelece uma curva para corrigir o erro do ADC (SYSTEMS, 2023).

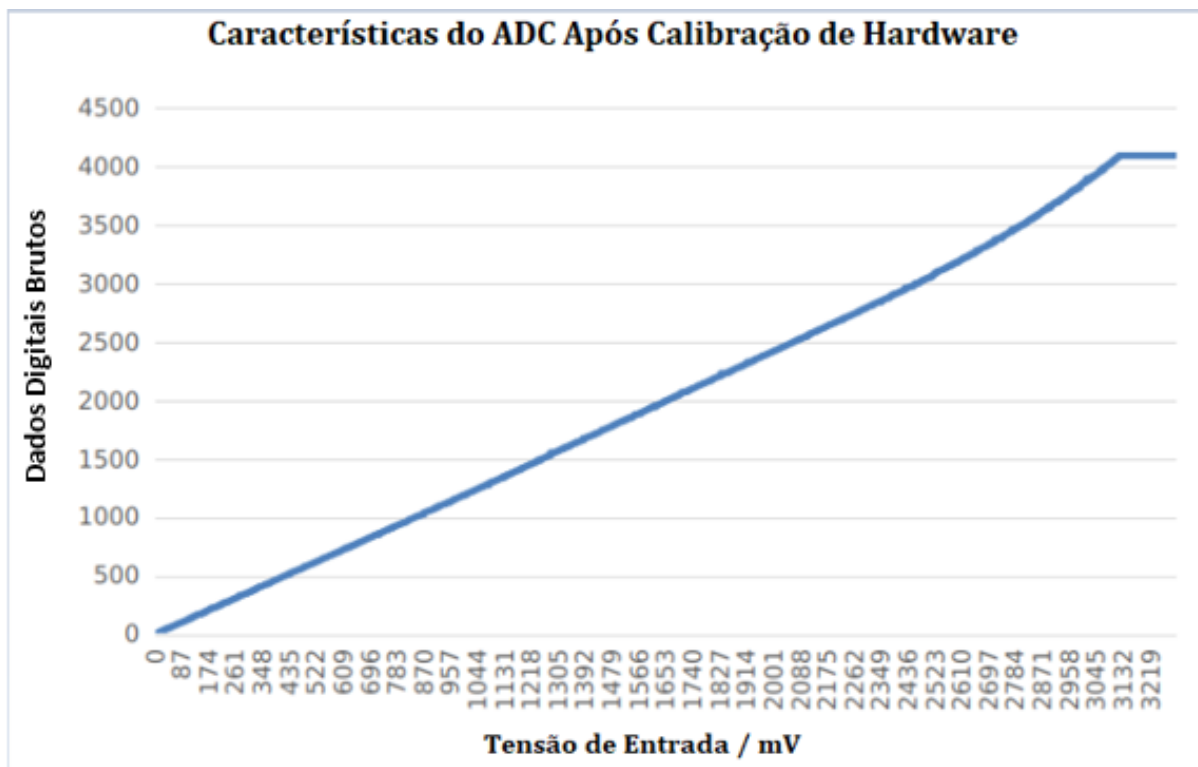
Para reduzir o ruído no ADC, pode-se adicionar um capacitor de *bypass* cerâmico de 100 nF ao pad de entrada do ADC em uso. Além disso, a multamostragem pode ser aplicada

para uma atenuação adicional do ruído. Desde 2019, a Espressif implementou a calibração de *hardware*, que agora é realizada internamente pelo driver ADC (SYSTEMS, 2023).

Apesar das melhorias adotadas, a linearidade do ADC pode ainda apresentar variações que afetam a precisão das medições, exigindo ajustes e calibrações adicionais para obter resultados mais consistentes.

Para fins de comparação, apresento a curva característica do ADC do ESP32, conforme publicada pela Espressif, juntamente com a curva obtida experimentalmente em minhas medições. Esta abordagem permite uma análise detalhada, evidenciando a não linearidade do ADC do ESP32.

Figura 7 – Curva característica ADC ESP2

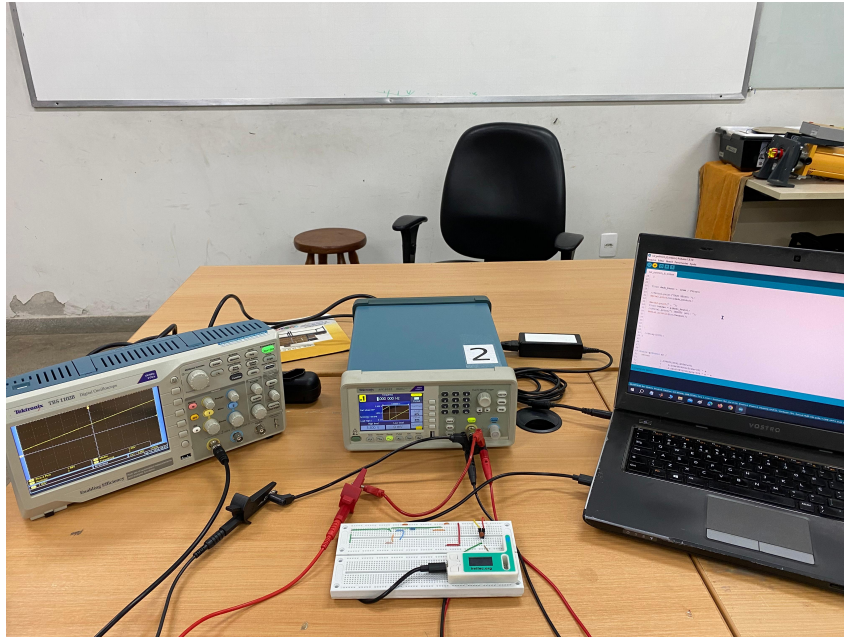


Fonte:(ESPRESSIF, 2025).

A curva apresentada na Figura 7, divulgada pela Espressif, demonstra que o comportamento do ADC no ESP32 apresenta variações que podem comprometer a precisão das medições. Essas variações exigem a aplicação de técnicas de correção para assegurar resultados confiáveis, viabilizando o uso do ADC em aplicações que demandam maior exatidão nos valores convertidos.

A Figura 9 foi obtida a partir de um gerador de função, configurado para gerar um sinal de onda dente de serra com frequência de 1 Hz, variando de 0 a 3,3 V. Esse sinal foi então injetado diretamente no ADC do ESP32 para análise de seu comportamento e posterior calibração do conversor analógico-digital.

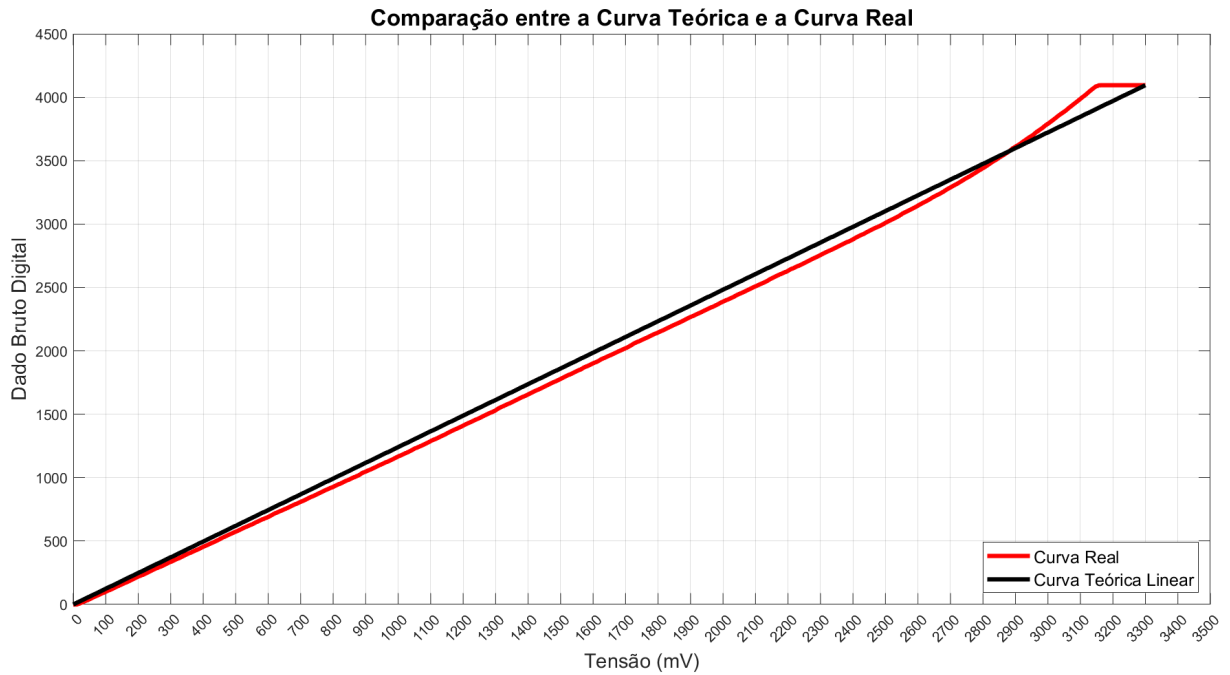
Figura 8 – Teste experimental ADC ESP32 LoRa



Fonte: Elaborada pelo autor, 2024.

Para uma análise mais detalhada, foi utilizada uma curva teórica para comparação com os resultados experimentais obtidos a partir dos dados coletados. A comparação demonstrou que o ADC do ESP32 não apresenta comportamento linear, conforme evidenciado pela curva experimental. Devido à não linearidade, é necessário implementar correções adequadas para assegurar a precisão das medições.

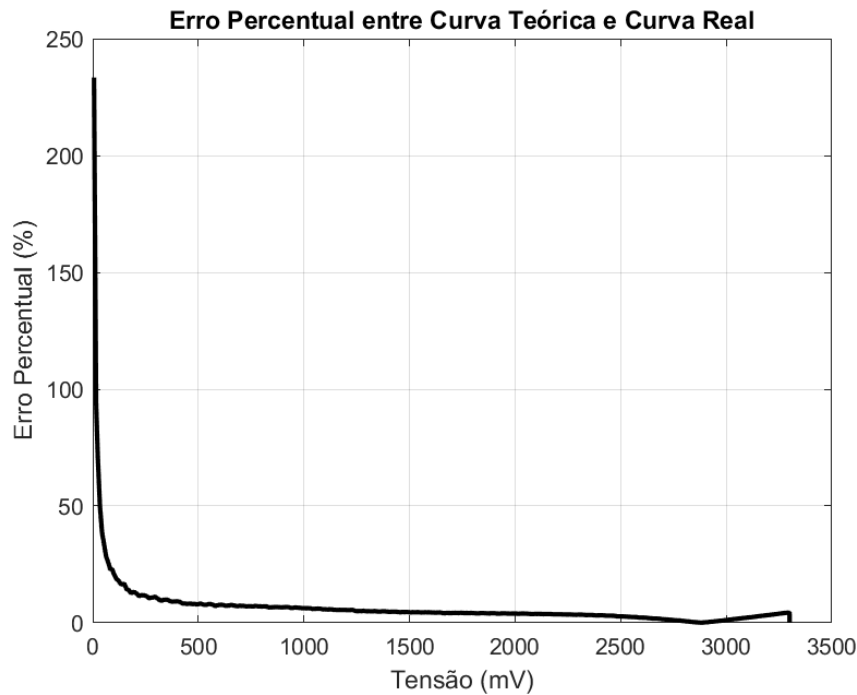
Figura 9 – Curva experimental ESP32 LoRa



Fonte: Elaborada pelo autor, 2024.

Para quantificar a magnitude do erro na curva experimental, foi calculado o erro percentual ao longo de toda a curva de medição. A análise dos resultados, conforme ilustrado na Figura 10, revela discrepâncias significativas entre os valores medidos e os valores esperados, destacando a necessidade de calibração do ADC do ESP32. O erro absoluto médio registrado foi de 58,52mV, enquanto o valor médio do erro percentual foi de 6,92%. Esses números indicam que o desempenho do ADC está abaixo do ideal, tornando indispensável o ajuste para aprimorar a precisão e reduzir as inconsistências nas leituras.

Figura 10 – Erro percentual entre curva teórica e curva real



Fonte: Elaborada pelo autor, 2024.

Com base nos dados obtidos da curva experimental, serão implementadas três abordagens distintas de correção para o ADC do ESP32, utilizando ajustes de curvas, como regressão linear, regressão polinomial e a biblioteca de calibração da Espressif. A regressão linear será aplicada para estabelecer uma relação direta entre os valores medidos pelo ADC e as tensões reais, sendo adequada para corrigir erros com comportamento não linear. Por outro lado, a regressão polinomial permitirá capturar variações não lineares mais complexas, proporcionando melhor ajuste, especialmente nas extremidades da faixa de medição.

A terceira estratégia utilizará a biblioteca de calibração da Espressif, que incorpora algoritmos avançados projetados para corrigir o comportamento do ADC de forma mais precisa e eficiente. O objetivo dessas três abordagens é avaliar a redução dos erros de leitura, comparar a eficácia de cada método e identificar qual oferece o melhor desempenho na correção dos desvios apresentados pelo ADC, otimizando assim a precisão das medições em todo o intervalo de operação.

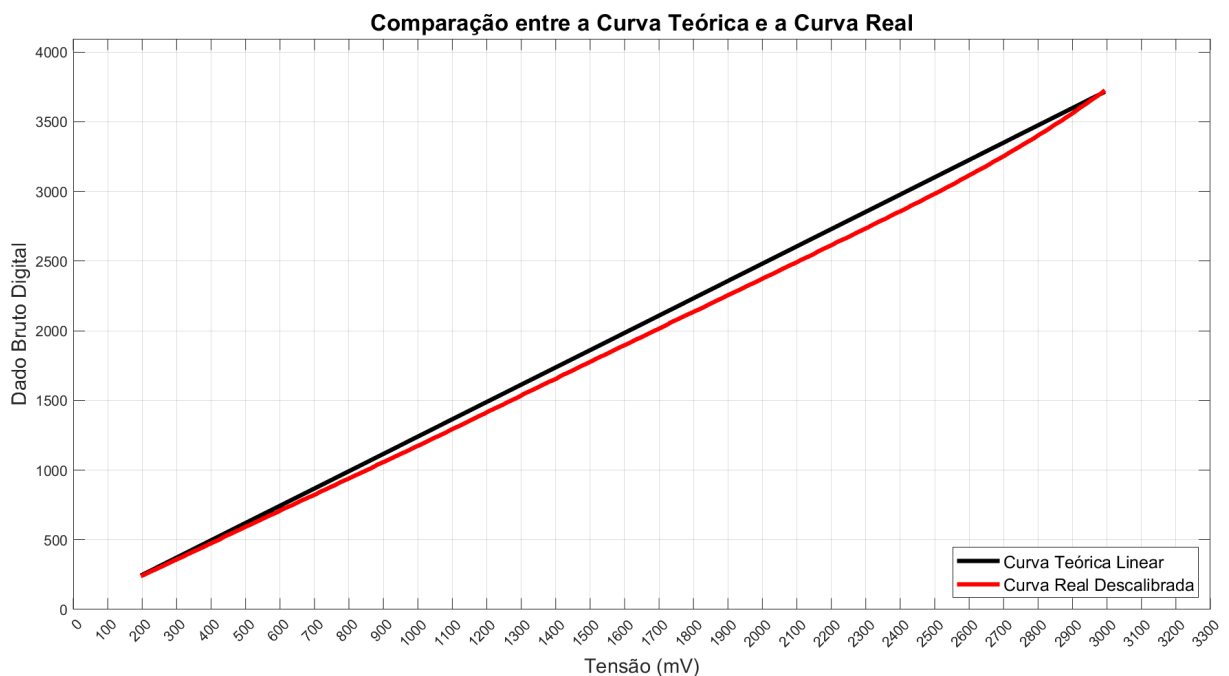
### 3.3.1 Resultados da linearização do ADC ESP32 LoRa

Conforme discutido, foram implementados três métodos de linearização para o ADC do ESP32 LoRa: regressão linear, regressão polinomial e a biblioteca de calibração oficial da Espressif. Cada técnica foi aplicada para minimizar os desvios não lineares intrínsecos ao ADC do ESP32, permitindo uma significativa redução nos erros de leitura, especialmente em comparação aos testes iniciais realizados sem qualquer correção.

Devido às limitações identificadas no ADC do ESP32, que apresenta inconsistências nas extremidades do intervalo de 0 a 3300 mV, foi definido um intervalo de operação ajustado entre 200 mV e 3000 mV para este projeto, evitando as áreas mais propensas a distorções. Essa faixa foi escolhida também por ser compatível com o sensor de pressão utilizado, cuja saída analógica varia de 335 mV a 2949 mV, assegurando que as leituras permaneçam dentro da região mais linear do ADC.

Os resultados da calibração e linearização serão apresentados em três gráficos comparativos, cada um destacando a performance após a aplicação dos métodos. Esses gráficos permitem uma análise detalhada da eficácia de cada técnica de linearização, demonstrando a redução de erros e a melhoria na linearidade das leituras.

Figura 11 – ADC descalibrado limitado 200mV a 3000mV



Fonte: Elaborada pelo autor, 2024.

Conforme expresso na Figura 11 para o intervalo de operação do ADC do ESP32, que vai de 200 mV a 3000 mV, foi observado um erro absoluto médio de 61,4 mV e um erro percentual médio de 4,0%. Esses resultados demonstram que as extremidades da faixa de leitura do ADC no ESP32 apresentam uma distorção significativa, afetando a linearidade das medições e comprometendo a precisão.

Inicialmente, a calibração será realizada por meio de regressão polinomial, gerando curvas de correção. Nesse processo, foram obtidas duas curvas de sétima ordem, uma de quarta ordem e outra de primeira ordem. Para determinar as funções, foram coletados 50 valores de referência, variando a tensão de 0 a 3300 mV por meio de um divisor de tensão. Os valores de tensões foram medidos com um multímetro, já os dados brutos digitais foram registrados

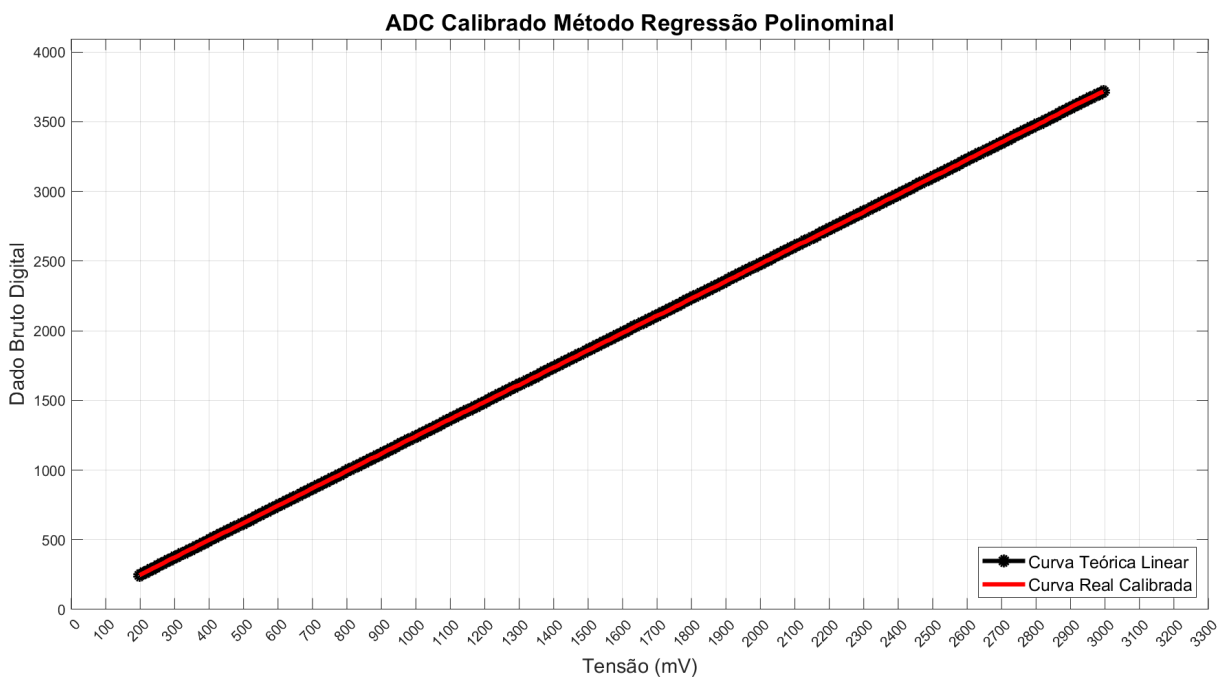
através do monitor serial da IDE do Arduino. Para garantir a precisão e a confiabilidade dos resultados, o desvio padrão foi calculado para cada amostra, fornecendo uma análise estatística da variabilidade dos dados.

A utilização de três funções polinomiais, em vez de uma única, se deve à necessidade de aumentar a precisão da calibração em diferentes faixas de operação do ADC. Ao empregar uma única função polinomial para representar todo o intervalo de trabalho do ADC, frequentemente surgem erros de ajuste, especialmente em dispositivos que apresentam distorções ou não-linearidades em faixas específicas, como ocorre no ADC do ESP32.

Dividir o intervalo de operação em subintervalos e aplicar uma função polinomial específica a cada um deles permite que cada polinômio seja otimizado para uma faixa de valores mais estreita, onde os dados experimentais são mais facilmente modelados com precisão. Esse método de ajuste segmentado, ou por partes, reduz significativamente o erro de calibração, já que cada polinômio se ajusta melhor à curva real em seu intervalo específico.

Assim, as duas funções de sétimo grau foram utilizadas para capturar a complexidade e a não-linearidade nas regiões mais sensíveis, enquanto as funções de quarto e primeiro grau foram aplicadas em faixas onde a variação era mais previsível. Esse modelo segmentado resulta em um ajuste mais eficiente e preciso do ADC, reduzindo o erro em cada faixa e proporcionando uma calibração mais precisa do que uma única função.

Figura 12 – ADC calibrado método regressão polinomial limitado 200mV a 3000mV



Fonte: Elaborada pelo autor, 2024.

A Figura 12 mostra o comportamento do ADC do ESP32 após a calibração utilizando o método de regressão polinomial, comparando a curva calibrada (em vermelho) com a linha

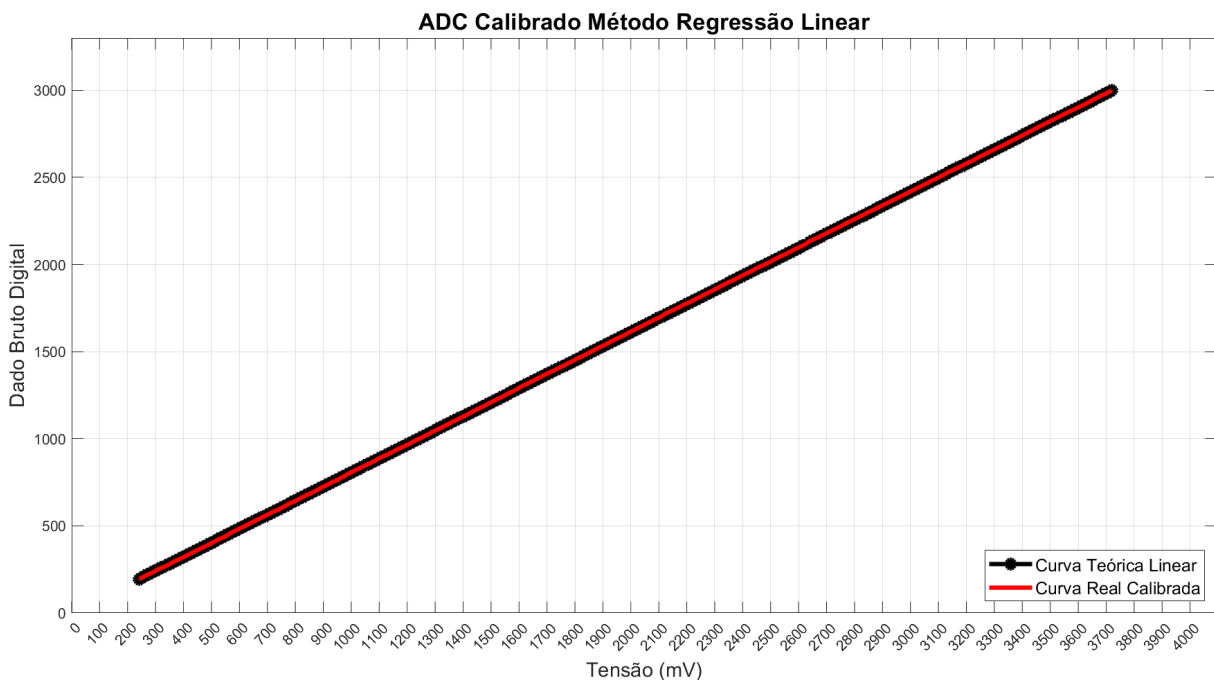
teórica (em preto).

A linha preta representa a curva teórica ideal de uma relação linear entre o valor de entrada analógico em mV e o dado bruto digital. Já a curva em vermelho representa os dados obtidos após a aplicação das funções de calibração polinomial, demonstrando um alinhamento quase perfeito com a linha teórica ao longo de toda a faixa de 200 mV a 3000 mV.

A proximidade entre as duas curvas indica que o método de calibração foi bem-sucedido em corrigir as distorções não-lineares típicas do ADC do ESP32. O erro absoluto médio foi reduzido para aproximadamente 1,3962 mV, enquanto o erro percentual médio caiu para 0,11535%. Esses resultados indicam que a calibração foi eficaz, praticamente eliminando as variações detectadas antes da correção.

O método de calibração descrito a seguir envolve a modelagem de 15 retas de calibração, todas baseadas nos mesmos 50 valores coletados anteriormente. Essas retas foram geradas a partir de intervalos específicos, sempre buscando o melhor comportamento linear para o sistema.

Figura 13 – ADC calibrado método regressão linear limitado 200mV a 3000mV

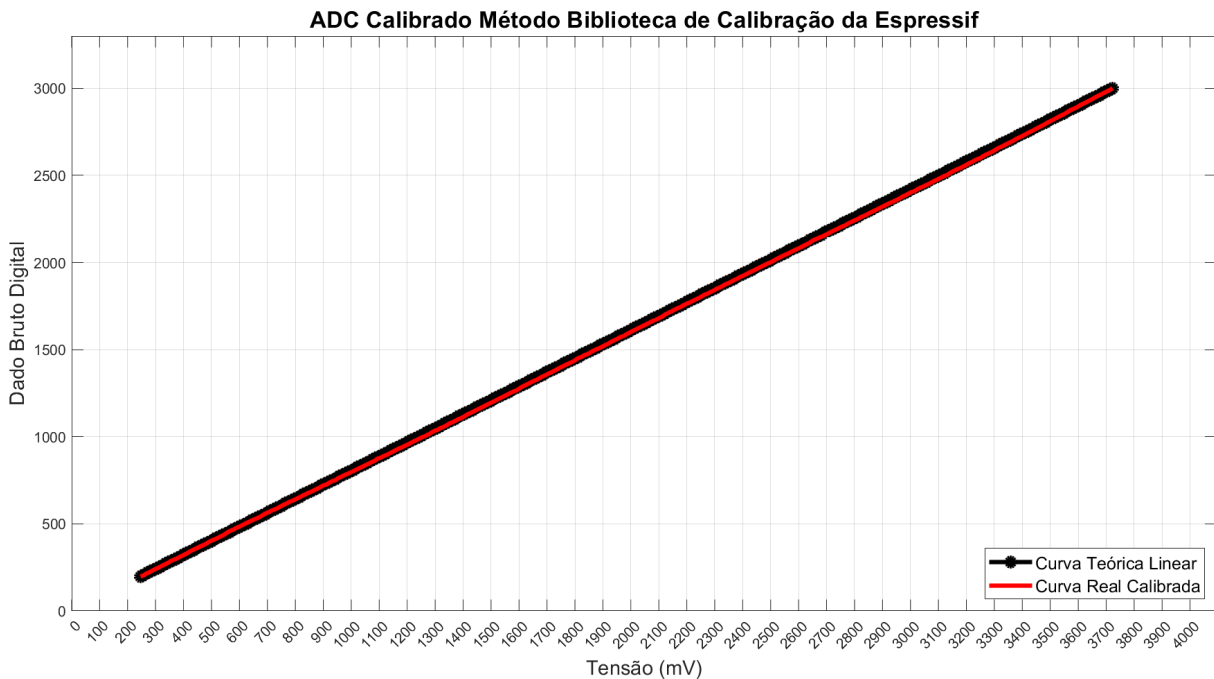


Fonte: Elaborada pelo autor, 2024.

A Figura 13 ilustra o comportamento do ADC do ESP32 após a calibração utilizando o método de regressão linear. A proximidade entre as curvas pós-calibração e a referência demonstra a eficácia do método em reduzir a não linearidade do ADC do ESP32. O erro absoluto médio foi reduzido para aproximadamente 1,8986 mV, e o erro percentual médio é igual a 0,15578%, indicando uma significativa melhora na precisão das leituras. Embora a regressão linear tenha proporcionado uma correção eficaz e um desempenho satisfatório, a calibração utilizando o método de regressão polinomial apresentou resultados ainda mais precisos.

O método de calibração descrito a seguir é baseado em uma biblioteca de calibração desenvolvida pela Espressif.

Figura 14 – ADC calibrado método biblioteca de calibração da Espressif limitado 200mV a 3000mV



Fonte: Elaborada pelo autor, 2024.

A Figura 14 ilustra o desempenho do ADC do ESP32 após a calibração utilizando a biblioteca de calibração fornecida pela Espressif. Assim como nos dois métodos descritos anteriormente, a proximidade entre as curvas calibradas e a curva de referência demonstra a eficiência da calibração em corrigir a não linearidade do ADC. Como resultado, foi alcançado um erro absoluto médio de aproximadamente 10,1607 mV e um erro percentual médio de 0,6703%, representando uma melhoria significativa na precisão das leituras.

Diante dos resultados obtidos, a calibração por regressão polinomial demonstrou maior eficiência em corrigir as imprecisões do ADC do ESP32, superando os outros métodos em termos de precisão. Esse método será, portanto, adotado ao longo do projeto, uma vez que apresentou uma taxa de erro significativamente menor e uma maior proximidade com os valores de referência. A escolha se justifica pela capacidade da regressão polinomial de ajustar com precisão a não linearidade inerente do ADC, resultando em medições mais consistentes e confiáveis.

### 3.4 Condicionamento de sinal

Devido à diferença de tensão entre o sensor e o microcontrolador, é fundamental realizar um condicionamento de sinal. Esse processo ajusta o sinal analógico de saída do sensor, que opera a 5V, para a faixa de 3,3V suportada pelo ESP32 LoRa. O condicionamento deve garantir

a preservação da linearidade do sinal, assegurando que a transferência dos dados adquiridos pelo sensor ao microcontrolador seja exata e sem distorções. Isso é fundamental para garantir a integridade das medições e a confiabilidade na aquisição de dados analógicos.

Tendo isso em mente, será utilizado um divisor de tensão resistivo para ajustar a saída do sensor à faixa de 3,3V na entrada do ESP32 LoRa. Conhecendo-se os valores de  $V_{in}$  (tensão de entrada) e  $V_{out}$  (tensão de saída desejada), podemos definir um valor apropriado para  $R_1$  ou  $R_2$  e, a partir disso, calcular o outro resistor para obter a tensão de 3,3V. As fórmulas para divisor de tensão e divisor de corrente baseiam-se nas Leis de Kirchhoff e na Lei de Ohm.

A tensão de saída  $V_{out}$  do divisor de tensão é dada pela Equação 3.1:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2}, \quad (3.1)$$

onde:

- $V_{in}$  é a tensão de entrada,
- $V_{out}$  é a tensão desejada na entrada do ESP32,
- $R_1$  e  $R_2$  são os resistores do divisor de tensão.

A partir dessa relação, podemos determinar os valores de  $R_1$  e  $R_2$  para alcançar a tensão desejada de 3,3V.

### 3.5 Sensor de pressão XGZP6847A

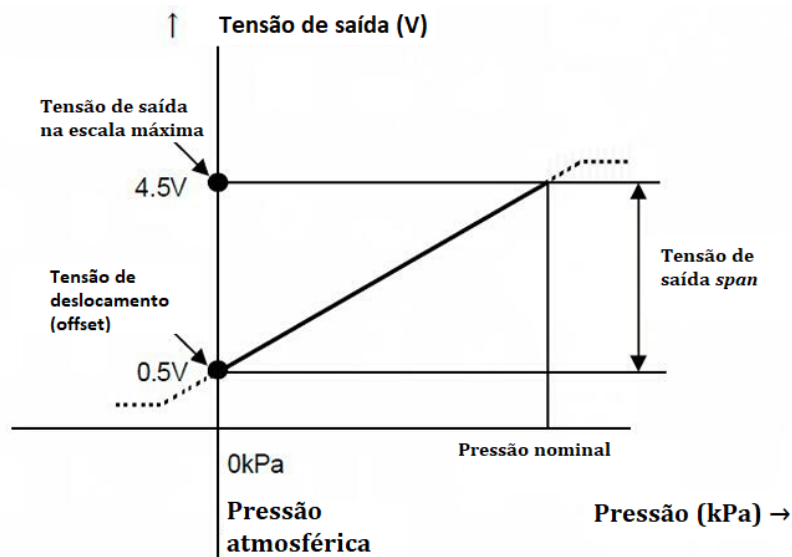
Esta etapa diz respeito a verificar a precisão e consistência das medições feitas pelo sensor de pressão XGZP6847A. Os testes operacionais incluirão resultados obtidos em diferentes níveis de pressão, garantindo que as leituras estejam alinhadas com as expectativas de desempenho e as especificações técnicas do dispositivo. A análise visa confirmar a operação precisa e estável do sensor em toda a sua faixa de medição, validando a adequação dos dados para aplicações que exigem alta confiabilidade e precisão.

O experimento será montado em um protoboard, onde o sensor XGZP6847A será conectado a um sistema de fornecimento de pressão controlada. Um cilindro de ar comprimido medicinal será utilizado para injetar pressão no orifício de medição do sensor, enquanto uma válvula reguladora de pressão, acoplada a um manômetro, permitirá o ajuste preciso da pressão aplicada.

À medida que a pressão for ajustada, os valores analógicos gerados pelo sensor serão monitorados e comparados com as referências especificadas no datasheet do componente. O foco será verificar se a linearidade apresentada na Figura 15 extraída do datasheet do fabricante

corresponde à realidade, e se a resposta do sensor em diferentes níveis de pressão está em conformidade com as especificações técnicas, garantindo a confiabilidade dos dados obtidos.

Figura 15 – Curva de pressão XGZP6847A



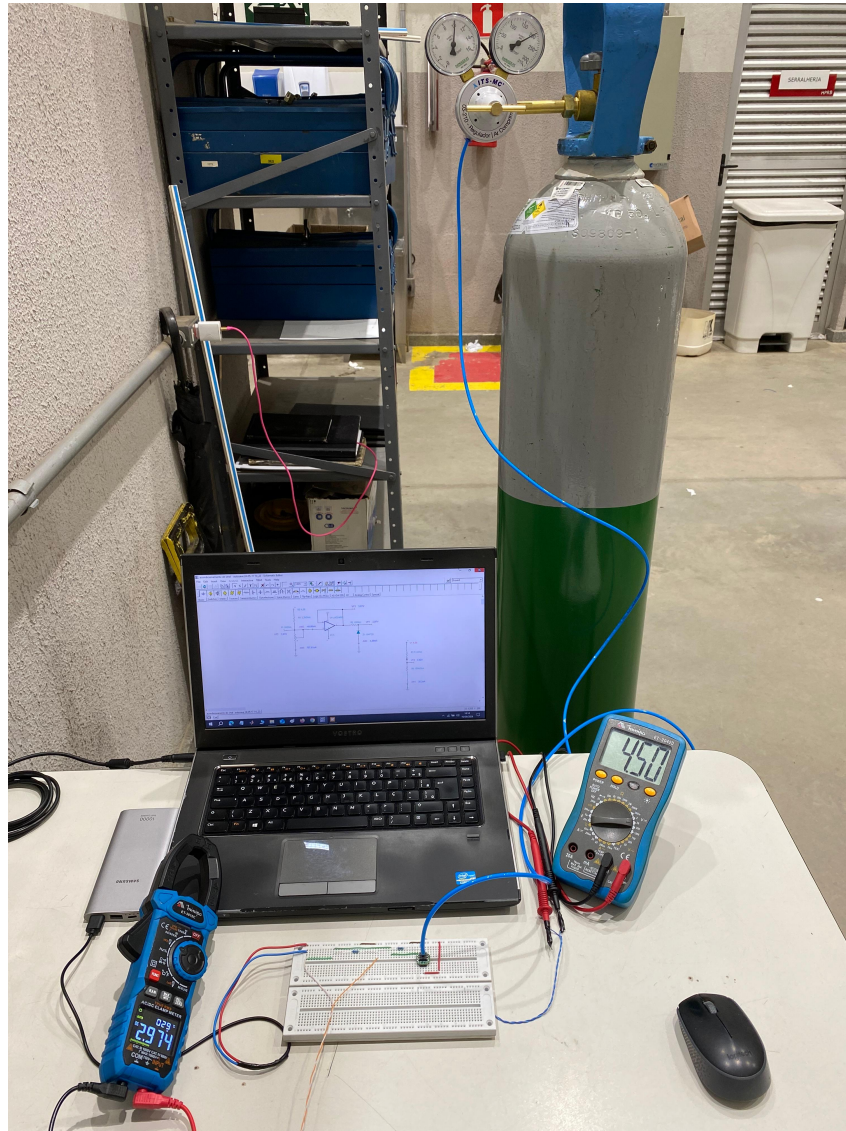
Fonte: (CFsensor, 2024).

### 3.5.1 Resultado teste funcional sensor de pressão XGZP6847A

Durante a fase de validação, o sensor de pressão XGZP6847A foi submetido a testes para avaliação de sua precisão e estabilidade nas medições. Conforme ilustrado na Figura 16, os testes consistiram na aplicação de pressão controlada por meio de um cilindro de ar comprimido medicinal. A pressão foi regulada com o uso de uma válvula reguladora de pressão e monitorada por um manômetro com resolução mínima de 100 mbar.

A pressão foi aplicada em incrementos de 100 mbar, com um tempo de estabilização de 2 minutos em cada ponto, para permitir a leitura adequada do sensor. Os valores medidos pelo sensor foram comparados com os valores de referência do manômetro.

Figura 16 – Teste funcional sensor de pressão XGZP6847A



Fonte: Elaborada pelo autor, 2024.

A coleta de dados foi realizada com o suporte de dois multímetros e registrados na Tabela 3. Um deles monitorou a saída analógica direta do sensor, cuja faixa de operação está especificada em 0,5V a 4,5V, enquanto o outro aferiu a tensão após o divisor resistivo ajustado para uma saída máxima de 3,3V, adequada ao ADC do ESP32. Este procedimento permitiu uma análise detalhada e comparativa da resposta do sensor com os valores de referência especificados no datasheet.

Para assegurar que os dados coletados estivessem dentro dos limites de tensão aceitáveis pelo ADC do ESP32, realizamos o condicionamento de sinal utilizando um divisor de tensão resistivo. Inicialmente, foi fixado o valor de  $R_2 = 100\text{ k}\Omega$ . Com base nos valores conhecidos de tensão de entrada e saída, calculamos o valor adequado para o resistor  $R_1$  de modo a obter a tensão de saída desejada.

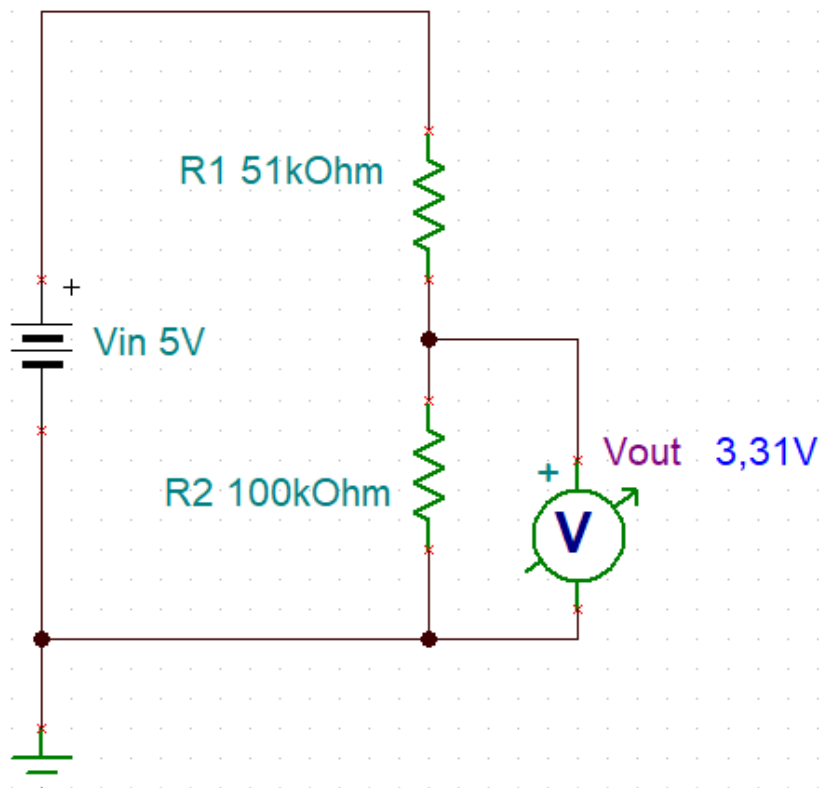
Parâmetros utilizados:

- $V_{in} = 5V$  Tensão de saída do sensor de pressão,
- $V_{out} = 3.3V$  Tensão desejada na entrada do ADC,
- $R_2 = 100k\Omega$  1% Resistor de referência no divisor de tensão.

Aplicando esses valores à Equação (3.1) do divisor de tensão, obteve-se um valor calculado de  $R_1 \approx 51k\Omega$ . Assim, foi escolhido um resistor comercial de  $51k\Omega$  com tolerância de 1% para  $R_1$ , garantindo uma redução eficiente e segura da tensão de saída do sensor para níveis compatíveis com o ADC do ESP32.

O uso de um resistor de precisão de 1% em um divisor de tensão pode impactar a exatidão das medições, pois sua tolerância permite uma variação de até 1% do valor nominal. No projeto, o resistor  $R_1$  foi calculado como  $R_1 \approx 51k\Omega$ , mas, devido à tolerância, seu valor real pode variar entre  $R_1 = 50,49k\Omega$  e  $R_1 = 51,51k\Omega$ . Essa variação pode influenciar a tensão de saída  $V_{out}$ , afetando a leitura do ADC do ESP32 e, conseqüentemente, as medições de pressão.

Figura 17 – Circuito divisor de tensão



Fonte: Elaborada pelo autor, 2024.

Embora o erro introduzido por um resistor de 1% seja geralmente pequeno, ele pode ser significativo em aplicações que exigem alta precisão, afetando a calibração do sistema. No entanto, para este projeto, essa tolerância é aceitável, pois não será necessário medir pressões inferiores a 1 mbar. Para aplicações em que a exatidão é essencial, recomenda-se o uso de resistores com tolerância menor, como 0,1% ou 0,05%. Em aplicações comuns, um resistor de 1% é uma alternativa econômica, garantindo um desempenho adequado sem comprometer significativamente a qualidade dos dados.

Neste projeto, foi possível utilizar um divisor resistivo para o condicionamento de sinal devido à alta impedância de entrada do ADC do ESP32. Essa característica minimiza a carga no circuito divisor, permitindo que a tensão de saída seja medida de forma precisa sem afetar significativamente o sinal, o que garante a integridade dos dados coletados.

Com base na análise da Tabela 3 e nas especificações do *datasheet* do sensor de pressão XGZP6847A, os testes práticos realizados com o sensor, integrado a um circuito divisor de tensão, comprovaram uma resposta linear e estável para pressões na faixa de 0 a 10 bar. Esse desempenho está em pleno acordo com as características técnicas do sensor, que indicam uma relação linear entre a pressão aplicada e o sinal de saída em sua faixa de operação.

Tabela 3 – Teste funcional do sensor de pressão XGZP6847A

<b>Pressão em Bar</b>	<b>Saída Sensor de Pressão (0,5V a 4,5V)</b>	<b>Saída Divisor de Tensão (Limitada a 3,3V)</b>
0	0,50V	0,34V
1	0,93V	0,62V
2	1,31V	0,87V
2,5	1,50V	1,00V
3	1,71V	1,14V
4	2,09V	1,39V
5	2,50V	1,65V
6	2,90V	1,92V
7	3,28V	2,17V
7,5	3,48V	2,30V
8	3,67V	2,43V
9	4,07V	2,69V
10	4,47V	2,97V

Fonte: Elaborada pelo autor, 2024.

Com base nos testes experimentais conduzidos com o auxílio de um manômetro de escala graduada em incrementos de 100 mbar, foi possível confirmar que o sensor de pressão XGZP6847A possui resolução suficiente para detectar variações de pressão a cada 100 mbar. A cada incremento de 100 mbar, a saída do sensor apresenta um aumento de aproximadamente 40 mV, o que permite ao sistema de aquisição interpretar essas variações. Essa resolução de leitura de 100 mbar atende plenamente aos requisitos do projeto, garantindo que mudanças significativas na pressão sejam detectadas para esta aplicação.

Conforme mencionado anteriormente, o ADC do ESP32 não suporta tensões maiores que 3,3V. Para adaptar a saída do sensor, foi utilizado um divisor de tensão, e as especificações do sistema foram definidas da seguinte forma.

O *span* representa a diferença entre o valor máximo e o valor mínimo da faixa de saída ajustada:

$$Span = 2950\text{ mV} - 335\text{ mV} = 2615\text{ mV}.$$

Isso significa que a faixa total de variação de saída do sensor, após o divisor de tensão, é de 2615 mV para o intervalo de 0 a 10 bar.

O *range* do ADC, conforme ajustado pelo divisor de tensão, é de:

$$Range = 335\text{ mV a } 2950\text{ mV}.$$

A resolução do ADC depende do número de passos digitais (ou níveis discretos) que ele utiliza para representar o sinal analógico entre 335 mV e 2950 mV.

Assumindo que:

- 335 mV gera aproximadamente o valor digital 416.
- 2950 mV gera aproximadamente o valor digital 3661.

Assim, o número de passos ou níveis disponíveis no ADC para esse *range* será:

$$\text{Níveis Disponíveis} = 3661 - 416 = 3245 \text{ níveis}.$$

Agora, a resolução em termos de mV por passo digital é dada por:

$$\text{Resolução} = \frac{Span}{\text{Níveis Disponíveis}} = \frac{2615\text{ mV}}{3245} \approx 0,806\text{ mV/passos}.$$

Essa resolução indica que o ADC consegue detectar variações de aproximadamente 0,806 mV para cada incremento no valor digital.

O fundo de escala do sistema, considerando o *range* ajustado pelo divisor de tensão, será:

- Tensão de fundo de escala: 2950 mV.
- Valor digital correspondente: 3661 (aproximadamente).

Para determinar a relação entre o incremento de pressão no sensor e a resposta do ADC, partimos dos valores:

- *Span*: 2615 mV.
- Pressão máxima:  $P_{\max} = 10000$  mbar.
- Pressão mínima:  $P_{\min} = 0$  mbar.

A variação total de pressão é:

$$\Delta P = P_{\max} - P_{\min} = 10000 \text{ mbar} - 0 \text{ mbar} = 10000 \text{ mbar}.$$

Portanto, a relação em mV por mbar é:

$$\text{Relação} \left( \frac{\text{mV}}{\text{mbar}} \right) = \frac{\text{Span}}{\Delta P} = \frac{2615 \text{ mV}}{10000 \text{ mbar}} = 0,261 \frac{\text{mV}}{\text{mbar}}.$$

Essa relação de 0,261 mV/mbar indica que, para cada aumento de 1 mbar na pressão, a saída do sensor aumenta em 0,261 mV. Comparando essa sensibilidade com a resolução do ADC de 0,806 mV por passo digital, podemos concluir que:

- a) A cada incremento de 100 mbar, a saída do sensor aumenta em 26,1 mV, o que equivale a aproximadamente 32 passos digitais do ADC;
- b) Isso assegura que o sistema possa detectar incrementos de 100 mbar com precisão suficiente, pois a variação no valor digital será significativa.

Essas configurações garantem que a saída do sensor esteja devidamente ajustada à faixa de entrada do ADC do ESP32, permitindo uma leitura precisa e confiável dentro dos limites de operação do microcontrolador.

Diante das análises e dos ajustes de calibração implementados, é possível concluir que o sistema desenvolvido, com base no ESP32, proporciona uma solução viável e eficiente para a leitura de pressão dentro dos parâmetros estabelecidos. Ao utilizar um divisor de tensão, conseguimos adaptar a saída do sensor para a faixa de operação do ADC do ESP32, mantendo a precisão das leituras ao evitar a saturação da entrada analógica.

O cálculo do *span*, do *range* e da resolução do ADC revelou que o sistema é capaz de detectar variações de 0,806 mV para cada incremento digital, garantindo uma sensibilidade adequada para aplicações que demandam precisão. Com uma relação de 0,261 mV/mbar, cada incremento de 100 mbar corresponde a uma variação de 26,1 mV, ou cerca de 32 passos digitais do ADC.

Além disso, a faixa de operação de 0 a 10 bar está alinhada com as especificações do sensor, permitindo uma leitura completa da faixa de pressão sem comprometimento de dados. Os testes realizados também indicam que o ESP32 pode processar essas informações com confiabilidade, representando uma alternativa de baixo custo e flexível para aplicações que exigem monitoramento contínuo e resposta rápida a variações de pressão.

Em resumo, o sistema apresentado cumpre os critérios de precisão e alcance operacional necessários para este projeto, mostrando-se apropriado para aplicações que exigem monitoramento de pressão.

Quadro 2 – Parâmetros do sensor de pressão e ADC

<b>Parâmetro</b>	<b>Valor</b>	<b>Descrição</b>
<b>Span</b>	2615 mV	Diferença entre o valor máximo e mínimo da faixa de saída ajustada do sensor.
<b>Range (faixa)</b>	335 mV a 2950 mV	Faixa de tensão após o divisor de tensão (valores ajustados para o ADC do ESP32).
<b>Resolução (ADC)</b>	0,806 mV/passos	Resolução do ADC, representando a variação de tensão mínima que o ADC consegue detectar.
<b>Fundo de Escala</b>	2950 mV	Tensão máxima que o sistema pode medir, após o divisor de tensão.
<b>Valor Digital FS</b>	3661	Valor digital correspondente à tensão máxima do fundo de escala no ADC.
<b>Relação mV/mbar</b>	0,261 mV/mbar	Sensibilidade do sensor, indicando a variação de tensão para cada 1 mbar de pressão.
<b>Incremento de Pressão (100 mbar)</b>	26,1 mV	Variação de tensão correspondente a um incremento de 100 mbar.
<b>Incremento ADC (100 mbar)</b>	32 passos digitais	Variação no valor digital do ADC para cada incremento de 100 mbar de pressão.

Fonte: Elaborada pelo autor, 2025.

### 3.6 Placa de circuito impresso

A prototipagem é uma etapa essencial no desenvolvimento de circuitos eletrônicos. A utilização de uma *protoboard* permite a rápida montagem e teste de circuitos, facilitando a experimentação e a validação de ideias. No entanto, apesar das vantagens oferecidas pela *protoboard* na fase inicial de um projeto, a transição para uma Placa de Circuito Impresso (PCB) é crucial para a implementação final de qualquer dispositivo eletrônico.

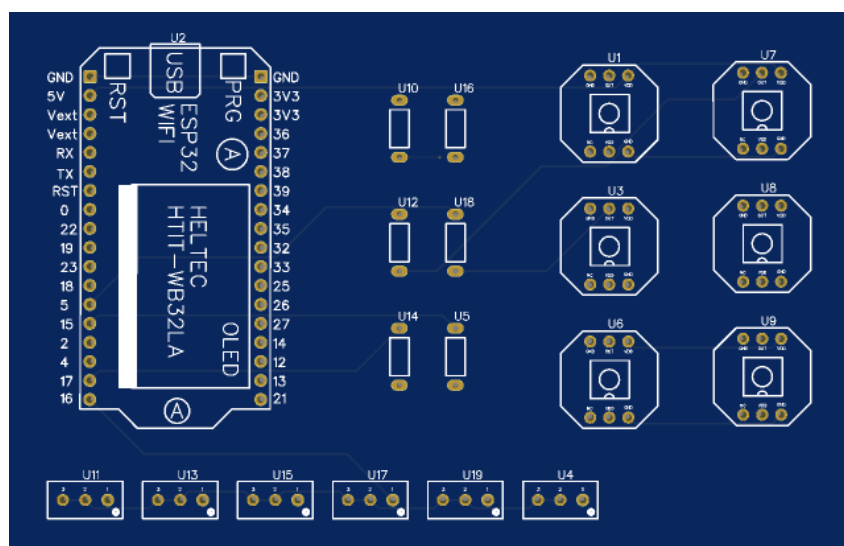
Para essa etapa do projeto, vamos trabalhar com o *software* EasyEDA, uma ferramenta específica para desenvolvimento de PCB. O primeiro passo será criar um esquemático do circuito. Nesse estágio, é necessário inserir e conectar corretamente todos os componentes, incluindo o filtro capacitivo, os *pads* para inserção do ESP32 LoRa e o divisor de tensão resistivo, que ajustará a tensão do sensor à faixa de 3,3V.

Após a conclusão do esquemático, o próximo passo é convertê-lo para a vista de *layout* do PCB. Nesta etapa, deve-se posicionar os componentes na placa e usar as ferramentas de roteamento do EasyEDA.

Depois de finalizar o *layout*, é preciso gerar os arquivos *Gerber*, que contêm todas as informações necessárias para a fabricação da PCB. Esses arquivos podem ser enviados diretamente para uma fabricante de PCBs, para dar início à produção. Assim, o EasyEDA oferece um fluxo de trabalho completo, desde a criação do esquemático até a produção final da PCB para a central de alarmes.

A Figura 18 ilustra a aparência da placa PCB que está em desenvolvimento.

Figura 18 – Placa PCB em desenvolvimento



Fonte: Elaborada pelo autor, 2024.

### 3.6.1 Projeto e fabricação da PCB

Esta etapa do projeto visa a montagem da PCB destinada à aquisição dos dados de pressão, fundamental para a implementação do sistema de monitoramento. A construção da PCB foi precedida pela validação do circuito eletrônico em *proto-board*. Nessa fase preliminar, foram realizados testes de linearização do ADC do ESP32, avaliação funcional dos sensores e condicionamento de sinal utilizando um divisor resistivo, garantindo a adequação das leituras de pressão dentro da faixa suportada pelo microcontrolador.

O design do circuito eletrônico e o roteamento da PCB foram realizados com o auxílio do *software* EasyEDA, que permitiu a criação precisa do *layout* da placa, organizando os componentes e otimizando o percurso dos sinais. Um aspecto essencial no projeto foi a inclusão de um plano de terra, que garante uma referência comum para todos os componentes e minimiza interferências eletromagnéticas. Esse plano de terra é especialmente importante para manter a integridade do sinal e reduzir ruídos indesejados que poderiam afetar a precisão das leituras de pressão.

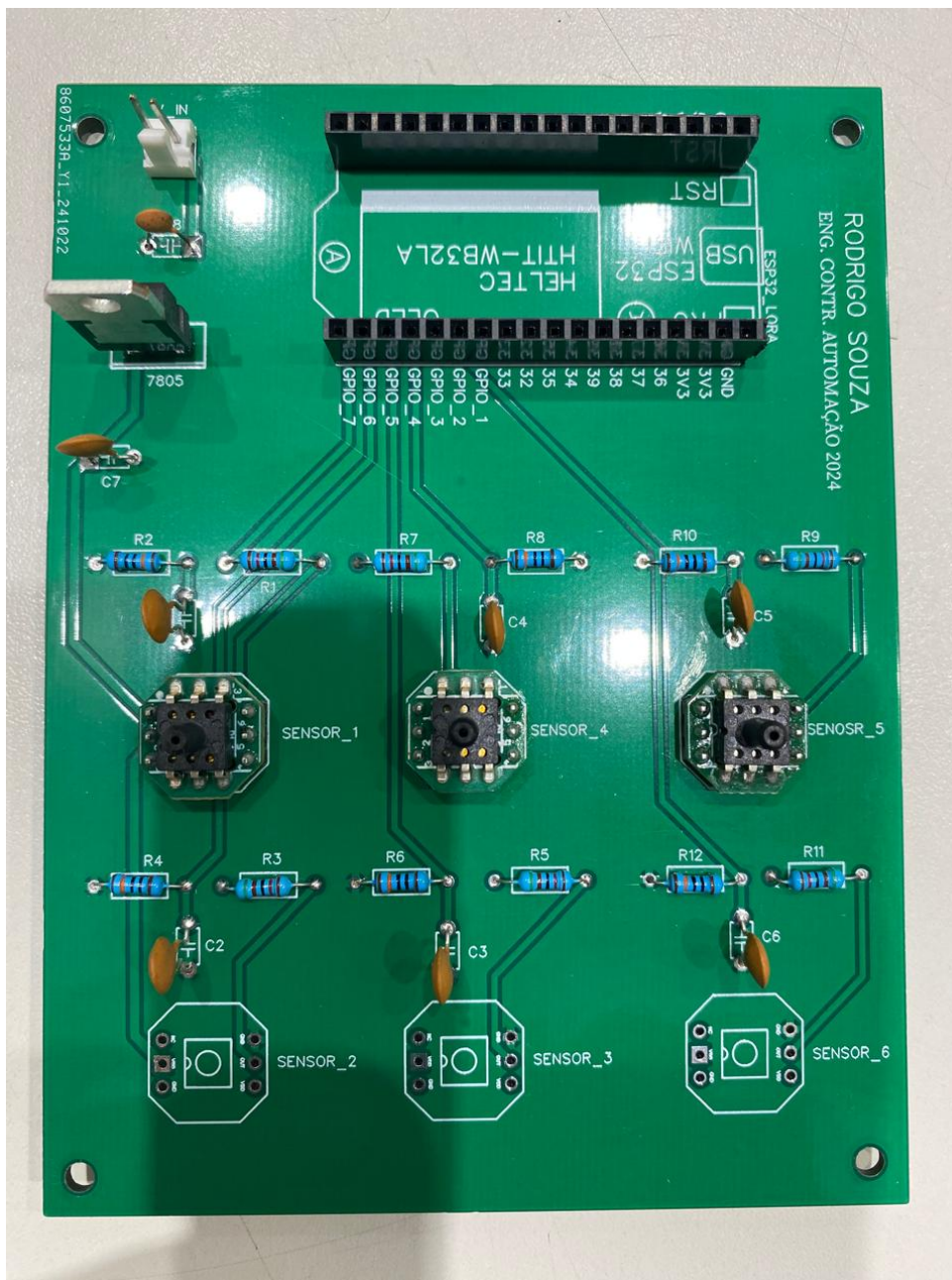
Além disso, foram integrados capacitores de filtro em pontos estratégicos do circuito. Esses capacitores atuam na redução de ruídos de alta frequência e estabilizam a alimentação, evitando flutuações que poderiam introduzir erros nas medições e comprometer a qualidade dos dados coletados.

O arquivo *Gerber*, contendo todos os dados necessários para a fabricação da PCB, foi gerado a partir do projeto desenvolvido no *software* EasyEDA. Este arquivo inclui informações detalhadas sobre o *layout* da placa, como trilhas, furos, planos de terra, e posicionamento dos componentes, permitindo que o fabricante reproduza fielmente o design projetado.

Após a finalização e verificação do arquivo *Gerber*, este foi enviado para a empresa JLCPCB, encarregada da fabricação da PCB. Esse processo assegura que a placa seja produzida com precisão, seguindo rigorosamente as especificações do projeto, o que é fundamental para garantir qualidade e confiabilidade na montagem e na futura implementação do sistema de aquisição de dados de pressão.

A Figura 19 apresenta a placa de dupla camada finalizada, pronta para ser utilizada no sistema.

Figura 19 – Placa de circuito impresso final



Fonte: Elaborada pelo autor, 2024.

### 3.7 Montagem do protótipo

O desenvolvimento do protótipo será realizado no laboratório do IFMG Campus Betim, pois é onde estão disponíveis os recursos e equipamentos necessários para prototipagem e instrumentos de medição. Caso seja necessário, parte do trabalho poderá ocorrer no setor de manutenção, especialmente para integração final e testes em campo.

A aquisição de todos os materiais necessários para a execução do projeto será conduzida diretamente pelo autor, Rodrigo de Souza Gonçalves, garantindo a conformidade com as especificações técnicas e os requisitos do projeto.

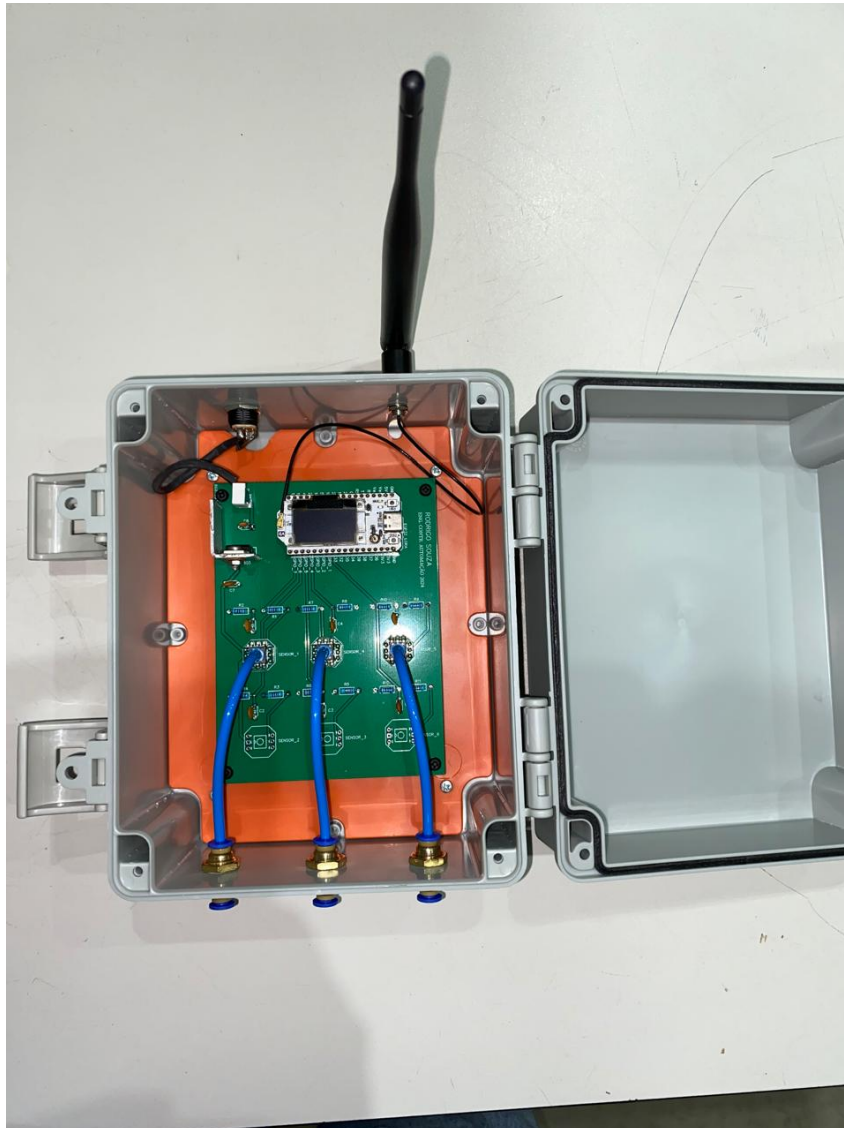
Figura 20 – Montagem final central de alarmes



Fonte: Elaborada pelo autor, 2024.

A Figura 20 mostra o início da montagem da central de alarmes, que funcionará como módulo transmissor. O primeiro passo foi realizar as furações para fixar os conectores pneumáticos, responsáveis por receber as pressões do sistema. Em seguida, foram instalados a antena 2 (modelo TX915-JKS-IPX20) e o conector Jack P4. Por fim, fixou-se o espelho, juntamente com a placa PCB previamente montada, contendo todos os componentes devidamente soldados.

Figura 21 – Central de alarmes módulo transmissor

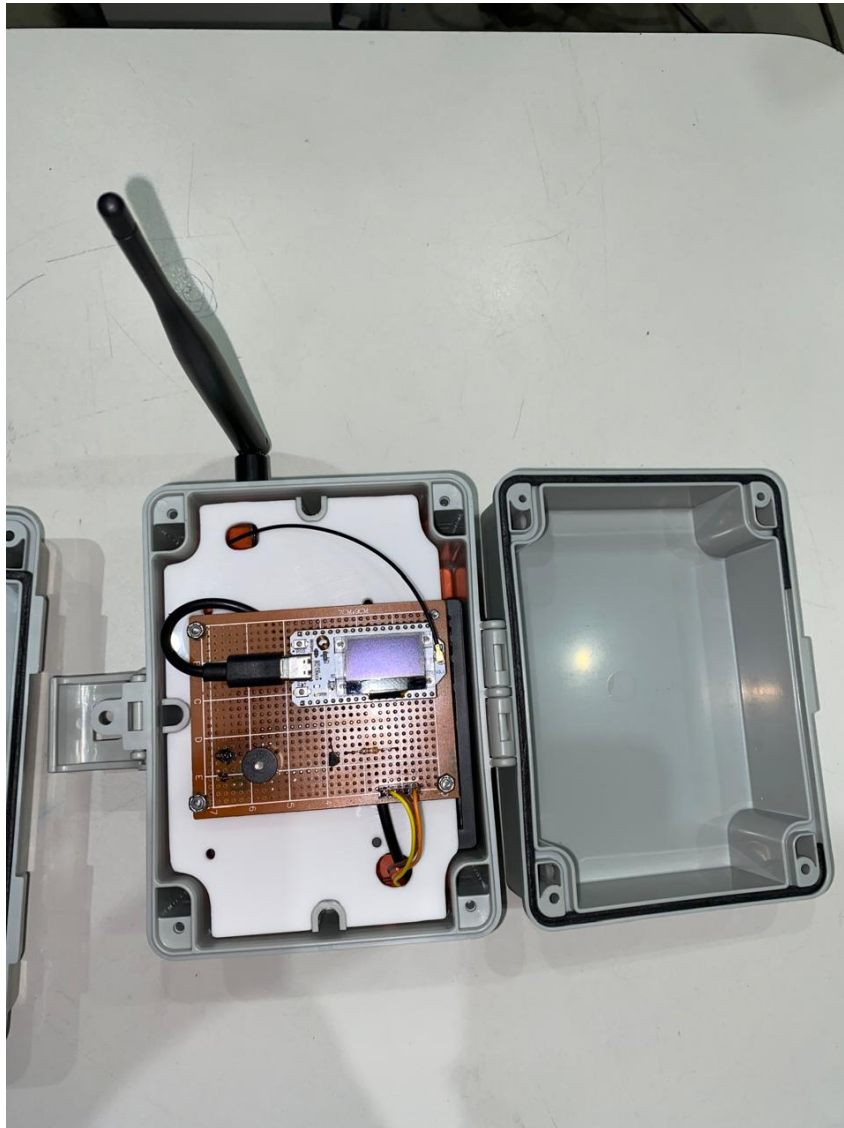


Fonte: Elaborada pelo autor, 2024.

A Figura 21 apresenta a montagem final do hardware instalado em uma caixa hermética, projetada para proteger os componentes eletrônicos de condições ambientais adversas, como poeira e umidade. O sistema é composto por uma placa de circuito impresso (PCB) contendo o módulo ESP32 LoRa com *display* integrado, utilizado para aquisição e transmissão de dados em tempo real.

A estrutura conta com três conectores inferiores do tipo pneumáticos de engate rápido, o que possibilita a instalação rápida e segura da central. Esses engates são conectados à PCB por mangueiras pneumáticas azuis, as quais possibilitam a entrada de sinais provenientes de sensores analógicos. A antena externa TX915-JKS-IPX20 está instalada no topo da caixa, garantindo a comunicação eficiente via protocolo LoRa.

Figura 22 – Central de alarmes módulo receptor



Fonte: Elaborada pelo autor, 2024.

A Figura 22 apresenta a montagem interna do módulo receptor do sistema LoRa, desenvolvido para integrar o sistema de comunicação sem fio. O dispositivo utiliza uma antena externa modelo TX915-JKS-IPX20, instalada na parte superior da caixa hermética, garantindo eficiência na recepção de dados via protocolo LoRa.

Adicionalmente, o módulo conta com a integração de um *buzzer*, implementado como um sistema de alerta sonoro local, capaz de sinalizar eventos de forma eficiente. A alimentação do dispositivo é fornecida por meio de um cabo USB conectado à placa do módulo ESP32, que, por sua vez, é alimentada pela saída USB de um Raspberry Pi.

Figura 23 – Central de alarmes com módulos transmissor e receptor



Fonte: Elaborada pelo autor, 2024.

A Figura 23 ilustra a central de alarmes com a integração de *hardware* finalizada. Os módulos já estão prontos para receber o *firmware* e iniciar as operações. O módulo transmissor será instalado no posto de enfermagem do pronto socorro, enquanto o módulo receptor ficará alocado no setor de manutenção do Hospital Regional, garantindo uma comunicação eficiente entre esses dois pontos estratégicos.

### 3.8 Programação ESP32 LoRa

A implementação de um sistema com ESP32 LoRa envolve várias etapas essenciais, desde a programação do dispositivo para ler dados analógicos de um sensor até a transmissão desses dados via protocolo de comunicação LoRa. A seguir, detalhamos os passos técnicos necessários para configurar tanto o módulo transmissor quanto o receptor.

Para programar as placas, utilizamos a IDE do Arduino devido à sua compatibilidade com diversas bibliotecas especializadas. Essas bibliotecas oferecem APIs robustas que simplificam a implementação de protocolos de comunicação, permitindo a configuração de parâmetros como frequência de operação, potência de transmissão e recepção de dados. Além disso, facilitam

o gerenciamento eficiente das funções de hardware das placas, como ADCs e GPIOs. Essa abordagem assegura uma integração confiável entre os dispositivos.

A biblioteca utilizada neste projeto foi a Heltec ESP32 LoRa V3 juntamente com a Radio Lib. O repositório da biblioteca Heltec ESP32 LoRa V3 fornece instruções detalhadas para a instalação e configuração da biblioteca, facilitando a programação e integração do ESP32 LoRa. Esta biblioteca permitiu a implementação eficiente da comunicação LoRa, oferecendo exemplos práticos de transmissão e recepção de dados. A documentação apresenta aspectos técnicos como a configuração de parâmetros de comunicação, gerenciamento de energia e otimização de desempenho.

O primeiro passo foi definir qual das placas atuaria como transmissor e qual seria o receptor. Essa decisão foi fundamental para a configuração dos parâmetros de comunicação LoRa e para a implementação da calibração do ADC. Após essa definição, estabelecemos os seguintes parâmetros técnicos para o módulo LoRa:

- a) Frequência de operação: `#define FREQUENCY 915,0` – Frequência de 915 MHz, adequada para a comunicação LoRa em conformidade com as regulamentações locais;
- b) Largura de banda: `#define BANDWIDTH 62,5` – Define a largura de banda para a comunicação, balanceando alcance e taxa de dados;
- c) Fator de espalhamento: `#define SPREADING_FACTOR 12` – Determina o fator de espalhamento, afetando a taxa de transmissão e a sensibilidade do sinal;
- d) Potência de transmissão: `#define TRANSMIT_POWER 20` – Configurada para 20 dBm, garantindo um alcance satisfatório sem exceder os limites permitidos;
- e) Taxa de codificação: `#define CODING_RATE 5` – Define a taxa de codificação para correção de erros durante a transmissão.

Adicionalmente, foi implementada a biblioteca para calibração do ADC, essencial para garantir a precisão das leituras analógicas. Como apenas uma das placas seria responsável pela transmissão dos dados, a calibração foi concentrada nessa unidade, utilizando curvas de linearização com funções de regressão polinomial conforme dito anteriormente.

Para assegurar maior confiabilidade e precisão nos dados obtidos pelos sensores conectados ao ADC do ESP32 LoRa, foi implementado um algoritmo de cálculo da média de 10.000 amostras consecutivas. Essa abordagem é essencial para reduzir o impacto de ruídos e flutuações no sinal analógico, proporcionando leituras mais estáveis. Calcular a média de várias amostras permite suavizar os valores obtidos, resultando em dados mais confiáveis para transmissão e processamento.

Como o ESP32 LoRa possui um *display* integrado, configuramos sua utilização por meio do protocolo I2C. Esse recurso foi utilizado para visualizar, em tempo real, as leituras

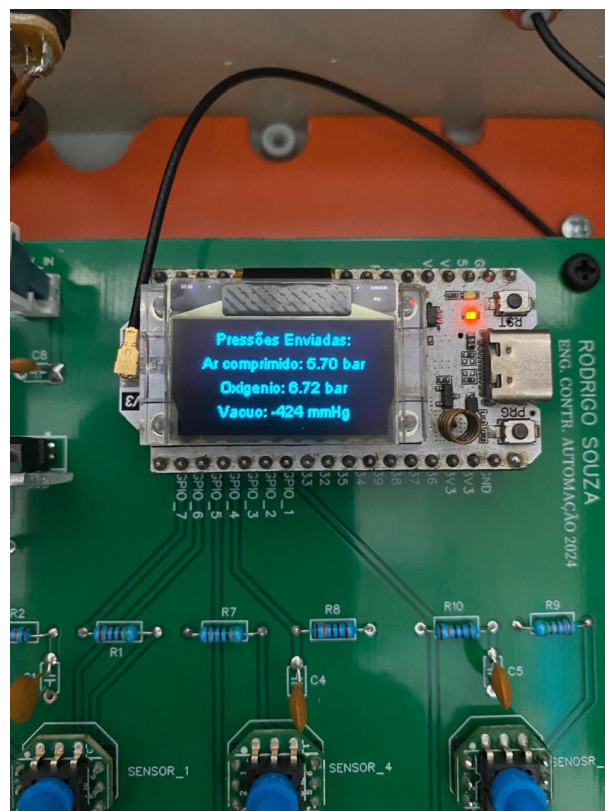
das pressões medidas pelo sistema. Essa funcionalidade tornou-se essencial para a validação em campo, permitindo verificar diretamente no local se as leituras realizadas pelo ESP32 eram consistentes e condiziam com as condições reais de operação.

Outro elemento implementado no código foi a função de indicação de transmissão de dados por meio de um LED. Esse recurso foi configurado para piscar durante o envio dos pacotes LoRa, confirmando visualmente que os pacotes contendo as pressões estavam sendo enviados.

Para prevenir travamentos no código, foi implementado o *Watchdog Timer* (WDT) no microcontrolador ESP32. Essa ferramenta monitora continuamente a execução do programa e, caso o sistema entre em um estado de loop infinito ou travamento inesperado, o WDT aciona automaticamente um reset no dispositivo, garantindo a restauração da operação normal.

O WDT opera como um temporizador que deve ser regularmente alimentado pelo software enquanto este funciona corretamente. Se o programa não reiniciar o WDT dentro de um intervalo de tempo predefinido, o temporizador interpreta isso como uma falha no sistema. Como medida corretiva, ele dispara uma ação de recuperação, que geralmente consiste em reiniciar completamente o dispositivo.

Figura 24 – Transmissor Esp32 LoRa



Fonte: Elaborada pelo autor, 2024.

A Figura 24 ilustra o módulo ESP32 LoRa em operação, com o código devidamente carregado. O dispositivo está realizando a aquisição de leituras de pressão em tempo real, com

os valores medidos sendo exibidos no *display* integrado. Simultaneamente, esses dados são transmitidos utilizando o protocolo de comunicação LoRa.

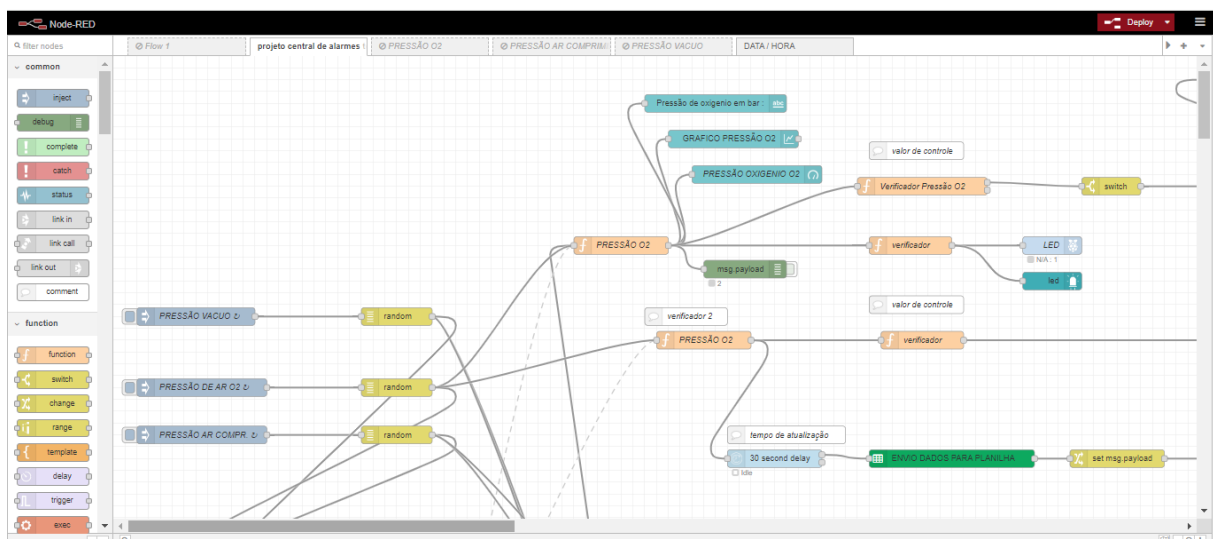
O pacote contendo as informações de pressão é enviado ao módulo ESP32 LoRa receptor, instalado na unidade de manutenção. Este módulo receptor foi configurado com os mesmos parâmetros operacionais do transmissor, garantindo a compatibilidade e a integridade na comunicação entre os dispositivos.

### 3.9 Node-RED

Node-RED é uma ferramenta de programação visual baseada em fluxo, desenvolvida inicialmente pela *International Business Machines Corporation (IBM)*, que se destaca pela sua capacidade de conectar dispositivos, APIs e serviços de forma rápida e eficiente. Utilizando uma abordagem de programação orientada a fluxo, Node-RED permite que desenvolvedores criem aplicações complexas de maneira intuitiva, sem a necessidade de escrever grandes blocos de código. Isso é feito através de uma interface gráfica que permite arrastar e soltar "nós" representando funções, dispositivos e APIs, e conectá-los para definir o fluxo de dados e ações do sistema.

A Figura 25 ilustra o ambiente de programação do Node-RED, destacando como é fácil e intuitivo criar novos fluxos. Através de uma interface gráfica simples, basta arrastar e soltar nós para construir o fluxo desejado. Essa abordagem visual facilita o desenvolvimento e a configuração de lógica complexa, mesmo para quem não possui experiência avançada em programação.

Figura 25 – Ambiente de programação no Node-Red



Fonte: Elaborada pelo autor, 2024.

O fluxo do sistema de monitoramento e alarme para pressão de oxigênio, ar comprimido e vácuo no Node-RED será realizado da seguinte maneira.

Os dados dos sensores de pressão serão recebidos por meio de um ESP32 LoRa, que atuará como um receptor. Este dispositivo utilizará uma conexão USB serial para transmitir suas leituras em tempo real ao Node-RED. Assim que os dados chegam, o Node-RED os captura imediatamente e os direciona para a etapa de processamento. Isso permite que as informações dos sensores sejam integradas de forma eficiente e em tempo real ao sistema, possibilitando um monitoramento contínuo e preciso.

Uma vez recebidos, os dados dos sensores passam por uma camada de processamento, onde são aplicadas lógicas específicas para validação e análise. O sistema verifica se as leituras dos sensores estão dentro dos limites pré-definidos como normais, avaliando se cada leitura se encontra entre os valores mínimos e máximos aceitáveis.

Se os valores estiverem dentro dos limites estabelecidos, o sistema simplesmente atualiza o *dashboard* com essas informações, permitindo o monitoramento em tempo real sem a necessidade de ações adicionais. No entanto, se alguma leitura ultrapassar os limites aceitáveis, o sistema imediatamente avança para a próxima etapa, que é a geração de alarmes. Isso garante que apenas condições críticas ou anormais desencadeiem alertas, evitando falsos positivos e mantendo a operação do sistema eficiente e focada em situações que realmente exigem atenção.

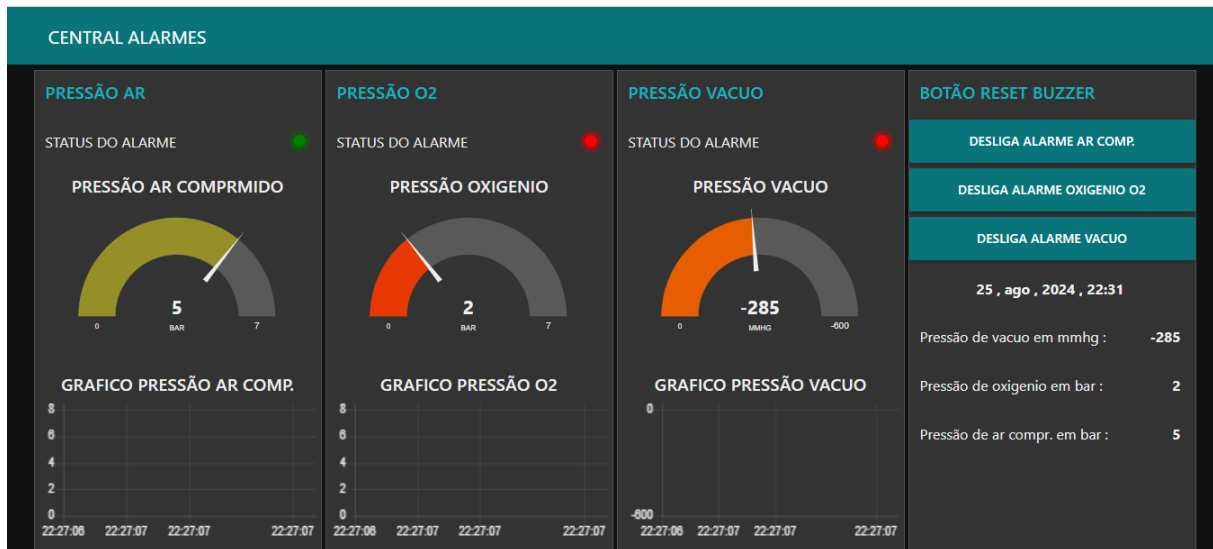
Se o processamento identificar condições anormais, como leituras fora dos limites seguros, o sistema imediatamente gera um alerta. Esses alertas podem ser configurados para acionar diversos mecanismos, tais como:

- a) Alarmes visuais e sonoros: Ativação de luzes de aviso ou sirenes para alertar os operadores no local;
- b) Notificações remotas: Envio de mensagens via SMS, Telegram ou WhatsApp para notificar a equipe de manutenção sobre a situação;
- c) Logs de evento: Para registrar eventos e condições anormais em um banco de dados para análises futuras, será utilizada a API do Google Sheets.

Os dados e alertas são integrados em um *dashboard* no Node-RED, que é constantemente atualizado em tempo real. Esta interface gráfica oferece uma visão completa das leituras atuais de pressão de oxigênio, ar comprimido e vácuo, utilizando indicadores visuais intuitivos para facilitar a compreensão do estado do sistema. Além disso, o *dashboard* disponibiliza gráficos históricos, que apresentam tendências e variações ao longo do tempo, auxiliando na análise de desempenho e na identificação de possíveis anomalias.

A Figura 26 apresenta um modelo de *dashboard* da central de alarmes após a conclusão do projeto. Ela demonstra como os dados e alertas são organizados e exibidos de maneira clara e funcional, proporcionando uma visão completa e acessível do sistema de monitoramento.

Figura 26 – Dashboards central alarmes



Fonte: Elaborada pelo autor, 2024.

Este fluxo operacional do Node-RED cria uma central de alarme proativa, que não apenas monitora as pressões dos sistemas em tempo real, mas também oferece ferramentas para análise, visualização e resposta imediata a condições críticas.

### 3.10 Raspberry Pi

O Raspberry Pi é um microcomputador de baixo custo, com dimensões aproximadas de um cartão de crédito, que proporciona grande versatilidade e desempenho em uma variedade de aplicações.

Para este projeto, será utilizado o modelo Raspberry Pi 4 Modelo B, que apresenta um processador quad-core de 64 bits de alto desempenho, suporte para dois monitores em resoluções de até 4K por meio de um par de portas micro HDMI, até 4 GB de RAM, LAN sem fio de banda dupla de 2,4/5,0 GHz, *Bluetooth* 5.0, *Gigabit Ethernet*, USB 3.0. Para o usuário final, o Raspberry Pi 4 Modelo B fornece desempenho de *desktop* comparável a sistemas de PC x86 de nível básico. Para maiores informações o Quadro 3 pode ser consultado (FOUNDATION, 2024).

No sistema de monitoramento e geração de alarmes, o Raspberry Pi atua como a unidade central de processamento e coleta de dados. Ele é responsável por receber e consolidar informações transmitidas pelos módulos ESP32 LoRa, que podem incluir parâmetros críticos como pressão de oxigênio, ar comprimido e vácuo. Além disso, o Raspberry Pi executa o Node-RED, uma plataforma de automação e integração que realiza o processamento e a gestão dos dados

Quadro 3 – Especificações do Raspberry Pi 4

<b>Especificações</b>	<b>Detalhes</b>
Processador	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) SoC de 64 bits a 1,8 GHz
Memória	4 GB LPDDR4-3200 SDRAM
Conectividade Sem Fio	2,4 GHz e 5,0 GHz IEEE 802.11ac, Bluetooth 5.0, BLE
Ethernet	Gigabit Ethernet
Portas USB	2 portas USB 3.0; 2 portas USB 2.0
GPIO	Cabeçalho GPIO padrão Raspberry Pi de 40 pinos (compatível com placas anteriores)
Saída de Vídeo	2 portas micro-HDMI (até 4kp60 suportado)
Portas de Display	Porta de exibição MIPI DSI de 2 pistas
Porta de Câmera	Porta de câmera MIPI CSI de 2 pistas
Áudio/ Vídeo	Porta de áudio estéreo e vídeo composto de 4 polos
Decodificação de Vídeo	H.265 (decodificação 4kp60), H264 (decodificação 1080p60, codificação 1080p30)
Gráficos	OpenGL ES 3.1, Vulkan 1.0
Armazenamento	Slot para cartão Micro-SD
Alimentação	5 V DC via conector USB-C (mínimo 3 A) 5 V DC via cabeçalho GPIO (mínimo 3 A) Power over Ethernet (PoE) habilitado (requer PoE HAT separado)
Temperatura de Operação	0 – 50 graus C ambiente

Fonte: Elaborada pelo autor, 2024.

localmente. Esse arranjo permite que o Raspberry Pi execute funções de análise e tratamento de dados em tempo real, otimizando a integração dos sensores e proporcionando uma resposta ágil às condições monitoradas.

Utilizar o Raspberry Pi como concentrador central de dados em um sistema de monitoramento e alarme traz diversas vantagens, como confiabilidade, capacidade de processamento, flexibilidade de conectividade e custo-benefício. Sua integração com os módulos ESP32 LoRa e o Node-RED cria um sistema capaz de coletar, processar e exibir dados críticos, proporcionando uma solução para monitoramento em tempo real e resposta a situações de alerta.

### 3.11 Implementação do Node-RED no Raspberry Pi

Para que o Raspberry Pi se torne o concentrador da central de alarmes, é necessário realizar a instalação do sistema operacional. Neste projeto, foi utilizado o Raspberry Pi OS devido à sua compatibilidade e suporte com o hardware utilizado. A preparação iniciou-se com o uso de um cartão SD, no qual a imagem do sistema operacional foi gravada utilizando a ferramenta *Raspberry Pi Imager*.

Após a instalação do sistema operacional no Raspberry Pi, foi adicionado o *software* necessário para viabilizar a configuração e operação da central de alarmes. Para esse projeto, conforme dito anteriormente, utilizamos o Node-RED, uma ferramenta de programação visual que permite a integração de sensores, atuadores e serviços. A instalação do Node-RED foi realizada localmente, configurando o ambiente de desenvolvimento para os fluxos específicos da aplicação.

Para organizar a programação no Node-RED e garantir a modularidade do sistema, foram implementados cinco fluxos principais, cada um destinado a uma funcionalidade específica da central de alarmes. A seguir, são descritos os fluxos desenvolvidos:

- a) Dashboard: Responsável pela interface de usuário, permitindo o monitoramento em tempo real das variáveis do sistema;
- b) Parâmetros: Gerencia as configurações gerais do sistema, como limites de alerta, tempo de envio de mensagem via Telegram e tempo de envio de dados para planilha no *Gsheet*;
- c) Pressão de ar comprimido: Monitora a pressão do sistema de ar comprimido, identificando desvios que possam comprometer a operação;
- d) Pressão de vácuo: Realiza o monitoramento da pressão de vácuo, garantindo a detecção de falhas ou condições anormais no sistema;
- e) Pressão de oxigênio: Supervisiona a pressão do sistema de oxigênio, assegurando que esteja dentro dos limites de segurança;
- f) Alarme buzzer: Gerencia a ativação de alertas sonoros, sendo acionado automaticamente em caso de valores críticos detectados nos fluxos de pressão.

Essa estrutura de fluxos facilita a manutenção e a expansão do sistema, promovendo uma organização clara e eficiente no ambiente de desenvolvimento do Node-RED.

Para viabilizar a alimentação do Node-RED com os dados recebidos através do protocolo LoRa, foi implementada uma conexão serial utilizando a interface USB do Raspberry Pi, conectada à porta USB tipo C do módulo ESP32 LoRa. Essa configuração permitiu a transferência dos dados de forma eficiente e em tempo real.

O protocolo *Universal Asynchronous Receiver-Transmitter* (UART) foi empregado para estabelecer a comunicação serial entre o ESP32 LoRa e o Raspberry Pi. Essa escolha garantiu a transmissão dos dados das medições realizadas em campo, como pressão de ar comprimido, vácuo e oxigênio, diretamente para o Raspberry Pi, possibilitando o processamento imediato dos valores no ambiente do Node-RED.

Para que os dados pudessem ser processados corretamente no Node-RED, foi utilizado o nó (`serial in`), responsável por receber as informações transmitidas pelo módulo LoRa. Em seguida, foi implementado um nó de função (`function`) cuja lógica identifica qual gás medicinal está associado aos dados recebidos na transmissão.

Essa estrutura foi projetada para garantir que mensagens fora do formato esperado sejam automaticamente descartadas, aguardando a próxima transmissão válida. Essa abordagem assegura que apenas dados consistentes e devidamente formatados, como as medições específicas de pressão de ar comprimido, vácuo e oxigênio, sejam processados. Dessa forma, o tratamento das informações ocorre de maneira individualizada e confiável dentro dos fluxos configurados no Node-RED.

Após a identificação da mensagem, os dados são exibidos no *dashboard*. No entanto, dentro do fluxo de parâmetros, o valor do gás medicinal é analisado. Caso o valor esteja fora do limite especificado, uma notificação é enviada imediatamente pelo Telegram, informando que o gás medicinal em questão apresenta baixa pressão. Além disso, um sinal sonoro é ativado na central, permanecendo ativo até que o problema de baixa pressão seja resolvido. Essa abordagem garante uma resposta rápida a condições críticas, assegurando maior segurança e confiabilidade no sistema.

Com o objetivo de registrar e monitorar o histórico do estado das pressões, foi utilizado o *Google Sheets* (*GSheet*) como base de dados. Esse recurso é alimentado automaticamente a cada 60 segundos com os valores das pressões monitoradas. O registro contínuo desses dados é altamente útil para análises de desempenho, como a identificação de possíveis perdas de eficiência em equipamentos, por exemplo, na bomba de vácuo.

A partir da análise dos dados históricos, é possível detectar tendências de queda de rendimento, permitindo a antecipação de intervenções corretivas, como a manutenção preventiva. Essa abordagem contribui significativamente para a redução de falhas inesperadas e para a maximização da confiabilidade operacional do sistema.

Figura 27 – Dashboard final central de alarmes



Fonte: Elaborada pelo autor, 2024.

A Figura 27 apresenta o *dashboard* da central de alarmes final que foi projetado para oferecer uma interface de monitoramento em tempo real, proporcionando ao operador uma visão clara e detalhada das condições dos sistemas de gases medicinais. A seguir, detalha-se a usabilidade das principais funcionalidades:

- Monitoramento em tempo real:** Cada variável monitorada, como pressão de ar comprimido, pressão de oxigênio e pressão de vácuo, é exibida em medidores circulares que mostram os valores atuais de forma destacada. Essas informações são acompanhadas por gráficos históricos, permitindo a visualização de tendências ao longo do tempo;
- Indicadores de status de alarme:** Acima de cada seção, há um indicador visual de status do alarme, representado por um ponto luminoso. A cor verde indica que os valores estão dentro dos limites esperados, enquanto a mudança para cor vermelha indica uma anomalia, ou uma baixa pressão;
- Gráficos históricos:** A interface inclui gráficos que exibem a variação das pressões ao longo do tempo, facilitando a análise de tendências e a detecção de irregularidades, como quedas de pressão;
- Controle do buzzer:** Um botão interativo, denominado “*desligar buzzer*”, permite que o operador silencie manualmente, por um intervalo de tempo, o alarme sonoro da central, caso uma condição crítica esteja controlada;
- Painel de dados detalhados:** À direita da interface, um painel exibe informações complementares, como os valores atuais das pressões monitoradas em suas respectivas unidades (*bar* e *mmHg*);

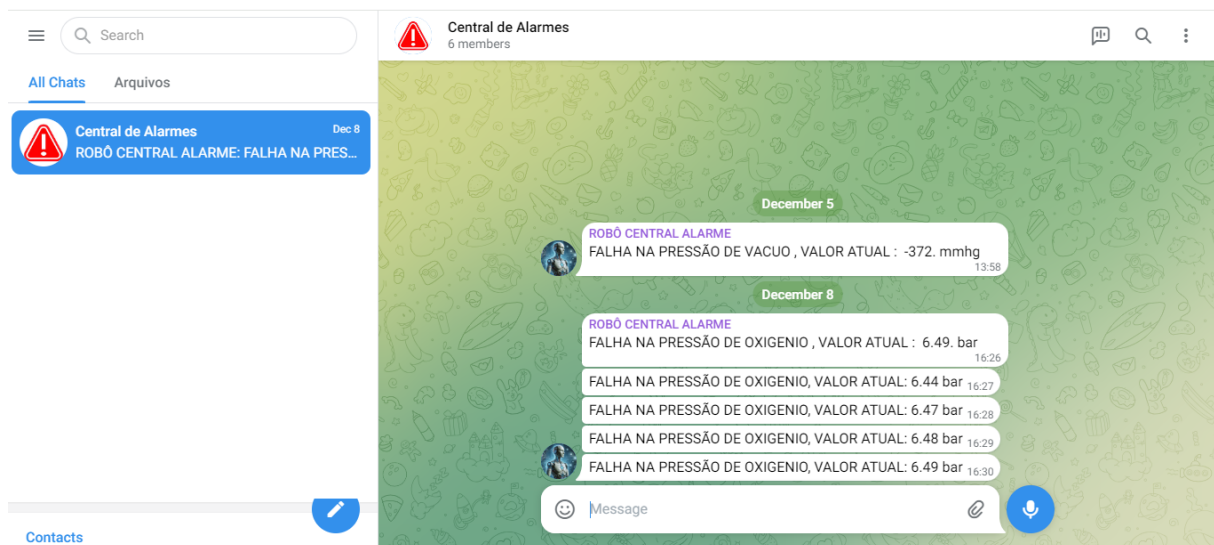
- f) Estrutura modular: A organização modular do *dashboard* assegura que as diferentes variáveis sejam exibidas de forma isolada, proporcionando maior clareza e facilitando a interpretação dos dados;
- g) Notificações automatizadas: Em situações onde os valores monitorados ultrapassam os limites estabelecidos, o *dashboard* envia notificações automáticas via *pop-ups* para alertar os operadores sobre a necessidade de intervenção.

A usabilidade do *dashboard* foi projetada com foco na simplicidade e eficiência operacional, permitindo ao usuário identificar rapidamente quaisquer desvios nos parâmetros monitorados.

Ao longo do período de avaliação, o uso do Node-RED instalado no Raspberry Pi mostrou-se uma opção eficaz para a central de alarmes, integrando os sensores e os serviços requeridos de maneira confiável. O sistema conseguiu monitorar em tempo real os parâmetros críticos, como pressão de ar comprimido, vácuo e oxigênio, apresentando os dados no *dashboard* de maneira clara e acessível.

Além disso, a integração com o módulo ESP32 LoRa, utilizando comunicação via UART, garantiu a transmissão dos dados de campo com baixa latência e alta precisão. A lógica implementada nos nós de função permitiu descartar mensagens inválidas e tratar as informações de maneira individualizada, assegurando que apenas os valores consistentes fossem processados e utilizados pelo sistema.

Figura 28 – Notificações recebidas via Telegram



Fonte: Elaborada pelo autor, 2024.

A Figura 28 ilustra o recebimento de uma notificação automática via Telegram, o que se mostrou essencial para alertar rapidamente sobre condições críticas, como a baixa pressão de gases medicinais. Além do mais, o alarme sonoro, configurado para permanecer ativo até a resolução do problema, reforçou ainda mais a capacidade de resposta imediata do sistema.

Figura 29 – Base dados central de alarmes

	A	B
1	HISTORICO PRESSÃO DE VACUO	
14407	PRESSÃO DE VACUO, VALOR ATUAL: -445 mmHg, 09-12-2024 15:02:53	
14408	PRESSÃO DE VACUO, VALOR ATUAL: -469 mmHg, 09-12-2024 15:03:53	
14409	PRESSÃO DE VACUO, VALOR ATUAL: -441 mmHg, 09-12-2024 15:04:53	
14410	PRESSÃO DE VACUO, VALOR ATUAL: -469 mmHg, 09-12-2024 15:05:53	
14411	PRESSÃO DE VACUO, VALOR ATUAL: -456 mmHg, 09-12-2024 15:06:53	
14412	PRESSÃO DE VACUO, VALOR ATUAL: -478 mmHg, 09-12-2024 15:07:53	
14413	PRESSÃO DE VACUO, VALOR ATUAL: -468 mmHg, 09-12-2024 15:08:53	
14414	PRESSÃO DE VACUO, VALOR ATUAL: -458 mmHg, 09-12-2024 15:09:53	
14415	PRESSÃO DE VACUO, VALOR ATUAL: -464 mmHg, 09-12-2024 15:10:53	
14416	PRESSÃO DE VACUO, VALOR ATUAL: -439 mmHg, 09-12-2024 15:11:53	
14417	PRESSÃO DE VACUO, VALOR ATUAL: -471 mmHg, 09-12-2024 15:12:53	
14418	PRESSÃO DE VACUO, VALOR ATUAL: -447 mmHg, 09-12-2024 15:13:53	
14419	PRESSÃO DE VACUO, VALOR ATUAL: -477 mmHg, 09-12-2024 15:14:53	
14420	PRESSÃO DE VACUO, VALOR ATUAL: -439 mmHg, 09-12-2024 15:15:53	
14421		
14422		
14423		
14424		

Fonte: Elaborada pelo autor, 2024.

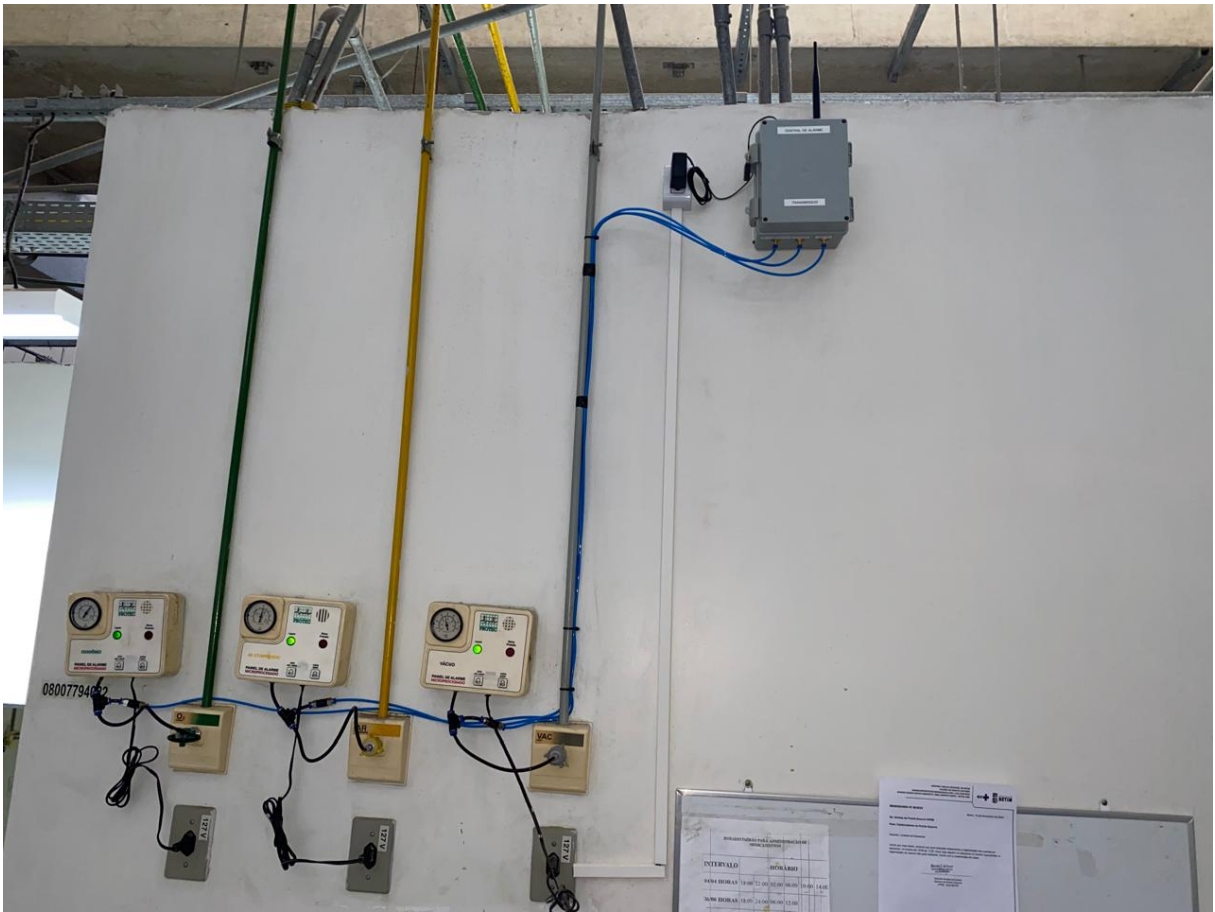
A Figura 29 apresenta o registro contínuo dos dados no *Google Sheets*, o que proporcionou uma base histórica, permitindo análises detalhadas sobre o desempenho dos equipamentos, como a detecção de possíveis quedas de eficiência na bomba de vácuo, ar comprimido e oxigênio. Com isso, é possível antecipar manutenções preventivas, reduzindo o risco de falhas inesperadas e aumentando a confiabilidade do sistema.

Esses resultados destacam a eficácia da solução desenvolvida, evidenciando sua capacidade de atender aos requisitos de monitoramento em aplicações críticas.

### 3.12 Instalação dos módulos transmissor e receptor

A instalação do módulo transmissor da central de alarmes foi realizada no posto de enfermagem do pronto socorro. A escolha desse local deve-se à facilidade de acesso e à disponibilidade de pontos para a conexão adequada da central de alarmes. O posto de enfermagem já contava com os pontos de conexão necessários para integrar a central de alarmes ao sistema existente, incluindo conexões elétricas e pneumáticas. Essa infraestrutura pré-existente minimizou a necessidade de adaptações adicionais, reduzindo os custos e o tempo de instalação.

Figura 30 – Posto de enfermagem módulo transmissor



Fonte: Elaborada pelo autor, 2024.

A Figura 30 mostra a instalação já realizada do módulo transmissor no posto de enfermagem do pronto socorro.

Figura 31 – Manutenção módulo receptor



Fonte: Elaborada pelo autor, 2024.

A Figura 31 ilustra a instalação realizada no setor de manutenção do Hospital Público de Betim. A escolha desse local foi estratégica, pois permite a visualização em tempo real das pressões dos gases medicinais, garantindo uma resposta imediata em caso de disparo de alarmes. Além disso, o alerta sonoro emitido pela central de alarmes pode ser facilmente ouvido por qualquer funcionário presente no ambiente, assegurando que a equipe seja rapidamente notificada sobre eventuais irregularidades no sistema. Outro benefício é que o *dashboard* do sistema pode ser acessado remotamente via endereço IP, proporcionando maior flexibilidade no monitoramento e gerenciamento das informações.

Figura 32 – Alcance da comunicação com ESP32 LoRa



Fonte: Elaborada pelo autor, 2024.

A Figura 32 apresenta a posição final de instalação do sistema, com uma distância aproximada de 46 metros entre o módulo transmissor ESP32 LoRa e o módulo receptor ESP32 LoRa. Durante os testes, não foram observadas perdas de pacotes nas transmissões, o que atende às expectativas, uma vez que nos testes iniciais a comunicação entre os módulos alcançou até 150 metros.

Inicialmente, o sistema estava configurado para enviar um pacote a cada 15 segundos contendo os valores de todos os gases medicinais de forma concatenada. Por exemplo:

```
ar_comprimido: 7,26 bar, oxigênio: 6,7 bar, vácuo: -450 mmHg.
```

Apesar da resistência da comunicação LoRa a interferências, ocasionalmente as mensagens chegavam ao receptor com caracteres corrompidos, prejudicando a análise dos dados na programação realizada no Node-RED.

Para solucionar o problema, o método de transmissão foi ajustado. Agora, a cada 9 segundos, é enviado o valor de apenas um gás medicinal por vez. Essa abordagem trouxe diversas melhorias:

- a) Otimização da largura de banda – A transmissão de menos caracteres reduz o volume de dados enviados, garantindo um uso mais eficiente da largura de banda disponível;

- b) Redução do tempo de transmissão – Mensagens mais curtas minimizam o período em que os dados ficam "no ar", diminuindo o risco de colisões e interferências com outros dispositivos;
- c) Maior eficiência na correção de erros – Com um volume menor de dados, os mecanismos de detecção e correção tornam-se mais eficazes, aumentando a confiabilidade da comunicação.

Essas melhorias tornaram o sistema mais estável, confiável e adequado ao ambiente, garantindo a transmissão precisa dos dados essenciais. A implementação de alarmes preventivos reforça a segurança operacional, pois qualquer variação nos parâmetros críticos, como ar comprimido a 6,5 bar, oxigênio a 6,5 bar e vácuo a -360 mmHg, dispara alertas imediatos. Isso permite uma resposta ágil da equipe técnica para corrigir falhas antes que comprometam o funcionamento dos equipamentos hospitalares, garantindo a continuidade das operações.

Não foi possível identificar com precisão a fonte da interferência no ambiente. No entanto, a instalação da central ocorreu próxima a equipamentos como aparelhos de raio-X e tomógrafos, que emitem sinais eletromagnéticos e podem estar contribuindo para o problema. Diante dessa situação, optou-se por ajustar os parâmetros de comunicação LoRa como alternativa para reduzir os efeitos da interferência e garantir a confiabilidade da transmissão de dados, tornando o sistema mais robusto em ambientes com altos níveis de ruído eletromagnético.

Para garantir a segurança e a integridade dos dados transmitidos, foi implementada uma rede privada de comunicação. Essa rede utiliza uma palavra de sincronização (*sync word*), que atua como um identificador exclusivo, garantindo que apenas dispositivos autorizados consigam participar da comunicação.

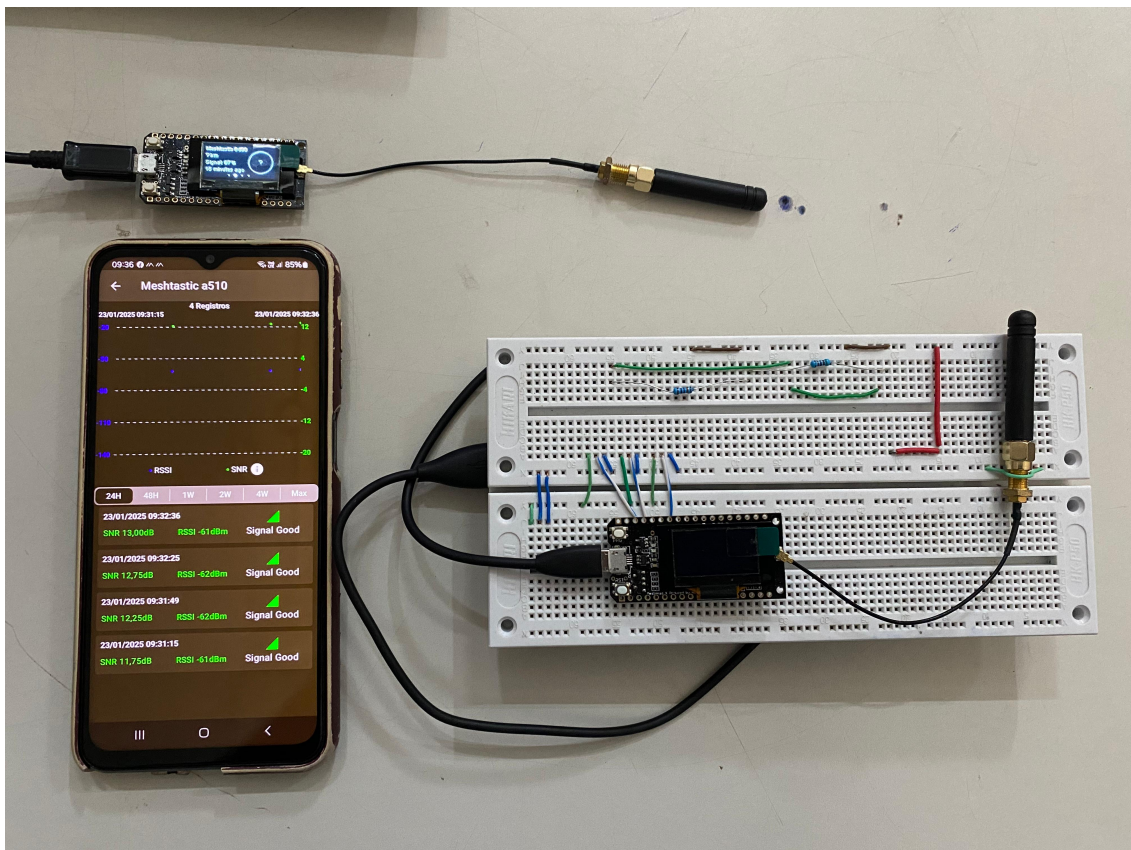
A palavra de sincronização é essencial no protocolo LoRa, pois permite que os dispositivos filtrem pacotes de dados, ignorando transmissões que não correspondam à configuração da rede privada. Dessa forma, evita-se que dispositivos não autorizados interceptem ou interfiram na troca de informações, aumentando a confiabilidade e a proteção contra acessos indevidos.

Além disso, essa abordagem contribui para a redução de interferências externas e melhora a estabilidade da comunicação, tornando o sistema mais seguro e eficiente, especialmente em ambientes com alta densidade de dispositivos sem fio.

## 4 RESULTADOS

Este capítulo descreve os materiais e softwares empregados no desenvolvimento da Central de Alarmes com ESP32 LoRa e fornece um quadro detalhado dos componentes físicos e softwares utilizados no sistema.

Figura 33 – Esp 32 LoRa transmissor e receptor



Fonte: Elaborada pelo autor, 2024.

A Figura 33 apresenta o ESP32 LoRa atuando como transmissor e receptor. A partir dos dados obtidos nos testes de transmissão, foi possível confirmar que o ESP32 LoRa é capaz de enviar e receber dados dentro do Hospital Público de Betim.

### 4.1 Materiais usados no projeto

O Quadro 4 detalha a lista de materiais utilizados no desenvolvimento do projeto da Central de Alarmes, apresentando de forma clara e organizada os itens necessários para sua execução.

Quadro 4 – Lista de materiais utilizados no projeto central de alarmes

ITEM	PRODUTO	QUANT.	VALOR UNIT.	VAL TOTAL
1	Adaptador Micro HDMI	1	R\$ 29,45	R\$ 29,45
2	Antena LoRa 915 MHz	2	R\$ 24,03	R\$ 48,06
3	Buzzer	1	R\$ 1,27	R\$ 1,27
4	Cabo Conversor HDMI p/ VGA	1	R\$ 32,62	R\$ 32,62
5	Cabo tipo C	1	R\$ 25,61	R\$ 25,61
6	Caixa Plástica PVC 160x120	1	R\$ 50,50	R\$ 50,50
7	Caixa Plástica PVC 190x150	1	R\$ 70,10	R\$ 70,10
8	Capacitor 100nf cerâmico	7	R\$ 0,25	R\$ 1,75
9	Capacitor 33nf cerâmico	1	R\$ 0,27	R\$ 0,27
10	Cartão micro SD	1	R\$ 29,74	R\$ 29,74
11	Case Raspberry Pi 4	1	R\$ 19,28	R\$ 19,28
12	Conector KK 3961-2 Fêmea 3,96mm	1	R\$ 1,20	R\$ 1,20
13	ESP32 LoRa	2	R\$ 132,73	R\$ 265,46
14	Espaçadores M3	8	R\$ 2,40	R\$ 19,20
15	Fonte AC/DC 12,8V	1	R\$ 35,02	R\$ 35,02
16	Fonte tipo C Raspberry Pi	1	R\$ 58,54	R\$ 58,54
17	Jack P4	1	R\$ 3,90	R\$ 3,90
18	LM7805	1	R\$ 2,07	R\$ 2,07
19	Placa PCB	5	R\$ 33,85	R\$ 169,25
20	Placa Perfurada 9x7	1	R\$ 1,70	R\$ 1,70
21	Raspberry Pi 4 4GB RAM	1	R\$ 575,00	R\$ 575,00
22	Resistor 100k ohms	6	R\$ 0,30	R\$ 1,80
23	Resistor 390 ohms	1	R\$ 0,13	R\$ 0,13
24	Resistor 51k ohms	6	R\$ 0,30	R\$ 1,80
25	Sensor de Pressão XGZP6847A	3	R\$ 59,99	R\$ 179,97
26	Transistor BC337	1	R\$ 0,36	R\$ 0,36
27	Tubo Acoplamento PM4	3	R\$ 3,37	R\$ 10,11
<b>TOTAL</b>				<b>R\$ 1.634,16</b>

Fonte: Elaborada pelo autor, 2024.

## 4.2 Lista de *software*

A lista de *softwares* utilizados no projeto abrange ferramentas essenciais para programação, processamento de dados e monitoramento do sistema. Cada *software* foi selecionado para desempenhar uma função específica, contribuindo desde o desenvolvimento do *firmware* até a análise e visualização dos dados coletados. A seguir, são detalhados os principais softwares empregados e suas respectivas aplicações no projeto.

Quadro 5 – Softwares utilizados no projeto

<b>Software</b>	<b>Utilização no Projeto</b>
MATLAB	Utilizado para processamento de dados de forma gráfica.
IDE Arduino	Empregado para a programação do ESP32 LoRa, desenvolvendo o <i>firmware</i> para transmissão e recepção dos dados.
Excel	Utilizado para extrair funções de calibração.
MeshStatic	Utilizado para simulação da rede LoRa, permitindo a avaliação da conectividade e cobertura dentro do ambiente hospitalar.
Node-RED	Utilizado para a criação de uma interface gráfica de monitoramento, permitindo a visualização e gerenciamento dos alarmes em tempo real.
Tina	Utilizado para simulação de circuitos eletrônicos.

Fonte: Elaborada pelo autor, 2025.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

O desenvolvimento deste sistema de central de alarmes utilizando ESP32 LoRa e Raspberry Pi demonstrou ser uma solução eficiente e responsiva para o monitoramento de variáveis críticas em ambientes hospitalares. A integração dessas tecnologias permitiu criar um sistema de baixo consumo de energia, com capacidade para transmitir dados sem a necessidade de cabos de comunicação, características ideais para atender às necessidades de segurança e conectividade em ambientes tão sensíveis.

O uso do ESP32 com tecnologia LoRa destacou-se como um diferencial, assegurando comunicação contínua e confiável mesmo em ambientes com possíveis interferências. Já o Raspberry Pi desempenhou um papel crucial ao oferecer uma interface gráfica intuitiva, permitindo a visualização clara e eficiente dos parâmetros dos gases medicinais, facilitando o monitoramento e a tomada de decisões.

Além da eficiência técnica, o projeto contribuiu significativamente para a segurança hospitalar ao minimizar os riscos associados a possíveis falhas no monitoramento dos gases medicinais. Sua implementação destaca a relevância de adotar tecnologias modernas e acessíveis em ambientes que exigem altos níveis de confiabilidade e precisão.

Com um custo total de R\$ 1.634,16, o projeto demonstrou uma excelente relação custo-benefício, comprovando sua viabilidade para implementação. Essa acessibilidade econômica revela que é possível desenvolver soluções tecnológicas de grande impacto com recursos limitados, mantendo a eficiência e a confiabilidade. Ademais, o baixo custo torna essa proposta particularmente atraente para instituições que procuram inovar e otimizar processos essenciais, como o monitoramento hospitalar, sem ultrapassar seus orçamentos.

Em resumo, o projeto atingiu seus objetivos ao desenvolver uma solução tecnológica inovadora, acessível e financeiramente viável, com impacto direto e positivo na segurança hospitalar. A proposta destacou a relevância da automação como uma ferramenta indispensável para a modernização e otimização de sistemas críticos em áreas de alta responsabilidade, como o setor de saúde.

### 5.1 Trabalhos futuros

Como possibilidades de aprimoramento, sugere-se a inclusão de novas variáveis de controle, como o monitoramento dos níveis dos reservatórios inferior e superior de água, além da temperatura da sala do tomógrafo. O acompanhamento da climatização é crucial, pois o tomógrafo não opera adequadamente em temperaturas elevadas. A coleta de dados sobre a temperatura da sala permitiria identificar possíveis falhas no sistema de ar-condicionado e agir preventivamente, evitando interrupções no funcionamento do equipamento.

Já o monitoramento dos níveis dos reservatórios de água é igualmente importante, con-

siderando que o uso de água é essencial para diversas atividades hospitalares. A falta de água representa um problema significativo, podendo comprometer o atendimento e aumentar os riscos operacionais. Esses avanços não apenas reforçariam a confiabilidade do sistema, mas também ampliariam sua abrangência, tornando-o ainda mais eficaz em futuras versões.

## REFERÊNCIAS

- AMARAL, A. E. S. **Desenvolvimento de um sistema supervisorio para monitoramento contínuo de sinais**. 84 p. Dissertação (Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação)) — Universidade Federal do Rio Grande do Norte, Natal, 2025. Disponível em: <<https://repositorio.ufrn.br/handle/123456789/62130>>. Acesso em: 25 jan. 2025.
- ANATEL. **Resolução nº 680 da ANATEL**. 2017. Disponível em: <<https://informacoes.anatel.gov.br/legislacao/resolucoes/2017/936-resolucao-680>>. Acesso em: 14 mai. 2024.
- (ANVISA), A. N. de V. S. **Gases medicinais: Informações gerais**. Brasília, DF: Agência Nacional de Vigilância Sanitária (ANVISA), 2020. Disponível em: <<https://www.gov.br/anvisa/pt-br/setorregulado/regularizacao/medicamentos/gases-medicinais/informacoes-gerais#:~:text=1.,ou%20modifica%C3%A7%C3%A3o%20de%20fun%C3%A7%C3%B5es%20fisiol%C3%B3gicas.>> Acesso em: 29 mar. 2024.
- AUGUSTIN, A.; YI, J.; CLAUSEN, T.; TOWNSLEY, W. M. **A Study of LoRa: Long Range & Low Power Networks for the Internet of Things**. 2016. 1466 p. Disponível em: <<<https://www.mdpi.com/1424-8220/16/9/1466>>>. Acesso em: 23 set. 2024.
- AUTOMATION, H. **WiFi LoRa 32 (V3)**. 2025. Disponível em: <<https://heltec.org/project/wifi-lora-32-v3/>>. Acesso em: 03 fev. 2025.
- BRIDI, A. C.; SILVA, R. C. L. d.; FARIAS, C. C. P. d.; FRANCO, A. S.; SANTOS, V. d. L. Q. d. Tempo estímulo-resposta da equipe de saúde aos alarmes de monitorização na terapia intensiva: implicações para a segurança do paciente grave. **Revista Brasileira de Terapia Intensiva**, SciELO Brasil, v. 26, p. 28–35, 2014.
- CFsensor. **Air Pressure Sensor XGZP6847A**. 2024. Disponível em: <<https://cfsensor.com/product/air-pressure-sensor-xgzp6847a/>>. Acesso em: 13 mai. 2024.
- COMER, D. E. **Interligação de Redes com TCP/IP: Princípios, Protocolos e Arquitetura**. 6ª ed.. ed. São Paulo, SP: Elsevier Brasil, 2016. v. 1. 36–56 p.
- CORPORATION, S. **LoRa and LoRaWAN: A Technical Overview**. 2024. Disponível em: <<https://www.semtech.com/uploads/technology/LoRa/lora-and-lorawan.pdf>>. Acesso em: 13 dez. 2024.
- EBYTE. **Datasheet LoRa Module**. 2023. Disponível em: <<https://www.es-ebyte.com/Uploadfiles/Files/2023-4-3/202343113916132.pdf>>. Acesso em: 03 fev. 2025.
- ESPRESSIF. **ESP-IDF ADC API Reference (ESP32-S3)**. 2025. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32s3/api-reference/peripherals/adc.html>>. Acesso em: 03 fev. 2025.
- FOUNDATION, R. P. **Raspberry Pi 4 Model B Specifications**. 2024. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>>. Acesso em: 15 set. 2024.
- GÁS, O. **Sinalização de Tubulação para Gases Medicinais: Oxigênio, Ar Comprimido e Vácuo**. 2023. Disponível em: <<https://www.oxigasinstalacao.com.br/produto/sinalizacao-tubulacao-gases-medicinais-oxigenio-ar-comprimido-e-vacuoo/>>. Acesso em: 10 out. 2024.

INSTRUMENTS, T. **Application Note A Basic Guide to I2C**. 2013. Texas Instruments Application Report. Disponível em: <<https://www.ti.com/lit/pdf/sbaa565>>. Acesso em: 14 mai. 2024.

LEENS, F. An introduction to i2c and spi protocols. **IEEE Instrumentation & Measurement Magazine**, v. 12, n. 1, p. 8–13, 2009.

LOPEZ, M. P.; ABREU, F. R. Central de gases medicinais: coração de uma instituição hospitalar. **Caderno de Estudos Tecnológicos**, v. 1, n. 1, 2013.

MAGRANI, E. **A internet das coisas**. [S.l.]: Edições Cândido, 2022. Disponível em: <<https://repositorio.fgv.br/server/api/core/bitstreams/b50af2ba-b001-4b1d-a1ad-5df985f6d1bb/content>>. Acesso em: 10 jun. 2024.

OLIVEIRA, A. E. C. d.; MACHADO, A. B.; SANTOS, E. D. d.; ALMEIDA, É. B. d. Fadiga de alarmes e as implicações para segurança do paciente. **Revista Brasileira de Enfermagem**, SciELO Brasil, v. 71, p. 3035–3040, 2018.

PROJETADO, M. **O Protocolo I<sup>2</sup>C**. 2024. Disponível em: <<https://mundoprojetado.com.br/i2c/>>. Acesso em: 6 mai. 2024.

PROTOTYPE, F. **AN1200.22 - LoRaWAN™ Specification**. 2016. Disponível em: <<https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>>. Acesso em: 19 abr. 2024.

SILVA, R. C. L. d. Fadiga de alarmes: quando os alarmes se tornam uma ameaça à segurança nas unidades de cuidados intensivos. **Revista de Enfermagem UFPE on line**, Recife, v. 8, n. 1, 2014. Disponível em: <<https://periodicos.ufpe.br/revistas/index.php/revistaenfermagem/article/view/9597>>. Acesso em: 17 Mai. 2024.

SYSTEMS, E. **ESP-IDF ADC Calibration Documentation for ESP32-S3**. 2023. Disponível em: <[https://docs.espressif.com/projects/esp-idf/en/v5.3.1/esp32s3/api-reference/peripherals/adc\\_calibration.html](https://docs.espressif.com/projects/esp-idf/en/v5.3.1/esp32s3/api-reference/peripherals/adc_calibration.html)>. Acesso em: 11 set. 2024.

## **APÊNDICE A – MANUAL TÉCNICO DE OPERAÇÃO E MANUTENÇÃO DO PROTÓTIPO DA CENTRAL DE ALARMES**

Este manual tem como objetivo fornecer instruções detalhadas para a operação e manutenção do protótipo da Central de Alarmes, desenvolvido para monitorar sistemas críticos, como ar comprimido, oxigênio e vácuo.

O documento está estruturado para guiar usuários e técnicos em todas as etapas necessárias para o correto funcionamento da central, abrangendo os seguintes tópicos:

- a) Descrição do sistema: Explicação das funcionalidades, componentes e especificações técnicas do protótipo;
- b) Instalação e configuração: Procedimentos para instalação da Central Transmissora e Receptora, garantindo conexão segura com os sensores e a fonte de energia;
- c) Operação: Instruções para uso diário, monitoramento em tempo real e interpretação dos alarmes visuais e sonoros;
- d) Manutenção preventiva e corretiva: Recomendações para inspeção periódica, testes de alarmes e solução de possíveis falhas, incluindo a substituição de sensores e conexões defeituosas;
- e) Segurança: Diretrizes para armazenamento, reinicialização do sistema e cuidados gerais para prolongar a vida útil do equipamento;
- f) Parâmetros de vigilância: Valores ideais de operação para cada gás monitorado, garantindo que qualquer desvio seja identificado imediatamente;
- g) Resultados esperados: Monitoramento contínuo e confiável, com alarmes precisos e resposta eficaz a condições críticas.

Este manual serve como um guia para garantir a operação eficiente e segura da Central de Alarmes.

---

# Manual Técnico de Operação e Manutenção do Protótipo da Central de Alarmes

## 1. Introdução

Este manual detalha os procedimentos para instalação, teste, uso e manutenção do protótipo da central de alarmes, utilizando ESP32 LoRa e Raspberry Pi 4 com objetivo de monitorar sistemas críticos (ar comprimido, oxigênio e vácuo).

---

## 2. Instruções de Instalação

### 2.1. Requisitos de Instalação

- **Hardware:** A Central de Alarmes Transmissora deve ser instalada em um local próximo a uma fonte de energia elétrica e aos pontos de conexão de ar comprimido, oxigênio e vácuo, garantindo facilidade de acesso e minimizando a extensão das conexões. A Central de Alarmes Receptora deve ser posicionada em um local seguro, próximo a uma fonte de energia elétrica, para assegurar o funcionamento contínuo e estável do sistema.

### 2.2. Passos para Instalação

1. **Configuração Central Alarmes Receptora:**
    - O equipamento já vem previamente configurado com os valores de controles ideais para Hospital Regional
    - Caso seja necessário alterar algum parâmetro, poderá ser feita na aba lateral (CONFIGURAÇÕES).
  2. **Conexão da Central Transmissora:**
    - A Central é projetada com conectores de engate rápido, permitindo uma conexão fácil e segura. Basta conectar a mangueira pneumática ao ponto correspondente ao gás medicinal que será monitorado.
  3. **Comunicação:**
    - Assim que o equipamento for devidamente instalado e conectado à fonte de energia, que pode ser de 127V ou 220V, ele iniciará a comunicação de forma automática.
- 

## 3. Utilização do Protótipo

### 3.1. Operação Diária

- O protótipo realiza monitoramento em tempo real e aciona alarmes visuais e sonoros no caso de desvios nos parâmetros.
- Utilize o monitor conectado ao Raspberry Pi para visualizar os dados.

### 3.2. Interpretação de Alarmes

- **Alarme Normal:** Todos os sistemas operam dentro dos limites configurados.
  - **Alarme Crítico:** Valores fora dos limites operacionais seguros; requer intervenção imediata.
- 

## 4. Checagem e Manutenção

### 4.1. Manutenção Preventiva

- **Periodicidade:** Realizar a cada 15 dias.
- **Atividades:**
  - Verificar conexões dos sensores e integridade dos cabos.
  - Testar a resposta dos alarmes com simulações de valores críticos.

### 4.2. Manutenção Corretiva

- Substitua sensores ou conexões que apresentarem falhas.
  - Reinicie Central Receptora retirando da tomada e ligando na tomada em caso de travamentos.
- 

## 5. Funcionamento e Guarda

### 5.1. Armazenamento do Protótipo

- Guarde Central em local seco e protegido de impactos quando não estiverem em uso.
- Utilize uma caixa organizadora para evitar danos aos componentes.

### 5.2. Reinicialização

- Reinicie o sistema mensalmente para evitar falhas acumuladas.
- 

## 6. Parâmetros de Vigilância e Alarmes

- **Ar Comprimido:** 6.5 bar.
- **Oxigênio:** 6.5 bar.
- **Vácuo:** -360mmHg.

Os parâmetros devem ser revisados de acordo com as necessidades específicas do ambiente monitorado.

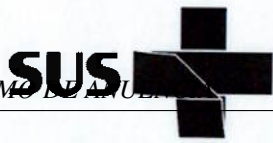
## 7. Resultados Esperados e Avaliação

- **Resultados:**
    - Monitoramento contínuo, confiável e com alarmes precisos.
    - Visualização em tempo real das pressões dos gases medicinais.
  - **Avaliação:**
    - Testar o protótipo simulando condições críticas para garantir resposta correta.
    - Revisar os alarmes gerados em relação aos eventos reais.
- 

Este manual garante que o protótipo seja utilizado de forma segura e eficiente, com foco na simplicidade e na funcionalidade do sistema.

## **APÊNDICE B – TERMO DE ANUÊNCIA**

O Termo de Anuência fornecido formaliza a autorização para a realização da pesquisa intitulada "Implementação de uma central de alarme hospitalar com ESP32 LoRa e Raspberry Pi". Este documento é essencial, pois garante o consentimento oficial da instituição para a execução do projeto em suas dependências.



Betim, 16 de dezembro de 2024.

### TERMO DE ANUÊNCIA


Em resposta à solicitação de aprovação para realização de pesquisa pelo pesquisador Rodrigo de Souza Gonçalves, responsável pelo projeto: **“Implementação de uma central de alarme hospitalar com ESP32 LORA E RASPBERRY P1”**, que tem como objetivo: projetar e implementar uma central de alarme hospitalar para monitorar os gases medicinais que seja capaz de trazer maior segurança para o ambiente hospitalar, registramos que:

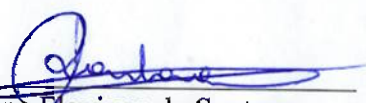
- 1- o projeto de pesquisa apresentado é coerente e apresenta método adequado;
- 2- o fomento à pesquisa e ao desenvolvimento técnico e científico, bem como a integração ensino e serviço, fazem parte das diretrizes da Educação em Saúde do SUS/Betim;
- 3- a execução do projeto foi autorizada pela Coordenação do Núcleo de Ensino e Pesquisa (NEP) do HPRB;
- 4- o SUS Betim não terá nenhuma despesa decorrente da participação nessa pesquisa;
- 5- os resultados desta pesquisa deverão ser apresentados à Secretaria Municipal de Saúde de Betim.

Diante de tais considerações, autorizamos a execução desse trabalho. Ressaltamos a necessidade de que o autor deixe explícito, em toda e qualquer publicação feita a partir desta pesquisa, a participação do SUS/Betim.

À disposição para quaisquer esclarecimentos.

Atenciosamente,

  
Daniela Cristina da Silveira  
Diretora de Gestão do Trabalho e Educação em Saúde – SMS/SUS Betim

  
Jaqueline Flaviana de Santana  
Secretária Municipal de Saúde  
SMS/SUS Betim