

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**GUSTAVO STRONCZEK BELO**

**SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE SMART HOME**

**GUARAPUAVA**

**2023**

**GUSTAVO STRONCZEK BELO**

**SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE SMART HOME**

**IOT SYSTEM FOR SMART HOME MONITORING AND CONTROL**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Mecânica do Curso de Bacharelado em Engenharia Mecânica da Universidade Tecnológica Federal do Paraná.  
Orientador(a): Prof. Dr. Vlademir Aparecido Freire Junior

**GUARAPUAVA**

**2023**



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**GUSTAVO STRONCZEK BELO**

**SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE SMART HOME**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Mecânica do Curso de  
Bacharelado em Engenharia Mecânica da  
Universidade Tecnológica Federal do Paraná.

Data de aprovação: 05/dezembro/2023

---

Vlademir Aparecido Freire Junior  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Henrique Ajuz Holzmann  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Hilson Henrique Daum  
Doutorado  
Universidade Tecnológica Federal do Paraná

**GUARAPUAVA**

**2023**

## RESUMO

Nos últimos anos as abordagens que consideram o conceito de Internet das Coisas têm se tornado cada vez mais frequentes em diversos estudos, devido à sua crescente aplicação em diversas áreas nos dias atuais. A Internet das Coisas (IoT) é amplamente empregada em setores como indústria, logística, saúde, agricultura, segurança e, cada vez mais, em residências. Devido ao fato de utilizarem dispositivos eletrônicos de custo relativamente baixo para a automação e a maior acessibilidade à internet, torna-se cada vez mais fácil implementar sistemas de automação considerando o IoT, proporcionando maior comodidade aos usuários e, por vezes, acessibilidade para pessoas com necessidades especiais ou deficiências. Assim, esse conceito adquire uma grande importância para a população em geral. O presente trabalho propõe o desenvolvimento de um sistema IoT capaz de monitorar o consumo energético e controlar equipamentos residenciais, onde, através do uso da placa ESP8266 ESP-12E V3 com o *firmware* Tasmota instalado, em conjunto com uma conexão MQTT a um aplicativo, é possível realizar o controle de maneira remota das funções liga/desliga, além de monitorar o consumo de energia de equipamentos de maneira não invasiva, propiciando assim, um melhor gerenciamento da operação e consumo dos dispositivos residenciais do usuário. Após testes, o sistema desenvolvido obteve resultados satisfatórios, garantindo a confiabilidade do controle e dos dados de monitoramento exibidos no aplicativo.

**Palavras-chave:** Internet das Coisas; casa inteligente; MQTT; ESP8266; gerenciamento residencial.

## ABSTRACT

In recent years, approaches considering the concept of the Internet of Things (IoT) have become increasingly prevalent in various studies, owing to its growing application across different fields today. The Internet of Things (IoT) is widely employed across industries such as manufacturing, logistics, healthcare, agriculture, security, and increasingly in households. Due to the use of relatively low-cost electronic devices for automation and increased internet accessibility, implementing automation systems considering IoT has become progressively more feasible, offering greater convenience to users and sometimes accessibility for individuals with special needs or disabilities. Consequently, this concept holds significant importance for the general population. This study proposes the development of an IoT system capable of monitoring energy consumption and controlling residential appliances. Through the utilization of the ESP8266 ESP-12E V3 board with installed Tasmota firmware, coupled with an MQTT connection to an application, remote control of on/off functions is achievable. Additionally, it enables the non-invasive monitoring of energy consumption by appliances, facilitating improved management of the operation and usage of user's residential devices. Following tests, the developed system yielded satisfactory results, ensuring the reliability of both control and monitoring data displayed within the application.

**Keywords:** Internet of Things; smart home; MQTT; ESP8266; residential management.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Conexão de aparelhos à Internet.....	16
Figura 2 - Automação no ambiente residencial.....	18
Figura 3 - Microcontrolador ATmega328P DIP-28.....	19
Figura 4 - Arduino Uno.....	19
Figura 5 - ESP8266 ESP-12E V3.....	20
Figura 6 - Raspberry Pi 4.....	21
Figura 7 - Publicação de mensagens no MQTT Broker.....	23
Figura 8 - Interruptor Sonoff Basic R2.....	25
Figura 9 - Interface do Tasmota.....	26
Figura 10 - Diagrama esquemático.....	27
Figura 11 - Montagem do sistema.....	28
Figura 12 - Página inicial do HiveMQ.....	29
Figura 13 - Painel de uso do HiveMQ.....	29
Figura 14 - Criação de credenciais do HiveMQ.....	30
Figura 15 - Informações de conexão do HiveMQ.....	30
Figura 16 - Página inicial do GitHub.....	31
Figura 17 - Terminal do Gipod.....	32
Figura 18 - Terceira etapa para a compilação do Tasmota.....	33
Figura 19 - Página de <i>download</i> do Tasmotizer 1.2.....	34
Figura 20 - Interface do Tasmotizer 1.2.....	35
Figura 21 - IP de acesso ao Tasmota.....	35
Figura 22 - Interface inicial do Tasmota .....	36
Figura 23 - Aba de Configurações.....	37
Figura 24 - Aba de configurações do módulo.....	37
Figura 25 - Configuração da saída digital e entrada analógica.....	38
Figura 26 - Interface pré-configurada.....	39
Figura 27 - Temporizador.....	39
Figura 28 - Configuração da aba MQTT no Tasmota.....	40
Figura 29 - Confirmação de conexão com o broker MQTT.....	41
Figura 30 - Pré-calibração do sensor de corrente.....	42
Figura 31 - Aplicativo utilizado.....	43
Figura 32 - Configuração do MQTT no aplicativo.....	44
Figura 33 - Configuração do <i>switch</i> do relé.....	45
Quadro 1 - Inputs para os painéis.....	46
Figura 34 - Painel do aplicativo.....	50
Figura 35 - Bancada de testes.....	51
Figura 36 - Monitoramento através do aplicativo.....	52
Figura 37 - Alicates amperímetro.....	53

<b>Figura 38 - Comparação da corrente aferida.....</b>	<b>53</b>
<b>Figura 39 - Validação do sistema.....</b>	<b>54</b>

## LISTA DE TABELAS

Tabela 1 - Lista de componentes.....	27
--------------------------------------	----

## LISTA DE ABREVIATURAS E SIGLAS

MIT	<i>Massachusetts Institute of Technology</i>
IoT	<i>Internet of Things</i>
UIT	União Internacional de Telecomunicações
ABINC	Associação Brasileira de Internet das Coisas
EIBA	<i>European Installation Bus Association</i>
NAHB	<i>National Association of Home Builders</i>
AURESIDE	Associação Brasileira de Automação Residencial
SI	Sistema integrado
MQTT	<i>Message Queuing Telemetry Transport</i>
QoS	<i>Quality of Service</i>
TLS	<i>Transport Layer Security</i>
S.I.	<i>Sine loco</i> (sem local)
IA	Inteligência artificial
PCI	Placa de circuito impresso

## LISTA DE SÍMBOLOS

P	Potência	[W]
V	Tensão	[V]
I	Corrente	[A]
E	Consumo	[kWh]
P <sub>m</sub>	Potência média	[W]
ΔT	Tempo de medição	[h]

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos.....</b>	<b>13</b>
1.1.1	Objetivos Gerais.....	13
1.1.2	Objetivos Específicos.....	13
<b>1.2</b>	<b>Justificativa.....</b>	<b>14</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>15</b>
<b>2.1</b>	<b>Internet das Coisas.....</b>	<b>15</b>
<b>2.2</b>	<b>Smart Home ou Domótica.....</b>	<b>17</b>
<b>2.3</b>	<b>Microcontroladores e microprocessadores.....</b>	<b>18</b>
2.3.1	Arduino.....	19
2.3.2	ESP.....	20
2.3.3	Raspberry Pi.....	21
2.3.4	Definição do dispositivo.....	22
<b>2.4</b>	<b>Protocolo de comunicação IoT.....</b>	<b>22</b>
<b>2.5</b>	<b>Tasmota.....</b>	<b>24</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS.....</b>	<b>27</b>
<b>3.1</b>	<b>Componentes eletrônicos e montagem.....</b>	<b>27</b>
<b>3.2</b>	<b>Configuração do MQTT.....</b>	<b>28</b>
<b>3.3</b>	<b>Instalação do Tasmota.....</b>	<b>31</b>
3.3.1	Compilação firmware.....	31
3.3.2	Instalação com o Tasmotizer.....	33
<b>3.4</b>	<b>Configuração do Tasmota.....</b>	<b>36</b>
3.4.1	Configuração das entradas e saídas.....	36
3.4.2	Configuração dos temporizadores.....	39
3.4.3	Configuração do MQTT.....	40
<b>3.5</b>	<b>Aplicativo MQTT.....</b>	<b>42</b>
<b>4</b>	<b>RESULTADOS.....</b>	<b>51</b>
<b>4.1</b>	<b>Resultados experimentais.....</b>	<b>51</b>
<b>4.2</b>	<b>Validação.....</b>	<b>53</b>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>55</b>
<b>5.1</b>	<b>Trabalhos futuros.....</b>	<b>55</b>
	<b>REFERÊNCIAS.....</b>	<b>57</b>

## 1 INTRODUÇÃO

Com a criação da Internet em 1969 pelos Estados Unidos durante a Guerra Fria, a disponibilidade aos civis em 1991 e desde então a crescente acessibilidade da mesma ao redor do mundo, a população sente cada vez mais a necessidade de estar conectada a informações de forma prática e rápida. Antes do advento da Internet, a comunicação a longas distâncias e a disseminação de notícias sobre acontecimentos ao redor do mundo eram processos extremamente demorados, podendo levar meses até que as pessoas tivessem conhecimento desses eventos. Hoje em dia é muito fácil se comunicar com pessoas fisicamente distantes, levando segundos para uma mensagem chegar ao outro lado do mundo, e até mesmo notícias podem ser acompanhadas praticamente em tempo real, o que há um século atrás seria totalmente inviável.

Com o aumento do uso da Internet também surgiu o interesse de conectar aparelhos comuns com uma conexão sem fio para monitoramento e controle, o que possibilita a exibição de informações relevantes sobre determinada atividade ou até mesmo o controle desses aparelhos à distância, trazendo grande comodidade ao usuário. Devido a isso, de acordo com Morais et al. (2018), em 1999 foi introduzido pelo pesquisador britânico Kevin Ashton do *Massachusetts Institute of Technology* (MIT) o termo Internet das Coisas, do inglês, *Internet of Things* (IoT). Um exemplo de aplicação do IoT que pode ser citado é um *smartwatch* durante um treino de corrida, que conectado a Internet pode registrar todos os dados durante o treino e depois mostrá-los em um *smartphone*, desde gráficos exibindo os batimentos cardíacos, ritmo de treino, elevação, calorias gastas, velocidade, entre outros dados relevantes.

O IoT atualmente é utilizado em diversas áreas, como na indústria, logística, saúde, agricultura, segurança, casas inteligentes, do inglês, *smart homes* e em maiores escalas em cidades inteligentes, do inglês, *smart cities*. Essa tecnologia combinada com a inteligência artificial (IA) tem como vantagem a otimização de tarefas de modo a poupar tempo ao usuário e auxiliar na tomada de decisões mais assertivas durante suas atividades. O IoT apresenta como principal vertente o monitoramento que possibilita a redução do tempo para acesso às informações que estão sendo analisadas. Sem o IoT seria difícil gerenciar vários dados sem estar

presencialmente no local, assim seria necessário o deslocamento para aferir os dados ou até solicitar a outra pessoa verificar e repassar os dados via Internet.

Junior e Farinelli (2019), evidenciam que devido ao crescente desenvolvimento do IoT, a automação residencial se tornou mais ampla, no entanto, a ideia de automação residencial não é recente, seu início vem da década de 1980, com a automação de tarefas simples como, por exemplo, abrir e fechar o portão com controle remoto e automatização da iluminação, já com o propósito de trazer mais comodidade ao usuário.

De acordo com Moraes et al. (2018), pode ser visto que ao redor do mundo o acesso à Internet está cada vez mais difundido, facilitando assim a implementação de sistemas IoT. Segundo pesquisas realizadas por Magrani (2018), nas últimas décadas, bilhões de pessoas se conectaram ao mundo digital. Dados recentes da União Internacional de Telecomunicações (UIT) mostram que 95% da população global já vive em áreas cobertas com rede celular (2G ou mais) e 84% têm acesso à banda larga móvel.

A segurança residencial também se beneficiou com o surgimento da IoT. Atualmente, existem câmeras de segurança que oferecem a possibilidade de conexão e acesso remoto, viabilizando o monitoramento em tempo real do ambiente residencial. Além disso, é possível integrar sensores de movimento, portas e janelas aos sistemas de alarme, possibilitando o envio de notificações para dispositivos móveis diante de qualquer atividade suspeita. Essas são apenas algumas das muitas estratégias disponíveis para aprimorar a segurança residencial. Dessa forma, a automação não se limita apenas a controlar as funções básicas, mas também se expande para fornecer um gerenciamento completo dos equipamentos e sistemas de segurança residencial.

Junior e Farinelli (2019), destacam ainda que, além do uso em segurança e automação residencial, a tecnologia de IoT também pode ser aplicada para cuidados com a saúde e bem-estar de idosos, crianças e animais de estimação. Para os idosos, por exemplo, é possível desenvolver sistemas que monitorem a temperatura corporal, a frequência cardíaca e a detecção de quedas, proporcionando um ambiente mais seguro e confortável para a terceira idade. Já para as crianças, os sistemas de monitoramento do sono, que utilizam câmeras e microfones, são muito populares atualmente. E para os animais de estimação, os sistemas de controle de alimentação remoto são bastante utilizados, permitindo que seus donos possam ter certeza de que

seus *pets* estão sendo alimentados de forma adequada, mesmo quando não estão em casa.

Como pode ser visto, sistemas IoT possuem inúmeros benefícios independente da área de aplicação. Diante disso, este trabalho visa contribuir no desenvolvimento de um sistema de monitoramento e controle de dispositivos residenciais de baixo custo, com isso, pretende-se proporcionar maior comodidade e acessibilidade para os usuários em suas atividades residenciais.

## **1.1 Objetivos**

Esta seção tem o intuito de apresentar os objetivos gerais e os objetivos específicos que são almejados a serem alcançados durante o desenvolvimento deste trabalho.

### **1.1.1 Objetivos Gerais**

Este trabalho tem como objetivo desenvolver um sistema IoT para monitoramento e controle de dispositivos residenciais. Por meio da combinação de um sistema desenvolvido com um aplicativo de acesso gratuito, disponível nas principais lojas de aplicativos, com o intuito de monitorar o consumo de energia de dispositivos e controlar suas operações remotamente.

### **1.1.2 Objetivos Específicos**

- Estudar as características de sistemas IoT e seus protocolos de comunicação;
- Verificar os componentes necessários para o desenvolvimento do sistema, junto à análise de conceitos para dimensionamentos, montagens e implementações;
- Executar a montagem do sistema de monitoramento e controle;
- Estabelecer conexão entre o sistema de monitoramento e controle por meio de um smartphone;
- Validação do sistema de monitoramento e controle com resultados experimentais.

## 1.2 Justificativa

O avanço tecnológico vem mudando a forma como as pessoas vivem e se relacionam com o ambiente em que estão inseridas. Um dos segmentos que tem sido impactado por essa transformação é o setor residencial, com o surgimento das chamadas *smart homes*. Nesse contexto, a utilização de aparelhos conectados à Internet vem crescendo de forma significativa, oferecendo inúmeras vantagens para seus usuários, como o controle e automação de diversas atividades cotidianas, além da coleta de dados relevantes para aprimoramento dessas funções.

Assim, este trabalho é desenvolvido justamente para explorar as possibilidades oferecidas pelas casas inteligentes, destacando sua capacidade de aumentar a comodidade do usuário e tornar acessível o uso de aparelhos que seriam difíceis ou até inviáveis de serem utilizados por pessoas com deficiência ou mobilidade reduzida.

Todo o trabalho visa o desenvolvimento de um sistema de baixo custo, conectado a um aplicativo gratuito, que fará com que os usuários tenham acesso às funcionalidades e controles de equipamentos residenciais a partir de seus dispositivos móveis, como *smartphones* e/ou *tablets*, que tenham conexão com a internet.

Através de um sistema de monitoramento e controle é possível traçar estratégias que possibilitam a redução do gasto com energia elétrica, trazendo economia financeira e aumento da vida útil dos aparelhos. Um controle energético adequado reduz os impactos ambientais relacionados à sua fonte de energia, principalmente a respeito da geração de energia de forma não renovável, como de usinas termelétricas e termonucleares, que causam impactos ambientais irreparáveis. Dessa forma, o presente trabalho se justifica pela importância do tema, que tem despertado cada vez mais interesse em função dos benefícios que podem ser oferecidos à sociedade e até mesmo ao meio ambiente.

## 2 REFERENCIAL TEÓRICO

Nesta seção, serão abordados os principais temas relevantes para o desenvolvimento do trabalho.

### 2.1 Internet das Coisas

De acordo com Costa (2019), o termo Internet das Coisas, do inglês, *Internet of Things* (IoT), frequentemente é utilizado de forma genérica a objetos que vêm se conectando à Internet com determinadas capacidades de processamento, tornando-as “inteligentes” para executar determinadas funções nas casas, ambientes de trabalho, cidades e indústrias.

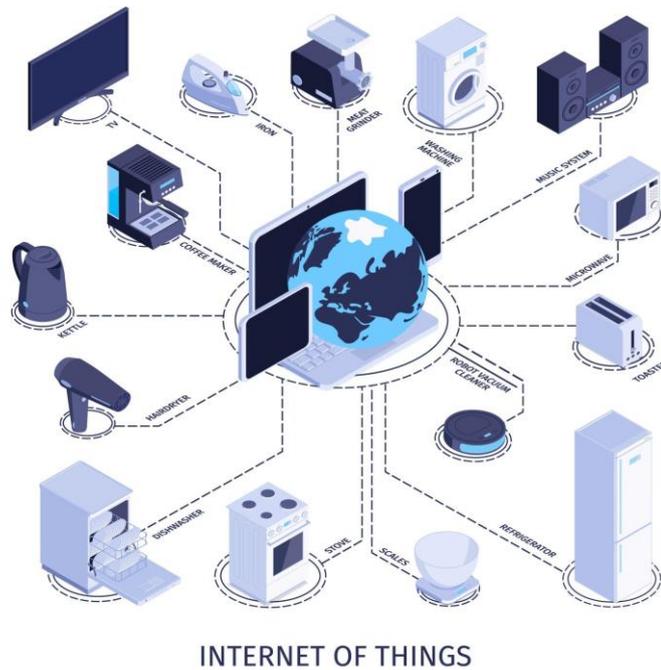
Rosner e Kenneally (2018), durante sua definição de IoT, diz que o termo se refere a dispositivos, muitas vezes desempenhando funções diferentes das que usualmente executam, têm a capacidade de sentir, acumular e analisar dados e se comunicar através de redes.

Ashton (2017), o criador do termo *Internet of Things*, destaca que o termo não se refere apenas a conectar objetos do cotidiano à Internet. Ativar uma torradeira pela Internet tem pouco valor: você ainda precisa ir até lá e colocar o pão. O que define a Internet das coisas é a captura de dados. No início do século XX os computadores obtinham seus dados geralmente por pessoas através de teclados, sendo uma grande limitação, porque pessoas conseguem inserir manualmente apenas dados relativamente simples, e como o mundo é um sistema complexo, a entrada de humanos restringiu muito os primeiros computadores.

Embora a solução de deixar os computadores sentirem o mundo por si mesmo fosse evidente, não era de fácil implementação. A detecção do computador até o século XXI não conseguiu realizar todo o seu potencial pois os sensores trabalham melhor em uma rede. Ashton (2017), ainda faz uma analogia relacionando o corpo humano, através do sistema nervoso, onde o cérebro combina informações com memórias e as interpreta ou dá sentido a elas. Uma pessoa pode ver uma torta, sentir o cheiro de maçã, tocar e sentir sua temperatura, assim, ao fazer a correlação de todas essas entradas é possível identificar uma torta de maçã quente ou fria, mostrando, dessa forma, que nossos olhos, nariz, ouvido, assim por diante, tem valor limitado, a menos que estejam conectados. Por fim Ashton (2017, p. 9) conclui:

Este é o significado da Internet das Coisas: sensores conectados à Internet, comportando-se de forma semelhante à Internet, fazendo conexões abertas e ad hoc, compartilhando dados livremente e permitindo aplicações inesperadas, para que os computadores possam compreender o mundo ao seu redor e se tornar o sistema nervoso da humanidade.

**Figura 1 - Conexão de aparelhos à Internet**



**Fonte: Freepik (2021)**

É notável o crescimento do IoT ao redor do mundo. De acordo com um estudo desenvolvido pela TGT Consult e a Associação Brasileira de Internet das Coisas (ABINC), até 2025, mais de 27 bilhões de dispositivos estarão conectados à Internet. Segundo o autor do estudo e analista da TGT Consult/ISG, David de Paulo Pereira, desde a divulgação do Plano Nacional de IoT, houve um notável avanço e amadurecimento do mercado de serviços relacionados à Internet das Coisas.

Para o analista, o avanço das tecnologias de IoT no último ano foi impulsionado por fatores cruciais, como o crescente entendimento dos benefícios dessa tecnologia e o amadurecimento do mercado, juntamente com a aceleração da transformação digital causada pela pandemia, que aumentou o uso de dispositivos conectados.

## 2.2 *Smart Home* ou Domótica

Ambos os termos se referem a automação residencial, permitindo que os dispositivos eletrônicos realizem tarefas de forma autônoma ou sejam controlados remotamente, sem a necessidade de se deslocar fisicamente até o dispositivo para comandá-lo.

Nesse contexto, os sistemas domóticos podem incluir, mas sem se restringir, o controle centralizado ou não de segurança, iluminação, climatização, audiovisuais, comunicação de voz, e mesmo de telefonia, além de outros sistemas que possam proporcionar a melhoria de fatores cotidianos como comodidade, conforto, eficiência energética e segurança. (Junior e Farinelli, 2019, p.16)

Junior e Farinelli (2019) salientam que, para implementar a Domótica, são necessários o uso de três classes de componentes, que são os sensores, que tem por objetivo receber os dados do ambiente, fornecendo-os para os controladores, que fazem a análise dos dados recebidos para assim determinar qual ação deve ser executada pelo atuador, que pode modificar ou alterar o ambiente.

Inúmeros aparelhos podem ser automatizados, desde os mais simples, como uma cafeteira que irá passar o café em um determinado horário, até os mais complexos como caldeiras residenciais para trabalhar de acordo com a temperatura do ambiente. A área da saúde também pode ser aplicada na domótica, onde os aparelhos podem monitorar os sinais vitais, administrar medicamentos com horários e quantidades específicas, e até sensores de quedas, como o desenvolvido por Torres (2018), especialmente sobre quedas de idosos, na dissertação “Tecnologia Assistiva para Detecção de Quedas: Desenvolvimento de Sensor Vestível Integrado ao Sistema de Casa Inteligente”, onde através de um sensor inercial contendo acelerômetro e giroscópio, é possível detectar a queda e fazer a notificação da mesma.

Cezar (2020), menciona que em todo o mundo, existem várias entidades e organizações que se dedicam a regulamentar e promover a adoção da domótica. Em países onde essa tecnologia já está mais disseminada, como países da Europa e os Estados Unidos, há uma série de associações e entidades dedicadas a estabelecer padrões e diretrizes para a implementação da domótica em residências. Algumas dessas entidades incluem a *European Installation Bus Association* (EIBA) na Europa e a *National Association of Home Builders* (NAHB) nos Estados Unidos, bem como outras organizações similares em diferentes regiões do mundo.

No Brasil, em fevereiro de 2000, foi registrada a AURESIDE (Associação Brasileira de Automação Residencial), que tem como missão fomentar a adoção de tecnologias de Automação Residencial no país, através de informações e conhecimentos de maneira segmentada, com capacitações, treinamentos, cursos, divulgação de artigos e vídeos, dentre outras formas (AURESIDE, 2020).

A Figura 2, exibe algumas das automações que podem ser inseridas em uma casa, trazendo conforto e segurança para o usuário.

**Figura 2 - Automação no ambiente residencial**



Fonte: Freepik (2022)

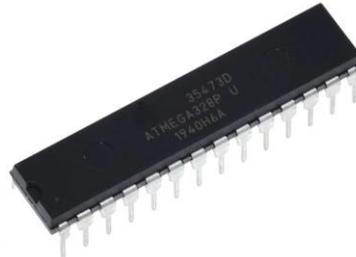
### 2.3 Microcontroladores e microprocessadores

De acordo com Próspero (2020), os microcontroladores são dispositivos que atuam como o controlador do sistema de automação, projetados em um sistema integrado (SI), que resumidamente, possuem um microprocessador programável que irá tomar as decisões do processo, uma memória RAM para guardar informações de maneira temporária, uma memória ROM para registrar as entradas e saídas e uma placa mãe onde todos os componentes são conectados.

Complementa Próspero (2020), que além dos componentes principais, podem ser conectados a este sistema integrado outros componentes para aumentar as funcionalidades da plataforma de desenvolvimento, como conexões USB, módulos de conexões com fio ou Wi-Fi e até mesmo mais portas de entradas ou saídas. Na Figura

3 pode ser visualizado o Microcontrolador ATmega328P DIP-28, que é utilizado nas placas Arduino Uno.

**Figura 3 - Microcontrolador ATmega328P DIP-28**



**Fonte: Byteflop (2017)**

“O microcontrolador pode ser compreendido como um computador simples, enquanto o microprocessador é apenas uma parte desse computador, essencial a todos eles, mas insuficiente para operar sozinho.” Lenz e Torres (2019, p.14).

Nas próximas subseções serão abordadas as principais características de três placas de circuito impresso que contém microcontrolador ou microprocessador, muito utilizados para automação residencial, mostrando as suas principais vantagens e desvantagens, para assim ser feita a escolha do mais adequado para o projeto proposto.

### 2.3.1 Arduino

Arduino é uma das plataformas de prototipagem eletrônicas mais populares do mercado. Na Figura 4, tem-se a placa do modelo Uno, que Oliveira e Zanetti (2015) apontam como um dos modelos mais vendidos e bastante indicado para iniciantes na plataforma.

**Figura 4 - Arduino Uno**



**Fonte: Ralcomp (2019)**

Ao optar pelo arduino, é possível desfrutar de inúmeras vantagens, algumas delas são:

- É relativamente fácil de usar e programar através de seu *software* de programação gratuito, o Arduino IDE, onde sua programação é baseada em C/C++;
- Sua comunidade é ativa, compartilham seus projetos, dicas e soluções de problemas, tornando o aprendizado mais fácil devido a grande quantidade de conteúdo na Internet;
- Seu preço é baixo, facilitando assim a compra para quem deseja um projeto de baixo orçamento;
- Possui grande variedade de *shields*, que são placas que podem ser acopladas ao arduino para aumentar suas funcionalidades, como *shields* com *touchpads*, expansores de entradas e saídas, conectores *Sd card*, botões, *display*, entre outros.

Próspero (2020), aponta que há uma ampla variedade de placas Arduino disponíveis, cada uma com suas características e aplicações específicas. Entre os modelos mais conhecidos estão o Arduino Uno, Arduino Nano e Arduino Mega, onde dependendo do tipo de aplicação, um dos modelos será mais adequado para o uso.

### 2.3.2 ESP

Os dispositivos da família ESP, são placas que possuem microcontroladores embutidos. Semelhantemente aos arduinos, o ESP possui vários tipos de módulos que possuem o mesmo microcontrolador, no caso da Figura 5, pode ser visualizado o Módulo ESP8266 ESP-12E.

Figura 5 - ESP8266 ESP-12E V3



Fonte: Microwat (2016)

Ao optar pelo ESP, é possível desfrutar de inúmeras vantagens, algumas delas são:

- Pode ser usado para fazer a programação a Arduino IDE, sendo necessário apenas a instalação de uma biblioteca na IDE para fazer a sincronização com o ESP;
- Em alguns dispositivos pode ser utilizado o *firmware* Tasmota, que substitui o original do fabricante permitindo que os usuários controlem seus dispositivos de forma mais flexível;
- Semelhantemente ao arduino, sua comunidade é ativa, dessa forma é fácil encontrar conteúdo relacionado ao ESP na Internet;
- Seu preço é baixo, dando mais acessibilidade a sua compra;
- Possui módulo Wi-Fi de fábrica, assim não há a necessidade de gastos adicionais para conseguir conectar o microcontrolador à Internet.

### 2.3.3 Raspberry Pi

Conforme destacado por Próspero (2020), o Raspberry Pi se diferencia dos demais pois é considerado um microcomputador que não contém um microcontrolador, mas apenas um microprocessador. A Figura 6, mostra um Raspberry Pi 4, onde pode ser visto que sua estrutura é de placa única, chamada de sistema integrado, em que todos os componentes são soldados em uma única placa, diferente de desktops e notebooks comuns, onde, em geral, seus componentes são apenas conectados entre si.

**Figura 6 - Raspberry Pi 4**



**Fonte: Tectudo (2019)**

Ao optar pelo Raspberry Pi, é possível desfrutar de inúmeras vantagens, algumas delas são:

- O poder de processamento do Raspberry Pi é superior ao arduino e Esp, permitindo a execução de tarefas complexas com mais facilidade;
- Possui vários tipos de conexões com a Internet, como Wi-Fi, Bluetooth, Ethernet;
- Semelhante aos outros microcontroladores citados, a comunidade do Raspberry é muito ativa, assim, não é difícil encontrar conteúdo para estudo;
- Levando em consideração todas as suas características, seu preço é relativamente baixo.

#### 2.3.4 Definição do dispositivo

Um dos principais objetivos deste trabalho é deixar a reprodução do sistema desenvolvido de fácil acesso, por isso deve-se fazer a escolha do dispositivo que tenha o melhor custo benefício para o usuário. Todos os três dispositivos analisados conseguem executar o projeto proposto de maneira satisfatória, assim, o microcontrolador escolhido será o que possui o menor custo.

Inicialmente pode-se descartar o Raspberry Pi, que apesar de possuir um *hardware* superior que o Arduino e o ESP, seu preço é relativamente mais alto que os outros dois, e no projeto proposto, não há a necessidade de uma grande capacidade de processamento, pois o dispositivo precisa apenas executar tarefas relativamente simples.

O ESP8266 ESP-12E V3 possui um preço relativamente baixo, semelhante ao do arduino UNO, mas com a vantagem de possuir de fábrica um módulo Wi-Fi, assim, não há a necessidade de gastos adicionais com um *shield* que seria necessário no arduino.

Assim, a melhor escolha a ser tomada é a utilização do ESP8266 ESP-12E V3, visto que possui o menor custo junto a uma estrutura capaz de atender os requisitos do projeto de maneira satisfatória.

## 2.4 Protocolo de comunicação IoT

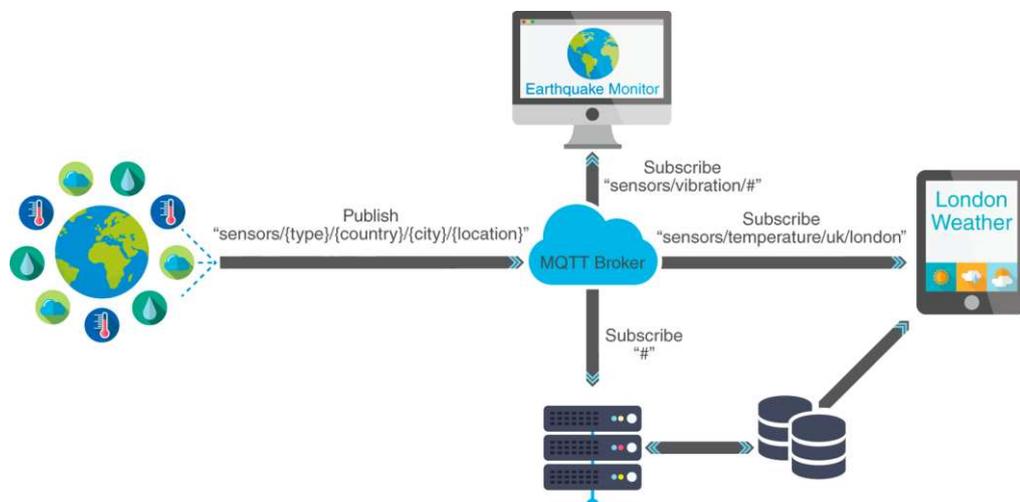
O protocolo de comunicação utilizado nesse estudo é o protocolo MQTT (*Message Queuing Telemetry Transport*), pois apresenta várias características que

fazem com que esse protocolo seja amplamente utilizado na automação residencial, principalmente devido ao seu baixo consumo de recursos, alta confiabilidade e alto nível de segurança.

Como apontado por Moraes e Hayashi (2021), o protocolo MQTT é um protocolo baseado em um modelo cliente/servidor sobre TCP/IP, muito utilizado em sistema IoT devido ao seu baixo consumo de processamento, de bateria e banda de comunicação, fazendo com que os dispositivos conectados se comuniquem de forma eficiente, mesmo conectados banda limitada ou instável, como redes móveis ou com conexões de Internet lentas. Neste protocolo, os dispositivos são conectados a um servidor nomeado *broker* que é responsável pelo gerenciamento e comunicação dos dispositivos através do paradigma *Publish/Subscribe*.

O processo *Publish* é responsável por publicar informações em um determinado tópico do *broker*, para que através do *Subscribe* os dispositivos se conectem ao tópico e consigam fazer a leitura dos dados que foram publicados. A leitura dos tópicos depende da retentividade em que as informações foram publicadas, caso não exista retentividade, logo após a publicação os dados serão apagados, assim, apenas usuários já inscritos no tópico terão acesso a informação. Para casos onde existe retentividade, os dados publicados ficarão disponíveis no tópico até o momento em que o usuário fizer a remoção, desta forma, mesmo que o usuário utilize o processo *Subscribe* após um período de tempo, a informação irá se manter disponível para ser lida. A Figura 7 representa uma conexão do tipo MQTT, onde dados sobre fenômenos naturais são publicados e lidos através de um Broker.

**Figura 7 - Publicação de mensagens no MQTT Broker**



Fonte: Adaptado de Stack Overflow (2017)

Maschietto et al. (2021), destacam que o protocolo MQTT apresenta três níveis de *Quality of Service* (QoS), cada um indicando o esforço do servidor para garantir a entrega da mensagem. Para o QoS 0 (*at most once*) não existe confirmação do recebimento da mensagem, fazendo com que o emissor não tenha certeza que a entrega ocorreu com sucesso. Para o QoS 1 (*at least once*) o emissor tem certeza que pelo menos uma mensagem será entregue, no qual a confirmação é feita pelo destinatário, enviando uma mensagem ao emissor para confirmar o recebimento. Por fim, tem-se o QoS 2 (*exactly once*), permitindo que a mensagem seja entregue exatamente uma vez através de um sistema de confirmação nos dois sentidos de tráfego.

Para um sistema IoT deve ser levado em consideração quem tem acesso às informações e acesso dos equipamentos, para proteger a privacidade e até, em casos extremos, a saúde física dos usuários, dessa forma, é utilizado um mecanismo de autenticação, com usuário e senha, garantido o acesso das informações para apenas entidades confiáveis. Moraes e Hayashi (2021) complementam a necessidade de combinar o TLS (Transport Layer Security) com o MQTT, criptografando os dados que estão sendo enviados para o *broker*, fornecendo mais uma camada de proteção para que seja garantida a integridade dos dados.

## 2.5 Tasmota

Segundo a página oficial do Tasmota no *GitHub*, o Tasmota é um *open source firmware* criado e mantido por Theo Arends para os microcontroladores ESP8266, ESP32, ESP32-S ou ESP32-C3, que são desenvolvidos pelo fabricante chinês *Espressif*. O desenvolvimento desse *firmware* iniciou devido a dispositivos que possuem o ESP8266 em sua estrutura, especialmente o interruptor *Sonoff*, que segundo sua descrição no site *Smartkits*, é um relé Wi-Fi muito utilizado na automação residencial devido a sua praticidade e ótimo custo benefício.

O *Sonoff* possui um *firmware* que não é aberto para modificações, assim, através da instalação do Tasmota, o microcontrolador se torna desbloqueado para controlar dispositivos e fazer leitura de sensores, o que não era possível com o *firmware* padrão do *Sonoff* que era barrado para controlar apenas um dispositivo.

**Figura 8 - Interruptor Sonoff Basic R2**

Fonte: IoTudo (2022)

Guimarães (2020), cita que *firmware* é feito para ser executado especificamente em um determinado microcontrolador/microprocessador, realizando um controle de baixo nível do *hardware*, ou seja, possui um controle direto do *hardware* do equipamento, diferentemente do *software*, que não interage diretamente com o *hardware*, podendo assim ser executado em diferentes dispositivos sem necessidade de alterações.

A utilização do Tasmota muitas vezes vem da necessidade da padronização dos dispositivos utilizados na automação residencial, que podem ser de fabricantes diferentes, assim, geralmente, cada uma possui um *firmware* e aplicativo específico para seu dispositivo. Com a instalação do Tasmota nos dispositivos, é possível controlar todos apenas utilizando uma interface, aumentando a comodidade para o usuário. Além disso, também pode ser usado para dispositivos como o Nodemcu v3 esp-12e, facilitando a programação para a maioria das atividades, que em grande parte, podem ser feitas através de caixa selecionáveis e listas suspensas, permitindo o desenvolvimento de automações residenciais para indivíduos com pouca familiaridade em programação.

Outra grande vantagem que pode ser citada é suporte ao protocolo MQTT do Tasmota, dessa forma é possível desenvolver projetos que podem ser acessados pela Internet sem a necessidade de uma rede local. Na Figura 9, pode ser visualizada a interface web do Tasmota, usada para configurar e controlar dispositivos que possuem *firmware* Tasmota. Nessa interface é possível acessar diversas configurações, como configurar a rede Wi-Fi, configurar o protocolo MQTT, definir a hora, configurar sensores, definir regras de automação, entre outras coisas.

**Figura 9 - Interface do Tasmota**



**Fonte: Autoria própria (2023)**

### 3 MATERIAIS E MÉTODOS

Nesta seção, serão abordados os materiais e métodos utilizados para o desenvolvimento do sistema proposto.

#### 3.1 Componentes eletrônicos e montagem

Foram utilizados componentes eletrônicos de fácil acesso para o desenvolvimento do projeto e para a montagem dos componentes eletrônicos, foi utilizada uma placa de prototipagem (*protoboard*). Através da Tabela 1, podem ser visualizados os componentes utilizados no desenvolvimento do projeto.

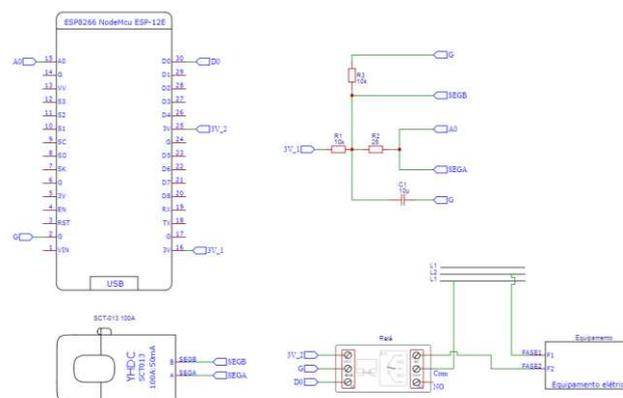
**Tabela 1 - Lista de componentes**

Componente	Quantidade
Módulo ESP8266 ESP-12E	1
Módulo Relé 4 canais 5V	1
Sensor SCT-013 100A	1
Capacitor 10 $\mu$ 50V	1
Resistor 10K $\Omega$	2
Resistor 26 $\Omega$	1
Plugue macho 10A	1
Plugue fêmea 10A	1
Placa de prototipagem	1
Fio paralelo 2x4mm <sup>2</sup>	2m

Fonte: Autoria própria (2023)

Para a montagem, elaborou-se um diagrama esquemático por meio do *software* EasyEDA, a fim de facilitar a montagem em bancada. O diagrama esquemático pode ser visualizado na Figura 10.

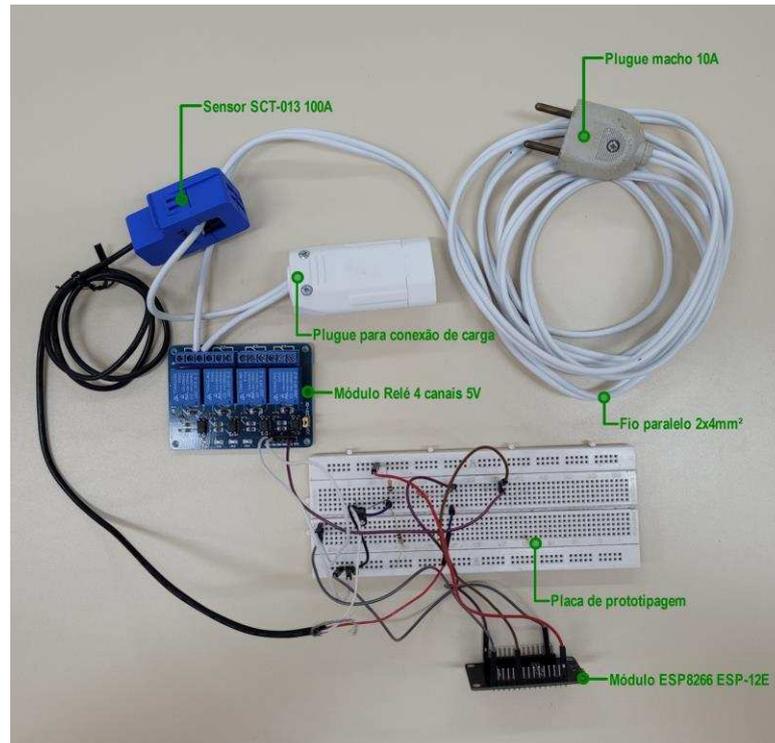
**Figura 10 - Diagrama esquemático**



Fonte: Autoria própria (2023)

Através do diagrama esquemático, utilizando os componentes listados na Tabela 1, foi realizada a montagem do sistema em bancada. A representação do sistema após a conclusão da montagem pode ser visualizada através da Figura 11.

**Figura 11 - Montagem do sistema**



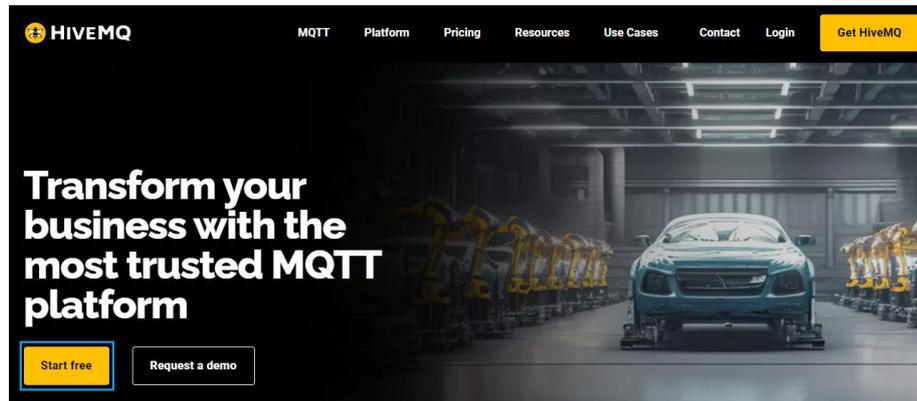
Fonte: Autoria própria (2023)

### 3.2 Configuração do MQTT

Para o acesso remoto ao sistema, é necessário utilizar um broker MQTT. Uma das melhores opções disponíveis atualmente é o broker da HiveMQ, que possui um plano gratuito onde pode-se conectar até 100 aparelhos com um tráfego mensal de até 10 GB, o que é uma quantidade suficiente de aparelhos para uma automação residencial.

Primeiramente deve-se acessar o site da HiveMQ (disponível em <https://www.hivemq.com/>) e clicar sobre o botão "Start free". A Figura 12 mostra a página inicial do site da HiveMQ.

Figura 12 - Página inicial do HiveMQ

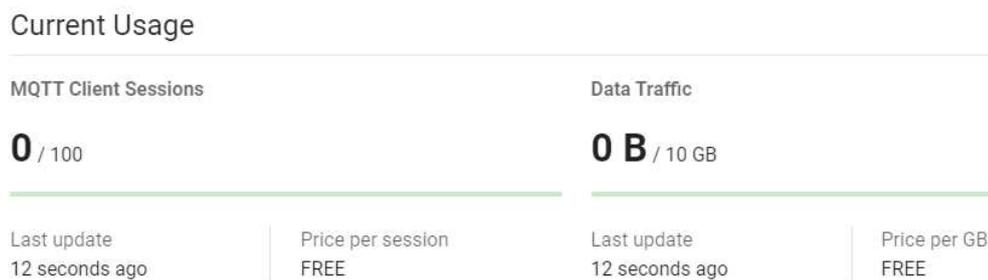


Fonte: HiveMQ (2023)

Ao ser redirecionado, estarão disponíveis três planos. Assim, deve-se utilizar o primeiro plano, “*HiveMQ Cloud*”, que pode ser utilizado diretamente do navegador, sem a necessidade de *downloads* ou instalações. Em seguida, deve-se fazer um cadastro para acesso ao painel de configurações. Para isso, deve-se usar a aba “*Sign Up*”, ou fazer *login* diretamente com o GitHub, Google ou LinkedIn. Após o *login*, preenche-se um breve questionário com nome, cargo e empresa para prosseguir para o painel de configurações do HiveMQ.

Com o cadastro finalizado, já se tem acesso às configurações do MQTT Broker. Na aba “*Overview*”, através do painel da Figura 13, é possível ver a quantidade de aparelhos conectados e o tráfego mensal para acompanhamento.

Figura 13 - Painel de uso do HiveMQ



Fonte: Autoria própria (2023)

Para dar início à configuração do MQTT, é essencial criar um nome de usuário e uma senha, assegurando a segurança do sistema. Essas credenciais podem ser geradas no painel “*Credentials*” do HiveMQ, inserindo o nome de usuário no campo “*Username*” e a senha nos campos “*Password*” e “*Confirm Password*”. Deve-se

certificar que a lista suspensa “*Permission*” está selecionada como “*Publish and subscribe*” para permitir a publicação e leitura de informações, do mesmo modo da Figura 14.

**Figura 14 - Criação de credenciais do HiveMQ**

Credentials

Currently you have not created any credentials. Fill out the following form to create an access credentials pair and limit access to your HiveMQ Cloud MQTT instance. To learn more [check out our Security Fundamentals guide](#).

Username \*  
Tasmota  
At least 5 characters

Password \*  
Utfpr2023  
At least 8 characters, 1 digit, 1 uppercase character

Confirm Password \*  
.....  
Passwords must match

Permission \*  
Publish and Subscribe  
Add permissions to limit access

> CREATE CREDENTIAL

**Fonte: Autoria própria (2023)**

Para conectar-se ao Tasmota, além do nome de usuário e senha, são necessários o URL e a porta do broker. Essas informações podem ser visualizadas no painel inicial, conforme a Figura 15.

**Figura 15 - Informações de conexão do HiveMQ**

Connect your first client

**Your Connection Settings**  
Your cluster is already up and running!  
Use these connection setting to connect to your cluster.

**Cluster URL**  
0d2a09bf4cc54bc5b200d490fdbb198e.s2.eu.hivemq.cloud

**Port**  
8883

**Websocket Port**  
8884

**Fonte: Autoria própria (2023)**

### 3.3 Instalação do Tasmota

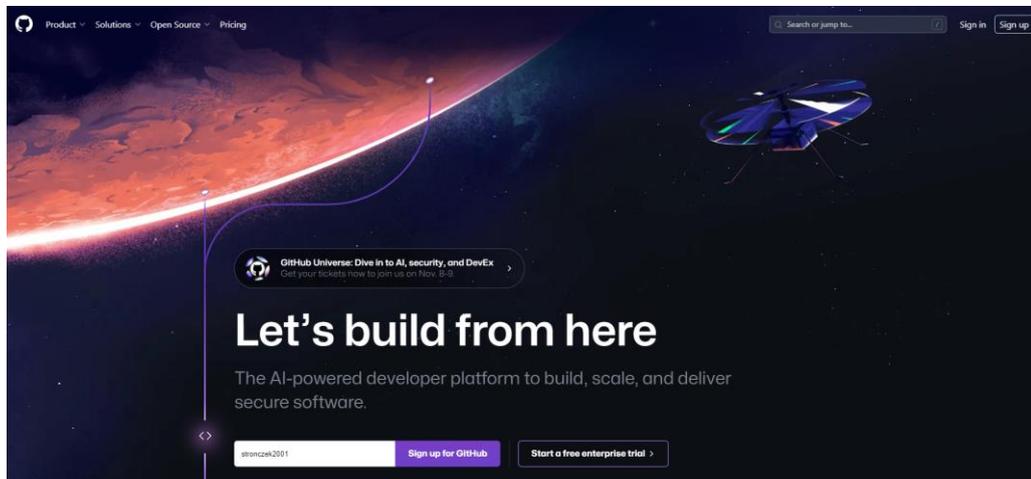
O propósito desta seção é ilustrar o processo de compilação e instalação do *firmware*. Isso se deve ao fato de que as versões pré-compiladas disponíveis no site

oficial do Tasmota não incluem todos os recursos necessários para a execução do projeto, especialmente no que diz respeito à conexão ao broker MQTT disponibilizado pela HiveMC.

### 3.3.1 Compilação firmware

Conforme a Figura 16, deve-se entrar no site oficial do GitHub (disponível em <https://github.com>) e fazer o *login*. Caso o usuário não possuir uma conta no GitHub, basta clicar no botão “*Sign up*” para efetuar o cadastro.

Figura 16 - Página inicial do GitHub



Fonte: GitHub (2023)

Com o *login* concluído, na barra de busca deve-se buscar pelo termo “*tasmocompiler*”. Para utilizar o Tasmocompiler, deve-se ir até a seção “*How to start using Tasmocompiler?*” e clicar sobre o botão “*link to Gipod*”, para que seja aberta a aba desenvolvimento e seja feita configuração e compilação do *firmware*.

Primeiramente deve-se fazer login no Gitpod, para isso, deve-se clicar sobre o botão “*Continue with GitHub*” e autorizar o uso do mesmo. Em seguida da sincronização do Gitpod ao GitHub, deve-se escolher o editor de código-fonte, onde deve-se selecionar dentro da lista suspensa o editor “*Vs Code*”. O tema altera apenas a interface do editor, assim, será mantido o tema padrão “*System*”.

Ao clicar no botão “*Continue*”, será solicitado a responder a um breve questionário. Em seguida, será necessário fornecer um número de telefone válido para receber um código de verificação. Após inserir o código enviado para o número fornecido, o espaço de trabalho será carregado.

Com o redirecionamento do Gipod, o terminal irá importar o Tasmocompiler, e após um tempo surgirá a mensagem “*Server started on port 3000*”, indicando que a importação foi concluída com sucesso, conforme a Figura 17.

**Figura 17 - Terminal do Gipod**

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[ ] Gitpod Task 1: node + v

Collecting sniffio>=1.1 (from anyio<5,>=3.4.0->starlette<0.32,>=0.19->platformio)
  Downloading https://files.pythonhosted.org/packages/c3/a0/5dba8ed157b0136607c7f2151db698850069e8d1fae123dc3391e8cfd9f/sniffio-1.3.0-py3-none-any.whl
Collecting exceptiongroup; python_version < "3.11" (from anyio<5,>=3.4.0->starlette<0.32,>=0.19->platformio)
  Downloading https://files.pythonhosted.org/packages/ad/83/b71e58866f156a39f029417e4c8ca4bc74908c8d4de9e8842ab54dc8c344/exceptiongroup-1.1.3-py3-none-any.whl
Collecting zipp>=0.5 (from importlib-metadata; python_version < "3.8")<-click<8.2,>=8.0.4->platformio)
  Downloading https://files.pythonhosted.org/packages/5b/fa/c9e82bbe1af626aadf08af563905eb87cab83fde0a0a08963510521047/zipp-3.15.0-py3-none-any.whl
Installing collected packages: bottle, tabulate, semantic-version, typing-extensions, sniffio, exceptiongroup, idna, anyio, starlette, packaging, marshmallow, colorama, h11, wsproto, ajsonrpc, pyelft
ools, zipp, importlib-metadata, click, uvicorn, pyserial, charset-normalizer, certifi, urllib3, requests, platformio
The script tabulate is installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
The script async-json-rpc-server is installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
The script uvicorn is installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
The scripts pyserial-miniterm and pyserial-ports are installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
The script normalizer is installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
The scripts pio, piodebugdb and platformio are installed in '/home/gitpod/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed ajsonrpc-1.2.0 anyio-3.7.1 bottle-0.12.25 certifi-2023.7.22 charset-normalizer-3.3.1 click-8.1.7 colorama-0.4.6 exceptiongroup-1.1.3 h11-0.14.0 idna-3.4 importlib-metadata-6.7.
0 marshmallow-3.19.0 packaging-23.2 platformio-6.1.11 pyelftools-0.29 pyserial-3.5 requests-2.31.0 semantic-version-2.10.0 sniffio-1.3.0 starlette-0.29.0 tabulate-0.9.0 typing-extensions-4.7.1 urllib
3-2.0.7 uvicorn-0.22.0 wsproto-1.2.0 zipp-3.15.0
Server started on port 3000

```

**Fonte: Autoria própria (2023)**

Nessa página, realiza-se a configuração do Tasmota para compilação. Na primeira etapa, o usuário deve clicar sobre o botão "*Download Source*", onde o código-fonte do Tasmota será baixado. Após o *download*, é necessário clicar em "*Next*" para avançar para a próxima etapa.

Na segunda etapa, é realizada a configuração do Wi-Fi. Nesse processo, ao fornecer o login e senha, o módulo será automaticamente conectado à internet sempre que for ligado.

Na terceira etapa, realiza-se a configuração dos recursos extras desejados para inclusão no *firmware*. Inicialmente, escolhe-se a placa ESP8266 como "*Generic*" para reconhecer o módulo. Em seguida, são apresentados todos os recursos que o Tasmota irá suportar. Por padrão, a interface web, regras e temporizadores estão selecionados. Para este projeto específico, além dos recursos já selecionados por padrão, é necessário marcar a caixa "*MQTT over TLS*", conforme a Figura 18, uma vez que o broker MQTT a ser utilizado requer tráfego seguro TLS.

**Figura 18 - Terceira etapa para a compilação do Tasmota**

3 Select features

Select board your project is based on

ESP8266

Generic     Wemos/NodeMCU     Shelly-type     SonOff Zigbee Bridge

---

ESP32

Generic     Webcam     Odroid-Go     M5Stack Core2     Solo1     ESP32 C3

ESP32 S2     ESP32 S3

Which features should be included in final binary firmware? [↔](#)

<input type="checkbox"/> Air/gas sensors	<input type="checkbox"/> Amazon Alexa	<input type="checkbox"/> Arduino slave	<input type="checkbox"/> Display Vcc	<input type="checkbox"/> Displays (I2C/SPI)	<input type="checkbox"/> Distance sensors
<input type="checkbox"/> Domicz	<input type="checkbox"/> Energy sensors	<input type="checkbox"/> Home Assistant	<input type="checkbox"/> IO port expander	<input type="checkbox"/> IR support	<input type="checkbox"/> KNX
<input type="checkbox"/> Light sensors	<input type="checkbox"/> mDNS discovery	<input type="checkbox"/> Modbus Bridge	<input checked="" type="checkbox"/> MQTT over TLS	<input type="checkbox"/> RF transceiver	<input checked="" type="checkbox"/> Rules
<input type="checkbox"/> Script	<input type="checkbox"/> SD card/LittleFS	<input type="checkbox"/> Shutters and Blinds	<input type="checkbox"/> Temp/Hum sensors	<input checked="" type="checkbox"/> Timers	<input type="checkbox"/> Tuya MCU
<input checked="" type="checkbox"/> Web interface	<input type="checkbox"/> WS2812 Leds				

**Fonte: Autoria própria (2023)**

Na quarta etapa, efetua-se a configuração de parâmetros extras, os quais, neste caso, serão deixados em branco, uma vez que não será necessário utilizar nenhum parâmetro extra.

A última etapa é destinada para a escolha da versão do Tasmota e a linguagem. Para esse projeto será escolhida a versão mais recente, que atualmente é a v13.1.0 junto com a linguagem inglesa. Após a seleção de versão e linguagem basta clicar sobre o botão “*Compile*”, para dar início a compilação.

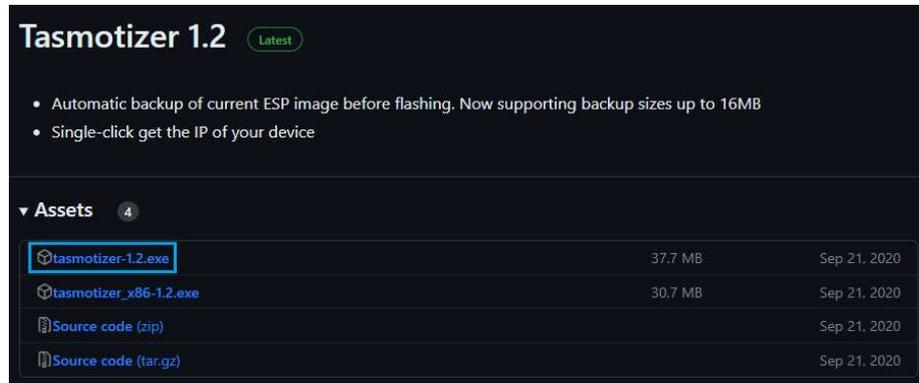
Por fim, após o fim da compilação, pode-se baixar o *firmware*, onde, para isso, utilizaremos o botão “Firmware.bin” para iniciar o *download*.

### 3.3.2 Instalação com o Tasmotizer

Para instalar o Tasmota através de um arquivo .bin, é necessário usar um *software* específico, como o Tasmotizer. Para a instalação do Tasmotizer, é necessário baixar a versão mais recente do *software* no GitHub, que atualmente é versão 1.2, acessando o *link* disponível no site oficial do Tasmota (disponível em <https://github.com/tasmota/tasmotizer/releases>). Ao acessar o site, deve-se escolher a versão do executável que corresponde à arquitetura do computador utilizado na instalação. Para um computador com um sistema operacional de 64 bits, deve-se baixar a opção "tasmotizer 1.2.exe". Para sistemas operacionais de 32 bits, o executável que deve ser baixado é o "tasmotizer X86-1.2.exe". A Figura 19 mostra a

página de *download* do Tasmotizer e a opção que deve ser baixada para um sistema operacional de 64 bits.

Figura 19 - Página de *download* do Tasmotizer 1.2



Fonte: Autoria própria (2023)

Após concluir o *download*, basta conectar o módulo através de um cabo USB ao computador e executar o Tasmotizer para acessar sua interface de configuração. Para a configuração, é necessário selecionar a porta à qual o módulo está conectado através da lista suspensa "*Select port*". Após selecionar a porta serial, deve-se abrir o arquivo bin que foi baixado anteriormente na seção de *download* do site oficial do Tasmota. Para isso, basta clicar no botão "*open*" e selecionar o arquivo no sistema operacional. O Tasmotizer oferece a opção de formatar o microcontrolador antes da instalação. Assim, deve-se marcar a caixa "*Erase before flashing*" para garantir a formatação do microcontrolador. Por fim, basta clicar no botão "*Tasmotize!*" para iniciar a instalação do *firmware*. A Figura 20 exhibe a interface do Tasmotizer 1.2 após a configuração para a instalação do Tasmota.

**Figura 20 - Interface do Tasmotizer 1.2**



**Fonte: Aatoria própria (2023)**

Ao clicar no botão "Tasmotize!", o sistema iniciará a instalação e, ao concluir a instalação do *firmware*, uma notificação será exibida.

Caso a rede de internet não tenha sido preenchida durante a compilação, o módulo irá criar uma rede Wi-Fi à qual devemos nos conectar. Isso nos permitirá configurar a conexão do microcontrolador. Conectado à rede Wi-Fi gerada pelo módulo, deve-se acessar o IP 192.168.4.1 através do navegador. Por padrão, este IP é utilizado para configurar a rede Wi-Fi. Ao acessar o IP, serão listadas as redes Wi-Fi disponíveis. Deve-se selecionar a rede desejada e inserir a senha correspondente na caixa de texto "*WiFi Password*". Após isso, basta clicar no botão "Save" para concluir a configuração.

Após a conclusão da configuração do Wi-Fi, semelhante a Figura 21, a página exibirá o endereço IP que será usado para acessar a interface padrão do Tasmota,.

**Figura 21 - IP de acesso ao Tasmota**



**Fonte: Aatoria própria (2023)**

## 3.4 Configuração do Tasmota

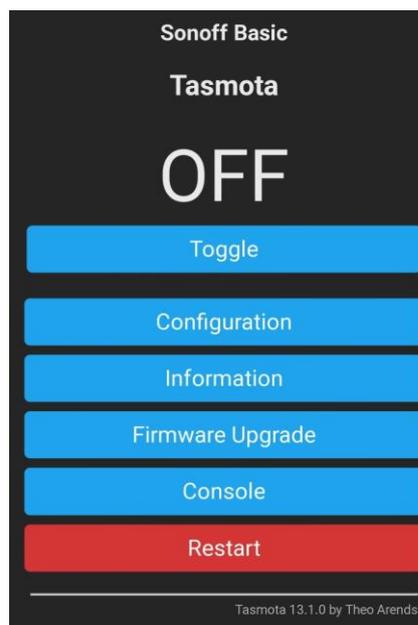
### 3.4.1 Configuração das entradas e saídas

Com a instalação de Tasmota, será possível acessar a interface através do IP obtido ao fim da instalação, onde será feito a configuração e controle do projeto. A Figura 22 mostra a interface inicial do Tasmota após sua instalação.

A versão inicial do Tasmota é configurada com base em um *Sonoff*, ou seja, um interruptor inteligente. Dessa forma, na tela inicial, existe um botão nomeado "*Toggle*" para fazer o controle desse interruptor. Com um clique sobre esse botão, a condição da digital do módulo é alternada. Na condição *off*, ou seja, desligada, o módulo fornece 0V. Já na condição *on*, ou seja, ligado, o módulo fornece 3,3V.

Outras seções estão disponíveis para acesso na página inicial do Tasmota. A aba "*Configuration*" é responsável pelas configurações básicas, incluindo comportamentos das entradas e saídas digitais do módulo, configurações Wi-Fi, configurações do MQTT, entre outras. A seção "*Information*" exibe informações básicas do Tasmota e suas conexões. A aba "*Firmware Upgrade*" é usada para atualizar a versão do Tasmota. O console é uma seção para visualizar informações em tempo real e inserir *inputs* de configurações do firmware. Por fim, o botão "*Restart*" tem a função de recarregar as informações da tela.

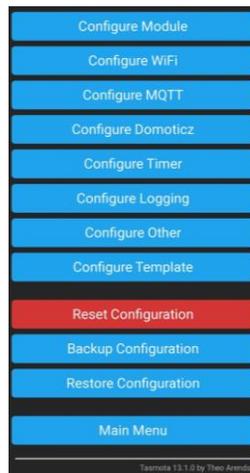
Figura 22 - Interface inicial do Tasmota



Fonte: Autoria própria (2023)

Inicialmente, é realizada a configuração da entrada analógica para o sensor de corrente não invasivo e a saída digital para controle do relé. Para isso, deve-se clicar sobre o botão "*Configuration*". Ao acessar a página de configurações, semelhante a Figura 23, é necessário entrar na seção "*Configure Module*", que é responsável por definir o comportamento de todas as entradas e saídas digitais ou analógicas do módulo ESP8266.

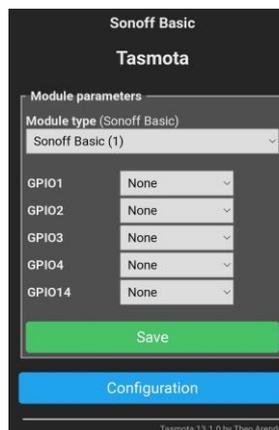
**Figura 23 - Aba de Configurações**



**Fonte: Autoria própria (2023)**

Ao acessar a página de configuração do módulo, a primeira etapa é alterar o tipo de módulo. Isso pode ser feito através da lista suspensa "*Module Type*", destacada na Figura 24, onde por padrão está selecionada como "*Sonoff Basic (1)*". Essa configuração inicial é específica para as entradas e saídas disponíveis em um *Sonoff Basic*.

**Figura 24 - Aba de configurações do módulo**



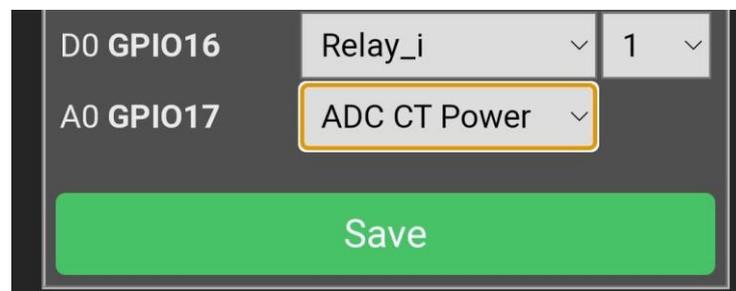
**Fonte: Autoria própria (2023)**

Atualmente, o Tasmota não oferece suporte nativo para o módulo Esp8266. Portanto, é necessário selecionar o módulo "*Generic (0)*". Esta opção fornecerá suporte satisfatório para o nosso módulo, permitindo a configuração adequada das entradas e saídas.

Neste projeto, é utilizada uma entrada analógica para receber os dados do sensor de corrente e uma saída digital para o controle do relé. Dessa forma, o pino D0 será empregado para a saída digital, e o pino A0 será designado para a entrada analógica, possibilitando a configuração adequada para a leitura do sensor e o controle do relé.

Ao clicar na lista suspensa, será possível escolher várias configurações pré-definidas para serem utilizadas, como *sliders*, botões, entre outros. Para o controle do relé, será selecionada a opção "*Relay\_i*". Dessa forma, na página inicial do Tasmota, ficará disponível um botão de liga e desliga para o controle do relé. Para a entrada analógica, será utilizada a pré-configuração "*ADC CT Power*", que faz a leitura dos dados do sensor e os apresenta de forma simples para leitura. A Figura 25 evidencia a configuração do módulo para controle do relé e leitura do sensor de corrente.

**Figura 25 - Configuração da saída digital e entrada analógica**



**Fonte: Autoria própria (2023)**

Com essa pré-configuração, a interface já está pronta para controlar o relé e exibir a leitura do sensor. A Figura 26 destaca a página inicial do Tasmota, na qual é possível controlar o relé por meio do botão nomeado "*Toggle*" e monitorar a voltagem, corrente, potência e consumo por meio dos valores apresentados.

**Figura 26 - Interface pré-configurada**



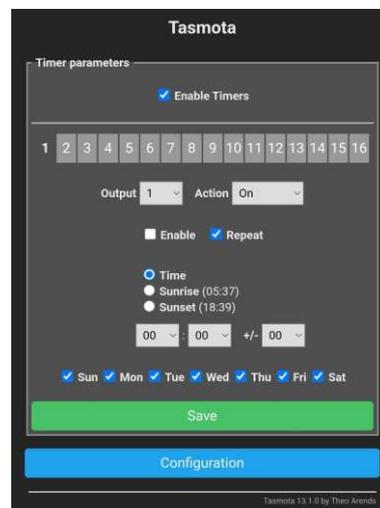
Fonte: A autoria própria (2023)

### 3.4.2 Configuração dos temporizadores

Para configurar os temporizadores no Tasmota, é necessário acessar a aba “*Configuration*” e, em seguida, a aba “*Configure Timer*”. Para ativar os temporizadores, é preciso marcar a caixa de seleção “*Enable Timers*”. O Tasmota suporta até 16 temporizadores, mas neste caso, apenas os temporizadores 1 e 2 serão utilizados.

No temporizador 1, de acordo com a Figura 27, o usuário deve selecionar o “*Output 1*”, responsável pelo controle do relé. A ação deve ser definida como “*On*” para ativar o relé quando o temporizador for acionado. Deve-se ativar a opção “*Time*” para que o temporizador leve em consideração o horário configurado. Todas as 7 caixas de seleção referentes aos dias da semana em que o temporizador deve funcionar devem ser marcadas, para que ele não seja restrito a dias específicos.

**Figura 27 - Temporizador**



Fonte: A autoria própria (2023)

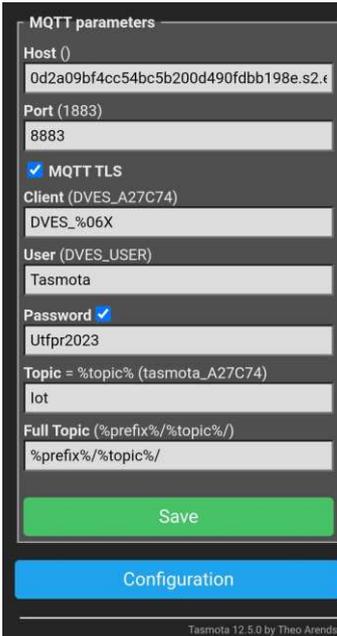
No caso do temporizador 2, a mesma configuração do temporizador 1 deve ser utilizada, com exceção da opção “*Action*”, que deve ser alterada para “*Off*”. Dessa forma, o relé será desativado no horário programado.

### 3.4.3 Configuração do MQTT

Para configurar o MQTT no Tasmota, é necessário acessar a aba “*Configuration*” e, em seguida, a aba “*Configure MQTT*”. Na página de configuração do MQTT, o usuário deve preencher as informações fornecidas pelo broker da HiveMQ. Primeiramente, é necessário preencher o campo “*Host*”, disponibilizado como “*Cluster URL*”, juntamente com o preenchimento do campo “*Port*”. Além disso, a opção “*MQTT TLS*” deve ser marcada, pois o broker MQTT requer esse protocolo de segurança para a comunicação.

Os campos “*User*” e “*Password*” devem ser preenchidos com o nome de usuário e senha, respectivamente, conforme configurado na seção “*Credentials*” na HiveMQ. Por fim, é necessário definir um nome para o tópico da comunicação MQTT no campo “*Topic*”, que pode ser um nome arbitrário, como “*IoT*”, por exemplo. Após preencher todas as informações necessárias, como destacado na Figura 28, basta clicar no botão “*Save*” para salvar a configuração.

Figura 28 - Configuração da aba MQTT no Tasmota



The screenshot displays the 'MQTT parameters' configuration interface. It includes the following fields and settings:

- Host ()**: 0d2a09bf4cc54bc5b200d490fdbb198e.s2.€
- Port (1883)**: 8883
- MQTT TLS**
- Client (DVES\_A27C74)**: DVES\_%06X
- User (DVES\_USER)**: Tasmota
- Password**: Utfpr2023
- Topic = %topic% (tasmota\_A27C74)**: lot
- Full Topic (%prefix%/topic%)**: %prefix%/topic%

A green 'Save' button is located at the bottom of the form. Below the form is a blue 'Configuration' button. The footer of the interface reads 'Tasmota 12.5.0 by Theo Arends'.

Fonte: Autoria própria (2023)

Após configurar o MQTT no Tasmota, é possível utilizar a aba “Console” para confirmar a conexão. Se a mensagem “Connected” for exibida, como na Figura 29, significa que a conexão entre o Módulo ESP8266 e o broker MQTT foi estabelecida com sucesso.

**Figura 29 - Confirmação de conexão com o broker MQTT**

The screenshot shows the Tasmota console interface with the following log output:

```

Tasmota
00:00:00.001 MDW: ESP8266EX
00:00:00.050 CFG: Loaded from flash at F7,
00:00:00.055 QPC: Count 1
00:00:00.072 Project tasmota - Tasmota Ver:
00:00:00.561 WIF: Connecting to AP1 TASMOT
00:00:03.762 WIF: Connected
00:00:04.015 HTP: Web server active on Iot
17:32:46.047 MQT: Attempting connection...
17:32:48.599 MQT: TLS connected in 2546 ms
17:32:48.600 MQT: MFLN not supported by TL
17:32:48.601 MQT: Connected
17:32:49.219 MQT: tele/Iot/LWT = Online (r
17:32:49.541 MQT: cmd/Iot/POWER =
17:32:51.115 MQT: tele/Iot/INFO1 = {"Info1
17:32:51.747 MQT: tele/Iot/INFO2 = {"Info2
17:32:52.383 MQT: tele/Iot/INFO3 = {"Info3
17:32:53.204 MQT: stat/Iot/RESULT = {"POWEI
17:32:54.025 MQT: stat/Iot/POWER = OFF
17:32:55.092 QPC: Reset
17:32:58.834 MQT: tele/Iot/STATE = {"Time"
17:32:59.656 MQT: tele/Iot/SENSOR = {"Time

```

Below the log output, there is an input field labeled "Enter command" and a blue button labeled "Main Menu". At the bottom of the interface, it says "Tasmota 12.5.0 by Theo Arends".

**Fonte: Autoria própria (2023)**

É necessário configurar o tempo de atualização da comunicação MQTT, que por padrão no Tasmota é de 300 segundos. Neste caso, a taxa de atualização será reduzida para o mínimo aceito pelo Tasmota, que é de 10 segundos. Para realizar essa alteração, basta utilizar o comando “TelePeriod 10” no console.

Antes de iniciar a configuração do aplicativo MQTT, é necessário realizar uma pré-calibração do sensor por meio do console. Utilizando um alicate amperímetro e uma carga conectada ao sistema, é possível aferir a corrente real consumida. Comparando esse valor com o indicado na página inicial do Tasmota, é possível estabelecer uma relação, obtendo-a ao dividir a corrente real pela indicada no sensor de corrente não invasivo.

Além disso, deve-se verificar o valor exibido na página web do Tasmota sem nenhum dispositivo conectado. Caso esse valor não seja igual a 0 Amperes, é necessário definir um limite mínimo de corrente para que seja contabilizada.

Para a calibração, utiliza-se a seguinte estrutura: “AdcParam 7,<param1>,<param2>,<param3>,<param4>”. O <param1>, em nosso caso, deve ser igual a 0, pois é o valor destinado a dispositivos de corrente não invasivos. O

<param2>, por padrão, é utilizado o valor 2146. Deve-se multiplicar esse valor pela relação entre a corrente real e o valor indicado no Tasmota. No exemplo, a corrente real foi aproximadamente 3,65 vezes maior que a corrente indicada no Tasmota, resultando em um input de 7832.9. O <param3> é a tensão da rede, que por padrão é 0.230 (230V). Neste caso, faz-se um ajuste prévio para 127V, ou seja, seu input é 0.127. O <param4> é o valor mínimo necessário para ativar o sistema, assim, após testes, o valor adequado foi de 0,85 Amperes, dessa forma, seu input foi 0.85. Portanto, como pode ser visualizado na Figura 30, o comando inserido ao console foi "AdcParam 7,0,7832.9,0.127,0.85".

**Figura 30 - Pré-calibração do sensor de corrente**

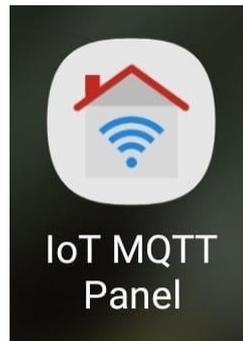


**Fonte: Autoria própria (2023)**

### 3.5 Aplicativo MQTT

Para a conexão MQTT, é utilizado um aplicativo gratuito disponível nas principais lojas de aplicativos. Isso permite que todo o controle e monitoramento ocorram de forma global, sem a necessidade de estar conectado à rede local onde o módulo Esp8266 está vinculado, o que é um requisito para acessar a página web do Tasmota.

O aplicativo utilizado é o "IoT MQTT Panel", destacado na Figura 31, que é caracterizado por uma interface simples e recursos adequados para o projeto.

**Figura 31 - Aplicativo utilizado**

**Fonte: Autoria própria (2023)**

Após o primeiro acesso ao aplicativo, torna-se necessário configurar a conexão com o broker MQTT. O usuário deve clicar sobre o botão *"Setup a connection"* e, em seguida, sobre o ícone *"Add Dashboard"* para escolher o nome da Dashboard. O nome da dashboard é arbitrário, e neste caso, foi nomeado "Meu Dashboard".

Com o Dashboard criado, é necessário configurar a conexão MQTT de maneira semelhante ao realizado na interface web do Tasmota. Inicialmente, deve-se escolher um nome para o broker no campo *"Connection name"*. Neste caso, o nome escolhido foi "Meu Broker". No campo *"Client ID"*, é necessário informar um nome para o dispositivo em uso; neste caso, foi nomeado "Meu\_Smartphone".

No campo *"Broker Web/IP address"*, é preciso fornecer a URL, e no campo *"Port"*, a porta, ambos disponibilizados na página web do HiveMQ. Na lista *"Network protocol"*, deve-se selecionar "TCP-SSL" para estabelecer a conexão adequadamente com o broker da HiveMQ. Por fim, as credenciais *"Username"* e *"Password"* devem ser preenchidas conforme criadas anteriormente na página do broker, juntamente com a seleção da opção *"Auto connect"* para garantir que a conexão seja estabelecida automaticamente. A Figura 32 evidencia o painel MQTT do aplicativo após a sua configuração.

**Figura 32 - Configuração do MQTT no aplicativo**

Connection name \*  
Meu Broker

Client ID  
Meu\_Smartphone

Broker Web/IP address \*  
0d2a09bf4cc54bc5b200d490fdbb198e

Port \*  
8883

Network protocol  
TCP-SSL

Manage SSL certificates

Add Dashboard

Meu Dashboard

Additional options

Username  
Tasmota

Password  
.....

Auto connect

Keep alive \*  
60

Enable will message

Notify on disconnect

CANCEL SAVE

**Fonte: Autoria própria (2023)**

Com a conexão ao broker concluída, o usuário deve clicar sobre o broker criado para configurar os painéis destinados ao controle e monitoramento. Ao acessar o botão "Add Panel", uma aba com atuais 21 painéis será aberta. No entanto, serão utilizados apenas cinco: o *Switch*, um botão com duas posições que muda de estado com um clique; o *Gauge*, um medidor que indica o valor de um parâmetro por meio de uma agulha; o *Line Graph*, que desenha um gráfico de linha com base nos valores obtidos de um parâmetro; o *Date & Time Picker*, utilizado para selecionar dias e horários; e o *Layout Decorator*, que é utilizado como caixa de texto para organizar o painel.

Primeiramente, o controle de ligar e desligar o relé será configurado por meio de um *switch*. Ao clicar sobre o ícone do *switch*, a página de configuração será aberta. Os painéis compartilham uma estrutura semelhante, com o campo "Panel name" destinado ao nome do painel que será exibido na dashboard. O campo "Topic" é designado para o tópico onde a publicação de informações ocorrerá. Quanto ao "Subscribe Topic", sua função é realizar leituras de dados publicados no MQTT.

No contexto de um *switch*, é necessário especificar a informação a ser publicada no tópico anteriormente definido, dependendo da posição do *switch*.

“Payload on” é responsável pela mensagem publicada ao mudar o *switch* para a posição ligada, enquanto “Payload off” trata da mensagem publicada ao desligar o *switch*. Em casos em que a mensagem precisa ser publicada ou lida como JSON devido à presença de múltiplas informações no tópico específico, é necessário selecionar a caixa “Payload is JSON Data”. Neste cenário, os campos “JsonPath for subscribe” e “Json pattern for publish” podem ser preenchidos para configurar a leitura e a publicação, respectivamente. Além disso, a lista suspensa QoS (*Quality of Service*) deve ser configurada com o número 2 para garantir a entrega de apenas uma mensagem MQTT.

Assim, para configurar o *switch* encarregado do controle do relé, como destacado na Figura 33, o “Panel name” será definido como “Relé”. O campo “Topic” será “cmnd/lot/POWER”, enquanto o campo “Subscribe Topic” será “stat/lot/POWER”. As entradas “Payload on” e “Payload off” serão configuradas como “ON” e “OFF” respectivamente. Por fim, o QoS deve ser alterado para 2, e o usuário pode clicar sobre o botão “Save” para salvar a configuração do painel.

**Figura 33 - Configuração do *switch* do relé**

← Edit panel

Panel name \*  
Relé

Disable dashboard prefix topic

Topic \*  
cmnd/lot/POWER

Subscribe Topic  
stat/lot/POWER

Payload on \*  
ON

Payload off \*  
OFF

Switch color

Use icon switch

Enable notification

Payload is JSON Data

Show received timestamp

Show sent timestamp

Confirm before publish

Retain      QoS    2

CANCEL    SAVE

**Fonte: Autoria própria (2023)**

Para os painéis restantes, a configuração é realizada de maneira análoga, preenchendo os campos conforme os requisitos específicos de cada painel. O Quadro 1 apresenta os valores dos campos que devem ser preenchidos em todos os painéis.

**Quadro 1 - Inputs para os painéis**

<b>Switches</b>	
<b>Relé</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Relé
2. Topic	cmnd/lot/POWER
3. Subscribe Topic	cmnd/lot/POWER
4. Payload on	ON
5. Payload off	OFF
6. QoS	2
<b>Temporizador para ligar</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Temporizador para ligar
2. Topic	cmnd/lot/TIMER1
3. Payload on	{"Enable":1}
4. Payload off	{"Enable":0}
5. QoS	2
<b>Temporizador para desligar</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Temporizador para desligar
2. Topic	cmnd/lot/TIMER2
3. Payload on	{"Enable":1}
4. Payload off	{"Enable":0}
5. QoS	2
<b>Escolha de Tensão</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	127V - 220V
2. Topic	cmnd/lot/AdcParam
3. Payload on	7,0,7832.9,0.220,0.85

4. Payload off	7,0,7832.9,0.127,0.85
5. QoS	2
<b>Gauges</b>	
<b>Tensão</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Tensão (V)
2. Topic	tele/lot/SENSOR
3. Payload min	0
4. Payload max	360
5. Arc color 1 (verde)	130
6. Arc color 2 (azul)	230
7. JsonPath for subscribe	[\$['ANALOG'].['CTEnergy'].['Voltage']]
8. QoS	2
<b>Corrente</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Corrente (A)
2. Topic	tele/lot/SENSOR
3. Payload min	0
4. Payload max	100
5. Arc color 1 (verde)	10
6. Arc color 2 (amarelo)	20
7. JsonPath for subscribe	[\$['ANALOG'].['CTEnergy'].['Current']]
8. QoS	2
<b>Potência</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Potência (W)
2. Topic	tele/lot/SENSOR
3. Payload min	0
4. Payload max	5000
5. Arc color 1 (verde)	1000
6. Arc color 2 (amarelo)	1500

7. JsonPath for subscribe	<code>[\$['ANALOG'].['CTEnergy'].['Power']]</code>
8. QoS	2
<b>Line Graph</b>	
<b>Consumo</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Consumo (kWh)
2. Topic	tele/lot/SENSOR
3. JsonPath for subscribe	<code>[\$['ANALOG'].['CTEnergy'].['Energy']]</code>
4. QoS	2
<b>Data &amp; Time Picker</b>	
<b>Horário para ligar</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Horário para ligar
2. Topic	cmdnd/lot/TIMER1
3. Picker type	Time
4. Json pattern for publish	<code>{"Time":&lt;hour&gt;:&lt;minute&gt;}</code>
5. QoS	2
<b>Horário para desligar</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Horário para desligar
2. Topic	cmdnd/lot/TIMER2
3. Picker type	Time
4. Json pattern for publish	<code>{"Time":&lt;hour&gt;:&lt;minute&gt;}</code>
5. QoS	2
<b>Layout Decorator</b>	
<b>Seção de controle</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Controle
2. Title alignment	Center
3. Text size	25px
<b>Seção de monitoramento</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Monitoramento

2. Title alignment	Center
3. Text size	25px
<b>Seção dos Temporizadores</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Temporizadores
2. Title alignment	Center
3. Text size	25px
<b>Seção de escolha de Tensão</b>	
<b>Campo</b>	<b>Valor</b>
1. Panel name	Tensão
2. Title alignment	Center
3. Text size	25px

**Fonte: Autoria própria (2023)**

Após a criação de todos os painéis, é possível organizar os mesmos clicando sobre o ícone de três pontos localizado ao lado direito de cada painel. Com as opções “*Move before*” e “*Move after*”, é possível posicioná-los para cima e para baixo, respectivamente.

Dessa forma, conforme mostrado na Figura 34, conclui-se a configuração integral do aplicativo MQTT, conferindo controle preciso sobre o relé por meio do *switch*, permitindo alternar entre as condições ligado e desligado. O monitoramento abrange os *Gauges* de tensão, corrente e potência, além do *Line Graph* que traça o consumo acumulado ao longo do tempo.

Os dois *switches* dos temporizadores desempenham papel no controle da operação dos respectivos temporizadores, em conjunto com os dois *Data & Time Picker* responsáveis pelos horários. Caso o temporizador esteja ativo, a ação de ligar ou desligar é executada de acordo com as configurações estabelecidas. Por último, o *switch* facilita a escolha da tensão residencial, alternando entre 127 Volts e 220 Volts conforme necessário.

**Figura 34 - Painel do aplicativo**



**Fonte: Autoria própria (2023)**

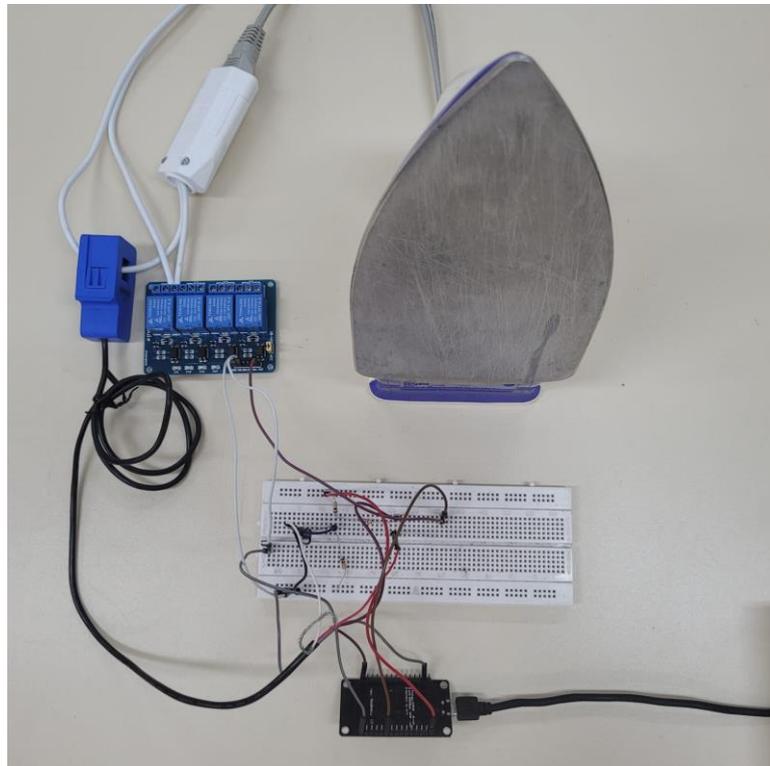
## 4 RESULTADOS

Este capítulo tem como objetivo apresentar os resultados obtidos com a montagem do sistema em bancada, além de sua validação no monitoramento e controle do sistema gerenciando um dispositivo residencial.

### 4.1 Resultados experimentais

Após a montagem do sistema, é necessário aferir as variáveis de potência, corrente e tensão para garantir que o sistema está fornecendo informações corretas ao usuário. Além disso, o sistema permite controlar o relé por meio de um botão do tipo *switch* ou por temporizadores. Ao ativá-los, é possível definir um horário para o dispositivo ligar ou desligar, permitindo assim um maior nível de gerenciamento e controle. Para os testes do sistema, foram utilizados dois equipamentos residenciais: uma sanduicheira e um ferro de passar. A Figura 35 evidencia o sistema conectado com um ferro de passar.

**Figura 35 - Bancada de testes**



**Fonte: Autoria própria (2023)**

A determinação da potência e do consumo de energia é realizada internamente pelo Tasmota, levando em consideração o valor da corrente medida em conjunto com a tensão selecionada pelo usuário. O cálculo da potência é executado utilizando a Equação 1, em que P representa a potência (W), V a tensão (V) e I a corrente (A).

$$P = V \cdot I \quad (1)$$

Para aferir o consumo de energia, é utilizada a Equação 2, em que E representa o consumo de energia (kWh), Pm é a potência média durante toda a medição (W) e ΔT corresponde ao tempo de medição (h).

$$E = \frac{(Pm \cdot \Delta T)}{1000} \quad (2)$$

A potência média (W) é calculada pelo somatório de todas as potências aferidas sobre a quantidade de leituras, conforme a Equação 3, com n sendo o número de leituras e P(i), com i de 1 a n, sendo cada valor de potência registrado pelo Tasmota.

$$Pm = \frac{1}{n} \sum_{i=1}^n P(i) \quad (3)$$

A Figura 36 ilustra o monitoramento e controle de um ferro de passar conectado a uma rede elétrica de 127V. Nessa configuração, o equipamento registra um consumo de corrente de 7,009A, correspondendo a uma potência consumida de aproximadamente 890W.

**Figura 36 - Monitoramento através do aplicativo**



**Fonte: Autoria própria (2023)**

## 4.2 Validação

Para a validação e aferição, foi utilizado um alicate amperímetro da marca Politem, cujas especificações técnicas apontam um erro máximo de  $\pm 2\%$ . Isso proporciona uma precisão satisfatória para a comparação com os resultados obtidos no painel do aplicativo. A Figura 37 ilustra o alicate amperímetro utilizado para aferição.

**Figura 37 - Alicate amperímetro**



**Fonte: Autoria própria (2023)**

Ao posicionar o alicate amperímetro em um dos fios condutores que alimentam o equipamento elétrico, neste caso do ferro de passar, é possível verificar a corrente elétrica consumida. Conforme apresentado nas Figuras 38 e 39, a variação entre a leitura do alicate amperímetro e a exibida no painel do aplicativo foi de aproximadamente 1,28%, mantendo-se dentro da margem de erro esperada para o alicate amperímetro. Isso valida a precisão dos dados obtidos demonstrando que o sistema exibe valores corretos em relação a medição real.

**Figura 38 - Comparação da corrente aferida**



**Fonte: Autoria própria (2023)**

**Figura 39 - Validação do sistema**

**Fonte: Autoria própria (2023)**

Além da aferição das medições das variáveis de tensão, corrente e potência, foram realizados diversos testes quanto ao acionamento (liga/desliga) remoto dos dispositivos juntamente com a verificação do controle de temporização alterando os tempos de liga/desliga a fim de garantir que operação correta e segura do sistema.

Com a validação do sistema, sabe-se que as informações fornecidas na interface do Tasmota ou na interface do aplicativo após a atualização MQTT estão corretas. Com isso, devem ser utilizados aparelhos elétricos respeitando os limites operacionais do relé, sensor, fiação elétrica e seus conectores. Para o sistema montado em bancada, o sensor SCT-013 suporta uma corrente máxima de 100A, enquanto a fiação de 4mm<sup>2</sup> tem um limite de aproximadamente 28A. Já o relé e os conectores possuem aplicações máxima de 10A, sendo, portanto, o ponto limitante.

Assim, é recomendado conectar aparelhos com consumo máximo de 10A e voltagem de 127V ou 220V. Isso permite o uso de aparelhos com potência máxima de 2200W, que possibilita o gerenciamento do consumo de energia e controle de diversos dispositivos residenciais, respeitando os limites de segurança operacional dos componentes utilizados no sistema.

## 5 CONCLUSÃO

O presente trabalho atingiu seus objetivos, centrados no desenvolvimento de um sistema IoT para monitoramento e controle de dispositivos residenciais. A sincronização eficaz do sistema com um aplicativo gratuito proporciona aos usuários a capacidade de monitorar o consumo de energia dos dispositivos e controlar suas operações remotamente. Os objetivos específicos, desde o estudo das características de sistemas IoT até a validação do sistema com resultados experimentais, foram meticulosamente alcançados, consolidando a fundamentação teórica na prática.

A justificativa inicial, ancorada no impacto do avanço tecnológico no setor residencial e na ascensão das *smart homes*, foi plenamente respaldada pelo desenvolvimento do sistema proposto. A ênfase na acessibilidade para pessoas com deficiência ou mobilidade reduzida destaca a relevância social do trabalho. O sistema de baixo custo, conectado a um aplicativo gratuito, concretiza a proposta de facilitar o controle residencial por meio de dispositivos móveis, proporcionando aos usuários acesso intuitivo e remoto.

A contribuição ambiental do trabalho é evidente, uma vez que a implementação de estratégias de controle energético visa não apenas a economia financeira, mas também a redução dos impactos ambientais relacionados à geração de energia não renovável. Assim, a importância do tema se reforça, não apenas pelos benefícios imediatos oferecidos à sociedade, mas também pelo papel potencialmente transformador na mitigação dos impactos ambientais. Em síntese, este trabalho não só cumpriu seus objetivos como também evidenciou a importância das soluções tecnológicas inovadoras na promoção da eficiência, acessibilidade e sustentabilidade no contexto residencial.

### 5.1 Trabalhos futuros

Para futuros trabalhos, é recomendável substituir o módulo relé de 10A por um módulo com uma capacidade de corrente superior, como os modelos com suporte a 30A que são facilmente encontrados no mercado. Ademais, a troca dos plugues macho e fêmea por conectores de 20A resultará em um aumento do limite atual de 10A para 20A, dobrando a capacidade de potência. Isso viabiliza a conexão de dispositivos de até 4400W utilizando uma rede de 220V.

Outra sugestão de melhoria consiste em utilizar todos os canais disponíveis no relé para controlar diferentes equipamentos elétricos. Isso expandirá o controle residencial, não se limitando a apenas um dispositivo. A inclusão de botões de pressão (*push buttons*) no sistema eletrônico para controlar individualmente cada canal do módulo relé seria uma adição valiosa. Isso permitiria o controle manual mesmo sem a presença de um dispositivo móvel, oferecendo maior flexibilidade ao sistema.

Por último, considera-se a substituição da protoboard por uma placa de circuito impresso (PCI). Essa troca aumentará a durabilidade do projeto e o tornará mais compacto e seguro. Além disso, a utilização de uma placa de circuito impresso proporcionará uma montagem mais estável e confiável do sistema.

## REFERÊNCIAS

ASHTON, Kevin. **Making sense of IoT**: How the Internet of Things became humanity's nervous system. [S.l]: Hewlett packard, 2017. Disponível em: <[http://book.itep.ru/depositary/iot/HPE\\_Aruba\\_IoT\\_eBook\\_English.pdf](http://book.itep.ru/depositary/iot/HPE_Aruba_IoT_eBook_English.pdf)>. Acesso em: 14 de mai. de 2023.

CEZAR, Edvandro Roberto da Silva. A domótica criando conforto e segurança. **Unianchieta**, Jundiaí, v. 3, n. 2, p.10, jul./dez. 2020. Disponível em: <<https://revistas.anchieta.br/index.php/RevistaUbiquidade/article/view/1674/1499>>. Acesso em: 5 de jun. de 2023.

COSTA, Manuella de Farias Nardelli. Internet das coisas: A proteção da privacidade em um mundo conectado. **Enap**, Brasília, v. 1, n. 1, p.44, 2019. Disponível em: <<https://repositorio.enap.gov.br/jspui/handle/1/4125>>. Acesso em: 14 de mai. de 2023.

GUIMARÃES, Fábio. Firmware e IDE – Aula 4 – MC. **Mundo projetado**, 18 de fev. de 2020. Disponível em: <https://mundoprojetado.com.br/firmware-e-ide/>. Acesso em: 12 de mai. de 2023.

IoT: até 2025, mais de 27 bilhões de dispositivos estarão conectados. **ABINC**, 2022. Disponível em: <https://abinc.org.br/iot-ate-2025-mais-de-27-bilhoes-de-dispositivos-estarao-conectados/>. Acesso em: 13 de mai. de 2023.

JÚNIOR, Sérgio Luiz S.; FARINELLI, Felipe A. **Domótica**: Automação residencial e casas inteligentes com Arduino e Esp826. São Paulo: Editora Saraiva, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788536530055/pageid/0>>. Acesso em: 03 abr. 2023.

LENZ, Maikon L.; TORRES, Fernando E. **Microprocessadores**. Porto Alegre: Grupo A, 2019. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788595029736/pageid/0>>. Acesso em: 14 de mai. de 2023.

MASCHIETTO, Luís G. et al. **Arquitetura e Infraestrutura de IoT**. Porto Alegre: Grupo A, 2021. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9786556901947/pageid/0>>. Acesso em: 6 de mai. de 2023.

MORAES, Alexandre de; HAYASHI, Victor T. **Segurança em IoT**: Entenda os riscos e ameaças em Internet das Coisas. Rio de Janeiro: Editora Alta Books, 2021. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788550816548/epubcfi/6/2/%5B%3Bvnd.vst.idref%3Dcover%5D!/4/2/2%4051:2>>. Acesso em: 1 de mai. de 2023.

MORAIS, Izabelly Soares de et al. **Introdução a Big Data e Internet das Coisas (IoT)**. Porto Alegre: Grupo A, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788595027640/pageid/0>>. Acesso em: 10 de abr. de 2023.

OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P. **Arduino Descomplicado**: Como Elaborar Projetos de Eletrônica. São Paulo: Editora Saraiva, 2017. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788536518114/pageid/2>>. Acesso em: 15 de abr. de 2023.

PRÓSPERO, Israel. Microcontroladores: o que são e aplicações. **EESC jr**, 18 de ago. de 2020. Disponível em: <https://eescjr.com.br/blog/microcontroladores-o-que-sao-e-aplicacoes/#:~:text=Quando%20analisamos%20os%20microcontroladores%20mais,Arduino%2C%20ESP%20e%20Raspberry%20Pi> . Acesso em: 10 de abr. de 2023.

Protocolo MQTT: O Que é, Como Funciona e Vantagens. **Automação Industrial**, 2022. Disponível em: <https://www.automacaoindustrial.info/mqtt/>. Acesso em: 18 de nov. de 2023.

ROSNER, Gilad; KENNEALLY, Erin. Clearly Opaque: Privacy Risks of the Internet of Things. **SSRN**, [S.l]. v. 1, n. 1, p.151, 2018. Disponível em: <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3320656](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3320656)>. Acesso em: 14 de mai. de 2023.

SMART KITS. **Smart Kits**. Descrição do sonoff. Disponível em: <https://www.smartkits.com.br/sonoff-mini-r2#:~:text=O%20Sonoff%20%C3%A9%20um%20rel%C3%A9,praticidade%20e%20%C3%B3timo%20custo%20benef%C3%ADcio>. Acesso em: 12 de mai. de 2023.

TASMOTA. **Tasmota**. Sobre o tasmota. Disponível em: <https://tasmota.github.io/docs/About/>. Acesso em: 29 de mar. de 2023.

TOLEDO, Victor. Tudo sobre o Raspberry Pi 4: veja especificações e preço do mini PC. **TechTudo**, 25 de set. de 2019. Disponível em: <https://www.techtudo.com.br/noticias/2019/09/tudo-sobre-o-raspberry-pi-4-veja-especificacoes-e-preco-do-mini-pc.ghtml> . Acesso em: 13 de abr. de 2023.

TORRES, Guilherme Gerzson. **Tecnologia Assistiva para Detecção de Quedas**: desenvolvimento de sensor vestível integrado ao sistema de casa inteligente. 2018. 71 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/180824/001072583.pdf?sequenc e=1>>. Acesso em: 16 de mai. de 2023.