

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO ENGENHARIA ELETRÔNICA

ALEXANDRE PEDRO SANTANA
ALINE NUNES SCREMIN
WILLIAN PASQUETI DE ALMEIDA

**AWA – SISTEMA DE MONITORAMENTO RESIDENCIAL DE GLP COM
ALERTA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2021

ALEXANDRE PEDRO SANTANA
ALINE NUNES SCREMIN
WILLIAN PASQUETI DE ALMEIDA

**AWA – SISTEMA DE MONITORAMENTO RESIDENCIAL DE GLP COM
ALERTA**

Trabalho de conclusão de curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso II, do Curso Engenharia Eletrônica do Departamento Acadêmico de Eletrônica da Universidade Tecnológica Federal do Paraná – UTFPR, Campus Curitiba, como requisito parcial para a obtenção do título.

Orientadora: Prof^a. Dr^a. Keiko Veronica Ono Fonseca

CURITIBA
2021

ALEXANDRE PEDRO SANTANA
ALINE NUNES SCREMIN
WILLIAN PASQUETI DE ALMEIDA

AWA – SISTEMA DE MONITORAMENTO RESIDENCIAL DE GLP COM ALERTA

Este Trabalho de Conclusão de Curso de Graduação foi apresentado como requisito parcial para obtenção do título de Engenheiro Eletrônico, do curso de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica (DAELN) outorgado pela Universidade Tecnológica Federal do Paraná (UTFPR). Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Curitiba, 26 de abril de 2021.

Prof. Dr. Robinson Vida Noronha
Coordenador de Curso
Engenharia Eletrônica

Prof^a. Dr^a. Carmen Caroline Rasera
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia Eletrônica do DAELN

BANCA EXAMINADORA

Prof^a. Dr^a. Keiko Veronica Ono Fonseca
Universidade Tecnológica Federal do Paraná
Orientador

Prof. Dr. Daniel Fernando Pigatto
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luciano Scandelari
Universidade Tecnológica Federal do Paraná

*Aos nossos pais, amigos e a todos aqueles
que estiveram nos apoiando nesta jornada.*

AGRADECIMENTOS

Agradecemos primeiramente a Deus pela oportunidade de estudar em uma universidade renomada e contribuir para o desenvolvimento tecnológico da nação.

Agradecemos a nossas famílias e amigos, pelo apoio e incentivo tanto durante a realização da monografia quanto o decorrer de todo curso.

E por fim, agradecemos aos professores do Departamento de Eletrônica, em especial à nossa orientadora Prof^a. Dr^a. Keiko pelo apoio e a passagem de conhecimento, necessária à nossa formação profissional.

RESUMO

SANTANA, A. P. SCREMIN, A. N. ALMEIDA, W. P. **AWA – SISTEMA DE MONITORAMENTO RESIDENCIAL DE GLP COM ALERTA.** 2021 98f. Trabalho de Conclusão de Curso. (Engenharia Eletrônica), Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

A Internet das Coisas (IoT) permite a comunicação de máquinas com máquinas, como um novo modelo de internet. A partir disso, este trabalho apresenta uma solução que emprega a IoT no desenvolvimento de uma plataforma para auxiliar no controle e monitoramento do consumo do gás (GLP), além de emitir um alerta em caso de detecção de vazamentos por meio de um aplicativo. Este projeto apresenta um hardware e uma plataforma de software, denominada AWA, composta por microcontrolador ESP32, responsável pela comunicação, sensores (Célula de carga e MQ-6, para o monitoramento do peso do botijão e a detecção de vazamentos), e o atuador HX711, que amplifica o sinal proveniente dos sensores, transmitindo para o microcontrolador. Essas informações coletadas do AWA são enviadas pelo microcontrolador para um aplicativo executado em um smartphone, o qual disponibiliza os dados ao usuário. Dessa forma, o consumo diário de GLP e o alerta de possíveis vazamentos pode possibilitar ao consumidor um maior controle de gastos e ainda, o mais importante, contribuir para evitar possíveis acidentes envolvendo botijão de gás.

Palavras-chave: *IoT, Balança Digital, comunicação, microcontrolador, consumo de GLP, vazamento de gás.*

ABSTRACT

SANTANA, A. P.; SCREMIN, A. N. ALMEIDA, W. P. **AWA – RESIDENTIAL BUTTER LGP MONITORING SYSTEM WITH ALERT.** 2021. 98f. Term paper. (Electronic Engineering), Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

The Internet of Things (IoT) allows the communication of machines with machines, like a new internet model. From this, this work presents a solution that employs IoT in the development of a platform to assist in the control and monitoring of gas consumption (LPG), in addition to issuing an alert in case of leak detection through an application. This project features a hardware and a software platform, called AWA, composed of ESP32 microcontroller, responsible for communication, sensors (load cell and MQ-6, for monitoring the weight of the cylinder and detecting leaks), and the actuator HX711, which amplifies the signal from the sensors, transmitting to the microcontroller. This information collected from AWA is sent by the microcontroller to a mobile application which makes the data available to the user. Thus, the daily consumption of LPG and the alert of possible leaks can enable the consumer to control their expenses with gas and, most importantly, to possibly avoid accidents involving gas cylinders.

Keywords: *IoT, Digital Scale, communication, microcontroller, LPG consumption, gas leak.*

LISTA DE FIGURAS

Figura 1. Dispositivos IoT x População.	13
Figura 2. Projeto AWA.	16
Figura 3. Integração idealizada dos componentes.	17
Figura 4. Relação dos componentes do sistema para o correto funcionamento do equipamento.	18
Figura 5. Protótipo do hardware de Medeiros e Santos.	21
Figura 6. Modelos de botijões de gás.	24
Figura 7. Célula de carga.	27
Figura 8. Esquemático das células de carga com módulo HX711.	28
Figura 9. Equação da ponte de Wheatstone.	29
Figura 10. Módulo HX711.	30
Figura 11. Sensor de gás MQ-6.	32
Figura 12. Microcontrolador ESP32 e sua pinagem.	33
Figura 13. Comparativo entre ESP32, ESP8266 e Arduino UNO R3.	34
Figura 14. Exemplo de um diagrama de bloco no contexto de IoT.	35
Figura 15. Arquitetura TCP/IP.	37
Figura 16. Visão geral da arquitetura do Firebase Cloud Messaging.	42
Figura 17. Representação do uso de Firebase.	43
Figura 18. Balança marca Wincy Casa.	45
Figura 19. Medição do multímetro da corrente de entrada.	47
Figura 20. Esquemático do circuito de entrada.	50
Figura 21. Circuito para o microcontrolador.	51
Figura 22. Circuito do microcontrolador e seus periféricos.	51
Figura 23. Circuito utilizado para HX711 e periféricos.	53
Figura 24. Circuito sugerido pelo datasheet.	53

Figura 25. Roteamento da placa.	54
Figura 26. Posicionamento dos componentes.....	56
Figura 27. Entrada da placa no forno de refusão.	56
Figura 28. Forno de refusão.....	57
Figura 29. Inspeção visual da placa.	57
Figura 30. Visão macro de funcionamento do sistema AWA.	59
Figura 31. Funcionamento do Setup do firmware.....	60
Figura 32. Funcionamento do Loop do firmware.Fonte: Autoria própria.	61
Figura 33. Mapa de navegação do aplicativo.	63
Figura 34. Activity_main.xml.....	64
Figura 35. Classe slide.java.	66
Figura 36. Código executado quando o button ‘SALVAR DADOS’ é pressionado.	67
Figura 37. Recebimento dos dados da activity ajuste, armazenamento de dados e recuperação.	69
Figura 38. Recuperação do Token e envio para o Firebase.	70
Figura 39. Ícone de Botijão de Gás para instruções de como se conectar à rede Wi-fi.....	71
Figura 40. Passo a passo para a conexão do aplicativo no Wi-fi.	72
Figura 41. Inclusão de dados no aplicativo.	73
Figura 42. Mensagem de erro devido a não conectividade com a internet.	74
Figura 43. Mensagem informativa para colocar o botijão em cima da balança.75	
Figura 44. Botão iniciar ativado na tela.	76
Figura 45. Mensagem de alerta de vazamento de gás detectado pelo sensor.77	
Figura 46. Teste da balança AWA com pesagens a partir de 13kg.....	81
Figura 47. Representação gráfica do consumo de gás (Teste com 13kg inicial).	82

Figura 48. Representação gráfica de vazamento de gás com um isqueiro próximo ao sensor.....	83
Figura 49. Representação gráfica de vazamento de gás da boca do fogão próximo ao sensor.....	84
Figura 50. Aviso de alerta de vazamento de gás no aplicativo.....	85

LISTA DE TABELAS

Tabela 01. Comparativo da AWA com os trabalhos relacionados.....	21
Tabela 02. Número de registros de vazamentos com GLP no estado de SP....	26
Tabela 03. Sensores da série MQ e suas funções.....	31
Tabela 04. Especificações da Balança Wincy casa.....	46
Tabela 05. Afastamento do recipiente de GLP para redes elétricas.....	46
Tabela 06. Tabelas de fios AWG.....	48
Tabela 07. Largura mínima da trilha de acordo com a corrente.....	55
Tabela 08. Critérios de diretrizes de qualidade.....	78
Tabela 09. Comparação de pesagens entre a AWA e balança convencional.....	80
Tabela 10. Comparação de pesagens de um botijão de gás completa entre a AWA e balança convencional.....	80

SUMÁRIO

1	INTRODUÇÃO	13
1.1	IDENTIFICAÇÃO DO PROBLEMA	14
1.2	SISTEMA IDEALIZADO	15
1.3.	PREMISSAS DO PROJETO	18
1.4	OBJETIVOS	19
1.4.1	OBJETIVO GERAL	19
1.4.2	OBJETIVOS ESPECÍFICOS	19
1.5	ESTRUTURA DO TRABALHO	19
1.6	TRABALHOS RELACIONADOS	20
2	REVISÃO BIBLIOGRÁFICA	22
2.1	GÁS LIQUEFEITO DE PETRÓLEO (GLP)	22
2.1.1	PERIGOS DO GLP	25
2.2	CÉLULA DE CARGA	26
2.2.1	PONTE DE WHEATSTONE	28
2.3	MÓDULO HX711	30
2.4	SENSORES MEDIDORES DE GÁS SÉRIE MQ	31
2.5	MICROCONTROLADOR	32
2.5.1	MICROCONTROLADOR ESP32	33
2.6	INTERNET DAS COISAS (IoT)	35
2.6.1	ARQUITETURA DA IOT	35
2.7	PROTOCOLOS DE COMUNICAÇÃO IOT	37
2.7.1	FIREBASE	40
2.7.1.1	Firestore Realtime Database	41
2.7.1.2	Firestore Cloud Messaging	42
2.7.2	FIREBASE x MQTT	43

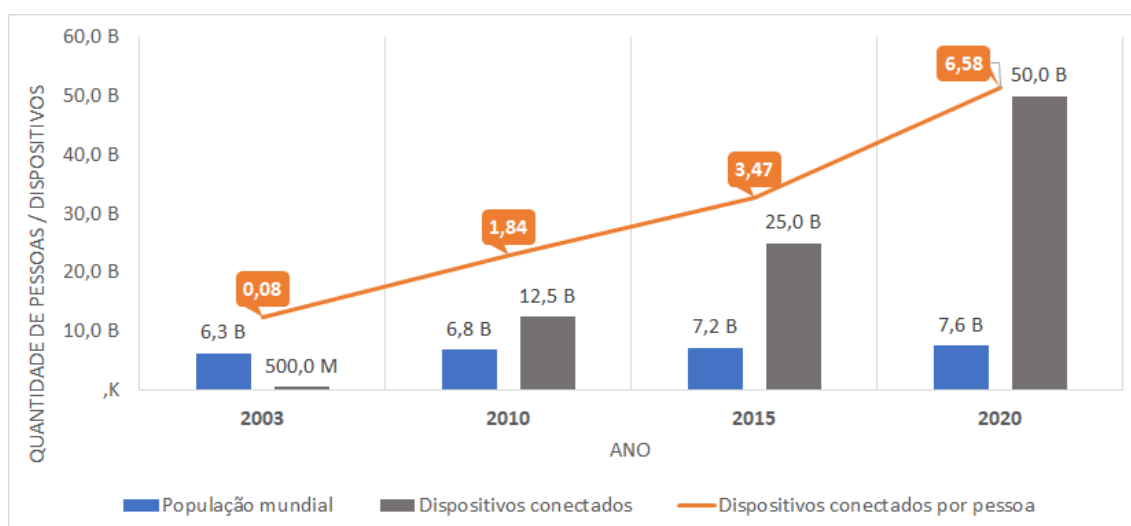
3	MATERIAIS E MÉTODOS	45
3.1	HARDWARE	45
3.1.1	ESTRUTURA DA BALANÇA	45
3.1.2	FONTE DE ENERGIA	46
3.1.3	ESQUEMÁTICO DO PROJETO	49
3.1.3.1	Circuito de entrada	49
3.1.3.2	Circuito para o microcontrolador	50
3.1.3.3	Circuito para o HX711	52
3.1.3.4	Roteamento	54
3.1.4	FABRICAÇÃO E MONTAGEM DA PLACA	55
3.1.5	FIRMWARE - ESP 32	58
3.2	APLICATIVO	61
3.2.1	DESENVOLVIMENTO DO APLICATIVO	61
3.2.1.1	Gráfico	64
3.2.1.2	Configuração Wi-fi	65
3.2.1.3	Ajuste	66
3.2.1.4	Tela principal	68
3.2.1.5	Notificações	70
3.2.2	FUNCIONAMENTO DO APLICATIVO	70
3.2.3	DIRETRIZES DE QUALIDADE	77
4	RESULTADOS E DISCUSSÕES	80
4.1	CALIBRAÇÃO DA AWA	80
4.2	TESTES DE INTEGRAÇÃO AWA X APLICATIVO	81
4.3	TESTE DO SENSOR DE GÁS	82
5	CONCLUSÕES	86
6	REFERÊNCIAS BIBLIOGRÁFICAS	88

1 INTRODUÇÃO

Atualmente a Internet virou onipresente, abrangendo quase todos os cantos do mundo e está presente na vida humana de maneiras inimagináveis. Em consequência disso surgiu o conceito da Internet das Coisas (do inglês *Internet of Things* - IoT), sendo muito utilizado em carros, casas, geoprocessamento, entre outros (SILVA, 2017).

Segundo PATACA (2020), cerca de 50 bilhões de dispositivos estão conectados à rede, totalizando uma taxa de 6,8 dispositivos por pessoa. A figura 1 demonstra o crescimento de dispositivos IoT x população dos últimos anos.

Figura 1. Dispositivos IoT x População.



Fonte: SINGH; SINGH, 2015.

O principal objetivo do IoT é tornar objetos “inteligentes”, sendo assim monitorados e/ou controlados à distância, através de um smartphone ou outro dispositivo conectado à rede, fazendo com que esses objetos desempenhem tarefas e compartilhem informações (SINGH; SINGH, 2015).

No contexto de simplificar o cotidiano das pessoas, a proposta do presente trabalho é apresentar uma solução de monitoramento do consumo de um botijão de gás liquefeito de petróleo (GLP) com base em dados fornecidos

pelo usuário, como por exemplo, o tipo do botijão (P13 e P45), a tara do botijão, através da utilização de sensores de carga acoplados em uma balança.

Além da gestão automatizada de nível de gás, também será apresentado um sistema de alerta que identifica possíveis vazamentos. A interface máquina-homem será realizada através do smartphone por meio de um aplicativo Android, onde o cliente configura a conexão WiFi e as características do seu botijão, como tipo (P13 ou P45) e tara do vasilhame e, acessa os dados que permitem estimar o percentual de GLP restante do botijão em uso em sua residência (consumo, volume disponível, possíveis vazamentos) evitando que o mesmo fique sem gás (GLP) para as suas tarefas diárias.

1.1 IDENTIFICAÇÃO DO PROBLEMA

É provável que poucos brasileiros tenham a real dimensão da importância social e econômica do Gás Liquefeito de Petróleo para o país. Hoje, o GLP, mais conhecido como “gás de cozinha”, tem maior penetração em território nacional do que a energia elétrica, a água encanada e a coleta de esgoto. São mais de 42 milhões de residências – ou 95% do total de domicílios do Brasil – e mais de 150 mil empresas regularmente atendidas por uma ampla e eficiente rede de distribuição do GLP, que está presente em 100% dos municípios brasileiros. Trata-se de um setor vital para a sociedade e que, portanto, deve estar ao alcance, sempre, de toda a população (SINDIGÁS, 2015).

Focando na utilização residencial do GLP, um problema identificado na rotina dos brasileiros foi a dificuldade de mensurar o volume de combustível restante no botijão. Existem alguns métodos para tentar prever o final do gás, como levantar o botijão de gás para estimar o peso, pois quando está próximo do fim, o estado líquido do gás muda para gasoso, tornando-o mais leve que o peso inicial. Outro método conhecido é observar a cor da chama do fogão que altera do azul intenso, quando o botijão está cheio, para um tom avermelhado quando está no seu fim, porém esses métodos populares não são muito precisos em relação à quantidade de GLP no recipiente (SINDIGÁS, 2015).

Outro problema identificado em residências foi a possibilidade de vazamentos no momento da substituição do botijão vazio por um cheio ou também, devido às más condições ou prazo de validade vencidos tanto da mangueira quanto do regulador que possuem validade de 5 anos e devem ter a certificação do INMETRO. Conforme indicado pelo CORPO DE BOMBEIROS DO PARANÁ (2021), para verificar vazamentos durante a instalação do botijão, normalmente se aplica uma espuma de sabão sobre a válvula, se a espuma borbulhar é porque há vazamento.

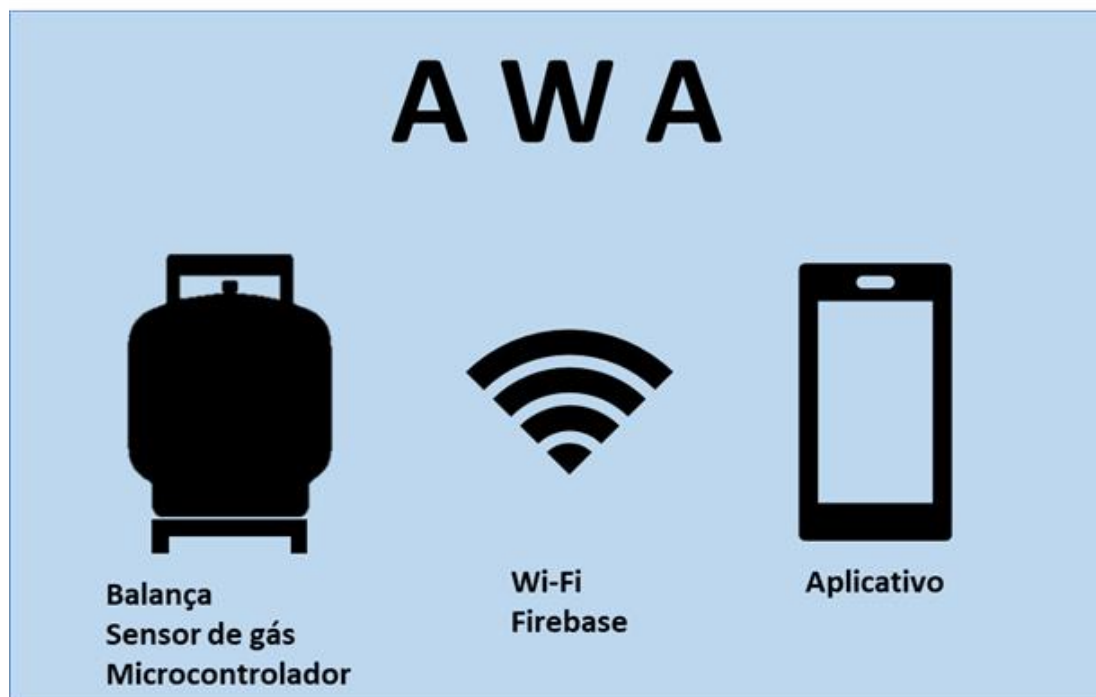
O vazamento de gás também pode ser identificado através do seu forte odor, devido ao aditivo Mercaptano que é inserido nos gases de cozinha para alertar as pessoas do risco decorrente de vazamentos, pois tanto o propano quanto o butano são inodoros (SINDIGÁS, 2015).

A incerteza e a insegurança geradas pelos fatores acima citados motivaram a necessidade de desenvolver uma alternativa para solucionar estes problemas. Sendo assim, este trabalho de conclusão de curso apresenta uma proposta de solução baseada no monitoramento e estimativa do volume de GLP disponível no botijão através do seu peso e a emissão de um alerta em casos de vazamentos. A solução proposta faz uso de tecnologias IoT.

1.2 SISTEMA IDEALIZADO

O sistema idealizado possui o intuito de simplificar o cotidiano das pessoas que utilizam botijão de gás. Denominado AWA, o projeto monitora o peso do botijão de gás, indicando volume de gás e gerando um alerta de volume estimado menor que 10% e se há vazamentos, pode ser dividido em 2 partes, o conjunto de hardware e firmware constituídos pela balança e pelo ESP 32, e o aplicativo Android, conforme demonstrado na figura 2.

Figura 2. Projeto AWA.

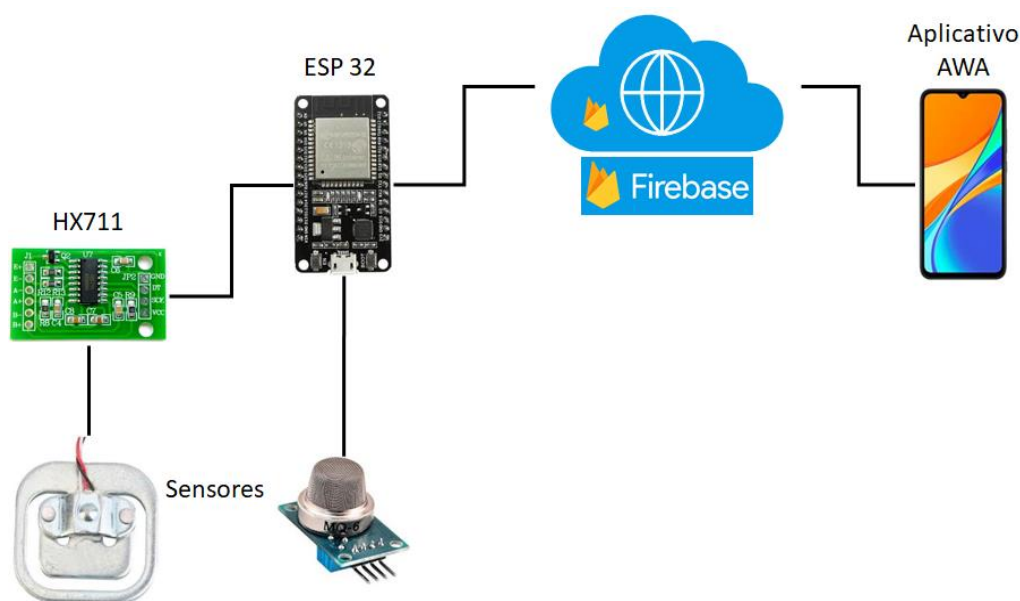


Fonte: Autoria própria.

Para dar início às medições, o usuário precisará fornecer, através do aplicativo, as características do botijão de gás, tara e tipo do botijão, que pode variar de acordo com os tamanhos. Em residências os tipos encontrados são P13, o mais comum, e P45. Após o *input* dessas informações, é necessário ajustar o botijão sobre a AWA e, por meio de um botão no aplicativo, as pesagens serão iniciadas.

O microcontrolador juntamente com os sensores, as células de carga e o sensor de vazamento de gás, fará a captação dos dados de peso (balança) e dados de vazamento de gás (sensor) e transmitirá informações da porcentagem restante e se há vazamentos para a base de dados, representado na figura 3.

Figura 3. Integração idealizada dos componentes.



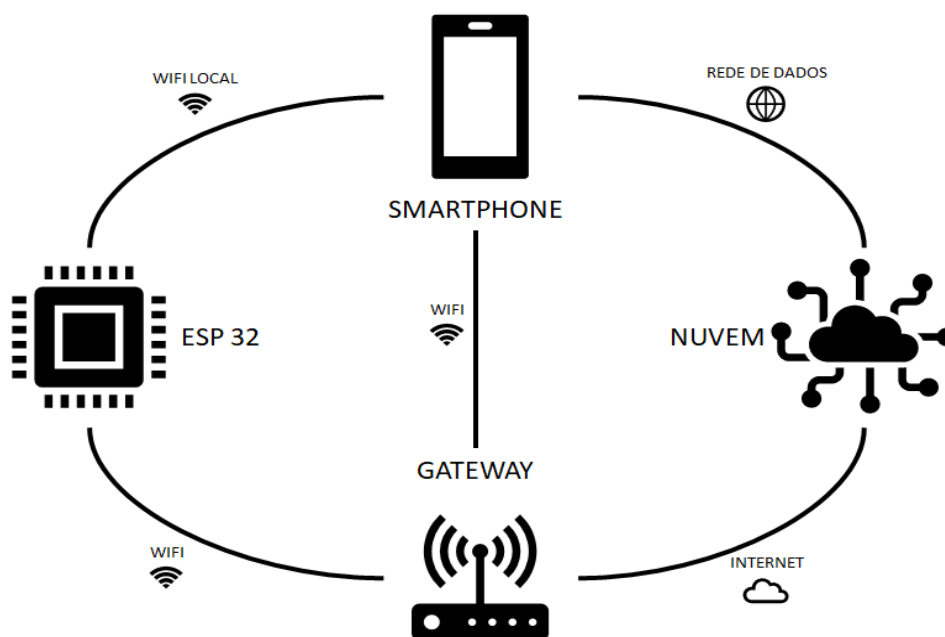
Fonte: Autoria própria.

Através do aplicativo AWA, o cliente saberá informações sobre sua média de consumo do botijão de gás medido e a porcentagem de quanto ainda falta para o término do gás presente no botijão, basta estar conectado a uma rede de dados com internet. O consumidor será informado por meio de notificações no celular, através do aplicativo, quando o botijão estiver com 50%, 25% e menos que 10% da carga total, para atentar sobre a substituição do botijão de gás do seu estabelecimento.

A AWA também emite alertas ao smartphone através do aplicativo sobre possíveis vazamentos, por meio de um sensor que detecta a presença de gás butano no ambiente, contribuindo assim para a prevenção de acidentes.

Para facilitar a compreensão das configurações da AWA, foi desenvolvido o diagrama ilustrado pela figura 4, que representa a relação dos componentes do sistema para o correto funcionamento do equipamento.

Figura 4. Relação dos componentes do sistema para o correto funcionamento do equipamento.



Fonte: Autoria própria.

1.3. PREMISSAS DO PROJETO

Para o correto funcionamento deste projeto, tem-se como premissas:

- ✓ Rede WI-FI 2,4 GHz disponível no local de instalação para a conectividade da AWA;
- ✓ Ponto de energia elétrica 127V a 1,8 metros, de acordo com a NPT 028 do Corpo de Bombeiros do Paraná, para alimentação da AWA;
- ✓ 1(um) botijão de gás (P-5, P-13 ou P-45);
- ✓ Cliente possuir um smartphone com memória e poder de processamento com suporte Wi-Fi e dados de internet, executando o sistema operacional Android;
- ✓ Usuário interessado em produtos que utilizam a tecnologia IOT.

1.4 OBJETIVOS

1.4.1 OBJETIVO GERAL

Desenvolver um protótipo com um sistema apto para mensurar o peso de um botijão de gás, indicando o percentual da quantidade ainda existente, e capaz de identificar se há algum vazamento, tudo por meio de uma aplicação de IoT.

1.4.2 OBJETIVOS ESPECÍFICOS

- a) Desenvolver Firmware para o microcontrolador ESP32 com sensores e atuadores para comunicação com a internet;
- b) Desenvolver a PCI (Placa de Circuito Impresso) para integração dos componentes;
- c) Desenvolver um aplicativo de celular que armazene temporariamente e processe dados recebidos do microcontrolador para fornecer uma visualização e monitoramento;
- d) Desenvolver comunicação entre o Firebase, ESP32 e o aplicativo;
- e) Testar funcionalidade do AWA e aplicativo;
- f) Documentar o desenvolvimento do projeto.

1.5 ESTRUTURA DO TRABALHO

Nesta primeira seção foram introduzidos e discutidos o problema, a motivação e a proposta de um sistema de monitoramento que estima o peso de um botijão de gás, e detecta possíveis vazamentos e que informa através de um aplicativo Android.

A seguir são apresentados, via uma revisão bibliográfica, os conceitos técnicos utilizados no trabalho, a descrição da metodologia utilizada para realização deste trabalho e os testes das funcionalidades do aplicativo na prática. Finalmente, os resultados e discussões e as considerações finais são apresentados nas seções 5 e 6 deste documento.

1.6 TRABALHOS RELACIONADOS

Com a evolução da tecnologia e o advento do IoT, inúmeras soluções estão em desenvolvimento para simplificar atividades da rotina do usuário. Com o GLP não é diferente, trabalhos relacionados têm sido desenvolvidos recentemente.

Em Dal Ponte (2018) é apresentado um sistema capaz de monitorar o consumo de gás por meio de um aplicativo Android. Para isso foram utilizados sensores (4 células de carga) que se comunicam com um módulo ESP32, que por sua vez envia os dados para o aplicativo através do protocolo MQTT. Além disso, o peso atual do gás também é enviado para uma tabela criada no Google Drive.

Para Vargas e Pinto (2016), a ideia principal não é monitorar a quantidade de gás restante, e sim se há vazamento ou não. É enviada uma notificação ao usuário por meio de SMS para alertar sobre o vazamento, além disso há um alerta sonoro para alertar quem estiver no local. Os principais componentes de hardware utilizados foram: sensor de gás (MQ-6), módulo Arduino UNO para interpretar os dados do sensor e Shield GSM/GRPS SIM 900 para enviar SMS.

Por fim, em Medeiros e Santos (2017), é apresentada uma plataforma inteligente para o monitoramento de gás de cozinha. Os dados obtidos do sensor de gás MQ-2 e de força resistiva são processados no Arduino UNO e enviados para um servidor Web por meio de um Shield Ethernet, e são recuperados em um aplicativo Android. No aplicativo, o usuário visualiza a quantidade de gás restante no botijão e acessa o histórico de medições. Além disso, há um buzzer,

no hardware junto ao botijão, responsável por emitir um alerta sonoro quando há vazamento. Um protótipo do hardware é mostrado na Figura 5.

Figura 5. Protótipo do hardware de Medeiros e Santos.



Fonte: MEDEIROS; SANTOS (2017).

Dessa forma, a Tabela 1 representa o comparativo entre os trabalhos mencionados acima e o sistema idealizado pelo projeto AWA. O trabalho de Medeiros e Santos (2017) apresenta a melhor solução quanto ao hardware e tipo de alerta, porém o AWA possuirá a vantagem de notificar através do aplicativo por conexão Wifi, e arquivar todos os dados registrados em nuvem.

Tabela 01. Comparativo da AWA com os trabalhos relacionados.

	AWA (2021)	DAL PONTE (2018)	MEDEIROS E SANTOS (2017)	VARGAS E PINTO (2016)
MÉTODO DE MEDIÇÃO	4 CÉLULAS DE CARGAS	4 CÉLULAS DE CARGAS	1 FORÇA RESISTIVA	X
MICROCONTROLADOR	ESP 32	ESP 32	ATMEGA328	ATMEGA328
SISTEMA OPERACIONAL	ANDROID	ANDROID	ANDROID	X
APLICAÇÃO	FIREBASE	MQTT	SERVIDOR PHP	SMPP
CONEXÃO	WI-FI	WI-FI	ETHERNET	GSM / GRPS
BANCO DE DADOS	FIREBASE	GOOGLE DRIVE	MySQL	X
MONITORAMENTO DE VAZAMENTO DE GÁS	SIM	X	SIM	SIM
NOTIFICAÇÕES	ATRAVÉS DO APLICATIVO	ATRAVÉS DO APLICATIVO	BUZZER	SMS
PLACA DE CIRCUITO IMPRESSO PRÓPRIA	SIM	X	X	X

Fonte: Autoria própria.

2 REVISÃO BIBLIOGRÁFICA

2.1 GÁS LIQUEFEITO DE PETRÓLEO (GLP)

O gás liquefeito de petróleo (GLP) é o combustível composto essencialmente por dois gases extraídos do petróleo: o butano e o propano. É comum também, conter em sua composição, outros hidrocarbonetos como o etano. O GLP é incolor e para tornar mais seguro a sua utilização, adiciona-se um composto à base de enxofre, de modo a torná-lo perceptível ao olfato humano em casos de vazamento (ANP, 2020).

Comparado com a outras fontes de energia, o GLP tem vantagem nos itens abaixo:

- ✓ É de fácil manuseio, transporte e armazenagem, o que proporciona grande disponibilidade;
- ✓ Oferece economia significativa em relação a outras fontes de energia;
- ✓ Gera energia limpa, que não produz resíduos pesados e não agride o meio ambiente;
- ✓ É armazenado em recipientes próprios, garantia de fornecimento contínuo.

Por suas características e aplicabilidade, mais de 90% da população brasileira depende da distribuição de Gás GLP todos os dias, seja em residências – na cozinha e no banho – ou em atividades ligadas à indústria, comércio, prestações de serviços ou agronegócio (CONSIGÁS, 2017).

Os botijões de uso doméstico devem ser fabricados de acordo com a Associação Brasileira de Normas Técnicas (ABNT), que estabelece os requisitos para fabricação, armazenamento e requalificação dos recipientes para a certificação das distribuidoras (SINDIGÁS, 2015).

Um dos órgãos de fiscalização vinculado ao setor de GLP é o Inmetro – Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, autarquia vinculada ao Ministério do Desenvolvimento, Indústria e Comércio Exterior, que


fiscaliza os botijões, especialmente em relação ao peso correto do produto em cada recipiente e os sistemas de medição do GLP a granel (SINDIGÁS, 2015).

Segundo a ABNT (Associação Brasileira de Normas Técnicas), a NBR 8460:2020, que especifica os requisitos mínimos exigíveis para peças e segurança, os métodos de ensaios, projeto, fabricação, alteração e utilização dos recipientes transportáveis destinados ao acondicionamento de gás liquefeito de petróleo (GLP). Há também a norma NBR 15186:2005, onde estabelece a base de armazenamento, envasamento e distribuição de GLP, bem como seu projeto e construção.

O GLP pode ser envasado em diversos tipos de cilindros e pesos (Figura 4). Outra informação que contém em sua embalagem é a tara (peso do recipiente vazio), a qual pode ser normalmente encontrada em sua alça. A tara varia de acordo com o processo adotado por cada fabricante na produção dos recipientes de gás, sendo aqueles os responsáveis por fazer a manutenção dos botijões e o reabastecimento do mesmo (MEDEIROS, SANTOS, 2017).

Segundo a Liquigás (2020), o modelo P13 é o mais usado em residências seguido do P45. Na figura 6 seguem as principais características dos modelos de botijões existentes no mercado.

Figura 6. Modelos de botijões de gás.



Nome		P-8	P-13	P-20	P-45
Kg		8	13	20	45
Diâmetro (mm)		300	360	310	376,5
Altura (mm)		464	476	878	1299
Aplicação		Fogões domésticos	Fogões domésticos	Empilhadeiras e balonismo.	Condomínios, indústrias, hospitais, bares, restaurantes, farmácias, balonismo entre outras.
Material		Aço carbono de 2,5 a 3,0 mm de espessura	Aço carbono de 2,5 a 3,0 mm de espessura	Aço carbono de 2,5 a 3,0 mm de espessura	Aço carbono de 2,5 a 3,0 mm de espessura
Principal região Consumidora		São Paulo, Mato Grosso, Rio de Janeiro, Piauí, Maranhão, Sergipe e Rondônia	Todas as regiões do Brasil	Todas as regiões do Brasil	Todas as regiões do Brasil
Pressão interna		2 a 7 Kg/cm ²	2 a 7 Kg/cm ²	2 a 7 Kg/cm ²	2 a 7 Kg/cm ²
Dispositivo de Segurança		Plugue fusível	Plugue fusível	Válvula sobre pressão	Válvula sobre pressão
Padrão da Válvula		Norma ABNT NBR 8614	Norma ABNT NBR 8614	Norma ABNT NBR 8614 (consumo)	Norma ABNT NBR 8614 (consumo)

Fonte: SINDIGÁS, 2015.

2.1.1 PERIGOS DO GLP

O GLP é um produto que deve ser manuseado com cuidado, devido sua alta capacidade de formar mistura explosiva com o ar, o qual entra em ignição com muita facilidade, ocasionando acidentes graves com pessoas e instalações. O GLP, quando inalado em grande quantidade, pode provocar tonturas e irritações no sistema respiratório, olhos e queimaduras na pele, além de produzir um efeito anestésico e levar à morte (FDE, 2009).

Para a segurança do consumidor, os botijões devem ser armazenados em áreas bem ventiladas e distantes de vias públicas com rigoroso controle de possíveis fontes de ignição (ASMIRG, 2009). Acessórios como regulador e mangueira certificados pelo INMETRO são indispensáveis, os quais possuem prazo de validade de 5 anos da fabricação. Passado este prazo, podem apresentar trincas, fissuras e outros defeitos. Os botijões possuem também um dispositivo de segurança (plugue fusível) que, em caso de aumento da pressão interna, libera o GLP impedindo que ocorra um vazamento e/ou explosão do vasilhame (MOREIRA, 2015).

Estima-se que o vazamento do conteúdo de um simples botijão doméstico (P13) possa formar uma nuvem explosiva de até 200.000 litros de mistura com o ar em condições ambientais normais (SOBRINHO, 2014). Esses vazamentos são extremamente perigosos, pois envolve o risco de explosão e incêndio. Por se tratar de um produto químico invisível e muito inflamável, uma pequena faísca no gás pode provocar uma explosão e um incêndio de grande magnitude (MOREIRA, 2015).

O mais grave acidente industrial com GLP ocorreu em 1984 em San Juanico - México que vitimou fatalmente cerca de 600 moradores que viviam nos arredores de uma planta de grande capacidade de armazenagem do produto. (FUNDACENTRO, 2014). Dados do Sindicato Nacional das Empresas Distribuidoras de Gás Liquefeito de Petróleo (SINDIGÁS) mostram que de 360

milhões de vasilhames vendidos no Brasil por ano, os incidentes com vazamento chegam próximo de 7 mil.

Ainda segundo dados do Sindigás (2015), 51% dos acidentes ocorrem na instalação inadequada dos recipientes, 23% no uso inapropriado, 13% estão relacionados a má conservação dos recipientes e 14% a outras circunstâncias.

No ano de 2015, foram registrados 3.391 vazamentos com GLP no estado de São Paulo, resultando em um aumento de 6,13% em relação ao ano anterior, conforme demonstrado na Tabela 02. No mesmo ano, foram registrados 358 incêndios com GLP no estado de São Paulo (LIQUIGÁS, 2020).

Tabela 02. Número de registros de vazamentos com GLP no estado de SP.

NATUREZA	Vazamento de GLP engarrafado			
	2012	2013	2014	2015
ANO				
CAPITAL	1.522	1.529	1.239	1.305
GRANDE SP	758	919	717	763
INTERIOR	1.162	1.464	1.239	1.323
TOTAL	3.442	3.912	3.195	3.391

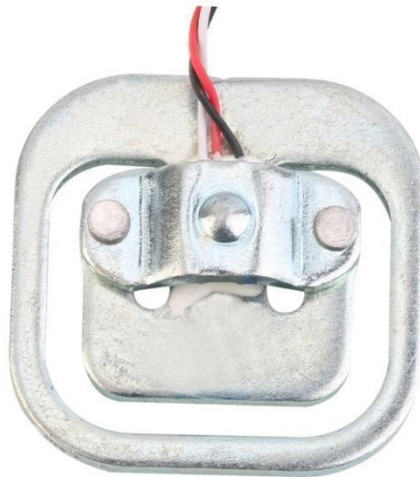
Fonte: Banco de Dados do SIOPM (LIQUIGÁS, 2020).

2.2 CÉLULA DE CARGA

Uma célula de carga é um transdutor que converte a massa (kg) em uma saída elétrica mensurável (mV). Geralmente, são usadas no mínimo duas células de carga, uma para medir compressão e a outra tensão (forças aplicadas em direções diferentes) (VIDAL, 2017).

Embora existam muitas variedades de sensores de força, as células de carga são o tipo mais comumente utilizado (Figura 7). Trata-se de um anel de alumínio ou de aço inox, com perfil projetado para sofrer deformação conforme o peso atua sobre ela, e a medição dessa deformação, feita por um extensômetro (OMEGA, 2020).

Figura 7. Célula de carga.

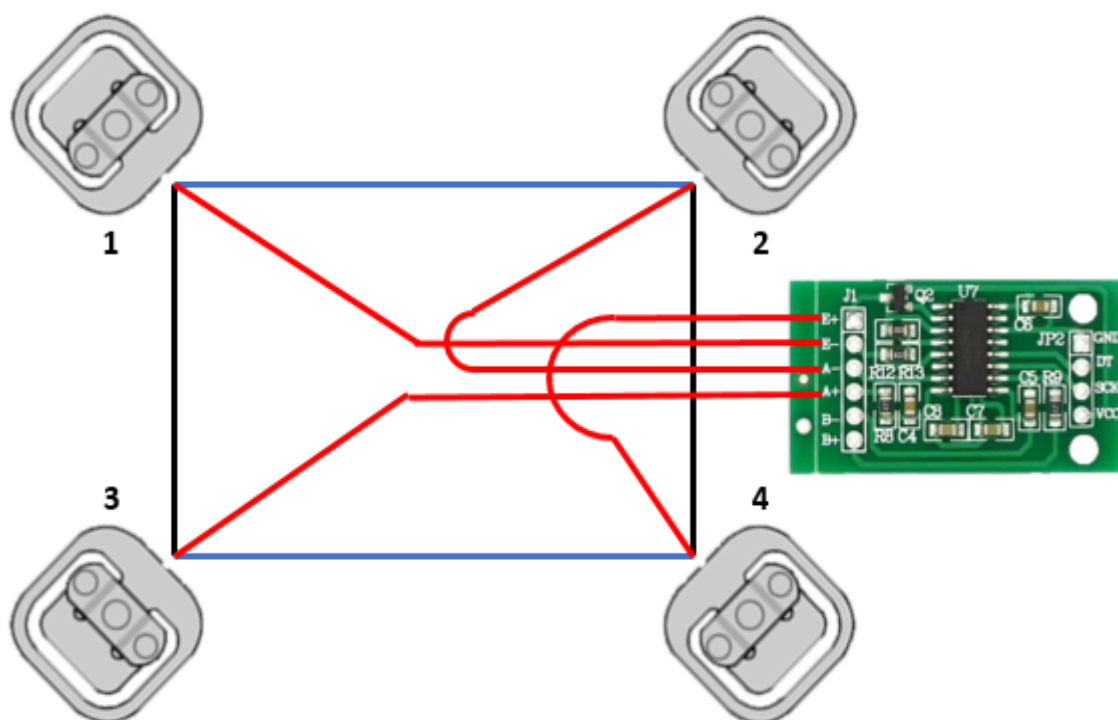


Fonte: VIDAL, 2017.

Segundo RIPPER NETO (2007), extensômetros resistivos medem deformação específica e seu princípio de funcionamento está baseado no fato que a resistência elétrica de um condutor varia proporcionalmente à sua deformação linear. Tais sinais elétricos passam por um processo de amplificação e conversão, para que o dispositivo de controle possa receber e processar essa informação.

Na figura 8, tem-se a representação do esquemático das células de carga juntamente com o módulo conversor e amplificador HX711, formando assim uma ponte de Wheatstone, um circuito elétrico capaz de medir pequenas mudanças na resistência correspondentes à tensão.

Figura 8. Esquemático das células de carga com módulo HX711.



Fonte: Autoria própria.

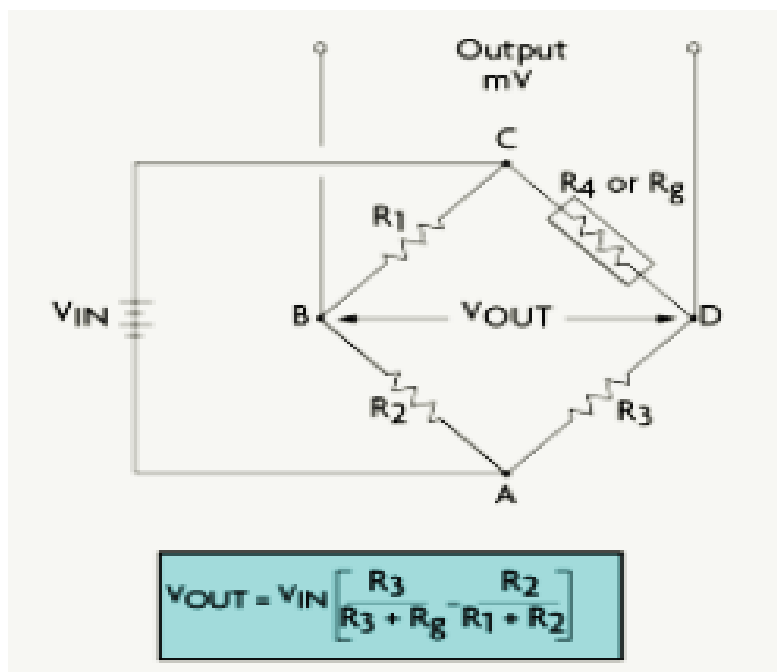
2.2.1 PONTE DE WHEATSTONE

A ponte de Wheatstone é um circuito muito utilizado em instrumentação eletrônica, pois por meio dela é possível medir, além de resistência elétrica, diversas outras grandezas físicas, como temperatura, força e pressão. Para isso, basta utilizar sensores ou transdutores que convertem as grandezas a serem medidas em resistência elétrica dada em volts (MARKUS, 2009).

Segundo OMEGA (2020), a equação da ponte de Wheatstone afirma que, se R_1 , R_2 , R_3 e R_4 forem iguais, e uma tensão, V_{IN} , for aplicada entre os pontos A e C, a saída entre os pontos B e D não mostrará nenhuma diferença de potencial. No entanto, se R_4 for alterado para algum valor que não seja igual a R_1 , R_2 e R_3 , a ponte ficará desequilibrada e existirá uma tensão nos terminais de saída. Em uma configuração conhecida como ponte G, o sensor de

deformação variável tem resistência R_g , enquanto os outros braços são resistores de valor fixo, conforme demonstrado na figura 9.

Figura 9. Equação da ponte de Wheatstone.



Fonte: OMEGA, 2020.

O sensor, no entanto, pode ocupar um, dois ou quatro braços da ponte de Wheatstone, dependendo da aplicação. A tensão total, ou tensão de saída do circuito (V), é equivalente à diferença entre a queda de tensão em R_1 entre R_4 , ou R_g . Normalmente, em uma ponte de Wheatstone com quatro elementos, dois sensores são ligados na compressão e dois na tração.

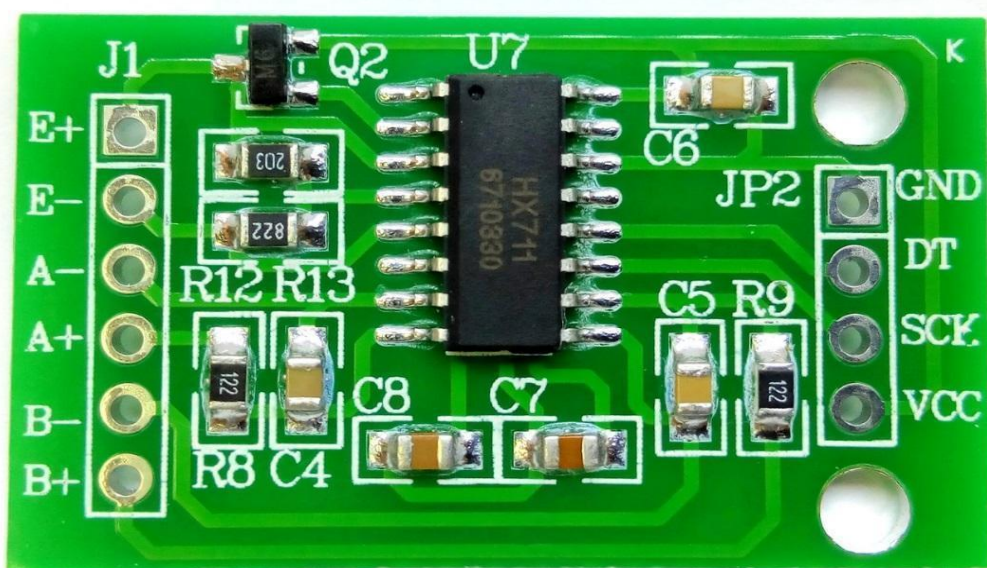
Para medidores localizados em trechos adjacentes, a ponte fica desequilibrada proporcionalmente à diferença de deformação. Para medidores em trechos opostos, a ponte equilibra proporcionalmente à soma das deformações (OMEGA, 2020).

2.3 MÓDULO HX711

Células de carga não oferecem dados com grandezas elétricas na faixa de leitura de alguns equipamentos de medição, sendo necessário um amplificador de sinal e um conversor análogo/digital (ALVES, 2016).

Como informado por seu datasheet, o HX711 (Figura 10) é um amplificador de instrumentação e conversão analógico-digital de 24-bit de resolução.

Figura 10. Módulo HX711.



Fonte: AVIA Semicondutor datasheet HX711.

Para seu funcionamento é necessária uma fonte de 2,7 – 5V e a transmissão dos dados convertidos ocorre através de uma interface serial, que utiliza um pino de dados e um de pulsos (DOUT e SCK). Além de ser de baixo consumo (menor que 1,5 mA) e ainda conta com a facilidade de encontrar bibliotecas (BIBLIOTECA HX711) ou projetos, auxiliando na programação e utilização.

2.4 SENSORES MEDIDORES DE GÁS SÉRIE MQ

Os sensores da série MQ (*Quality Monitoring*) funcionam como dispositivos de detecção na presença de algum gás (KARVINEN e KARVINEN, 2014).

Existem uma série de sensores de gases MQ, os quais possuem um pequeno aquecedor para que esteja em temperatura de referência no seu interior com um sensor eletroquímico, tornando-o sensível aos gases à temperatura ambiente, conforme demonstrado na Tabela 03 (ALVES, MENDONÇA, 2018).

Tabela 03. Sensores da série MQ e suas funções.

TIPO DO SENSOR	SENSÍVEL PARA
MQ-02	Metano, butano, GLP e fumaça
MQ-03	Álcool, Etanol e Fumaça
MQ-04	Metano e Gás CNG
MQ-05	Gás natural e GLP
MQ-06	GLP e gás butano
MQ-07	Monóxido de carbono
MQ-08	Gás hidrogênio
MQ-09	Monóxido de carbono e gases inflamáveis

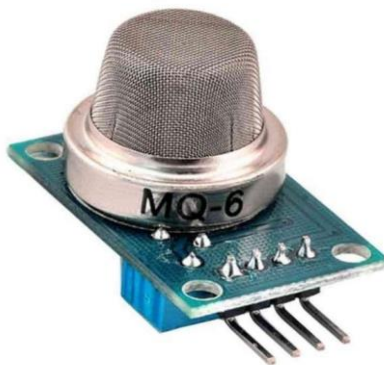
Fonte: ALVES, MENDONÇA, 2018.

O Sensor de Gás MQ- 6 (figura 11) é um componente com a função de analisar a presença de gás GLP (Gás de Cozinha), Propano, Isobutano e Gás Natural Liquefeito no ambiente (CANDIDO, *et.al.* 2017).

Seu módulo possui sensor de alta sensibilidade ao monóxido de carbono em ampla faixa de temperatura. Tem vantagens como a vida útil longa, baixo custo e estabilidade nas leituras. Seu aquecedor interno utiliza tensão elétrica de 5V e sua sensibilidade pode ser ajustada por meio do potenciômetro, sendo calibrado no ponto de acionamento. Além de ser de fácil instalação, possui um sinal de saída capaz de ser captado por outros dispositivos, ou seja, o sinal pode

ser lido por um processador e comunicar-se eletronicamente por uma rede de dados (CORREIA *et al*, 2017).

Figura 11. Sensor de gás MQ-6.



Fonte: (CANDIDO, *et.al*. 2017).

O gás de cozinha contém uma mistura de gases que em uma concentração de 1000 ppm já torna o ambiente asfixiante (FISPQ, 2015). Para isso, o sensor MQ-06 possui uma faixa de sensibilidade que varia em concentrações entre 100 ppm a 10.000 ppm, onde é possível detectar em vários níveis de diferentes gases no ambiente (PINTO, 2016).

Os sensores de gás MQ-2 e MQ-5 também possuem sensibilidade ao GLP e poderiam ser utilizados neste projeto, porém no momento da aquisição dos materiais apenas o MQ-6 estava disponível.

2.5 MICROCONTROLADOR

Os microcontroladores vêm revolucionando os projetos eletrônicos digitais devido à enorme versatilidade de hardware e software que oferecem. Por ser conhecido como Microprocessador de um Chip, opera em apenas um componente todos os elementos de um sistema microprocessado completo,

Outros microcontroladores foram avaliados para o desenvolvimento desse projeto, o ESP8266 e Arduino UNO R3, porém após algumas pesquisas, concluiu-se que o ESP32 encaixava melhor para o projeto pois é mais robusto em questão de Hardware, baixo custo de aquisição, maior rapidez, menor consumo de energia, grande memória e tem Wi-Fi integrado. Esse componente ainda demonstra ser mais versátil, pois além do clássico módulo de comunicação Wi-Fi, apresenta um sistema com processador Dual Core, Bluetooth híbrido e múltiplos sensores embutidos, tornando a construção de sistema como internet das coisas (IoT) muito mais simples e compacto, facilitando qualquer projeto futuro (CURTO, 2018).

A Figura 13 apresenta um comparativo do ESP32 com os dois microcontroladores citados.

Figura 13. Comparativo entre ESP32, ESP8266 e Arduino UNO R3

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
<u>Clock</u>	160MHz	80MHz	16MHz
<u>WiFi</u>	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Fonte: CURTO, 2018.

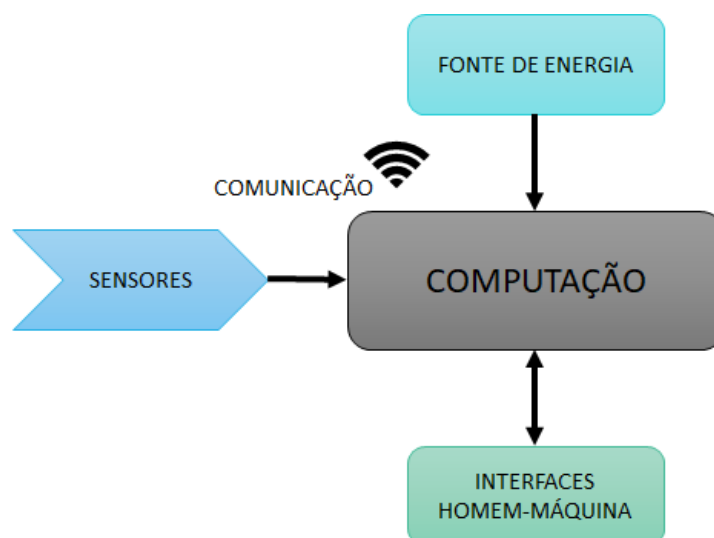
Em referência aos fatos acima, pode-se pensar na viabilidade de usar o ESP32 como um microcontrolador para executar projetos de baixo custo e alta conectividade com componentes de sistemas de automação residencial.

2.6 INTERNET DAS COISAS (IoT)

A IoT veio para oferecer a conectividade entre vários objetos do mundo físico que estão conectados à rede, através de sensores capazes de capturar e enviar eventos às plataformas de comunicação, que receberem os dados e as informações, propiciando, de forma inteligente, a manipulação de objetos interconectados (COLOMBO & LUCCA FILHO, 2018)

A Figura 14 representa, em uma forma resumida, o diagrama de blocos e as características de um objeto capaz de integrar uma IoT.

Figura 14. Exemplo de um diagrama de bloco no contexto de IoT.



Fonte: A autoria própria.

2.6.1 ARQUITETURA DA IOT

Ao longo do desenvolvimento da IoT, diferentes arquiteturas têm sido estudadas por pesquisadores (SETHI & SARANGI, 2017). De fato, uma arquitetura universal da IoT ainda não está definida, já que o cenário atual para Internet das Coisas é caracterizado pela presença de aplicativos construídos

exclusivamente que operam em uma arquitetura de rede já projetada, e à medida que os sistemas de IoT avançam (SILVA, 2017).

No geral, qualquer arquitetura que tenha sido mantida nas fases iniciais da pesquisa de IoT precisará ser revista e modificada pois a maioria das obras relacionadas à arquitetura IoT partiram do princípio de redes de sensores sem fio, porém no futuro, grande número de diferentes meios serão utilizados para permitir a comunicação entre dispositivos heterogêneos (CASTELLANI, *et. al.*, 2010).

O modelo de arquitetura TCP/IP, criado pelo DoD (Departamento de Defesa dos Estados Unidos) tornou-se popularmente utilizado (BRITO, *et. al.*, 2018). O modelo TCP/IP (Protocolo de Controle de Transmissão / Protocolo de Internet) (Figura 15) possui apenas quatro camadas: Aplicação, Transporte, Internet e Host/rede ou Acesso à rede, e foi projetado especialmente para ser o protocolo utilizado na Internet, tendo como principal característica o suporte direto a comunicação entre redes de diversos tipos (CURSO REDE DE COMPUTADORES, 2020).

Neste trabalho adota-se para o desenvolvimento do projeto essa arquitetura TCP/IP, o qual tem como característica a transferência de dados, robustez, controle de fluxo e segurança (SCRIMGER *et al.*, 2002), conforme é apresentado na figura abaixo para facilitar o entendimento dos protocolos e redes por camadas do TCP/IP IP

Figura 15. Arquitetura TCP/IP.

Camadas	Protocolos
Camada de Aplicação	HTTP MQTT
Camada de Transporte	TCP
Camada de Rede	IPV4
Camada Física	IEEE 802.11 IEEE 802.15.4

Fonte: Adaptado de CURSO REDE DE COMPUTADORES, 2020.

2.7 PROTOCOLOS DE COMUNICAÇÃO IOT

A Internet das Coisas (IoT), integra dispositivos de diversas tecnologias e protocolos de comunicação, que permitem a sua implementação. A diversidade de aplicações pode ainda requerer a escolha de sistemas de comunicação, como por exemplo Lora, Zigbee e Wi-fi, entre outros, que melhor se adequem aos requisitos destas aplicações. A seguir, discute-se sobre estas tecnologias e sua adequação para este projeto.

LoRa (*Long Range*, do inglês - Longo alcance) consiste em uma técnica de modulação que permite a transferência de longa gama de informações com uma baixa taxa de transferência. É uma tecnologia de camada física, que transmite os sinais na banda ISM SUB-GHz (868 MHz, 915 MHz e 433 MHz) em diferentes padrões de frequência. Utilizando frequências mais baixas do que as bandas ISM típicas (2,4 ou 5,8 GHz), a técnica permite a cobertura de uma grande área, especialmente em locais fechados. Em vez de usar a transmissão de *narrow band*, emprega uma forma de modulação *Spread Spectrum Chirp* (CSS), que distribui informações sobre diferentes canais de frequência e taxas

de dados (LEAL, 2019). Essa proposta está focada em acessos a dispositivos a até 15 quilômetros de distância, utilizando uma estação de rádio de longo alcance, além da sua facilidade de desenvolvimento, baixo consumo de energia e custo de implementação (XAVIER, 2018). Pode ser adequada para aplicações e serviços como tarifação de água, energia, entre outros tipos (BRITO, *et. al.*, 2018).

O Zigbee é um conjunto de especificações desenvolvido pela *Zigbee Alliance* para utilização em *smart home* e IoT, que define as camadas subsequentes às camadas estabelecidas pelo IEEE 802.15.4, referente a Internet, transporte e aplicação do modelo TCP/IP, oferecendo serviços de segurança, tolerância a erros e conexão de novos dispositivos (FRENZEL, 2013). Suas características incluem baixo consumo de energia, taxa de transmissão e custo de implementação. Seu alcance é limitado até 100 metros, e necessita de repasse das informações pelos nós da rede, em múltiplos saltos até alcançar o seu destino de longas distâncias (DING, *et. al.*, 2015).

O protocolo Bluetooth, definido no padrão IEEE 802.15 foi projetado para as redes conhecidas como WPAN a se tornou comercialmente utilizado em vários dispositivos. Suas aplicações são transferência de dados, ligação de periféricos como fones de ouvidos, sistemas de som, tanto a celulares, gadgets, dispositivos IoT, entre outros (OLIVEIRA, 2017). Uma variação do Bluetooth é conhecida como BLE (*Bluetooth Low Energy*), com baixo consumo de energia e alcance limitado até 50 metros. Opera na banda *Industrial Scientific Medical* (ISM) de 2,4GHz e define 40 canais de radiofrequência (RF) com espaçamento de 2MHz (GOMES, 2001).

O padrão IEEE 802.11 é internacionalmente conhecido como Wi-Fi, acrônimo para *Wireless Fidelity*. O primeiro padrão 802.11 a obter grande aceitação da indústria foi o 802.11b. Embora os produtos 802.11b sejam todos baseados no mesmo padrão, há uma preocupação que produtos de diferentes fornecedores interrompam com precisão (STALLINGS, 2006). O termo Wi-Fi quer dizer um conjunto de especificações para redes locais sem fio (*Wireless Local Area Network* (WLAN)), e tem como principal característica conectar

dispositivos em redes locais sem fio. Esses padrões trabalham em uma faixa definida 2,4GHz a 5GHz, e essas redes possuem um número finito de elementos conectados na rede (OLIVEIRA, 2017).

Dessa forma, para o projeto, a comunicação escolhida foi a tecnologia Wi-Fi pois, além de ser mais flexível, oferece suporte a vários dispositivos como computadores, smartphones, tablets, entre outros, e possui conectividade com o microcontrolador escolhido, o ESP32 (dadas suas características e preço acessível, comparando com o seu antecessor ESP8266), sem a necessidade de compra de um módulo compatível, além de estar presente em 79,1% dos domicílios no Brasil, segundo a matéria da TOKARNIA - Repórter da Agência Brasil, 2020.

Como dizia SILVA (2017), os principais desafios da IoT para os anos seguintes são: a regulamentação, padronização de protocolos e garantia de segurança, o qual podem ser tornar vulneráveis a ataques. Ainda não foi definido um padrão de protocolos para todas as aplicações IoT, pois cada uma tende a ser eficiente para apenas um único problema. Existem um bom número de protocolos de rede para IoT aplicáveis nos aparelhos como HTTP, MQTT, CoAP, UPnP, XMPP, entre outros, mas cada qual com uma tecnologia e aplicações distintas SILVA (2017).

Em aplicações IoT, para o funcionamento dos protocolos mencionados, é utilizada a comunicação TCP (Transmission Control Protocol) pois permite o envio de dados de forma ordenada e garante a entrega por meio do IP (*Internet Protocol*). Um protocolo de comunicação que tem recebido destaque nos últimos anos para a área de Internet das Coisas é o protocolo MQTT, lançado pela IBM em 1999 e que possui um modelo de publicação/assinatura, do inglês *publish/subscribe* (SONI, MAKWANA, 2017).

Esse modelo de comunicação *publish/subscribe* é baseado na troca descentralizada de mensagens, conhecidas como eventos. É um conjunto de clientes que publicam esses eventos (produtores), os quais são encaminhados para clientes que registraram interesse em recebê-los (consumidores) (SILVESTRE, 2005).

Recentemente, foi publicada a versão 5.0 do protocolo MQTT (OASIS 2019), que além de especificar as mensagens, presentes nas versões anteriores, introduz um novo tipo de mensagem denominado AUTH, que tem como finalidade permitir a implementação de mecanismos de autenticação estendida.

Os objetivos da utilização do MQTT são minimizar os requerimentos de recursos de dispositivo e de largura de banda tentando garantir confiabilidade e garantia de entrega. O protocolo segue o modelo cliente/servidor. Os dispositivos sensores são clientes que se conectam a um servidor (chamado de broker) usando TCP (JAFFEY, 2014).

2.7.1 FIREBASE

Apesar das características positivas do MQTT, o projeto apresentado utilizará o Firebase, o qual é uma plataforma BaaS (*Backend as a Service*) obtida pela Google em 2014, que propõe auxiliar desenvolvedores a construir aplicativos de qualidade para iOS, Android ou Web já que todo o Backend será configurado e gerenciado pelo Firebase (MORIBE, 2016).

Backend as a Service é um modelo de serviço de computação em nuvem que atua como um componente de *middleware*, permitindo, aos desenvolvedores, conectar seus aplicativos a serviços em nuvem através de *Software Development Kit* (SDK) e *Application Programming Interface* (API) (BATSCHINSKI, 2016).

Quando os clientes usam o Firebase, atuam como "controladores de dados" com relação a todos os dados pessoais fornecidos ao Google que estejam associados ao Firebase. O Google, nesse caso, atua como "processador de dados". Em 2018, o GDPR (Regulamento geral de proteção de dados) incluiu termos de segurança e processamento de dados para todos os serviços do Firebase (FIREBASE, 2020c).

O Firebase é certificado de acordo com os principais padrões de privacidade e segurança, como ISO e ISO 27018. Além disso, seus serviços criptografam dados em trânsito usando HTTPS e isolam logicamente os dados do usuário, e mantendo assim as informações pessoais seguras (FIREBASE, 2020c).

A partir da integração de dispositivos (por exemplo, microcontroladores) e dos recursos disponíveis do Firebase é possível desenvolver aplicações em Internet das Coisas. O ecossistema de produtos do Firebase subdivide-se em dois grupos, sendo um deles direcionado ao desenvolvimento e testes do aplicativo, como o *Firebase Realtime Database*, e outro grupo voltado à expansão e envolvimento com o público-alvo da aplicação, destacando o *Firebase Cloud Messaging*, conforme apresentados a seguir (FIREBASE, 2018).

2.7.1.1 Firebase Realtime Database

O *Firebase Realtime Database* é um banco de dados hospedado na nuvem. Os dados são sincronizados simultaneamente e armazenados como JSON (*JavaScript Object Notation*) com todos os usuários conectados (QUESSADA *et. al*, 2019).

O *Realtime Database* é um banco de dados NoSQL e, por isso, tem otimizações e funcionalidades distintas de um banco de dados relacional. A API do *Realtime Database* foi desenvolvida para autorizar apenas operações que possam ser executadas com agilidade. Isso possibilita uma ótima experiência em tempo real que atende a milhões de usuários sem comprometer a capacidade de resposta (FIREBASE, 2020a).

Para este recurso é utilizada a sincronização de dados, ao contrário de típicas solicitações HTTP. Assim que algum dado é alterado, todos os dispositivos conectados recebem as atualizações, em milissegundos. Além disso, se a conexão é perdida, o estado atual do dispositivo é sincronizado com o estado atual do servidor após restabelecida a conexão (FIREBASE, 2020a).

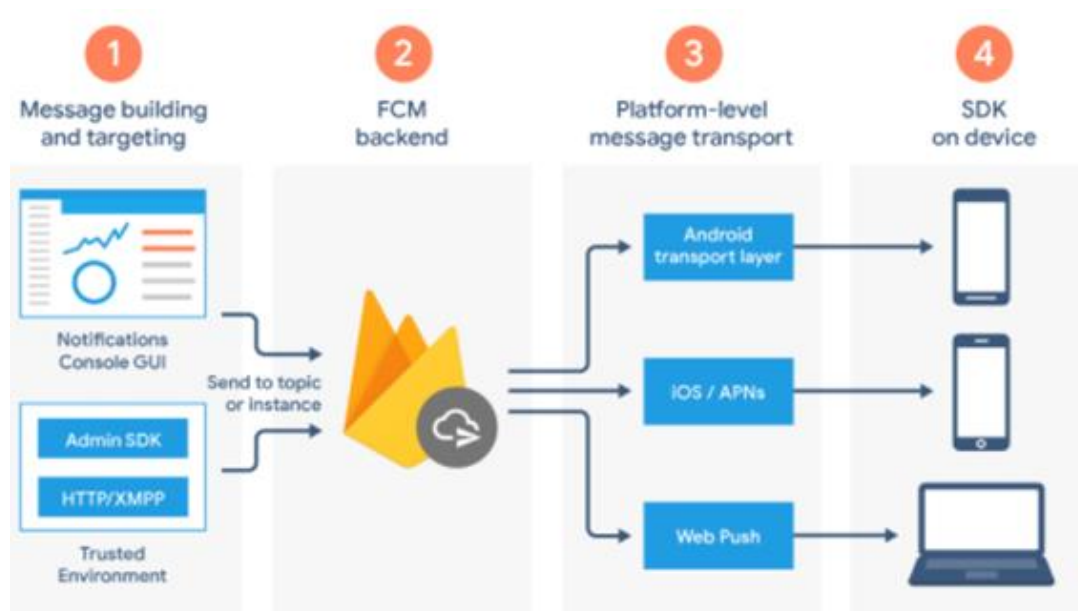
2.7.1.2 Firebase Cloud Messaging

O *Firestore Cloud Messaging* (FCM) permite a entrega confiável de mensagens e notificações, podendo avisar o aplicativo de um usuário que os dados estão disponíveis para sincronização, promovendo interações e retenção de usuários (LUZ, SILVA, 2017).

O FCM permite o envio de mensagens para os Apps dos usuários em várias plataformas, podendo ser enviada para um único ou um grupo de dispositivos ou por tópicos (enviando mensagens de notificação e mensagens de dados). Para casos de uso como mensagens instantâneas, uma mensagem pode transferir um *payload* de até 4 KB para um app cliente (FIREBASE, 2020b).

Conforme descrito pelo FIREBASE (2021), os componentes que constituem a arquitetura do *Firestore Cloud Messaging* (FCM) conforme apresentados nas 4 (quatro) etapas abaixo, e ilustrado na Figura 16.

Figura 16. Visão geral da arquitetura do Firestore Cloud Messaging.



Fonte: FIREBASE, 2020.

1 – Ferramentas para escrever ou criar solicitações de mensagens;

2 - O *back-end* do FCM, que (entre outras funções) aceita solicitações de mensagens, realiza a distribuição de mensagens por tópicos e gera metadados de mensagens, como o ID da mensagem;

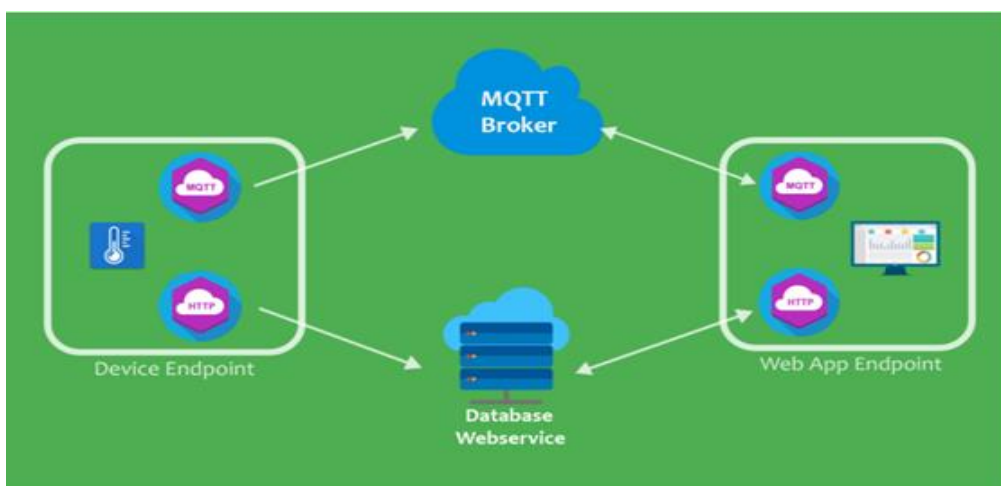
3 - Uma camada de transporte no nível da plataforma, que encaminha a mensagem para o dispositivo de destino, processa a entrega da mensagem e aplica a configuração específica da plataforma quando apropriado;

4 - O SDK do FCM no dispositivo do usuário, em que a notificação é exibida ou a mensagem é processada.

2.7.2 FIREBASE x MQTT

A partir da comunicação, os dados são enviados e alocados em um banco de dados para que possam ser analisados, e então executadas as ações necessárias. Para realizar tal procedimento, uma das maneiras pode ser alcançada através do MQTT. A figura 17 mostra a representação desta solução:

Figura 17. Representação do uso de Firebase.



Fonte: FREECODECAMP (2019).

Conforme representado, o dispositivo responsável pela leitura de dados dos sensores realiza o envio dos dados para o *broker* (MQTT servers), e através do *broker* um *smartphone* pode recuperar esses dados. Porém, segundo FREECODECAMP (2019), para que haja a inclusão de banco de dados em nuvem, tanto o dispositivo quanto o *smartphone* devem fazer requisições HTTP para salvar os dados, ou seja, deve ser implementado dois clientes: um MQTT e outro HTTP. Com o serviço do Firebase, o *Realtime Database*, é possível implementar um protótipo de IoT sem algumas das complicações mencionadas acima.

Inicialmente foi realizada, na prática, a comunicação via MQTT entre o microcontrolador e o *smartphone*. Porém, como descrito anteriormente percebeu-se então a necessidade de armazenar os dados em nuvem. Chegou-se, assim, ao serviço de bancos de dados do Firebase e comunicação via requisições HTTP. Estas características permitiram substituir a comunicação via MQTT para Firebase.

Os Bancos de Dados em Tempo Real são apropriados para gerenciar grandes volumes de dados compartilhados em aplicações simultâneas. Esses bancos são similares aos convencionais, porém não possui as restrições temporais para transações e manipulação de dados (GROSSMANN, 2018)

Para realizar a conexão do banco de dados com os clientes, o Firebase mantém uma conexão bidirecional aberta através de websockets, permitindo uma maior interação entre um navegador e um Web Site. Assim, qualquer alteração no banco de dados, os clientes conectados são informados, e os dados salvos (FERNANDES, 2013).

Portanto, devido a integração da comunicação entre clientes e o banco de dados em tempo real, o Firebase se torna uma opção mais atrativa para a implementação de um protótipo de IOT. Além disso, o Firebase ainda conta com diversos serviços dos quais podem ser implementados notificações push para *smartphone* e autenticações de segurança via aplicativo (FREECODECAMP, 2019).

3 MATERIAIS E MÉTODOS

No decorrer deste capítulo, serão apresentados e explicados todos os componentes utilizados no projeto e como ocorreu a integração entre eles.

3.1 HARDWARE

3.1.1 ESTRUTURA DA BALANÇA

Uma balança comercial da marca Wincy Casa (figura 18) foi utilizada para darmos início ao esqueleto da AWA. Além da sua estrutura, as 4 células de cargas permaneceram para utilização no projeto.

Figura 18. Balança marca Wincy Casa.



Fonte: Autoria própria.

Na Tabela 04, pode ser visto as especificações da balança utilizada.

Tabela 04: Especificações da Balança Wincy casa.

ESPECIFICAÇÕES	RESULTADOS
CAPACIDADE	180 kg
RESOLUÇÃO	0,1 g
PLATAFORMA	Vidro temperado
DIMENSÕES	28 cm x 28 cm x 2 cm

Fonte: Adaptado do Manual da Balança Wincy casa.

3.1.2 FONTE DE ENERGIA

Para fornecer energia ao circuito, optou-se por uma fonte Modelo AC-DC 127 – 5V (fonte de celular), pois, de acordo com o datasheet do regulador de entrada (Datasheet B1117), é necessária uma tensão de 5V para que seja fornecido ao circuito a tensão de 3,3V.

É necessário se adequar a distância mínima de afastamento entre o recipiente de GLP e a rede elétrica. De acordo com a NPT 028 - Corpo de Bombeiros do Paraná (2014), item 5.3.1 afirma que “os recipientes transportáveis trocáveis ou abastecido no local (capacidade volumétrica igual ou inferior a 0,5 m³) e os recipientes estacionários de GLP (capacidade volumétrica maior que 0,5 m³) devem ser situados no exterior das edificações, em locais ventilados e obedecendo aos afastamentos mínimos” conforme Tabela 05.

Tabela 05. Afastamento do recipiente de GLP para redes elétricas.

Nível de Tensão (kV)	Distância mínima (m)
Menor ou igual a 0,6	1
Entre 0,6 e 23	3
Maior ou igual a 23	7,5

Fonte: NPT 028, 2014.

Dessa forma, foi adicionado um cabo de 2 metros para que se adeque a Norma de Procedimento Técnico 028.

Na sequência, foi realizada a medição da corrente na entrada do regulador de tensão para que pudéssemos dimensionar a espessura do cabo. A corrente aferida foi de 120 mA, conforme mostrado na figura 19.

Figura 19. Medição do multímetro da corrente de entrada.



Fonte: Aatoria própria.

Assim, de acordo com a Tabela 06 a seguir, o fio deve possuir número AWG (*American Wire Gauge*) igual ou inferior a 30, para que se tenha uma margem de segurança maior, além da menor dificuldade em encontrar este tamanho, foi escolhido o cabo AWG 20. O qual corresponde ao diâmetro mínimo do fio como 0,8118 mm e secção transversal de 0,52 mm², com uma resistência de 32,69 ohms/Km.

A queda de tensão referente ao cabo pode ser calculada por:

$$V = I \times R, \text{ assim } V = 0,12 \times (2 \times 2 \times 0,03269) = \mathbf{0,0157 \text{ V}}$$

onde R (resistência) foi expressa como $2 \times L \times 0,03269$;

onde L é o comprimento do cabo, é multiplicado por 2 pois se considera o par positivo/negativo.

Deste modo, a queda de tensão comparada à tensão de 5 volts é 0,31%.

Tabela 06. Tabelas de fios AWG.

Número AWG	Diâmetro (mm)	Secção (mm ²)	Número de espiras por cm	kg por Km	Resistência (ohms/Km)	Capacidade (A)
0	11,86	107,2			0,158	319
0	10,4	85,3			0,197	240
0	9,226	67,43			0,252	190
0	8,252	53,48			0,317	150
1	7,348	42,41		375	1,4	120
2	6,544	33,63		295	1,5	96
3	5,827	26,67		237	1,63	78
4	5,189	21,15		188	0,8	60
5	4,621	16,77		149	1,01	48
6	4,115	13,3		118	1,27	38
7	3,665	10,55		94	1,7	30
8	3,264	8,36		74	2,03	24
9	2,906	6,63		58,9	2,56	19
10	2,588	5,26		46,8	3,23	15
11	2,305	4,17		32,1	4,07	12
12	2,053	3,31		29,4	5,13	9,5
13	1,828	2,63		23,3	6,49	7,5
14	1,628	2,08	5,6	18,5	8,17	6
15	1,45	1,65	6,4	14,7	10,3	4,8
16	1,291	1,31	7,2	11,6	12,9	3,7
17	1,15	1,04	8,4	9,26	16,34	3,2
18	1,024	0,82	9,2	7,3	20,73	2,5
19	0,9116	0,65	10,2	5,79	26,15	2
20	0,8118	0,52	11,6	4,61	32,69	1,6
21	0,723	0,41	12,8	3,64	41,46	1,2
22	0,6438	0,33	14,4	2,89	51,5	0,92
23	0,5733	0,26	16	2,29	56,4	0,73
24	0,5106	0,2	18	1,82	85	0,58
25	0,4547	0,16	20	1,44	106,2	0,46
26	0,4049	0,13	22,8	1,14	130,7	0,37
27	0,3606	0,1	25,6	0,91	170	0,29
28	0,3211	0,08	28,4	0,72	212,5	0,23
29	0,2859	0,064	32,4	0,57	265,6	0,18
30	0,2546	0,051	35,6	0,45	333,3	0,15
31	0,2268	0,04	39,8	0,36	425	0,11

32	0,2019	0,032	44,5	0,28	531,2	0,09
33	0,1798	0,0254	56	0,23	669,3	0,072
34	0,1601	0,0201	56	0,18	845,8	0,057
35	0,1426	0,0159	62,3	0,14	1069	0,045
36	0,127	0,0127	69	0,1	1338	0,036
37	0,1131	100	78	0,089	1700	0,028
38	0,1007	0,0079	82,3	0,07	2152	0,022
39	0,0897	0,0063	97,5	0,056	2696	0,017
40	0,0799	0,005	111	0,044	3400	0,014
41	711	0,004	126,8	0,035	4250	0,011
42	0,0633	0,0032	138,9	0,028	5312	0,009
43	0,0564	0,0025	156,4	0,022	6800	0,007
44	0,0503	0,002	169,7	0,018	8500	0,005

Fonte: Tabela de Conversão - AWG - UFRGS.

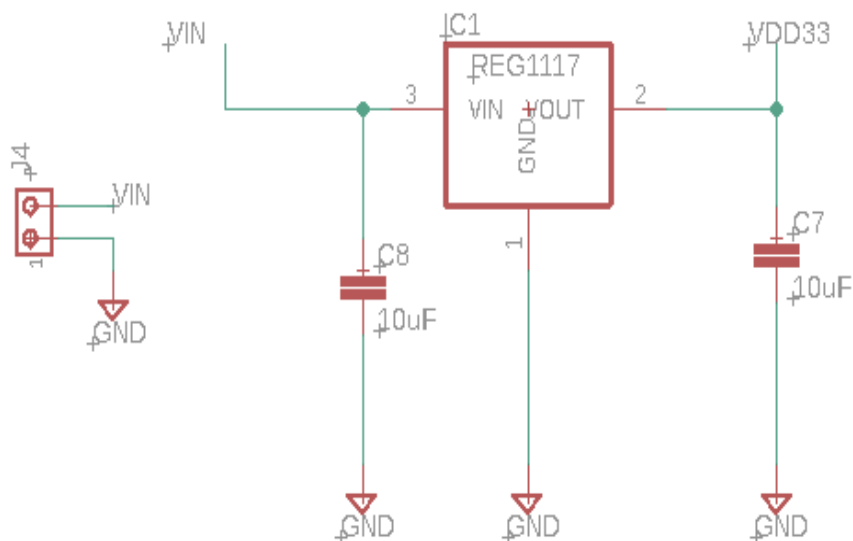
3.1.3 ESQUEMÁTICO DO PROJETO

3.1.3.1 Circuito de entrada

O circuito abaixo (Figura 20), tem por função receber a tensão de entrada e regular para 3,3V, alimentando, assim, o restante do circuito. É composto principalmente pelo circuito integrado SMD 1117.

Os capacitores, assim com seus valores, foram incluídos devido à sugestão de aplicação típica presente no datasheet.

Figura 20. Esquemático do circuito de entrada.



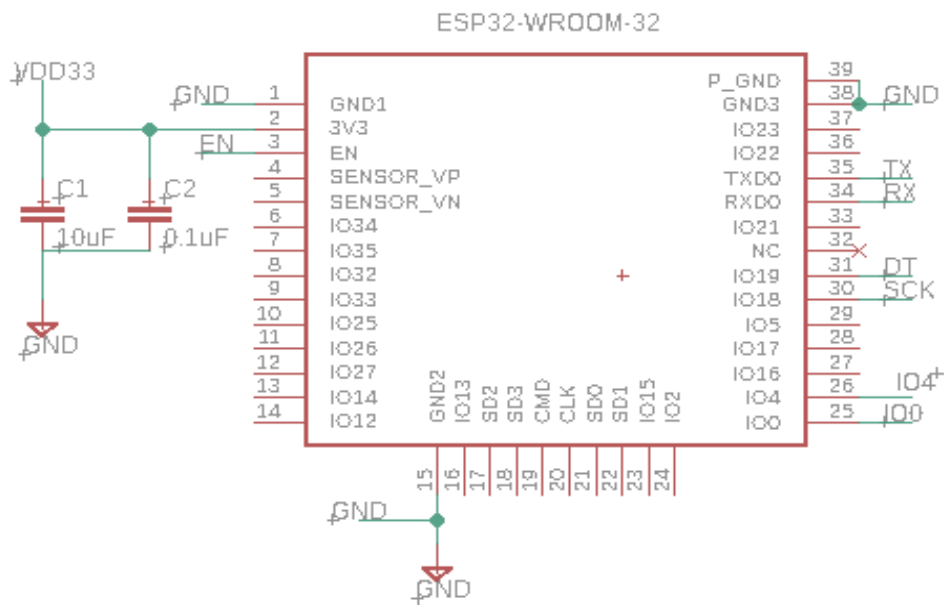
Fonte: Autoria própria.

3.1.3.2 Circuito para o microcontrolador

Os valores de C1 e C2 são indicados no datasheet e servem como capacitores de desacoplamento, ou seja, filtram os ruídos da faixa de alimentação.

Os valores de R6 e C9 também são indicados no datasheet e tem a finalidade de garantir o fornecimento de energia durante a inicialização (Figura 21).

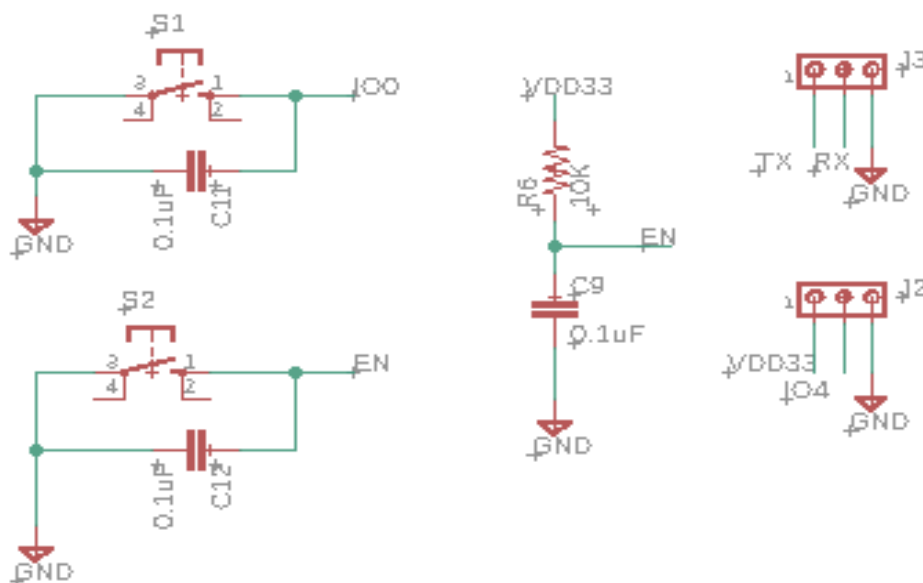
Figura 21. Circuito para o microcontrolador.



Fonte: Autoria própria.

O conector J3 é responsável pela ligação entre o microcontrolador e dispositivo utilizado para gravar o firmware, e o conector J2 une o microcontrolador ao sensor de gás MQ-2 (Figura 22).

Figura 22. Circuito do microcontrolador e seus periféricos.



Fonte: Autoria própria.

Os valores de C1 e C2 são indicados no datasheet e servem como capacitores de desacoplamento, ou seja, filtram os ruídos da faixa de alimentação, conforme MARTO (2016). Os valores de R6 e C9 também são indicados no datasheet e tem a finalidade de garantir o fornecimento de energia durante a inicialização.

O conector J3 é responsável pela ligação entre o microcontrolador e dispositivo utilizado para gravar o firmware. Já o conector J2 une o microcontrolador ao sensor de gás MQ-2.

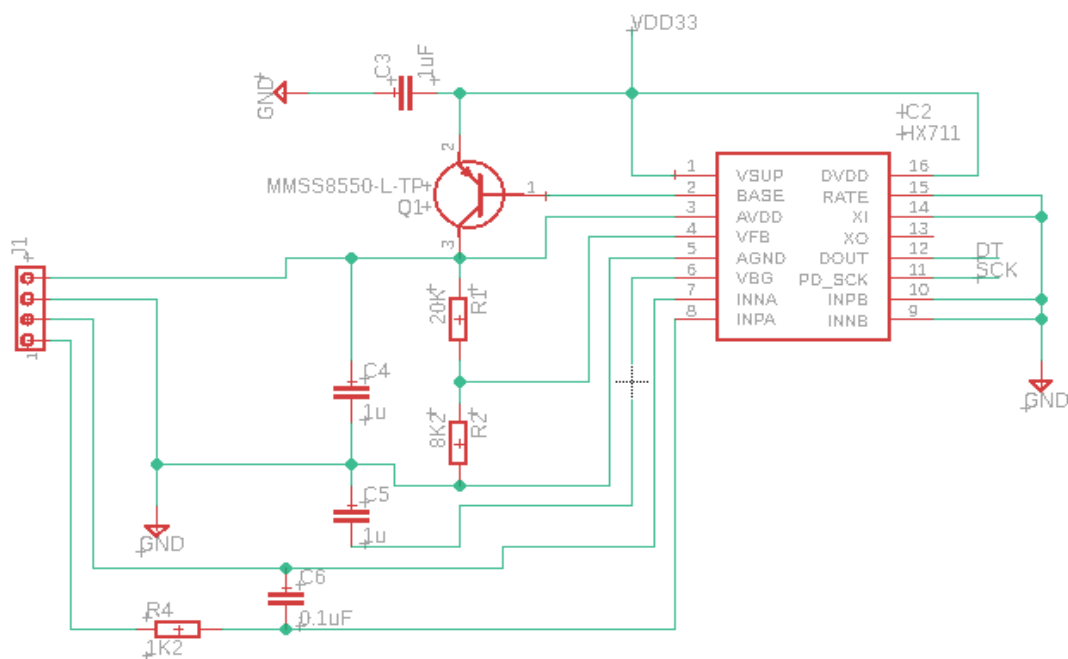
Foram adicionados ainda dois botões: um para reinicializar o microcontrolador, e outro para pôr o ESP-32 em modo de gravação. Tanto para resetar quanto para gravar os pinos EN e IO0 devem estar em nível lógico zero, respectivamente.

Os capacitores C11 e C12 foram incluídos de acordo com esquemático do Kit de desenvolvimento da Espressif, fabricante do chip ESP-32 (ESP32WROOM_32 Peripheral Schematics, 2021).

3.1.3.3 Circuito para o HX711

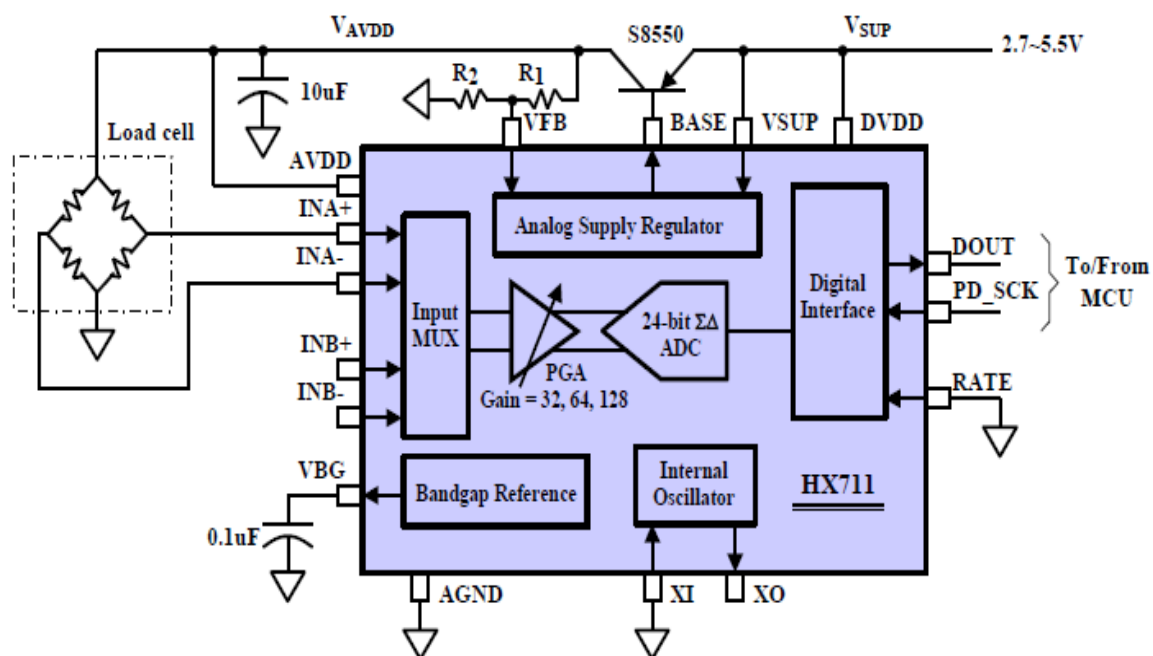
O circuito utilizado (Figura 23) foi o mesmo do módulo HX711 (Elecrow), utilizado para testes. Este circuito baseia-se no sugerido pelo datasheet (Figura 24).

Figura 23. Circuito utilizado para HX711 e periféricos.



Fonte: Autoria própria.

Figura 24. Circuito sugerido pelo datasheet.

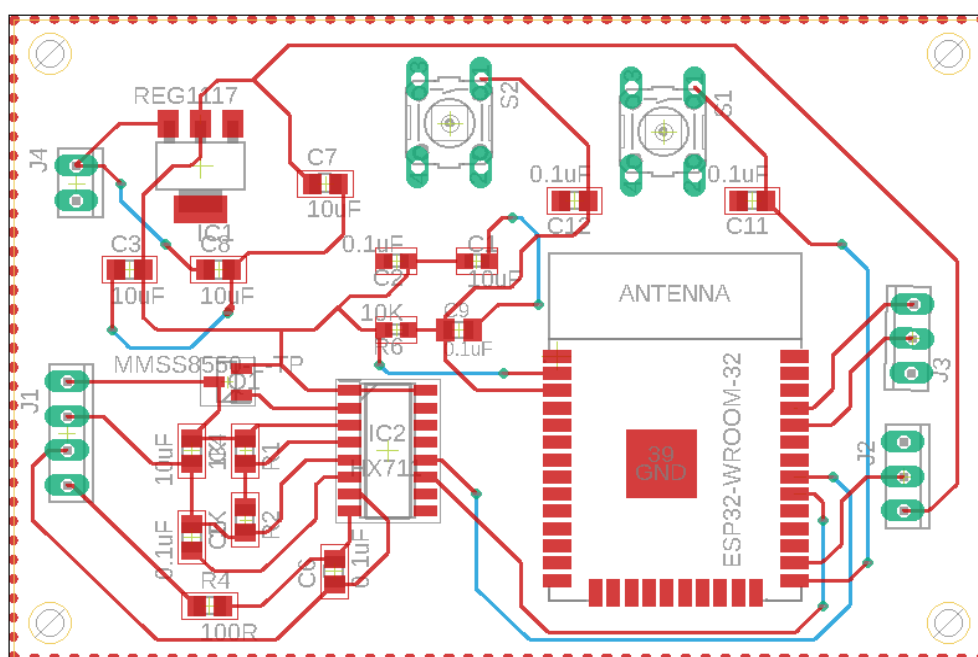


Fonte: Datasheet HX711.

3.1.3.4 Roteamento

Para assegurar um bom funcionamento e durabilidade, o projeto deve possuir um excelente desenho da placa de circuito impresso (PCI), com espessura de trilhas e disposição dos componentes seguindo algumas regras para garantir que o diagrama esquemático se comporte como desejado (LIMA, 2010). A placa do projeto AWA possui dois *layers*, com espessura de cobre de 0,035 mm (1 onça) e a espessura total é de 1,57 mm. Optou-se por esses valores por serem valores típicos para a produção da placa (Figura 25).

Figura 25. Roteamento da placa.



Fonte: LIMA, 2010.

Outra questão importante no desenho das trilhas é a distância entre trilhas adjacentes ou entre pontos eletricamente distintos. A largura de trilha é de 0,254 mm (10 mils). Para esse valor de espessura de trilha e espessura de cobre a corrente é de 0,8 A, conforme a Tabela 07.

Tabela 07. Largura mínima da trilha de acordo com a corrente.

Largura da trilha (cobre)	Corrente (A)	
	1 oz	2 oz
5 mils	0,5	0,7
10 mils	0,8	1,4
20 mils	1,4	2,2
30 mils	1,9	3
50 mils	2,5	4
100 mils	4	7

Fonte: LIMA, 2010.

Também foi implantado um plano “terra”. A finalidade desta técnica é diminuir substancialmente as interferências de modo comum e diferencial, já que múltiplos pontos de aterramento não permitem o acoplamento de impedância comum.

Dessa forma, conforme afirma o CUTRI (2005), o ruído associado ao loop de terra é atenuado, pois a baixa resistência e indutância do plano terra reduzem a impedância da fonte.

3.1.4 FABRICAÇÃO E MONTAGEM DA PLACA

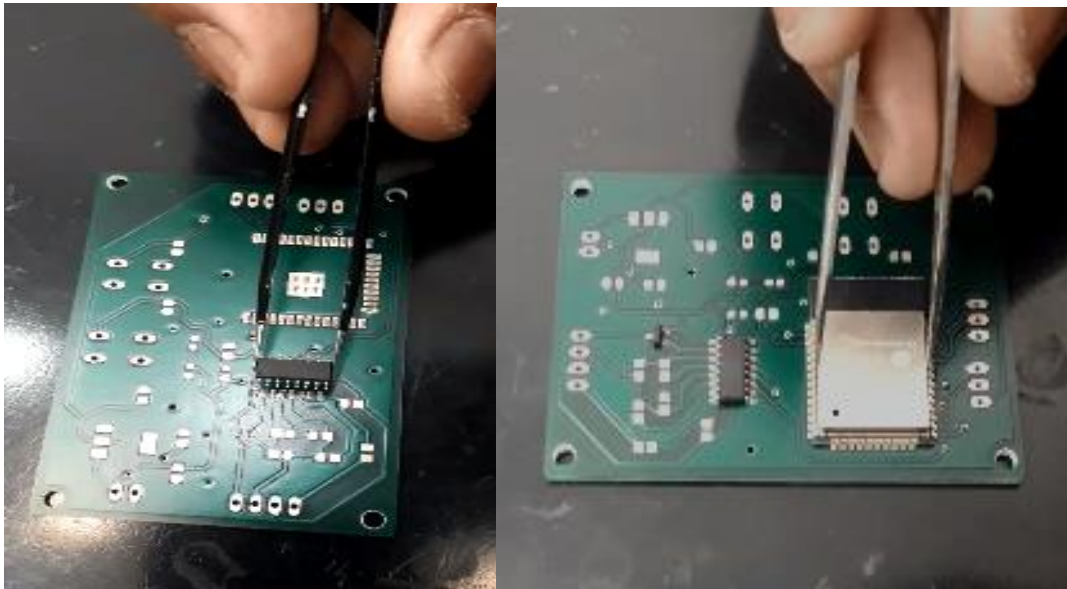
A aprovação da proposta foi condicionada, pela Comissão de TCC, ao desenvolvimento da placa do circuito impresso do sistema.

A placa de circuito impresso foi fabricada no laboratório EFAB administrado pelo Prof.Dr. Fernando Cardoso Castaldo e foi montada na APTAMAX - Indústria e Comércio de Produtos Eletrônicos Eireli.

Inicialmente foi realizada a fabricação da placa eletrônica, a partir de um desenvolvimento do projeto da PCI. A placa de cobre crua passou por tratamentos com produtos químicos, pintura e revelação das trilhas, e novos banhos com cobre, níquel, estanho e banho de prata.

Para a montagem da placa, foram depositados em seus respectivos pads, pasta de solda composta de Sn63/Pb37, a fim de posicionar os componentes, conforme demonstrado na figura 26.

Figura 26. Posicionamento dos componentes.



Fonte: Autoria própria.

Após a inserção dos componentes, a placa foi colocada em um forno de refusão (Figura 27) com temperatura controlada para realizar a solda.

Figura 27. Entrada da placa no forno de refusão.



Fonte: Autoria própria.

A utilização do forno de refusão (Figura 28) é muito importante, pois evita soldas frias ou calor em excesso na placa, não comprometendo a soldagem de seus componentes.

Figura 28. Forno de refusão.



Fonte: Autoria própria.

Após saída da placa do forno de refusão, foi realizada a inspeção visual da peça (figura 29), a fim de garantir que a placa não possui nenhum problema de solda.

Figura 29. Inspeção visual da placa.



Fonte: Autoria própria.

3.1.5 FIRMWARE - ESP 32

O programa do microcontrolador ESP32 foi desenvolvido através da IDE do Arduino e foi necessário a instalação de um driver para a IDE do Arduino reconhecer a placa ESP32, utilizamos uma publicação do K TECNOLOGIA (2018) Tecnologia para tal configuração.

Foi utilizado a biblioteca HX711master, disponibilizada pelo fabricante, que realiza a interface do conversor analógico-digital (ADC) do módulo HX711 para leitura das células de carga (BIBLIOTECA HX711 MASTER).

Após alguns testes e o entendimento do funcionamento da biblioteca HX711master, foi possível o incremento do código com as nossas condições especificadas anteriormente:

- ✓ Conexão com o Wi-Fi;
- ✓ Conexão com o Firebase;
- ✓ Análise dos dados recebidos pelos sensores.

A conectividade do WI-FI é realizada através da BIBLIOTECA WI-FI MANAGER-ESP32, que é um gerenciador de conexão Wi-fi com portal de configuração web. Para o correto funcionamento, foi necessário a instalação da biblioteca de servidor DNS e Web, fornecidas pelo mesmo criador (BIBLIOTECA WI-FI MANAGER-ESP32)

Utilizamos a BIBLIOTECA FIREBASE ESP32 CLIENT tanto para conexão, envio de dados e envio de notificações para o Firebase. Após as conexões, a balança realiza uma calibração da sua tara, através da função “.tare()” da biblioteca HX711.

As configurações do tipo do botijão e da sua tara são inseridas pelo usuário através do aplicativo e são buscadas pelo microcontrolador na base do Firebase.

Após todas as configurações e o usuário autorizar o início da leitura pelo aplicativo, o ESP32 dará início a pesagem realizando cerca de 200 medidas,

calcula a média e o percentual de gás ainda presente no botijão, e envia ao banco de dados do Firebase. Se a porcentagem for igual a 50%, 25% ou abaixo de 10% é enviado uma notificação para smartphone.

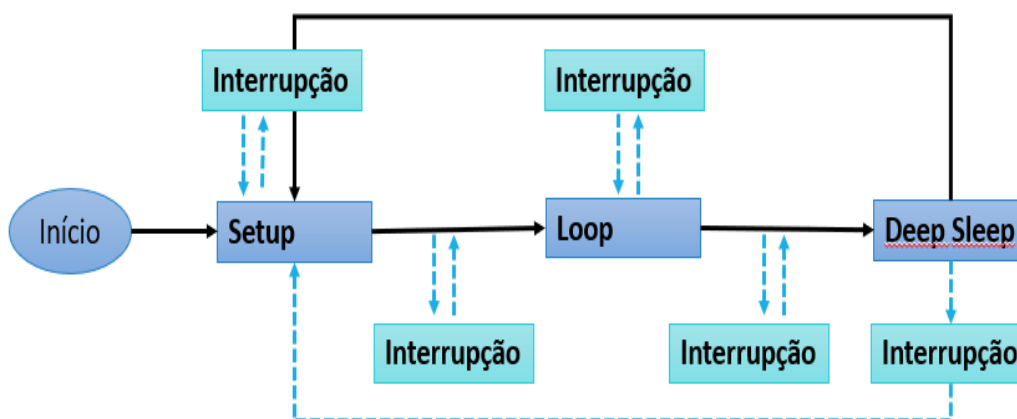
Além disso, temos uma interrupção que monitora a todo momento, qualquer sinal do GPIO que está acoplado no sensor MQ-6, a fim de avisar qualquer sinal de perigo devido a vazamentos.

A fim de economizar energia, após cada leitura, o microcontrolador entrará em sono profundo, através da função *deep sleep*. Neste modo apenas o controlador RTC, periféricos RTC (incluindo coprocessador ULP) e memória RTC ficam ativos. O despertar do microcontrolador, pode ser realizado de 5 maneiras:

- ✓ Timer;
- ✓ External wakeup (ext0);
- ✓ External wakeup (ext1);
- ✓ ULP coprocessor wakeup;
- ✓ Touchpad.

Neste projeto utilizamos o despertar por Timer e por External wakeup (ext0). Nas figuras 30, 31 e 32, os diagramas descrevem o funcionamento geral do programa.

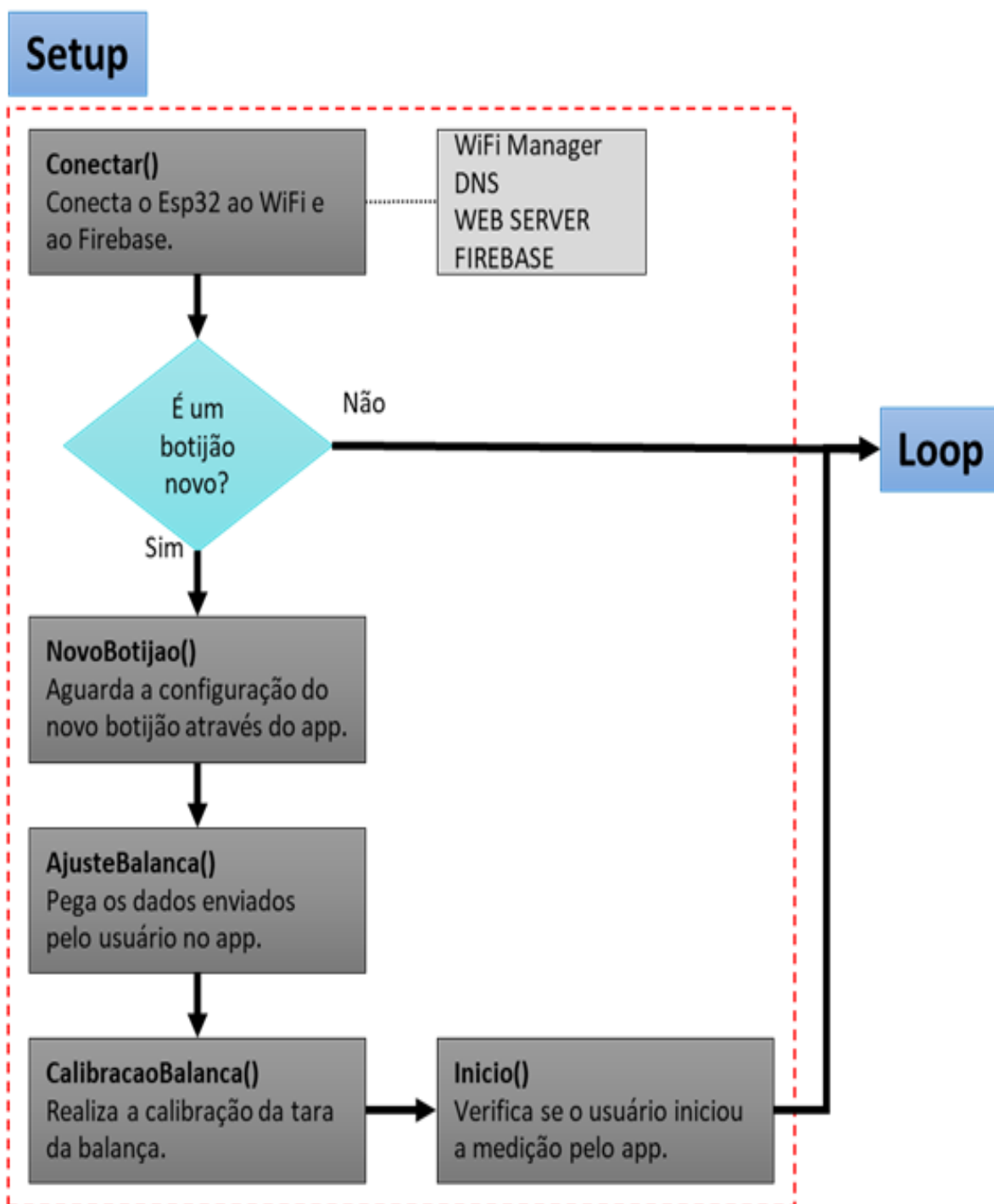
Figura 30. Visão macro de funcionamento do sistema AWA.



Fonte: Autoria própria.

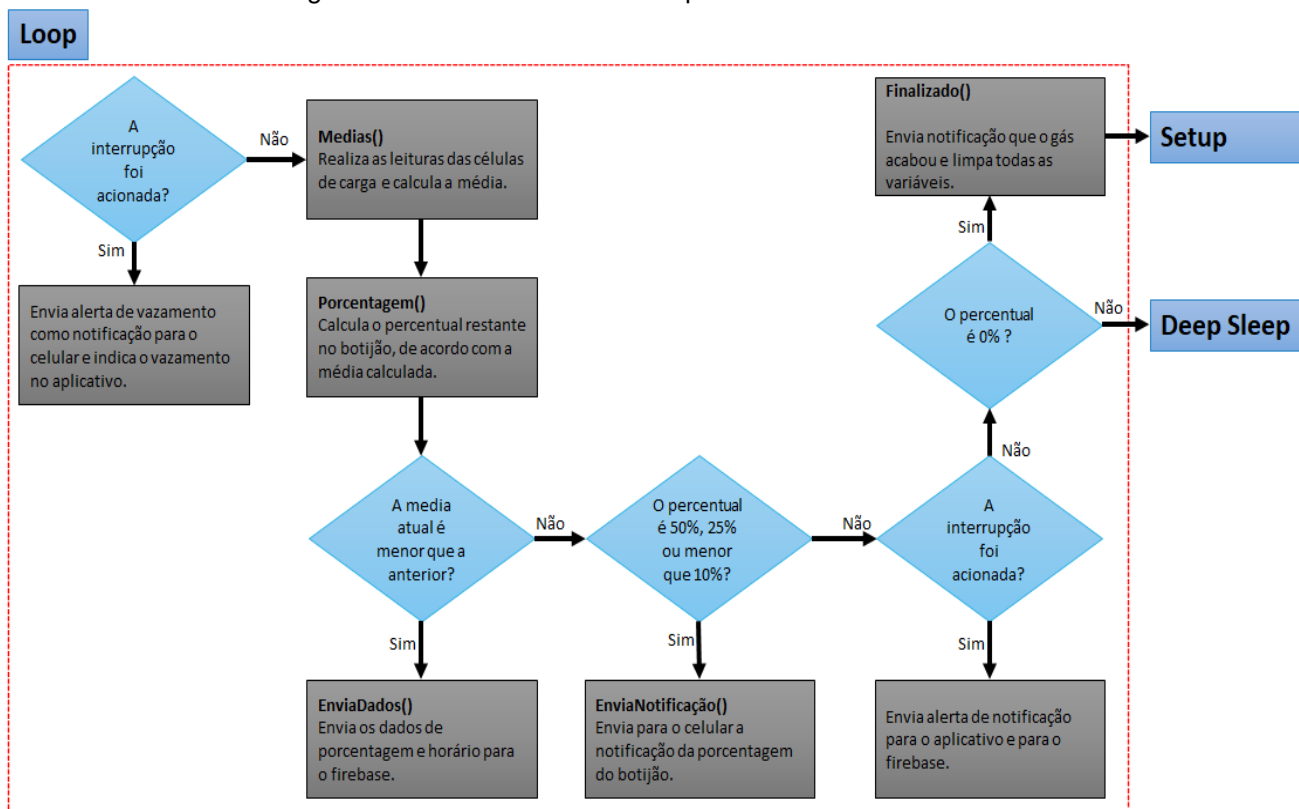
A figura acima, mostra a visão macro de funcionamento do firmware desenvolvido para o ESP32. Pode ser observado que a interrupção realizada pelo sensor MQ-6 pode ser ativada em todos os estágios, setup, loop ou deep sleep, porém neste último, a interrupção retorna para o setup.

Figura 31. Funcionamento do Setup do firmware.



Fonte: Autoria própria.

Figura 32. Funcionamento do Loop do firmware.



Fonte: Autoria própria.

O código completo e as bibliotecas utilizadas no projeto do microcontrolador podem ser acessados no Github (AWA-SISTEMA DE MONITORAMENTO, 2021).

3.2 APLICATIVO

3.2.1 DESENVOLVIMENTO DO APLICATIVO

O desenvolvimento do aplicativo tem como principal função de interagir com o usuário, conforme listado abaixo:

- ✓ Selecionar o tipo do botijão de gás (P05 - P13 - P45);

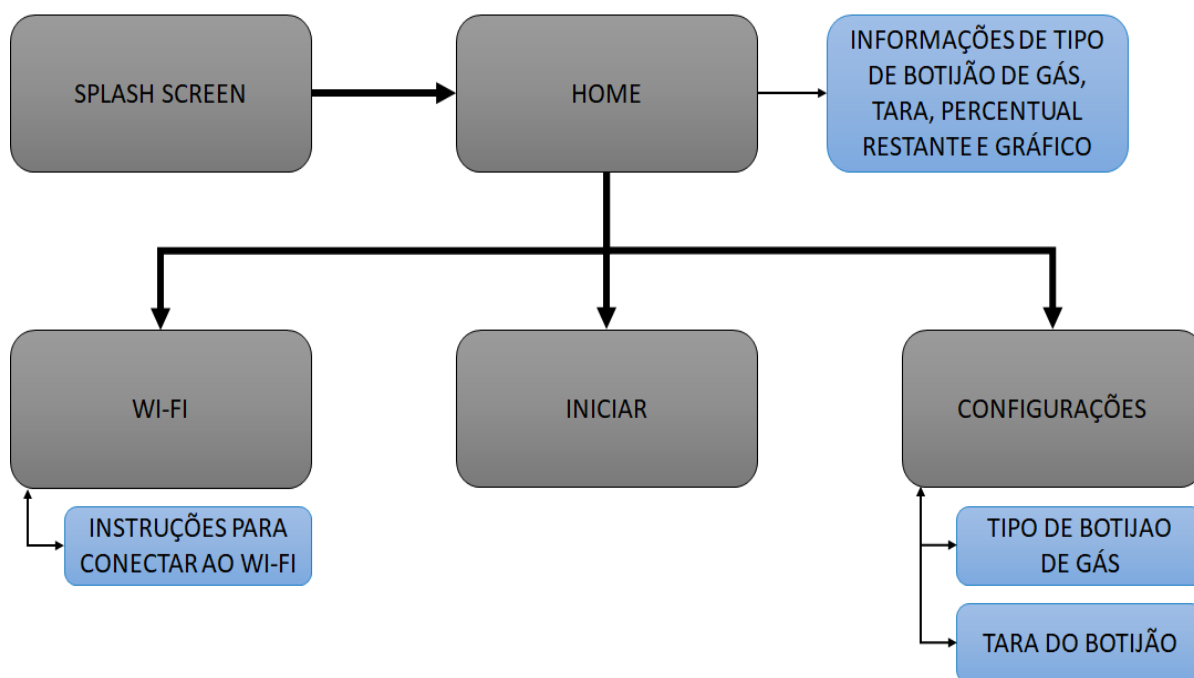
- ✓ Inserir a tara do botijão de gás;
- ✓ Configurar a rede Wi-fi do dispositivo (balança);
- ✓ Verificar a quantidade em percentual de gás restante;
- ✓ Acompanhar o consumo de gás durante os dias, através de um gráfico;
- ✓ Receber notificações/alerta de gás restante de 50%, 25% e inferior a 10%;
- ✓ Receber notificações/alerta de vazamento de gás.

O aplicativo foi desenvolvido para a plataforma Android, uma plataforma desenvolvida pela Google voltada para dispositivos móveis, totalmente aberta e livre (*Open Source*). Inicialmente o sistema Android foi desenvolvido pelo Google e atualmente essa plataforma é mantida pela OHA (*Open Handset Alliance*), conforme explica SILVA (2015).

Para fazer a programação do aplicativo em java foi utilizado o ambiente de desenvolvimento integrado (IDE) Android Studio. Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995. É a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na Web e softwares corporativos (JAVA, 2021). Já o Android Studio é baseado no IntelliJ IDEA, Ambiente de desenvolvimento integrado escrito em JAVA para o desenvolvimento de software de computador, oferecendo mais recursos para aumentar a produtividade na criação de um aplicativo Android.

A seguir é ilustrado o mapa de navegação do aplicativo, que é o caminho pelo qual o usuário pode percorrer entre as telas (figura 33).

Figura 33. Mapa de navegação do aplicativo.

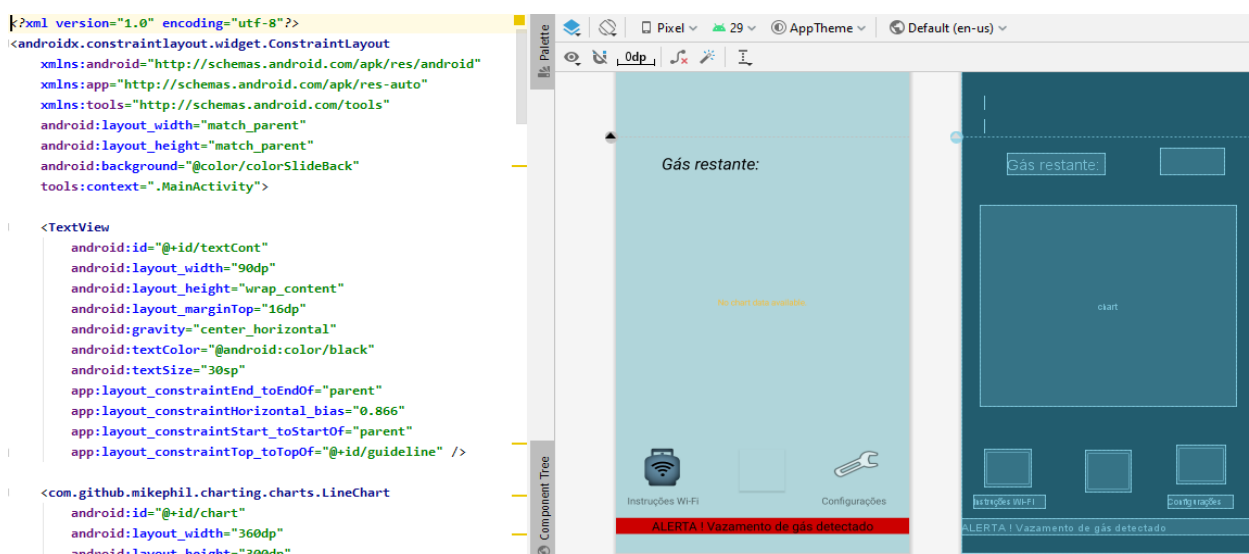


Fonte: Autoria própria.

Para compreender o desenvolvimento do aplicativo é importante conhecer o que é uma *Activity*, o qual é a tela na aplicação, responsável por definir e exibir um layout com componentes de interface de usuário (*views*), como botões, textos e imagens, por exemplo, e controlar os eventos recebidos, como destaca MOTTA (2019). É nesta interface que o usuário interage com o aplicativo. E para organizar e posicionar os componentes há algumas opções de Layout, entretanto a opção escolhida para as *activitys* foi a *ConstraintLayout*. Nesse tipo de layout os componentes de interface são posicionados a partir de uma referência criada entre os demais componentes.

A seguir (figura 34) é exibido a janela referente à *activity* principal. Nela é possível observar, na aba direita, como os componentes são posicionados. E na aba esquerda, o código xml, no qual também podem ser adicionados e configurados os botões, textos e imagens da *activity*.

Figura 34. Activity_main.xml.



Fonte: Autoria própria.

O código completo do aplicativo pode ser visto no Github AWA-SISTEMA DE MONITORAMENTO, 2021.

3.2.1.1 Gráfico

No aplicativo, o gráfico é utilizado para facilitar a visualização do consumo de gás durante um período, constituindo um importante componente no aplicativo. Para a construção do gráfico foi utilizada a BIBLIOTECA MPANDROID CHART, desenvolvida por Philipp Jahoda, e a versão utilizada foi a 3.0.2.

Esta biblioteca é responsável por receber o timestamp e a porcentagem restante do gás, e plotar nos eixos 'x' e 'y'. O timestamp é convertido para o formato 'dd/MM/yy' através da classe HourAxisValueFormatter, e é visualizado no eixo 'x' do gráfico. Já a porcentagem de gás é visualizada no eixo 'y'.

O gráfico foi inserido na *activity* principal (*activity_main.xml*). E para facilitar a interação com o usuário foi usado um marcador sobre o ponto do gráfico que o usuário pressiona. Assim que pressionado, um marcador surge

com as informações da data e da porcentagem de gás restante. Para isso, foi implementado uma outra *activity* e uma classe responsável por inserir e atualizar os dados nas posições corretas.

3.2.1.2 Configuração Wi-fi

Para realizar a configuração Wi-fi da balança o usuário deve seguir os passos descritos do *slide*, que é acessado através de um *button* na *activity* principal. A biblioteca utilizada para implementar a classe *slide* foi BIBLIOTECA MATERIAL-INTRO, versão 2.0.0, desenvolvida por Jan Heinrich Reimer.

Para implementá-lo foi criado uma *activity* chamada *activity_slide*. E na classe *Slide* foram configurados as mensagens e imagens que devem aparecer em cada passo (Figura 35).

Figura 35. Classe slide.java.

```

setButtonBackVisible(false);
setButtonNextVisible(false);

addSlide(new SimpleSlide.Builder().title("Conectando sua balança ...").
    description("Ligue sua balança e siga os passos a seguir").
    image(R.drawable.gas_image2).
    background(R.color.colorSlideBack).build());

addSlide(new SimpleSlide.Builder().
    title("1º Passo").description("Esteja a menos de 2 metros da balança").
    image(R.drawable.gas_image2).
    background(R.color.colorSlideBack).build());

addSlide(new SimpleSlide.Builder().
    title("2º Passo").description("Conecte-se à rede 'AWA', com senha '12345678' ").
    image(R.drawable.rede_senha).
    background(R.color.colorSlideBack).build());

addSlide(new SimpleSlide.Builder().
    title("3º Passo").description("Abra o navegador e digite o endereço '192.168.4.1' ").
    image(R.drawable.url_wifi_manager).
    background(R.color.colorSlideBack).build());

addSlide(new SimpleSlide.Builder().
    title("4º Passo").description("Escolha qual rede você deseja conectar sua balança ").
    image(R.drawable.wifi_scan).
    background(R.color.colorSlideBack).build());

```

Fonte: Autoria própria.

3.1.2.3 Ajuste

Para configurar o tipo de gás e a tara do botijão foi criado uma *activity*, composta por:

- ✓ Dois *TextView* que mostram as mensagens para selecionar e inserir o tipo de gás e a tara do botijão;
- ✓ Um *RadioGroup* com dois *RadioButton*, responsáveis pela seleção de apenas uma opção de tipo de gás;
- ✓ Um *EditText*, responsável por adquirir a tara do botijão. Possui entrada apenas para números e vírgula;

- ✓ Um *Button*, responsável por salvar os dados no Firebase, e voltar para *activity* principal com a passagem dos dados.

Já na classe ajuste, foram realizados os procedimentos para passagem dos dados para *activity* principal através de uma *Intent*, e o envio dos dados do tipo do gás e valor da tara do gás para o Firebase.

Além disso, foi implementado um *Toast*, que após o botão de “SALVAR DADOS” ser pressionado, verifica se um dos campos está vazio. E assim exibe uma mensagem informando que todos os campos devem ser preenchidos (figura 36). Para a seleção do tipo do gás, o usuário deve selecionar obrigatoriamente entre 3 opções (P05, P13 e P45). Para a tara do botijão, o usuário insere os dados que é informado no vasilhame. Essa entrada de dados permite apenas que o cliente insira números e vírgula; assim, limitando a possibilidade inserir dados não condizentes com a tara do botijão.

Figura 36. Código executado quando o button ‘SALVAR DADOS’ é pressionado.

```
public void Salvar(View view) {
    EditText tara = findViewById(R.id.editTextTara);
    valorTara = tara.getText().toString();
    ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo netInfo = cm.getActiveNetworkInfo();

    if (netInfo != null && netInfo.isConnectedOrConnecting()) {

        if (tipoGas == null || valorTara == null || valorTara.equals("")) {
            Toast.makeText(getApplicationContext(), text: "Preencha todos os campos", Toast.LENGTH_LONG).show();
        } else {
            valorTaraFloat = Float.parseFloat(valorTara);
            referenciaUser.child("TaraBotijao").setValue(valorTaraFloat);
            referenciaUser.child("TipoBotijao").setValue(tipoGas);
            referenciaUser.child("NovoBotijao").setValue(1);
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            intent.putExtra( name: "tipoGas", tipoGas);
            intent.putExtra( name: "tara", valorTara);
            Toast.makeText(getApplicationContext(), text: "Dados salvos com sucesso", Toast.LENGTH_LONG).show();
            startActivity(intent);
        }
    }
    else{
        Toast.makeText(getApplicationContext(), text: "É necessário conectar à internet", Toast.LENGTH_LONG).show();
    }
}
}
```

Fonte: Autoria própria.

3.1.2.4 Tela principal

A tela principal, é a tela que surge logo após o *splash screen* (tela de abertura de um aplicativo) ser exibido. E é composta por:

- ✓ Cinco *TextView*, dos quais dois são responsáveis por exibir o tipo de gás e a tara do botijão, dois são responsáveis por exibir a quantidade de gás restante logo acima do gráfico, e um por exibir um alerta quando um vazamento de gás é detectado;
- ✓ Três *Button*, um responsável por acessar a tela de ajustes, outro pela tela de configuração do Wi-fi da balança, e outro para comunicar à balança que ela já pode iniciar o envio dos dados;
- ✓ Um *Chart*, é o gráfico responsável por exibir os dados da porcentagem de gás restante em um período;
- ✓ Também foi utilizado um *Guideline* para auxiliar a posicionar os componentes acima.

A classe principal é responsável por:

- ✓ Receber os dados pelo Firebase, através do método 'addValueListener'. Este método verifica se há alteração dos dados (evento) e recebe os valores. Foram utilizados para receber os valores de porcentagem de gás, timestamp e alerta de gás ('0' ou '1');
- ✓ Gerar o *Token*, através do método 'instanceIdResult.getToken'. O *Token* é necessário para realizar as notificações 'push'.
- ✓ Enviar o valor do Token e quando pressionado o *Button* 'Iniciar', enviar o valor '1'.
- ✓ Armazenar localmente os dados do *Token*, tara do botijão e tipo do gás. Foram realizados através do recurso 'SahredPreferences' (Figura 37). A utilização de um banco de dados local foi necessária, pois quando o aplicativo é reiniciado esses dados são utilizados para exibir na tela principal o valor da tara do botijão e do tipo de gás selecionados. Já o *Token* é enviado para o Firebase.

- ✓ Outro recurso muito importante foi o de ativar persistência em disco, o que significa que mesmo quando o smartphone não tiver acesso à internet, o aplicativo mantém o estado. Pois, os dados são gravados localmente no dispositivo. Isso pode ser feito através do método 'setPersistenceEnabled(true)'.

Figura 37. Recebimento dos dados da activity ajuste, armazenamento de dados e recuperação.

```
public void recuperarDadosAjuste(){
    SharedPreferences preferences = getSharedPreferences("ARQUIVO_PREFERENCIA", mode: 0);
    SharedPreferences.Editor editor = preferences.edit();

    tipoGas_local = preferences.getString( key: "tipoGas", defValue: "zero");
    tara_local = preferences.getString( key: "tara", defValue: "zero");

    if(tipoGas_local=="zero" && tara_local=="zero") {
        valorTipoGas.setText("Selecione o tipo de gás e a tara !");
        valorTara.setText("");
    }else{
        valorTipoGas.setText("Tipo de gas: " + tipoGas_local);
        valorTara.setText("Tara do butijão: " + tara_local + " g");
    }

    Bundle dados = getIntent().getExtras();
    if(dados != null) {
        valorTipoGas.setText("Tipo de gas: " + dados.getString( key: "tipoGas"));
        valorTara.setText("Tara do butijão: " + dados.getString( key: "tara") + " Kg");
        editor.putString("tipoGas", dados.getString( key: "tipoGas"));
        editor.putString("tara", dados.getString( key: "tara"));
        editor.commit();
    }else{
        //valorTipoGas.setText("Selecione o tipo de gás e a tara !");
    }
}
```

Fonte: Autoria própria.

3.1.2.5 Notificações

Para o recebimento das notificações foi criada a classe 'MyFirebaseMessagingService', que é responsável por tratar as notificações que são enviadas pelo ESP32. Nesta classe, é sobrescrito o método 'onNewToken', chamado durante a comunicação para criar o Token, necessário para enviar a notificação pelo ESP32. Na classe principal o Token é recuperado e enviado para o ESP32 através do Firebase (Figura 38).

Figura 38. Recuperação do Token e envio para o Firebase.

```
public void recuperarToken(){
    FirebaseInstanceId.getInstance().getInstanceId().addOnSuccessListener(new OnSuccessListener<InstanceIdResult>() {
        @Override
        public void onSuccess(InstanceIdResult instanceIdResult) {
            referenciaUser.child("Token").setValue(instanceIdResult.getToken());
        }
    });
}
```

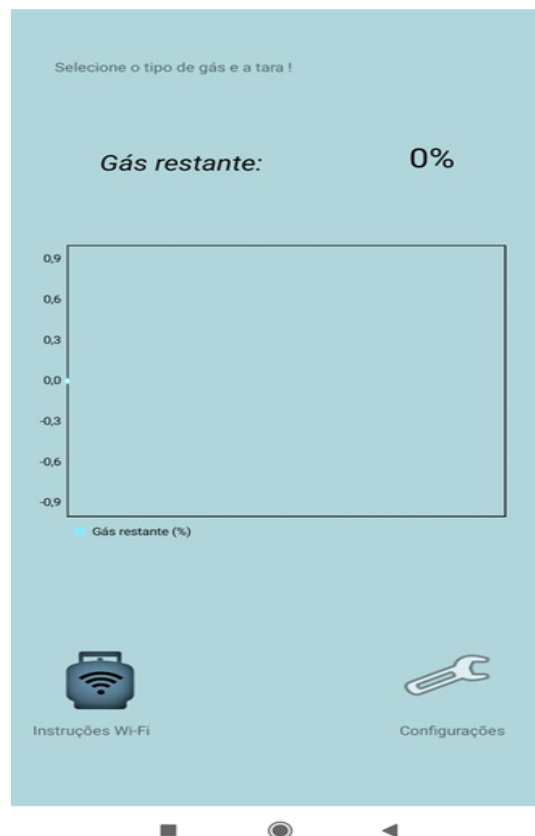
Fonte: Autoria própria.

3.2.2 FUNCIONAMENTO DO APLICATIVO

Após realizada a instalação do aplicativo, é imprescindível que seja configurada a rede Wi-fi na qual a balança irá se conectar, pois somente com a rede configurada a balança conectará à rede e iniciará a comunicação com o servidor do Firebase.

Para configurá-la basta clicar no ícone de botijão com o símbolo de rede Wi-fi, no canto inferior esquerdo, conforme indicado na Figura 39.

Figura 39. Ícone de Botijão de Gás para instruções de como se conectar à rede Wi-fi.

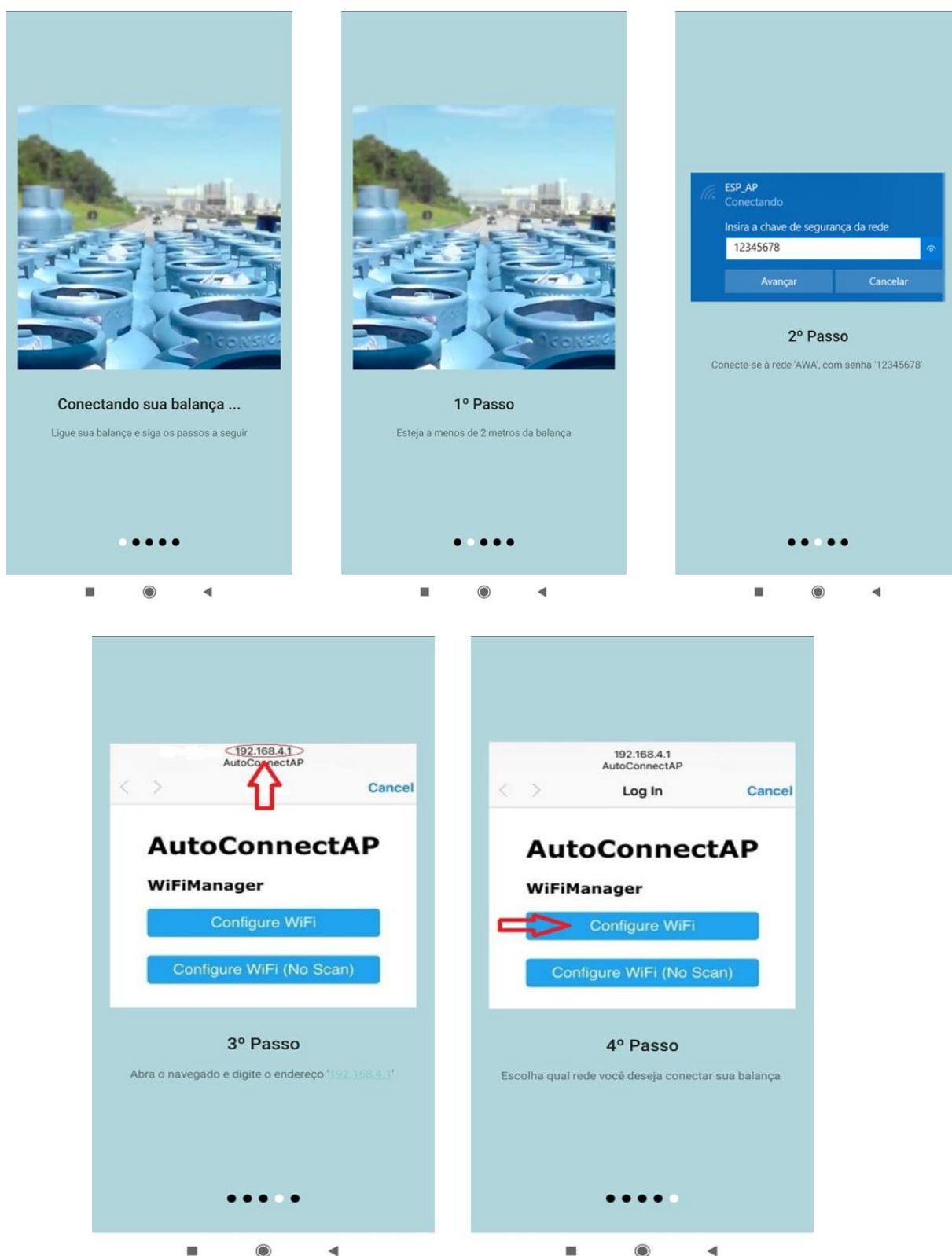


Fonte: Autoria própria.

Então, abrirá em uma nova tela um slide, contendo os passos a seguir, conforme apresentado na Figura 40.

O Slide pode ser definido como um componente que utiliza o ato de deslizar na tela como recurso de interação. Neste caso, o ato de deslizar ocasiona a passagem entre telas.

Figura 40. Passo a passo para a conexão do aplicativo no Wi-fi.



Fonte: Autoria própria.

Na tela principal do aplicativo, na parte superior, é mostrado o tipo de gás e a tara em quilogramas, demonstrado na figura 41. Se ainda não foi selecionado o tipo de gás e a tara, aparecerá uma mensagem para inserir os dados.

Figura 41. Inclusão de dados no aplicativo.

A imagem mostra a interface de usuário de um aplicativo em uma tela com fundo azul claro. No topo, o texto "Selecione o tipo de gás:" precede três opções de seleção por rádio: "P05", "P13" e "P45". Abaixo, o texto "Digite a tara do botijão (Kg):" está seguido por uma linha horizontal para entrada de texto. Um botão retangular com o texto "SALVAR DADOS" está centralizado. Abaixo do botão, há um ícone de um triângulo invertido com um ponto de exclamação. No rodapé, há duas linhas de texto em itálica: "Atente-se ao prazo de validade da mangueira de gás e do regulador de pressão." e "Está impresso numa tarjeta amarela, com o código de segurança NBR-8613." Na base da tela, os ícones de navegação padrão de um sistema Android são visíveis.

Fonte: Autoria própria.

Para inserir os dados basta clicar no ícone de “ferramenta”, no canto inferior direito. Assim que clicado, abrirá uma nova tela para serem realizadas as devidas configurações. Inicialmente, é necessário selecionar o tipo de gás entre P05, P13 e P45. Após isso é necessário digitar a tara do botijão de gás (em quilogramas).

Depois de inserido os dados, é preciso ainda clicar no botão “SALVAR DADOS”, pois só assim os dados serão enviados para o servidor do Firebase, para que o microcontrolador possa recuperar esses dados e enviar o percentual de gás restante corretamente.

O celular deverá estar conectado à rede de internet para este envio de dados, caso contrário, uma mensagem aparecerá na tela, conforme a figura 42.

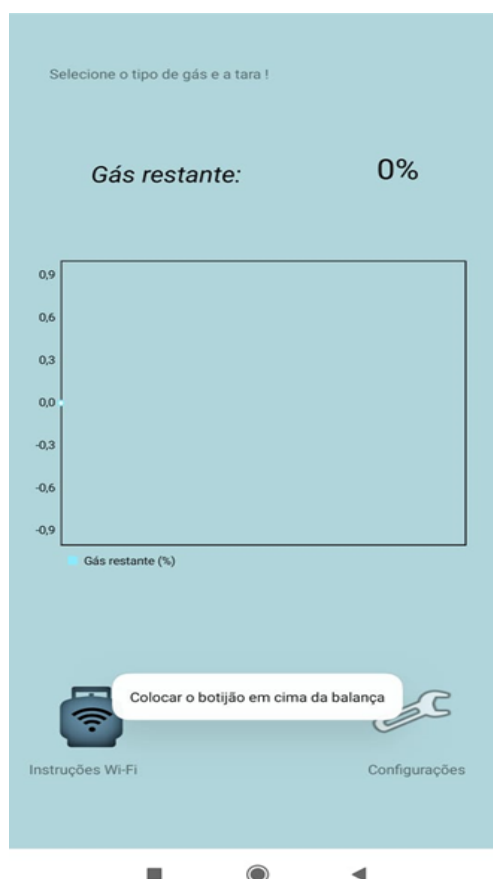
Figura 42. Mensagem de erro devido a não conectividade com a internet.



Fonte: Autoria própria.

Concluído os passos anteriores, é necessário esperar que a mensagem informando para colocar o botijão na balança apareça, conforme a figura 43.

Figura 43. Mensagem informativa para colocar o botijão em cima da balança.

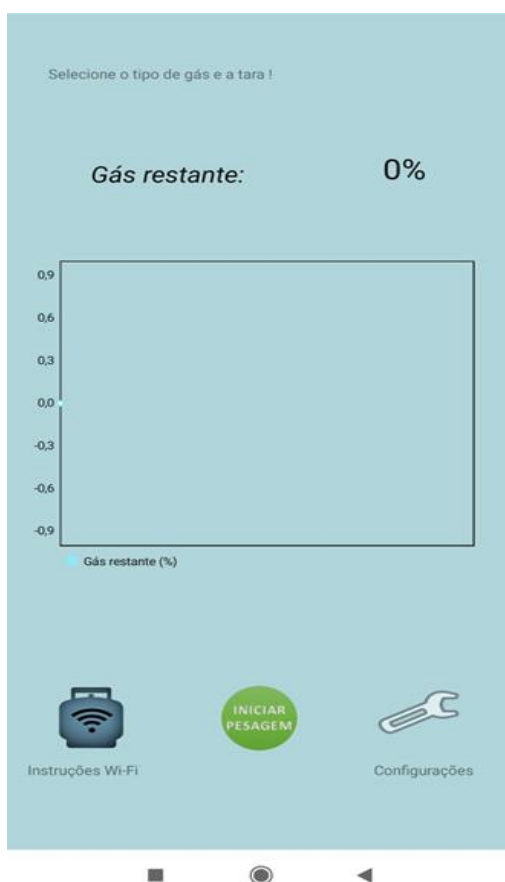


Fonte: Autoria própria.

Após alguns segundos a mensagem some da tela e aparecerá o ícone “iniciar”, localizado entre os ícones de “ferramenta” e “botijão com símbolo de Wi-fi”, assim, basta o usuário clicar neste ícone para a balança iniciar a pesagem do gás, e o posterior envio de dados.

A porcentagem de gás restante atual poderá ser visualizada acima do gráfico, conforme figura 44. Logo abaixo, no gráfico, é possível acompanhar a porcentagem de gás restante através dos dias.

Figura 44. Botão iniciar ativado na tela.

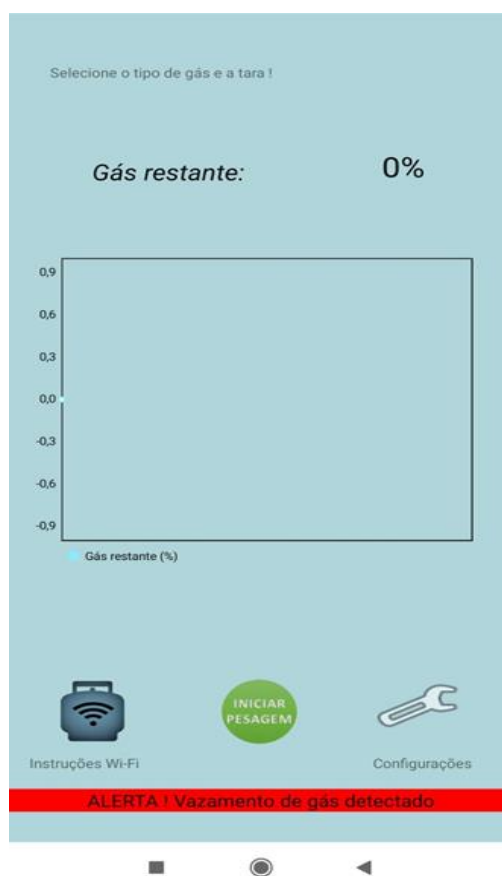


Fonte: Autoria própria.

Quando a porcentagem de gás restante for igual a 50%, 25% e igual ou menor que 10%, será enviado uma notificação ao usuário, alertando o volume de gás. As notificações abaixo de 10% serão enviadas toda vez que o volume diminuir 1% até que atinja 0%.

Uma notificação também será enviada caso haja um vazamento de gás, e na tela principal do aplicativo também haverá uma mensagem destacada para indicar o vazamento, conforme demonstrado na figura 45.

Figura 45. Mensagem de alerta de vazamento de gás detectado pelo sensor.



Fonte: Autoria própria.

3.2.3 DIRETRIZES DE QUALIDADE

Para garantir a qualidade do aplicativo, a Google define uma série de diretrizes que são recomendadas para oferecer ao usuário uma experiência consistente e intuitiva (DEVELOPER ANDROID, 2021).

Baseado nestas diretrizes foram analisados uma série de critérios. Estes critérios englobam principalmente a experiência visual, embora o desempenho e a estabilidade também tenham sido observados. Conforme apresentado na Tabela 08, na primeira coluna é feita uma descrição de critério, na segunda coluna o teste realizado, e na terceira coluna o resultado e comentários.

Tabela 08. Critérios de diretrizes de qualidade.

Descrição	Teste realizado	Resultado e comentários
Compatibilidade com a navegação padrão pelo botão “Voltar”.	Em cada tela e na caixa de diálogo foi testado o botão “Voltar”.	Funcionou normalmente, com o retorno esperado em cada situação.
Compatibilidade com a navegação por gestos.	Após ativada a navegação por gestos, foi testada em cada tela e na caixa de diálogos.	Assim, como o item anterior, funcionou normalmente. Recursos como a navegação por gestos é importante para aprimorar a experiência de interação com o usuário, ficando a critério do usuário qual tipo de navegação escolher.
Preservação e restauração do estado e dados.	Foi retomado o aplicativo pelo seletor de “Aplicativos recentes”, após “forçar a parada” e ao abrir pela tela inicial. Além da verificação entre telas.	Funcionou como esperado, sem perdas de dados. A adoção de banco de dados local e em nuvem foi imprescindível para a solução de perda de dados.
Definição dos canais de notificação, prioridade e tempo limite.	As notificações de alerta e volume de gás foram acionadas simultaneamente. E foram visualizadas com o smartphone bloqueado e desbloqueado, na gaveta de notificações.	Foi utilizado o canal de notificação padrão do Firebase. A prioridade e o tempo limite foram definidos no firmware do ESP32.
Compatibilidade com as orientações “paisagem” e “retrato”.	Rotação do smartphone, em todas as telas do aplicativo.	Apenas a opção “retrato” foi habilitada, pois a disposição dos componentes nesta orientação ficou mais bem ajustada.
Exibição dos gráficos, textos, imagens e outros elementos	Foram realizados todos os testes anteriores, ou seja, a navegação por todas as telas	Não houve nenhum dano perceptível dos elementos. A pouca quantidade de imagens,

sem distorção, desfoque ou pixelização perceptível.	e caixas de diálogos, inicialização do aplicativo por diferentes modos, rotação do celular, acionamento de notificações.	e estáticas, contribuíram para diminuir a possibilidade de erros. O elemento que necessitou maior atenção foi o gráfico, pois é um componente que pode se apresentar de modos diferentes dependendo do tempo e da porcentagem do gás. A implementação de um marcador nos pontos de intersecção foi primordial para facilitar uma visualização mais precisa.
Apresentação de falhas, ou outros erros como “O Android não está respondendo”.	Realização dos testes anteriores.	Durante a criação do aplicativo apresentou erros frequentes como “O Android não está respondendo” ou o encerramento súbito do aplicativo. Porém conforme surgiam, foi capaz solucioná-los, até que não foram encontrados mais.
Desempenho na inicialização e renderização.	Realização dos testes anteriores.	Desempenho satisfatório na inicialização (menor que 2 segundos). A renderização também não apresentou problemas, pois não é uma característica do aplicativo a variação de quadros, como animação.

Fonte: Autoria própria.

Assim como os critérios, os testes realizados também são sugeridos pela Google. Embora pareçam simples de verificar, estes testes são importantes para que não passe despercebido nenhum equívoco na criação do aplicativo.

Critérios referentes à segurança, privacidade e a integração com a Google Play não foram considerados, pois não foi o enfoque neste projeto. Porém, futuramente serão itens que terão prioridade para a continuidade do projeto.

4 RESULTADOS E DISCUSSÕES

4.1 CALIBRAÇÃO DA AWA

A biblioteca HX711master utilizada para a leitura dos dados enviados pelo HX711, possui uma variável chamada de fator_calibracao que auxilia na calibração dos dados verificados com o peso real do objeto pesado pelo sistema.

Após a fabricação da PCI, o valor de fator_calibracao foi alterado pois os primeiros testes foram realizados através de protoboard e muitos jumpers, o que pode interferir na leitura do HX711. Um pacote de 1 kg (um quilograma) de arroz foi utilizado como medida para essa calibração.

Com o auxílio de uma balança convencional, foram realizados testes de comparação das medidas que eram obtidas pela AWA e balança convencional.

Na tabela 09, foi realizada a aferição do peso de um pacote de 1 kg de arroz, que é onde temos que ter mais assertividade na medição pois será quando o botijão estiver no fim.

Tabela 09. Comparação de pesagens entre a AWA e balança convencional.

PESAGENS	1	2	3	4	5	6	7	8	9	10	Média
AWA	0,99	0,99	1,03	1,03	1,00	1,03	1,00	0,98	0,98	1,01	1,004
Convencional	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00

Fonte: Aatoria própria.

Na Tabela 10, foi realizada a aferição do peso de botijão de gás completo.

Tabela 10. Comparação de pesagens de um botijão de gás completa entre a AWA e balança convencional.

PESAGENS	1	2	3	4	5	6	7	8	9	10	Média
AWA	27,04	27,01	27,02	27,05	27,01	27,05	27,01	26,98	27,00	27,03	27,02
Convencional	27,20	26,90	27,00	27,00	27,10	27,10	27,00	27,00	26,90	27,00	27,02

Fonte: Aatoria própria.

4.2 TESTES DE INTEGRAÇÃO AWA X APLICATIVO

No dia 6 de fevereiro de 2021, foram realizados testes de integração do aplicativo e do hardware simulando o consumo de gás.

Foi utilizado um balde d'água com o valor de 13 kg, aferidos na balança convencional e a vazão da água foi realizada através de uma mangueira para outro balde d'água (Figura 46). O monitoramento foi realizado a cada minuto, durante aproximadamente uma hora, até esvaziar o balde.

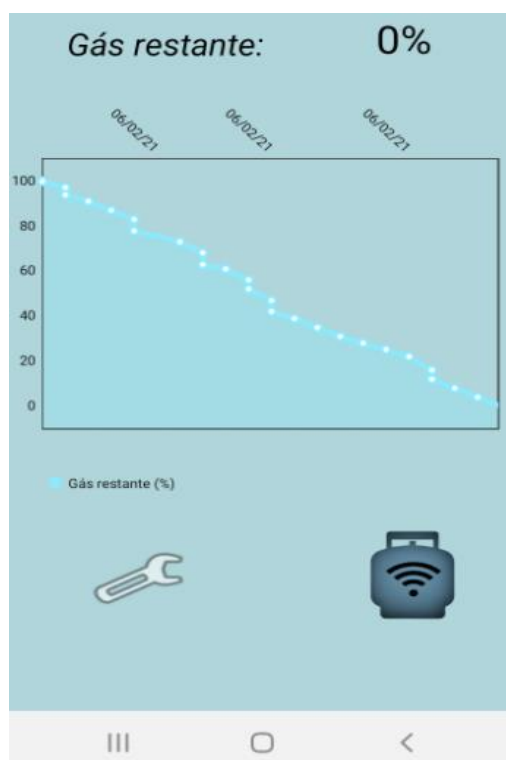
Figura 46. Teste da balança AWA com pesagens a partir de 13kg.



Fonte: Autoria própria.

O resultado do teste pôde ser visto através do aplicativo e na figura 47.

Figura 47. Representação gráfica do consumo de gás (Teste com 13kg inicial).



Fonte: Autoria própria.

A principal finalidade é originar dados simples e de fácil interpretação sobre o botijão de gás cadastrado nas telas anteriores, tais como a quantidade de gás restante (em porcentagem), além da data da última atualização do consumo e a duração estimada deste gás.

Quando a porcentagem de gás restante for igual a 50%, 25% e igual ou menor que 10% será enviado uma notificação ao usuário, alertando o volume de gás. As notificações abaixo de 10% serão enviadas toda vez que o volume diminuir 1% até que atinja 0%.

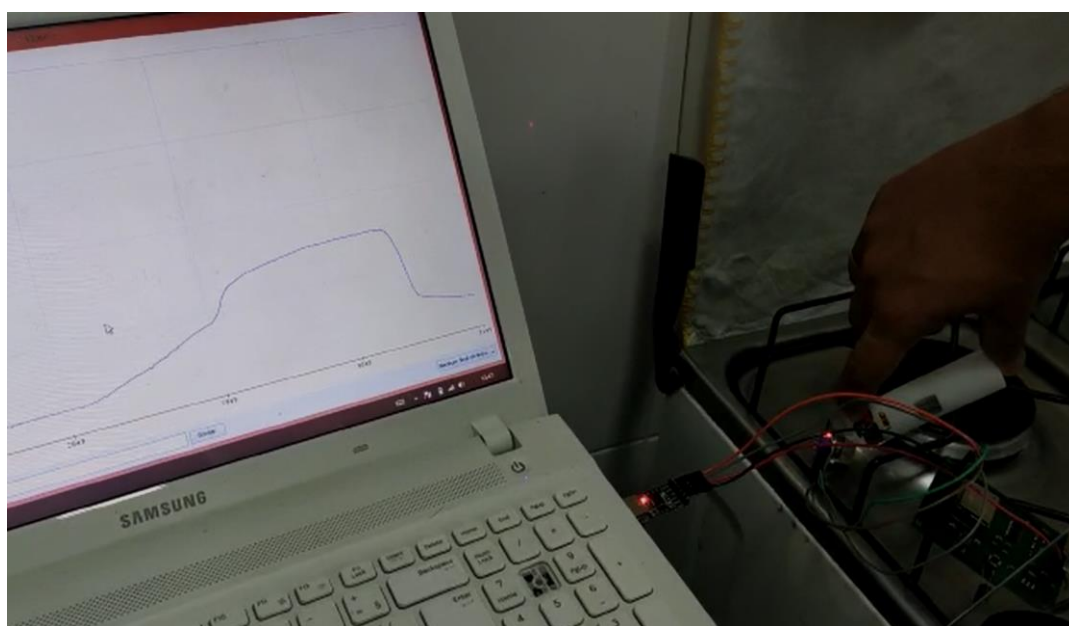
4.3 TESTE DO SENSOR DE GÁS

Para testar o sensor MQ-6, foram utilizadas duas formas de vazamento de gás: a utilização de um isqueiro, conforme representado na Figura 48 e a

abertura da vazão de gás na boca do fogão, conforme Figura 49, para que pudesse fazer uma simulação de forma segura envolvendo poucas quantidades de gás no local, sem oferecer nenhum risco.

O teste para verificar o funcionamento, realizado através da liberação de gás por um isqueiro acionado manualmente perto do sensor, foi possível verificar através do *Serial Plotter* disponibilizado no IDE do Arduino a subida da curva, como podemos ver na imagem a seguir.

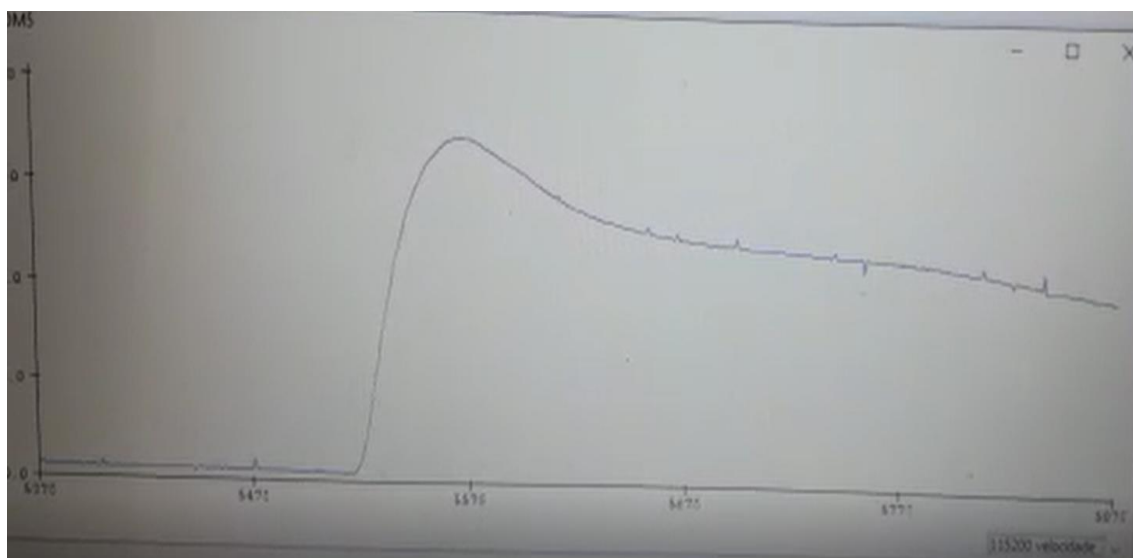
Figura 48. Representação gráfica de vazamento de gás com um isqueiro próximo ao sensor.



Fonte: Autoria própria.

Já no teste com a abertura da boca do fogão, é perceptível a diferença na curva de sensibilidade do sensor e também a demora para zerar o gráfico (Figura 49) indicando que o gás ainda permanece no ambiente depois de algum tempo.

Figura 49. Representação gráfica de vazamento de gás da boca do fogão próximo ao sensor.

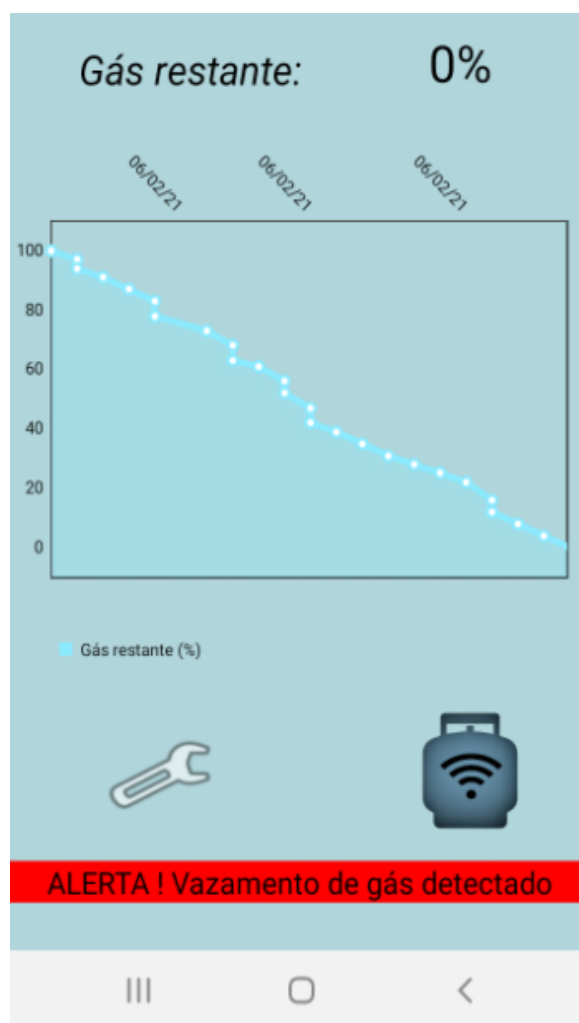


Fonte: Autoria própria.

No mesmo dia que realizamos o teste de simulação de consumo do gás, realizamos os testes do sensor de gás, o alerta é indicado através de uma notificação no smartphone e alerta no aplicativo conforme figura 50.

Dados como esse de ocorrência de vazamentos serão coletados em tempo real, onde o sensor identifica vazamentos, e enviará estes dados ao aplicativo pela internet. Assim, os usuários poderão realizar monitoramentos mesmo à distância, trazendo na segurança da sua vida e de seu lar ou estabelecimento.

Figura 50. Aviso de alerta de vazamento de gás no aplicativo.



Fonte: Aatoria própria.

A demonstração da utilização do aplicativo pode ser vista através do *link* <https://www.youtube.com/watch?v=pLKwv440JRg>. Neste vídeo, é realizada a conexão da AWA com a internet, configuração da tara e do tipo do botijão de gás, início das pesagens e o envio do alerta de vazamento de gás na tela do aplicativo.

5 CONCLUSÕES

A idealização deste projeto partiu como premissa monitorar o consumo de gás e garantir a segurança do usuário quanto ao vazamento de gás, de forma simples e eficiente. Para alcançar esse objetivo foram desenvolvidos o firmware, a placa de circuito de impresso, o aplicativo e a comunicação entre eles. Em seguida foi possível testar e aprovar o resultado do projeto.

O desenvolvimento do firmware foi realizado sem grandes problemas, devido a imensa documentação e exemplos para o microcontrolador ESP-32. Seu módulo Wi-fi integrado pode ser visto como sua principal vantagem, favorecendo muito para o desenvolvimento do projeto. Assim, foi fácil perceber por que é considerado um dos principais microcontroladores quando se refere a IoT. Além disso, a vasta utilização em práticas com microcontroladores durante o curso contribuiu muito para a ausência de maiores complicações.

Para o desenvolvimento da PCI, foram utilizadas além das informações presentes nos datasheets, os módulos do microcontrolador ESP-32 e o módulo do CI HX711 como modelo. Assim, reduziu-se qualquer erro tanto no esquemático quanto para rotear a placa. Ainda do mesmo modo foi cometido um pequeno equívoco no esquemático, porém foi possível fazer a reparação com um pequeno jumper.

O aplicativo desenvolvido conseguiu cumprir com suas funções. A partir dele o usuário pode interagir com a balança de maneira fácil e intuitiva. Alguns aspectos merecem ser destacados: o gráfico, que permite ao usuário um modo rápido e claro da visualização do consumo de gás; a possibilidade de o usuário conectar a balança à rede a partir do aplicativo; e a possibilidade de o usuário visualizar os dados do Firebase mesmo estando desconectado.

A comunicação entre aplicativo, Firebase e microcontrolador foi um ponto que recebeu muita atenção, pois o sucesso do projeto depende fundamentalmente de uma eficaz comunicação entre o ESP32 e o aplicativo. Na plataforma do Firebase foi possível visualizar os dados e simular a comunicação de modo simples e eficaz. Assim, o Firebase demonstrou ser uma ferramenta de

boa confiabilidade, satisfazendo a comunicação necessária para o projeto. Além disso, o Firebase ainda conta com uma gama de serviços que podem ser utilizados para incrementar o projeto futuramente.

A partir dos testes e simulações da balança, foi possível observar que ela possui uma excelente aferição de peso, devido fundamentalmente ao CI HX711, CI específico para trabalhar com células de carga; e ao firmware, que permitiu traduzir os dados vindos do sensor e calibrá-los. O sensor de vazamento de gás se comportou conforme esperávamos, sua sensibilidade atendeu aos propósitos do projeto e pôde ser observada de forma eficiente através de um gráfico no computador.

Portanto, de forma geral, podemos concluir que o trabalho conseguiu atingir seus objetivos. A balança comportou-se de modo eficaz em todos os aspectos.

Algumas melhorias podem ser consideradas para a continuidade e escalabilidade do projeto, como por exemplo:

- ✓ A autenticação do usuário para a maior segurança dos dados;
- ✓ Desenvolvimento da calibração da AWA através do aplicativo pelo cliente;
- ✓ Desenvolvimento de um aplicativo para smartphones com o sistema operacional IOS;
- ✓ Disponibilizar o aplicativo nas lojas online Android e IOS;
- ✓ Implementação de alertas sonoros no aplicativo de smartphone e no dispositivo AWA, tanto ao final do gás no recipiente quanto para indicar vazamentos;
- ✓ Desenvolvimento de módulos sensores de gás compatíveis com a AWA para a detecção de vazamentos de gás em outros cômodos da residência;
- ✓ Desenvolvimento da tela para registrar a validade da mangueira e do regulador.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ABNT - **Associação Brasileira de Normas Técnicas**. Disponível em: <https://www.abntcatalogo.com.br>. Acessado em 28 de outubro de 2020.

ALENCAR, D. S. **Implementação de um protótipo de automação residencial voltado para segurança**. Trabalho de Conclusão de Curso. Universidade Federal do Pará, Tucuruí/PA. 2019.

ALVES, R. G. F. **Dosagem programada de matéria-prima para os diversos processos da indústria**. Revista FATEC Garça. Faculdade de Tecnologia Garça, 2016.

ALVES, S. R. P. MENDONÇA, T. D. **Desenvolvimento de protótipo robótico móvel para monitoramento gasoso em ocorrências do tipo “vazamento de GLP”**. Monografia (Graduação) – Curso de Formação de Oficiais BM-MA, Universidade Estadual do Maranhão, 2018.

ANP – **Agência Nacional do Petróleo, Gás Natural e Biocombustíveis**. 2020. Disponível em: <http://www.anp.gov.br/petroleo-e-derivados2/glp>. Acessado em: 25 de maio de 2020.

ASMIRG - Associação Brasileira dos Revendedores de GLP. **Cartilha revendedor GLP**. Brasil, 2009. Disponível em:< <http://www.sindicombustiveis-al.com.br/wpcontent/uploads/2013/08/cartilha-posto-revendedor-combustiveis.pdf>> Acessado em: 08 de novembro de 2020.

AWA-SISTEMA DE MONITORAMENTO. Bibliotecas utilizadas. 2021. Disponível em: <https://github.com/alinescremin/awa-sistemademonitoramento/tree/main>. Acessado em: 10 de abril de 2021.

BATSCHINSKI, G. **Backend as a Service: Prós e Contras**. 2016. Disponível em: <https://www.infoq.com/br/news/2016/07/backend-pros-e-contras/>. Acesso em: 23 de novembro de 2020.

BIBLIOTECA HX711. Disponível em: <https://github.com/bogde/HX711>. Acessado em: 08 de novembro de 2020.

BIBLIOTECA HX711 MASTER. Disponível em: <https://github.com/sparkfun/HX711-Load-Cell-Amplifier>. Acessado em: 10 de novembro de 2020.

BIBLIOTECA FIREBASE ESP32 CLIENT. Disponível em: <https://github.com/mobizt/Firebase-ESP32>. Acessado em 08 de novembro de 2020.

BIBLIOTECA MATERIAL-INTRO. Disponível em: <https://github.com/heinrichreimer/material-intro>. Acessado em 15 de novembro de 2020.

BIBLIOTECA MPANDROID CHART. Disponível em: <https://github.com/PhilJay/MPAndroidChart#documentation>. Acessado em 08 de novembro de 2020.

BIBLIOTECA WI-FI MANAGER-ESP32. Disponível em: <https://github.com/zhouhan0126/WI-FIMANAGER-ESP32>. Acessado em: 10 de novembro de 2020.

BRITO, L. L. F., MIRANDA NETO, M. DE OLIVEIRA, M. R. MORAES, I. A., MUNIZ, V. A. O. **Protocolos de comunicação para Internet of Things (IoT)**. Intercursos Revista Científica. Ciências Exatas - UEMG. Ituiutaba, MG, 2018.

CANDIDO, L. UTILI, M. ZUCCHETTI, M. CIAMPICHETTI, A. CALDERONI,, P. **Development of advanced hydrogen permeation sensors to measure Q2 concentration in lead-lithium eutectic alloy**. Fusion Engineering and Design. V. 124. 2017.

CASTELLANI, A. P., BUI, N., CASARI, P., ROSSI, M., SHELBY, Z., ZORZI, M. **Architecture and protocols for the Internet of Things: a case study**. 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). Mannheim, Germany, 2010.

COLOMBO, J. F.; LUCCA FILHO, J. **INTERNET DAS COISAS (IoT) E INDÚSTRIA 4.0: revolucionando o mundo dos negócios**. Revista Interface Tecnológica. Faculdade de Tecnologia (FATEC), Taquaritinga, SP, Brasil, 2018.

CONSIGÁS - GLP – **GÁS LIQUEFEITO DE PETRÓLEO: Fonte de energia eficiente, econômica e ecologicamente correta**. 2017. Disponível em <https://www.consigaz.com.br/gas-glp/>. Acessado em 06 de novembro de 2020.

CORPO DE BOMBEIROS DO PARANÁ. **Cuidados com o gás de cozinha ou GLP**. 2021. Disponível em: <http://www.bombeiros.pr.gov.br/Pagina/Gas-de-cozinha-ou-GLP>. Acessado em 07 de abril de 2021.

CORREIA, A. H. M.; DE LIMA, P. Y. C. C.; SANTOS, T. V. M. **Sistema de proteção contra vazamento de gás inflamável e acúmulo de fumaça em ambientes fechados**. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. BRASIL, 2017.

Curso de Redes de Computadores. **Apostila de Internet e Arquitetura TCP/IP**. volume I, 2ª ed. PUC-RIO/CCE, 2020. Disponível em: <http://www.jairo.pro.br/redesII/tcp_ip.pdf > Acessado em 11 de maio de 2021.

CURTO, Blog do. **Conhecendo o ESP32: Destrinchando a estrutura do ESP32 e suas funções importantes**. Curto Circuito. 2018. Disponível em: <https://www.curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>. Acessado em 04 de abril de 2021.

CUTRI, R. **Técnicas de projeto de compatibilidade eletromagnética (CEM) no desenvolvimento de placas de circuito impresso**. 2005.

DAL PONTE, G. B. **Desenvolvimento de um sistema de monitoramento remoto de consumo de gás envasado utilizando tecnologia IoT**. Trabalho de Conclusão de Curso. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Florianópolis. 2018.

DEVELOPER ANDROID, **Principais critérios de qualidade do app**. 2021. Disponível em: <https://developer.android.com/docs/quality-guidelines/core-app-quality#CR-9>. Acessado em 09 de abril de 2021.

DING, Y. HONG, S. H., KIN, R. LU, J., LEE, Y. H., XIAOBING, A. XU, L. **Experimental Investigation of the Packet Loss Rate of Wireless Industrial Networks in Real Industrial Environments**. IEEE International Conference on Information and Automation, Lijiang, 2015.

ESP32 WROOM 32 Peripheral Schematics. 2021. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. Acessado em 22 de novembro de 2020.

FDE- **Fundação para o Desenvolvimento da Educação. Manual de uso e segurança de instalações de gás em escolas**. 2. ed. São Paulo, 2009. Disponível em: <http://file.fde.sp.gov.br/portalfde/Arquivo/DocRedeEnsino/ManualGas.pdf> Acessado em: 07 de novembro de 2020.

FERNANDES, P. M. C. **Desenvolvimento de uma Framework Real Time Web para HTML5**. Universidade Atlântica, Barcarena, 2013.

FIREBASE. **Produtos**. 2018. Disponível em: <https://firebase.google.com/products-build?hl=pt-br#grow-features>. Acesso em: 04 de abril de 2021.

FIREBASE - **Firestore Realtime Database**. 2020a. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>. Acessado em 23 de novembro de 2020.

FIREBASE - **Firestore Cloud Messaging**. 2020b. Disponível em: <https://firebase.google.com/docs/cloud-messaging>. Acessado em 23 de novembro de 2020.

FIREBASE - **Privacidade e segurança no Firebase**. 2020c. Disponível em: <https://firebase.google.com/support/privacy?hl=pt-br> Acessado em 24 de novembro de 2020.

FIREBASE - **Visão geral da arquitetura do FCM.** 2021. Disponível em: <https://firebase.google.com/docs/cloud-messaging/fcm-architecture>. Acessado em 12 de abril de 2021.

FISPQ - **FICHA DE INFORMAÇÕES DE SEGURANÇA DE PRODUTO QUÍMICO – GLP.** 2015. Disponível em: <https://www.liquigas.com.br/wps/wcm/connect/a94a95004640d0be8ce7cddd2947447b/FISPQ+GLP+rev+01-11.pdf?MOD=AJPERES>. Acessado em 28 de janeiro de 2021.

FREECODECAMP. **Prototipagem da Internet das coisas com Firebase: como fazer mais com menos.** 2019. Disponível em: <https://www.freecodecamp.org/news/iot-prototyping-with-firebase-doing-more-with-less-2f5c746dac8b/>. Acessado em 21 de novembro de 2020.

FRENZEL, L. **What's the difference between IEEE 802.15.4 and Zigbee wireless?** Electronic Design, 2013.

FUNDACENTRO. **Inspeções de equipamentos: Estudo de casos. Caso 058: O desastre de San Juanico.** 2014. Disponível em: <http://inspecaoequipto.blogspot.com.br/2014/02/caso-058-o-desastre-de-sanjuanico-1984.html>>Acessado em: 04 de abril de 2021.

GOMES, C., OLLER, C., PARADELLS, J. **Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology,** Sensors, Basileia, 2001.

GROSSMANN, G. H. **IoT Smart Lock (ISL): Sistema de fechadura inteligente, utilizando protocolos de internet das coisas.** Monografia de especialização. Universidade Tecnológica Federal do Paraná. 2018.

INTERNACIONAL BUSINESS MACHINES CORPORATION; EUROTECH. **MQTT specification version 3.1.** out. 2010. Disponível em: public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>. Acessado em: 04 de setembro de 2020.

JAFFEY, T. **MQTT and CoAP, IoT protocols**. 2014. Disponível em: <http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php>. Acessado em: 04 de setembro de 2020.

JAVA. **Obtenha informações sobre a tecnologia Java**. 2021. Disponível em: <https://www.java.com/pt-BR/about/> Acesado em: 03 de abril de 2021.

K TECNOLOGIA, F. Blog do Fernando K. **Introdução ao ESP32**. 2018. Disponível em: <https://www.fernandok.com/2017/11/introducao-ao-esp32.html> Acessado em: 22 de novembro de 2020.

KARVINEN, K.; KARVINEN, T. **Primeiros Passos com Sensores: Perceba o mundo usando eletrônica, Arduino e Raspberry Pi**. Novatec Editora, 2014.

LEAL, G. F. **WaterManna: a IoT na gestão de recursos hídricos**. Dissertação (Mestrado) - Universidade Estadual de Maringá, Centro de Tecnologia, Departamento de Informática, Programa de Pós-Graduação em Ciência da Computação, 2019.

LIMA, C. B.. **Técnicas de Projetos Eletrônicos com os Microcontroladores AVR**. Edição do autor - Clube de Autores, 1ª ed. 2010. Disponível em https://professorpetry.com.br/Bases_Dados/Apostilas_Tutoriais/Projeto_PCI_Charles.pdf Acessado em 22 de novembro de 2020.

LIQUIGÁS. **O que é GLP?** 2020. Disponível em: <https://www.liquigas.com.br/wps/portal/>. Acessado em 28 de outubro de 2020.

LIQUIGÁS. **22º Prêmio Fiesp de Mérito Ambiental**. 2020. Disponível em: www.fiesp.com.br/arquivo-download/?id=214426. Acessado em: 02 de abril de 2021.

LUZ, T. S. SILVA, V. E. O. **DOE VIDA: Aplicativo para Captação de Doadores de Sangue**. Trabalho de Conclusão de Curso. Coordenação dos cursos de Informática – Tecnologia em Análise e Desenvolvimento de Sistemas. Jataí, 2017.

MARKUS, O. **Circuitos Elétricos - Corrente Contínua e Corrente Alternada**. 8 Ed. São Paulo Érica, 2009.

MARTO, J. **Capacitores de desacoplamento em projetos de alta frequência**. 2016. Disponível em: <https://www.embarcados.com.br/capacitores-de-desacoplamento/>. Acessado 22 de novembro de 2020.

MEDEIROS, G. V. S. SANTOS, M. S. **Smartgás: uma plataforma inteligente para monitoramento de gás de cozinha**. Trabalho de Conclusão de Curso (Técnico em Informática para Internet) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal, 2017.

MOREIRA, A. M. **Segurança na utilização de gás liquefeito de Petróleo**. Trabalho de Conclusão de Curso. Universidade Federal do Espírito Santo. Vitória/ ES. 2015.

MOTTA, D. **Activity: o que é, como é iniciada e ciclo de vida básico**. 2019. Disponível em: <https://douglasmotta.com.br/2019/06/22/activity-o-que-e-como-e-iniciada-e-ciclo-de-vida-basico/>. Acessado em: 26 de novembro de 2020.

MORIBE, F. **Firestore – Vantagens de um BaaS para sua Startup**. 2016. Disponível em: <https://medium.com/@fgmoribe/firebase-vantagens-de-um-baas-para-sua-startup-38fd3891329a>. Acesso em: 23 de novembro de 2020.

NETO, H. V. **Microcontroladores MCS51**. Universidade Tecnológica Federal do Paraná. Curitiba, 2002.

NPT 028 – Norma de Procedimento Técnico: **Manipulação, armazenamento, comercialização e utilização de gás liquefeito de petróleo (GLP)**. Corpo de Bombeiros BM/7. 2014. Disponível em: http://www.bombeiros.pr.gov.br/sites/bombeiros/arquivos_restritos/files/documento/2018-12/NPT_028.pdf. Acessado em: 02 de março de 2021.

OASIS - **MQTT Version 5.0 OASIS**. 2019 Standard. Disponível em: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>. Acessado em: 23 de novembro de 2020.

OLIVEIRA, S. D. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**, Novatec, 1ª Edição, vol. 1, São Paulo, 2017.

OMEGA Engineering Inc. **What is a Load Cell & How Does it Work?** 2020. Disponível em <https://www.omega.com/en-us/resources/load-cells>. Acessado em 08 de novembro de 2020.

PATACA, C. **A INTERNET DAS COISAS: Tipologias, Protocolos e Aplicações**. Atlantic International University (AIU), Honolulu, Hawaii, 2020.

PINTO, A. C. V. **DESENVOLVIMENTO DE SISTEMA DE MONITORAMENTO DE GÁS LP COM ALARME POR SMS**. Monografia de Graduação em Engenharia de Controle e Automação. Universidade Federal de Ouro Preto. Minas Gerais, 2016.

QUESSADA, M. S. CRISTIANI, A. L. RANZANI JR, P. L. PEREIRA, R. S. LEAL, M. P. MENEGUETTI, R. I. **Desenvolvimento de um Modelo de Mobilidade Urbana em Tempo Real para Simuladores de Rede**. 2019: ANAIS ESTENDIDOS DO XXXVII SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS. Instituto Federal de São Paulo. 2019.

RIPPER NETO, Arthur Palmeira. **Vibrações Mecânicas**. Rio de Janeiro: e-papers, 2007.

SCRIMGER, R. *et al.* **TCP/IP: a Bíblia**. Rio de Janeiro: Elseiver, 2002.

SETHI, P. & SARANGI, S. R. **Internet of Things: Architectures, Protocols, and Applications**. Journal of Electrical and Computer Engineering. Volume 2017, Article ID 9324035, 25 pages. 2017.

SILVA, L. A. **Apostila de Android – Programando passo a passo**. Programação básica (versão android studio). 2015. Disponível em <https://www.leonardoleandrodev.com.br/downloads/apostila-android.pdf>. Acessado em: 26 de novembro de 2020.

SILVA, L. J. **Internet das coisas**. Trabalho de Conclusão de Curso. Universidade do Sul de Santa Catarina – UNISUL. Palhoça/SC, 2017.

SILVESTRE, B. O. **Serviços de Notificação de Eventos Baseados em Publish/Subscribe**. 2005. Disponível em: <http://www-di.inf.puc-rio.br/~endler/courses/semGSD/monografias/bruno-silvestre.pdf> Acessado em 28 de outubro de 2020.

SINDIGÁS- Sindicato Nacional das Empresas Distribuidoras de Gás Liquefeito de Petróleo. **Gás LP no Brasil: Manual de segurança para o posto revendedor de GLP**. Normas para o Armazenamento de Recipientes Transportáveis de Gás Liquefeito de Petróleo. 2015. Disponível em: http://www.sindigas.com.br/Download/Arquivo/Manual_Revendedor_SINDI_GAS_Ago2011_09set2011_SITE_634515322748649721.pdf Acessado em: 07 de novembro de 2020.

SINGH, S., SINGH, N. **Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce**. International Conference on Green Computing and Internet of Things (ICGCIoT). pp. 1577–1581, 2015.

SOBRINHO, V, F. **Segurança com gás liquefeito de petróleo-GLP**. Fundacentro. São Paulo, 2014. Disponível em: <http://www.fundacentro.gov.br/Arquivos/sis/EventoPortal/AnexoPalestraEvento/Seguran%C3%A7a%20GLP%20.pdf>. Acessado em 08 de novembro de 2020.

SONI, D.; MAKWANA, A. **A survey on MQTT: a protocol of internet of things (IoT)**. In: International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017). 2017.

STALLINGS, W. **Data and computer communications**. 8a. ed. Upper Saddle River, NJ: Prentice Hall, 2006.

Tabela de Conversão - AWG. **Tabela de fios AWG com tabela de conversão para medidas métricas**. Disponível em:

http://www.if.ufrgs.br/~mittmann/tabela_de_fios.pdf. Acessado em 08 de novembro de 2020.

TOKARNIA, M. **Um em cada 4 brasileiros não tem acesso à internet, mostra pesquisa:** Número representa 46 milhões que não acessam a rede. 2020. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2020-04/um-em-cada-quatro-brasileiros-nao-tem-acesso-internet>. Acessado em: 29 de novembro de 2020.

VARGAS E PINTO, A. C. **Desenvolvimento de sistema de monitoramento de gás LP com alarme por SMS.** Monografia (Graduação) – Universidade Federal de Ouro Preto. Escola de Minas. Colegiado do Curso de Engenharia de Controle e Automação e Técnicas Fundamentais. 2016.

VIDAL, V. **Balança digital com Arduino e célula Strain Gauge.** 2017. Disponível: <https://blog.eletrogate.com/balanca-digital-com-arduino-aprenda-a-usar-a-celula-de-carga/> Acessado em: 21 de novembro de 2020.

XAVIER, R. C. **Entenda a importância do protocolo Lora na agricultura 4.0.** 2018. Disponível em <https://www.khomp.com/pt/protocolo-lora-na-agricultura-40/>. Acessado em: 04 de Abril de 2021.