

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**SAYMON RAFAEL DA CUNHA SILVA**

**SISTEMA DE MONITORAMENTO E CONTROLE DE  
ENERGIA UTILIZANDO O MICROCONTROLADOR ESP32**

**PONTA GROSSA**

**2021**

**SAYMON RAFAEL DA CUNHA SILVA** ✉

**SISTEMA DE MONITORAMENTO E CONTROLE DE  
ENERGIA UTILIZANDO O MICROCONTROLADOR ESP32**

**ENERGY MONITORING AND CONTROL SYSTEM  
USING THE ESP32 MICROCONTROLLER**

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Josmar Ivanqui ✉

**PONTA GROSSA**

**2021**



Este Trabalho de Conclusão de Curso está licenciado sob uma Licença Creative Commons Atribuição–NãoComercial–SemDerivações 4.0 Internacional.

**SAYMON RAFAEL DA CUNHA SILVA**

**SISTEMA DE MONITORAMENTO E CONTROLE DE ENERGIA UTILIZANDO O  
MICROCONTROLADOR ESP32**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Elétrica da Universidade  
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 03/dezembro/2021

---

Josmar Ivanqui  
Doutorado  
Universidade Tecnológica Federal do Paraná (UTFPR)

---

Cristhiane Gonçalves  
Doutorado  
Universidade Tecnológica Federal do Paraná (UTFPR)

---

Virgínia Helena Varotto Baroncini  
Doutorado  
Universidade Tecnológica Federal do Paraná (UTFPR)

**PONTA GROSSA  
2021**

Dedico este trabalho primeiramente a mim, pela tenacidade. À minha família e aos meus amigos, pelos momentos de ausência. Por fim, ao meu orientador, pelo incentivo durante todo o projeto.

## **AGRADECIMENTOS**

O presente trabalho não poderia ser finalizado sem a colaboração de diversas pessoas e instituições às quais presto meus agradecimentos. Certamente, esses parágrafos não irão abranger a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre estas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

A minha família, pelo carinho, incentivo e total apoio em todos os momentos da minha vida.

Ao meu orientador, que me mostrou os caminhos a serem seguidos e pela confiança depositada.

A todos os professores e colegas do departamento, que ajudaram de forma direta e indireta na realização e conclusão deste trabalho.

A todos os demais que de alguma forma contribuíram para meu crescimento pessoal e profissional.

“Uma máquina pode fazer o trabalho de cinquenta pessoas comuns. Nenhuma máquina pode fazer o trabalho de uma pessoa extraordinária.” (HUBBARD, 1998, tradução).<sup>1</sup>

---

<sup>1</sup> “*A machine can do the work of fifty ordinary people. No machine can do the work of an extraordinary person.*” (HUBBARD, 1998).

## RESUMO

O avanço tecnológico trouxe muitas facilidades ao dia a dia dos seres humanos, atualmente é possível supervisionar e controlar entradas e saídas de energia de qualquer lugar do mundo. Esse tipo de recurso é largamente utilizado no âmbito industrial, com tudo, apresenta muitas limitações no meio residencial como alto custo, tecnologias limitadas e escassez no mercado. Os microcontroladores, associados à determinados equipamentos eletrônicos, são uma das chaves para resolver muitos desses obstáculos, com estes componentes, que nada mais são que pequenos computadores programáveis, é possível implementar um sistema de automação residencial, responsável pelo controle, monitoramento e armazenamento de dados de consumo de energia das saídas elétricas de uma casa ou qualquer outro tipo de unidade consumidora de baixa tensão. Esses sistemas concentram-se no uso consciente de energia, conceito chave da eficiência energética. A automação residencial ainda é uma das respostas mais viáveis aos problemas de acessibilidade das mais diversas pessoas e ambientes.

**Palavras-chave:** Automação; Residencial; Monitoramento; Controle; Energia.

## **ABSTRACT**

Technological advances have brought many facilities to the daily lives of human beings, nowadays it is possible to supervise and control inputs and produce energy from anywhere in the world. This type of resource is widely used in the industrial scope, however, it has many limitations in the residential environment, such as high cost, limited technologies and scarcity in the market. Microcontrollers, associated with certain electronic equipment, are one of the keys to solving many problems, with these components, which are nothing more than small programmable computers, it is possible to implement a home automation system, responsible for the control, monitoring and storage of data. consumption of energy from the electrical outlets of a house or any other type of low-voltage consumer unit. These systems focus on the conscious use of energy, an efficient keying concept. Home automation is still one of the most viable answers to the accessibility problems of the most diverse people and environments.

**Keywords:** Automation; Residential; Monitoring; Control; Energy.



## LISTA DE ILUSTRAÇÕES

Figura 1 – ESP32 Pinagem . . . . .	20
Figura 2 – Diagrama de blocos do ESP32 . . . . .	21
Figura 3 – Interface de desenvolvimento do Arduino . . . . .	25
Figura 4 – Sensor de corrente não invasivo HMCT103C . . . . .	28
Figura 5 – Circuito Integrado 74HC595 . . . . .	29
Figura 6 – Módulo rele 8 unidades . . . . .	30
Figura 7 – Comportamento de um Web Server . . . . .	32
Figura 8 – Composição de Preço da fatura de Energia . . . . .	33
Figura 9 – E-mail de notificação enviado pelo ESP32 . . . . .	36
Figura 10 – Fluxograma de Processo da Programação . . . . .	38
Figura 11 – Circuito Eletrônico do Projeto . . . . .	39
Figura 12 – Declaração das variáveis do sensor de corrente e do 74HC595 . . . . .	40
Figura 13 – Circuito em cascata 74HC595 . . . . .	41
Figura 14 – Circuito elétrico saída dos reles . . . . .	41
Figura 15 – Página Web . . . . .	43
Figura 16 – Conteúdo aba início . . . . .	44
Figura 17 – Conteúdo aba consumo . . . . .	45
Figura 18 – Conteúdo aba configurações . . . . .	46
Figura 19 – Conteúdo aba desenvolvedor . . . . .	47
Figura 20 – Solicitação de <i>login</i> . . . . .	47
Figura 21 – Código Base64 no algoritmo . . . . .	48
Figura 22 – Cancelamento de <i>login</i> ou caso <i>logout</i> . . . . .	48
Figura 23 – Como encontrar o editor de Script Google Sheets . . . . .	49
Figura 24 – Bancada de testes do projeto . . . . .	54
Quadro 1 – Dados técnicos ESP32 . . . . .	21
Quadro 2 – Exemplo prático de codificação Base64 . . . . .	31
Quadro 3 – Comparação com outras tecnologias . . . . .	50
Quadro 4 – Comparação de custos . . . . .	52
Quadro 5 – Componentes utilizados . . . . .	53

## LISTA DE TABELAS

Tabela 1 – Comparação de Microcontroladores . . . . .	34
Tabela 2 – Dados armazenados no Google Sheets - Lâmpada 12W . . . . .	49
Tabela 3 – Composição do custo médio do projeto . . . . .	51
Tabela 4 – Dados Aferidos 1 . . . . .	55
Tabela 5 – Dados Aferidos 2 . . . . .	55

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

### SIGLAS

API	<i>Application Programming Interfaces</i>
CI	Circuito Integrado
CLP	Controlador Lógico Programável
HTML	Linguagem de Marcação de Hipertexto, do inglês <i>HyperText Markup Language</i>
HTTP	Protocolo de transferência de Hipertexto, do inglês <i>Hypertext Transfer Protocol</i>
IA	Inteligência Artificial
IDC	<i>International Data Corporation</i>
IDE	Ambiente de desenvolvimento integrado, do inglês <i>Integrated Development Environment</i>
IHM	Interface Homem Máquina
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
IP	Protocolo de comunicação da internet, do inglês <i>Internet Protocol</i>
PCI	Placas de Circuito Impresso
PNE	Portadoras de necessidades especiais
RTC	Relógio de Tempo Real, do inglês <i>Real Time Clock</i>
SDK	<i>Software Development Kit</i>
TIR	Taxa interna de retorno
TMA	Taxa mínima de atratividade
URL	Localizador Uniforme de Recursos, do inglês <i>Uniform Resource Locator</i>
UTFPR	Universidade Tecnológica Federal do Paraná
VPL	Valor presente líquido
W3C	<i>World Wide Web Consortium</i>
WHATWG	<i>Web Hypertext Application Technology Working Group</i>

### ACRÔNIMOS

ANEEL	Agência Nacional de Energia Elétrica
CSS	Folha de estilos em cascata, do inglês <i>Cascading Style Sheets</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	ESTADO DA ARTE	13
1.2	O PROBLEMA	14
1.3	PREMISSA	14
1.4	OBJETIVOS	14
1.4.1	Objetivo Geral	14
1.4.2	Objetivos Específicos	15
1.5	JUSTIFICATIVA	15
1.6	METODOLOGIA	15
1.7	DESCRIÇÃO DO DESENVOLVIMENTO	16
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>17</b>
2.1	DOMÓTICA	17
2.2	MICROCONTROLADORES	18
2.2.1	ESP32	19
2.3	LINGUAGENS DE PROGRAMAÇÃO	21
2.3.1	Linguagem C++	22
2.3.2	Linguagem HTML	23
2.3.2.1	<i>Cascading Style Sheets</i>	23
2.4	AMBIENTES DE DESENVOLVIMENTO INTEGRADO	24
2.4.1	A interface do Arduino IDE	25
2.4.2	Bibliotecas	26
2.4.2.1	<i>Github</i>	27
2.5	EQUIPAMENTOS AUXILIARES	27
2.5.1	Sensor de Corrente não Invasivo HMCT103C	27
2.5.2	Circuito Integrado 74HC595	28
2.5.3	Módulo rele	29
2.6	CONVERSÃO DE BASE	30
2.7	CIBERSEGURANÇA	31
2.8	<i>WEB SERVER</i>	31
2.9	CONSUMO DE ENERGIA E TARIFAÇÃO GRUPO B - RESIDENCIAL	32
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>34</b>
3.1	DESIGNAÇÃO DO MICROCONTROLADOR	34
3.2	FLUXOGRAMA DO PROJETO	35
3.3	CIRCUITO ELÉTRICO DO PROJETO	37
3.4	<i>WEB SERVER</i> DO PROJETO	42
3.4.1	Aba Início	43
3.4.2	Aba Consumo	44
3.4.3	Aba Configurações	45
3.4.4	Aba Desenvolvedor	46
3.4.5	HTTP Authentication	46
3.5	BANCO DE DADOS PLANILHAS GOOGLE	48
3.6	COMPARAÇÃO COM AS TECNOLOGIAS EXISTENTES NO MERCADO	50
3.6.1	Comparação de custos	51
3.7	<i>PAYBACK</i>	52

<b>4</b>	<b>RECURSOS E RESULTADOS</b> . . . . .	<b>53</b>
4.1	RECURSOS . . . . .	53
4.2	RESULTADOS . . . . .	54
<b>5</b>	<b>CONCLUSÕES</b> . . . . .	<b>57</b>
5.1	TRABALHOS FUTUROS . . . . .	58
	<b>REFERÊNCIAS</b> . . . . .	<b>59</b>

## 1 INTRODUÇÃO

Controle e monitoramento de variáveis são procedimentos rotineiros dentro dos diversos campos industriais. A análise de dados obtidos por sensoriamento ou por aquisição manual é uma ferramenta que possibilita a prevenção de futuras falhas e o controle praticamente instantâneo de uma ponto de energia.

Com a crescente demanda por inovações tecnológicas a indústria está sempre adaptando seus equipamentos, hoje alguns inversores de frequência com a Interface Homem Máquina (IHM) da Schneider<sup>®</sup>, por exemplo, permitem a monitoração e em alguns casos, até o controle do consumo de energia do próprio equipamento e de outros conectados a ele.

Guardada as devidas proporções, ao mesmo tempo, as tecnologias do âmbito residencial também evoluem continuamente, Amazon Alexa, *Smart Control*, e algumas tecnologias independentes, são exemplos de equipamentos que permitem um determinado nível de automação residencial.

Os principais infortúnios das destas tecnologias de automação residencial, também conhecidos como domótica, são o elevado custo, a complexidade de compatibilidade e de futuras expansões. Além disso, nenhuma delas permite a monitoração da energia consumida como acontece de forma análoga nos equipamentos industriais.

É imprescindível que na *Era Atual*, conhecida como *Digital* (ou da *Informação*), caracterizada principalmente pela acessibilidade e conectividade, não exista um equipamento de baixo custo e versátil no mercado capaz de realizar controle e monitoramento de consumo de energia elétrica residencial com acompanhamento por *smartphones*.

### 1.1 ESTADO DA ARTE

O forte investimento em aperfeiçoamento de Inteligência Artificial (IA) provoca o surgimento de tecnologias que podem ser consideradas acessíveis, como por exemplo, a Alexa da Amazon, e uma série de microcontroladores, tais como: ESP32, Arduino, Raspberry Pi Pico, além de sensores mais potentes e versáteis, entre outros equipamentos disponíveis.

Apontada como uma das principais tecnologias atuais a IA poderá gerar investimentos de R\$ 2,4 bilhões de reais para o Brasil. Essa é a estimativa do *International*

*Data Corporation* (IDC), consultoria especializada em inteligência de mercado, conforme entrevista publicada na revista eletrônica Olhar Digital por (SHIMABUKURO, 2021) a demanda atual está cada vez mais propensa e receptível quando se diz respeito a sistemas automatizados, e o que antes era limitado ao uso industrial, agora com uma conexão à *internet* e dispositivo de integração às redes é facilmente adaptado para as residências.

## 1.2 O PROBLEMA

A ausência de uma ferramenta acessível que forneça informações instantâneas de consumo de energia em uma unidade consumidora, pode ser sim considerado um dos fatores da má gestão de energia, afinal, como se pode controlar o que não se conhece?

## 1.3 PREMISSA

Através de um ambiente de desenvolvimento integrado, e o emprego de uma série de dispositivos auxiliares como módulos relés, sensores e fontes de alimentação, programar um microcontrolador ESP32 com linguagem C++ e HTML/CSS, de modo que o algoritmo resultante apto à tornar este microcontrolador uma ferramenta que possa ser implantada no mercado e atenda os requisitos de controle e monitoramento de energia.

## 1.4 OBJETIVOS

### 1.4.1 Objetivo Geral

Desenvolver um sistema, que monitore e controle os pontos de energia de uma unidade consumidora sem a necessidade de trocar equipamentos instalados por outros. Que seja de fácil instalação. Esteja apto a expansões e adaptações. Que tenha baixo custo e acessível a qualquer público.

#### 1.4.2 Objetivos Específicos

- Estudo e comparação das tecnologias existentes no mercado;
- Desenvolvimento de um algoritmo de controle e monitoramento;
- Especificação de linguagens de programação e metodologias;
- Desenvolvimento da programação de um microcontrolador;
- Montagem e análise de um protótipo; e o
- Comparativo entre o dispositivo desenvolvido e o oferecido pelo mercado.

#### 1.5 JUSTIFICATIVA

A correta gestão de energia, também denominado de "Eficiência Energética", é um tema corriqueiro nas indústrias e entre profissionais que atuam na área das Engenharias. Saber como utilizar a energia de maneira correta, evita uma série de problemas ambientais e econômicos.

Uma tecnologia que permita o controle, a monitoração e sendo de fácil instalação, possibilitaria o usuário identificar os problemas pontuais e desta maneira resolver e/ou minimizar os seus prejuízos, tal produto ainda poderia ser considerado uma nova referência aos padrões de acessibilidade físicas, para os portadores de necessidades especiais, e digitais da sociedade atual.

#### 1.6 METODOLOGIA

Utilizando-se de uma estratégia exploratória, o presente trabalho defende o emprego de sistemas que possibilitem a automatização residencial (domótica).

Posteriormente um estudo comparativo entre microcontroladores será levantado, no mesmo será explicado os motivos pelo quais selecionado o microcontrolador para esta aplicação.

Após a realização dessas etapas, é importante realizar uma pesquisa sobre os componentes eletrônicos que podem ser associados ao microcontrolador, como módulos relés, circuitos integrados, sensores, e afins. Essa pesquisa abrange informações tais como: o funcionamento, especificações da ficha de dados, adaptabilidade, entre outras.

Em seguida será tratado o desenvolvimento do algoritmo contido no microcon-



trolador que o permitirá suprir e resolver os problemas e objetivos propostos.

Com o circuito executado, ou seja, microcontrolador programado e associados aos devidos componentes eletrônicos, é de grande valia a realização de um teste de bancada, este que testar a funcionalidade do sistema.

Quando validado o teste de bancada, e observados os resultados "esperados *versus* obtidos", é inevitável a curiosidade da comparação com os produtos já existentes no mercado *versus* o aqui construído. Essa etapa irá apresentar se o tema em questão conseguiu alcançar os objetivos desejados e desta forma, solucionando o problema proposto.

## 1.7 DESCRIÇÃO DO DESENVOLVIMENTO

No Capítulo 2 será realizada revisão da literatura, a qual tem por essência apresentar uma breve introdução e esclarecer conceitos e definições que serão citados ao longo da elaboração desse projeto.

No Capítulo 3 serão apresentados os estudos realizados para a escolha do ESP32 como microcontrolador desse projeto, além de explicações e ilustrações sobre o algoritmo desenvolvido e as ferramentas nele envolvidas. Esse capítulo também apresenta ilustrações sobre o *web server* desenvolvido e aborda o tempo de retorno do investimento do projeto.

Os recursos, humanos e físicos, e os resultados obtidos no projeto serão expostos no Capítulo 4, com explicações mais detalhadas, quadro de componentes e tabelas de dados aferidos.

Por fim no Capítulo 5 será apresentada a conclusão da monografia analisando e comparando os resultados obtidos, além de sugestões de trabalhos futuros utilizados nesse tema.

## 2 REVISÃO DA LITERATURA

### 2.1 DOMÓTICA

A automação industrial teve maior visibilidade e destaque entre as décadas de 50 e 60, com a criação da General Motors®, o Controlador Lógico Programável (CLP), desde então diversas tecnologias foram criadas e adaptadas para se comunicarem com o mesmo.

Esses controladores são especializados em controle e monitoramento de máquinas e equipamentos industriais, e são um grande marco na área de tecnologia. Apresentam diversas vantagens de processos, desde economia de energia e tempo a procedimentos de manutenção.

Com tudo, a automação não se limita apenas a área industrial, o termo domótica provém da união da palavra romana *domus* (casa) e da palavra robótica. Automação residencial e casas inteligentes são sinônimos mais conhecidos de domótica.

A automação residencial assim como a industrial, também busca automatizar os processos, porém domésticos. Esses processos domésticos na maioria das vezes apresentam características centralizadas (remotas) ou apenas permitem que algo se controle de maneira autônoma.

Mas diferente do âmbito industrial, a domótica não tem como foco o desempenho de processos, além disso ela também pode proporcionar uma maior comodidade, conforto, economia de energia e segurança. Ferramentas como sensores, atuadores e dispositivos com algum meio de comunicação embutido são essenciais para incorporar esse tipo de tecnologia conforme citado em (JUNIOR, 2018).

Apesar de existirem vários meios de comunicação entre os dispositivos, como por exemplo *bluetooth* e infra-vermelho, o *Wi-Fi* e outros sinais que permitem a conexão à *internet* são fundamentais para um conceito que está em paralelo com a automação, a Internet das Coisas, do inglês *Internet of Things* (IoT).

A IoT está correlacionado com a conexão de equipamentos à *internet*, e que por consequência acaba facilitando o cotidiano das pessoas, pois, é possível por meio de um celular ou uma máquina com acesso à *internet*, controlar e monitorar o desempenho de um ou mais equipamentos específicos, conforme encontra-se em (MORAIS *et al.*, 2018).

Esse conceito é tão importante atualmente que fabricantes de equipamentos industriais como Schneider<sup>®</sup>, Siemens<sup>®</sup> e até a nacional WEG<sup>®</sup>, já incorporam em CLP, inversores de frequência e outros equipamentos de controle a compatibilidade com a *internet*. Alguns ainda possuem seu próprio *web server* para supervisão.

Contudo, é muito improvável utilizar esses equipamentos no âmbito residencial, é inviável e insensato utilizar um inversor de frequência para controlar o motor de um portão elétrico, em virtude do elevado custo do equipamento mesmo que seja o mais simples disponível comercialmente. Outro detalhe importante sobre esses equipamentos industriais, são que eles são de uso dedicado (também chamados de sistemas embarcados), resultando em aplicações com fins específicos dos mesmos.

Com isso algumas desenvolvedoras criaram alguns dispositivos do tipo *Open Source* (Código aberto) de desenvolvimento, ou seja, que é possível modificar, especificar e até criar a maneira a qual ele irá atuar, o melhor exemplo a citar desses dispositivos são os microcontroladores, como o Arduino<sup>®</sup>, o ESP32, o PIC, entre outros.

## 2.2 MICROCONTROLADORES

A máquina de tear deu origem ao primeiro sistema automatizado, que utilizavam o método de cartões perfurado, esse método foi utilizado por anos e se assemelha em um aspecto dos sistemas modernos, que é o uso binário de dados. Esse método, apesar de útil para época, apresentava uma série de problemas, como por exemplo a programação, que dificilmente poderia ser alterada, como descrito por Lenz.

Com a evolução da eletrônica digital, foi possível implementar dispositivos com capacidade de memorização (na época as chamadas válvulas termiônicas), e mais além até os transistorizados (circuitos integrados).

Lenz afirma que as grandes vantagens dessa tecnologia era a diminuição considerável de espaço, consumo, tempo de processamento e a reprogramação do sistema, o que implicava cada vez mais o avanço tecnológico.

Lenz também expõe, que foi só em 1969, a Intel expôs o primeiro microprocessador comercializável de 4 bits. Apesar de semelhantes em alguns aspectos, o microcontrolador tem menor poder de processamento, podendo ser até chamado de "computador simples", mas a grande vantagem é que diferente do microprocessador, ele não necessita do auxílio de outros componentes para armazenar dados. Os microcontro-

ladores ganharam o mercado em meados de 1980. A estrutura de um microcontrolador comumente contém:

- Memória volátil para manipular e armazenar dados em tempo de processamento;
- Memória não volátil para receber as instruções do programador e permitir que o microcontrolador salve o algoritmo;
- Gerador de *clock* interno (trem de pulso de onda quadrada); e
- Uma série de periféricos que facilitam a integração do microcontrolador com outros componentes, tais como: geração de ondas por meio da modulação por largura de pulso, conversor analógico digital, temporizadores, comparadores, entre outros.

Ao longo do tempo mais empresas entraram no ramo de fabricação de microcontroladores, como a Microchip® com os famigerados PICs, com isso uma série de modelos de microcontroladores foram criadas, e conseqüentemente os controladores foram divididos em famílias. Cada uma com sua peculiaridade: memória, desempenho, velocidade, versatilidade, métodos e linguagens de programação diferentes.

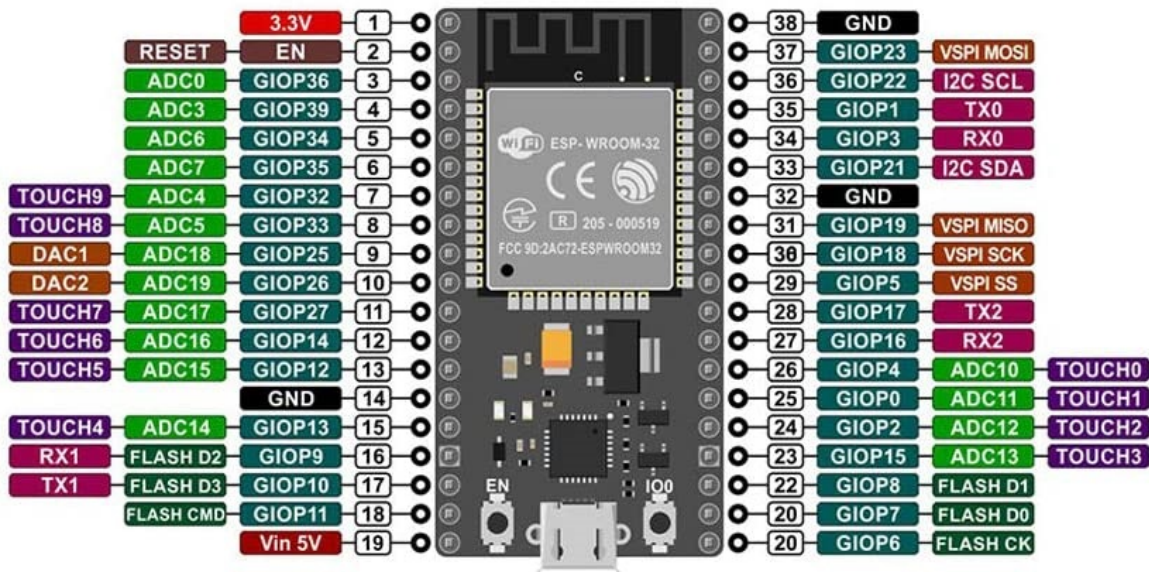
### 2.2.1 ESP32

O microcontrolador ESP32 projetado pela *Espressif Systems*® é lançado em 2016, e são notório pelas suas características técnicas como velocidade de processamento, acessibilidade e conectividade vem ganhando espaço entre os programadores, conforme (SANTOS; LARA JUNIOR, 2019).

O ESP32 é constituído por um robusto processador, que pode ser *single* ou *dual-core* de 32 bit (com dois núcleos físicos de processamento), além de ser superior em várias categorias quando comparado à outros microcontroladores. Possui também um limitador de tensão de 5V para proteção casos a alimentação de energia seja excedente, e também há um circuito de Relógio de Tempo Real, do inglês *Real Time Clock* (RTC) para manter a data e a hora atualizados em caso de quedas de energia, entre outros periféricos. Uma imagem detalhada de sua pinagem pode ser observada na Figura 1, além de algumas de suas funcionalidades em relação ao pinos.

Fisicamente o ESP32 não possui uma memória *EEPROM*, contudo, sua memória *Flash* interna de 16 MB pode disponibilizar até 4 MB para esse fim. Alguns desenvol-

Figura 1 – ESP32 Pinagem



Fonte: Albuquerque, Yure .

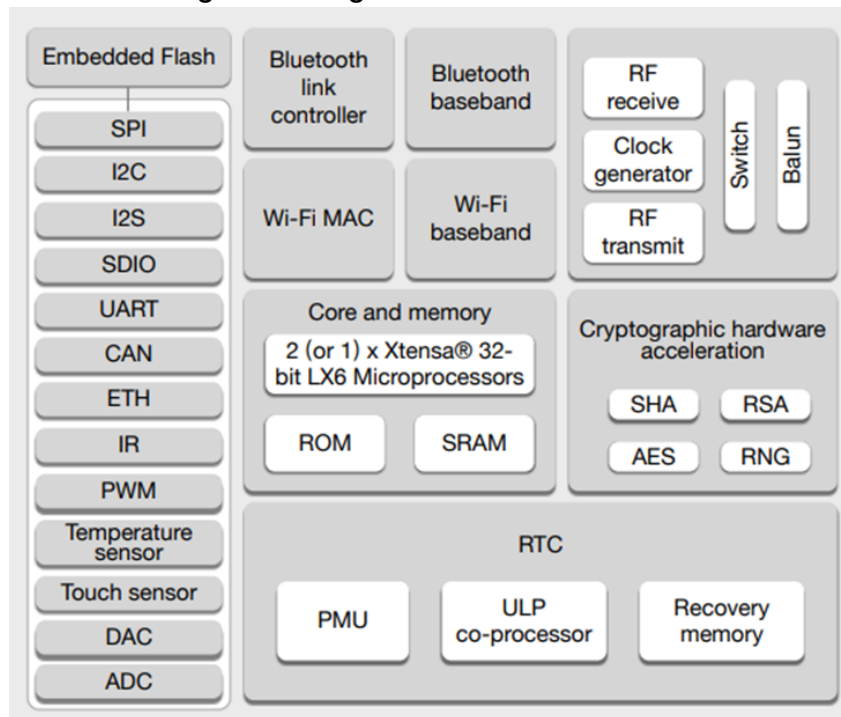
vedores independentes já disponibilizaram bibliotecas específicas "EEPROM.h" para aplicações que demandem armazenamentos de dados, elas podem ser encontradas no portal *github*.

Quanto à conectividade do ESP32, (SANTOS; LARA JUNIOR, 2019) relatam que esse microcontrolador apresenta dois módulos ímpares de integração, incorporados ao seu *chip*, com acesso a redes de transmissão sem fio, através de ondas de rádio, representadas pelo protocolo *bluetooth* e o *Wi-Fi*. Os periféricos presentes em uma placa ESP32 podem ser melhor visualizados em um diagrama de blocos, ilustrado na Figura 2, geralmente encontrado em uma ficha de dados, do inglês: *datasheet*.

Com suas dimensões de pequeno porte, pode ser facilmente incorporado em placas de circuito impressos e outros circuitos eletrônicos. A programação do ESP32 pode advir-se pela linguagem C/C++, e além da programação em seu ambiente próprio, o *Software Development Kit* (SDK), esse microcontrolador também pode ser programado via interface Arduino IDE.

Deve-se atentar com a programação via Arduino IDE, pois, apesar de não apresentar complicações nenhuma ao ESP32, é necessário realizar o *download* dos *drivers* e de bibliotecas internas da IDE referente aos ESPs. Algumas outras especificações técnicas importantes desse microcontrolador podem ser observadas no Quadro 1.

Figura 2 – Diagrama de blocos do ESP32



Fonte: ESPRESSIF SYSTEMS.

Quadro 1 – Dados técnicos ESP32

Característica	Especificação
Dimensões	27,5 x 51,0 x 7,0 mm
Tensão de Alimentação	4,5 à 12 V <sub>DC</sub>
Tensão de Nível Lógico	3,3 V <sub>DC</sub>
Corrente de Consumo	80 à 500 mA
Temperatura de Operação	-40 à 85 °C

Fonte: autoria própria (2021).

### 2.3 LINGUAGENS DE PROGRAMAÇÃO

As linguagens de programação são amplamente conhecidas e utilizadas por profissionais e amadores da área de computação. Para que as linguagens de programação possam ser utilizadas, elas precisam ter sido projetadas e implementadas, cientistas precisaram descobrir do que os programadores necessitam e o que eles esperam de uma determinada linguagem, conforme Melo.

Ao mesmo tempo, um programa, também denominado de algoritmo, é uma ferramenta abstrata que manipula códigos, ele necessita ser implementado em um mecanismo físico (computadores, CLPs, microcontroladores, etc.) para que este código possa ser executado. A construção de algoritmos é influenciada pela linguagem de

programação, logo:

Uma linguagem de programação é o conjunto de recursos que podem ser compostos para construir programas específicos, mais um conjunto de regras de composição que garantem que todos os programas podem ser implementados em computadores com qualidade apropriada. (MELO, 2003).

Uma linguagem eficiente para dada situação é aquela que consegue sanar todos os requisitos do problema. Que é expressiva, que possui paradigmas (um meio de qualificação da linguagem), que pode ser implementada e além de tudo isso, é eficaz para aquela situação. Alguns exemplos de linguagem de programação são: JavaScript, Python, Ladder, PHP, C (e suas derivações). Doravante neste trabalho, daremos foco para aquelas linguagens que mais se adequam a nossa solução.

### 2.3.1 Linguagem C++

Manzano, discorre que a linguagem C, foi desenvolvida por Dennis M. Richie com intuito de desenvolver um sistema operacional em 1972. Aplicada principalmente para escrever programas que utilizassem recursos internos de máquina, de forma muito mais simples que a sua concomitante, a linguagem *Assembly*.

Ainda, segundo Manzano, apesar de se assemelharem no nome, a linguagem C++ (desenvolvida em 1979 por Bjarne Stroustrup e constantemente revisada) e a C, são duas linguagens diferentes. Ela abrange quatro paradigmas de programação, com abstração de dados, programação genérica, estruturada e orientada a objetos.

Um algoritmo construído pela linguagem C, contempla um conjunto de funções, a principal delas é usualmente denominada "*main*", além das já pré definidas da linguagem, outras podem ser construídas conforme a complexidade do programa. A construção de um código em C, precisa obedecer uma série de regras dessa linguagem, regras de nomenclatura, regras de variáveis, regras de escrita entre outras.

Outra grande vantagem dessa linguagem é que ela pode ser executada em diversos sistemas operacionais, com isso há disponível diversas plataformas compiladoras *open source* ou *freewares*, além dos diversos ambientes de desenvolvimentos para essa linguagem, conforme descrito em (MANZANO, 2014).

### 2.3.2 Linguagem HTML

Desenvolvida na década de 90, a linguagem HTML, "Linguagem de Marcação de Hipertexto, do inglês *HyperText Markup Language*", recebeu uma série de adaptações e reestruturações logo nos seus primeiros anos de existência.

Em 2004, a Apple, Opera e Mozilla propuseram à W3C (*World Wide Web Consortium*), portadora dos direitos do código HTML, uma melhoria da versão HTML4 para a HTML5, porém essa melhoria só aconteceu em 2006, quando a W3C se aliou à WHATWG (*Web Hypertext Application Technology Working Group*).

A grande característica da linguagem HTML5 é que é possível acessar, manipular e formatar elementos e atributos da linguagem utilizando JavaScript e CSS, desse modo as páginas web (ou *web servers*) construídas através de HTML5 podem ter abundantes ambientes interativos e diversas outras opções visuais.

Das principais vantagens da nova versão HTML, além da compatibilidade com JavaScript e *Cascading Style Sheets* (CSS) pode-se citar: a redução da necessidade de *plugins*, melhor manipulação de erro, independência do dispositivo (as mesmas marcações serão utilizadas e renderizadas em diferentes tipos de dispositivos).

Dos novos recursos da HTML5, têm-se principalmente: O elemento "canva", que permite a criação de desenho e gráficos, reprodução de mídias (áudio e vídeo), novos atributos para campos de formulário e tipos de dados, novas APIs.

As APIs (*Application Programming Interfaces*) extensões que disponibilizam aos programadores uma maneira simplificada de escreverem funções de acesso ao banco de dados, de validações de formulários, de comunicação bidirecional com o servidor, e de diversas outras funções mais complexas como cita (TERUEL, 2014).

#### 2.3.2.1 *Cascading Style Sheets*

Com a implementação da linguagem CSS (Folha de estilos em cascata, do inglês *Cascading Style Sheets*) que é traduzido como uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) ao criar uma página web, torna-se desnecessário a utilização de diversos comandos e revisões em HTML. As Folhas em Cascata também permitem o desenvolvimento de uma página web mais harmoniosa, com o emprego de uma série de elementos visuais.



Segundo Miletto, o CSS altera a maneira de organização das páginas, sendo o responsável na formatação da aparência da mesma, de modo que o HTML passa a ser utilizado somente como elemento de estruturação da página. Esse fator faz com que a atualização do algoritmo seja simplificada e automática para todas as demais páginas que compõe o código, como por exemplo a de um *site*.

Neste trabalho foi utilizado uma ferramenta que já dispõe destes recursos, que está disponível no link [Font Awesome](#), em que o mesmo disponibiliza códigos em HTML que podem ser encaixados no código fonte com o intuito puramente estético. Nesse portal há uma vasta gama de símbolos que não estão presentes na tabela ASCII, porém com a extensão disponibilizada para membros do *site* é possível realizar a exportação desses símbolos únicos, como por exemplo o símbolo de *on/off*.

## 2.4 AMBIENTES DE DESENVOLVIMENTO INTEGRADO

Apesar dos microcontroladores oferecem a vantagem de serem programados para diversos fins, isso, além de requerer um certo nível de conhecimento do programador, também é importante que fique à disposição as ferramentas as quais ele poderá utilizar para que facilitem a sua programação.

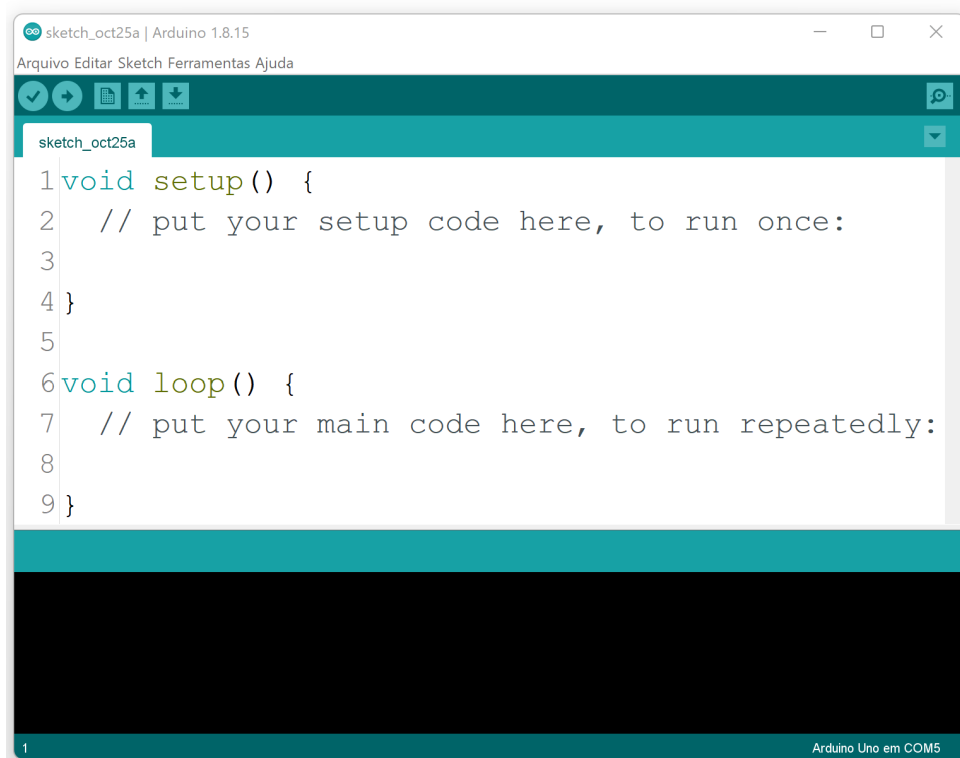
O termo integrado se relaciona a estas ferramentas, que agrupam um conjunto de soluções já incorporadas em seus ambientes de desenvolvimento. No Ambiente de desenvolvimento integrado, do inglês *Integrated Development Environment* (IDE) também é possível compilar, gravar e, em alguns, até simular o algoritmo, além de que esses *softwares* também conseguem detectar erros grosseiros de programação, atividade designada como "depuração".

Usualmente cada fabricante possui sua própria IDE, como por exemplo a Microchip® com o MPLAB para o PIC, mas assim como cada família de microcontrolador possui sua particularidade cada IDE também possui a sua. Possuir simultaneamente compilador, gravador e depurador embutido, permitir a conexão com microcontroladores de outras marcas, leiaute, versatilidade e entre outros são atributos únicos que cada empresa desenvolve de acordo com a própria identidade. Cabe ressaltar ainda, que há uma gama de IDEs desenvolvidas por terceiros, que tem como atrativo a facilidade de programação sobre as mesmas, na sua grande maioria comerciais.

### 2.4.1 A interface do Arduino IDE

O Arduino IDE é um *software* gratuito usado principalmente para escrever de maneira simples códigos em linguagens C e C++, e compilar algoritmos compatíveis para os diversos módulos de microcontroladores Arduinos, porém sua versatilidade permite a extensão para outros tipos de microcontroladores, como os ESPs, PICs, e até mesmo o novo Raspberry e o Pi Pico, pode-se visualizar a imagem da interface na versão 1.8.15 na Figura 3.

**Figura 3 – Interface de desenvolvimento do Arduino**



**Fonte: Autoria Própria.**

Estão disponíveis para sistemas operacionais como: MAC, Windows, Linux e roda na plataforma Java<sup>®</sup> que vem com funções e comandos embutidos que desempenham um papel vital para depuração, edição e compilar o código no ambiente.

O código escrito, na plataforma IDE irá gerar um arquivo com extensão "ino" e outro com extensão "hex", sendo este último um arquivo de código binário que é então transferido para a memória do microcontrolador.

O ambiente IDE contém principalmente duas partes básicas, o editor e compilador, onde o primeiro é usado para escrever o código necessário e o segundo é usado para compilar e enviar o código binário para o microcontrolador.

É possível também realizar uma série de expansões na plataforma, as chamadas bibliotecas (arquivos em extensão .h), que carregam novas funções e metodologias novas para melhor adequação ao algoritmo desejado, conforme Mohamed e Ali al Dahoud, e assim, diminuir o esforço do programador.

#### 2.4.2 Bibliotecas

A IDE possui uma vasta gama de bibliotecas já instaladas, mas novas também podem ser implementadas no algoritmo via IDE para facilitar a escrita do código, no portal *github* são encontradas os mais diversos tipos de bibliotecas adaptadas para a IDE, desde funções matemáticas avançadas a integração de sensores, basta baixar-las e seguir o processo de instalação explicado na página da própria biblioteca.

A inclusão de bibliotecas específicas é indispensável para o algoritmo de Monitoramento e Controle de Energia, são elas: “EmonLib.h”, “WiFi.h”, “ShiftRegister74HC595.h” e “HTTPClient.h”, a sintaxe para incluir qualquer biblioteca no Arduino IDE é: #include “NomeDaBiblioteca.h”

A “HTTPClient.h” é uma Biblioteca para fazer solicitações HTTP GET, POST e PUT de forma simplificada em um *Web Server*. Será utilizada para realizar a comunicação do *Web Server* do projeto com uma aplicação criada nas no Google Sheets (Planilhas Google), onde será realizado o armazenamento de dados do projeto conforme descrito em McEwen.

A biblioteca “EmonLib.h” é necessária para conversões digitais de 12 bits, com ela é possível ler os valores de corrente ou tensão de um sensores que realizem esse tipo de medição (no estudo em questão, Sensor de Corrente não Invasivo HMCT103C).

A “WiFi.h” é essencial para qualquer projeto que necessite uma conexão à *internet*, ela possibilita que o microcontrolador (que deve possuir um *Ethernet Shield* acoplado ou associado) seja capaz de responder a uma solicitação HTTP, com ela também é possível criar um *Web Server* simples, permitindo a criação de uma página web via HTML ou/e CSS capaz de enviar ou receber os valores de entradas e saídas do microcontrolador. (ARDUINO, 2021b).

Ademais a “ShiftRegister74HC595.h” é voltado para o Circuito Integrado 74HC595, ela simplifica o uso de desses registradores deslocamento. Essa biblioteca permite definir pinos individuais e se encarrega de deslocar os bytes para escrita ou leitura.

Também permite realizar a associação de vários 74HC595 para aumentar ainda mais o número de portas do microcontrolador. (ARDUINO, 2021a).

#### 2.4.2.1 Github

A plataforma GitHub® é um serviço de hospedagem e repositório Git. Enquanto o Git é um tipo de *software*, o GitHub fornece uma interface gráfica baseada na web. Ele também fornece controle de acesso e vários recursos de assessoramento, como tutoriais, códigos fontes e ferramentas básicas de gerenciamento de tarefas para cada projeto, expõe em Finley. O Github também pode ser utilizado como uma rede social entre os usuários.

## 2.5 EQUIPAMENTOS AUXILIARES

Os equipamentos listados a seguir, são os principais componentes auxiliares que contribuem para que o algoritmo desenvolvido e interpretado pelo microcontrolador desempenhe todas as funções designadas à este projeto.

### 2.5.1 Sensor de Corrente não Invasivo HMCT103C

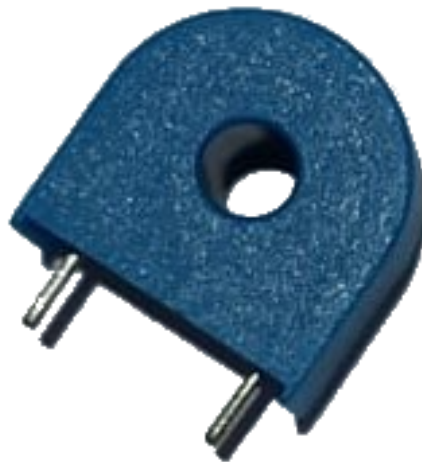
Os sensores HMCT103C (ou HWCT) são peças chaves para a execução desse projeto, conforme pode ser observado na Figura 4, são equipamentos de efeito Hall (funcionam baseados na diferença de potencial transversal a um fluxo de corrente elétrica, que surge na presença de um campo magnético perpendicular à corrente) que operam como um secundário selado de um transformador de corrente, enquanto o condutor que será medido o valor de corrente funciona como o primário desse transformador, especificado em (ETC DATASHEET, 2021).

Especificações:

- Microtransformador de corrente de precisão integrado;
- Permite medição em corrente alternada de 5mA a 5A;
- Corrente primária nominal em 50/60 Hz: 5 A;
- Corrente primária máxima em 50/60 Hz: 20 A;
- Relação dos enrolamentos:  $N_p/N_s = 1/2500$  (1/1000);

- Resistência DC do enrolamento a 20°C: 155  $\Omega$ ;
- Precisão RL  $\leq$  10  $\Omega$ : 2%;
- Temperatura de operação: -40 a 85°C.

**Figura 4 – Sensor de corrente não invasivo HMCT103C**



**Fonte: Autoria Própria.**

A grande vantagem de se utilizar esse tipo de sensor é que sua característica não invasiva permite realizar a leitura de corrente de um condutor sem a necessidade de rompê-lo (amperímetros devem ser colocados em série em um circuito elétrico), esse fator aumenta a segurança ao manusear o sensor e evita esforços adicionais de acoplamento ao realizar as medições, além de ser disponível por um preço acessível.

### 2.5.2 Circuito Integrado 74HC595

Os 74HC/HCT595 são dispositivos CMOS de alta velocidade e são compatíveis com o CIs TTL de baixa potência, é em sua suma definição um registrador de deslocamento serial de 8 estágios com um registro de armazenamento e saídas de 3 estados. O registro de deslocamento e registro de armazenamento tem *clocks* separados, extraído de (NXP SEMICONDUCTORS, 2021).

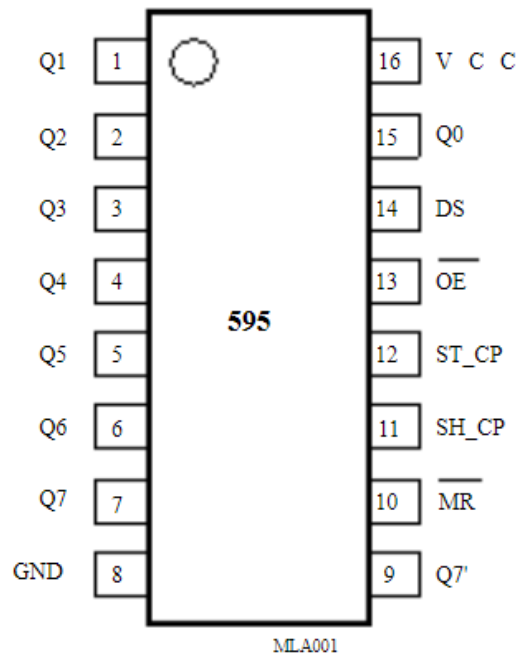
O registrador de deslocamento tem uma entrada serial (DS) e uma saída serial padrão (Q7') para cascata. Também é fornecido com *reset* assíncrono (LOW ativo) para todos os oito estágios de registro, a Figura 5 ilustra a pinagem desse CI.

Esses CIs também podem ser utilizados como expansores de portas digitais de alguns microcontroladores como o Arduino<sup>®</sup> e o ESP32, e a sua função cascata

permite expandir ainda mais o número de portas, além de serem circuitos de baixo custo, e podem ser facilmente substituídos em uma placa de circuito impresso.

É essencial a implementação da biblioteca “ShiftRegister74HC595.h” para facilitar a construção do algoritmo que permita que o 74HC595 funcione como expensor de portas.

**Figura 5 – Circuito Integrado 74HC595**



**Fonte: NXP Semiconductors.**

### 2.5.3 Módulo rele

O rele é um equipamento eletromecânico que funciona como um interruptor, quando há circulação de corrente elétrica em suas bobinas internas provoca a existência de um campo eletromagnético, que aplica uma força de atração nos contatos internos do rele, o rele, dependendo da quantidade de pinos, pode possuir contatos Normalmente aberto (NA), (Normalmente fechado (NF) ou ambos.

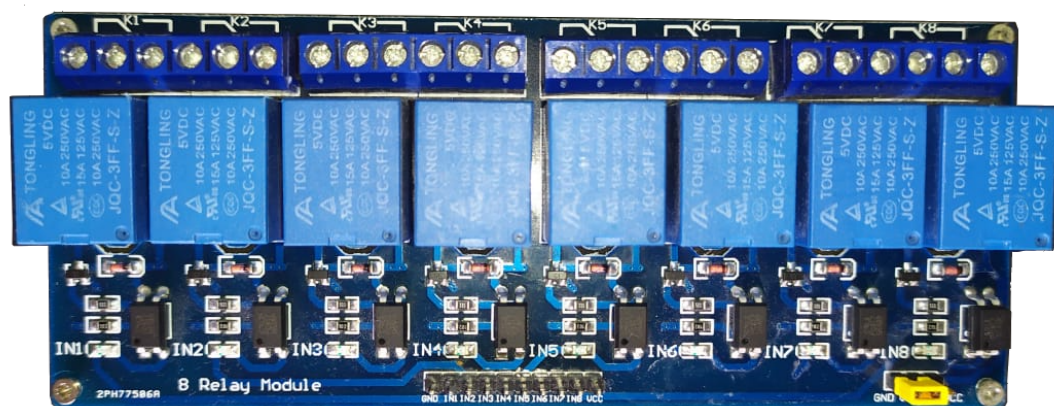
Os *shields* são módulos que podem ser facilmente conectados, sobrepostos a uma placa Arduino, tendo como finalidade estender as capacidades internas da placa principal como aparece em (SILVA; NARDOTO; JORGE, 2016).

Os módulos reles estão disponíveis no mercado nas mais diversas especificações, com 1, 2, 4, 8 e até 16 reles acoplados, com espaço para adaptações. Além

disso alguns *shields* podem conter foto acopladores, que são dispositivos de segurança que isolam o circuito elétrico acoplado ao módulo rele do microcontrolador, na Figura 6 pode-se observar um módulo com oito unidades reles.

ROSSI explica que o funcionamento do optoacoplador ou foto acoplador baseia-se no efeito fotoelétrico, no qual um feixe de luz infravermelha, produzido pelo LED quando se aplica uma tensão entre os terminais 1 e 2, polariza a base de um fototransistor, permitindo a condução entre coletor e emissor.

**Figura 6 – Módulo rele 8 unidades**



**Fonte: Autoria própria.**

## 2.6 CONVERSÃO DE BASE

Base64 é um tipo de codificação (ou decodificação) com finalidade de representar sequências definidas de caracteres, como por exemplo uma palavra qualquer, em uma linguagem diferente da 8-bits.

É utilizada para codificar dados binários quando estes precisam ser transferidos por uma entrada tipo caixa de texto, um exemplo é mostrado no Quadro 2. Esse método garante que os dados permaneçam intactos durante a transferência. Alguns aplicativos de e-mail, armazenamento de dados em XML ou JSON são as principais aplicações do Base64, descrito em (JOSEFSSON, 2006).

Quadro 2 – Exemplo prático de codificação Base64

<b>Conteúdo do texto</b>	E	S	P
<b>ASCII</b>	69	83	80
<b>Bit Padrão</b>	0 1 0 0 0 1 0 1	0 1 0 1 0 0 1 1	0 1 0 1 0 0 0 0
<b>Índice</b>	17	21	16
<b>Codificação Base64</b>	R	V	Q

Fonte: Aatoria própria (2021).

## 2.7 CIBERSEGURANÇA

Em 2020 foi constatado pela Symantec que o Brasil é o terceiro país que mais sofre ataques cibernéticos em aparelhos conectados à *internet*, conforme (FANTINATO, 2021). Níveis de cibersegurança são recursos imprescindíveis atualmente para proteção de dados. Uma definição apropriada de cibersegurança é a seguinte:

Cibersegurança é a prática proteger sistemas e ativos de informação tais computadores, servidores, dispositivos móveis, redes e sistemas eletrônicos entre outros, contra ameaças ou ataques cibernéticos. (PORTAL DA INDÚSTRIA, 2021).

Fabricantes como Siemens® e Scheneider® por exemplo, possuem certificado Achilles nível 2 em equipamentos como CLP e inversor de frequência com compatibilidade IoT. Essa característica comprova que o sistema desses equipamentos possuem defesas efetivas contra invasões externas via protocolos de comunicações *Ethernet*, IP, TCP e algumas outras. (SIEMENS, 2021).

A literatura de Pinheiro apresenta que a proteção de dados pessoais de uma residência e de quem nela reside, assim como no âmbito industrial, também necessitam de proteção, informações mesmo que irrelevantes devem ser mantidas em segurança, pois acima de tudo são pessoais. No Brasil, em 2018, foi sancionada a Lei Geral de Proteção de Dados Pessoais (Lei Nº. 13.709/2018), que preceitua direitos para os titulares e deveres para as empresas.

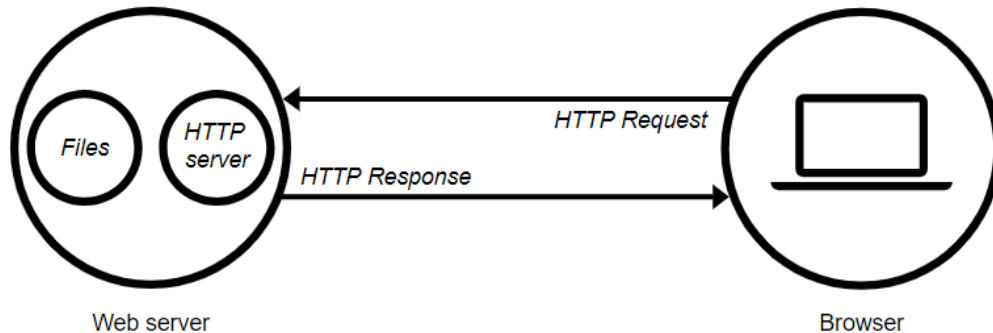
## 2.8 WEB SERVER

Um *Web Server* ou Servidor Web, é um tipo de aplicação de armazenagem e processamento de informações, e que retorna esses dados como uma página web, tais páginas podem ser acessadas utilizando seu nome de domínio (DNS) através dos navegadores de *internet*. Essa comunicação entre o navegador e o servidor é realizada



via protocolo HTTP (Hypertext Transfer Protocol), e necessita uma conexão com à *internet*.

**Figura 7 – Comportamento de um Web Server**



**Fonte: (MOZZILA).**

Para publicar uma página web, é necessário ou um servidor web estático ou um dinâmico. Um Servidor Estático, se caracteriza pelo envio de arquivos tal como foram criados e armazenados, e consiste em um *hardware* com um servidor HTTP. Analogamente um servidor dinâmico atualiza os arquivos hospedados antes de enviá-los ao navegador através do servidor HTTP, e consiste em servidor web estático com *software* adicional, mais comumente um servidor de aplicações e um banco de dados, como descrito em (MOZZILA, 2021a).

## 2.9 CONSUMO DE ENERGIA E TARIFAÇÃO GRUPO B - RESIDENCIAL

Para a aplicação das tarifas de energia elétrica, os consumidores são identificados por classes e subclasses de consumo, residencial, industrial, comercial, rural, poder público, iluminação pública, serviço público e consumo próprio.

No Brasil, as unidades consumidoras são classificadas em dois grupos tarifários, Grupo A e Grupo B. O agrupamento é definido em função do nível de tensão em que são atendidos e em função da demanda (kW). As unidades consumidoras atendidas em tensão abaixo de 2.300 volts são classificadas no Grupo B (baixa tensão), e o Grupo A se enquadra em tensões maiores ou iguais a 2.300 volts. O Grupo B é dividido em subgrupos, de acordo com a atividade do consumidor, residencial e residencial baixa renda (B1), Rural e cooperativa de eletrificação rural (B2), as demais classes (B3) e por fim iluminação pública (B4), extraído da (ANEEL, 2021).

A tarifa de energia corresponde ao valor cobrado por unidade de energia

(R\$/kWh) que contempla os custos desde a geração até a disponibilização aos consumidores. Os segmentos que contribuem para o cálculo da tarifa são os custos com a aquisição de energia elétrica, custos relativos ao uso do sistema de distribuição, custos relativos ao custo do sistema de transmissão, perdas técnicas e não técnicas encargos setoriais e impostos. Por meio de resoluções, a Agência Nacional de Energia Elétrica (ANEEL) publica o valor da tarifa de energia sem os tributos, por classe de consumo e, com base nesses valores, as distribuidoras de energia incluem os tributos (PIS, Cofins, ICMS e CIP) e emitem a fatura de energia aos consumidores, estudo apresentado por (BARROS, 2019).

**Figura 8 – Composição de Preço da fatura de Energia**

Valores Faturados							
NOTA FISCAL CONTA DE ENERGIA ELETRICA no. [REDACTED] Serie B							
Emitida em 01/08/2021							
Produto Descricao	Un	Consumo	Valor Unitario	Valor Total	Base de Calculo	Aliq. ICMS	
01 ENERGIA ELETRICA CONSUMO	kWh	151	0,822649	124,22	124,22	29,00%	
02 ENERGIA CONS. B. VERMELHA P2	kWh			21,11	21,11	29,00%	
03 CONT ILUMIN PUBLICA MUNICIPI				17,26			
Base de Calculo do ICMS:		145,33	Valor ICMS:		42,14	Valor Total da Nota Fiscal: 162,59	

**Fonte: Autoria Própria.**

A fatura de energia da Figura 8, é referente à concessionária COPEL, responsável pela distribuição de energia no Paraná. A energia elétrica de consumo, é determinada pela potência (P) do equipamento (em kW) pelo tempo (t) em que ele se encontra acionado, o valor de potência pode ser encontrado na embalagem ou no próprio equipamento, com esses valores em nota, basta multiplica-los pela tarifa (\$) atribuída pela concessionária, para obter o valor à pagar da conta, conforme a equação:

$$C = P \times t \times \$ \quad (1)$$

A potência também pode ser calculada pela multiplicação da corrente pela tensão de entrada, com tudo, para isso são necessários instrumentos de medição, o amperímetro e o voltímetro (ou um multímetro para os dois).

### 3 DESENVOLVIMENTO

#### 3.1 DESIGNAÇÃO DO MICROCONTROLADOR

Como visto na Seção 2.2, após a invenção e comercialização do primeiro microcontrolador uma vasta gama desses componentes com características únicas de cada fabricante foram ganhando espaço no mercado.

Contudo, também apresentam suas limitações, sejam elas técnicas ou comerciais, e em alguns casos, alguns apresentam capacidades inferiores quando comparados à outros. Para esse estudo levou-se em consideração o Arduino UNO R3 (ATmega328), ESP32 e o PIC16F877a.

Para uma escolha adequada, alguns fatores devem ser levados em conta, essencialmente os de desempenho e a disponibilidade de funções, como número de portas, compatibilidade *Wi-Fi*, memórias e, para este caso, o custo. A Tabela 1 compara as tecnologias em questão.

**Tabela 1 – Comparação de Microcontroladores**

<b>Microcontrolador</b>	<b>ATmega328</b>	<b>ESP32</b>	<b>PIC16F877a</b>
Custo	R\$ 72,00	R\$ 77,28	R\$ 35,90
Portas Analógicas	6	18	2
Portas Digitais	14	34	33
Memória <i>Flash</i>	32 KB	16 MB	14,3 KB
EEPROM	1 KB	4 MB (da Flash)	256 Bytes
Velocidade do <i>Clock</i>	16 MHz	80 a 240 MHz	0 a 20 MHz
Módulo <i>Wi-Fi</i> embutido	Não	Sim	Não
<i>Bluetooth</i> Integrado	Não	Sim	Não

**Fonte: Aatoria própria (2021).**

Com base no exposto, cada equipamento apresenta suas vantagens e desvantagens, mesmo sendo possível conectar-se à internet com um Arduino UNO R3 associando um módulo *Wi-Fi* à ele, ainda continua sendo muito mais vantajosa a utilização do ESP32 pois já possui este módulo especificamente, analogamente ao compararmos o ESP32 com o PIC.

Todos os custos da Tabela 1 foram obtidos do mesmo *E-commerce*, Baú da Eletrônica.

## 3.2 FLUXOGRAMA DO PROJETO

Conforme descrito na Seção 3.1, e na Tabela 1, o ESP32, é o microcontrolador ideal para um projeto de Controle e Monitoramento de Energia, pois, ele não necessita de nenhum outro acessório suplementar para atingir os requisitos de um código de programação elementar para domótica, tornando-o o microcontrolador com melhor custo benefício se comparado aos outros, pois não é necessário o investimento monetário em outros módulos auxiliares, pois já dispõe do que serão utilizados neste projeto, tais como: *Wi-Fi* e *Bluetooth*.

O desenvolvimento do algoritmo desse projeto vai além do elementar, com isso é necessário a utilização de outros componentes para melhor adequação do projeto proposto.

Utilizando a plataforma de programação do Arduino IDE com os devidos ajustes para comunicação com o ESP32, o primeiro passo para a programação é a declaração das condições iniciais, estas que passarão as informações de estado, tipo e afins das variáveis do código. É nessa etapa que as bibliotecas citadas Seção 2.4.2 são incluídas.

O intervalo de tempo é extremamente pequeno de execução de cada instrução, pois o mesmo opera numa elevada frequência (considerando a frequência máxima de 240 MHz, teria-se um período de 4,2 ns (nanosegundos)), que deve ser multiplicado pelo número de linhas do código - Ciclos da CPU, inicia-se a invocação do *Web Server* e em seguida a conexão com o banco de dados que será utilizado.

Para isso, vale ressaltar que as variáveis de conexão à internet, bem como nome da rede *Wi-Fi* e senha devem estar condizentes, caso contrário, o ESP32 não fará a conexão com à internet. Esse erro pode ser tratado de diferentes maneiras, variando de programador para programador.

Já sabe-se que *cybersegurança* é um fator importante a ser considerado, então uma vez *online*, o *Web Server* fará uma requisição de Protocolo de transferência de Hipertexto, do inglês *Hypertext Transfer Protocol* (HTTP) do usuário, esses dados digitados são tratados utilizando codificação e decodificação fazendo uso da técnica BASE64.

Se o *login* e senha digitados forem os mesmos que os cadastrados (via programação na IDE), o usuário terá acesso ao conteúdo da página *web* do projeto.

Essa página contempla a interface com o usuário e possui uma área de *logout*

e quatro abas, sendo elas:

- Início;
- Consumo;
- Configurações; e
- Desenvolvedor.

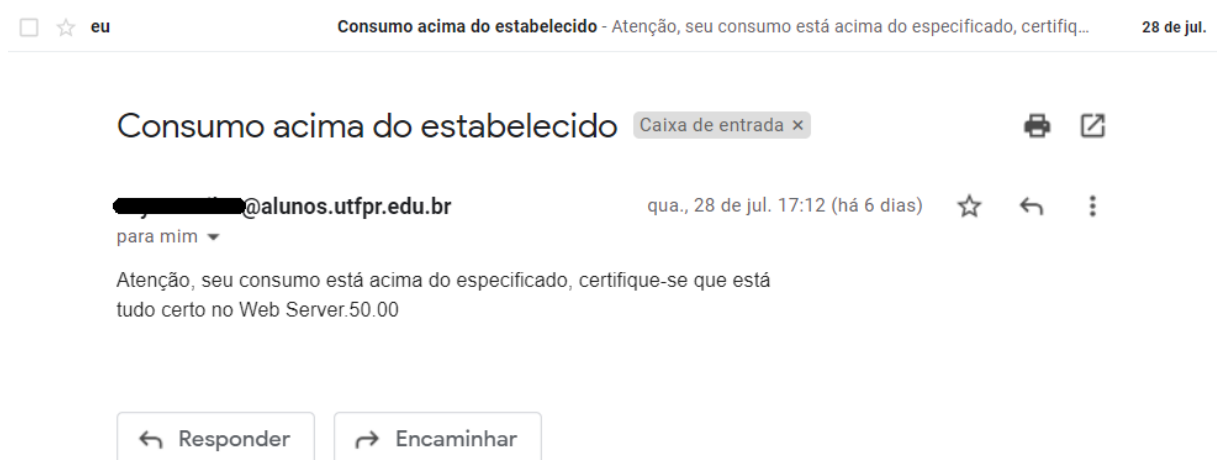
Cada uma com sua finalidade específica.

A medição de corrente via sensor *HMCT103C* se inicia junto com a conexão do *Web Server*, contudo os valores de consumo necessitam o preenchimento dos valores de tarifação e tensão, que se inicialmente encontram-se zerados.

Inicialmente é feita a calibração, e estes dados são atualizados no sistema na aba Configurações.

Outra característica importante da aba Configurações é o campo "Valor do Alarme", que é uma variável da função de notificação via e-mail. A condição de "Alarme" é disparado da seguinte maneira, o usuário especifica um valor máximo de consumo e se o valor medido for maior que o especificado, uma mensagem é enviada, conforme modelo que pode ser visto na Figura 9, se depois de seis horas, o consumo ainda for maior que o especificado, outra mensagem será enviada, e assim por diante até que o valor medido seja menor que o determinado. Vale ressaltar que essa função depende tanto do código desenvolvido no Arduino IDE quanto no código do desenvolvido em *Script do Google Sheets*®.

**Figura 9 – E-mail de notificação enviado pelo ESP32**



**Fonte: autoria própria (2021).**

Simultaneamente, existe outra função associada ao *script do Google Sheets*® que é responsável pelo armazenamento dos valores de consumo medido, a cada duas

horas, o ESP32 enviará os dados obtidos para uma planilha do *Google* o dia e hora no formato: dd/MM/aaaa HH:mm com o respectivo valor de consumo. A cada iteração, automaticamente os valores são colocados um abaixo do outro. Tanto o intervalo de duas horas quanto o de seis horas da função de alarme podem ser alterados via código no Arduino IDE.

O formato da data, a quantidade de variáveis armazenadas na planilha, o conteúdo do e-mail e outras informações já destacadas, são passíveis de alteração via códigos IDE e *Google Sheets*<sup>®</sup>, porém demandam conhecimento do usuário, contudo, estas alterações deve ser implementadas pelo desenvolvedor e sofrendo apenas ajustes no textitstart-up. A Figura 10 apresenta o fluxograma do projeto desenvolvido.

### 3.3 CIRCUITO ELÉTRICO DO PROJETO

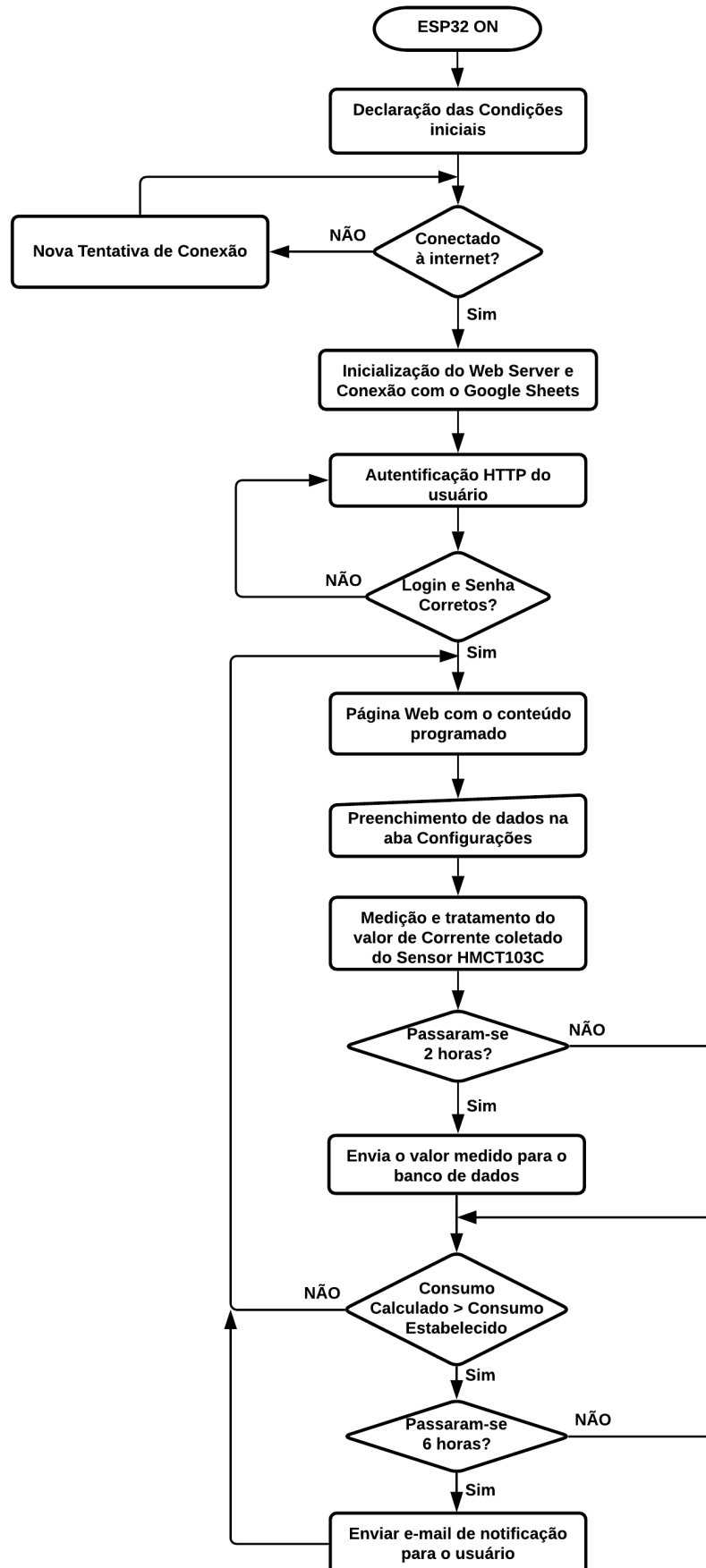
Para o funcionamento do projeto, não é necessário somente a programação do microcontrolador, é preciso interligar os pinos do ESP32 com os componentes, como o expensor de portas, sensores de corrente e outros periféricos.

A Figura 11 representa o circuito eletrônico do projeto, entretanto ainda resta a medição individual, que é capturada nas saídas dos reles e se destina aos objetos de controle, como tomadas e lâmpadas. A Figura 11, foi realizada via *software* Eagle da Autodesk<sup>®</sup>, utilizando a licença de estudante, com esse *software* também é possível realizar o desenho da Placas de Circuito Impresso (PCI) do circuito, caso desejado.

No circuito pinos específicos do ESP32 são conectados ao 74HC595 e ao HMCT103C, esses pinos necessitam respeitar as características conforme Figura 1 (pinos analógicos para o HMCT), além de serem declarados no algoritmo programado do projeto. Também é importante ressaltar que para sensores de leitura analógica, como é o caso do HMCT103 os pinos do ESP32 também sejam analógicos, as bibliotecas de sensores já se encarregam de enviar o algoritmo correto nas funções carregadas, como é o caso da "EmonLib.h" para o sensor de corrente.

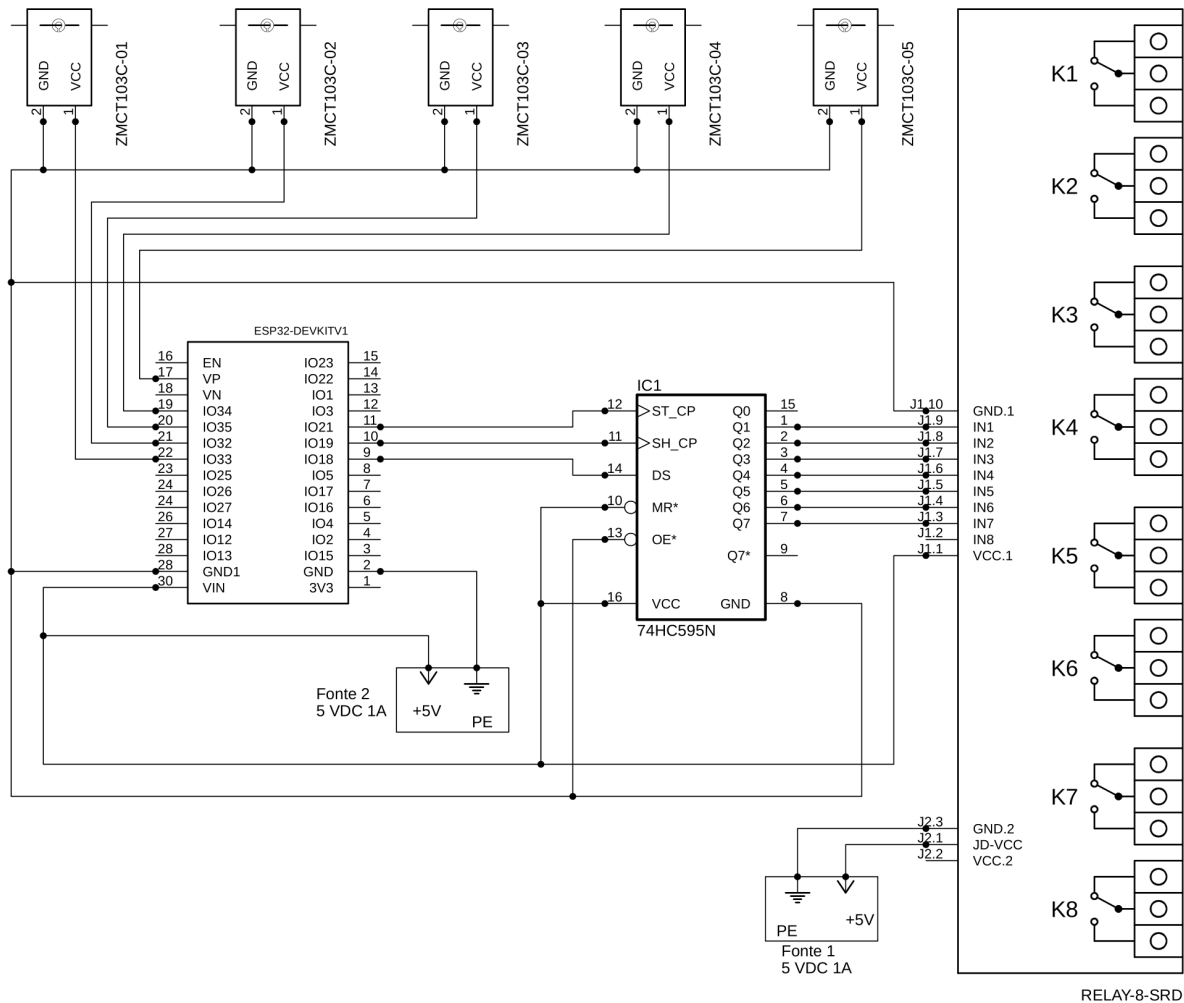
Analogamente a biblioteca "ShiftRegister74HC595.h" exerce o mesmo com o registrador de deslocamento 74HC595. E como visto na Figura 11, três pinos do ESP32 possibilitam a utilização de até oito saídas, e o mais excepcional, é que com a função cascata do Circuito Integrado (CI) é possível interligar vários outros 74HC595 utilizando ainda somente os três pinos do ESP32, para isso é necessário uma simples alteração

Figura 10 – Fluxograma de Processo da Programação



Fonte: autoria própria (2021).

Figura 11 – Circuito Eletrônico do Projeto



Fonte: autoria própria (2021).

no código e na ligação do circuito.

Quanto aos reles, com a utilização de uma fonte de alimentação separada o circuito apresentará maior segurança, dessa forma são reduzidas as chances de danos ao ESP32 em caso de condições adversas. Os reles não demandam adaptações no código, pois a placa utilizada já possui as medidas de segurança (opto acoplador e afins) e de conexões com os reles, as saídas dos reles funcionam como interruptores, com tudo são acionadas pelos pinos de entradas na lateral esquerda da placa, conectados ao 74HC595, que já possui o funcionamento adequado ao algoritmo.

Os reles atuam como um interruptor no circuito, no caso das tomadas (carga), o rele, quando acionado faz com que o circuito em questão fique aberto, interrompendo a circulação de corrente. Esse fator pode mostrar-se vantajoso, pois evita o modo *stand-by* dos aparelhos (modo espera). As ligações elétricas nas saídas dos rele podem ser



**Figura 12 – Declaração das variáveis do sensor de corrente e do 74HC595**

```

//----- VÁRIAVEIS SENSOR CORRENTE -----

#define CURRENT_CAL 1.883 //VALOR DE CALIBRAÇÃO (DEVE SER AJUSTADO EM PARALELO
                          //COM UM MULTÍMETRO MEDINDO A CORRENTE DA CARGA)

#define pinoSensor1 36 //PINO ANALÓGICO EM QUE O SENSOR1 ESTÁ CONECTADO
EnergyMonitor emon1; //INSTÂNCIA DO SENSOR DE CORRENTE 1
double currentDraw1; //VARIABLE QUE ARMAZENARÁ OS VALORES COLETADOS

#define pinoSensor2 32 //PINO ANALÓGICO EM QUE O SENSOR2 ESTÁ CONECTADO
EnergyMonitor emon2;
double currentDraw2;

#define pinoSensor3 33 //PINO ANALÓGICO EM QUE O SENSOR3 ESTÁ CONECTADO
EnergyMonitor emon3;
double currentDraw3;

#define pinoSensor4 34 //PINO ANALÓGICO EM QUE O SENSOR4 ESTÁ CONECTADO
EnergyMonitor emon4;
double currentDraw4;

#define pinoSensor5 35 //PINO ANALÓGICO EM QUE O SENSOR5 ESTÁ CONECTADO
EnergyMonitor emon5;
double currentDraw5;

//----- VÁRIAVEIS 74HC595 -----

#define pinData 18
#define pinClock 19
#define pinLatch 21
#define quantidadeCI 1

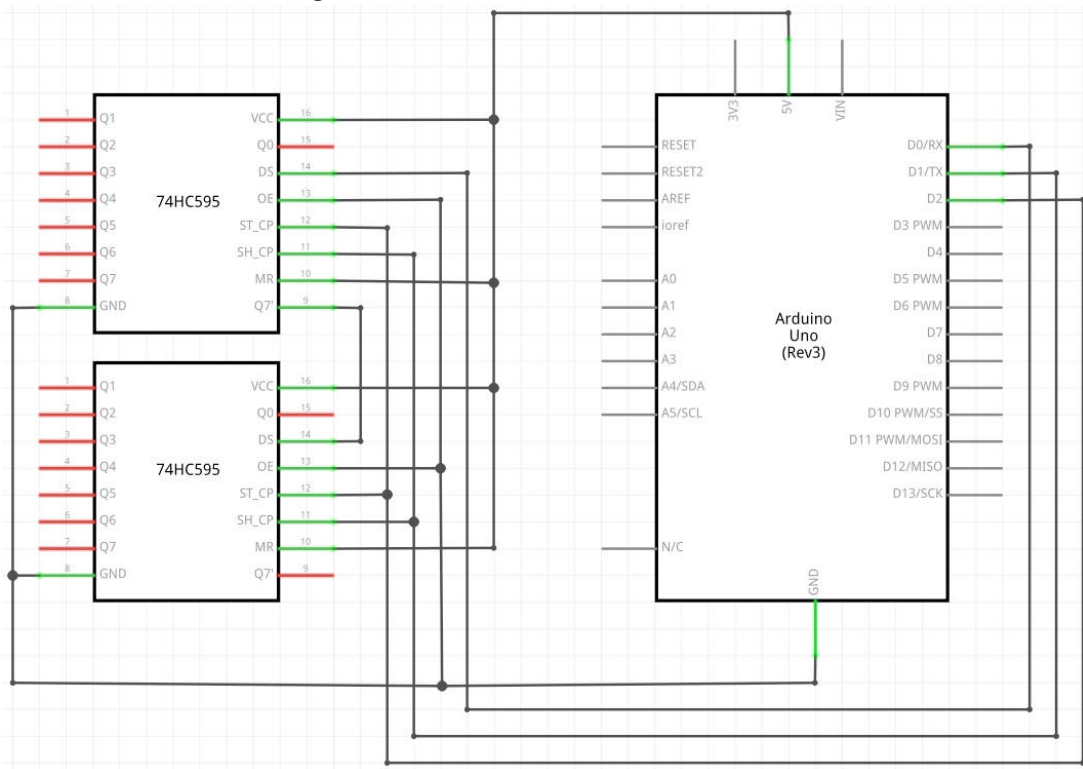
ShiftRegister74HC595<quantidadeCI> rd(pinData, pinClock, pinLatch);
// parametros: <numero de registradores> (pino data, pino clock, pino latch)
//OBS: Para mais de 1 registrador atentar-se ao circuito (cascata)

int pin1_rd = 1;
int pin2_rd = 2;
int pin3_rd = 3;
int pin4_rd = 4;
int pin5_rd = 5;
int pin6_rd = 6;
int pin7_rd = 7;

```

**Fonte: autoria própria (2021).**

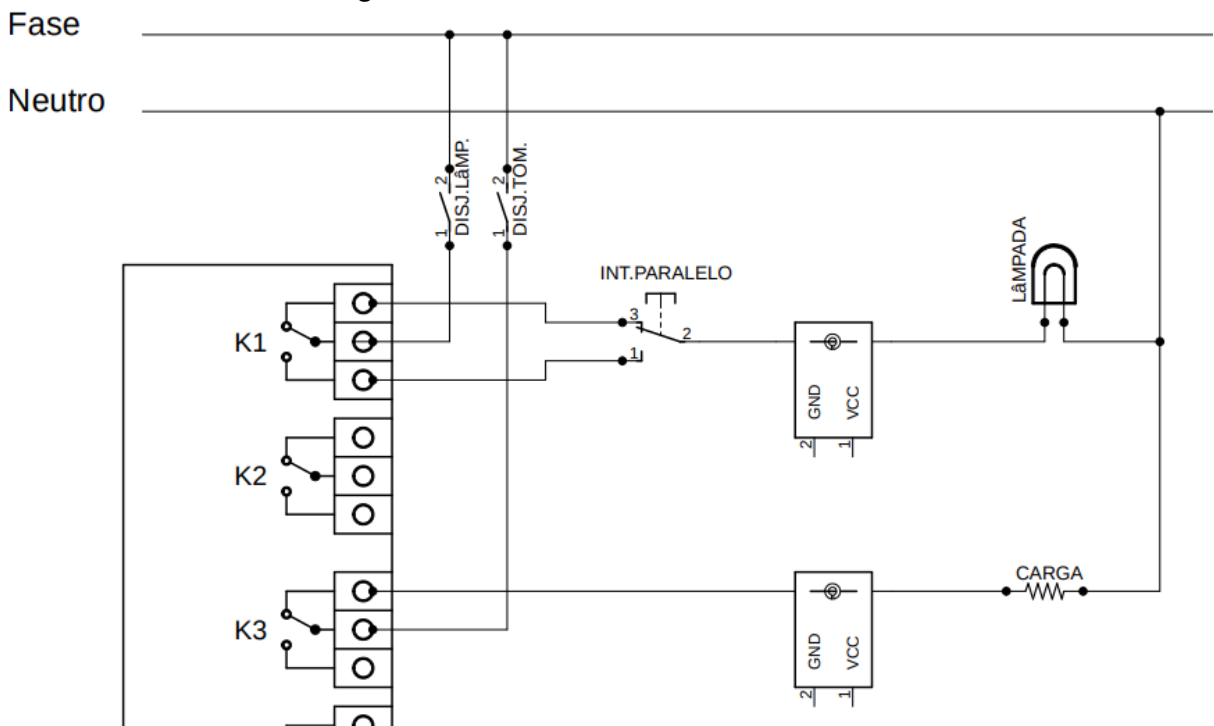
Figura 13 – Circuito em cascata 74HC595



Fonte: Denk, Timo.

observadas na Figura 14, utiliza-se os contatos normalmente fechados do rele, pois em casos de falta de energia, os pontos continuarão a funcionar normalmente.

Figura 14 – Circuito elétrico saída dos reles



Fonte: autoria própria (2021).

O *stand-by* é um modo de economia de energia, contudo, Mattede alerta que a soma de todos os aparelhos em *stand-by* de uma residência pode totalizar até 12% do consumo total de energia elétrica. Dito isso, ao analisar o projeto, verifica-se que se a saída do rele for associada a um filtro de linha, os resultados obtidos podem ser ainda melhores, pois há a possibilidade de interromper o modo *stand-by* de vários aparelhos de uma só vez.

Atualmente, o cenário energético do Brasil está em uma situação crítica, como consequência da crise hídrica que afeta os reservatórios das usinas hidrelétricas, afetando diretamente o sistema tarifário aplicado aos brasileiros. O preço da energia elétrica ficou mais caro a partir de julho (2021) com o reajuste na bandeira tarifária vermelha patamar 2, a cobrança extra passou de R\$ 6,24 para R\$ 9,49 a cada 100 kWh consumidos – o que corresponde a uma alta de 52%, conforme reportagem publicada no (G1 GLOBO, 2021). Esse fator pode aumentar ainda mais a porcentagem da influência de aplicações como o *stand-by* no valor da conta de energia.

### 3.4 WEB SERVER DO PROJETO

A página web do projeto é a ponte entre as funções do algoritmo e o usuário, é onde está localizada a IHM. Esse *web server* é do tipo estático, e pode ser chamado de *web server* síncrono, possui algumas limitações se comparado com o assíncrono (dinâmico), porém atende as demandas do projeto.

O HTTP é a base da comunicação da internet, fazendo com que *sites* possam ser encontrados facilmente. Portanto, quando um usuário digita a URL do *site*, o navegador gera uma solicitação e envia para o IP colocado na URL, (MOZZILA). Para a obtenção do IP via ESP32 é necessário a utilização do monitor serial (com ele é possível enviar e receber informações na forma de texto), o IP será interpretado como uma URL de *site* no navegador, que direcionará para página *web* do usuário (programada via HTML/CSS na Arduino IDE), conforme é apresentado na Figura 15.

Uma vez acessada a página, serão solicitados *login* e senha do usuário via *HTTP Authentication*, que se concedida, dará acesso ao conteúdo da página web. Essa página está dividida em abas e uma área comum, para melhor interação do usuário, e cada uma dessas sessões possui uma função. A nomenclatura das abas foi definida pelo programador, e poderiam ser diferentes alterando-as via código no Arduino IDE,

contudo o nome não altera em nada o funcionamento do projeto.



**Fonte: autoria própria (2021).**

A página web além de ser compatível com a maioria dos navegadores presentes em um computador, também demonstrou ser compatível com os navegadores dos dispositivos móveis, tanto IOS como Android, contudo com alguma alteração no leiaute.

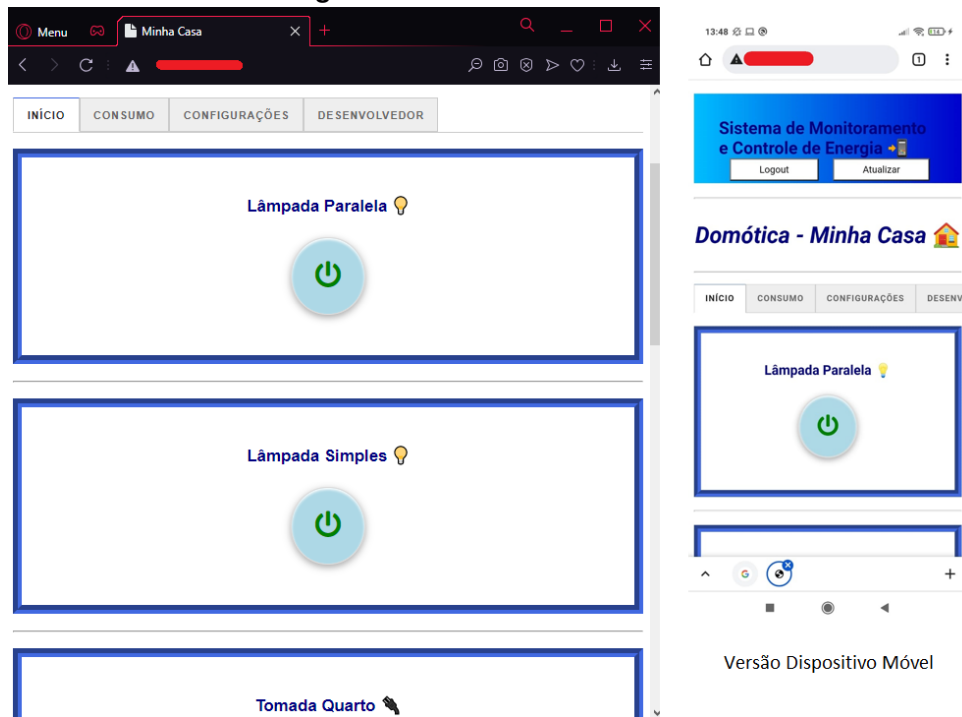
Os botões de "logout" e "atualizar" são para deslogar do *web server* e atualizar as variáveis, respectivamente, o microcontrolador atualiza constantemente os valores medidos, contudo, como se trata de um *web server* estático, essas informações só são enviadas atualizadas quando alguma interação é realizada na página.

### 3.4.1 Aba Início

A aba início, que tem seu leiaute conforme apresentado na Figura 16, é responsável por representar os botões, que alterarão o estado dos rele, *on/off*. É no leiaute desses botões que foram utilizados códigos do *Font Awesome*.

Ao pressionar um desses botões, o respectivo interpretará o código C/C++ encarregado de alterar o estado do rele. O módulo rele em questão é acionado em nível baixo, então, ao inicializar o ESP32 é importante que na declaração das condições iniciais, ele envie sinal de nível alto para os reles, para que os mesmos se iniciem desligados e ao pressionar o botão o estado de inverte, acionando ou desligando o respectivo rele.

Figura 16 – Conteúdo aba início



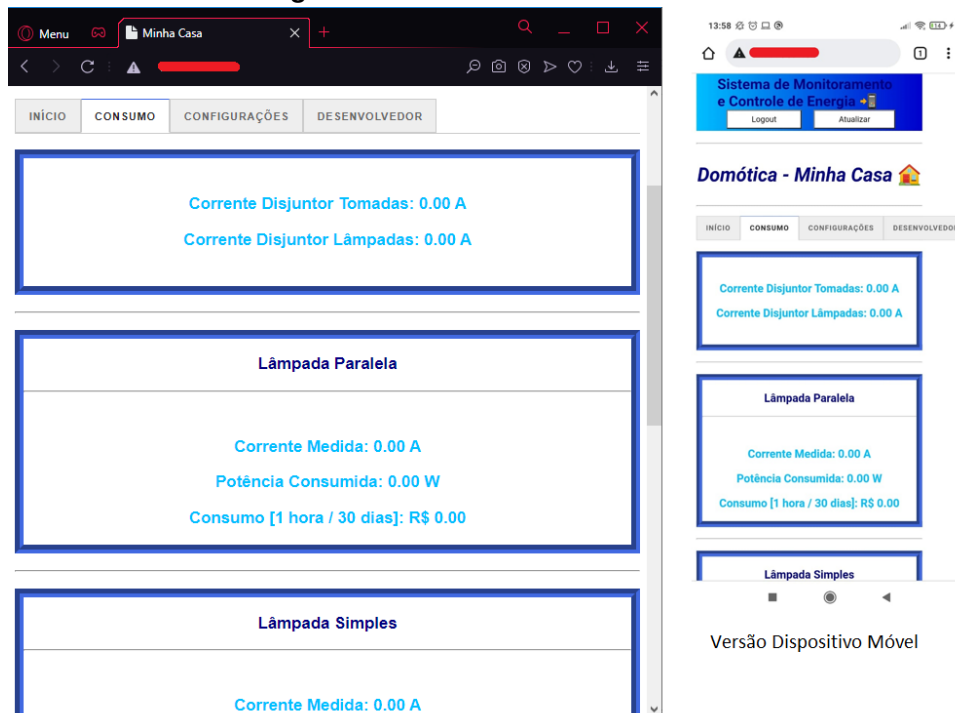
Fonte: autoria própria (2021).

### 3.4.2 Aba Consumo

Já na aba consumo, conforme apresentado na Figura 17, afim de possibilitar uma leitura de dados de forma clara e concisa evitar, é nela onde está localizado as medições e o resultado dos cálculos resultados das funções do algoritmo. É uma aba puramente visual, não há interação com o usuário nela. São esses valores apresentados que serão enviados para o banco de dados.

Para os cálculos serem realizados no algoritmo, é importante que as informações localizadas na aba configurações sejam preenchidas, pois diferente dos botões de estado dos rele, esses cálculos necessitam o preenchimento de algumas variáveis. Outro fator é a calibração do sensor de corrente HMTTC, no *github* existe um valor pré definido, contudo pode estar incoerente com a região/situação atual de uso, para a correta medição é necessário a aferição do sensor. Medir a corrente conhecida de um circuito é uma ótima técnica para a obtenção da calibração ideal.

Figura 17 – Conteúdo aba consumo



Fonte: autoria própria (2021).

### 3.4.3 Aba Configurações

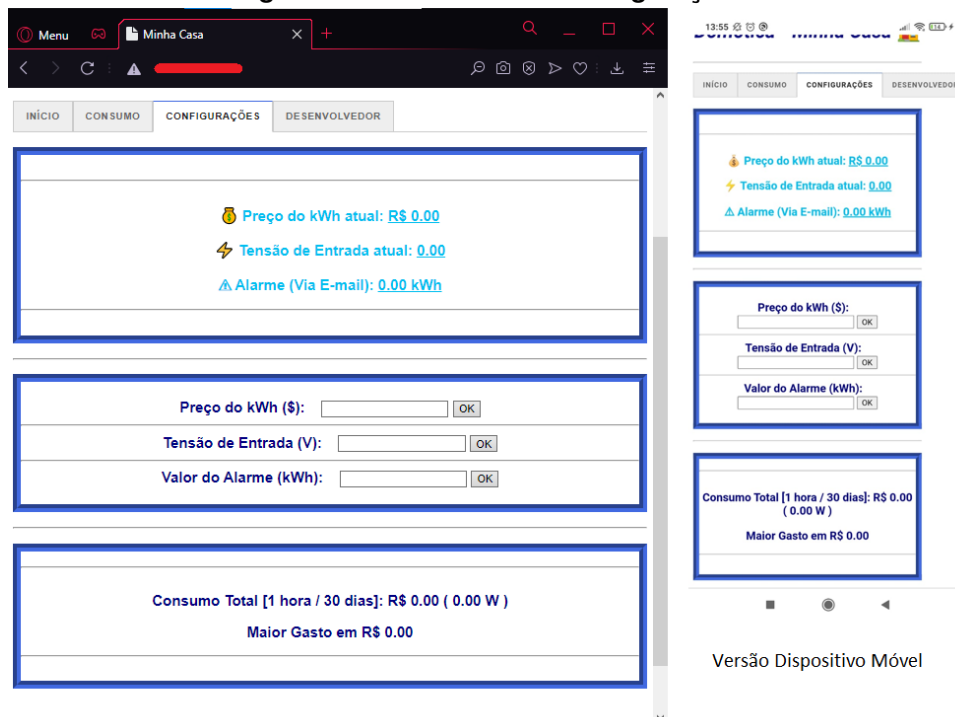
Nas configurações encontra-se a interface mais importante para a medição dos valores, é nessa aba que o usuário informa ao *software* o valor da tensão e da tarifa de energia, esses valores não alteram o funcionamento dos botões, são valores para cálculos, portanto precisam ser preenchidos de maneira correta.

O ideal é medir com um instrumento a tensão da residência, pode existir diferença no valor medido dos pontos devido às quedas de tensões, com tudo essa discrepância pode ser desconsiderada. O valor da tarifa pode ser obtido no *site* da concessionária ou na conta de energia.

Outro campo de preenchimento, é o campo do Alarme, que envia notificações via e-mail para o usuário, a notificação é enviada caso o valor medido de consumo seja maior do que o definido pelo usuário. O valor mínimo de entrada do Alarme é 10, puramente por decisão do programador, pode ser alterado via código.

Com esses três valores preenchidos o algoritmo consegue apresentar os valores de medições na página web, esses valores podem ser alterados a qualquer instante. Na Figura 18 é apresentado o resultado da página de configurações desenvolvida para este sistema.

**Figura 18 – Conteúdo aba configurações**



Fonte: autoria própria (2021).

#### 3.4.4 Aba Desenvolvedor

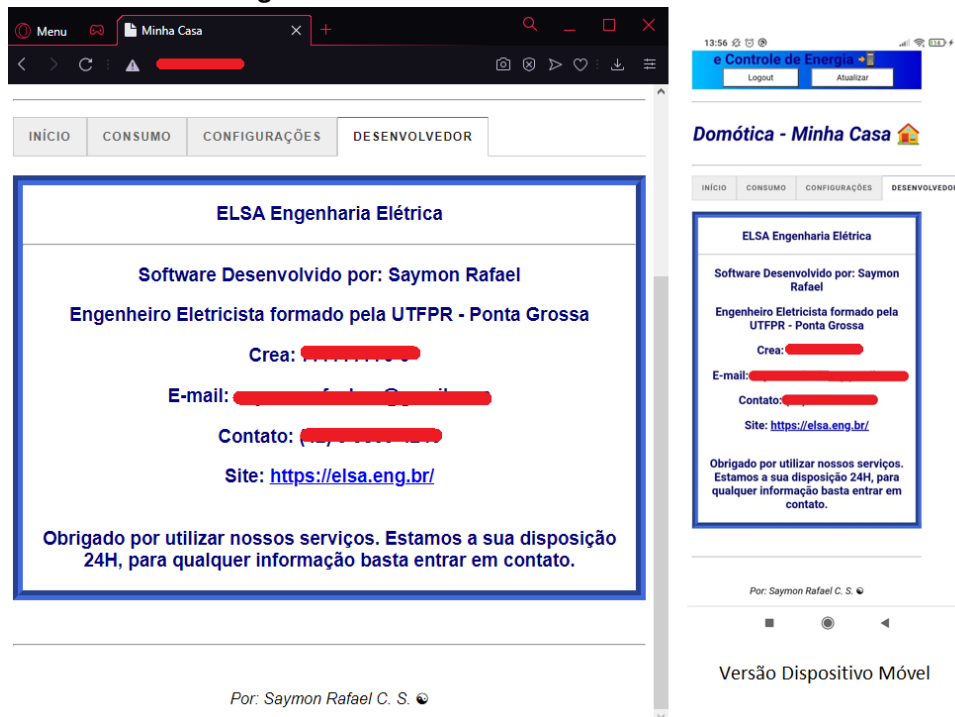
Em algumas situações, o usuário pode decidir alterar algum conteúdo do *software*, seja em termos de funcionalidade ou estético, ou no pior caso, decorrente de algum problema técnico ocorrido, na aba desenvolvedor encontram-se as informações de contato do programador para agilizar a solução do problema. É uma aba puramente informativa, conforme mostrado na Figura 19.

#### 3.4.5 HTTP Authentication

O HTTP fornece uma estrutura geral para controle de acesso e autenticação, o servidor responde ao cliente com uma mensagem do tipo 401 (Não autorizado), quando for esta a situação. Um cliente que deseja autenticar-se com um servidor pode fazer isso incluindo um campo de cabeçalho de solicitação *www-Authenticate* com as credenciais. O usuário apresentará uma solicitação de senha e usuário, conforme é mostrado na Figura 20 e, em seguida, emitirá uma solicitação incluindo o cabeçalho *Authorization* se o acesso for concedido, conforme instrução obtida da (MOZZILA, 2021b).

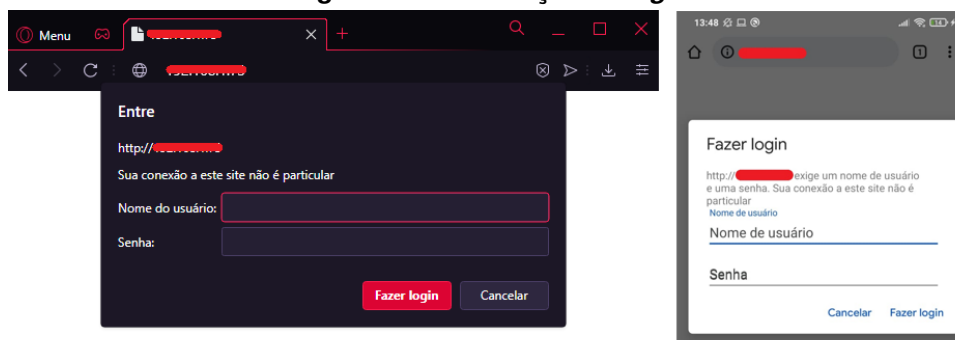
O usuário é cadastrado via algoritmo, nesse projeto, existe apenas um cadastro

Figura 19 – Conteúdo aba desenvolvedor



Fonte: autoria própria (2021).

Figura 20 – Solicitação de login



Fonte: autoria própria (2021).

de *login* e senha permitidos, com tudo, a inclusão de acesso de mais de um usuário é algo simples e possível de se realizar via código. As credenciais do usuário são interpretadas e codificadas e decodificadas a partir da técnica BASE64, como demonstra o código da Figura 21.

Um ponto importante a ressaltar, é que decodificação BASE64 diferencia caracteres maiúsculos e minúsculos, então é importante digitar as credenciais conforme elas foram codificadas. Para este projeto que tem ainda cunho didático, não foi imposto um limite de tentativa de *login* em caso de informações não autorizadas, e qualquer situação diferente, como fechar o *pop-up* de *login*, será tratado pelo algoritmo, conforme Figura 22 e caso o usuário solicite o *logout*, ele será direcionado para mesma área de



**Figura 21 – Código Base64 no algoritmo**

```
//-----
//----- VERIFICAÇÃO DA SENHA (UTILIZANDO BASE64) -----
//-----

if (cabecalho.indexOf("c2F5bW9uOjEyMzQ1Ng==") >= 0) {

    //BASE64 CODE: c2F5bW9uOjEyMzQ1Ng== --> BASE64 DECODE: 'saymon:123456'
    //SE A SENHA FOR CORRETA, O ALGORITMO DÁ CONTINUIDADE.
```

**Fonte: autoria própria (2021).**

tratamento da situação descrita.

**Figura 22 – Cancelamento de *login* ou caso *logout***



**Clique [Aqui](#) ou Atualize a pagina para logar novamente.**

**Fonte: autoria própria (2021).**

### 3.5 BANCO DE DADOS PLANILHAS GOOGLE

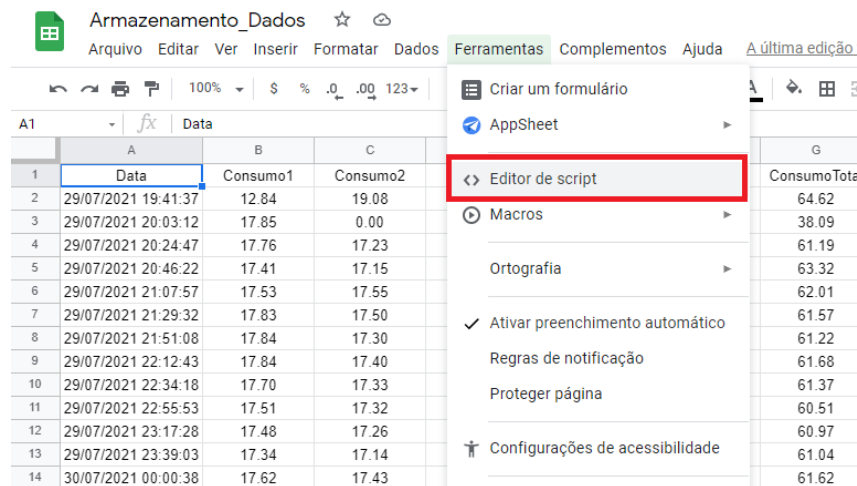
Apesar de muitos usuários preferirem o conhecido Excel da Microsoft®, as planilhas googles são mais vantajosos para o projeto em questão, pois são compatíveis e simples de programar, além de gratuita.

São muito as semelhantes ao Excel, e por serem uma aplicação *online*, é muito mais atraente realizar o armazenamento de dados, com isso não é necessário utilizar a memória do ESP32, o que implicaria em um algoritmo mais complexo. Possuem um recurso de macros, que possibilita a automatização de tarefas, e podem ser programadas em um editor de *Script* conforme demonstra a Figura 23.

Para a coleta de dados, as planilhas devem se comunicar com o microcontrolador, a biblioteca "HTTPClient.h" realiza essa tarefa, pois possuem funções específicas para esse fim, como já explicado no Seção 2.4.2.

Com tudo a biblioteca por si só não é o suficiente, é necessário a adaptação do algoritmo no microcontrolador e no editor de *Script*, essas duas programações são encontradas no portal *github*, mas com funcionalidades diferentes, por isso precisaram ser alteradas para se adequar à aplicação do projeto.

**Figura 23 – Como encontrar o editor de Script Google Sheets**



Fonte: autoria própria (2021).

Não encontrou-se um número limite de informações a serem enviadas, fica a critério do programador definir quais variáveis serão armazenadas, nesse projeto apenas o dia, hora e consumo total são armazenados, pois são as variáveis de intuito de estudo, como exposto na Tabela 2.

**Tabela 2 – Dados armazenados no Google Sheets - Lâmpada 12W**

Data	Corrente (A)	Potência (W)	Tensão (V)
22/08/2021 21:48	0,00	0,00	0,00
22/08/2021 22:48	0,11	13,70	125,9
22/08/2021 23:48	0,10	12,90	125,9
23/08/2021 00:47	0,11	13,57	125,9
23/08/2021 01:47	0,10	12,12	125,9
23/08/2021 02:47	0,10	12,41	125,9
23/08/2021 03:47	0,10	12,66	125,9
23/08/2021 04:46	0,10	12,13	125,9
23/08/2021 05:46	0,10	12,45	125,9
23/08/2021 06:46	0,10	12,58	125,9
23/08/2021 07:46	0,10	12,15	125,9

Fonte: Autoria própria (2021).

O ESP32 envia as variáveis no formato *String* para as planilhas, isso impossibilita a realização de cálculos e gráficos com as mesmas, uma solução simples para esse problema é deixar as células do *Google Sheets*<sup>®</sup> no formato americano, pois, o causador disso é que os números decimais são separados por ponto, e não por vírgula, conforme estabelece a norma brasileira, essa conversão de dados por si só já permite realizar cálculos com os valores de consumo. Quanto a data, nada se altera, já que no código do *Script* ela é declarada no formato dd/MM/aaaa HH:mm.

As possibilidades de tratamento desses dados coletadas são inúmeras, pois as

planilhas *Google* são uma ferramenta de análise, então é possível construir diversos tipos de gráficos, cenários e simulações.

### 3.6 COMPARAÇÃO COM AS TECNOLOGIAS EXISTENTES NO MERCADO

As plataformas *Open Source* possibilitam a criação de diversos algoritmos para diversos fins, como já visto, é possível adaptar um ESP32 como um sistema de controle de energia. Com tudo, já existem algumas tecnologias no mercado com finalidades semelhantes.

Alguns pontos importantes devem ser levados em consideração ao comparar-se tecnologias, como a quantidade de canais que podem ser controlador, a alimentação, e outros fatores considerados nesse projeto. O Quadro 3 contempla informações sobre as capacidades desse projeto e de outros equipamentos.

**Quadro 3 – Comparação com outras tecnologias**

<b>Característica</b>	<b>Projeto</b>	<b>Alexa</b>	<b>Smart Plug</b>	<b>Redgtech</b>
Comando de Voz	Não	Sim	Não	Não
Alimentação	Tomada	Bateria	Tomada	Tomada
Controle de TV	Tomada	Diversos modelos	Único	Tomada
Controle de lâmpada	Sim	Lâmpadas <i>Wi-Fi</i>	Único	Sim
Canais	1 a 64+	*	Único	1 a 32
Monitoramento Energia	Sim	Não	Sim	Não
Armazenamento Dados	Sim	Não	Não	Não

**Fonte: autoria própria (2021).**

O asterisco indicado na coluna da Alexa, significa que a quantidade de canais desse equipamento depende dos acessórios compatíveis, já que para a Alexa controlá-los é necessário uma conexão *Wi-Fi* entre eles. Além disso a Alexa possui uma grande vantagem, que é o seu comando de voz.

O *smart plug*, apesar de ser um acessório independente, pode ser acoplado a uma Alexa, possui um aplicativo próprio de monitoramento, porém sem armazenamento. A grande desvantagem, assim como a Alexa, é que para cada ponto a ser controlado é necessário um *Smart Plug*, tornando-o inviável financeiramente se comparado a outros.

Outro equipamento encontrado, com aplicações semelhantes ao deste projeto, é a placa da Redgtech Automação<sup>®</sup>, uma empresa especializada em domótica. A placa dedicada da Redgtech apresenta vantagens como custo viável, vários canais e controle

com temporizador.

Cada tecnologia possui vantagem em determinada aplicação, algumas possuem tecnologias mais avançadas, outras melhor suporte. Com tudo, o projeto em questão, possui o diferencial em relação aos outros que é o fator de monitoramento de energia e armazenamento de dados em tempo real.

### 3.6.1 Comparação de custos

A mão de obra, matéria prima e os impostos são os principais componentes do valor final de uma mercadoria. Para o projeto em questão, para fins comparativos, considerou-se uma demanda de 32 canais, e a hora do projetista R\$ 30,32 (EAD UNIVALI, 2021), além de não ser considerado o preço do frete em nenhum dos estudos, conforme Tabela 3.

**Tabela 3 – Composição do custo médio do projeto**

Item	Descrição	Quantidade	Preço unitário	Preço total
1	ESP32	01	R\$ 41,75	R\$ 41,75
2	74HC595	05	R\$ 1,30	R\$ 6,48
3	Modulo rele (16)	02	R\$ 104,69	R\$ 209,38
4	Sensor corrente NE	05	R\$ 9,98	R\$ 49,90
5	Fonte 3,3 V	01	R\$ 8,72	R\$ 8,72
6	Fonte 5 V	01	R\$ 8,73	R\$ 8,73
7	Hora Projetista	16	R\$ 30,31	R\$ 485,07
			<b>TOTAL</b>	<b>R\$ 810,00</b>

Fonte: Autoria própria (2021).

Os preços são referentes ao ano de 2021, para esses custos, os acessórios foram cotados pelo *E-commerce* do Mercado Livre®, com tudo, esse custo poderia ser refinado em caso de compras por lote, dessa maneira o custo da mercadoria por unidade tende a diminuir. Esses valores poderiam ser encontrados diferentes em outros *E-commerces*, mas para intuito de estudo, manteve-se o Mercado Livre®. O Quadro 4 apresentará uma comparação de valores (em reais) das tecnologias citadas na Quadro 3, vale ressaltar que os valores já incluem impostos.

Considerando que o algoritmo do projeto já foi desenvolvido, a mão de obra pode ter sua quantidade refinada, mas para isso seriam necessárias algumas aplicações e estudos mais específicos, contudo, em casos comerciais, esse valor pode ser diluído com o valor da margem de lucro. Custos de instalação foram desconsiderados.

**Quadro 4 – Comparação de custos**

<b>Característica</b>	<b>Projeto</b>	<b>Alexa</b>	<b>Smart Plug</b>	<b>Redgtech</b>
Custo	R\$ 810,00	R\$ 569,05	R\$ 99,00	R\$ 1.599,00
Site	Mercado Livre	Amazon	Amazon	Redgtech automação
Data	14/08/21	14/08/21	14/08/21	14/08/21

**Fonte: autoria própria (2021).**

### 3.7 PAYBACK

O *Payback* (retorno de pagamento), é um indicador extremamente utilizado atualmente para cálculo de prazo de retorno de um investimento, pode ser entendido como o tempo de retorno do investimento inicial até o momento no qual o ganho acumulado se iguala ao valor deste investimento, conforme explica (MORAES, 2021).

Para o estudo de *Payback* desse projeto, considerou-se o custo de investimento de R\$ 810,00, uma média de R\$ 200,00 por mês de consumo de energia e 17% de economia nas faturas de energia, 12% conforme citado por Mattede e mais 5% de utilização baseada no monitoramento. Com essas informações o *Payback* simples calculado foi de 24 meses.

O tempo de *Payback* varia com o retorno que o investimento trará, no caso exposto, a média é de R\$ 36,00 por mês. Esse estudo foi realizado de maneira puramente teórica, já que para uma simulação mais precisa, seria necessário a implementação do projeto, com pelo menos 6 meses de uso contínuo.

## 4 RECURSOS E RESULTADOS

### 4.1 RECURSOS

Para o correto planejamento de um projeto, é necessário levar-se em conta os recursos humanos e físicos. Os humanos são aqueles provenientes das pessoas que se dispõem a dedicar-se ao trabalho. Os físicos são os materiais, equipamentos e afins.

Este projeto não contou com nenhum tipo de suporte financeiro, doação de equipamentos ou empréstimos, além disso, devido a situação da pandemia do Coronavírus iniciada no primeiro semestre de 2020, não foi possível a utilização dos laboratórios da UTFPR.

Quanto aos recursos físicos demandados, todos foram providos por investimento próprio, e seguem descritos na Quadro 5.

**Quadro 5 – Componentes utilizados**

Item	Descrição	Quantidade
1	Matriz de contato 400 furos	2
2	Soquete de tomada	2
3	Soquete de lâmpada	2
4	Disjuntores Monofásico	2
5	Interruptor Paralelo	1
6	Barramento de cobre	1
7	Trilho disjuntores	1
8	Fontes de tensão 5 V	2
9	Lâmpada de LED	1
10	Lâmpada fluorescente	1
11	Lâmpada incandescente	1
12	Multímetro	1
13	ESP32	1
14	HMCT103C	5
15	Módulos 8 reles	1
16	74HC595	2
17	Tábua MDF 50x50 cm	1
18	Cabo 2,5 mm (cinza/vermelho)	2
19	Organizador de fios	0.3
20	<i>Jumpers</i>	-
21	<i>Notebook</i>	1

Fonte: autoria própria (2021).

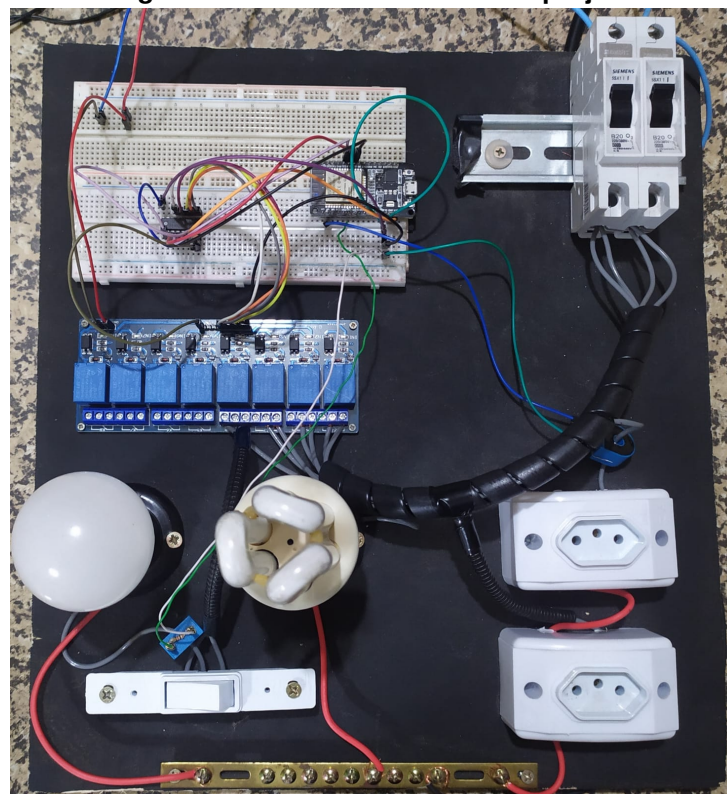
A quantidade dos cabos no item 18 do Quadro 5 está representada em metros, quantidade de *jumpers* não foi contabilizada, apesar de uma quantidade considerável

de *jumpers* machos e fêmeas em utilização. Também não levou-se em conta possíveis despesas de energia e afins no período de testes.

## 4.2 RESULTADOS

Para a análise de resultados desse projeto, improvisou-se uma bancada de MDF com disjuntores, tomadas, receptáculo de lâmpadas e interruptores. O intuito é simular os pontos de energia de uma residência, contudo tomado as devidas proporções, conforme é apresentado na Figura 24. Utilizou-se dois tipos de lâmpadas, a LED e a fluorescente, um carregador de celular e um ferro de solda com potências conhecidas para os testes.

**Figura 24 – Bancada de testes do projeto**



**Fonte: autoria própria (2021).**

Ao utilizar a biblioteca "EmonLib.h", a mesma é responsável por realizar a maioria dos procedimentos matemáticos para a obtenção do valor de corrente eficaz, então, com um multímetro, mede-se o valor da tensão na fonte de alimentação em que a bancada está conectada, e conforme a primeira lei de Ohm e suas derivações, Equação (2), que define a relação proporcional de uma diferença de potencial  $U$ , em

volts, entre dois pontos de resistência intrínseca  $R$ , em ohms, onde percorre uma corrente elétrica  $I$ , em amperes.

$$U = R \times I \quad (2)$$

É por meio desse conceito que o algoritmo escrito no ESP32, calcula o valor de potência medido nos equipamentos acoplados. Todos os valores foram registrados no *Google Sheets*, que armazenou os dados periodicamente. Considerando uma tensão medida de 125,9 V, as Tabela 4 e Tabela 5 foram elaboradas.

**Tabela 4 – Dados Aferidos 1**

<b>Descrição</b>	<b>Lâmpada LED</b>	<b>Lâmpada Fluorescente</b>	<b>Ferro de Solda</b>
Potência Etiqueta (W)	4,7	12	30
Potência Calculada (W)	6,13	12,39	37,2
Erro Percentual (%)	30,43%	3,25%	24,00%

**Fonte: Autoria própria (2021).**

Como na etiqueta do carregador de celular da marca utilizada, não fornecia informações de potência, e somente de corrente, ao contrário da Tabela 4 e Tabela 5, foi construída comparando valores de corrente.

**Tabela 5 – Dados Aferidos 2**

<b>Descrição</b>	<b>Carregador de Celular</b>
Corrente Etiqueta (A)	0,15
Corrente Medida (A)	0,12
Erro Percentual (%)	20%

**Fonte: Autoria própria (2021).**

Todos os testes foram realizado em um período de 72 horas contínuas de funcionamento, alternando manualmente as lâmpadas e a carga da tomada, os valores medidos e calculados são a média do número de medições realizadas, com um intervalo de medição de 60 minutos, com exceção do ferro de solda, que apresenta risco maior se conectado à tomada por muito tempo, este teve intervalo de medição de 30 segundos.

Ao todo o projeto se manteve conectado a energia por sete dias, contudo esse período deve ser dividido em dois. A primeira parte desse tempo durou três dias, foi dentro desse intervalo onde se realizaram a calibração do sensor e os testes dos rele e adjacentes. O segundo foi puramente para aquisição de dados e testes de bancadas, alternando as cargas, incluindo a retirada de todas.



Durante essa fase o projeto não demonstrou sinais de aquecimento, contudo, no teste de falta de energia, os valores de entrada eram perdidos, pois não utilizou-se a memória EEPROM do ESP32 para armazenar esses valores, dessa forma, em casos de falta de energia, o usuário necessitará digitar os valores de tensão medida, tarifa de energia e alarme.

## 5 CONCLUSÕES

Plataformas com capacidade *Open Source*, mostraram-se ser um grande diferencial na elaboração de uma tecnologia nova, o baixo custo no caso do ESP32 juntamente com a plataforma gratuita da Arduino IDE possibilitam a construção de um algoritmo próprio para determinada aplicação, com a possibilidade de inclusão dos mais diversos acessórios disponíveis no mercado. Essa liberdade do *Open Source* pode ser explorada por qualquer entusiasta com conhecimentos de programação.

Como o período de realização desse trabalho foi durante o distanciamento social, de maneira que as universidades se encontravam fechadas para aos alunos, o grande desafio foi realizar a montagem e testes sem o auxílio de um laboratório capacitado, pois instrumentos como osciloscópio, amperímetros e afins, são de custos elevados, e sem o suporte financeiro, não foi possível realizar testes mais precisos no projeto.

Pelo método indicativo de viabilidade econômico do Valor presente líquido (VPL), ao considerar-se a Taxa mínima de atratividade (TMA) o mesmo valor da taxa Selic de 5,25% ao ano (ou 0,43% ao mês) (MELHOR CAMBIO, 2021), obtêm-se uma Taxa interna de retorno (TIR) de 2%. A TIR é a taxa que zera o VPL, em outras palavras, é a taxa que faz com que todas as saídas sejam equivalentes às entradas de recursos, sempre trazidas a valor presente, na teoria, se a TIR for maior que a TMA, significa que o investimento é viável, conforme descrito em (ALVEZ, 2019). Provando, que o investimento na tecnologia abordada nesse projeto, é viável, considerando um prazo de 24 meses.

Mesmo que existam tecnologias semelhantes no mercado, nenhuma apresenta por si só a mesma capacidade exposta desse projeto. Controle e monitoramento com armazenamento de dados e sistema de notificação ao usuário, são, excelentes métodos de simplificação da redução de custo com energia no dia a dia de uma residência, comércio local, ou até uma indústria, tendo em conta o aumento considerável do custo das tarifas elétricas.

Além desses fatores financeiros e técnicos, este trabalho pode apresentar grandes soluções nos quesitos de acessibilidade física, proporcionando melhores condições às pessoas Portadoras de necessidades especiais (PNE), que muitas vezes não possuem condições igualitárias para usufruir das tecnologias mais simplistas do

mercado.

## 5.1 TRABALHOS FUTUROS

Esse projeto proporcionou conhecimento extra em diversas áreas não abordadas no curso de Engenharia Elétrica e apesar dos resultados obtidos terem sido considerados satisfatórios, algumas considerações precisam ser levadas em conta.

Como descrito no decorrer do trabalho, o *Web Server* é do tipo estático, e mesmo que funcional, não é a melhor opção para esse tipo de aplicação, pois o algoritmo necessita de uma iteração para atualizar os valores, e não é permitido o acesso simultâneo de vários usuários na *web page*. O ideal, seria a implementação da biblioteca "ESPAsyncWebServer.h", que possibilita a utilização de um *web server* dinâmico, como tudo isso requer maior conhecimento em programação JavaScript e HTML, bem como, o desenvolvimento de um aplicativo para dispositivos móveis, com compatibilidade IOS e Android.

Outras características que ficaram pendentes nesse projeto, são a proteção física e a placa de circuito impressa. Essas duas características são importantes caso o trabalho avançasse para a etapa de comercialização, uma caixa plástica para proteção de poeira e fatores físicos e a PCI contemplando todos os acessórios necessários.

Em comparação com equipamentos disponíveis comercialmente, tal como a Alexa que possui uma grande vantagem, que é o seu comando de voz, porém, apesar do cenário atual desse projeto não contemplar essa tecnologia, existem vários dispositivos que podem ser acoplados e programados juntamente ao ESP32 que possibilitam o controle via comando de voz.

## REFERÊNCIAS

ALBUQUERQUE, YURE. **ESP32 pinout – Guia Básico de GPIOs**. [S. l.: s. n.], 2021. Disponível em:

<https://blog.smartkits.com.br/esp32-pinout-guia-basico-de-gpios/>. Acesso em: 10 abr. 2021.

ALVEZ, V. **O que é TIR e VPL? Saiba como utilizar nos seus investimentos**.

[S. l.: s. n.], 2019. Disponível em:

<https://maisretorno.com/portal/o-que-e-tir-e-vpl>. Acesso em: 18 fev. 2019.

ANEEL. **Grupo B**. [S. l.: s. n.], 2021. Disponível em: [https://www.aneel.gov.br/home?p\\_p\\_id=101&p\\_p\\_lifecycle=0&p\\_p\\_state=maximized&p\\_p\\_mode=view&\\_101\\_struts\\_action=%5C%2Fasset\\_publisher%5C%2Fview\\_content&\\_101\\_returnToFullPageURL=http%5C%3A%5C%2F%5C%2Fwww.aneel.gov.br%5C%2Fhome%5C%3Fp\\_auth%5C%3DiwZkkstu%5C%26p\\_p\\_id%5C%3D3%5C%26p\\_p\\_lifecycle%5C%3D1%5C%26p\\_p\\_state%5C%3Dnormal%5C%26p\\_p\\_state\\_rcv%5C%3D1&\\_101\\_assetEntryId=15049535&\\_101\\_type=content&\\_101\\_groupId=656835&\\_101\\_urlTitle=grupo-b&inheritRedirect=true](https://www.aneel.gov.br/home?p_p_id=101&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&_101_struts_action=%5C%2Fasset_publisher%5C%2Fview_content&_101_returnToFullPageURL=http%5C%3A%5C%2F%5C%2Fwww.aneel.gov.br%5C%2Fhome%5C%3Fp_auth%5C%3DiwZkkstu%5C%26p_p_id%5C%3D3%5C%26p_p_lifecycle%5C%3D1%5C%26p_p_state%5C%3Dnormal%5C%26p_p_state_rcv%5C%3D1&_101_assetEntryId=15049535&_101_type=content&_101_groupId=656835&_101_urlTitle=grupo-b&inheritRedirect=true). Acesso em: 10 abr. 2021.

ARDUINO. **ShiftRegister74HC595**. [S. l.: s. n.], 2021. Disponível em:

<https://www.arduino.cc/reference/en/libraries/shiftregister74hc595/>. Acesso em: 10 abr. 2021.

ARDUINO. **Web Server**. [S. l.: s. n.], 2021. Disponível em:

<https://www.arduino.cc/en/Tutorial/LibraryExamples/WebServer>. Acesso em: 10 abr. 2021.

BARROS, B. F. d. **Geração, transmissão, distribuição e consumo de energia elétrica**. [S. l.]: Erica, 2019. (Eixos). ISBN 9788536508207. Disponível em:

<http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000015179&lang=pt-br&site=eds-live&scope=site>.

DENK, TIMO. **ShiftRegister74HC595 Arduino Library**. [S. l.: s. n.], 2019. Disponível em: <https://timodenk.com/blog/shift-register-arduino-library/>. Acesso em: 30 set. 2019.

EAD UNIVALI. **Engenheiro eletricista: salário, o que faz e mercado de trabalho**.

[S. l.: s. n.], 2021. Disponível em:

<https://ead.univali.br/blog/engenheiro-eletricista-salario>. Acesso em: 8 ago. 2021.

ESPRESSIF SYSTEMS. **ESP32 Datasheet**. [S. l.: s. n.], 2021. Disponível em:

<https://html.alldatasheet.com/html-pdf/1148023/ESPRESSIF/ESP32/6845/12/ESP32.html>. Acesso em: 10 abr. 2021.

ETC DATASHEET. **Datasheet Sensor ZMCT103C**. [S. l.: s. n.], 2021. Disponível em: <https://datasheetspdf.com/pdf/1408474/ETC/ZMCT103C/1>. Acesso em: 10 abr. 2021.

FANTINATO, G. **Por que o Brasil é tão propenso a ataques hacker?** [S. l.: s. n.], 2021. Disponível em: <https://www.tecmundo.com.br/seguranca/211787-brasil-tao-propenso-ataques-hacker.htm>. Acesso em: 10 abr. 2021.

FINLEY, K. **What Exactly Is GitHub Anyway?** [S. l.: s. n.], 2021. Disponível em: <https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>. Acesso em: 14 jun. 2012.

G1 GLOBO. **Energia em alta: veja perguntas e respostas sobre o sistema de bandeiras e o peso na conta de luz**. [S. l.: s. n.], 2021. Disponível em: <https://g1.globo.com/economia/noticia/2021/07/08/energia-em-alta-veja-perguntas-e-respostas-sobre-o-sistema-de-bandeiras-e-o-peso-na-conta-de-luz.ghtml>. Acesso em: 7 ago. 2021.

HUBBARD, E. **Epígrafe**. [S. l.: s. n.], 1998. Disponível em: [https://pt.wikiquote.org/wiki/Elbert\\_Hubbard#:~:text=%22A%5C%20m%C3%A1quina%5C%20pode%5C%20fazer%5C%20a, trabalho%5C%20de%5C%20um%5C%20homem%5C%20extraordin%C3%A1rio.%22](https://pt.wikiquote.org/wiki/Elbert_Hubbard#:~:text=%22A%5C%20m%C3%A1quina%5C%20pode%5C%20fazer%5C%20a, trabalho%5C%20de%5C%20um%5C%20homem%5C%20extraordin%C3%A1rio.%22). Acesso em: 10 abr. 2021.

JOSEFSSON, S. **The Base16, Base32, and Base64 Data Encodings**. [S. l.: s. n.], 2006. Disponível em: [https://www.hjp.at/doc/rfc/rfc4648.html#sec\\_4](https://www.hjp.at/doc/rfc/rfc4648.html#sec_4). Acesso em: 10 abr. 2021.

JUNIOR, S. L. S. **Domótica : automação residencial e casas inteligentes com Arduino e ESP8266**. [S. l.]: Erica, 2018. ISBN 9788536528120. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000016403&lang=pt-br&site=eds-live&scope=site>.

LENZ, M. L. **Microprocessadores**. [S. l.]: SAGAH, 2019. ISBN 9788595029736. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000015743&lang=pt-br&site=eds-live&scope=site>.

MANZANO, J. A. N. G. **Programação de computadores com C/C++**. [S. l.]: Erica, 2014. ISBN 9788536507620. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.00009366&lang=pt-br&site=eds-live&scope=site>.

MATTEDE, H. **Aparelhos em stand-by, gasto desnecessário?** [S. l.: s. n.], 2021. Disponível em: <https://www.mundodaeletrica.com.br/aparelhos-em-stand-by-gasto-desnecessario/>. Acesso em: 8 ago. 2021.

MCEWEN, A. **HTTPClient**. [S. l.: s. n.], 2021. Disponível em: <https://www.arduino.cc/reference/en/libraries/httpclient/>. Acesso em: 10 abr. 2021.

MELHOR CAMBIO. **Taxa Selic Hoje**. [S. l.: s. n.], 2021. Disponível em: <https://www.melhorcambio.com/taxa-selic>. Acesso em: 8 ago. 2021.

MELO, A. C. V. d. **Princípios de linguagem de programação**. [S. l.]: Blucher, 2003. ISBN 9788521203223. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000014996&lang=pt-br&site=eds-live&scope=site>.

MILETTO, E. M. **Desenvolvimento de software II: introdução ao desenvolvimento web com html, css, javascript e php**. [S. l.]: Bookman, 2014. ISBN 9788582601969. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000005390&lang=pt-br&site=eds-live&scope=site>.

MOHAMED, F.; ALI AL DAHOUD, A. Z. **Integrated Development Environment “IDE” For Arduino**. [S. l.: s. n.], 2018. Disponível em: [https://www.researchgate.net/profile/Mohamed-Fezari-2/publication/328615543\\_Integrated\\_Development\\_Environment\\_IDE\\_For\\_Arduino/links/5bd8c6d24585150b2b9206df/Integrated-Development-Environment-IDE-For-Arduino.pdf](https://www.researchgate.net/profile/Mohamed-Fezari-2/publication/328615543_Integrated_Development_Environment_IDE_For_Arduino/links/5bd8c6d24585150b2b9206df/Integrated-Development-Environment-IDE-For-Arduino.pdf). Acesso em: 10 abr. 2021.

MORAES, I. D. **O que é e como Calcular o Payback?** [S. l.: s. n.], 2021. Disponível em: <https://www.contabeis.com.br/noticias/30249/o-que-e-e-como-calcular-o-payback/>. Acesso em: 8 ago. 2021.

MORAIS, I. S. d. *et al.* **Introdução a big data e internet das coisas (IOT)**. [S. l.]: SAGAH, 2018. ISBN 9788595027640. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000014660&lang=pt-br&site=eds-live&scope=site>.

MOZZILA. **O que é um servidor web (web server)?** [S. l.: s. n.], 2021. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/What_is_a_web_server). Acesso em: 10 abr. 2021.

MOZZILA. **O que é um servidor web (web server)?** [S. l.: s. n.], 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Authentication>. Acesso em: 10 abr. 2021.

NXP SEMICONDUCTORS. **74HC595 Datasheet**. [S. l.: s. n.], 2021. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/15644/PHILIPS/74HC595.html>. Acesso em: 10 abr. 2021.

PINHEIRO, P. P. **Segurança digital : proteção de dados nas empresas**. [S. l.]: Atlas, 2020. ISBN 9788597026054. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000020636&lang=pt-br&site=eds-live&scope=site>.

PORTAL DA INDÚSTRIA. **Cibersegurança: o que é, importância e como se qualificar**. [S. l.: s. n.], 2021. Disponível em:

<https://www.portaldaindustria.com.br/industria-de-a-z/ciberseguranca/>. Acesso em: 10 abr. 2021.

ROSSI, G. P. **MÓDULOS DE HARDWARE PARA APLICAÇÃO EM ARQUITETURAS DESCENTRALIZADAS DE AUTOMAÇÃO RESIDENCIAL**. [S. l.: s. n.], 2018.

SANTOS, J. W.; LARA JUNIOR, R. C. d. **Sistema De Automação Residencial De Baixo Custo Controlado Pelo Microcontrolador Esp32 E Monitorado Via Smartphone**. [S. l.: s. n.], 2019.

SHIMABUKURO, I. **Investimentos em IA no Brasil podem chegar a R\$ 2,4 bilhões em 2021**. [S. l.: s. n.], 2021. Disponível em: <https://olhardigital.com.br/2021/02/05/noticias/investimentos-em-ia-no-brasil-podem-chegar-a-r-24-bilhoes-em-2021/>. Acesso em: 10 abr. 2021.

SIEMENS. **List of Achilles certified products - Industrial Security**. [S. l.: s. n.], 2021. Disponível em: <https://assets.new.siemens.com/siemens/assets/api/uuid:713a33b1e842eb8fb55d0c355d5ce22f1b2aad41/7838-07-siemens-achilles-certified-products.pdf>. Acesso em: 10 abr. 2021.

SILVA, C. B. S. d.; NARDOTO, D. R.; JORGE, M. d. M. **Desenvolvimento de um Sistema Modular Microcontrolado de Automação Residencial Via WI-FI**. [S. l.: s. n.], 2016.

TERUEL, E. C. **HTML 5 : guia prático**. [S. l.]: Erica, 2014. ISBN 9788536506067. Disponível em: <http://search.ebscohost.com/login.aspx?direct=true&db=edsmb&AN=edsmb.000009348&lang=pt-br&site=eds-live&scope=site>.