

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA
PARAÍBA

COORDENAÇÃO DO CURSO DE ENGENHARIA ELÉTRICA



HENRIQUE DE PAIVA ARAÚJO PONTES

Protótipo de Medição de Energia para Aplicações IOT

João Pessoa

2023

HENRIQUE DE PAIVA ARAÚJO PONTES

Protótipo de Medição de Energia para Aplicações IOT

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como parte dos requisitos para a obtenção do grau de Engenheiro Eletricista.

Orientador: Gilvan Vieira de Andrade Junior

João Pessoa
2023

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *Campus* João Pessoa

P814p Pontes, Henrique de Paiva Araújo
Protótipo de Medição de Energia para Aplicações IOT/
Henrique de Paiva Araújo Pontes. – 2023.
58 f.

TCC (Graduação – Bacharelado em Engenharia Elétrica) – Instituto Federal da Paraíba – IFPB / Coordenação de Engenharia Elétrica, 2023.

Orientador: Prof^o Gilvan Vieira de Andrade Junior.

1. Medidor de energia. 2. Microcontrolador. 3. ESP-32. 4. Softwares de Desenvolvimento. 5. Corrente elétrica. I. Título.

CDU 621.317:004

Bibliotecária responsável Ivanise Andrade Melo de Almeida – CRB15/96

HENRIQUE DE PAIVA ARAÚJO PONTES

Protótipo de Medição de Energia para Aplicações IOT

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como parte dos requisitos para a obtenção do grau de Engenheiro Eletricista.

Aprovado Pela Banca Examinadora em:

BANCA EXAMINADORA

Documento assinado digitalmente

 gov.br

GILVAN VIEIRA DE ANDRADE JUNIOR

Data: 08/01/2024 12:07:16-0300

Verifique em <https://validar.iti.gov.br>

Gilvan Vieira de Andrade Júnior

Orientador

Documento assinado digitalmente

 gov.br

GILVAN VIEIRA DE ANDRADE JUNIOR

Data: 08/01/2024 12:08:29-0300

Verifique em <https://validar.iti.gov.br>

Lincoln Machado De Araújo

Membro da Banca

Walmeran Jose Trindade Junior

Membro da Banca

Documento assinado digitalmente

 gov.br

WALMERAN JOSE TRINDADE JUNIOR

Data: 22/12/2023 18:13:10-0300

Verifique em <https://validar.iti.gov.br>

João Pessoa

2023

Agradecimentos

Reservo este espaço para agradecer à Deus e as pessoas que, direta ou indiretamente, contribuíram para a realização deste relatório. Entre estes, faço uma menção especial a meus colegas e familiares, que sempre me incentivaram e dedicaram momentos de atenção, solidariedade e companheirismo. Agradeço aos meus colegas de curso por todos os momentos que compartilhamos durante esses mais de quatro anos, em especial aos meus colegas Marcello Aires, Lucas Alves, Pedro Kelven, Almir Santos, Mateus Lucas e Bryan Santos que deram apoio e dicas valiosas no presente relatório. E também aos professores do IFPB que fazem com que a Instituição se torne uma segunda casa e sempre estiveram empenhados em formar profissionais éticos e bem preparados para a atuação no mercado de trabalho.

“You have to choose the move that feels right sometimes; that’s what intuition is.”
(Magnus Carlsen)

RESUMO

Este trabalho de conclusão de curso (TCC) tem como objetivo explorar a tecnologia de medidores de energia com o uso de IoT, descrevendo as vantagens e desafios envolvidos na sua implementação. O trabalho será dividido em três etapas: primeiro, será realizada uma revisão bibliográfica sobre o assunto, discorrendo sobre as tecnologias envolvidas. Em seguida, será desenvolvido um protótipo de medidor de energia com IoT, para avaliar a eficácia da tecnologia. Finalmente, serão apresentadas as conclusões e recomendações para melhorias futuras do trabalho.

Espera-se que este TCC contribua para a compreensão dos benefícios do uso de medidores de energia com IoT. Além disso, a implementação de um protótipo permitirá avaliar a viabilidade técnica e econômica dessa tecnologia em diferentes contextos além de educar o consumidor final quanto aos seus gastos em períodos distintos.

Palavras-chave: Medidor de Energia; Microcontrolador; ESP-32; IoT; Potência; kWh; Corrente Elétrica; Consumo elétrico.

ABSTRACT

This present academic work aims to explore the technology of energy meters using IoT, describing the advantages and challenges involved in its implementation. The work will be divided into three stages: first, a literature review will be conducted on the subject, discussing the technologies involved. Next, an IoT energy meter prototype will be developed to evaluate the effectiveness of the technology. Finally, the conclusions and recommendations for future improvements will be presented.

It is expected that this thesis will contribute to the understanding of the benefits of using IoT energy meters. Additionally, the implementation of a prototype will allow for the assessment of the technical and economic feasibility of this technology in different contexts, as well as educate the end consumer about their expenses in different periods.

Keywords: *Energy Meter; Microcontroller; ESP-32; IoT; Power; kWh; Electric Current; Electrical Consumption.*

LISTA DE FIGURAS

Figura 1 – Um ciclo de corrente alternada	15
Figura 2 – Tensão de Pico Eficaz e RMS	16
Figura 3 – Regra da mão direita	17
Figura 4 – ESP32 Pinout	19
Figura 5 – Visual Studio Code	24
Figura 6 – Imagem das bibliotecas instaladas utilizando o PlataformIO e o Visual Studio Code	25
Figura 7 – Plot saída serial do microcontrolador	28
Figura 8 – Software TablePlus	29
Figura 9 – Página <i>web</i> do GraphPad	29
Figura 10 – Imagem de todos os componentes utilizados para o protótipo	31
Figura 11 – Curva de Linearidade ADC	32
Figura 12 – Imagem do esquemático do projeto	33
Figura 13 – Protótipo montado	33
Figura 14 – Calibração do protótipo	35
Figura 15 – Captura de tela da saída serial do microcontrolador plotada	40
Figura 16 – Captura de tela dos registros gravados no banco de dados	41
Figura 17 – Captura de tela da página <i>web</i>	42
Figura 18 – Captura de tela da página <i>web</i> para dispositivos móveis	43
Figura 19 – Gráfico em tempo real da leitura da potência elétrica	44

LISTA DE TABELAS

Tabela 1 – Tabela com os materiais utilizados para o desenvolvimento do protótipo 30

LISTA DE ABREVIATURAS E SIGLAS

AC	<i>Alternating current</i>
ADC	<i>Analog-to-Digital Converter</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
A	<i>Amperes</i>
GND	<i>Ground</i>
GPIO	<i>General Purpose Input/Output</i>
HTML	<i>HyperText Markup Language</i>
IOT	<i>Internet of Things</i>
IDE	<i>Integrated Development Environment</i>
kWh	<i>Quilowatt-hora - Kilowatt-hour</i>
LED	<i>Light-Emitting Diode</i>
NDU	<i>Norma de Distribuição Unificada</i>
OLED	<i>Organic Light-Emitting Diode</i>
PHP	<i>Hypertext PreProcessor</i>
RMS	<i>Raiz do valor quadrático médio - Root Mean Square</i>
SDA	<i>Serial Data</i>
SCL	<i>Serial Clock</i>
SHA	<i>Secure Hash Algorithm</i>
V	<i>Volts</i>
VCC	<i>Common Collector Voltage</i>
W	<i>Watts</i>
Wi-fi	<i>Wireless Fidelity</i>

SUMÁRIO

	LISTA DE ABREVIATURAS E SIGLAS	8
1	INTRODUÇÃO	11
1.1	Objetivos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Conceitos Elétricos	13
2.1.1	Corrente Alternada	14
2.1.2	RMS	15
2.2	Eletromagnetismo	16
2.3	IoT	17
2.4	Microcontrolador	18
2.4.1	ESP-32	18
2.5	Linguagens Desenvolvimento do software protótipo	20
2.5.1	C++	20
2.5.2	PHP	20
2.5.3	Javascript	21
2.5.3.1	Chart.JS	22
2.5.4	HTML	22
2.5.5	MySQL	22
2.6	Softwares de Desenvolvimento	23
2.6.1	Visual Studio Code	23
2.6.2	PlataformIO	24
2.6.2.1	Bibliotecas PlataformIO	25
2.6.2.1.1	Arduino.h	25
2.6.2.1.2	Wifi.h	26
2.6.2.1.3	ZMPT101B.h	26
2.6.2.1.4	AdafruitGFX.h	27
2.6.2.1.5	HttpClient.h	27
2.6.3	BetterSerialPlotter	27
2.6.4	TablePlus	28
2.6.5	GraphPad	29
3	MATERIAIS E MÉTODOS	30
3.1	Materiais utilizados	30
3.2	Metodologia	31

3.2.1	Funcionamento dos componentes	31
3.2.2	Montagem do protótipo	32
3.2.3	Testes iniciais no microcontrolador	34
3.2.4	Conversão e calibração dos dados	34
3.2.5	Envio e armazenamento dos dados	35
3.2.6	Aplicação no microcontrolador	35
3.2.7	Aplicação no servidor	36
3.2.8	Consulta dos dados	38
3.2.9	Exibição dos dados na página web	38
4	RESULTADOS	40
4.1	Resultados obtidos diretamente do microcontrolador	40
4.2	Resultados demonstrados graficamente	41
5	CONCLUSÃO	45
	REFERÊNCIAS	47
	APÊNDICES	48
	Apêndice A - Código fonte na íntegra do código implementado no microcontrolador.	48
	Apêndice B - Código fonte do código php onde o servidor recebe os dados do microcontrolador e faz a inserção no banco de dados.	52
	Apêndice C - Código fonte das consultas MySQL dos dados armazenados no banco de dados.	54
	Apêndice D - Github do projeto com os arquivos desenvolvidos	57

1 INTRODUÇÃO

Estamos vivendo em uma época em que a tecnologia está se transformando rapidamente e cada vez mais dispositivos estão conectados à internet. A relevância deste tema é evidente no contexto da crescente preocupação com a eficiência energética e a sustentabilidade ambiental. A adoção de medidores de energia com IoT pode ajudar a reduzir o consumo de energia em residências e empresas, além de promover a transição para uma matriz energética mais limpa e sustentável.

Os medidores de energia elétrica são equipamentos utilizados pelas concessionárias de energia elétrica para registrar o consumo de seus clientes, e são fundamentais para a emissão de faturas de energia elétrica, estes medidores podem ser analógicos ou digitais, e funcionam através da medição da corrente elétrica que passa através deles, são essenciais para o funcionamento do setor de energia elétrica em todo o mundo.

Já a Internet das Coisas (IoT), é tecnologia em constante evolução e que tem sido aplicada em diversos setores, incluindo o de energia elétrica. Unificando os medidores de energia elétrica com IoT pode proporcionar vantagens significativas em termos de monitoramento e controle no consumo de energia.

Com a união da Internet das Coisas com os medidores se faz possível fazer um monitoramento da energia elétrica utilizada e com isso gerar gráficos e relatórios afim de enfatizar a eficiência energética consumida por uma residência, realizando a troca de equipamentos de alto consumo ou visualizando possíveis horários que demonstram desperdícios. Essa eficiência é um papel crucial no cenário contemporâneo, refletindo a busca por práticas sustentáveis e responsáveis no uso dos recursos naturais. Neste projeto como dito, tem objetivo minimização do desperdício de energia em diversas atividades.

Além do impacto ambiental positivo, a eficiência energética também representa uma abordagem economicamente vantajosa para a conta de energia do usuário. A implementação de práticas e tecnologias mais eficientes resulta em reduções nos custos operacionais para empresas e consumidores, promovendo uma economia financeira substancial a longo prazo.

Portanto, este TCC tem como objetivo demonstrar tudo isso com a unificação destas diretrizes e auxiliar na compreensão dos benefícios e desafios da implementação destas tecnologias juntas.

1.1 Objetivos

Este trabalho tem como objetivo geral desenvolver um protótipo de medidor de energia elétrica acoplado ao medidor residencial já existente e por meio de uma página *web* visualizar e analisar o consumo elétrico desta residência.

Já os objetivos específicos são:

- A:** Compreender os fundamentos de energia elétrica e conceitos de IOT;
- B:** Assimilar os conhecimentos em linguagens de programação (C, PHP, JavaScript, MySQL) juntamente com os conhecimentos de eletrônica;
- C:** Projetar o circuito e adquirir o que se faz necessário para sua montagem;
- D:** Aplicar os conhecimentos elétricos e eletrônicos para montagem e o bom funcionamento do protótipo;
- E:** Documentar os resultados obtidos por meio da elaboração e defesa deste TCC.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados as bases do conhecimento teórico utilizado no desenvolvimento do trabalho. Nesta seção será apresentado um pouco sobre os conceitos elétricos utilizados no projeto, conceitos de IoT, assim como das linguagens de programação e *softwares* utilizados ao longo do projeto.

2.1 Conceitos Elétricos

Tensão elétrica, corrente elétrica e potência elétrica são conceitos fundamentais da eletricidade e são utilizados em diversos contextos, desde a produção de energia elétrica até a utilização em dispositivos eletrônicos.

A tensão elétrica é a diferença de potencial elétrico entre dois pontos, medida em volts (V). Ela é responsável por impulsionar a corrente elétrica em um circuito elétrico, sendo comparada com a pressão da água em um encanamento, que impulsiona o fluxo de água.

A corrente elétrica é o fluxo de elétrons em um circuito elétrico, medido em ampères (A). Ela é produzida pelo movimento dos elétrons através dos condutores elétricos, impulsionados pela tensão elétrica presente no circuito. A corrente elétrica pode ser contínua ou alternada, dependendo do tipo de circuito elétrico.

A potência elétrica é a quantidade de energia elétrica consumida por um dispositivo elétrico por unidade de tempo, medida em watts (W). A potência elétrica pode ser utilizada para dimensionar a capacidade de um circuito elétrico e para comparar o consumo de energia de diferentes dispositivos.

Os medidores geralmente medem a potência operando por meio do princípio da indução eletromagnética. Em outras palavras, o campo magnético produzido induz o movimento rotativo do disco no medidor apenas quando há consumo de energia elétrica. Isso resulta no deslocamento das engrenagens e dos ponteiros do medidor. No protótipo em questão a potência é medida através de um sensor que também utiliza da indução eletromagnética como explicado no tópico de eletromagnetismo.

É importante destacar que a tensão elétrica, corrente elétrica e potência elétrica estão interligadas e afetam o desempenho e eficiência dos sistemas elétricos. Com isso, existem diversas fórmulas de cálculo de potência elétrica, que são utilizadas para determinar a quantidade de energia elétrica consumida por um dispositivo ou circuito elétrico, conceitos fundamentais para ter um conhecimento básico e entender o funcionamento dos dispositivos e sistemas elétricos. As principais fórmulas do cálculo de potência elétrica

são:

$$P = V \cdot I \quad (2.1)$$

Essa é a fórmula mais comum e a que será utilizada para calcular a potência elétrica. Ela utiliza a tensão elétrica (V) e a corrente elétrica (I) para determinar a potência elétrica (P) consumida por um dispositivo ou circuito elétrico.

$$P = I^2 \cdot R \quad (2.2)$$

Essa fórmula é utilizada para calcular a potência elétrica em circuitos elétricos que possuem uma resistência elétrica (R). Ela utiliza a corrente elétrica (I) e a resistência elétrica (R) para determinar a potência elétrica (P) consumida pelo circuito.

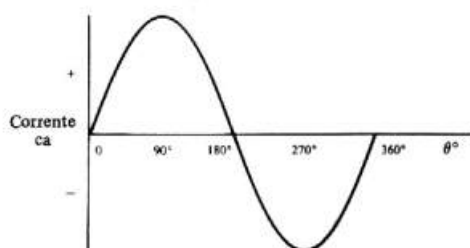
$$P = \frac{V^2}{R} \quad (2.3)$$

Essa fórmula é uma variação da fórmula anterior e é utilizada para calcular a potência elétrica em circuitos elétricos que possuem uma resistência elétrica (R). Ela utiliza a tensão elétrica (V) e a resistência elétrica (R) para determinar a potência elétrica (P) consumida pelo circuito.

Além dessas equações, existem outras fórmulas que são empregadas em contextos específicos, como por exemplo, para calcular a potência elétrica de motores elétricos. É fundamental ter em mente que o cálculo da potência elétrica desempenha um papel essencial no dimensionamento da capacidade de um circuito elétrico. Isso garante o correto funcionamento dos dispositivos elétricos e previne sobrecargas e danos aos equipamentos.

2.1.1 Corrente Alternada

A corrente alternada é um tipo de eletricidade que é usada em quase todas as residências do mundo. Ela é diferente da corrente contínua, pois o fluxo de elétrons muda constantemente de direção. A corrente alternada é produzida nas usinas elétricas e, em seguida, é transportada para as casas por meio de linhas de transmissão de alta tensão. Nas residências, a corrente alternada é transformada em uma tensão mais baixa por meio de um transformador e é fornecida aos pontos de consumo. Na figura abaixo esta uma demonstração de um ciclo da corrente alternada.

Figura 1 – Um ciclo de corrente alternada

Fonte: Gussow (2005)

É importante destacar que a frequência da corrente alternada residencial varia em diferentes regiões do mundo. No Brasil e em alguns outros países, a frequência é de 60 Hz, enquanto em outras regiões é de 50 Hz (ALUGAGERA, 2020). Essa frequência é determinada pela velocidade de rotação do gerador nas usinas elétricas e é mantida constante para garantir o correto funcionamento dos dispositivos elétricos que a utilizam. No Brasil como mencionado anteriormente utiliza-se por padrão a frequência de 60 Hz, ou sejam, a cada 1 segundo se completam 60 ciclos de uma oscilação positiva e uma oscilação negativa, como na figura 1.

Embora a corrente alternada seja considerada segura e confiável para uso doméstico, é fundamental adotar precauções para evitar choques elétricos. Recomenda-se que as instalações elétricas sejam realizadas por profissionais qualificados, pois isso garante a correta conexão dos fios e o cumprimento das normas de segurança.

2.1.2 RMS

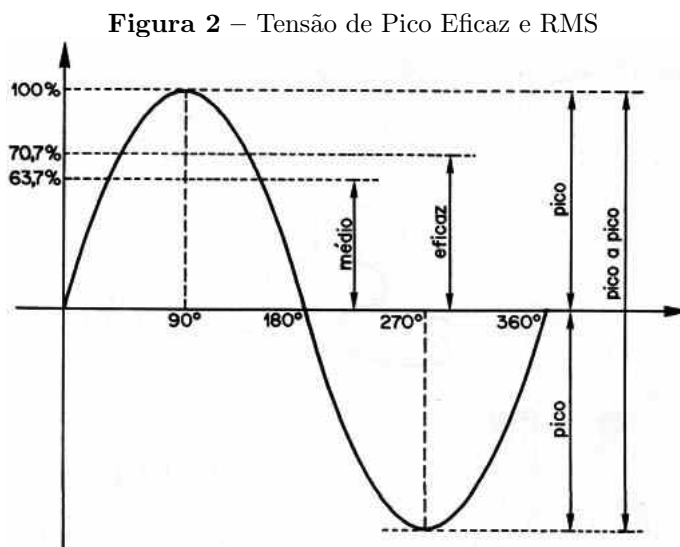
A tensão ou corrente elétrica RMS (Root Mean Square) é uma medida que representa a magnitude média de uma forma de onda ao longo do tempo, considerando tanto os valores positivos quanto os negativos. Essa medida é importante para determinar a quantidade de energia elétrica equivalente a uma corrente contínua de valor constante.

A importância do cálculo RMS reside no fato de que diferentes formas de onda variáveis podem produzir quantidades diversas de energia elétrica, dependendo de sua magnitude e duração. Por exemplo, uma onda quadrada com a mesma amplitude que uma onda senoidal gerará o dobro da energia elétrica se ambas estiverem presentes no mesmo intervalo de tempo. Assim, o valor RMS é utilizado como uma medida padronizada para comparar diferentes formas de onda em um circuito e compreender seu impacto energético.

Para calcular o valor RMS, é utilizada uma fórmula matemática específica. A relação entre o valor RMS e o valor de pico de uma função senoidal é uma constante de $1/\sqrt{2}$, ou aproximadamente 0,707. Portanto, para converter o valor de pico de uma onda senoidal em seu valor RMS, é necessário multiplicar o valor de pico por 0,707.

$$V_{\text{rms}} = 0.707 \times V_{\text{pico}} \quad (2.4)$$

Na figura 2 está demonstrado uma onda senoidal com os valores de pico, rms ,e média.



Fonte: Braga (2016)

Em resumo, o cálculo RMS é essencial para o projeto e dimensionamento de circuitos elétricos, permitindo a determinação da magnitude eficaz da corrente ou tensão elétrica. Ele também é utilizado em sistemas de monitoramento de energia elétrica, como medidores de consumo de energia elétrica, para medir com precisão a quantidade de energia elétrica consumida em um determinado período de tempo.

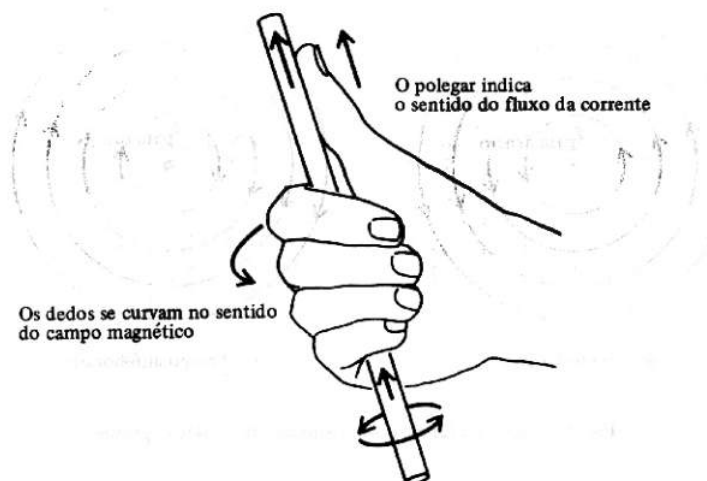
2.2 Eletromagnetismo

A indução eletromagnética é um fenômeno físico descoberto por Michael Faraday no século XIX. Ele observou que quando um campo magnético varia ou quando um condutor se move em relação a um campo magnético, uma corrente elétrica é induzida no condutor. Esse fenômeno é a base para o funcionamento de muitos dispositivos eletromagnéticos, incluindo o sensor de corrente TA12-100.

No caso do TA12-100, um núcleo ferromagnético é colocado próximo ao condutor pelo qual a corrente está fluindo. Quando a corrente elétrica passa pelo condutor, ela cria um campo magnético ao seu redor conforme a Lei de Ampere. Isso resulta em um fluxo magnético que de acordo com a Lei de Faraday, a variação desse fluxo magnético induz uma força eletromotriz no sensor. Essa força eletromotriz induzida é convertida em um sinal de saída proporcional à variação da corrente no condutor. Essa tensão induzida é então processada eletronicamente para fornecer uma saída que representa a corrente medida.

Aplicando a regra da mão direita se determina a direção do campo magnético em torno de um condutor com corrente elétrica, pode-se então usar a direção do campo magnético em relação ao sensor. Por isso o sensor possui uma direção correta para passar o condutor por dentro.

Figura 3 – Regra da mão direita



Fonte: Gussow (2005)

Essas leis e princípios da física são aplicados nos sensores de corrente não invasivos para medir a corrente elétrica em um condutor sem interromper o circuito, tornando-os uma ferramenta valiosa em diversas aplicações, já que ele é projetado para envolver o condutor, sem a necessidade de contato direto ou interrupção do fluxo de corrente. O que facilita e torna viável a qualquer pessoa fazer a instalação.

2.3 IoT

O termo Internet das Coisas (IoT), do inglês Internet of Things, foi proposto em 1999 por Kevin Ashton da MIT Auto Centre (ASHTON, 2009). A Internet das Coisas é uma tecnologia que permite a conexão e comunicação entre dispositivos, objetos e sistemas através da internet. A ideia é que objetos do dia-a-dia possam se conectar à internet e trocar informações, tornando o mundo mais inteligente e eficiente.

A IoT envolve a utilização de sensores, dispositivos conectados à internet, plataformas de gerenciamento de dados e algoritmos de inteligência artificial para coletar e analisar informações sobre o mundo físico. Essas informações são utilizadas para automatizar processos, melhorar a eficiência energética, otimizar a produção industrial, entre outras aplicações.

Por sua versatilidade, a IoT tem impactado diversos setores, desde a saúde, com a criação de dispositivos médicos inteligentes, até a indústria automotiva, com a criação de carros autônomos e conectados. A tecnologia também tem sido utilizada para melhorar

a gestão de cidades, com soluções de iluminação pública, monitoramento de tráfego e gerenciamento de resíduos.

Ainda que ofereça benefícios, a IoT também apresenta desafios, como a segurança dos dados e a privacidade dos usuários. A expansão da IoT tem sido objeto de discussão em vários setores, com o objetivo de garantir que os benefícios da tecnologia sejam alcançados de forma sustentável e responsável.

2.4 Microcontrolador

Microcontroladores são dispositivos eletrônicos que integram em um único chip um processador, memória, entradas e saídas, além de outras funcionalidades. Eles são amplamente utilizados em sistemas embarcados, onde o controle e a automação de dispositivos eletrônicos são necessários.

Estes dispositivos podem ser programados para realizar diversas tarefas, desde simples operações de controle de temperatura até sistemas mais complexos, como sistemas de comunicação sem fio. Eles são muito utilizados em aplicações industriais, automotivas, médicas, de segurança e muitas outras.

O processo de programação de um microcontrolador é realizado por meio de uma linguagem de programação específica, geralmente em um ambiente de desenvolvimento integrado (IDE). O programador deve conhecer bem a arquitetura do microcontrolador para poder desenvolver códigos eficientes e seguros.

Os microcontroladores estão cada vez mais presentes em nossas vidas, fazendo parte de equipamentos e dispositivos eletrônicos que utilizamos diariamente, como smartphones, televisores, sistemas de controle de acesso, entre outros. Com o avanço da tecnologia, espera-se que o uso de microcontroladores continue a crescer, tornando-se cada vez mais presentes em nossa rotina. Tanto é que nos dias atuais existem diversos microcontroladores de desenvolvimento que tem como ênfase a facilidade de uso e na acessibilidade para desenvolvedores iniciantes ou menos experientes. Eles são projetados para serem amigáveis ao usuário e possuem recursos que simplificam a programação e o teste de projetos eletrônicos. Alguns exemplos populares são, o Arduino, ESP8266, ESP32, PIC, entre outros.

2.4.1 ESP-32

O ESP-32 é um microcontrolador desenvolvido pela empresa chinesa Espressif Systems, que possui capacidades avançadas de comunicação sem fio e conectividade com a internet. Ele possui dois núcleos de processamento de 32 bits, o que o torna capaz de executar diversas tarefas ao mesmo tempo.

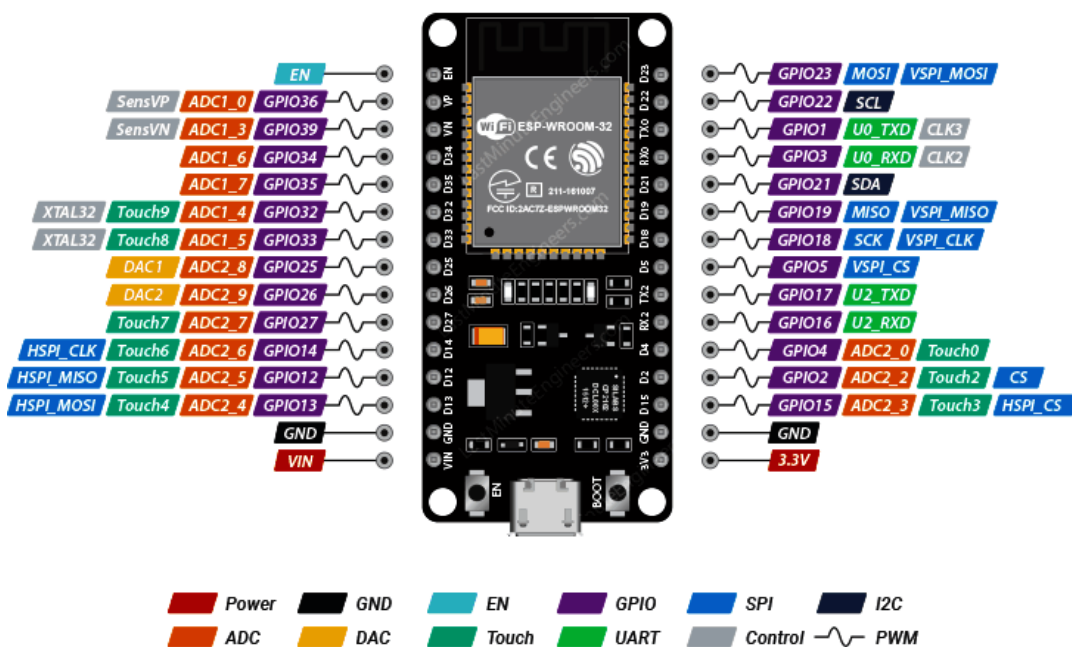
O ESP-32 possui um sistema integrado de comunicação Wi-Fi e Bluetooth, o que o torna ideal para aplicações de IoT (Internet das Coisas), como sistemas de automação residencial, sensores remotos, controle de equipamentos eletrônicos e muitas outras aplicações. Além disso, ele possui diversas interfaces de entrada e saída, como GPIOs, I2C, SPI, entre outras, o que permite a conexão com diversos dispositivos externos.

O microcontrolador possui uma arquitetura de baixo consumo de energia, o que o torna ideal para aplicações que utilizam baterias ou outras fontes de energia limitadas. Ele também possui uma ampla comunidade de desenvolvedores e usuários, o que garante a disponibilidade de diversas bibliotecas e ferramentas de desenvolvimento. Por estes motivos de conexão e várias portas analógicas que o ESP-32 foi escolhido para este protótipo.

Os pinos General Purpose Input/Output (GPIO) são responsáveis por receber ou enviar informações digitais a outros pinos, já os pinos ADC (Analog to Digital Converter) permitem também receber ou enviar informações a outros pinos, porém de entrada ou saída analógicas.

A Figura 4 ilustra o ESP-32 e sua pinagem, representando o que cada porta do dispositivo oferece.

Figura 4 – ESP32 Pinout



ESP32 Dev. Board Pinout

2.5 Linguagens Desenvolvimento do software protótipo

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software). Um software pode ser desenvolvido para rodar em um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução. Existem várias linguagens e elas servem para muitos propósitos. Alguns óbvios, como criar um software, outros menos, como controlar um carro ou uma torradeira. A linguagem de programação é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, que permite que um programador especifique precisamente quais os dados que o computador irá atuar, como irá ler, tratar e armazenar dados, e quais ações devem ser tomadas de acordo com as circunstâncias. de implementação de um código fonte que informará instruções de processamento ao computador. A seguir será citado as linguagens que foram utilizadas no protótipo.

2.5.1 C++

C++ é uma linguagem tanto de alto quanto de baixo nível, pois possuem recursos como gerenciamento automático de memória, sintaxe mais expressiva, tanto o que permite que sistemas desenvolvidos nessa linguagem sejam de alto desempenho, estáveis e seguros. que foi desenvolvida a partir da linguagem C. Foi criada na década de 1980 por Bjarne Stroustrup e tem sido amplamente utilizada desde então, principalmente no desenvolvimento de sistemas operacionais, softwares de jogos, aplicativos de desktop, sistemas embarcados, entre outros.

Uma das principais características do C++ é a sua capacidade de encapsular dados e funções em classes, permitindo uma maior modularização e reutilização de código. A linguagem também possui recursos avançados, como polimorfismo, herança, templates e sobrecarga de operadores, que facilitam o desenvolvimento de programas complexos.

Outra vantagem do C++ é a sua alta performance, que é possibilitada pelo seu compilador eficiente e pela possibilidade de trabalhar diretamente com a memória do computador. Porém, isso também exige que o programador tenha um bom entendimento de gerenciamento de memória e segurança do código.

A Base para programação do Arduino é o C/C++, o que a torna familiar para programadores que já estão familiarizados com essas linguagens. No entanto, ela possui algumas diferenças e simplificações para tornar a programação mais acessível, mas podemos dizer que a linguagem de programação utilizada na IDE do Arduino é C++.

2.5.2 PHP

PHP (Hypertext Preprocessor) é uma linguagem de programação de código aberto para construção de sites dinâmicos e aplicações *web*. Segundo (DOYLE, 2011), os pro-

gramas PHP rodam em um servidor *web* e disponibiliza páginas *web* aos visitantes que requisitarem. Estas páginas são dinâmicas, ou seja, o conteúdo muda automaticamente cada vez que são visualizadas.

O PHP também pode ser escrito sob o paradigma de orientação a objetos. Classes, propriedades e métodos; herança, polimorfismo, e encapsulamento. O PHP trata objetos da mesma maneira que referências e manipuladores, significando que cada variável tem uma referência a um objeto ao invés de uma cópia de todo o objeto.

Outra vantagem do PHP é a sua grande comunidade de desenvolvedores, que oferece uma ampla variedade de bibliotecas e frameworks para agilizar o desenvolvimento de aplicações. A linguagem também é fácil de integrar com outras tecnologias *web*, como HTML, CSS e JavaScript que será vista adiante.

2.5.3 Javascript

JavaScript é uma linguagem de programação versátil e amplamente utilizada para desenvolvimento *web*. Criada em 1995 por Brendan Eich, ela permite a criação de sites interativos, dinâmicos e responsivos. Sendo uma linguagem de script do lado do cliente, o JavaScript é executado diretamente no navegador, o que o torna uma escolha popular para adicionar interatividade e funcionalidade aos sites. (MOZILLA, 2023)

Uma das principais características do JavaScript é a capacidade de manipular o conteúdo de uma página *web* em tempo real. Por meio do chamado Document Object Model (DOM), o JavaScript permite que os desenvolvedores acessem e modifiquem os elementos HTML, estilos CSS e comportamentos de um site. Isso permite a criação de interfaces interativas e a atualização de conteúdo de forma dinâmica, sem a necessidade de recarregar a página.

Além disso, o JavaScript também é utilizado para a criação de animações, validação de formulários, manipulação de eventos, comunicação assíncrona com servidores através de requisições AJAX que foram bastante utilizadas para atualização de gráficos em tempo real no projeto. Ele oferece suporte a uma ampla gama de recursos e bibliotecas, como o popular framework jQuery, que simplifica tarefas comuns de desenvolvimento.

Hoje, ele é usado para desenvolver aplicativos móveis híbridos usando frameworks como o React Native e o Ionic, bem como para criar aplicativos de desktop usando tecnologias como o Electron. Essa versatilidade faz do JavaScript uma escolha essencial para os desenvolvedores que desejam criar aplicações modernas e interativas em diversas plataformas.

2.5.3.1 Chart.JS

Biblioteca de gráficos em JavaScript que foi adicionada para criar visualizações interativas de dados na página *web*. Ele oferece uma maneira simples e flexível de exibir dados em vários tipos de gráficos, como barras, linhas, pizzas e muito mais.

Ao utilizar o Chart.js, você pode criar gráficos dinâmicos e responsivos em seu site ou aplicativo da *web*. Ele é construído em cima da linguagem JavaScript e usa a *tag* HTML5 canvas para renderizar os gráficos.

O Chart.js também fornece métodos para atualizar e modificar os gráficos dinamicamente. Você pode adicionar ou remover dados, alterar as opções de exibição e atualizar o gráfico para refletir as alterações em tempo real.

Em resumo, o Chart.js é uma biblioteca JavaScript poderosa e versátil para criar gráficos interativos. Ele simplifica o processo de visualização de dados e oferece uma variedade de opções de personalização. Que facilitou e muito já que a página do usuário desenvolvida no projeto conta com seis gráficos distintos.

2.5.4 HTML

HTML é uma forma de dizer a um navegador como exibir uma página (WILLARD, 2009), oferecendo uma forma de marcar o texto com etiquetas (tags) que informam ao navegador como o texto está estruturado.

Com estas tags, é possível criar estruturas de conteúdo na forma de textos, imagens, vídeos e outros elementos, que podem ser visualizados em diferentes dispositivos e navegadores, o HTML é frequentemente combinado com outras tecnologias *web*, como CSS (Cascading Style Sheets) e JavaScript, para criar páginas mais sofisticadas e interativas.

Atualmente, HTML oferece uma série de recursos como textos, links, imagens e vídeos, além de incluir estilo de apresentação e scripts. Conforme conta (HENICK, 2010), HTML está em desenvolvimento constante desde 1992 e os navegadores evoluíram muito rapidamente. Um site com boa interface do usuário dá ferramentas para que os usuários naveguem com o mínimo de trabalho.

2.5.5 MySQL

O MySQL é um sistema de banco de dados relacional de código aberto, desenvolvido pela Oracle Corporation. (KOFILER, 2005) diz que MySQL é considerado por muitos, o mais rápido, confiável e barato sistema de banco de dados entre os outros sistemas existentes.

Com ele, é possível armazenar e gerenciar grandes quantidades de dados, permitindo o acesso a informações estruturadas por meio de linguagens de programação como

PHP, Java e Python. O sistema é compatível com diversas plataformas e sistemas operacionais, o que o torna uma opção versátil para desenvolvedores.

Além disso ele dispõe de funções onde é possível através do uso destas, fazer cálculos aritméticos para receber resultados já processados entre datas, como uma média ao longo de um ano com suas determinadas condições.

O MySQL é amplamente utilizado em todo o mundo, desde pequenos sites até grandes empresas, sendo uma das escolhas mais populares de sistemas de gerenciamento de bancos de dados relacionais.

2.6 Softwares de Desenvolvimento

Para implementar programas em um Arduino e também para o desenvolvimento da aplicação *web*, é necessário utilizar uma IDE, onde todo o código é escrito, são feitos testes para encontrar possíveis erros e no caso do Arduino para também transferir o programa para a placa. Além da IDE de desenvolvimento, outros softwares que auxiliam na visualização e tratamento dos dados também serão citados abaixo.

2.6.1 Visual Studio Code

O Visual Studio Code é um editor de código-fonte gratuito desenvolvido pela Microsoft, que oferece suporte a diversas linguagens de programação, incluindo JavaScript, TypeScript, Python, C++, entre outras. É um editor de código aberto, multiplataforma e gratuito, disponível para Windows, macOS e Linux. Nele também é possível instalar extensões que adicionam funcionalidades adicionais, como o recurso que iremos utilizar, o PlatformIO IDE, que é um ecossistema de código aberto para desenvolvimento de software para sistemas embarcados.

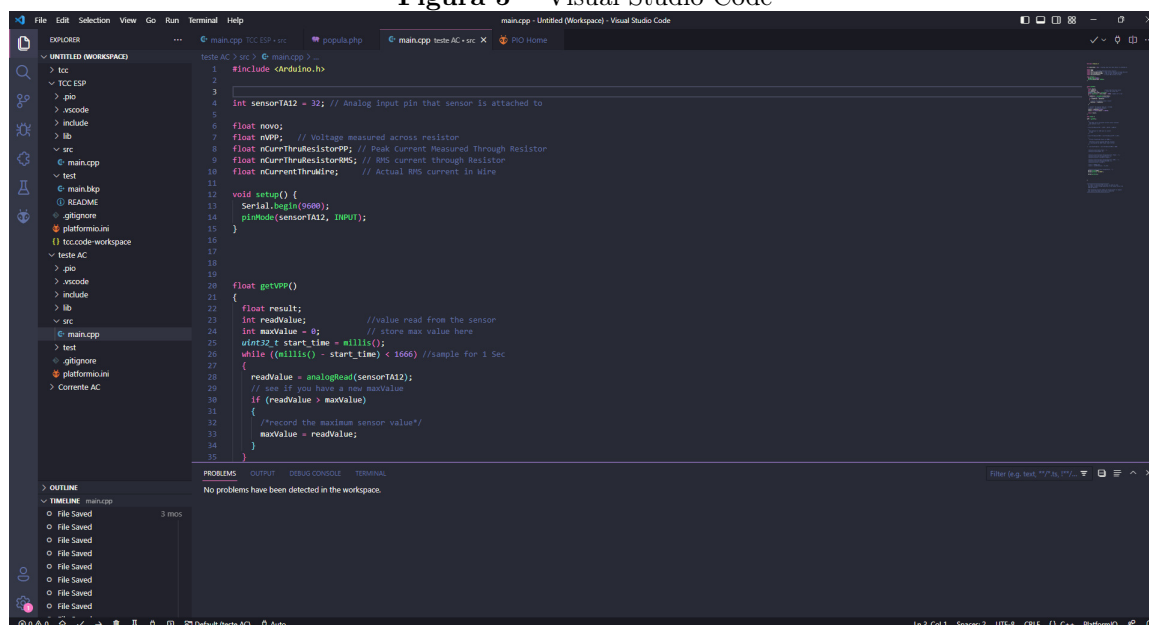
O software é altamente customizável, permitindo que os desenvolvedores personalizem sua interface, instalem extensões e ajustem as configurações para se adequarem às suas necessidades específicas de desenvolvimento. Ele possui recursos avançados de edição de texto, como realce de sintaxe, sugestões de código e integração com ferramentas de depuração e controle de versão.

Com sua ampla gama de recursos e ferramentas, o Visual Studio Code se tornou uma das escolhas mais populares para desenvolvedores em todo o mundo, ajudando a simplificar e agilizar o processo de desenvolvimento de software.

Ou seja, com ele ficou unificado a apenas um software, a utilização para a etapas de programação do projeto, tanto a parte Arduino, a parte de front-end da página *web*, como a parte de *back-end*.

Na figura 5 está a demonstração do software.

Figura 5 – Visual Studio Code



Fonte: Autoria própria

2.6.2 PlataformIO

PlataformIO é uma plataforma de desenvolvimento de código aberto para sistemas embarcados, projetada para facilitar o desenvolvimento de aplicativos para dispositivos IoT (Internet das Coisas). Ele é utilizado como uma extensão para o VS Code, e trás um ambiente de desenvolvimento integrado (IDE) que oferece suporte a uma variedade de plataformas e ferramentas, incluindo Arduino, ESP32, ESP8266, STM32, Raspberry Pi, entre outras.

Como ele oferece suporte para diversos tipos de placas, o que possibilita desenvolver para diversos microcontroladores utilizando apenas uma ferramenta. Assim que se cria o projeto, é selecionado o microcontrolador a ser utilizado no projeto para comunicação correta entre o software e o hardware. Ele também inclui um gerenciador de bibliotecas integrado, permitindo que os desenvolvedores acessem e gerenciem bibliotecas de código de terceiros.

Outra característica importante do PlataformIO é sua capacidade de fornecer suporte integrado para ferramentas de construção e depuração, incluindo compiladores e depuradores, tornando mais fácil para os desenvolvedores construir, testar e depurar seus aplicativos. Transfere para a placa

Ele ajuda a simplificar e agilizar o processo de desenvolvimento de aplicativos para sistemas embarcados, tornando mais fácil para os desenvolvedores criar aplicativos robustos e escaláveis para dispositivos IoT.

2.6.2.1 Bibliotecas PlatformIO

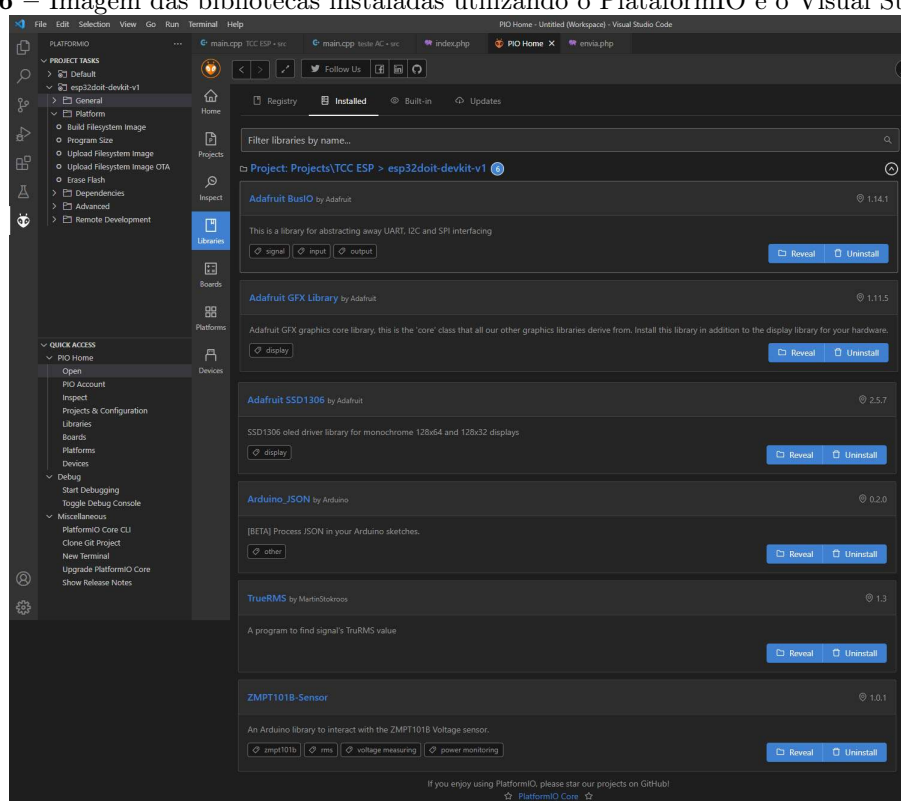
As bibliotecas no PlatformIO são pacotes de software reutilizáveis que contêm conjuntos de funções e classes que podem ser usadas para realizar tarefas específicas. Essas bibliotecas são criadas por desenvolvedores e comunidades para facilitar o desenvolvimento de projetos. O PlatformIO possui um gerenciador de bibliotecas embutido que permite pesquisar, instalar e gerenciar bibliotecas de maneira fácil e conveniente.

Essas bibliotecas fornecem uma variedade de funcionalidades, como comunicação com periféricos, manipulação de sensores, controle de motores, exibição de informações em *display's*, entre outras.

Com o uso adequado de bibliotecas, se pode adicionar facilmente funcionalidades extras ao seu projeto sem ter que desenvolvê-las do zero. Uma vez que uma biblioteca esteja instalada, você pode incluí-la no seu código fonte e usar suas funções e classes.

Um exemplo da tela de instalação das bibliotecas está demonstrado abaixo na figura 6

Figura 6 – Imagem das bibliotecas instaladas utilizando o PlatformIO e o Visual Studio Code



Fonte: Autoria própria

2.6.2.1.1 Arduino.h

Esta biblioteca é a principal a ser usada no projeto pois com ela é possível o programar utilizando funções do Arduino. Ela contém uma coleção de funções e macros que simplificam a programação de dispositivos Arduino, permitindo criar projetos de maneira mais rápida e fácil.

A biblioteca Arduino.h fornece acesso a recursos de hardware, como pinos de entrada e saída, comunicação serial e temporizadores. Ela também inclui funções úteis para realizar tarefas comuns, como a leitura de sensores ou a escrita em displays.

Outra característica vantagem da biblioteca Arduino.h é sua documentação detalhada e exemplos de uso que tem uma alta facilidade de encontrar recursos online. Isso permite que iniciantes e experientes aprendam a usar a biblioteca de maneira eficaz e rápida.

Em resumo, a biblioteca Arduino.h é um componente fundamental no desenvolvimento de projetos com dispositivos Arduino. Ela fornece uma interface simplificada para o hardware, permitindo que desenvolvedores controlem o funcionamento de seus dispositivos e criem projetos complexos de maneira mais fácil e rápida. Além do que ela permite a adição de bibliotecas próprias para o Arduino ao projeto.

2.6.2.1.2 Wifi.h

A biblioteca WiFi.h é uma biblioteca usada pelo Arduino que fornece funções para configurar e utilizar a conectividade Wi-Fi em placas compatíveis com o Arduino. Ela é especialmente útil quando se deseja conectar o Arduino a uma rede Wi-Fi, permitindo a comunicação com outros dispositivos conectados à mesma rede.

Com a biblioteca WiFi.h, é possível realizar configurar e conectar a placa a uma rede Wi-Fi com comunicação em redes locais ou até mesmo com a internet, dependendo da configuração da rede em que a placa está conectado. fornecendo o nome da rede (SSID) e senha, ele também possui funções de verificar o status da conexão do Wifi com a placa, entre outras funções relacionadas a rede sem fio.

Essas são apenas algumas das funcionalidades principais da biblioteca WiFi.h do Arduino. Ela permite ao Arduino se conectar a redes Wi-Fi.

2.6.2.1.3 ZMPT101B.h

Com a biblioteca ZMPT101B, os usuários podem adquirir os valores da tensão AC de maneira precisa e ajustada para posterior processamento. Isso é particularmente útil em projetos que exigem monitoramento da tensão da rede elétrica, como sistemas de automação residencial, dispositivos de controle de energia e aplicações de monitoramento de energia.

No geral, a biblioteca ZMPT101B inclui funções que simplificam a interação com o sensor ZMPT101B, permitindo que os desenvolvedores obtenham informações confiáveis sobre a tensão AC com mais facilidade e eficiência.

2.6.2.1.4 AdafruitGFX.h

A biblioteca AdafruitGFX é uma biblioteca de gráficos para placas e dispositivos compatíveis com o Arduino. Ela fornece um conjunto de funções que permitem criar e manipular elementos gráficos, como texto, formas geométricas e pixels, em displays ou telas conectadas ao Arduino.

No projeto será utilizado um display OLED e esta biblioteca se faz necessária para facilitar a escrita dos textos em diferentes tamanhos de fonte, cores e estilos. A biblioteca permite posicionar o texto em coordenadas específicas na tela e escolher fontes pré-definidas ou personalizadas.

Também é possível desenhar formas geométricas já que a biblioteca oferece funções para desenhar linhas, retângulos, círculos, elipses e outros elementos geométricos. Você pode especificar a posição, tamanho, cor e estilo dessas formas.

Simplificando, ela auxilia no processo de criação de elementos gráficos em displays, tornando mais fácil a criação de interfaces visuais ou visualizações personalizadas em projetos.

2.6.2.1.5 HttpClient.h

A biblioteca HTTPClient é uma biblioteca do Arduino que permite realizar solicitações HTTP (Hypertext Transfer Protocol) a servidores *web*. Ela oferece recursos para enviar requisições HTTP, receber e processar respostas, além de lidar com diferentes tipos de dados, como texto e arquivos.

Graças a biblioteca HTTPClient, é possível enviar solicitações HTTP, como GET, POST, PUT, DELETE e outras, para um servidor *web*. No projeto é especificado o URL de destino, os cabeçalhos HTTP necessários e os parâmetros da solicitação. E assim o servidor processa os dados após este envio.

Amplamente utilizada em projetos Arduino, esta biblioteca é essencial para os que requerem uma comunicação com servidores *web*, como acesso a APIs, envio de dados para serviços de nuvem ou obtenção de informações da *web*. Ela simplifica o processo de envio e recebimento de dados através do protocolo HTTP, facilitando a integração do Arduino com a internet.

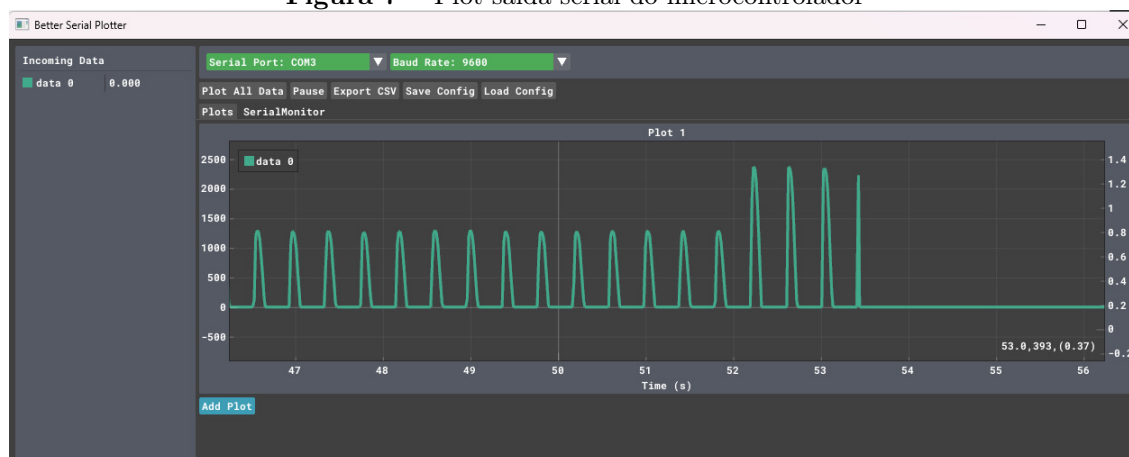
2.6.3 BetterSerialPlotter

BetterSerialPlotter é uma ferramenta gratuita de visualização de dados em tempo real para dispositivos Arduino. Ele permite que desenvolvedores exibam dados de sensores, entradas analógicas e outros valores gerados por seus projetos em gráficos, tabelas e visualizações de tendências.

O BetterSerialPlotter é uma alternativa mais poderosa e flexível ao Serial Plotter padrão incluído nas IDE's. Já que ela oferece uma interface gráfica, para que fosse feita a análise dos dados em tempo real. Além do que possui a capacidade de exportar dados para CSV e analisá-los em um software de planilhas.

Na figura 7 abaixo é demonstrado o software com os dados de leitura da corrente elétrica gerada por um secador de cabelo em duas velocidades distintas, obtidos na porta serial do próprio projeto.

Figura 7 – Plot saída serial do microcontrolador



Fonte: Autoria própria

2.6.4 TablePlus

O TablePlus é um software de gerenciamento de banco de dados que oferece uma interface intuitiva e eficiente para administrar várias bases de dados relacionais, incluindo MySQL. Ele é amplamente utilizado por desenvolvedores, administradores de banco de dados e analistas para gerenciar, consultar e manipular dados de bancos de dados.

Foi utilizada a versão gratuita do software que possui algumas limitações em relação a versão paga, como o número máximo de abas, janelas e filtros avançados, que não eram necessárias para o presente projeto. Sua interface amigável e limpa se torna o destaque que oferece uma experiência de usuário agradável, permitindo que os usuários naveguem pelas bases de dados, visualizem esquemas, editem registros e executem tarefas de gerenciamento com facilidade por estes motivos da escolha do software em questão.

Abaixo na figura 8 é demonstrado uma ilustração da ferramenta com a estrutura da tabela utilizada no projeto.

Figura 8 – Software TablePlus

The screenshot shows the TablePlus interface with the 'medicao' table selected. The table structure is as follows:

#	column_name	data_type	character_set	collation	is_nullable	column_default	extra	foreign_key	comment
1	idmedicao	int(10)	NULL	NULL	NO	NULL	auto_increment	EMPTY	→ EMPTY
2	idconsumidor	int(11)	NULL	NULL	NO	NULL	EMPTY	EMPTY	→ EMPTY
3	data	datetime	NULL	NULL	NO	NULL	EMPTY	EMPTY	→ EMPTY
4	tensao	float	NULL	NULL	NO	NULL	EMPTY	EMPTY	→ EMPTY
5	corrente	float	NULL	NULL	NO	NULL	EMPTY	EMPTY	→ EMPTY
6	watts	float	NULL	NULL	NO	NULL	EMPTY	EMPTY	→ EMPTY

Below the table structure, the index information is displayed:

index_name	index_algorithm	is_unique	column_name
PRIMARY	BTREE	TRUE	idmedicao

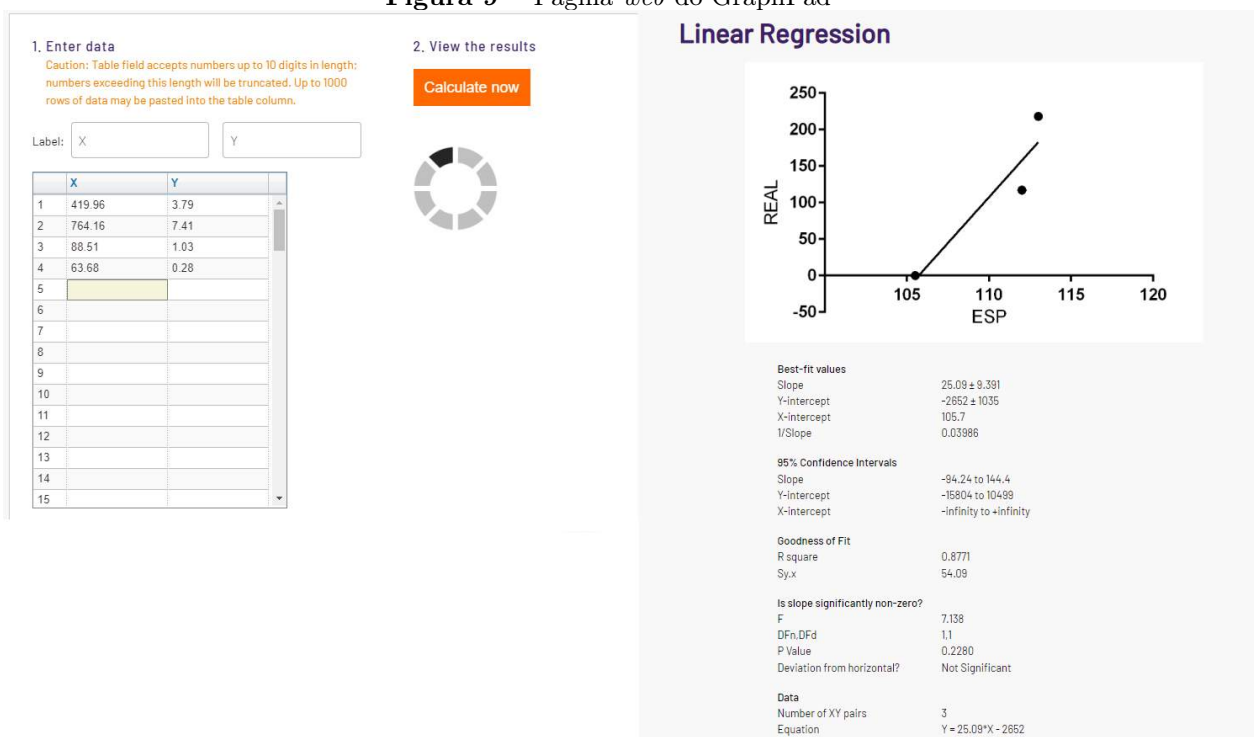
Fonte: Autoria própria

2.6.5 GraphPad

Ferramenta online (GRAPHPAD, 2022) para gerar a equação da reta, com ela você define os valores do eixo X e Y da reta e ela automaticamente gera a sua equação de reta, este software foi utilizado para equivalência do valor mostrado no Serial do ESP-32 comparado ao valor obtido no multímetro.

Na imagem 9 está um modelo feito com os valores obtidos no sensor.

Figura 9 – Página web do GraphPad



Fonte: Autoria própria

3 MATERIAIS E MÉTODOS

Neste capítulo será apresentada a metodologia usada para montagem e desenvolvimento do protótipo apresentado neste trabalho. Na primeira seção, será mostrado os materiais utilizados, suas características e o processo de montagem. Nas seções subsequentes, serão apresentados os métodos empregados, juntamente com trechos do código-fonte referentes tanto à implementação na placa ESP-32 quanto à construção do servidor *web*.

3.1 Materiais utilizados

Os materiais utilizados neste projeto foram selecionados com o objetivo de embasar e fundamentar o protótipo de forma consistente e flexível ao mesmo tempo, oferecendo uma facilidade de ajuste e uma pequena interface para o usuário final, para fornecer informações necessárias e alcançar o objetivo do protótipo em questão, foi preciso interligar ao microcontrolador um sensor de tensão (ZMPT101B) que foi conectado em paralelo a rede elétrica residencial, um sensor de corrente (TA12-100) com conexão não invasiva a rede elétrica e um *display* OLED 128x64 para exibir informações ao usuário sem a necessidade de um computador prontamente no local. Além de um *proto-board* e conectores do tipo WAGO para as conexões entre os materiais. Abaixo, na tabela 1 está a lista de todos os materiais utilizados assim como suas ilustrações na figura 10.

Tabela 1 – Tabela com os materiais utilizados para o desenvolvimento do protótipo

Item	Quantidade	Especificação
1	1	Microcontrolador ESP-PICO-V3-02
2	2	Proto-board 170 pontos
3	3	Conector de Emenda WAGO 3 Vias
4	1	Sensor de Tensão ZMPT101B
5	1	Sensor de Corrente TA12-100
6	1	Display OLED 128x64
7	1	Wattímetro
8	1	Multímetro com Alicates Amperímetro

Figura 10 – Imagem de todos os componentes utilizados para o protótipo

Fonte: Autoria própria

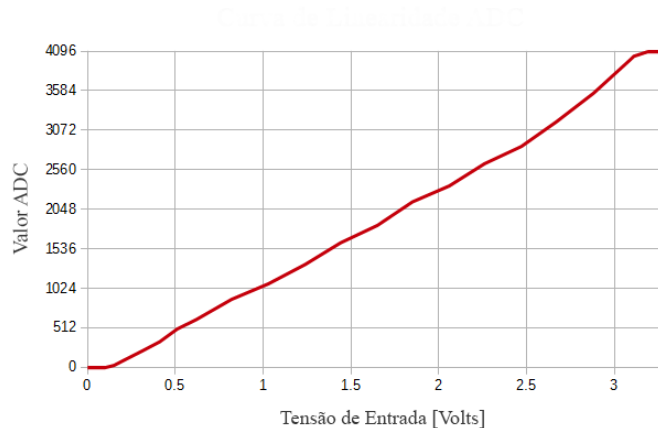
3.2 Metodologia

Nesta seção, será demonstrado o passo a passo e métodos para a montagem e execução final do protótipo.

3.2.1 Funcionamento dos componentes

No início do estudo, foram dedicados esforços para tentar o funcionamento do microcontrolador Arduino, porém após inúmeras tentativas de conciliar o microcontrolador com a rede sem fio Wifi sem sucesso, mesmo trocando componentes e firmwares dos mesmos, após essa falhas foi decidido a troca de placa. Com a troca para o ESP-32 foi necessário compreender em detalhes o funcionamento deste microcontrolador ESP-32, uma vez que este é o componente central do protótipo em questão. Por meio da análise da figura 4, foi possível identificar todos os terminais do microcontrolador e compreender as funções desempenhadas por cada um deles. A fim de desenvolver o projeto proposto, foram selecionados os pinos de leitura analógica, os terminais de alimentação da placa, assim como os terminais de SDA e SCL, que foram utilizados para integrar o *display* OLED ao sistema.

Além disso, foi imprescindível adquirir um conhecimento sobre a curva de linearidade das entradas ADC do ESP-32, conforme ilustrado na figura 11. Essa curva representa a relação entre a tensão recebida pelo microcontrolador e o valor de saída obtido na porta Serial. A compreensão dessa relação é fundamental para garantir a precisão e a confiabilidade das medições efetuadas pelo sistema.

Figura 11 – Curva de Linearidade ADC

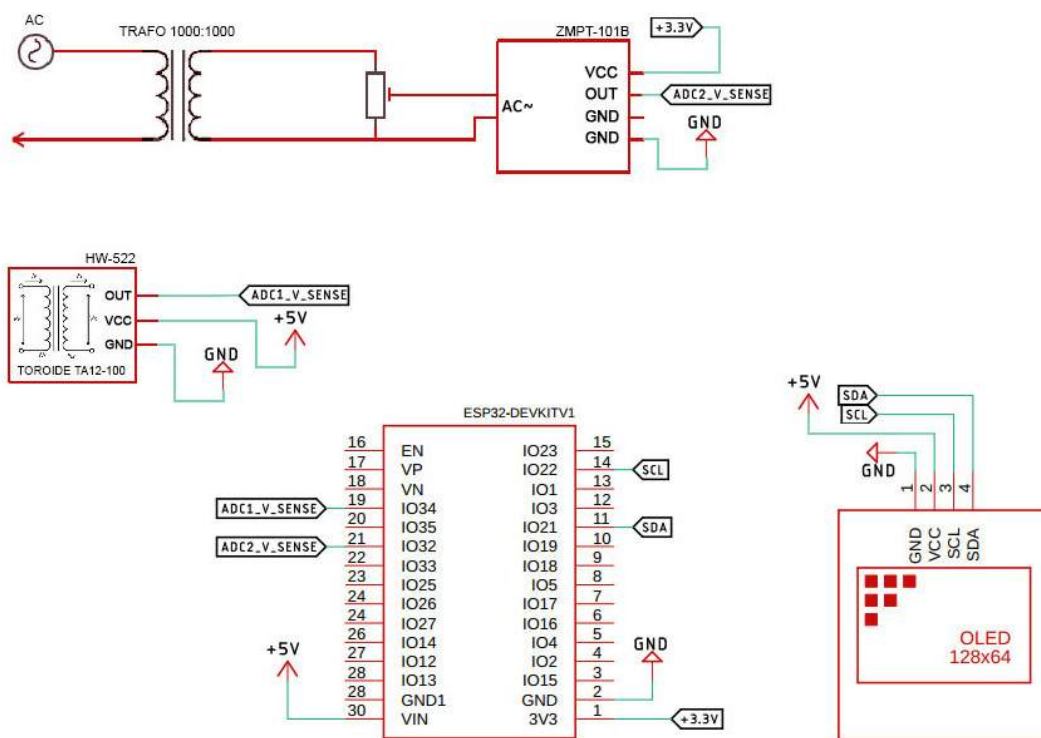
Fonte: ESP32Forum (2022)

Após o estudo detalhado do microcontrolador ESP-32, tornou-se necessário também realizar uma análise dos *datasheets* dos demais componentes eletrônicos presentes no projeto, tais como do *display* OLED (VISHAY, 2022), do sensor de tensão (DATASHEETSPDF, 2022) e do sensor de corrente (ELECTRONICOSCALDAS, 2022). Essa etapa teve como objetivo compreender as especificações de entrada de tensão e de dados em cada componente, analisando os pinos adequados para a correta conexão com a placa ESP-32, garantindo assim uma integração entre eles.

3.2.2 Montagem do protótipo

Com os componentes necessários a disposição, foi feito um esquemático para conexão dos componentes com o ESP-32 utilizando os pinos mencionados anteriormente, o esquemático está demonstrado na figura 12.

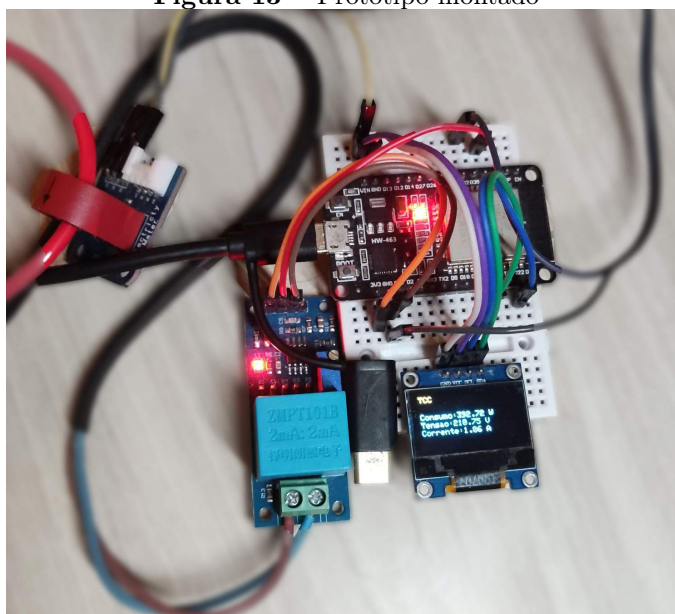
Figura 12 – Imagem do esquemático do projeto



Fonte: Autoria própria

Em seguida foi realizada a montagem do *hardware* no *protoboard* seguindo exatamente as ligações descritas pelo esquemático, com a montagem final demonstrada na figura 13.

Figura 13 – Protótipo montado



Fonte: Autoria própria

3.2.3 Testes iniciais no microcontrolador

Com a montagem do protótipo concluída foi feita a conexão do ESP-32 ao computador através de um cabo de dados, possibilitando a transferência do código-fonte contendo as instruções que o microcontrolador irá executar.

A fase inicial do projeto envolveu testes abrangentes no microcontrolador escolhido, como testes simples de piscar um LED e também o teste de conexão Wi-fi do dispositivo, testes estes visando verificar a integridade do dispositivo e o conhecimento de seu funcionamento.

Em seguida, para cada dispositivo periférico agregado ao microcontrolador, foi realizado a instalação de suas bibliotecas apropriadas e realizado um teste exemplo que era acompanhado dos arquivos de cada biblioteca, este foi um passo crucial para garantir a comunicação eficaz entre o microcontrolador e cada um destes dispositivos separadamente.

3.2.4 Conversão e calibração dos dados

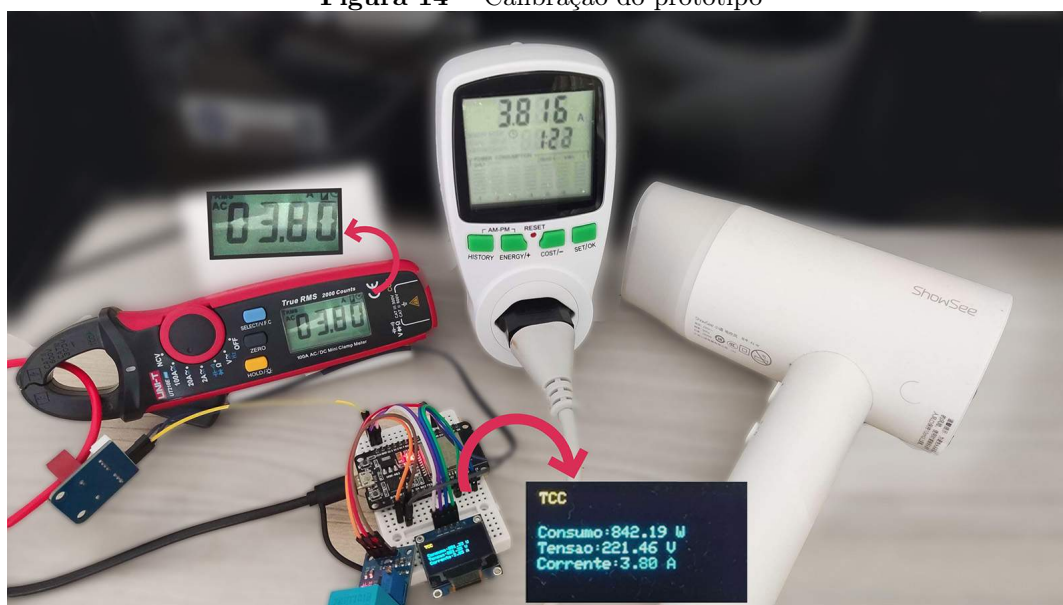
Nesta etapa foi feita a conversão dos valores que eram retornados pelo microcontrolador para valores em unidades de medidas utilizadas no Sistema Internacional de Unidades (SI), ou seja, Volts, Amperes, Watts.

Para a conversão da tensão foi utilizada a biblioteca ZMPT101B.h, a função desta biblioteca recebe a frequência e retorna o valor desejado em Volts, para checagem foram feitos testes na tomada residência e em um estabilizador com saída 110 V.

Já para a conversão da corrente elétrica para Watts outro método foi utilizado, foi criada uma função que checa continuamente os valores retornados por 1666 milissegundos, uma amostragem de 60 Hz, que é a frequência da rede local. Desta amostragem foi armazenado em uma variável o valor máximo, ou seja o valor de pico, e em seguida feito a multiplicação RMS para obter o valor em Amperes. Na averiguação foram realizados testes com aparelhos diferentes como secador de cabelo, liquidificador e ferro de solda.

A fim de garantir a acurácia das medições realizadas pelo protótipo, foi comparado o valor exibido no multímetro com o valor convertido obtido pelo microcontrolador, em seguida foi utilizado a ferramenta online GraphPad para geração da fórmula da equação de reta. Posteriormente feita a aplicação da fórmula matemática ao resultado de retorno, assim mantendo o valor em conformidade com o multímetro. O multímetro utilizado em questão foi o UNI-T modelo UT210E que de acordo com seu *datasheet* possui uma precisão de $\pm(2.5\%+5)$ para medida CA na escala de 20 A. A seguir na figura 14 está demonstração do processo de calibração do resultado do microcontrolador.

Figura 14 – Calibração do protótipo



Fonte: Autoria própria

3.2.5 Envio e armazenamento dos dados

Através da conexão Wi-fi se torna possível para o microcontrolador enviar as leituras obtidas tanto da corrente como da tensão elétrica, o envio destes dados é feito através do protocolo HTTP utilizando o método GET, além dos dados mencionados também é enviado ao servidor a data e hora do momento de envio, assim como o código identificador do cliente, esse por sua vez passa por um processo de criptografia com o método *hash* SHA-256 no lado do servidor para garantir uma melhor segurança ao sistema.

O Servidor *web* recebe estes dados e a partir deles faz a conferência do identificador do cliente, realiza o cálculo aritmético do valor da potência elétrica e em seguida faz a inserção destes registros no banco de dados como será demonstrado nos tópicos a seguir de aplicações.

3.2.6 Aplicação no microcontrolador

O código-fonte a seguir demonstra uma parte das instruções de execução presentes no microcontrolador do projeto, o trecho comentado apresenta na linha 20 como as variáveis contendo as leituras dos componentes são enviadas para o servidor *web*.

```

1
2 void loop(){
3
4 //CHAMADA DAS FUNCOES DE TENSAO/CORRENTE E ARMAZENAMENTO EM
5 //VARIÁVEIS
6 nVPP = getCorrente();
7 float tensaoMedida = sensorVolt.getVoltageAC(60);

```

```
7
8   v = tensaoMedida;
9   a = nCPP;
10  w = v * a * fp;
11
12  //FUNCAO DE ESCREVER NO DISPLAY
13  funcTela(v, w, a);
14
15  if (WiFi.status() == WL_CONNECTED){
16  // Checa conexao WiFi
17    HTTPClient http;
18
19    // serverName = ENDERECO WEB ONDE O ESP32 IRA ENVIAR A
    REQUISICAO
20    http.begin(serverName);
21    http.addHeader("Content-Type", "application/x-www-form-
    urlencoded");
22    sprintf(httpRequestData, "volt=%f&=%f&idcliente=%d&
    hashcliente=%s", tensaoMedida, a, idcliente, hashcliente);
23
24    int httpResponseCode = http.POST(httpRequestData); // Envia o
    POST request
25    if (httpResponseCode > 0){
26      // retorna resultado da resposta em http
27      String response = http.getString();
28      Serial.println(httpResponseCode);
29      Serial.println(response);
30    }
31    else{
32      Serial.print("Erro no envio POST: ");
33      Serial.println(httpResponseCode);
34    }
35    http.end();
```

Código 3.1 – Código implementado no microcontrolador

3.2.7 Aplicação no servidor

Na fase do servidor é quando cada requisição enviada pelo microcontrolador é lida, e assim essas informações são tratadas e exibidas ao usuário. Nesta fase, a aplicação efetua a associação do usuário ao seu código identificador, através do valor enviado na etapa anterior e no processo de criptografia, assim sabe-se a qual usuário pertence os

dados recebidos. Na aplicação do servidor também é feito o cálculo da potência consumida naquele instante, como o protótipo tem o foco principal residências e circuitos monofásicos, assumiu o fator potencia sendo 0.92, pois este é o fator de potência referência da NDU-001 (ENERGISA, 2023). Após este cálculo, é efetuado o armazenamento destas informações no banco de dados como demonstrado no código A.3 abaixo.

```
1 <?php
2
3     date_default_timezone_set('America/Recife');
4     $hora = date('Y-m-d H:i:s');
5
6     //CONEXAO COM O BANCO DE DADOS
7     $conexao = mysqli_connect('localhost', 'u559028162_root', '
8         Senha123');
9     $banco = mysqli_select_db($conexao, 'u559028162_tcc');
10    mysqli_set_charset($conexao, 'utf8');
11
12    //VARIAVEIS COM OS VALORES RECEBIDOS DO ESP-32
13    $idcliente = $_REQUEST['idcliente'];
14    $voltage = $_REQUEST['volt'];
15    $corrente = $_REQUEST['amp'];
16
17    if($idcliente){
18
19        //CHECA A CHAVE ENVIADA COM O ID DO CLIENTE
20        $hashvem = $_REQUEST['hash'];
21        $hashnum = (int)$idcliente*5;
22        $hashcalc = hash('sha256', $hashnum);
23
24        if($hashvem == $hashcalc){
25            //CHECA SE VIERAM OS VALORES DO ESP-32
26            if ($voltage && $corrente) {
27                $fp = 0.92;
28                //CALCULO DE POTENCIA EM WATTS
29                $watt = $voltage * $corrente * fp;
30
31                //INSERCAO NO BANCO DE DADOS
32                $query = "INSERT INTO medicao ('idconsumidor', '
33                    data', 'tensao', 'corrente', 'watts') VALUES
34                    ('" . $idcliente . "' ,'" . $hora . "' ,'" .
35                    $voltage . "' ,'" . $corrente . "' ,'" . $watt
36                    . "')";
```



```
32         $result = mysqli_query($conexao, $query) or die
           ('Errant query: ' . $query);
33     if($result){
34         echo "Gravou com sucesso <br/>";
35         $myfile = fopen("correnteagora.txt", "w");
36         fwrite($myfile, $watt);
37         fclose($fp);
38         echo "Txt OK";
39     }
40
41     }
42 }
43 }
44 ?>
```

Código 3.2 – Código fonte em PHP

3.2.8 Consulta dos dados

Neste etapa foi desenvolvido um algoritmo contendo várias consultas MySQL diferentes para cada tipo de gráfico, estas consultas utilizam funções do próprio MySQL como função de média, valor máximo, somatório, intervalo de tempo, etc, que foram fundamentais para realizar todas as filtragens necessárias para a disposição dos valores corretos ao usuário. Um trecho do algoritmo esta exemplificado no código 3.3 abaixo com uma consulta de consumo mensal, o código retorna a média de consumo de cada dia do mês corrente.

```
1
2 //CONSULTA FILTRANDO A MEDIA DE TODOS OS DIAS DO MES ATUAL
3     $query = "SELECT idmedicao, AVG(tensao) as tensao, AVG(watts)
              as watts, DAY(data) as dia FROM medicao WHERE DATE_SUB('data
              ',INTERVAL 1 DAY) And MONTH(data) = MONTH(CURDATE()) GROUP
              BY DAY(data) ORDER BY idmedicao ASC;";
4
```

Código 3.3 – Trecho da consulta MySQL

3.2.9 Exibição dos dados na página web

Os valores obtidos na seção anterior foram trabalhados e dispostos em uma página HTML contendo gráficos e tabelas afim de exibir uma interface apresentável e com facilidades de usabilidade para o usuário final, estes foram exibidos com o auxílio da biblioteca ChartJS em Javascript.

Para utilização e disposição de um gráfico em tempo real sem a necessidade do recarregar a página, o que não contribui para usabilidade do site, foi preciso utilizar do recurso AJAX do Javascript, através disto o site consegue fazer a consulta e atualizar o gráfico automaticamente dando mais dinâmica para o usuário.

O visual da página e estes gráficos citados serão exibidos no próximo capítulo no qual será exposto os resultados deste projeto.

4 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos no final do protótipo. Os resultados serão apresentados de forma separada, primeiramente serão mostrados os resultados obtidos pelo microcontrolador, com os resultados de retorno da porta serial. Na segunda parte, será apresentada a parte dos resultados da página *web*, que consiste na parte gráfica com os dados já tratados por completo e a forma como é disponibilizado ao usuário final.

Ao analisar esses resultados, estaremos aptos a avaliar até que ponto as propostas de solução alcançaram seus alvos, além de reconhecer possibilidades de melhorias e conserto de falhas.

4.1 Resultados obtidos diretamente do microcontrolador

A figura 15 demonstra os resultados obtidos diretamente da porta ADC do microcontrolador, no caso da imagem, pelo sensor de tensão, uma série de números inteiros que precisaram ser plotados para demonstrar se o resultado retornado estava de acordo com o teórico, ou seja, uma a onda senoidal da rede elétrica.

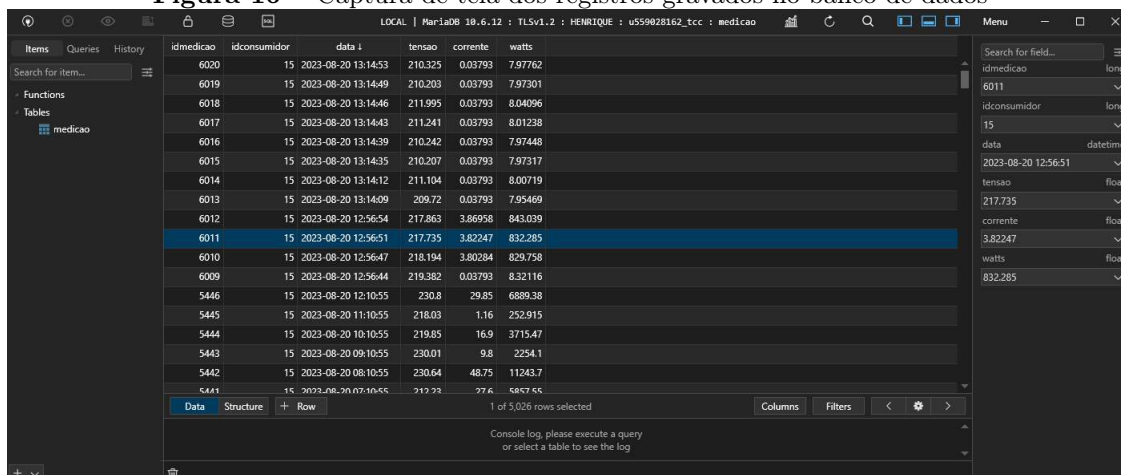
Figura 15 – Captura de tela da saída serial do microcontrolador plotada



Fonte: Autoria própria

Após feita a conversão e tratamento dos dados obtidos no microcontrolador, estes valores foram enviados ao servidor e gravados automaticamente a cada 3 segundos no banco de dados. Este intervalo de tempo foi escolhido para manter uma certa precisão dos valores consumidos pelo cliente. Deverá ser feita outra análise futura com mais práticas e com mais usuários para obter um intervalo ideal. A figura ?? demonstra cada registro gravado no banco de dados, alertando a preocupação sobre a quantidade de registros por minuto com o intervalo escolhido.

Figura 16 – Captura de tela dos registros gravados no banco de dados



idmedicao	idconsumidor	data	tensao	corrente	watts
6020	15	2023-08-20 13:14:53	210.325	0.03793	7.97762
6019	15	2023-08-20 13:14:49	210.203	0.03793	7.97301
6018	15	2023-08-20 13:14:46	211.995	0.03793	8.04096
6017	15	2023-08-20 13:14:43	211.241	0.03793	8.01238
6016	15	2023-08-20 13:14:39	210.242	0.03793	7.97448
6015	15	2023-08-20 13:14:35	210.207	0.03793	7.97317
6014	15	2023-08-20 13:14:12	211.104	0.03793	8.00719
6013	15	2023-08-20 13:14:09	209.72	0.03793	7.95469
6012	15	2023-08-20 12:56:54	217.863	3.86998	843.099
6011	15	2023-08-20 12:56:51	217.735	3.82247	832.285
6010	15	2023-08-20 12:56:47	218.194	3.80284	829.758
6009	15	2023-08-20 12:56:44	219.382	0.03793	8.32116
5446	15	2023-08-20 12:10:55	230.8	29.85	6889.38
5445	15	2023-08-20 11:10:55	218.03	1.16	252.915
5444	15	2023-08-20 10:10:55	219.85	16.9	3715.47
5443	15	2023-08-20 09:10:55	230.01	9.8	2254.1
5442	15	2023-08-20 08:10:55	230.64	48.75	11243.7
5441	15	2023-08-20 07:10:55	212.22	27.6	5867.65

Fonte: Autoria própria

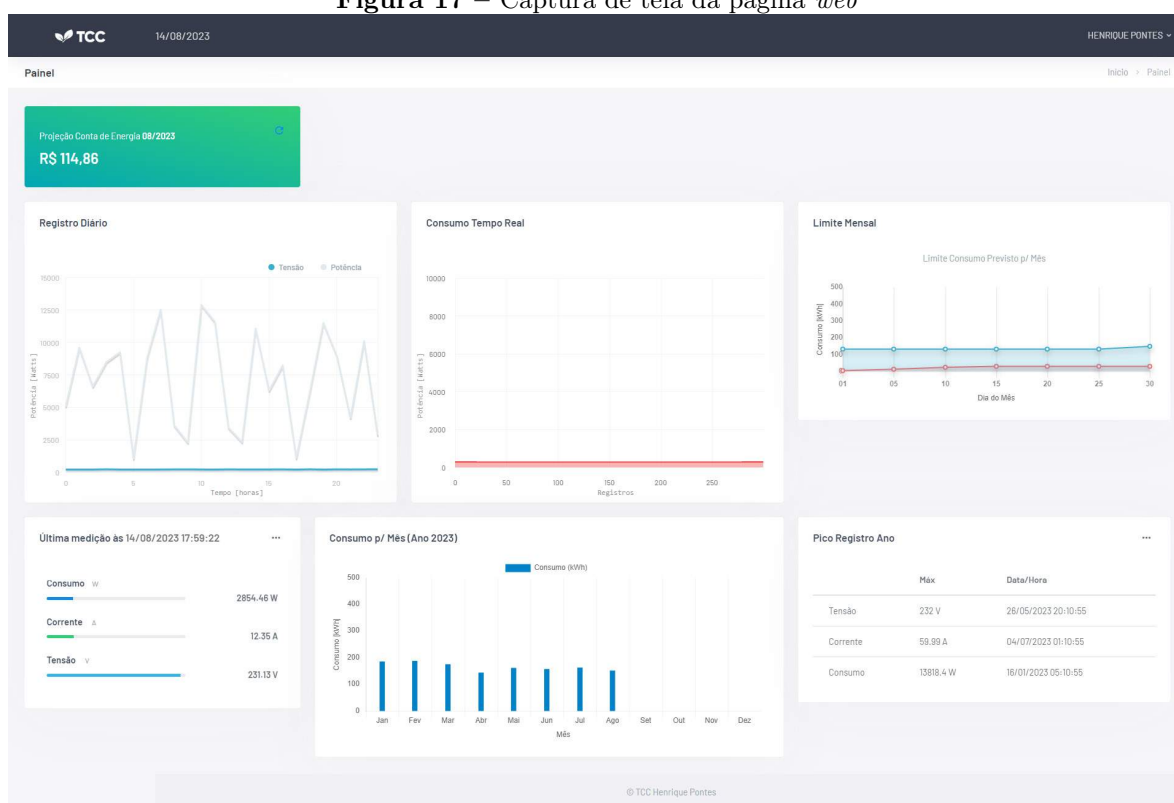
4.2 Resultados demonstrados graficamente

Os resultados do projeto foram apresentados de maneira clara e acessível por meio do site <<https://henriquedepaiva.com.br/tcc/>>. Nessa plataforma, os dados coletados e processados foram traduzidos em gráficos informativos e de fácil compreensão para o usuário.

Ao visitar o site, o usuário poderá explorar diferentes gráficos interativos que representam o consumo elétrico de diversas forma. Os gráficos, não apenas exibem os padrões de consumo ao longo do tempo, mas também oferecem a capacidade de interagir com os dados.

O site demonstra registros diários, mensal, anual e picos registrados ao longo do tempo. Além de uma projeção de quanto irá ser o valor da fatura se a média de consumo for mantida até o fim do mês. A figura 17 exemplifica estes gráficos ou acessando diretamente o link <<https://henriquedepaiva.com.br/tcc/>>.

Figura 17 – Captura de tela da página web



Fonte: Autoria própria

Outra característica do projeto também foi a atenção ao layout para dispositivos móveis. A interface do sistema foi projetada para garantir uma experiência de usuário otimizada em telas de diferentes tamanhos, como smartphones, tablets e computadores. Isso assegura que os usuários possam acessar de forma conveniente em qualquer lugar que esteja e independente do dispositivo que estiver utilizando. Isso é crucial, uma vez que a utilização de dispositivos móveis para acessar sites se tornou maior que computadores na era digital contemporânea. Um exemplo de diagramação da página em um celular encontra na figura 18 a seguir.

Figura 18 – Captura de tela da página *web* para dispositivos móveis

Fonte: Autoria própria

Além disso, o site proporcionou a funcionalidade de visualização em tempo real da leitura de consumo elétrico, possibilitando que usuário acompanhasse qualquer mudança ou variação abrupta no seu consumo e assim identificar tendências, picos e variações no consumo elétrico de maneira imediata e eficaz.

A visualização de gráficos em tempo real sem a necessidade de recarregar a página, se destaca como um elemento extra do projeto, enriquecendo significativamente a experiência do usuário ao proporcionar um acompanhamento dinâmico e imediato do consumo elétrico. A seguir na figura 19 um exemplo deste gráfico mencionado.

Figura 19 – Gráfico em tempo real da leitura da potência elétrica



Fonte: Autoria própria

5 CONCLUSÃO

O projeto em questão foi concluído com êxito, conseguindo atingir o pleno do funcionamento do protótipo ao término do processo e também atendendo as expectativas do mesmo, demonstrando sua capacidade de mensurar e exibir telemetria do consumo elétrico. Esse feito foi possível por meio da aplicação dos conhecimentos adquiridos tanto no Curso de Engenharia Elétrica quanto no Curso de Sistemas para a Internet. Adicionalmente, vale mencionar a contribuição do minicurso de Arduino, realizado durante o primeiro período do curso de Engenharia Elétrica e promovido pelo PET da instituição. Essa colaboração entre diferentes áreas de conhecimento e a integração de ensinamentos sobre novos microcontroladores disponíveis no mercado foram elementos essenciais para a concretização bem-sucedida desse desenvolvimento.

Durante o processo de desenvolvimento do protótipo, foram enfrentados alguns contratemplos. Um dos principais desafios foi a necessidade de trocar o microcontrolador originalmente planejado devido a incompatibilidades com a conexão Wi-fi. Essa mudança implicou em um processo de reconfiguração e reinstalação ao computador, o que demandou um gasto maior de tempo. Em adição de sua aquisição e também entender o seu funcionamento.

Além disso, houve a decisão pessoal de não utilizar a IDE padrão do Arduino para utilizar uma IDE PlataformIO, pois possuía mais facilidades e aprimoramentos para a eficiência do projeto. Essa transição por vezes demandou um investimento de tempo para familiarização com a nova ferramenta e entender onde realizavam alguns procedimentos.

A calibração do instrumento de mensuração também se apresentou como um desafio complexo. A obtenção de leituras precisas e confiáveis exigiu uma série de iterações e testes, e também houve algumas incompatibilidades entre bibliotecas utilizadas. Outro contratempo notável foi a dificuldade em encontrar determinados materiais específicos necessários para a construção do protótipo. A carência de sensores no mercado local retardou o progresso do projeto.

No contexto das perspectivas futuras, acredito que o projeto demonstra um potencial significativo para expansão e refinamento, visando aprimorar ainda mais sua funcionalidade e utilidade. Algumas das pretensões futuras vistas de imediato para agregar ao protótipo seria o desenvolvimento de uma placa de circuito impresso, isto não só otimizará o layout e a integração dos componentes, mas também promoverá uma abordagem mais profissional e robusta, juntamente com uma caixa personalizada para abrigar o protótipo, utilizando tecnologia de impressão 3D.

Outro ponto seria aprimorar a experiência do usuário e oferecer análises mais ri-

cas, a implementação de filtros no site pode proporcionar aos usuários a capacidade de segmentar e analisar os dados de consumo elétrico de maneira mais específica e personalizada.

Para evitar problemas com quedas de conectividade à Internet, o acoplamento de um módulo de cartão de memória ao microcontrolador para armazenar temporariamente os dados pode garantir que as informações não sejam perdidas. Dessa forma, quando a conexão for restaurada, os dados podem ser sincronizados com sucesso.

A exploração dessas pretensões futuras não apenas elevará a qualidade técnica do projeto, mas também demonstrará o comprometimento contínuo em melhorar e adaptar o sistema às necessidades dos usuários, contribuindo de maneira significativa para a eficiência e a conscientização no consumo elétrico.

Essa jornada, marcada por superações constantes, contribuiu significativamente para o desenvolvimento bem-sucedido do protótipo, além disso, essa experiência enriqueceu significativamente o repertório de conhecimentos pessoais. Ela também evidenciou de maneira marcante o potencial excepcional do curso, demonstrando sua capacidade de preparar os estudantes de forma abrangente e competente. Isso ressaltou a importância das vivências práticas em sala de aula. Ao mesmo tempo, reforçou a necessidade do embasamento teórico para aplicar essas experiências de maneira eficaz.

REFERÊNCIAS

- ALUGAGERA. *Lista de tensões e frequências (Hz) ao redor do mundo*. 2020. Disponível em: <<https://alugagera.com.br/noticias/tensao-frequencia-tomada-mundo>>.
- ASHTON, K. That “Internet of Things” Thing. *RFiD Journal*, 22, 97-114, p. 1, 2009.
- BRAGA, I. N. C. *Tensão de Pico Eficaz e RMS*. 2016. Disponível em: <<https://www.newtoncbraga.com.br/index.php/ideias-dicas-e-informacoes-uteis/177-ideias-praticas/12290-tensao-de-pico-eficaz-e-rms-ip1315.html>>.
- DATASHEETSPDF. *Datasheet ZMPT101B*. 2022. Disponível em: <<https://datasheetpdf.com/pdf/1031464/ETC/ZMPT101B/1>>.
- DOYLE, M. *Beginning PHP*. [S.l.]: Wrox, 2011.
- ELECTRONICOSCALDAS. *Datasheet TA12-100*. 2022. Disponível em: <http://www.electronicoscaldas.com/datasheet/TA12-TA12L-Series_YHDC.pdf>.
- ENERGISA. *NDU-001*. 2023. 44 p. Disponível em: <<https://www.energisa.com.br/Documents/Normas%20t%C3%A9cnicas/NDU%20001%20-%20Fornecimento%20de%20energia%20el%C3%A9trica%20a%20edifica%C3%A7%C3%B5es%20individuais%20ou%20agrupadas%20at%C3%A9%203%20unidades%20consumidoras.pdf>>.
- ESP32FORUM. *ESP32 Porta Analógica (ADC)*. 2022. Disponível em: <<https://www.esp32.com/viewtopic.php?t=1045>>.
- GRAPHPAD. *Tabela geradora da equação de reta*. 2022. Disponível em: <<https://www.graphpad.com/quickcalcs/linear1/>>.
- GUSSOW, M. *Eletricidade Básica 2ª Edição*. [S.l.]: Bookman, 2005.
- HENICK, B. *HTML Css: The Good Parts: Better Ways to Build Websites That Work*. [S.l.]: O'Reilly Media, 2010.
- KOFLER, M. *The Definitive Guide to MySQL 5*. [S.l.]: Apress, 2005.
- LASTMINUTEENGINEERS. *ESP32 Pinout*. 2022. Disponível em: <<https://lastminuteengineers.com/esp32-pinout-reference/>>.
- MOZILLA. *JavaScript Guide*. 2023. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>>.
- VISHAY. *Datasheet OLED 128x64*. 2022. Disponível em: <<https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf>>.
- WILLARD, W. *Html: A Beginner'S Guide*. [S.l.]: McGraw-Hill Education, 2009.

A APÊNDICES

A.1 Apêndice A - Código fonte na íntegra do código implementado no microcontrolador.

```

1
2 // BIBLIOTECA ARDUINO
3 #include <Arduino.h>
4
5 // BIBLIOTECA E PORTAS SENSOR TENSAO
6 #include <ZMPT101B.h>
7 ZMPT101B sensorVolt(34);
8
9 // BIBLIOTECA OLED
10 #include <Wire.h>
11 #include <Adafruit_GFX.h>
12 #include <Adafruit_SSD1306.h>
13
14 #define SCREEN_WIDTH 128 // OLED display largura pixels
15 #define SCREEN_HEIGHT 64 // OLED display altura pixels
16 // display conectado ao I2C (SDA, SCL)
17 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
18
19 // BIBLIOTECA WIFI E HTTP
20 #include <WiFi.h>
21 #include <HTTPClient.h>
22
23 //CONFIGURACAO WIFI
24 const char *ssid = "Henrique";
25 const char *password = "12345678";
26 const char *serverName = "https://www.henriquedepaiva.com.br/tcc/
    envia.php";
27 const char *hashcliente = "
    f369cb89fc627e668987007d121ed1eacdc01db9e28f8bb26f358b7d8c4f08ac
    ";
28
29
30 unsigned long lastTime = 0;
31 unsigned long inicializacao = 15;
32 int boot;
33 char httpRequestData[150];

```

```
34
35
36 unsigned long agora = 0;
37 int idcliente = 15;
38 int sensorTA12 = 32;
39 float nCPP;
40 float v;
41 float w;
42 float a;
43 float fp = 0.92;
44
45 void setup()
46 {
47
48     // BAUDRATE SERIAL OUTPUT
49     Serial.begin(9600);
50
51     sensorVolt.calibrateVoltage();
52
53     pinMode(sensorTA12, INPUT);
54
55     if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
56     { // Endere o 0x3D para 128x64
57         Serial.println(F("SSD1306 falhou"));
58     }
59     else
60     {
61         display.clearDisplay();
62     }
63
64     WiFi.begin(ssid, password);
65     Serial.println("Connecting");
66     while (WiFi.status() != WL_CONNECTED)
67     {
68         delay(500);
69         Serial.print(".");
70     }
71     Serial.println("");
72     Serial.print("Connected to WiFi network with IP Address: ");
73     Serial.println(WiFi.localIP());
74 }
75
```

```
76
77 // Impressao OLED
78 float funcTela(float v, float w, float a)
79 {
80     display.clearDisplay();
81     display.setTextSize(1);
82     display.setTextColor(WHITE);
83     display.setCursor(0, 0);
84     display.println("TCC Medidor IOT");
85     display.setCursor(0, 20);
86     display.print("Consumo:");
87     display.print(w, 2);
88     display.println(" W");
89     display.setCursor(0, 30);
90     display.print("Tensao:");
91     display.print(v, 2);
92     display.println(" V");
93     display.setCursor(0, 40);
94     display.print("Corrente:");
95     display.print(a, 2);
96     display.println(" A");
97     display.display();
98     return false;
99 }
100
101
102 //funcao corrente
103 float getCorrente()
104 {
105     float result;
106     int readValue; // value read from the sensor
107     int maxValue = 0; // store max value here
108     uint32_t start_time = millis();
109     while ((millis() - start_time) < 1666) // amostra em 1 Seg
110     {
111         readValue = analogRead(sensorTA12);
112         if (readValue > maxValue)
113         {
114             /*grava maior valor lido no sensor*/
115             maxValue = readValue;
116         }
117     }
```

```
118 // Conversao MEIA ONDA
119 result = maxValue * 0.3184;
120 // Conversao regressao linear
121 result = 0.01233 * result + 0.03793;
122 return result;
123 }
124
125 void loop()
126 {
127
128     nCPP = getCorrente();
129
130     float tensaoMedida = sensorVolt.getVoltageAC(60);
131
132     v = tensaoMedida;
133     a = nCPP;
134     w = v * a * fp;
135
136     funcTela(v, w, a);
137
138     if (WiFi.status() == WL_CONNECTED)
139     { // Checa conexao WiFi
140         HTTPClient http;
141         http.begin(serverName);
142         http.addHeader("Content-Type", "application/x-www-form-
143             urlencoded");
144
145         sprintf(httpRequestData, "volt=%f&=%f&idcliente=%d&
146             hashcliente=%s", tensaoMedida, a, idcliente, hashcliente);
147
148         int httpResponseCode = http.POST(httpRequestData); // Envia o
149             POST request
150         if (httpResponseCode > 0)
151         {
152             // retorna resultado da resposta em http
153             String response = http.getString();
154             Serial.println(httpResponseCode);
155             Serial.println(response);
156         }
157     }
158     else
159     {
160         Serial.print("Erro no envio POST: ");
161     }
162 }
```

```
157     Serial.println(httpResponseCode);
158   }
159   http.end();
160   //delay(1000);
161 }
162 else
163 {
164   Serial.println("Error in WiFi connection");
165 }
166 }
```

Código A.1 – Código implementado no microcontrolador

A.2 Apêndice B - Código fonte do código php onde servidor recebe os dados do microcontrolador e faz a inserção do banco de dados.

```
1
2
3 <?php
4   error_reporting(1);
5
6   date_default_timezone_set('America/Recife');
7   $hora = date('Y-m-d H:i:s');
8   $conexao = mysqli_connect('localhost', 'u559028162_root', '
9       Teste123');
10  $banco = mysqli_select_db($conexao, 'u559028162_tcc');
11  mysqli_set_charset($conexao, 'utf8');
12
13  //armazena as requisicoes vindas do microcontrolador em
14  variaveis
15  $idcliente = $_REQUEST['idcliente'];
16  $voltage = $_REQUEST['volt'];
17  $corrente = $_REQUEST['amp'];
18
19  //checagem e insercao dos dados no banco de dados
20  if($idcliente){
21      $hashvem = $_REQUEST['hashcliente'];
22      $hashnum = (int)$idcliente*5;
23      $hashcalc = hash('sha256', $hashnum);
24  }
```

```
25
26     if($hashvem == $hashcalc){
27
28
29     if ($voltage && $corrente) {
30         $watt = $voltage * $corrente;
31         $query = "INSERT INTO medicao ('idconsumidor','data','
                tensao', 'corrente', 'watts') VALUES ('".$idcliente."'
                ,'" . $hora . "', '" . $voltage . "', '" . $corrente . "
                ', '" . $watt . "') ";
32         $result = mysqli_query($conexao, $query) or die('Errant
                query: ' . $query);
33         if($result){
34             //gravacao do valor da corrente em arquivo texto
                para o consulta do grafico em tempo real
35             echo "Gravou com sucesso <br/>";
36             $myfile = fopen("correnteagora.txt", "w");
37             fwrite($myfile, $watt);
38             fclose($fp);
39             echo "Txt OK";
40         }
41     }
42 }
43 }
44
45 }
46
47 ?>
```

Código A.2 – Código implementado no microcontrolador

A.3 Apêndice C - Código fonte das consultas MySQL dos dados armazenados no banco de dados.

```
1 <?php
2
3     date_default_timezone_set('America/Recife');
4     $conexao = mysqli_connect('localhost', 'u559028162_root', '
5         Teste123');
6     $banco = mysqli_select_db($conexao, 'u559028162_tcc');
7     mysqli_set_charset($conexao, 'utf8');
8     $sql = mysqli_query($conexao, "select * from medicao") or die("
9         Erro");
10
11     //ULTIMO REGISTRO
12
13     $sqlultmio = mysqli_query($conexao, "SELECT * FROM medicao
14         ORDER by data desc LIMIT 1") or die("Erro");
15     while ($dadoagora = mysqli_fetch_assoc($sqlultmio)) {
16         $agora['data'] = $dadoagora['data'];
17         $agora['tensao'] = $dadoagora['tensao'];
18         $agora['corrente'] = $dadoagora['corrente'];
19         $agora['watts'] = $dadoagora['watts'];
20     }
21
22     //VALORES MAXIMOS ANO
23
24     $sqlmaxV = mysqli_query($conexao, "SELECT m.* FROM medicao m
25         WHERE m.tensao = (select max(subp.tensao) from medicao subp
26         where YEAR(subp.data) = '".date('Y')."');") or die("Erro");
27     while ($dadomax = mysqli_fetch_assoc($sqlmaxV)) {
28         $maxV['data'] = $dadomax['data'];
29         $maxV['tensao'] = $dadomax['tensao'];
30     }
31     $sqlmaxA = mysqli_query($conexao, "SELECT m.* FROM medicao m
32         WHERE m.corrente = (select max(subp.corrente) from medicao
33         subp where YEAR(subp.data) = '".date('Y')."');") or die("
34         Erro");
35     while ($dadomax = mysqli_fetch_assoc($sqlmaxA)) {
36         $maxA['data'] = $dadomax['data'];
37         $maxA['corrente'] = $dadomax['corrente'];
38     }
```


```
32     }
33     $sqlmaxW = mysqli_query($conexao, "SELECT m.* FROM medicao m
      WHERE m.watts = (select max(subp.watts) from medicao subp
      where YEAR(subp.data) = '".date('Y')."' );" ) or die("Erro");
34     while ($dadomax = mysqli_fetch_assoc($sqlmaxW)) {
35         $maxW['data'] = $dadomax['data'];
36         $maxW['watts'] = $dadomax['watts'];
37     }
38
39
40
41     //CONSUMO DI RIO    MEDIA POR HORA
42     $mediahoraV = array();
43     $mediahoraW = array();
44     for($i=0;$i<=24;$i++){
45         $mediahoraV[$i] = 0;
46         $mediahoraW[$i] = 0;
47     }
48
49     $sqlHora = mysqli_query($conexao, "SELECT idmedicao, AVG(
      tensao) as tensao, AVG(watts) as watts, HOUR(data) as
      hora FROM medicao WHERE DATE_SUB('data',INTERVAL 1 HOUR)
      And DATE(data) = CURDATE() GROUP BY HOUR(data) ORDER BY
      idmedicao ASC;" ) or die("Erro");
50     while ($dadohora = mysqli_fetch_assoc($sqlHora)) {
51         $hora = $dadohora['hora'];
52         $mediahoraV[$hora] = $dadohora['tensao'];
53         $mediahoraW[$hora] = $dadohora['watts'];
54     }
55 }
56
57
58
59     // CONSUMO MENSAL MEDIA POR MES
60
61     $mediaMesV = array();
62     $mediaMesW = array();
63     for($i=0;$i<=12;$i++){
64         $mediaMesV[$i] = 0;
65         $mediaMesW[$i] = 0;
66     }
67     $sqlMes = mysqli_query($conexao, "SELECT idmedicao, AVG(
```

```
        tensao) as tensao, AVG(watts) as watts, MONTH(data) as
        mes FROM medicao WHERE DATE_SUB('data',INTERVAL 1 MONTH)
        And YEAR(data) = YEAR(CURDATE()) GROUP BY MONTH(data)
        ORDER BY idmedicao ASC;") or die("Erro");
68 while ($dadoMES = mysqli_fetch_assoc($sqlMes)) {
69     $mes = $dadoMES['mes'];
70     $mediaMesV[$mes-1] = $dadoMES['tensao'];
71     $mediaMesW[$mes-1] = $dadoMES['watts'];
72
73
74 }
75
76
77 // CONSUMO MENSAL MEDIA POR DIA
78
79 $mediaDiaV = array();
80 $mediaDiaW = array();
81 for($i=0;$i<=30;$i++){
82     $mediaDiaV[$i] = 0;
83     $mediaDiaW[$i] = 0;
84 }
85 $sqlDia = mysqli_query($conexao, "SELECT idmedicao, AVG(
        tensao) as tensao, AVG(watts) as watts, DAY(data) as dia
        FROM medicao WHERE DATE_SUB('data',INTERVAL 1 DAY) And
        MONTH(data) = MONTH(CURDATE()) GROUP BY DAY(data) ORDER
        BY idmedicao ASC;") or die("Erro");
86 while ($dadoDIA = mysqli_fetch_assoc($sqlDia)) {
87     $dia = $dadoDIA['dia'];
88     $mediaDiaV[$dia] = $dadoDIA['tensao'];
89     $mediaDiaW[$dia] = $dadoDIA['watts'];
90
91 }
92
93
94 //var_dump($mediaDiaW);
95
96
97
98 ?>
```

Código A.3 – Código implementado no microcontrolador

A.4 Apêndice D - Github do projeto com os arquivos desenvolvidos

<https://github.com/henr1quep/tcc/>

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus João Pessoa
	Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, Joao Pessoa (PB)
	CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Restrito

Trabalho de Conclusão de Curso

Assunto:	Trabalho de Conclusão de Curso
Assinado por:	Henrique Pontes
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo da Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Henrique de Paiva Araújo Pontes, ALUNO (20152610140) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA, em 08/01/2024 12:25:01.

Este documento foi armazenado no SUAP em 08/01/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1047507

Código de Autenticação: 8a9b0a126d

