



**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA – EST**

MÁRCIA DOS SANTOS DE SA

**DESENVOLVIMENTO DE UM PROTÓTIPO DE DISPOSITIVO DE
BAIXO CUSTO BASEADO EM IOT (*INTERNET OF THINGS*) PARA
DETECÇÃO E PREVENÇÃO DE INCÊNDIOS EM AMBIENTES
RESIDENCIAIS**

Orientador: Fábio de Sousa Cardoso, Dr.

Manaus

2022

MÁRCIA DOS SANTOS DE SÁ

**DESENVOLVIMENTO DE UM PROTÓTIPO DE DISPOSITIVO DE
BAIXO CUSTO BASEADO EM IOT (*INTERNET OF THINGS*) PARA
DETECÇÃO E PREVENÇÃO DE INCÊNDIOS EM AMBIENTES
RESIDENCIAIS**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro Eletrônico(a).

Orientador: Fábio de Sousa Cardoso, Dr.

Manaus

2022

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia – EST

Reitor:

Cleinaldo de Almeida Costa

Vice-Reitor:

Cleto Cavalcante de Souza Leal

Diretor da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Eletrônica:

Bruno da Gama Monteiro

Banca Avaliadora composta por: Data da defesa: 25/01/2022.

Prof. Fábio de Sousa Cardoso, Dr. (Orientador)

Prof. Bruno da Gama Monteiro, Me.

Prof. Jozias Parente de Oliveira, Dr.

CIP – Catalogação na Publicação

Sá, Márcia de

Desenvolvimento de um protótipo de dispositivo de baixo custo baseado em Iot (*internet of things*) para detecção e prevenção de incêndios em ambientes residenciais / Márcia dos Santos de Sá; [orientado por] Fábio de Sousa Cardoso – Manaus: 2022 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2021.

1.IoT. 2.Detecção de incêndio, 3. protocolo UDP, 4. HTTP, 5. Sistema embarcados, 6. ESP8266, 7. baixo custo. I. Cardoso, Fábio de Sousa.

MÁRCIA DOS SANTOS DE SÁ

**DESENVOLVIMENTO DE UM PROTÓTIPO DE DISPOSITIVO DE
BAIXO CUSTO BASEADO EM IOT (*INTERNET OF THINGS*) PARA
DETECÇÃO E PREVENÇÃO DE INCÊNDIOS EM AMBIENTES
RESIDENCIAIS**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletrônico(a).

Nota obtida: 9.8 (Nove ponto oito)

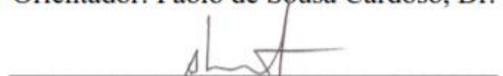
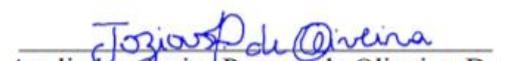
Aprovada em 25/01/2022

Área de concentração: Sistemas Embarcados, Internet das Coisas.

BANCA EXAMINADORA



Orientador: Fábio de Sousa Cardoso, Dr.


Avaliador: Bruno da Gama Monteiro, Me
Avaliador: Jozias Parente de Oliveira, Dr.

Manaus 2022

DEDICATÓRIA

Dedico este trabalho ao meu pai e principalmente a minha mãe (*in memoriam*) que já não se encontra mais presente entre nós.

AGRADECIMENTOS

A Deus por me manter saudável. Aos meus pais, e irmãos (Marcela e Marcelo), que sempre me apoiaram. Ao meu orientador pela paciência e incentivo. E aos meus amigos, Darc Pabla e Jezreel Souto, que nunca mediram esforços para me ajudar durante esse desafio. Meu muito obrigada.

RESUMO

O presente documento tem a finalidade de apresentar o desenvolvimento de um protótipo de dispositivo de baixo custo baseado em *Internet of Things* (IoT) para detecção e prevenção de incêndios em ambientes residenciais. Este sistema tem a capacidade de monitorar e coletar em tempo real, as variáveis do ambiente residencial, através dos sensores de temperatura e gases inflamáveis, e enviá-las via WiFi para a interface web. Além disso, o dispositivo deve atuar para desligar automaticamente a válvula reguladora de gás, caso seja detectado desvio nos parâmetros monitorados. Inicialmente, será apresentada a fundamentação teórica das tecnologias utilizadas neste projeto: conceito do ecossistema IoT, fases do gráfico de curva de crescimento de um incêndio, estudo sobre sensores e do módulo ESP8266. Além disso, são explorados os protocolos de comunicação I2C, HTTP e UDP e os métodos *GET* e *POST* usado para comunicação entre o microcontrolador e a aplicação web. Posteriormente, são apresentadas as etapas e materiais necessários à construção do protótipo IoT, seguida da descrição detalhada do experimento, ferramentas utilizadas e implementação do que foi citado anteriormente. Como resultado das experiências descritas ao longo do projeto, serão exibidos os valores dos parâmetros monitorados pelos sensores na interface web. Por fim, os dados obtidos mostraram que o dispositivo desenvolvido atende aos requisitos propostos ao detectar as variáveis de temperatura, gases inflamáveis e fumaça no ambiente residencial.

Palavras-chave: IoT, Detecção de incêndio, protocolo UDP, HTTP, Sistema Embarcados, ESP8266, baixo custo.

ABSTRACT

This document aims to present the development of a low-cost device prototype based on the Internet of Things (IoT) for fire detection and prevention in residential environments. This system has the ability to monitor and collect in real time, the variables of the residential environment, through temperature sensors and flammable gases, and send them via WiFi to the web interface. In addition, the device must act to automatically turn off the gas regulating valve, if deviation is detected in the monitored parameters. Initially, the theoretical foundation of the technologies used in this project will be presented: the concept of the IoT ecosystem, phases of the growth curve graph of a fire, a study on sensors and the ESP8266 module. In addition, the I2C, HTTP and UDP communication protocols and the GET and POST methods used for communication between the microcontroller and the web application are explored. Subsequently, the steps and materials necessary for the construction of the IoT prototype are presented, followed by a detailed description of the experiment, tools used and implementation of what was mentioned above. As a result of the experiences described throughout the project, the values of the parameters monitored by the sensors will be displayed in the web interface. Finally, the data obtained showed that the developed device meets the proposed requirements by detecting the variables of temperature, flammable gases and smoke in the residential environment.

Keywords: IoT, Fire detection, UDP protocol, HTTP, Embedded System, ESP8266, low cost.

LISTA DE FIGURAS

Figura 1 - Curva de evolução de um incêndio.....	15
Figura 2 - Sintomas causados por exposição ao monóxido de carbono (CO).....	16
Figura 3 - Comunicação UDP entre placas ESP's.....	19
Figura 4 - Módulo NodeMCU ESP12.....	21
Figura 5 - Curva característica de sensibilidade do sensor MQ2.....	22
Figura 6 - Sensor <i>chemiresistor</i> - MQ2.....	23
Figura 7 - Diagrama de bloco do ADS1115.....	24
Figura 8 - Sistema do protocolo de comunicação UDP.....	27
Figura 9 - Arquitetura do sistema proposto.....	29
Figura 10 - Biblioteca da placa NodeMCU ESP12 na Arduino IDE.....	31
Figura 11 - Fluxo de firmware do módulo sensor em diagrama de sequência.....	32
Figura 12 - Estrutura principal do <i>firmware</i> do módulo sensor.....	33
Figura 13 - Trecho do script da função <i>void controll</i>	33
Figura 14 - Fluxo de <i>firmware</i> do módulo atuador em diagrama de sequência.....	34
Figura 15 - Estrutura principal do firmware do módulo atuador.....	34
Figura 16 - Conexão do Sensor MQ2 com o módulo ADS1115.....	36
Figura 17 - Cálculo do módulo ADC no código.....	36
Figura 18 - Circuito do conversor de nível lógico.....	37
Figura 19 - Conexão elétrica do conversor de nível lógico com o módulo ADS1115.....	37
Figura 20 - Esquemático do circuito da placa módulo sensor.....	38
Figura 21 - Leiaute da placa de circuito impresso do módulo sensor.....	39
Figura 22 - Processo de fabricação da placa módulo sensor.....	40
Figura 23 - Soldagem dos componentes na placa módulo sensor.....	40
Figura 24 - Desenho da case do módulo sensor.....	41
Figura 25 - Protótipo do módulo sensor.....	41
Figura 26 - Descrição dos pinos do módulo ESP8266.....	42
Figura 27 - Circuito do módulo atuador.....	42
Figura 28 - Módulo carregador de bateria de lítio.....	43
Figura 29 - Módulo relé de 1 canal.....	43
Figura 30 - Processo de montagem do módulo atuador.....	44
Figura 31 - Protótipo do módulo atuador.....	44
Figura 32 - Cenário de teste de envio de dados entre placas ESP's.....	45
Figura 33 - Dados enviados para o módulo atuador.....	46
Figura 34 - Sistema Web.....	47
Figura 35 - Modelagem do banco de dados do sistema.....	47
Figura 36 - Conexão do banco de dados com o <i>Front-End</i>	48
Figura 37 - Código de tratamento de erro.....	49
Figura 38 - Protocolo HTTP no código do microcontrolador.....	49
Figura 39 - Saída de dados no navegador Web.....	50
Figura 40 - Dashboard da interface Web.....	50
Figura 41 - Processo de calibração do MQ2.....	51
Figura 42 - Parâmetros para os cenários dos testes de ambiente.....	52
Figura 43 - Recipiente descontaminado representando o status seguro.....	52
Figura 44 - Recipiente contaminado com fumaça representando o status crítico.....	53
Figura 45 - Aferição do sensor MQ2 no ambiente totalmente contaminado por fumaça.....	53
Figura 46 - Status indicando perigo no teste de temperatura.....	54

Figura 47 - Status seguro do teste de vazamento de gás inflamável.	55
Figura 48 - Status crítico dos testes de vazamento de gases inflamáveis.....	55
Figura 49 - Diretivas para classificação do teste de integração do sistema.....	56
Figura 50 - Cenário de teste de integração do protótipo de sistema.....	56
Figura 51 - Gráfico da coleta de dados da temperatura em 48h.....	58
Figura 52 - Valor médio da temperatura.	59
Figura 53 - Gráfico da coleta de dados do sensor de gases inflamáveis em 48h.....	59
Figura 54 - Gráfico de histograma do sensor de gases inflamáveis em 24h.	60
Figura 55 - Gráfico de histograma do sensor de gases inflamáveis em 48h.	60
Figura 56 - Gráfico do teste de detecção de gás.....	61
Figura 57 - Protótipo do dispositivo para detecção e prevenção de incêndios em ambientes residenciais.	62

SUMÁRIO

INTRODUÇÃO	13
1 REFERENCIAL TEÓRICO	15
1.1 AS ETAPAS DE UM INCÊNDIO	15
1.2 OS PERIGOS DA EXPOSIÇÃO AO MONÓXIDO DE CARBONO	16
1.3 GASES LIQUEFEITOS DE PETRÓLEO (GLP)	17
1.4 INTERNET OF THINGS	17
1.5 TECNOLOGIA WIFI 802.11	18
1.6 PROTOCOLO HTTP	18
1.7 USER DATAGRAM PROTOCOL (UDP)	19
1.8 MICROCONTROLADOR ESP8266	20
1.8.1 Módulo NodeMCUESP12	20
1.9 COMUNICAÇÃO I2C	21
1.10 SENSORES	21
1.10.1 Sensor Chemiresistors	22
1.10.2 Sensor de Temperatura	23
1.11 CONVERSOR ANALÓGICO-DIGITAL (ADC).....	24
1.12 SISTEMA OPERACIONAL EM TEMPO REAL	24
2 MATERIAIS E MÉTODOS	26
2.1 MÉTODO PROPOSTO	26
2.2 MATERIAIS UTILIZADOS NA CONSTRUÇÃO DO MÓDULO SENSOR	28
2.3 MATERIAIS UTILIZADOS NA CONSTRUÇÃO DO MÓDULO ATUADOR	28
3 IMPLEMENTAÇÃO DO PROJETO	29
3.1 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	30
3.2 IMPLEMENTAÇÃO DO <i>FIRMWARE</i>	32
3.2.1 O <i>firmware</i> do módulo sensor	32
3.2.2 O <i>firmware</i> do módulo atuador	34
3.3 DESENVOLVIMENTO DO PROTÓTIPO DO MÓDULO SENSOR	35
3.3.1 ADC para o sinal analógico do sensor MQ2	35
3.3.2 Circuito do conversor de nível lógico bidirecional	36
3.3.3 Esquemático do módulo sensor	38
3.3.4 Confecção da PCI módulo sensor	39
3.3.5 Montagem do protótipo do módulo sensor	40
3.4 DESENVOLVIMENTO DO MÓDULO ATUADOR	42
3.4.1 Módulo relé de 1 canal	43
3.4.2 Montagem do módulo atuador	44
3.5 PROTOCOLO DE COMUNICAÇÃO UDP.....	45
3.5.1 Teste de envio	45
3.6 DESENVOLVIMENTO DA INTERFACE WEB	46
3.6.1 Base de dados	47
3.6.2 Tratamento de erro	48
3.6.3 A comunicação com firmware do microcontrolador com o <i>Back-End</i>	49
3.6.4 Dashboard final da interface Web	50
3.7 TESTE DE VALIDAÇÃO DO PROTÓTIPO DO SISTEMA.....	51

3.7.1 Teste de fumaça.....	52
3.7.2 Teste de temperatura.....	54
3.7.3 Teste de vazamento de gases inflamáveis	54
3.7.4 Teste de integração do sistema	56
4 RESULTADOS OBTIDOS	58
4.1 COLETAS E ANÁLISE DE DADOS DO SISTEMA.....	58
4.2 RESULTADO DO TESTE DE INTEGRAÇÃO DO SISTEMA.....	61
CONCLUSÃO.....	63
REFERÊNCIAS.....	64
APÊNDICE A - CÓDIGO DO MÓDULO SENSOR.....	66

INTRODUÇÃO

Este trabalho tem como tema o desenvolvimento de um protótipo de dispositivo de baixo custo baseado em *Internet of Things* (IoT) para detecção e prevenção de incêndios em ambientes residenciais. Casas e apartamentos concentram quase metade dos incêndios registrados em Manaus, conforme dados da Secretaria de Segurança Pública (SSP-AM). Ao longo do ano de 2019, o Corpo de Bombeiros Militar do Amazonas (CBMAM) foi mobilizado para cessar chamadas em 109 ocorrências (G1, 2020).

O presente trabalho expõe a problemática da escassez de projetos voltados para dispositivos de detecção e prevenção de incêndios para ambientes residenciais. Além disso, os métodos convencionais de sistema de detecção de incêndio são baseados em sistemas cabeados monitorados por uma central que normalmente é instalada na portaria ou recepção, a maioria dos sistemas são apenas de alerta e não de prevenção, os dispositivos são de difícil aquisição para ambientes residenciais devido ao seu alto preço no mercado.

Este trabalho tem por hipótese a ideia de que é possível construir um dispositivo de baixo custo, comparado aos sistemas de mercado, para monitoramento em tempo real de cozinhas e áreas de risco em residências, utilizando uma plataforma de *hardware* denominada NodeMCU ESP12 que, através da rede WiFi envia pela internet os dados de sensores de temperatura, gás inflamável e de fumaça. O dispositivo deve atuar para desligar automaticamente a válvula reguladora de gás, caso seja detectado vazamento ou desvio nos parâmetros monitorados, habilitando o alarme sonoro e através de uma interface Web, notificar o usuário sobre as informações colhidas do ambiente. Objetivou-se, portanto, construir um protótipo que será responsável por detectar dados sensoriais, por meio dos sensores MQ2 e BME280, somado ao microcontrolador, para coleta, processamento e envio de dados. Os resultados de saída, serão exibidos por meio de uma interface para dispositivos *mobile*, além de habilitar o alerta sonoro, caso seja detectado desvio nos parâmetros monitorados.

O referido trabalho justifica-se por desenvolver um sistema capaz de monitorar e detectar em tempo real via rede IoT, vazamento de gás de cozinha, fumaça e as flutuações das variáveis de temperatura, além de diminuir acidentes com intoxicação por Gás Liquefeito de Petróleo (GLP). É importante ressaltar que incêndios em algumas comunidades podem gerar perdas de vidas, perda de bens e custos com a rede pública de saúde. Além da vantagem de não alterar a instalação elétrica existente. A pesquisa envolve a utilização de vários conceitos

estudados nas disciplinas de Engenharia Eletrônica, tais como: Fundamentos de Eletrônica, Análise de Circuitos Elétricos, Técnicas de Programação em Engenharia Eletrônica, Microprocessadores e Microcontroladores, Projetos de Sistemas Microprocessados, Análise de Sinais, Sensores e Instrumentação Eletrônica e Construção Eletrônica.

Este trabalho está dividido em quatro capítulos: referencial teórico, materiais e métodos, implementação e análise e interpretação dos resultados.

O primeiro capítulo está destinado ao referencial teórico, responsável pela revisão dos assuntos abordados dentro do escopo do projeto, bem como a definição das fases do gráfico de curva de crescimento de um incêndio e os perigos da exposição do corpo humano ao monóxido de carbono, produzidos desde a primeira fase de um incêndio até a extinção da combustão. Também, foi abordado o conceito do ecossistema IoT e os protocolos de comunicação HTTP e UDP e os métodos *GET* e *POST*, que serão abordados e implementados no projeto, bem como o *Analog Digital Converter* (ADC) para transformar um sinal analógico em um sinal digital. No entanto, os algoritmos utilizados neste trabalho foram desenvolvidos em linguagem C/C++. A linguagem de programação tem como objetivo a criação do firmware que será embarcado no microcontrolador através do ambiente de desenvolvimento integrado do Arduino IDE.

No segundo capítulo é apresentada a metodologia, nela serão descritas as etapas para construção do protótipo, quais matérias foram utilizadas, quais os procedimentos técnicos, quais as etapas seguidas para o desenvolvimento do algoritmo, da aquisição de dados.

No terceiro capítulo encontra-se a implementação, nela descreve-se a execução dos passos da metodologia, além de exibir os passos para a instalação e configuração das bibliotecas que serão utilizadas no firmware. Também são apresentados os diagramas elétricos das placas e sensores e a criação de uma interface web. Os programas feitos durante a implementação do projeto, são mostrados nos apêndices.

O quarto capítulo está destinado a análise e interpretação dos resultados. Neste capítulo estão os dados e parâmetros sistemáticos obtidos a partir do protótipo e a coleta de informações do ambiente.

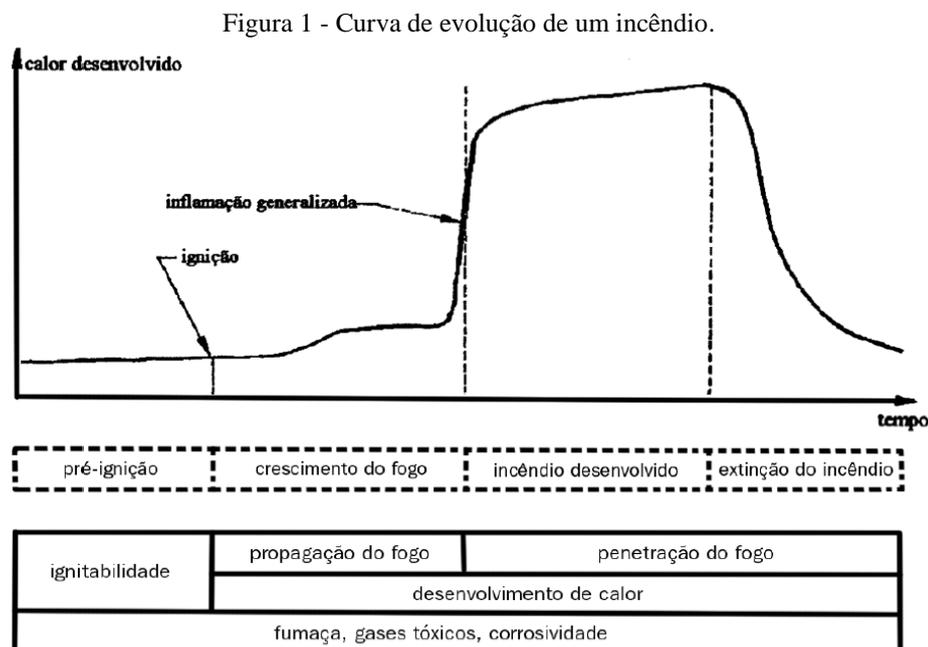
Por fim, na conclusão retoma-se o que foi proposto como hipótese e objetivo em comparação com os resultados obtidos, onde aborda a análise do comportamento do protótipo.

1 REFERENCIAL TEÓRICO

Neste capítulo, será realizada uma pesquisa bibliográfica visando definir as melhores ferramentas e técnicas para o cumprimento dos requisitos do projeto.

1.1 AS ETAPAS DE UM INCÊNDIO

O fogo é a consequência de uma reação química exotérmica que libera luz e calor, é também pode ser entendido como uma entidade gasosa emissora de radiação e decorrente da combustão. No entanto, é necessário levar em conta os principais mecanismos de transmissão do fogo, tais como a condução do calor, convecção do calor e a radiação de energia. É através da radiação de energia que o incêndio se manifesta, ou seja, um incêndio tem todo o fator químico do fogo. Entretanto, é um fenômeno que não tem controle sobre a combustão, alimentando-se dos diversos materiais do ambiente e sofrendo influências de vários fatores do ambiente tais como: Estado da matéria (sólido, líquido ou gás), calor específico, ponto de ignição, mistura inflamável, quantidade de calor, composição química, quantidade de oxigênio disponível e umidade. O gráfico da Figura 1 possui quatro fases distintas de um incêndio (CARLO et al., 2008).



Fonte: (CARLO et al., 2008, p. 71).

A primeira fase representa o crescimento da curva de pré-ignição do gráfico, em geral, tem duração entre cinco a vinte minutos até a ignição. Esse estágio produz pouco calor, mas com grande potencial para preencher ambientes com gases e fumaça. No segundo estágio ocorre a propagação e o crescimento do fogo no ambiente. Na terceira fase o incêndio irá se propagar por todo ambiente e todos os materiais entrarão em combustão atingindo temperaturas com valores acima de 1.100 °C. Por fim, na quarta fase o incêndio irá diminuir de intensidade na proporção que vai se consumindo os materiais combustíveis do ambiente. O desenvolvimento do projeto irá focar na primeira fase da curva que representa o processo inicial de ignição. E no momento da primeira fase que o sistema de detecção deve operar para notificar os residentes e consequentemente inibir o crescimento das chamas.

1.2 OS PERIGOS DA EXPOSIÇÃO AO MONÓXIDO DE CARBONO

Os gases e fumaça estão presentes desde a primeira fase do incêndio. À medida que a concentração de oxigênio diminui no ambiente e aumenta a quantidade de fumaça, inicia-se o desenvolvimento do monóxido de carbono (CO). Presente nos aquecedores de gás o monóxido de carbono é um gás inodoro, incolor e tóxico, é letal para os seres humanos e animais, pois quando se combina com as moléculas de hemoglobina (componente dos glóbulos vermelhos do sangue), reduzem a disponibilidade dessas moléculas para fazerem o transporte do oxigênio até o tecido do corpo. Com isso, o indivíduo tem consequentes desmaios e poderá morrer por asfixia. Sem falar, no fator psicológico, e o aumento da aceleração da respiração se dá também em função da angústia a que as vítimas estão submetidas, ocasionando o pânico e o desespero. A Figura 2 apresenta os sintomas típicos sofridos pelos humanos quando expostos ao monóxido de carbono, baseados no tempo de exposição e concentração (CARLO et al., 2008).

Figura 2 - Sintomas causados por exposição ao monóxido de carbono (CO).

CONCENTRAÇÃO (PPM)	SINTOMAS
35	nenhum sintoma adverso dentro de 8 horas de exposição
200	dor de cabeça após 2 a 3 horas de exposição
400	dor de cabeça e náusea após 1 a 2 horas de exposição
800	dor de cabeça, náusea e distúrbios após 45 minutos de exposição; morte em até 2 horas de exposição
1.000	perda da consciência
1.600	dor de cabeça, náusea e distúrbios após 5 a 10 minutos de exposição, perda da consciência após 30 minutos de exposição
12.800	efeitos fisiológicos imediatos, perda da consciência e risco de vida após 1 a 3 minutos de exposição

Fonte: (CARLO et al., 2008, p. 65).

1.3 GASES LIQUEFEITOS DE PETRÓLEO (GLP)

O GLP é a alcunha técnica do popularmente conhecido gás de cozinha, utilizado na maioria das residências no Brasil. Porém, como muitos imaginam, o GLP, não se trata de um único gás, mas sim de várias composições e misturas gasosas dos hidrocarbonetos. É formado pela combinação de butano, propano, isobutano, propeno e buteno, todos gases derivados do petróleo. No entanto, os gases predominantes são o propano e o butano. Em caso de vazamento, os gases por serem mais pesados que o ar, vão se acumulando ao nível do chão ao ponto de expulsar gradativamente o oxigênio do ambiente, causando asfixia em quem estiver no local. (SINDIGAS, 2021).

Para que o usuário perceba um possível vazamento de gás GLP, os fabricantes de GLP, introduziram em seu componente químico, um agente de mau cheiro. O odorante utilizado é o etil mercaptano, que possui odor característico e desagradável. O etil mercaptano é selecionado por não ser corrosivo, com baixo teor de enxofre. Teoricamente, o GLP é incolor, seja na forma líquida ou vapor. A nuvem branca, que aparece em torno de um ponto de vazamento, não é o produto em si, mas o vapor de água gelada condensado do ar circundante pela evaporação do líquido. Com todos esses elementos de segurança, ainda são vistos explosões e acidentes domésticos oriundos do vazamento de gás. Porém, a maioria dos acidentes, são causados pelo manuseio incorreto ou a falta de atenção do usuário (ABDULLAH, 2010).

1.4 INTERNET OF THINGS

A Internet of Things (IoT) emergiu dos avanços de várias áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. Entretanto, o termo IoT foi cunhado pelo pesquisador do *Massachusetts Institute of Technology* (MIT) Kevin Ashton no final da década de 90, durante uma apresentação de negócios feita à P&G (Procter & Gamble). Nesse cenário, Ashton explicou a ideia de utilizar etiquetas *Radio Frequency Identification* (RFID) na cadeia de suprimentos da multinacional. Em síntese, a IoT trata-se de um ecossistema que conecta objetos físicos, através de uma rede sem fio, para trocar, armazenar e coletar dados para consumidores, através de uma aplicação de software (ASHTON, 2009).

A IoT é uma palavra muito popular agora. É realmente um conceito importante que está fazendo mudanças na sociedade. Afinal, tudo está incorporado em uma infraestrutura de rede e são pilhas de protocolos aceitos pela comunidade IoT. Também, há dispositivos de gateway ou dispositivos intermediários que trocam dados com dispositivos de sensor e

fornece dados para servidores de análise de *Big Data*, que serão usados por vários serviços, incluindo automação residencial, serviços de saúde, serviços de gerenciamento de energia e serviços de emergência.

1.5 TECNOLOGIA WIFI 802.11

A WiFi é uma técnica de rede IoT amplamente utilizada. Isso ocorre porque está em todos os smartphones e dispositivos inteligentes. Baseada no padrão de comunicação sem fio 802.11 do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) ficou mundialmente conhecida como um acrônimo para *Wireless Fidelity*, em alusão à *HiFi*, termo usado para amplificadores de áudio de alta qualidade. Entretanto, o primeiro esboço do que seria uma norma referente ao padrão 802.11, nasceu em 1994 com uma velocidade que variava entre 5 e 11Mbps e trabalhava sobre uma frequência de 2.4GHz. No decorrer dos anos o padrão IEEE 802.11 apresentou várias versões (a/b/g/n/ac/ax), objetivando sua melhoria contínua.

Devido ao seu preço baixo, o padrão 802.11n, é bastante usado em redes domésticas, em sistemas embarcados. O padrão 802.11n consegue atingir uma velocidade de até 300 Mbps e suporta ambas as bandas *Industrial Scientific and Medical (ISM)* na faixa de 2,4 GHz e 5 GHz. Contudo, esse não é seu único diferencial em relação aos outros, seu sinal é mais estável e compatível com outros padrões. Além de disponibilizar o acesso ao *WiFi Direct*, permitindo que os dispositivos se conectem diretamente uns aos outros (OLIVEIRA, 2021).

1.6 PROTOCOLO HTTP

O protocolo de transferência de hipertexto (HTTP) foi desenvolvido para permitir a comunicação entre cliente-servidor. É a base de qualquer troca de dados na Web. No entanto, as mensagens enviadas pelo cliente, são chamadas de solicitações de requisições (requests), e as mensagens enviadas pelo servidor como resposta (responses). Os elementos que constituem as requisições são comumente chamados de método HTTP. Contudo, o método, normalmente é um verbo como *GET*, *POST*, *DELETE*, *PUT*, *PATCH* ou um substantivo como *OPTIONS* ou *HEAD* que estabelece qual procedimento o cliente quer fazer. Caracteristicamente, os dois métodos *HTTP* mais comuns são: *GET* e *POST* (MOZILLA, 2021).

As requisições do tipo *GET* é usado em operações para receber informações, sendo assim ele é utilizado nas funções de buscar todos os usuários, buscar as leituras das informações de todos os sensores, ou seja, retornando uma enorme quantidade de informações, no caso de

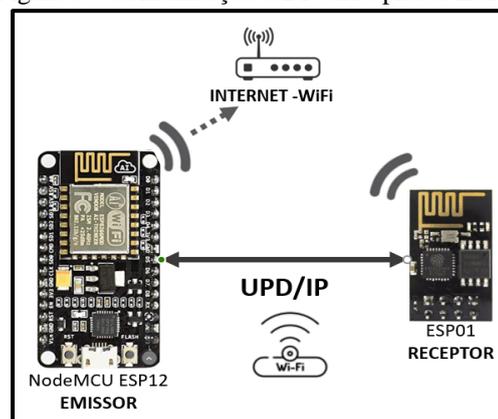
usuário por exemplo, são retornados todos os usuários, o que podemos chamar de método *index*. As requisições do método *POST*, são usadas para enviar informações para o sistema remoto, que podem ser usadas para inseri-las em um banco de dados. Este método aceita parâmetros no formato *JSON*.

1.7 USER DATAGRAM PROTOCOL (UDP)

O *User Datagram Protocol* (UDP) é um protocolo da camada de transporte desenvolvido em 1979 por John Postel. Um cientista da computação, conhecido por contribuir significativamente para o desenvolvimento da Internet. Contudo, o sistema a que a maioria das pessoas se refere como 'A Internet' não é apenas um protocolo: é uma pilha inteira de camadas de protocolos, geralmente denominada pilha TCP/IP. Entre essas pilhas de protocolo encontra-se, o UDP, que resulta em uma comunicação mais rápida e tem uma latência muito menor, porque não requer uma conexão aberta para enviar mensagens. Comparado a outros protocolos, o UDP realiza este processo de forma simples: envia pacotes diretamente para um computador de destino ou microcontrolador, indicando a ordem dos pacotes transferidos ou verificando se eles chegaram como pretendido. Os pacotes UDP são chamados de 'datagramas'. Basicamente, o protocolo UDP tem a função exclusiva de chaveamentos de pacote utilizando o endereço IP, determinado a origem e o destino da possível interface UDP/IP (POSTEL, 1980).

No contexto do sistema embarcado, a Figura 3, exibe o funcionamento básico do UDP. Quando um microcontrolador usa o UDP, os pacotes de dados se movem continuamente do emissor para o receptor, sem esperar a confirmação do receptor. Em caso de perda de pacotes de dados na extremidade receptora, o remetente não os enviará. Isso permite que o UDP se comunique com mais rapidez.

Figura 3 - Comunicação UDP entre placas ESP's.



Fonte: Autoria própria.

1.8 MICROCONTROLADOR ESP8266

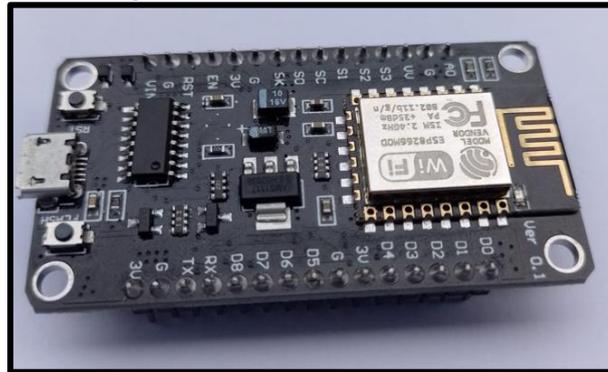
As características a serem analisadas ao escolher um microcontrolador são a arquitetura do processamento, o custo, o desempenho relacionado ao consumo de energia e a interface de comunicação para troca de dados com outros sistemas embarcados, sensores e firmware. Entretanto, o modelo escolhido para o projeto vem da família de microcontroladores ESP8266 produzido pela fabricante chinesa *Espressif*, um microcontrolador com base nas características analisadas e de baixo custo.

O ESP8266 é um microcontrolador de 32 bits que funciona na frequência-padrão de 80 MHz, podendo chegar a 160 MHz. Sua tensão nominal é de 3,3 volts e sua comunicação IEEE 802.11 b/g/n WiFi pode atuar no modo cliente ou no modo *WiFi Direct*. Também, em sua arquitetura encontram-se as principais comunicações seriais mais utilizadas na microeletrônica, que podem ser divididas em interfaces síncronas ou assíncronas. As comunicações síncronas compartilham o *clock* para que possam fazer a leitura e a escrita ao mesmo tempo, ao contrário das seriais assíncronas que não compartilham um sinal de *clock*. São exemplos de seriais de comunicação síncronas as interfaces SPI e I2C e as interfaces RS232 e USB são comunicação assíncronas. No entanto, comparando o ESP8266 com outras soluções Wi-Fi de mercado, é uma ótima opção para a maioria dos projetos de IoT (ESPRESSIF, 2020).

1.8.1 Módulo NodeMCUESP12

A placa NodeMCU ESP12, ilustrada na Figura 4, é uma plataforma de prototipagem de código aberto, com o chip ESP8266 integrado. O módulo é bastante acessível para aplicações de baixo custo, sendo ideal para aplicações voltadas a IoT. A placa vem com um regulador de tensão LDO (*Low Dropout*) para manter a tensão estável em 3,3 V capaz de fornecer com segurança até 600mA. A programação pode ser feita usando LUA, uma linguagem interpretada desenvolvida no Brasil por professores da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) ou usando a ferramenta de desenvolvimento integrado do Arduino. Além disso, o ESP12 possui um transceptor WiFi 802.11, para que não só possa interagir com a internet conectado a uma rede WiFi, mas também podemos configurar uma rede própria, permitindo que outros dispositivos se conectem diretamente ao ESP12. O módulo disponibiliza 11 pinos de I/O (entrada/saída) que facilitam a prototipação e montagem de projetos de circuitos (OLIVEIRA, 2021).

Figura 4 - Módulo NodeMCU ESP12.



Fonte: Autoria própria.

1.9 COMUNICAÇÃO I2C

O Protocolo de comunicação *Inter-Integrated Circuit* (I2C), é um protocolo de conexão de barramento multimestre para comunicação serial. Foi originalmente projetado pela *Philips Semiconductor* em 1982. Recentemente, é um protocolo amplamente usado para comunicação de curta distância. O protocolo prevê apenas duas vias bidirecionais: *Serial Data* (SDA) para dados seriais e *Serial Clock* (SCL). Teoricamente, é uma comunicação mestre-escravo, o microcontrolador assume a função de mestre do barramento e os demais dispositivos ligados a ele assumem a função de escravo. O diferencial dessa interface de comunicação é seu sistema de endereçamento, possibilitando que cada periférico escravo ligado a ele assumam um endereço e que todos se comuniquem no mesmo barramento. Então, conforme o *clock* avança, os bits de dados individuais são enviados para o microcontrolador (OLIVEIRA, 2021).

1.10 SENSORES

Um sensor é um dispositivo de entrada que detecta as mudanças físicas do ambiente e as converte em dados que podem ser interpretados por um ser humano ou uma máquina. Entretanto, o termo “dispositivo de entrada” na definição de um sensor significa que ele é parte de um sistema maior que fornece entrada para um sistema de controle principal, como um processador ou um microcontrolador. O exemplo mais simples de um sensor é o microfone, que converte a energia do som em um sinal elétrico que pode ser amplificado, transmitido, gravado e reproduzido (THOMAZINI; ALBUQUERQUE, 2009).

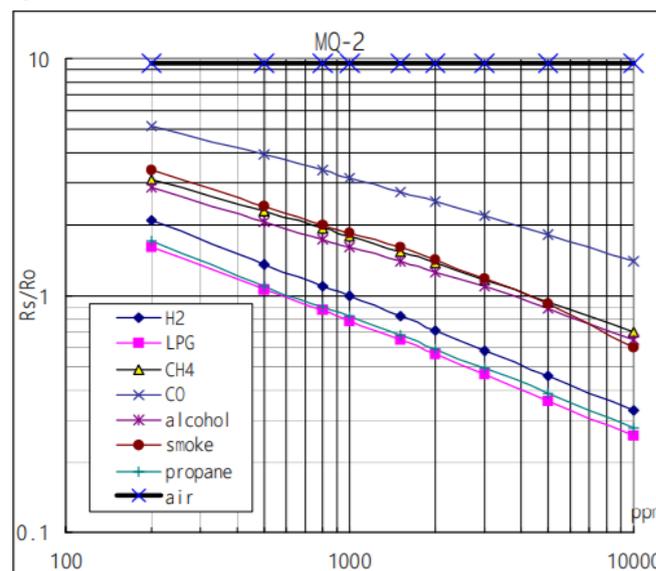
Os sensores estão na maioria dos produtos eletrônicos hoje em dia. Eles estão conduzindo uma revolução tecnológica chamada, "A Internet das Coisas". São usados nas

indústrias farmacêuticas e de semicondutores para medir e controlar as reações químicas necessárias para criar essas importantes matérias-primas. Os sensores de posição são encontrados em veículos e aeronaves. Sensores de movimento são usados em smartphones e sistemas de segurança. Sensores ergonômicos são encontrados em tablets, PCs e telefones. Os sensores químicos nos protegem dos perigos de incêndio e vapor tóxico. E o sensor de temperatura monitora o ambiente no qual ele é inserido. No entanto, o referido trabalho irá destacar os sensores químicos no contexto de proteção contra incêndio e gases tóxicos e os sensores de temperatura.

1.10.1 Sensor Chemiresistors

Na automação residencial os sensores *chemiresistors* tem fundamental importância na necessidade do monitoramento de gases inflamáveis e fumaça. Em particular o MQ2, da família MQ, cuja detecção é baseada na mudança da resistência elétrica em resposta à mudança química no ambiente. O sensor MQ2 funciona com uma tensão de 5 DC e consome cerca de 800mW. De acordo com o gráfico da Figura 5, a concentração mínima de gás que pode ser medida está em torno de 100ppm e a máxima em torno de 10000ppm. Em geral, o gráfico mostra que a resistência do sensor diminui com o aumento da concentração de gás. Iniciando-se em uma sensibilidade de detecção com 300ppm para gases inflamáveis, o suficiente para detectar concentrações de Gás Liquefeito do Petróleo (GLP), fumaça, propano, metano e monóxido de carbono em qualquer ambiente (FELIPEFLOP, [s.d.]).

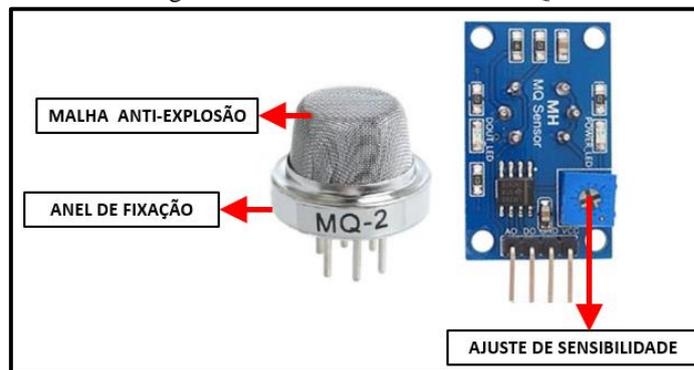
Figura 5 - Curva característica de sensibilidade do sensor MQ2.



Fonte: (FELIPEFLOP, [s.d.], p. 2).

O sensor é envolvido em duas camadas de malha fina de aço inoxidável chamada de rede anti-explosão, essa pequena estrutura garante que o elemento do aquecedor dentro do sensor não causará uma explosão. A malha de aço inoxidável é capaz de filtrar as partículas suspensas no ar de forma que apenas os elementos gasosos possam passar para dentro da câmara. A malha é ligada ao resto do corpo do sensor por meio de um anel de fixação revestido de cobre como mostra a Figura 6 (ESTEVAM JUNIOR; FARINELLI, 2019).

Figura 6 - Sensor *chemiresistor* - MQ2.



Fonte: Autoria própria.

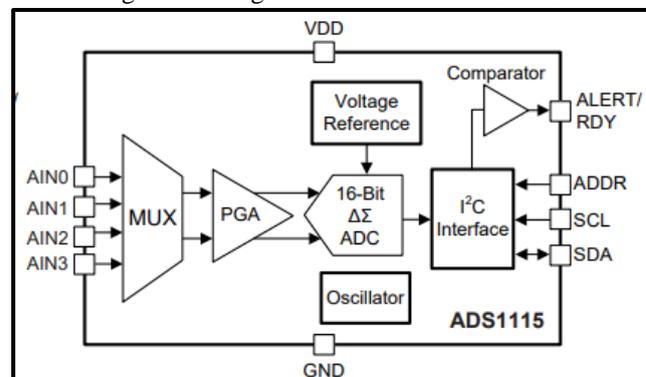
1.10.2 Sensor de Temperatura

O sensor de temperatura é um dispositivo de aferição que identifica variações da temperatura no ambiente a partir de uma característica física correspondente do dispositivo, como uma resistência elétrica, o campo eletromagnético (EMF) ou radiação térmica sendo que a maneira como um sensor de temperatura funciona depende da propriedade física que constitui o mesmo. Existem vários tipos de sensores térmicos comumente usados em sistemas embarcados. Entretanto, para o projeto o BME280 atinge um alto desempenho em todas as aplicações que requerem medição de umidade e temperatura aplicáveis à automação residencial, navegação, saúde e sistema de posicionamento global (GPS). O sensor de temperatura fornece um tempo de resposta rápido para uma aplicação rápida e alta precisão em uma ampla faixa de temperatura. A escolha do sensor BME280 deu-se, não apenas por suas características de precisão e desempenho, contudo o sensor tem um valor de mercado acessível e de fácil aquisição local. Além de ser um sensor digital, é compacto, funciona com um baixo consumo de energia. A conexão do sensor com o microcontrolador ocorre através da interface de comunicação I2C que pode ser alimentada usando de 1,71 a 3,6 volts, podendo aferir a umidade relativa do ar, de 0 a 100%, com precisão de $\pm 3\%$; como também mede a temperatura do ambiente, num intervalo de $-40\text{ }^{\circ}\text{C}$ a $85\text{ }^{\circ}\text{C}$, com precisão de $\pm 1,0\text{ }^{\circ}\text{C}$ (BOSCH, 2020).

1.11 CONVERSÃO ANALÓGICO-DIGITAL (ADC)

A maioria dos dispositivos que queremos aferir, por exemplo, os sensores de fumaça, variam continuamente em uma tensão analógica. Contudo, os dados dos sensores analógicos devem ser convertidos para um formato digital antes que possam ser interpretados por um microcontrolador. Para o projeto foi utilizado o módulo ADS1115, cuja arquitetura está ilustrada na Figura 7. É um conversor analógico-digital, responsável por converter um nível contínuo de tensão em uma sequência de bits. Este módulo possui uma resolução de 16 bits com leituras de até 860 amostras por segundo. Suas quatro entradas analógicas podem ser configuradas como *single ended* (terminal singular) ou como par diferencial. Além dessa característica, o módulo ADS1115 possui *Programmable Gain Amplifier* (PGA) interno de até 16 bits, que é responsável por levar os níveis dos sinais para níveis mais altos. O módulo pode ser alimentado por tensões entre 2 V a 5 V. O dispositivo utiliza o protocolo I2C para interface de comunicação, contudo, se um dispositivo analógico estiver conectado aos seus pinos SCL e SDA, a leitura de seus dados sofrerá uma conversão analógico-digital (TEXAS, 2018).

Figura 7 - Diagrama de bloco do ADS1115.



Fonte: (TEXAS, 2018, p. 1).

1.12 SISTEMA OPERACIONAL EM TEMPO REAL

Um sistema operacional em tempo real, comumente conhecido como RTOS, é um componente de software que alterna rapidamente entre as tarefas, dando a impressão de que vários programas estão sendo executados ao mesmo tempo em um único núcleo de processamento. No contexto da programação para *Internet of Things* (IoT) precisam considerar o sincronismo entre os dispositivos e a concorrência no tratamento de eventos. Por tanto, a programação deve considerar técnicas de sincronismo, possibilitando que um único programa

conte com várias tarefas sendo executadas ao mesmo tempo. Contudo, em sistemas em tempo real, o prazo de uma tarefa deve ser cumprido em todos os casos, caso contrário, a tarefa produzirá resultados inúteis no decorrer da depuração do programa (OLIVEIRA, 2021).

O *FreeRTOS* é um dos RTOS mais utilizados no mercado de sistema embarcados, projetado para ser compacto o suficiente para funcionar em um microcontrolador, pois os sistemas embarcados geralmente têm requisitos de tempo real. Um requisito de tempo real é aquele que especifica que o sistema embarcado deve responder a um determinado evento dentro de um tempo estritamente definido. Portanto, o *FreeRTOS* fornece a funcionalidade central de agendamento em tempo real, é uma comunicação entre tarefas. No *FreeRTOS*, uma tarefa de execução é chamada de thread (FREERTOS, 2021).

No caso do microcontrolador ESP8266, existem algumas restrições de interrupções, seu processamento não pode levar mais que 50 us para executar e têm um esquema de prioridades no qual interrupções de prioridade mais alta podem interromper aquelas de prioridade mais baixa. Entretanto, o ambiente de desenvolvimento integrado (IDE) do Arduino possui integração com o *FreeRTOS* para os microcontroladores da família de ESP8266, objetivando a resolução do problema de interrupção na depuração do código.

2 MATERIAIS E MÉTODOS

Neste capítulo é retratada a sequência de eventos para a criação do projeto, desde a pesquisa e instalação das bibliotecas no ambiente de desenvolvimento da IDE do Arduino ao sensoriamento do ambiente, processamento e envio de dados. O sistema consiste em um dispositivo de detecção e monitoramento, para coletar, processar e gerenciar informações dos níveis de temperatura, de gás inflamável e de fumaça em ambientes residenciais. A partir dos dados sensorizados em conjunto com a IoT.

Para praticidade é construção do projeto, o mesmo foi dividido em um módulo sensor, e um módulo atuador. O módulo sensor é formado pelos dispositivos de sensores de temperatura, fumaça e gás, um conversor analógico-digital e por um módulo NodeMCU ESP12.

O módulo atuador é formado por um módulo ESP01 da família de microcontroladores ESP8266 é um relé para ativar a solenoide de fechamento da válvula reguladora de gás. No entanto, o módulo atuador permanecerá estático até o momento que receber o comando provido do módulo sensor, no qual conterà a mensagem que o instruirá para fechar a válvula reguladora de gás.

2.1 MÉTODO PROPOSTO

Na primeira etapa foram feitas pesquisas na área de programação, dando ênfase nos conceitos de thread e linguagem de programação C++, para uso das bibliotecas do sensor BME280 e do conversor analógico-digital ADS1115, fornecidas pelo fabricante. Para instalar os drivers da placa NodeMCU ESP12, foi acessado o site do fabricante para download da biblioteca e instalação no ambiente de desenvolvimento Arduino IDE. Em seguida, foram realizadas pesquisas referentes ao microcontrolador ESP8266, bem como os protocolos de comunicação HTTP, UDP e os métodos de GET e POST.

Na segunda etapa, foi desenvolvido o *firmware* utilizando a interface de desenvolvimento IDE Arduino versão 1.8.16. Inicialmente foram incluídas as bibliotecas necessárias para o reconhecimento dos sensores e protocolos de conectividades com o microcontrolador. Em seguida, o *script* foi desenvolvido dentro da estrutura de threads, para que o script possa suportar tarefas paralelas. Ao passo que o desenvolvimento do *firmware* acompanha o diagrama de sequência.

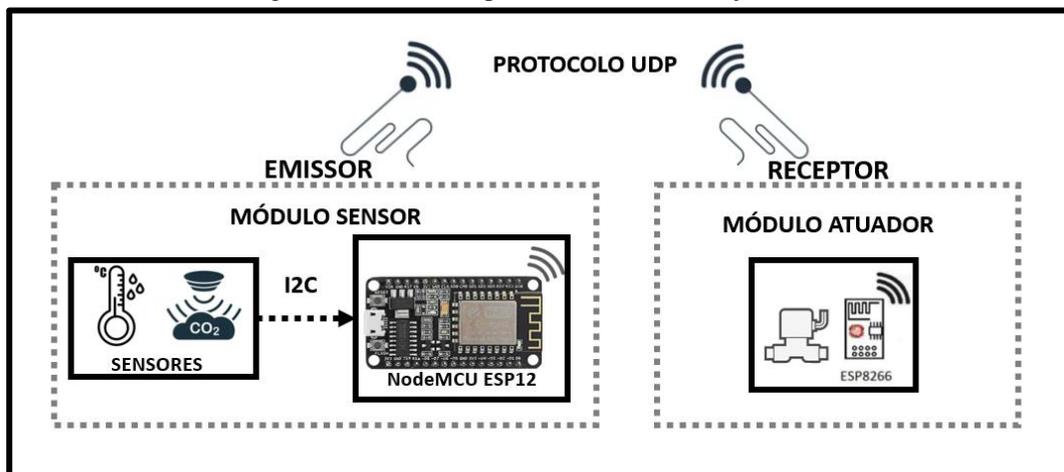
Na terceira etapa fez-se a confecção de um protótipo denominado módulo sensor, capaz de realizar o sensoriamento do ambiente residencial, através dos sensores MQ2 e

BME280. Efetuou-se o pré-aquecimento do sensor de gás e fumaça por um tempo mínimo de 48 horas. Isso garante que o elemento do aquecedor dentro do sensor não causará uma falsa detecção. Em seguida, foi efetuada a conexão do conversor analógico-digital (ADC) com o sensor MQ2, para transformar o sinal analógico produzido na saída do sensor em uma conexão de barramento I2C. Posteriormente, foi realizado o esquemático e o leiaute do circuito do módulo sensor no software gratuito KiCad versão 5.1.9. Detalhes do esquemático do circuito podem ser visualizados na seção 3.3.3. Em seguida foi feita a prototipação da placa de circuito impresso, utilizando o método de transferência térmica com corrosão por solução de Perclorato de Ferro. Por fim, foi feita uma inspeção geral da placa por meio de testes com multímetro, e o módulo sensor foi acondicionado dentro de um case impresso em impressora 3D. Objetivando a aquisição de dados e posterior transformação em informações, baseadas em análise de dados.

Na quarta etapa fez-se a confecção do módulo atuador, habilitado para fechar a válvula reguladora de gás, provenientes das informações enviadas do módulo sensor através do protocolo de comunicação UDP. Todo seu esquemático pode ser visualizado na seção 3.4. Além disso, o módulo atuador foi acondicionado dentro de um case impresso em impressora 3D para melhor visualização no ambiente.

Na quinta etapa foi realizada a configuração do protocolo de comunicação UDP, como também o teste de envio entre os módulos ESP12 e ESP01. Para o desenvolvimento desta tarefa foi necessário obter o endereço de IP do módulo ESP01. Em seguida, temos um ESP01, que está configurado como receptor e o ESP12 como emissor (Figura 8). Ao passo que o emissor envia uma mensagem de ativação para o receptor, quando as variáveis de temperatura e fumaça estão fora dos padrões estabelecidos dentro do ambiente de teste.

Figura 8 - Sistema do protocolo de comunicação UDP.



Fonte: Autoria própria.

Na sexta etapa foi desenvolvido uma interface Web, utilizando as tecnologias de desenvolvimento, HTML, CSS e *Javascript*, para o *Front-End*, objetivando apresentar uma aplicação intuitiva para o usuário. Para gravação de dados e comunicação com a *Back-End*, utilizou-se os scripts PHP e SQL. Posteriormente, foram criados os scripts para a comunicabilidade com a página web e o microcontrolador, por intermédio do protocolo HTTP e os métodos GET e POST. Por fim, na última etapa, foram realizados testes de validação, onde o módulo sensor e o módulo atuador foram testados simultaneamente em conjunto com a aplicação web, para validar a entrada e saída de dados e funcionalidade do sistema.

2.2 MATERIAIS UTILIZADOS NA CONSTRUÇÃO DO MÓDULO SENSOR

Para a confecção da placa do módulo sensor, foram utilizados os materiais listados abaixo:

- 1 módulo NodeMCU ESP12;
- 1 sensor de temperatura digital - BM280;
- 1 conversor analógico-digital ADS1115;
- 1 conversor de nível lógico 5V - 3.3V;
- 1 sensor de gás/fumaça MQ2;
- 2 leds, cores verde e vermelha;
- 5 resistores de 10k Ω ;
- 1 display Lcd Oled 12C 0.96”;
- 1 *Buzzer*;
- 1 placa de fenolite 5x7 cm;
- 1 barra de borne para encaixe do módulo NodeMCU ESP12;
- 1 conector P4 Fêmea;

2.3 MATERIAIS UTILIZADOS NA CONSTRUÇÃO DO MÓDULO ATUADOR

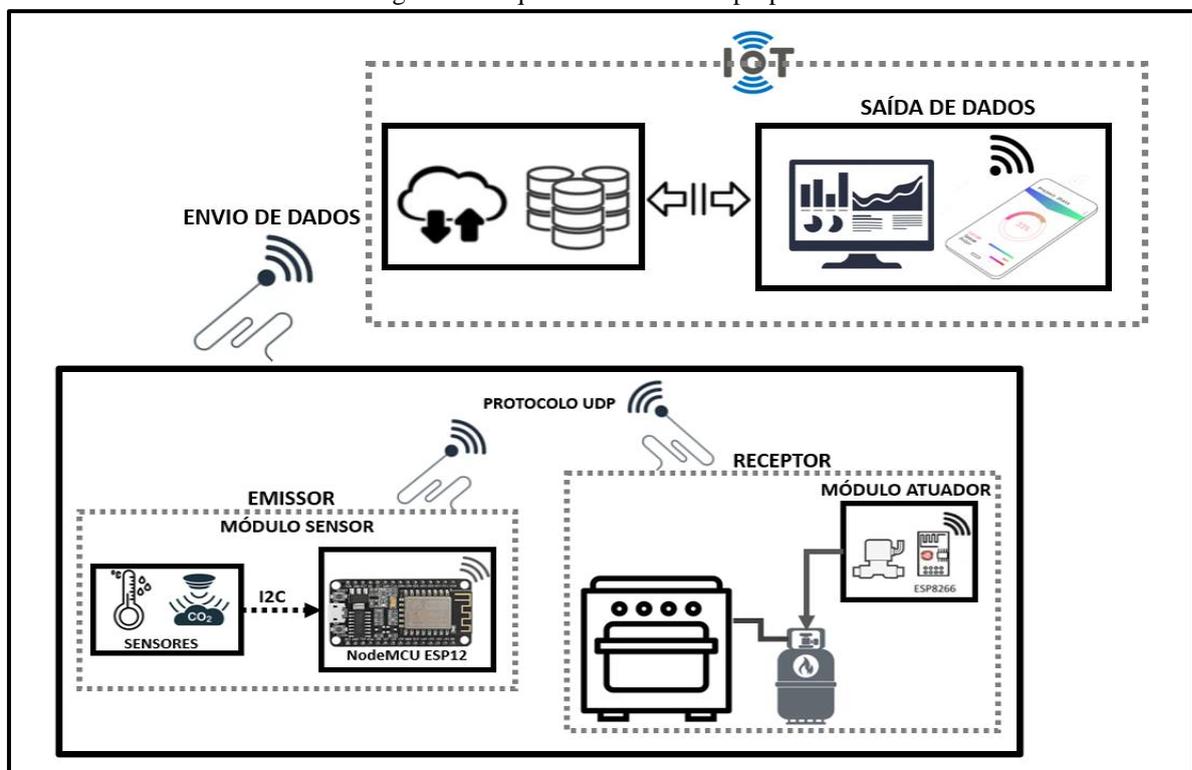
Para a confecção da placa do atuador, foram utilizados os materiais listados abaixo:

- 1 módulo ESP8266-01;
- 1 regulador de Tensão AMS1117 5v p/ 3.3v;
- 1 bateria de polímero de lítio 3.7V;
- 1 TP4056 carregador 5V;
- 1 módulo relé 1 canal;
- Mini solenoide 12V.

3 IMPLEMENTAÇÃO DO PROJETO

Este capítulo apresenta os procedimentos da construção do protótipo. Foi desenvolvido o firmware no ambiente da Arduino IDE, uma placa para o módulo sensor para o monitoramento e coleta de dados, uma placa para o módulo atuador, objetivando desligar a válvula de gás, caso seja detectado vazamento do mesmo. Além disso, foi desenvolvido uma interface web para notificar o usuário em tempo real sobre os eventos do ambiente protegido. A Figura 9 ilustra o sistema proposto.

Figura 9 - Arquitetura do sistema proposto.



Fonte: Autoria própria.

São apresentados neste capítulo os seguintes tópicos:

- preparação do ambiente de desenvolvimento;
- implementação do *firmware*;
- desenvolvimento do protótipo do módulo sensor;
- desenvolvimento do módulo atuador;
- protocolo de comunicação UDP;
- desenvolvimento da interface Web e integração com o *firmware*;
- testes de validação do protótipo do sistema.

3.1 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento deste trabalho foram instaladas algumas bibliotecas específicas que serão usadas na programação do módulo NodeMCU ESP12. No entanto, como o ESP12 não é um microcontrolador nativo do ambiente de desenvolvimento do Arduino. A comunidade do ESP8266 criou um driver de instalação para o Arduino IDE (versão 1.8.16), que permite programar a família de ESP8266 usando o ambiente de desenvolvimento e sua linguagem de programação. Primeiramente iremos abrir o Arduino IDE é na aba Arquivo/Preferências insira o endereço http://arduino.esp8266.com/stable/package_esp8266com_index.json no campo “URLs de gerenciador de placas adicionais”. Em seguida, navegue até o Gerenciador de Placas através do caminho, Ferramentas/Placa/Gerenciador de placas. No campo de pesquisa busque por ‘ESP8266’, é aparecerá uma única opção *esp8266 by ESP8266 Community versão 3.0.2*, instale o pacote clicando em “Instalar”. O resultado pode ser visualizado na Figura 10. Após a instalação dos drivers da placa, foi necessário instalar as demais bibliotecas, as quais serão descritas abaixo:

- *Adafruit_BME280.h*: trata-se da biblioteca do sensor de temperatura BME280, a mesma pôde ser incluída em Sketch/Incluir bibliotecas/Gerenciador de bibliotecas, e na caixa de pesquisa digite *Adafruit BME280*, em seguida clicando em instalar.

- *Adafruit_ADS1X15.h*: refere-se à biblioteca do módulo ADS1115, responsável por converter a saída analógica do sensor de gás e fumaça em uma sequência de bits, contudo, navegue até Sketch/Incluir bibliotecas/Gerenciador de bibliotecas, e na caixa de pesquisa digite ‘ADS1X15’, em seguida clicando em instalar.

- *ESP8266WiFi.h*: designada por disponibilizar diversas funções para conexão WiFi. Permitindo, que o projeto utilize funções de Client e Server.

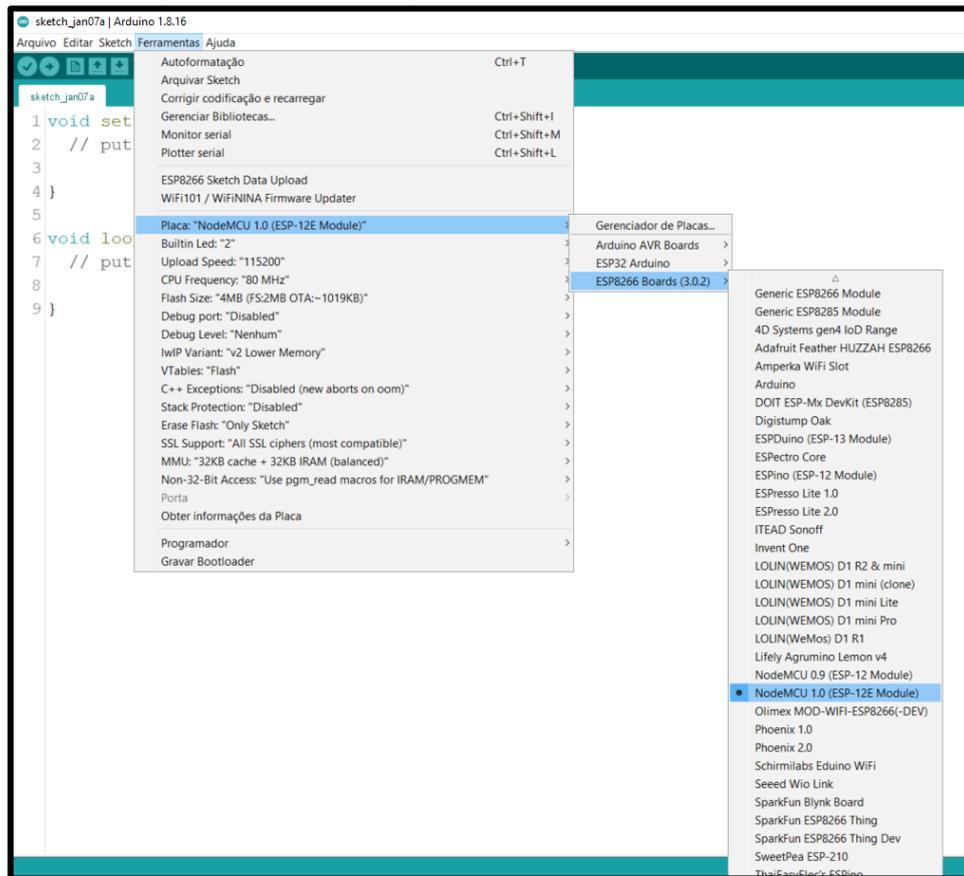
- *Wire.h*: responsável pela comunicação I2C entre os dispositivos e o microcontrolador, foi baixado diretamente do Sketch/Incluir Biblioteca/Gerenciar Bibliotecas do Arduino IDE.

- *SSD1306Wire.h*: seguindo o mesmo procedimento de instalação do módulo ADS1115, esta biblioteca é responsável por controlar o *display OLED* com o ESP8266.

- *ESP8266HTTPClient.h*: responsável pela comunicação entre o microcontrolador é o BackEnd da interface web. Esta biblioteca foi instalada ao mesmo tempo que os drivers da placa ESP8266.

- *math.h*: é uma biblioteca básica do Arduino, que permite a realização de operações matemáticas.

Figura 10 - Biblioteca da placa NodeMCU ESP12 na Arduino IDE



Fonte: Autoria própria.

Ao mesmo tempo que os drivers da placa ESP8266 foram instalados, também foi inserida automaticamente a biblioteca *WiFiUdp.h*. E incorporada ao projeto através da diretiva `#include <WiFiUdp.h>`, e a mesma tem a função de trocar dados entre placas ESP8266 com a conexão WiFi através do protocolo UDP.

No projeto se faz necessário o uso dos *threads*, para utilizá-la dentro da programação paralela do código do projeto, para que o mesmo possa suportar tarefas concomitantemente, dessa forma, foi necessário realizar o download diretamente do repositório do *github*, localizado no endereço <https://github.com/ivanseidel/ArduinoThread.git>. Após extrair o arquivo compactado, o mesmo foi copiado para o local de instalação da Arduino IDE, na pasta `Arduino/Libraries`. Após este procedimento a biblioteca pode ser inclusa utilizando a diretiva `#include "Thread.h"`. Abaixo estão alguns comandos importantes da biblioteca *Thread.h*:

- *Thread*: classe básica, que contém métodos para definir e executar call-backs.
- *ThreadController*: responsável por gerenciar vários Threads dentro do código.

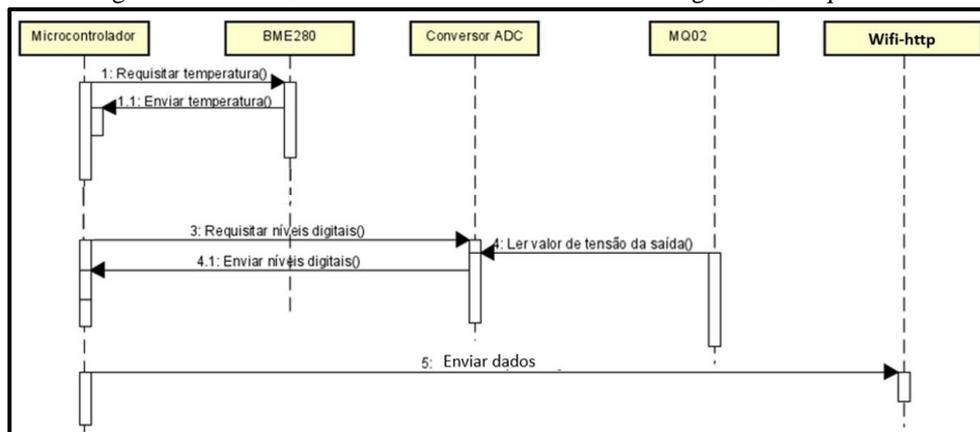
3.2 IMPLEMENTAÇÃO DO *FIRMWARE*

Nesta etapa será descrito o procedimento para implementação do *firmware*, e os métodos abordados. Quanto à linguagem de programação utilizada, o *firmware* dos módulos sensor e atuador são inteiramente programados em C++.

3.2.1 O *firmware* do módulo sensor

O fluxo do módulo sensor é apresentado na Figura 11. Nessa figura verifica-se que o microcontrolador ESP12 requisita os valores de temperatura, gás inflamável/fumaça dos sensores BME280, MQ02, respectivamente. Após obter essas informações, o microcontrolador cria os pacotes de dados e os envia, através da WiFi e protocolo http.

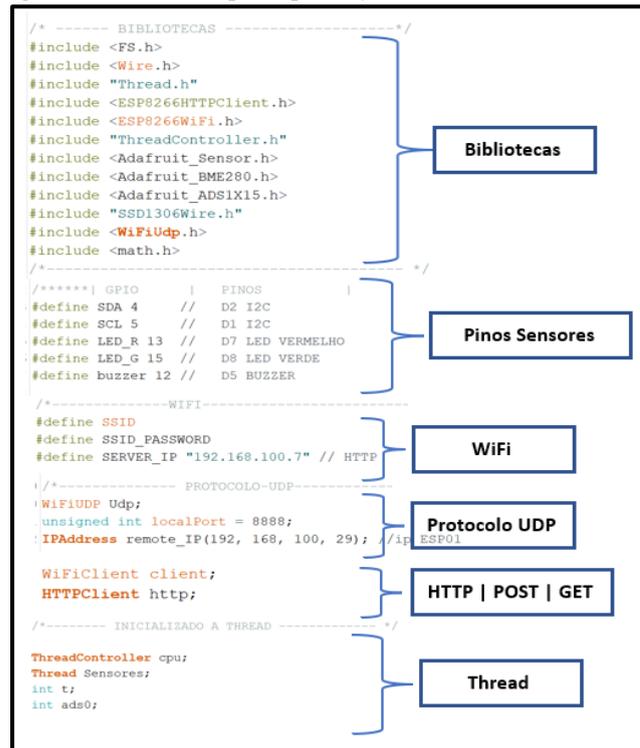
Figura 11 - Fluxo de firmware do módulo sensor em diagrama de sequência.



Fonte: Autoria própria.

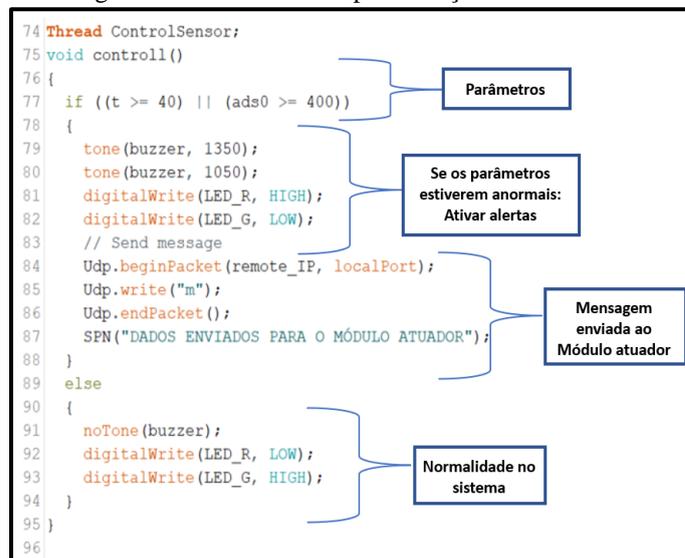
Na Figura 12, o *firmware* do módulo sensor usa as bibliotecas nativas do microcontrolador ESP8266: *Wire*, *Esp8266WiFi*. Outras bibliotecas foram utilizadas, como *Adafruit*: BME280 e ADS1015. Tratando-se da visualização de dados do módulo sensor utilizou-se a tela (display), importando dados da biblioteca SSD1306.

O script *ESP8266HTTPClient* realiza a conexão e captura os dados provenientes dos sensores no formato Json e disponibiliza para as aplicações web, utilizando-se do método POST e GET. Além disso, o script o *WiFi.UDP*, possibilita a transmissão de pequenos pacotes de informação a curtas distâncias, através da comunicação sem fio, entre o módulo sensor e o módulo atuador. Por fim, o *ThreadController.h*, é o principal método utilizado para a sincronia de eventos dentro do algoritmo, toda a estrutura do código utiliza-se de *Thread*. O código completo do módulo sensor encontra-se no apêndice A.

Figura 12 - Estrutura principal do *firmware* do módulo sensor.

Fonte: Autoria própria.

Na figura 13, exibe o trecho do script da função *void controll*, que realiza a operação dos parâmetros do ambiente, ativando todos os alertas estabelecidos, para isso, a função iniciar determinando se a variáveis estão abaixo ou acima do parâmetro estabelecido, é caso estejam acima, realiza a ativação do *buzzer* e leds, enviando uma mensagem para o módulo atuador gerando um comando de ativação é criando um sinal elétrico para ativar o relé.

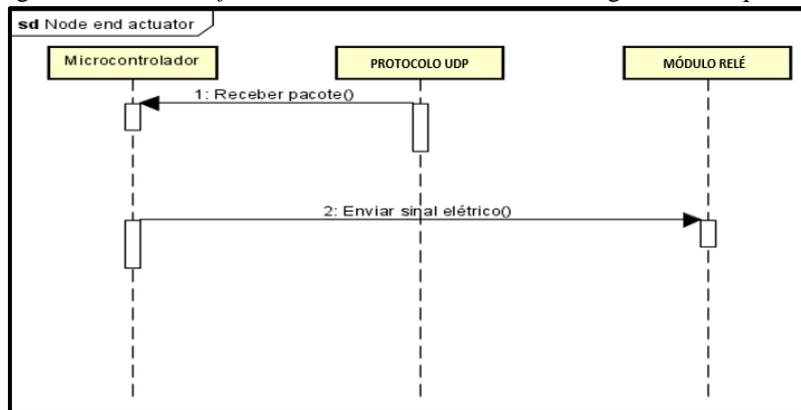
Figura 13 - Trecho do script da função *void controll*.

Fonte: Autoria própria

3.2.2 O *firmware* do módulo atuador

O *firmware* do módulo atuador, tem como estrutura principal o método, *WiFiUdp.h*, do protocolo de comunicação UDP e a funcionalidade de ativação do relé. O fluxo de *firmware* do módulo atuador (Figura 14) inicia com o microcontrolador ESP8266-01, recebendo os pacotes providos da tecnologia UDP. Dentre os pacotes recebidos ele identifica a existência do comando, e então cria o sinal elétrico para ativar o módulo relé.

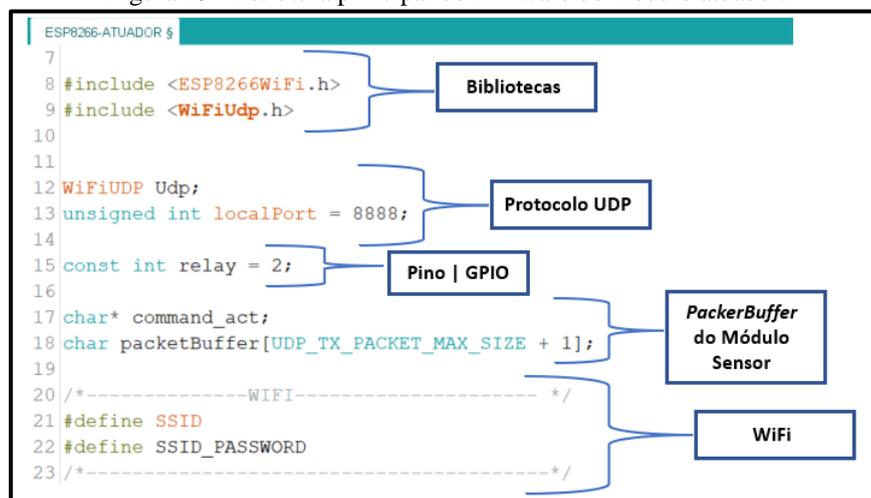
Figura 14 - Fluxo de *firmware* do módulo atuador em diagrama de sequência.



Fonte: Autoria própria.

O *firmware* do módulo atuador tem um objetivo simples, cria o sinal elétrico para ativar o módulo relé de 1 canal. Contudo, toda a estrutura do *firmware* é composta pelo script do *WiFi.UDP*, ao passo que realiza a conexão WiFi interna. Por fim, a estrutura principal do *firmware* do módulo atuador pode ser visualizada na Figura 15.

Figura 15 - Estrutura principal do firmware do módulo atuador.



Fonte: Autoria própria.

3.3 DESENVOLVIMENTO DO PROTÓTIPO DO MÓDULO SENSOR

Nesta etapa descreve-se como foi desenvolvido o sistema de hardware do protótipo. No entanto, o módulo sensor é formado pelos sensores MQ2 e BME280, que respectivamente são sensores de gás/fumaça e temperatura. Vale ressaltar que a conversão dos dados analógicos para digital é feita pelo componente ADS1115, em função do componente, MQ2, ter uma saída analógica em nível alto. Além disso, se faz necessário um regulador de tensão, a fim de reduzir a tensão de 5V para 3,3V, que é o nível de tensão aceitável dos circuitos projetados.

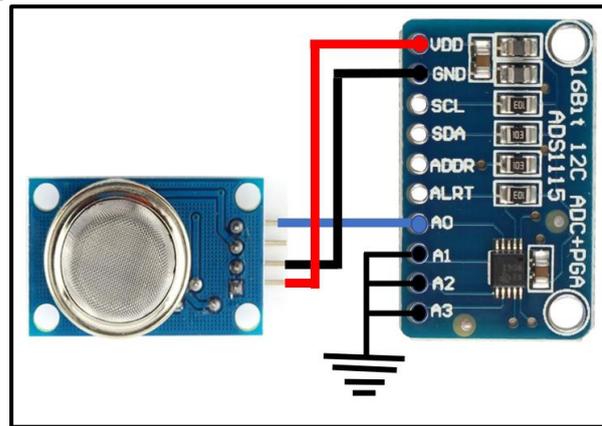
Todo o esquemático do circuito da placa do módulo sensor foi desenhado no software gratuito KiCad 5.1.9. Entretanto, toda parte de conexão com a rede WiFi quanto com a plataforma IoT também é realizada pelo módulo sensor.

3.3.1 ADC para o sinal analógico do sensor MQ2

O conversor analógico-digital (ADC) do NodeMCU ESP12 tem uma resolução de 10 bits, operando em uma faixa de tensão de 1,0 a 3,3 V. No entanto, o sensor MQ2, atua, dentro de uma faixa de sinal analógico. Para que o dispositivo MQ2 possa oferecer uma saída digital para o microcontrolador, é necessário conectá-lo a um dispositivo ADC antes que o sinal possa ser processado no NodeMCU ESP12.

Considerando o sensor MQ2 pré-aquecidos, o NodeMCU ESP12 requisita o valor de tensão analógica da saída do sensor MQ2. Porém, o módulo ADS1115 converte esse valor de tensão analógica para uma saída digital, de forma que a saída A0 do sensor MQ2 que está conectado à entrada AIN0 do ADS1115 podendo ser visto na Figura 16, enquanto as entradas AIN1, AIN2 e AIN3 são aterradas. A saída SDA e SCL do ADS1115 fazem parte do protocolo de comunicação *Inter-Integrated Circuits* (I2C), o qual se baseia numa sincronização do *clock* para transmitir a informação. Contudo, esta informação são os níveis digitais com resolução de 16 bits da respectiva tensão analógica do sensor MQ2.

Figura 16 - Conexão do Sensor MQ2 com o módulo ADS1115.



Fonte: Autoria própria.

Com o nível de tensão do sensor MQ2 já obtido, o microcontrolador faz um cálculo, para determinar a concentração de gases inflamáveis e de monóxido de carbono em partes por milhão (ppm), procedimento feito dentro do *firmware*, ilustrado na Figura 17. Assim, obtém-se as medidas das variáveis do sensor MQ2 de forma digital.

Figura 17 - Cálculo do módulo ADC no código.

```

45 /*-----Calculo do ADC ----- */
46 int readMQx(uint8_t pin)
47 {
48     float v = ReadVoltage(pin);
49     return ceil(10000 * (v / 5.0));
50 }
51 double ReadVoltage(byte pin)
52 {
53     float adc = ads.readADC_SingleEnded(pin);
54     float v = adc * 0.1875e-3;
55     return v;
56 }
57 /*-----Calculo do ADC ----- */

```

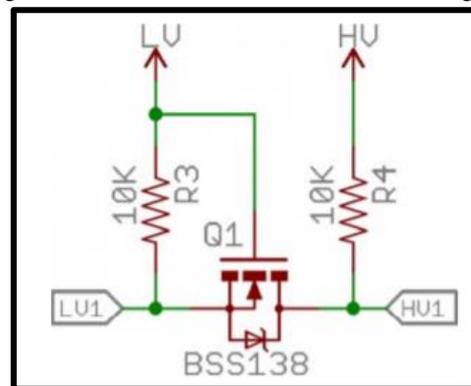
Fonte: Autoria própria.

3.3.2 Circuito do conversor de nível lógico bidirecional

É importante observar que os dois componentes mencionados anteriormente, tais como, o sensor MQ2 e o módulo ADC, operam com uma tensão de alimentação externa de 5V, portanto, suas saídas possuem fundo de escala de 5 V. Contudo, o NodeMCU ESP12 atua em uma faixa de tensão de 1,0 a 3,3 V. Para que todos os dispositivos naveguem na mesma faixa de tensão, é necessário conectá-los a um conversor de nível lógico antes que o sinal de tensão possa ser processado no NodeMCU.

O esquema do circuito da Figura 18, consiste em dois resistores de 10k cada. O dreno e o pino da fonte do MOSFET são puxados até o nível de tensão desejado. Para o perfeito estado de funcionamento do conversor de nível lógico, é necessário conectar a tensão lógica de alto nível, nomeada *High Voltage* (HV), com o pino de drenagem do MOSFET e a tensão lógica de baixo nível, intitulada no circuito como *Low Voltage* (LV), precisa ser conectada com a fonte do MOSFET.

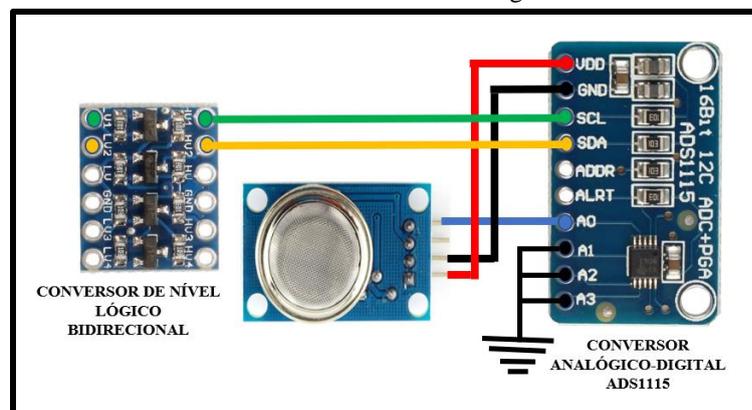
Figura 18 - Circuito do conversor de nível lógico.



Fonte: (SPARKFUN, 2015, p. 2).

Para o projeto, foi utilizado o conversor de nível lógico bidirecional, mostrado na Figura 19, o conversor é capaz de fazer um *step-down* de sinais 5V para 3,3V ou um *step-up* de sinais 3,3V para 5V. São 2 canais com 4 pinos cada, podendo operar simultaneamente ou isoladamente um do outro. Nos pinos centrais, os pinos *High Voltage* (HV), GND e *Low Voltage* (LV), são as tensões de referência. O conversor de nível lógico, está sendo usado para intermediar a interligação entre o canal I2C do conversor analógico-digital com o NodeMCU ESP12.

Figura 19 - Conexão elétrica do conversor de nível lógico com o módulo ADS1115.

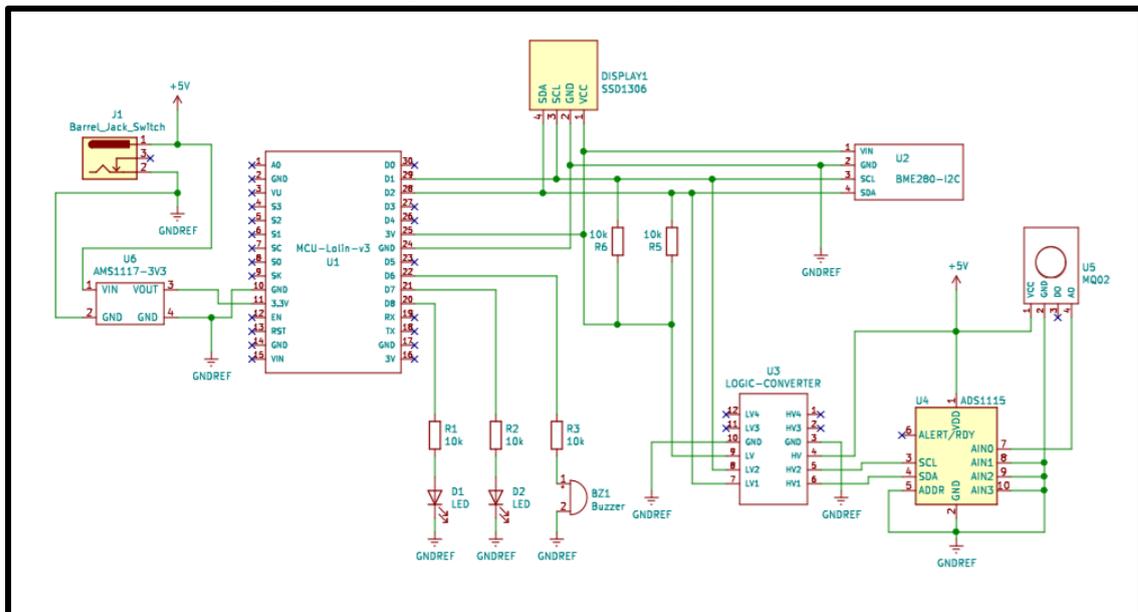


Fonte: Autoria própria.

3.3.3 Esquemático do módulo sensor

Na Figura 20, é possível verificar como os sensores se relacionam com NodeMCU ESP12 dentro do módulo sensor. O ESP12 inicia com a leitura dos valores de temperatura do sensor digital BME280, alimentado em um nível tensão de 3,3 V, utilizando o barramento *Inter-Integrated Circuits* (I2C) para transmissão das variáveis de temperatura, seu terminal SCL e SDA estão conectados aos GPIO's D1 e D2. Percebe-se que esse barramento possui dois resistores de *pull-up* de valor de 10 k Ω , conectados à tensão de 3,3 V provido pelo microcontrolador, configurações recomendadas pelo fabricante para este tipo de barramento. Todo o esquemático do circuito da placa foi desenvolvido na ferramenta KiCad na versão 5.1.9.

Figura 20 - Esquemático do circuito da placa módulo sensor.



Fonte: Autoria própria.

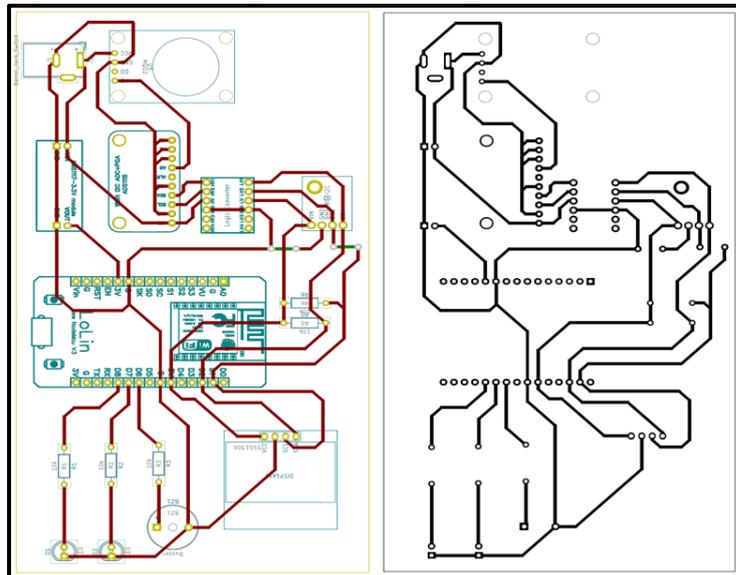
A saída A0 do sensor MQ2 está conectada à entrada AIN0 do ADS1115, enquanto as entradas AIN2 e AIN3 são aterradas, pois não são utilizadas. Seguidamente, o barramento SDA/SCL do ADS1115 são levados aos GPIO's D2 e D1 do NodeMCU ESP12 com intermédio do conversor de nível lógico bidirecional, o qual reduz a tensão de nível lógico alto de 5V para 3,3 V. Contudo, para o alerta sonoro no ambiente foi anexado ao circuito, um *buzzer* ativo, cuja entrada está conectada com o GPIO D6 do NodeMCU ESP12. O circuito, também apresenta dois leds de sinalização de cores vermelha e verde, ligados aos GPIO's D7 e D8 respectivamente. Objetivando sinalizar ao usuário sobre alterações no sistema. Com a finalidade de ajudar na visualização da depuração do resultado, foi anexado ao circuito um

display Oled cuja conexão com o microcontrolador é feita através do protocolo I2C. O esquemático do circuito que compõem o módulo sensor pode ser encontrado no apêndice A.

3.3.4 Confeção da PCI módulo sensor

Com o advento de tecnologias voltadas para softwares de eletrônica que auxiliam no desenho de esquemas elétricos e de leiautes da placa de circuito impresso (PCI). Tornou-se possível a construção do esquemático da PCI do módulo sensor, utilizando-se do *software* KiCad na versão 5.1.9. Na Figura 21, pode ser observado a disposição dos componentes e suas conexões elétricas.

Figura 21 - Leiaute da placa de circuito impresso do módulo sensor.

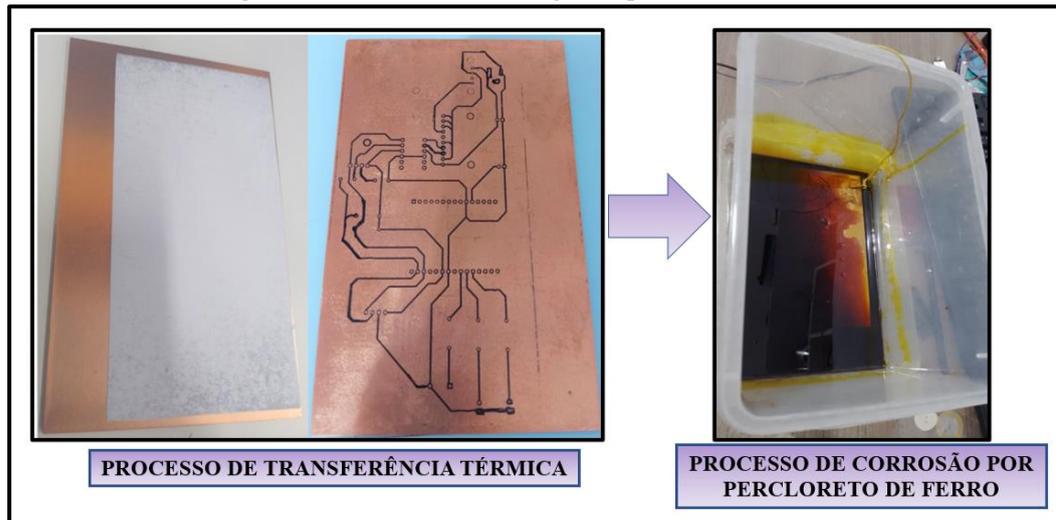


Fonte: Autoria própria.

Na Figura 22, verifica-se o processo de fabricação da placa, utilizando o método de transferência térmica com corrosão por solução de Percloroeto de Ferro. Para a fabricação das placas por este método deve-se:

1. Imprimir o layout da Figura 20 em escala 1:1, através de uma impressora a laser imprimindo em papel fotográfico de única face;
2. Transferir o leiaute impresso para a placa fenolite, na face cobreada, por aquecimento;
3. Mergulhar a placa fenolite na solução de percloroeto de Ferro, a qual deve permanecer em movimentos de inclinação leve durante o período de uma hora.

Figura 22 - Processo de fabricação da placa módulo sensor.



Fonte: Autoria própria.

3.3.5 Montagem do protótipo do módulo sensor

Finalizada a etapa de prototipação, obtém-se a placa com as trilhas de circuito elétrico de acordo com o esquemático do circuito, dando início ao processo de montagem da placa. Para isso, foram utilizados os seguintes itens: componentes eletrônicos definidos anteriormente, pinça metálica, alicates, solda de estanho, estação de solda, esquemático e multímetro. Após a soldagem dos componentes eletrônicos, obteve-se pôr fim a PCI da Figura 23. Preocupou-se em manter o sensor MQ2 na extremidade da placa, pois ele possui um aquecimento característico, e esta posição facilita a captação de gases do ambiente residencial.

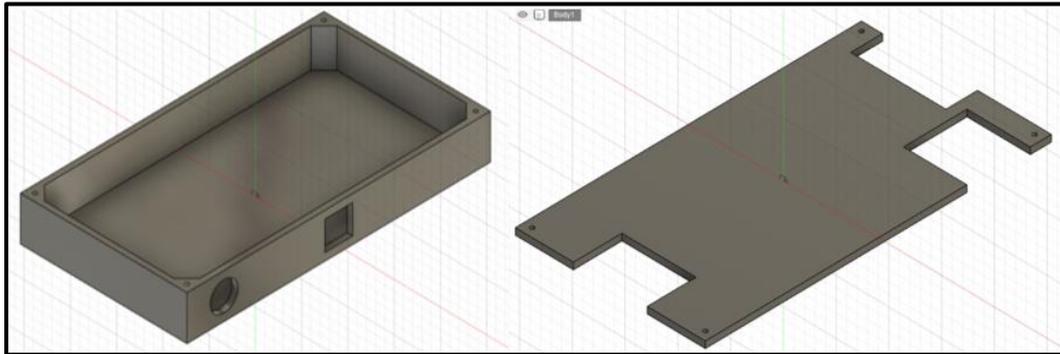
Figura 23 - Soldagem dos componentes na placa módulo sensor.



Fonte: Autoria própria.

A confecção da *case* tem como objetivo condicionar a placa de circuito, para que a mesma não sofra danificações físicas por processos mecânicos, tais como: quedas, contato humano e a interferência externa do ambiente. A *case* foi confeccionada em uma impressora 3D. A Figura 24 ilustra o fatiamento da case do módulo sensor.

Figura 24 - Desenho da case do módulo sensor.



Fonte: Autoria própria.

Por fim, a Figura 25 mostra a primeira versão do protótipo do módulo sensor, utilizado na etapa de coleta e monitoramento de ambiente.

Figura 25 - Protótipo do módulo sensor.



Fonte: Autoria própria.

A placa foi projetada para ser alimentada através de uma bateria de lítio, que possui capacidade de carga de 300mAh e tensão nominal de 3,7V, foi necessário utilizar um módulo carregador de bateria de lítio, com chip TP4056 integrado, tornando o módulo atuador portátil é recarregável, visto na Figura 28.

Figura 28 - Módulo carregador de bateria de lítio.



Fonte: Autoria própria.

3.4.1 Módulo relé de 1 canal

Um relé é uma chave operada eletricamente, como qualquer outra chave, pode ser ligada ou desligada, deixando passar ou não corrente. Podendo controlar tensões baixas, como os 3,3V fornecidos pelos GPIOs do ESP8266, bem como, permite-se controlar altas tensões de 12V, 24V ou tensão de rede de 110/220V. Portanto, para o projeto foi escolhido o módulo relé de 1 canal. Ele atende as necessidades do projeto referentes ao processo de receber o comando de desliga a válvula reguladora, tais comando oriundos do módulo sensor. Além disso, o módulo relé é de baixo valor no mercado, ilustrado na Figura 29.

Figura 29 - Módulo relé de 1 canal.

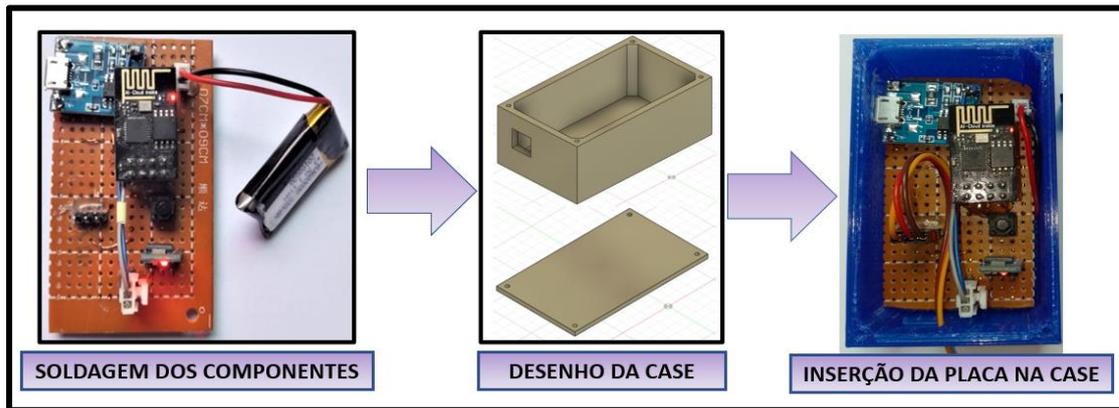


Fonte: Autoria própria.

3.4.2 Montagem do módulo atuador

Na prática, os componentes do circuito esquematizado são soldados na placa universal. Além disso, foi confeccionado em uma impressora 3D a case de proteção do módulo atuador. Após a soldagem dos componentes eletrônicos, a placa foi inserida na case, conforme a Figura 30.

Figura 30 - Processo de montagem do módulo atuador.



Fonte: Autoria própria

Por fim, a Figura 31 mostra o protótipo do módulo atuador, utilizado para desligar a válvula reguladora de gás. É possível observar em conjunto com o módulo atuador na saída do relé a mini solenoide, quando energizado com uma tensão de 12V aplicada nos terminais, é contraído e conforme é desenergizado a mola expande liberando o êmbolo.

Figura 31 - Protótipo do módulo atuador.



Fonte: Autoria própria.

3.5 PROTOCOLO DE COMUNICAÇÃO UDP

Como um dos requisitos estabelecidos pela hipótese, o dispositivo deve atuar para desligar automaticamente a válvula reguladora de gás, caso seja detectado vazamento ou desvio nos parâmetros de gás e fumaça. Para o cumprimento de tais condições, foi necessário estabelecer uma comunicação entre os sensores e a válvula de gás, esta comunicação foi executada através do protocolo de comunicação UDP, permitindo que a aplicação envie um datagrama encapsulado num pacote IPv4 ao destino especificado.

3.5.1 Teste de envio

Na montagem, o módulo sensor, está configurado como um emissor e o módulo atuador como receptor, visualizado na Figura 32. Os testes de envio entre os módulos ESP's foram executados em um ambiente controlável. O módulo sensor coleta os dados dos sensores, é quando ocorre uma perturbação nas variações do ambiente, como por exemplo, a inclusão de fumaça originada da queima de papel. O módulo emissor envia esses dados para o módulo atuador dentro de um pacote utilizando a tecnologia UDP.

Figura 32 - Cenário de teste de envio de dados entre placas ESP's.



Fonte: Autoria própria

Após receber o pacote de dados, o módulo atuador liga o led vermelho, informando fisicamente que a mensagem foi recebida é executada, como pode ser observado na Figura 33. Entretanto, os datagramas enviados durante o uso do UDP, só serão recebidos se porventura ocorra uma perturbação nas variáveis do ambiente, se o ambiente estiver descontaminado dos elementos de gás/fumaça, o módulo permanecerá em modo repouso na espera do comando. Com a finalidade de reduzir o consumo de energia dentro do sistema.

Figura 33 - Dados enviados para o módulo atuador.

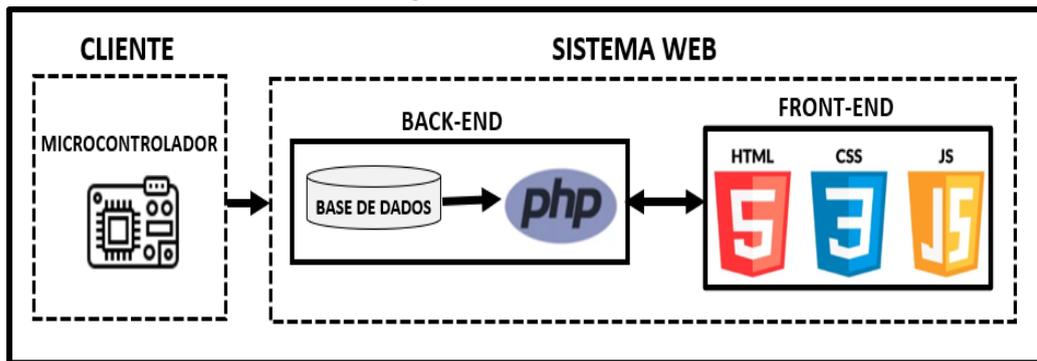


Fonte: Autoria própria.

3.6 DESENVOLVIMENTO DA INTERFACE WEB

Na Figura 34, ilustra a arquitetura do sistema web, visando a interface de interação com o usuário, através das camadas de *front-end*, *back-end* e base de dados. No entanto, na primeira camada, o microcontrolador faz o papel do cliente, é enviar as informações coletadas do ambiente para a base de dados. Em seguida, a linguagem PHP foi utilizada para interagir com o servidor, garantindo o acesso ao banco de dados do sistema. Para a estrutura e o estilo da página web, foram utilizadas camadas de CSS, HTML e JS. A programação de *front-end* da interface web foi desenvolvida por meio da ferramenta de desenvolvimento *Visual Studio Code* na versão 1.62.

Figura 34 - Sistema Web.

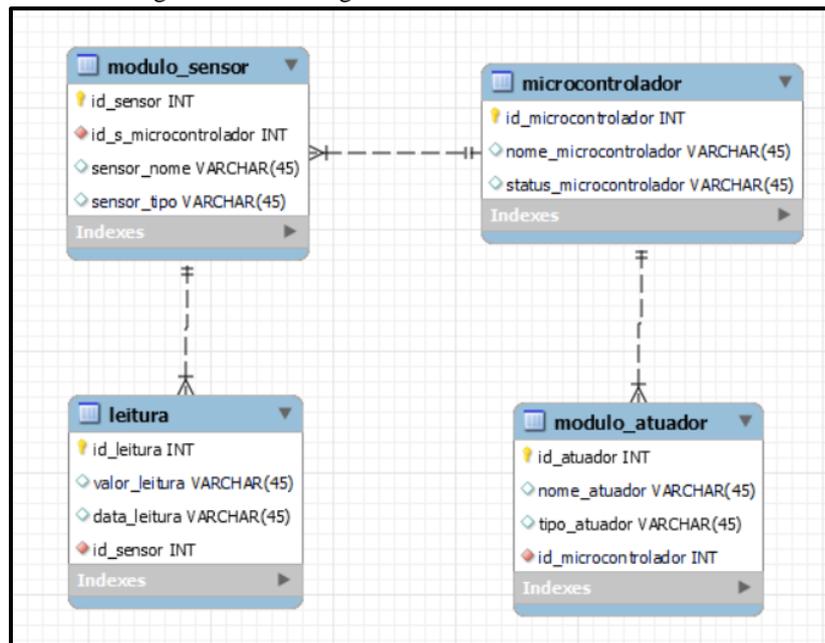


Fonte: Autoria própria.

3.6.1 Base de dados

Antes de desenvolver a interface web, foi necessário criar uma base de dados. Visando interrelacionar um conjunto de dados e organizá-los de tal forma que permita ao sistema a recuperação dos dados coletados. A definição do banco de dados se inicia pela construção do diagrama entidade-relacionamento (ER), ilustrado na Figura 35, elaborada na ferramenta de modelagem da *MySQL Workbench*, versão 8.0. O diagrama ER é uma representação visual dos dados com base no modelo ER e descreve como as entidades estão relacionadas umas às outras no banco de dados.

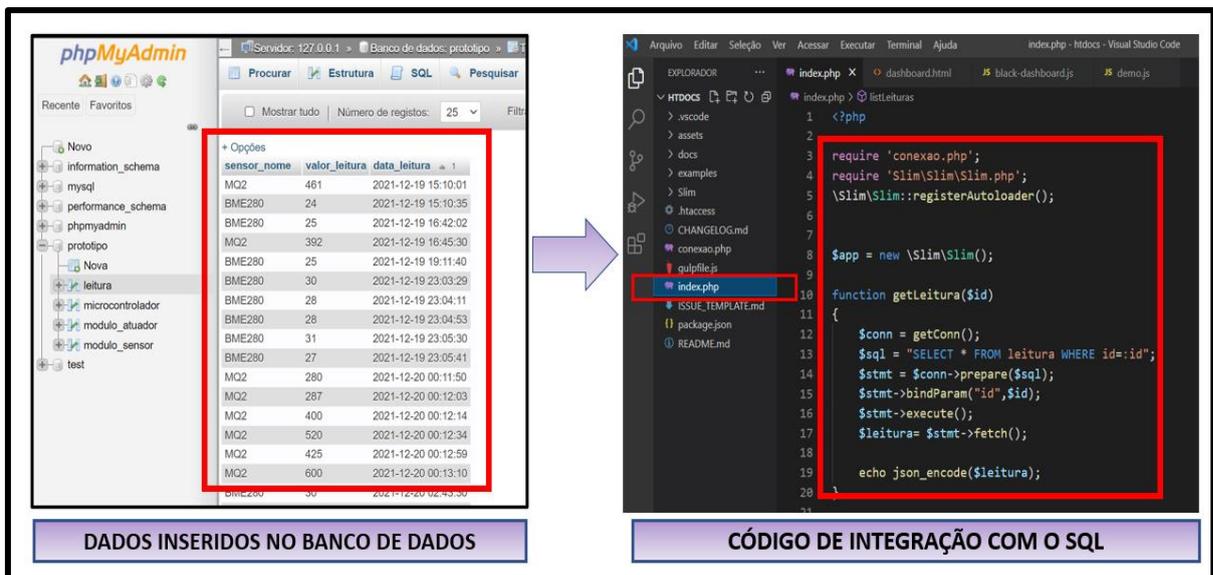
Figura 35 - Modelagem do banco de dados do sistema.



Fonte: Autoria própria.

Depois da modelagem do banco de dados, foi necessário estabelecer a comunicação entre o banco de dados e o *front-end* da página web, esta conexão foi possível através da linguagem PHP, que é um script do lado do servidor de código aberto, tipicamente usado para desenvolvimento web. No entanto, foi utilizado a ferramenta de software livre, *phpMyAdmin* versão 5.1.1, destinada a lidar com os scripts de administração do MySQL, na Figura 36 pode ser visualizada a interação, entre as ferramentas *phpMyAdmin* e *Visual Studio Code*.

Figura 36 - Conexão do banco de dados com o *Front-End*.



Fonte: Autoria própria.

3.6.2 Tratamento de erro

O objetivo do tratamento de erro dentro do código, é impossibilitar que o banco de dados grave informações corrompidas ou quando a própria rede corrompe essas informações. Além disso, quando o módulo sensor estiver desligado, o banco de dados não irá gravar informações. O código irá reconhecer um erro em tempo de execução. Em suma, o código de tratamento de erro foi criado dentro da tipagem dinâmica do *JavaScript* e pode ser visualizado na Figura 37.

Figura 37 - Código de tratamento de erro.

```

//console leitura ----
function onDataReceived(leituras) {

    if (leituras==null){ //Tratamento de Erro
        var chart_data = [0];
        var chart_labels = ["-"];
        var chart2_data = [0];
        var chart2_labels = ["-"];
    }
    else{
        var chart_data = [];
        var chart_labels = [];
        var chart2_data = [];
        var chart2_labels = [];
    }
}

```

Fonte: Autoria própria.

3.6.3 A comunicação com firmware do microcontrolador com o *Back-End*

Para a comunicação do NodeMCU ESP12 com o banco de dados da interface web, utilizou-se o método para envio de requisição HTTP do tipo *POST*, função criada dentro do código do microcontrolador, como pode ser visto na Figura 38. Essa requisição encapsula os dados, que serão enviados para o *script* do programa PHP.

Figura 38 - Protocolo HTTP no código do microcontrolador.

```

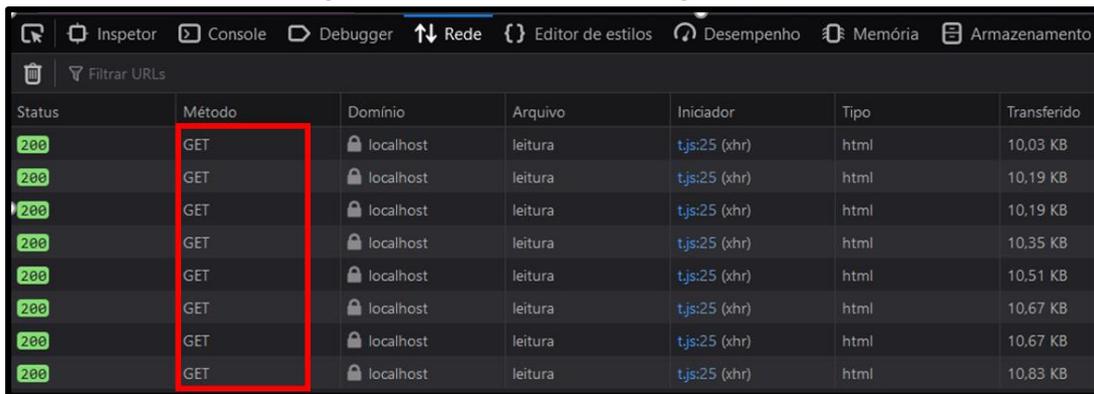
ESP12-22-12-Mp-ATUAL $
---
111 Thread MonitorSerial;
112 void mSerial()
113 {
114     if ((WiFi.status() == WL_CONNECTED)) {
115
116         WiFiClient client;
117         HTTPClient http;
118         SPT("[HTTP] begin...\n");
119         http.begin(client, "http://" SERVER_IP "/leitura"); //HTTP-rota
120         SPT("[HTTP] POST...\n");
121         int httpCode = http.POST("{\"valor\":" + String(t) + ", \"id_sensor\":" + String(2) + "}");
122
123         if (httpCode > 0) {
124             Serial.printf("[HTTP] GET... code: %d\n", httpCode);
125
126             if (httpCode == HTTP_CODE_OK) {
127                 const String& payload = http.getString();
128                 SPT("received payload:\n<<");
129                 SPT(payload);
130                 SPT(">>");
131             }
132         } else {
133             Serial.printf("[HTTP] POST... failed, error: %s\n", http.errorToString(httpCode).c_str());
134         }
135         http.end();
136     }
137 }

```

Fonte: Autoria própria.

O resultado dessa execução dentro do *script* PHP, gera uma saída *GET* que pode ser visualizada na saída do console do navegador (Figura 39).

Figura 39 - Saída de dados no navegador Web.



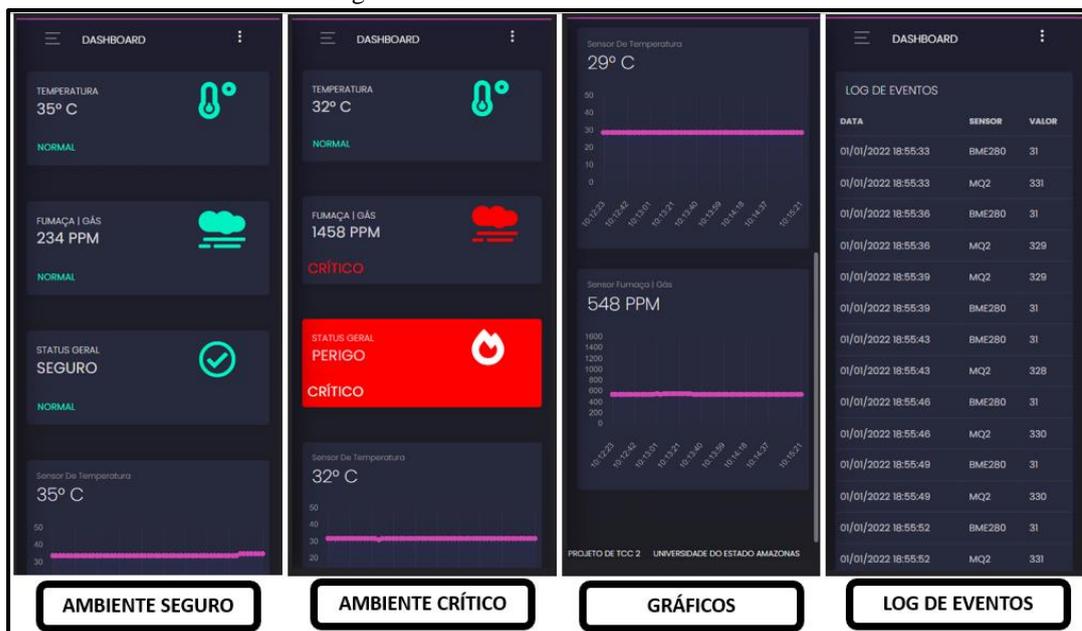
Status	Método	Domínio	Arquivo	Iniciador	Tipo	Transferido
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.03 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.19 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.19 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.35 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.51 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.67 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.67 KB
200	GET	localhost	leitura	t.js:25 (xhr)	html	10.83 KB

Fonte: Autoria própria.

3.6.4 Dashboard final da interface Web

O principal objetivo do desenvolvimento da interface web, é fornecer informações simples e de fácil interpretação acerca dos sensores instalados no ambiente residencial. No entanto o *dashboard* da aplicação web exibe de forma intuitiva os dados atuais dos sensores em forma de gráfico e no formato de *card*, as informações em formato de *card* mostram o estado em tempo real, se está normal ou crítico. Nos gráficos são exibidas informações de temperatura, gases e fumaça nas últimas 24 horas. Além disso, em uma outra aba denominada, log de eventos, o usuário tem acesso às informações detalhadas dos eventos extraídos da base de dados. Todo o sistema pode ser visto na Figura 40.

Figura 40 - Dashboard da interface Web.



Fonte: Autoria própria.

Feita a etapa da calibração, deve-se estabelecer parâmetros para o ambiente que será monitorado. Sabe-se que o sensor MQ2 tem uma concentração mínima de gases inflamáveis de 100ppm's e uma concentração máxima de 10000ppm. Porém, de acordo com seu datasheet, o gás GLP tem uma sensibilidade de detecção de 300ppm, dessa forma será estabelecido cenários de teste para detecção de gases inflamáveis e fumaça em torno de 400ppm. E os testes irão seguir o procedimento das diretivas para a classificação de cenários.

Será considerado um cenário normal ou crítico quando as leituras estiverem entre os seguintes intervalos conforme Figura 42.

Figura 42 - Parâmetros para os cenários dos testes de ambiente.

CENÁRIO DE TESTE - NORMAL - AMBIENTE SEGURO			
SENSORES	NORMAL	SINAL SONORO	INTERFACE WEB
TEMPERATURA	< 40°C	INATIVO	SEGURO
GASES INFLAMÁVEIS	< 400ppm	INATIVO	SEGURO
FUMAÇA	< 500 ppm	INATIVO	SEGURO
CENÁRIO DE TESTE - CRÍTICO - AMBIENTE PERIGO			
SENSORES	CRÍTICO	SINAL SONORO	INTERFACE WEB
TEMPERATURA	> 40°C	ATIVO	PERIGO
GASES INFLAMÁVEIS	> 400ppm	ATIVO	PERIGO
FUMAÇA	> 500 ppm	ATIVO	PERIGO

Fonte: Autoria própria.

3.7.1 Teste de fumaça

Para o cenário de teste do módulo sensor foi usado um recipiente transparente para simular uma câmara de gás. Primeiramente, o módulo sensor é inserido dentro do recipiente fechado, como visto na Figura 43.

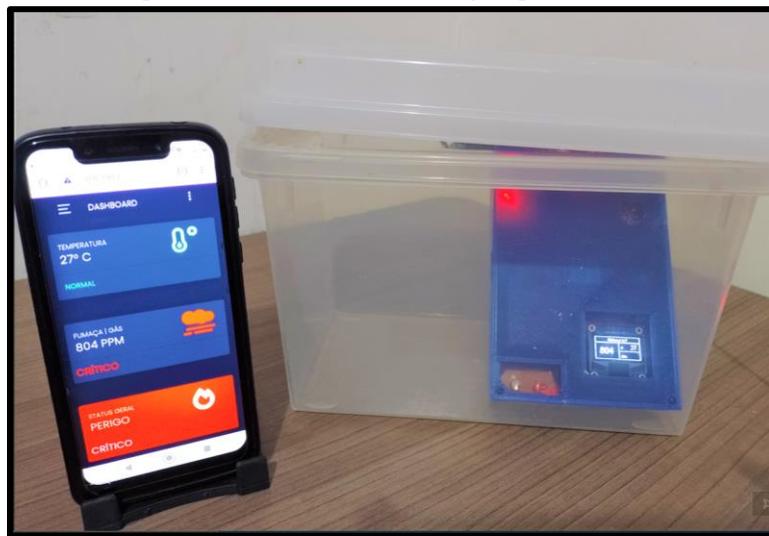
Figura 43 - Recipiente descontaminado representando o status seguro.



Fonte: Autoria própria.

Para alterar as variações do ambiente, foi inserido no recipiente transparente fumaça originada da queima de papel. A aferição do sensor é exibida no *dashboard*, mostrando o *status* crítico do ambiente e quantos ppm's foram detectados. Além disso, também pode ser visualizado em tempo real as informações do *display* do módulo sensor em sincronia com a aplicação web, informando ao usuário o status, perigo, do ambiente. Em vista disso, observou-se que o sensor e a aplicação web atuaram de acordo com os parâmetros estabelecidos, validando o teste de fumaça. Conforme observado na Figura 44.

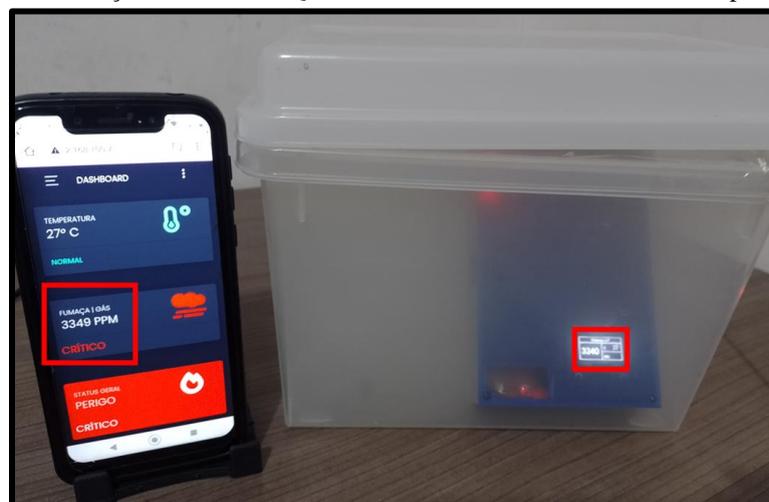
Figura 44 - Recipiente contaminado com fumaça representando o status crítico.



Fonte: Autoria própria

Na figura 45, o recipiente de teste foi totalmente contaminado, chegando a obter dados de saídas de 3349 ppm.

Figura 45 - Aferição do sensor MQ2 no ambiente totalmente contaminado por fumaça.

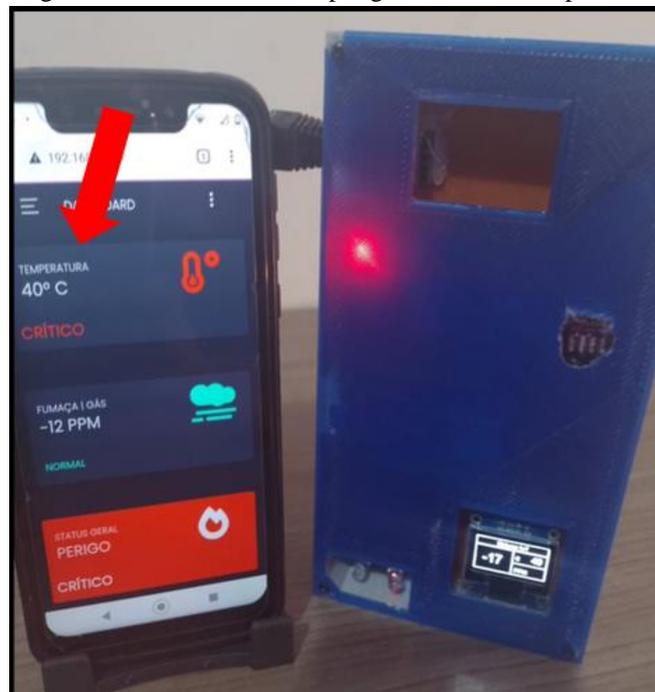


Fonte: Autoria própria.

3.7.2 Teste de temperatura

O procedimento para o teste de temperatura segue os parâmetros adotados no início da seção 3.7. Inicialmente, apresentando um status seguro, ou seja, abaixo de 40°C. Com o objetivo de alterar a temperatura do ambiente, foi utilizado um secador de cabelo em direção ao módulo sensor. Podendo ser observado em tempo real tanto no visor do display como na interface web, o valor crítico coletado é de 40°C, podendo ser visualizado na Figura 46. Conseqüentemente, foi validado o funcionamento do sensor de acordo com os parâmetros adotados.

Figura 46 - Status indicando perigo no teste de temperatura.



Fonte: Autoria própria.

3.7.3 Teste de vazamento de gases inflamáveis

De acordo com o Abdullah (2010), o gás GLP é composto de 70% de butano e 30% de propano, sendo o butano predominante para determinar as características do gás GLP. Em razão disso, o butano é a substância encontrada na maioria dos isqueiros portáteis usados para gerar fogo, tal objeto fará parte do teste de vazamento de gás inflamável. O módulo sensor foi incluído fisicamente no ambiente residencial, conforme visualizado na Figura 47, além disso o sistema deve obedecer a configuração, seguro, do cenário de teste.

Figura 47 - Status seguro do teste de vazamento de gás inflamável.



Fonte: Autoria própria.

Na Figura 48 é possível visualizar o status de perigo na interface web. Como também, o alerta sonoro é ativado conforme os parâmetros indicados para ativação do mesmo. Por fim, os testes de vazamento de gases inflamáveis foram executados em um ambiente controlável. Dessa forma, foi validado o funcionamento do sistema referente ao processo de detecção de vazamento de gases inflamáveis.

Figura 48 - Status crítico dos testes de vazamento de gases inflamáveis.



Fonte: Autoria própria.

3.7.4 Teste de integração do sistema

A fase de validação e integração do módulo sensor e do módulo atuador, tem como objetivo analisar o comportamento de todo o sistema. Na Figura 49, encontram-se as diretrizes para a classificação dos cenários de teste de integração do sistema.

Figura 49 - Diretivas para classificação do teste de integração do sistema.

CENÁRIO DE TESTE - NORMAL - AMBIENTE SEGURO				
SENSORES	NORMAL	SINAL SONORO	INTERFACE WEB	ATUADOR
TEMPERATURA	<40°C	INATIVO	SEGURO	ABERTO
GASES INFLAMÁVEIS	< 400ppm	INATIVO	SEGURO	ABERTO
FUMAÇA	< 500 ppm	INATIVO	SEGURO	ABERTO
CENÁRIO DE TESTE - CRÍTICO - AMBIENTE PERIGO				
SENSORES	CRÍTICO	SINAL SONORO	INTERFACE WEB	ATUADOR
TEMPERATURA	>40°C	ATIVO	PERIGO	FECHADO
GASES INFLAMÁVEIS	> 400ppm	ATIVO	PERIGO	FECHADO
FUMAÇA	> 500 ppm	ATIVO	PERIGO	FECHADO

Fonte: Autoria própria.

No intuito de testar o sistema num cenário de possível percepção de GLP, foi necessário instalar o protótipo em uma cozinha residencial (Figura 50).

Figura 50 - Cenário de teste de integração do protótipo de sistema.



Fonte: Autoria própria.

A princípio, o módulo sensor foi instalado no ambiente residencial e o módulo atuador na válvula reguladora de gás. A Figura 50 apresenta o cenário de teste com o protótipo instalado. Na ilustração é possível observar que ao receber o valor potencialmente perigoso de vazamento de GLP, ele irá tomar algumas medidas de precaução:

- a) o módulo sensor coleta a presença de gás perigoso e imediatamente envia os dados para o módulo atuador ao mesmo tempo ativa a sinalização sonora e o led vermelho, e envia uma mensagem para a interface web, indicando perigo;
- b) o módulo atuador recebe o comando do módulo sensor ativando o relé
- c) válvula reguladora de gás é automaticamente desligada por um solenoide.

Para executar o teste de vazamento de GLP, iniciou-se o processo de vazamento de gás pela saída do acendedor de chamas, o sistema detectou o vazamento na faixa de 200 ppm e o mesmo continuou aumentando até o valor de 800 ppm, indicando uma progressão perigosa no ambiente. Além disso, a interface web envia um alerta de perigo e o alerta sonoro é ativado conforme os parâmetros indicados para ativação do mesmo. Por fim, os testes de vazamento de GLP foram executados em um ambiente controlável. Os resultados obtidos do teste de integração do sistema, serão apresentados no capítulo 4, seção 4.2.

4 RESULTADOS OBTIDOS

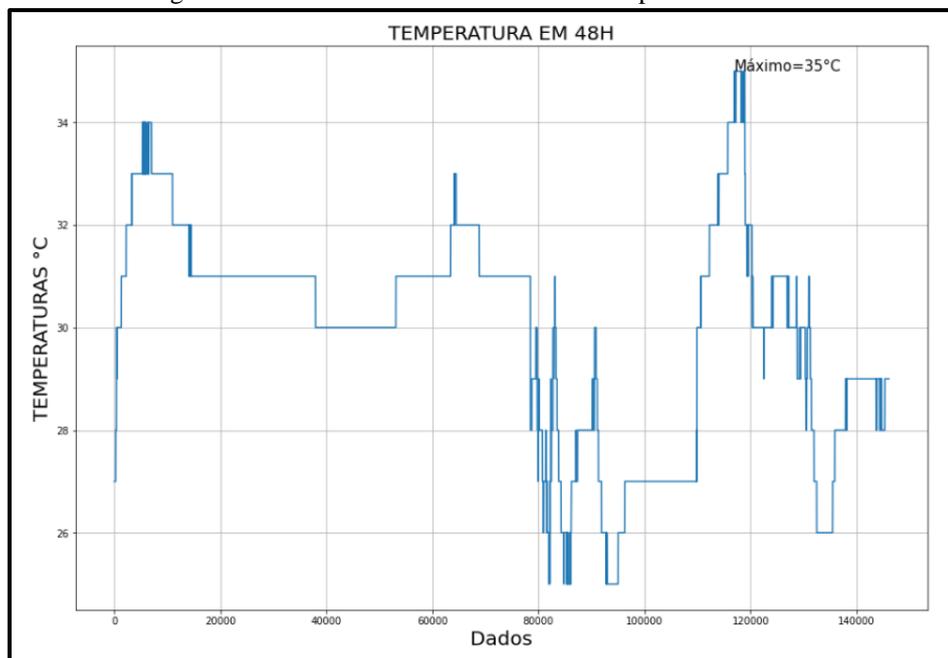
Este capítulo está separado em 2 seções onde são mostrados o desempenho da coleta e recuperação de dados do sistema e os testes de integração do sistema, exibindo o protótipo funcional do módulo sensor e do módulo atuador.

4.1 COLETAS E ANÁLISE DE DADOS DO SISTEMA

No período de 48h, foram coletados 146.125 dados, oriundos do módulo sensor, e com o auxílio do *Matplotlib*, uma biblioteca de *software* livre para plotar gráficos e visualizar dados, em linguagem *python*, foi possível efetuar uma modesta análise das informações da base de dados do sistema em forma de gráficos.

A Figura 51 mostra o gráfico de temperatura, observado a máxima temperatura de 35°C e a mínima temperatura de 25°C. Notou-se o processo de evolução da temperatura no transcorrer do dia.

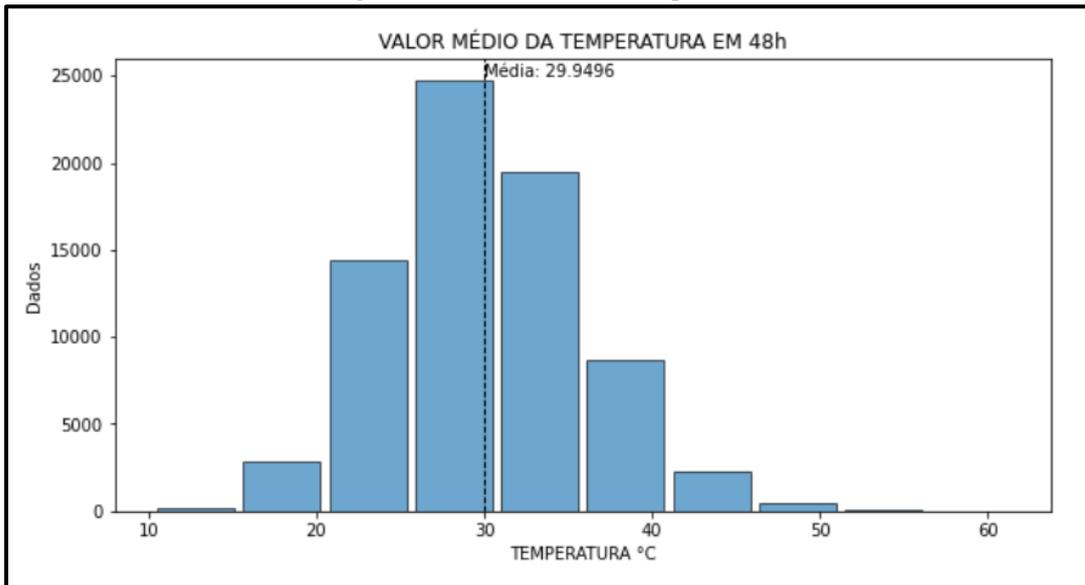
Figura 51 - Gráfico da coleta de dados da temperatura em 48h.



Fonte: Autoria própria.

No gráfico de histograma na Figura 52, verificou-se o valor médio da temperatura de 29.9496°C, foi observado que a temperatura está dentro dos parâmetros estabelecidos para a normalidade do ambiente referentes aos níveis de temperatura do capítulo 3 seções 3.7.4, ou seja, nenhuma ocorrência de eventos, críticos, ocorreram em 48h.

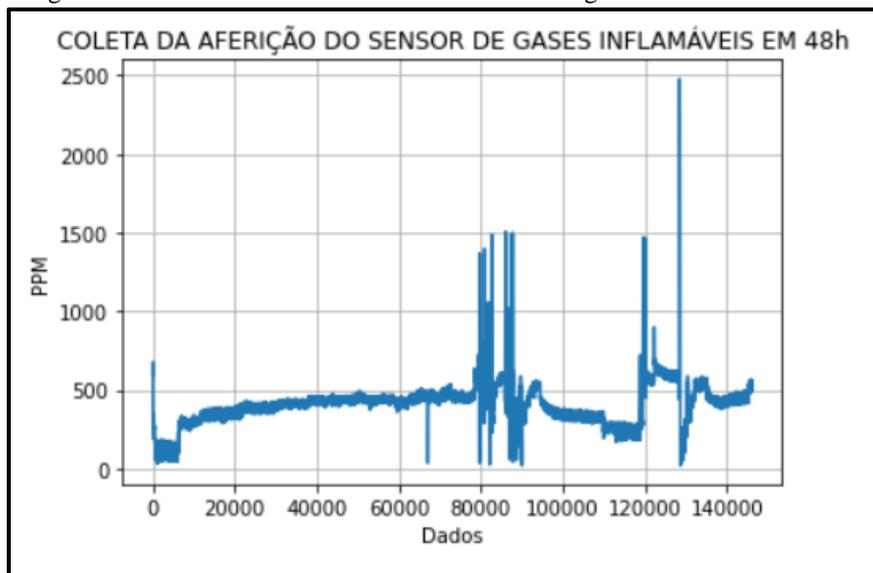
Figura 52 - Valor médio da temperatura.



Fonte: Autoria própria.

Para o sensor de gases inflamáveis e fumaça, a Figura 53 exibe o gráfico de coleta de dados, observado a máxima aferição de 2.478ppm e a mínima aferição de 25ppm. O Gráfico expõe as aferições dos testes realizados nesse período, isso explica porque em alguns pontos é possível identificar um elevado índice de ppm's.

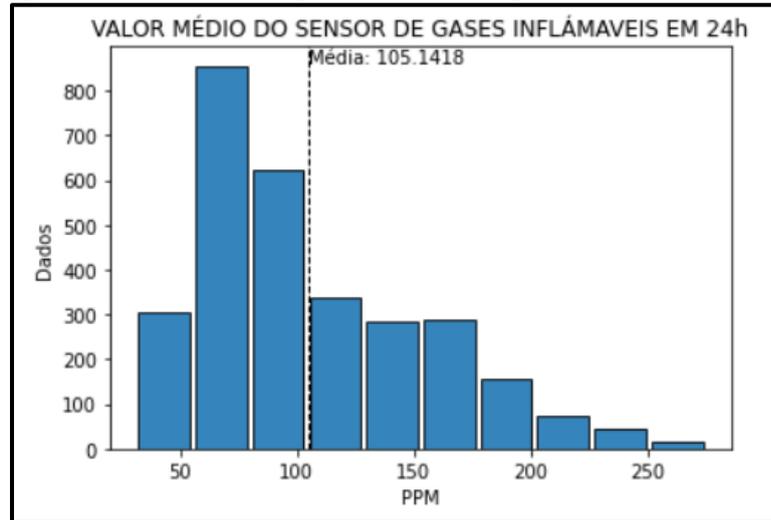
Figura 53 - Gráfico da coleta de dados do sensor de gases inflamáveis em 48h.



Fonte: Autoria própria.

Nas primeiras 24h, o gráfico do histograma da Figura 54, exibe um valor médio de 105ppm, é possível observar que a média está bem abaixo de 200ppm, atendendo aos parâmetros da curva de característica do sensor, fornecidos pelo *datasheet* do sensor MQ2.

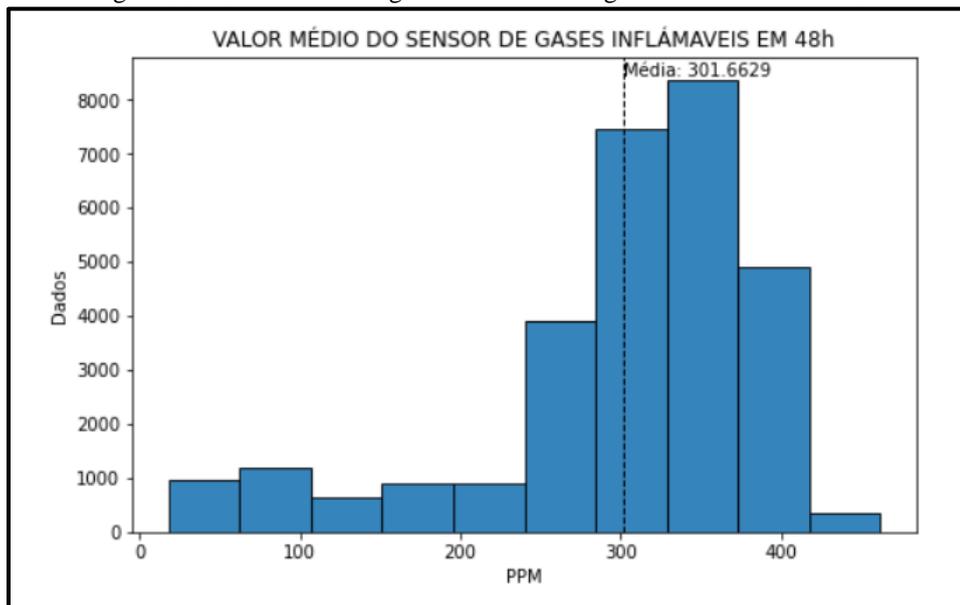
Figura 54 - Gráfico de histograma do sensor de gases inflamáveis em 24h.



Fonte: Autoria própria.

Porém, a média de 48h referente aos dados coletados, incluído as aferições dos testes realizados nesse período, exibindo um resultado de 301.6629ppm, podendo ser observado no gráfico de histograma da Figura 55. Apesar da concentração de alguns valores de teste, o sistema encontra-se dentro dos parâmetros estabelecidos.

Figura 55 - Gráfico de histograma do sensor de gases inflamáveis em 48h.



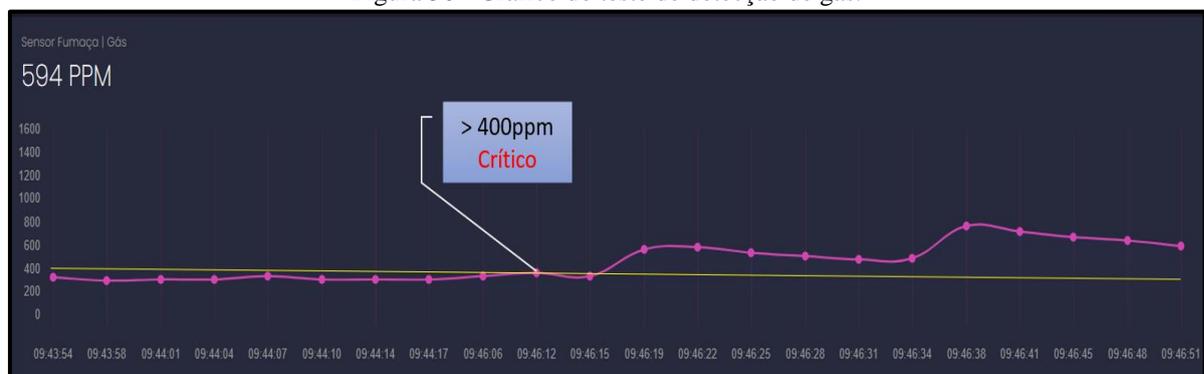
Fonte: Autoria própria.

Atendendo ao propósito de ser um dispositivo IoT o sistema é capaz de coletar, armazenar e enviar os dados continuamente, sem intermitência. O sistema transmite as informações referentes à temperatura, nível de gás inflamável e fumaça do ambiente residencial para o banco de dados, localizado em nuvem, e por meio da Internet, são disponibilizadas aos usuários.

4.2 RESULTADO DO TESTE DE INTEGRAÇÃO DO SISTEMA

Com base nos testes realizados de integração do sistema do protótipo final, identificou-se uma rápida detecção no sistema, podendo ser visualizado no gráfico da aplicação web do sistema (Figura 56). Como o acionamento é imediato a detecção do gás, tornam-se satisfatórios aos requisitos deste protótipo, tendo em vista que o módulo atuador aciona mediante as informações de perigo oriundas do módulo sensor. Também se obteve um resultado satisfatório em relação à sinalização sonora.

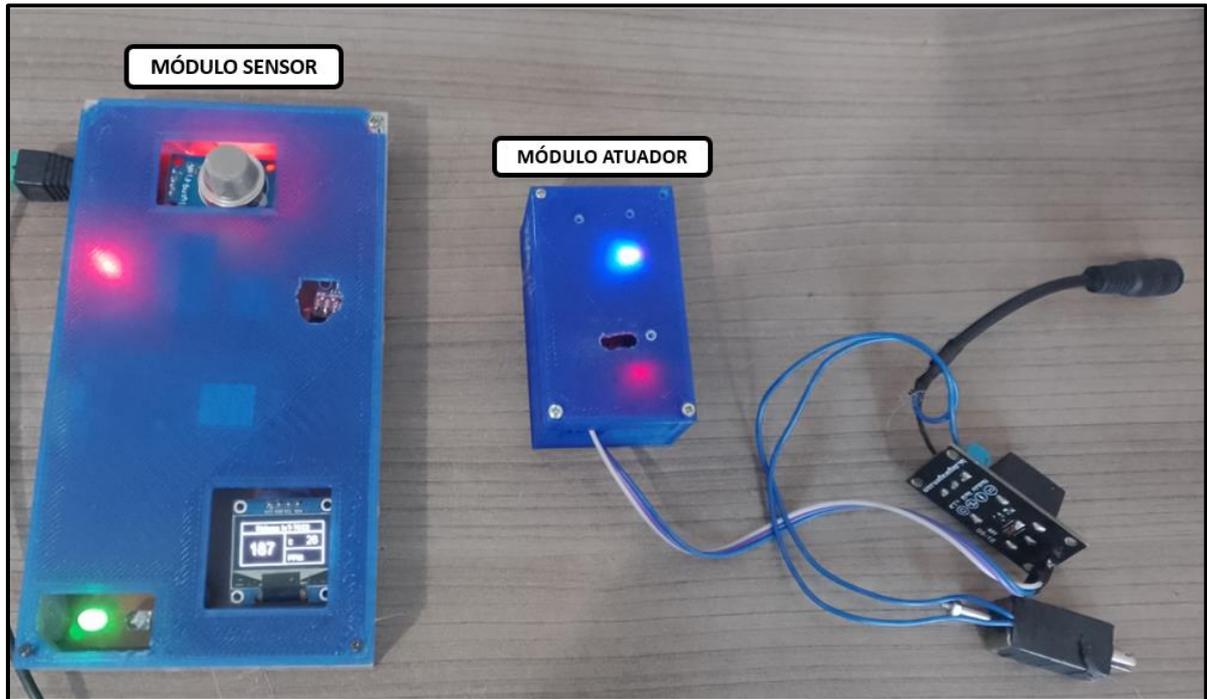
Figura 56 - Gráfico do teste de detecção de gás.



Fonte: Autoria própria.

Finalizados os procedimentos de simulação, montagem e integração do sistema, foi obtida a primeira versão do dispositivo IoT, ilustrada na Figura 57, para monitoramento em tempo real de baixo custo para detecção e prevenção de incêndios em ambientes residenciais. A comunicação entre o módulo sensor e o módulo atuador ocorre por intermédio do protocolo de comunicação UPD. Por fim, o dispositivo foi utilizado em conjunto com interface web para uma melhor interação com o usuário.

Figura 57 - Protótipo do dispositivo para detecção e prevenção de incêndios em ambientes residenciais.



Fonte: Autoria própria.

CONCLUSÃO

Neste trabalho foi desenvolvido um protótipo de dispositivo de baixo custo baseado em *Internet of Things* (IoT) para detecção e prevenção de incêndios em ambientes residenciais. Para tal objetivo, foram realizadas revisões bibliográficas sobre os assuntos referentes aos perigos da exposição do corpo humano ao monóxido de carbono produzido pela fumaça de um incêndio, sensores de temperatura, sensores *chemiresistors* de gases inflamáveis e fumaça. Além disso, foram abordadas pesquisas referentes às principais características do microcontrolador ESP8266, é um estudo detalhado dos protocolos de comunicação HTTP, bem como seus métodos de *GET* e *POST*. Também consta estudos sobre o protocolo UDP, visando a comunicação entre placas da família dos microcontroladores ESP8266 e os conceitos do ecossistema da internet das coisas.

A implementação da arquitetura do projeto foi realizada de forma satisfatória, de acordo com as etapas descritas no capítulo de materiais e métodos. Possibilitando realizar os testes de integração de sistema.

Referente a hipótese citada na introdução, e com base nos resultados obtidos, conclui-se que é possível desenvolver um dispositivo de baixo custo que é capaz de monitorar em tempo real, utilizando um dos múltiplos conceitos da IoT, no ambiente residencial, utilizando uma plataforma de *hardware* denominada NodeMCU ESP12. Foi corroborado, que é possível desligar automaticamente a válvula reguladora de gás, caso seja detectado vazamento ou desvio nos parâmetros monitorados, bem como habilitar o alerta sonoro. No que se refere a visualização dos dados, o dispositivo é capaz de enviar um alerta de perigo para a interface web, e o usuário poderá visualizar o alerta em uma página de *dashboard*.

Para trabalhos futuros sugere-se uma válvula solenoide adequada para válvula reguladora de registro de gás, a fim de obter mais segurança. Outra sugestão é acrescentar um sensor de fumaça, para que as leituras de fumaça e gás sejam identificadas de forma independente, objetivando uma melhor análise. Pode-se ainda, implementar dados de treino do algoritmo de *machine Learning* para criação do modelo que permite melhorar e otimizar o funcionamento do sistema de coleta de dados.

REFERÊNCIAS

ABDULLAH, Huzaifah. **A thesis submitted in fulfillment of the requirements for the award of the Degree of Bachelor of Chemical Engineering (Gas Technology)**. 2010.

Disponível em <

http://umpir.ump.edu.my/id/eprint/3289/1/CD5507_HUZAIFAH_ABDULLAH.pdf >

Acesso em 20 nov. 2021.

ALBERTO, Carlos. **Projetos com o Esp8266 para você estudar e praticar com IOT**.

1.ed. [s.l.]. CapSistema. 2021. E-book (3 p.). Disponível em: <

<https://play.google.com/books/reader?id=yz4oEAAAQBAJ&pg=GBS.PT3> >

Acesso em: 01 jul. 2021.

ASHTON, Kevin. **That ‘Internet of Things’ thing. Publicado no RFID Journal, 2009.**

Disponível em:< [https://www.itrco.jp/libraries/RFIDjournal-](https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf)

[That%20Internet%20of%20Things%20Thing.pdf](https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf)

>. Acesso em: 03 jun. 2021.

BOSCH. **BME280 Datasheet**. 2020. Disponível em <

<https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf> > Acesso em 13 out. 2021.

CARLO, U. et al. **A Segurança Contra Incêndio no Brasil**. 1ª ed. São Paulo: Projeto Editora, 2008. p. 65–71.

ESPRESSIF. **ESP8266EX Datasheet**. 2020. Disponível em<

https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>

Acesso em 13 Out. 2021. p. 65–71.

ESTEVAM JUNIOR, Sergio Luiz; FARINELLI, Felipe Adalberto. **Demótica: Automação Residencial e Casas Inteligentes com Arduino e ESP8266**. 1ª ed. São Paulo: Érica, 2019.

FELIPEFLOP. **MQ2 Datasheet**. [s.d.]. (2 p.). Disponível em<

https://www.filipeflop.com/img/files/download/Datasheet_Sensor_Gas_MQ2.pdf >

Acesso em 13 Out. 2021

FREERTOS. **What is An RTOS?**.2021. Disponível em <

<https://www.freertos.org/about-RTOS.html> > Acesso em 13 out. 2021

G1. **Residências concentram maioria de incêndios em Manaus, diz Corpo de Bombeiros**.

2020. Disponível em:< <https://g1.globo.com/am/amazonas/noticia/2020/03/24/residencias-concentram-maioria-de-incendios-em-manaus-diz-corpo-de-bombeiros.ghtml>>. Acesso em:

26 mai. 2021

MOZILLA. **Uma visão geral do HTTP**. 2021. Disponível em <

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview> > Acesso em 13 nov. 2021

OLIVEIRA, Sérgio de. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. 2.ed. São Paulo: Novatec. 2021. *E-book*. ISBN 978-65-86057-36-2. Disponível em: < https://www.amazon.com.br/Internet-Coisas-ESP8266-Arduino-Raspberry-ebook/dp/B08Z34JZC6/ref=sr_1_2?__mk_pt_BR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=Internet+das+Coisas+com+ESP8266%2C+Arduino+e+Raspberry+Pi&qid=1625233954&sr=8-2 >. Acesso em: 01 jul. 2021.

POSTEL, John. **User Datagram Protocol**. 1980. Disponível em < <https://dl.acm.org/doi/pdf/10.17487/RFC0768> > Acesso em 13 out. 2021.

SINDIGAS. **Evite os riscos do gás de cozinha (GLP)**. 2021. Disponível em < <https://www.sindigas.org.br/novosite/?p=14206> > Acesso em 13 out. 2021.

SPARKFUN. **Bi-Directional Logic Level Converter Hookup Guide**. 2015. Disponível em < https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/Bi-Directional-Logic-Level_HookupGuide.pdf > Acesso em 20 nov. 2021.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro. **Sensores Industriais - Fundamentos e Aplicações**. 8ª ed. São Paulo: Érica, 2009.

TEXAS. **ADS1115 Datasheet**. 2018. Disponível em < https://www.ti.com/lit/ds/symlink/ads1113.pdf?ts=1634086440228&ref_url=https%253A%252F%252Fwww.google.com.br%252F > Acesso em 13 out. 2021.

APÊNDICE A - CÓDIGO DO MÓDULO SENSOR

```

*****
AUTORA: MÁRCIA DOS SANTOS SE SÁ
UNIVERSIDADE DOS ESTADO DO AMAZONAS
NODEMCU-ESP12 - MÓDULO SENSOR
PROTOCOLOS: HTTP|UDP
*****

/* ----- BIBLIOTECAS ----- */

#include <FS.h>
#include <Wire.h>
#include "Thread.h"
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include "ThreadController.h"
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <Adafruit_ADS1X15.h>
#include "SSD1306Wire.h"
#include <WiFiUdp.h>
#include <math.h>
/*-----* /

#ifndef SPN
#define SPN(M) if(Serial)Serial.println(M)
#endif
#ifndef SPT
#define SPT(M) if(Serial)Serial.print(M)
#endif
/*-----Pinos do módulo NodeMCU-ESP12----- */

/*****| GPIO          | PINOS          | *****/
#define SDA 4          // D2 I2C
#define SCL 5          // D1 I2C
#define LED_R 13       // D7 LED VERMELHO
#define LED_G 15       // D8 LED VERDE
#define buzzer 12      // D5 BUZZER

/*-----WIFI----- */
#define SSID "MDS2GHZ"
#define SSID_PASSWORD "Linkdell072"
#define SERVER_IP "192.168.100.7" // HTTP

/*-----SENSORES-----*/
Adafruit_BME280 bme; // I2C
Adafruit_ADS1115 ads; /*16-bit version */
SSD1306Wire display(0x3c, SDA, SCL);

```

```

/*----- PROTOCOLO-UDP----- */
WiFiUDP Udp;
unsigned int localPort = 8888;
IPAddress remote_IP(192, 168, 100, 29); //ip ESP01
/*----- INICIALIZADO A THREAD ----- */
ThreadController cpu;
Thread Sensores;
int t;
int ads0;
/*-----Cálculo do ADC ----- */
int readMQx(uint8_t pin)
{
    float v = ReadVoltage(pin);
    return ceil(10000 * (v / 5.0));
}
double ReadVoltage(byte pin)
{
    float adc = ads.readADC_SingleEnded(pin);
    float v = adc * 0.1875e-3;
    return v;
}
/*----- */
void LeituraSensor()
{
    ads0 = readMQx(1);
    t = bme.readTemperature(); // Le a temperatura
}
/*----- */
Thread ControlSensor;
void controll()
{
    if ((t >= 40) || (ads0 >= 400))
    {

        digitalWrite(LED_R, HIGH);
        digitalWrite(LED_G, LOW);
        tone(buzzer, 1350);
        delay(150);
        tone(buzzer, 1050);
        delay(150);

        // Send message
        Udp.beginPacket(remote_IP, localPort);
        Udp.write("m");
        Udp.endPacket();
        SPN("DADOS ENVIADOS PARA O MÓDULO ATUADOR");
    }
    else
    {
        // noTone(buzzer);
    }
}

```

```

        digitalWrite(LED_R, LOW);
        digitalWrite(LED_G, HIGH);
    }
}

Thread MonitorSerial;
void mSerial()
{
    if ((WiFi.status() == WL_CONNECTED)) {

        WiFiClient client;
        HTTPClient http;
        SPT("[HTTP] begin...\n");
        http.begin(client, "http://" SERVER_IP "/leitura"); //HTTP-
rota
        SPT("[HTTP] POST...\n");
        int httpCode = http.POST("{\"valor\":" + String(t) + ",
\"id_sensor\":2}");

        if (httpCode > 0) {
            Serial.printf("[HTTP] GET... code: %d\n", httpCode);

            if (httpCode == HTTP_CODE_OK) {
                const String& payload = http.getString();
                SPT("received payload:\n<<");
                SPT(payload);
                SPT(">>");
            }
            else {
                Serial.printf("[HTTP] POST... failed, error: %s\n",
http.errorToString(httpCode).c_str());
            }
            http.end();

            http.begin(client, "http://" SERVER_IP "/leitura");
            int httpCode2 = http.POST("{\"valor\":" + String(ads0) + ",
\"id_sensor\":1}");
            if (httpCode2 == HTTP_CODE_OK) {
                const String& payload2 = http.getString();
                SPT("received payload:\n<<");
                SPT(payload2);
                SPT(">>");
            }

            http.end();
        }
        SPT("Temperatura: ");
        SPT(t);
        SPT(",");
        SPN("°C ");
    }
}

```

```

    SPT("GÁS: ");
    SPT(ads0);
    SPN(" ppm ");
    //Serial.print("Sending Data");
}
void setup()
{
    /*-----
    ##### Inicialização -- begin #####
    -----*/
    Serial.begin(115200);

    // Set device as a Wi-Fi Station
    WiFi.mode(WIFI_STA);
    WiFi.begin(SSID, SSID_PASSWORD);

    if ((WiFi.status() == WL_CONNECTED)) {
        WiFiClient client;
        HTTPClient http;
    }
    /*-----*/

    display.init(); /*Inicializar a tela */
    display.flipScreenVertically();

    unsigned status; /*inicialização BME */
    status = bme.begin(0x76, &Wire);
    ads.begin(); /*inicialização o ADS */
    Udp.begin(localPort);
    /*----- */
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(buzzer, OUTPUT);
    /*-----*/
    MonitorSerial.setInterval(3000);
    MonitorSerial.onRun(mSerial);

    Sensores.setInterval(1000);
    Sensores.onRun(LeituraSensor);

    ControlSensor.setInterval(1000);
    ControlSensor.onRun(controll);
    /*----- */
    cpu.add(&Sensores);
    cpu.add(&MonitorSerial);
    cpu.add(&ControlSensor);
}
/*----- */
void loop()

```

```

{
  cpu.run();
  telainicial();
  //thing.handle();
  //thing.stream(thing["data"]);
}

/**|-----|
    |           DISPLAY OLED CONFIG           |
    |-----| */

void telainicial()
{
  /*-----Apaga o display-----*/
  display.clear();
  // Desenha as molduras
  display.drawRect(0, 0, 128, 16);
  display.drawRect(0, 16, 128, 48);
  display.drawLine(67, 16, 67, 63);
  display.drawLine(67, 40, 128, 40);
  // Atualiza informacoes da temperatura
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_CENTER);
  display.drawString(64, 2, "Sistema IoT-TCC2");
  display.setFont(ArialMT_Plain_24);
  display.drawString(32, 26, String(ads0)); // MQ2
  // display.drawCircle(52, 32, 2);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_10);
  display.drawString(73, 24, "C");
  display.setFont(ArialMT_Plain_16);
  display.drawString(101, 19, String(t)); // temperatura
  display.setFont(ArialMT_Plain_10);
  display.drawString(73, 48, "PPM");
  display.setFont(ArialMT_Plain_16);
  display.drawString(101, 43, String()); // MQ Alguma coisa
  Aee!!

  display.display();
}

```