



UNIVERSIDADE FEDERAL DO MARANHÃO
Coordenação do Curso de Engenharia da Computação

Carlos Eduardo Nascimento Cajado

**DESENVOLVIMENTO DE SISTEMA PARA
GERÊNCIA E AUTOMAÇÃO DE
LABORATÓRIO USANDO ESP8266**

São Luís
2025

Carlos Eduardo Nascimento Cajado

DESENVOLVIMENTO DE SISTEMA PARA GERÊNCIA E AUTOMAÇÃO DE LABORATÓRIO USANDO ESP8266

Monografia apresentada à Coordenação do
Curso de Engenharia da Computação da
Universidade Federal do Maranhão, como
parte dos requisitos necessários para obtenção
do título de Bacharel em Engenharia da
Computação.

Orientador: Prof. Dr. Paulo Rogério de Almeida Ribeiro

São Luís

2025

Carlos Eduardo Nascimento Cajado

DESENVOLVIMENTO DE SISTEMA PARA GERÊNCIA E AUTOMAÇÃO DE LABORATÓRIO USANDO ESP8266

Monografia apresentada à Coordenação do
Curso de Engenharia da Computação da
Universidade Federal do Maranhão, como
parte dos requisitos necessários para obtenção
do título de Bacharel em Engenharia da
Computação.

Trabalho _____ em São Luís, 24 de fevereiro de 2025:

**Prof. Dr. Paulo Rogério de Almeida
Ribeiro**
Orientador

Prof. Dr. Pedro Baptista Fernandes
Examinador

**Profa. Dr. Luis Claudio de Oliveira
Silva**
Examinador

São Luís
2025

Agradecimentos

Primeiramente, agradeço a Deus por me conceder saúde e força para superar todas as dificuldades. À universidade, especialmente ao corpo docente, que contribuiu diretamente para minha jornada acadêmica e profissional. Ao meu orientador, Paulo Rogério, pelo suporte, correções e incentivo ao desenvolvimento deste projeto. Agradeço profundamente à minha família, em especial aos meus pais e à minha esposa amada, pelo amor, incentivo e apoio incondicional. E a todos que, de alguma forma, participaram da minha formação, direta ou indiretamente, meu sincero agradecimento.

*“E conhecereis a verdade,
e a verdade vos libertará.”*

João 8:32, em *Bíblia Sagrada*

Resumo

O controle automatizado possui grande relevância nas áreas de engenharia e ciência, assim como no âmbito residencial, onde a automação integra sistemas tecnológicos para oferecer controle, conforto e eficiência, abrangendo segurança, comunicação e consumo energético. Sua aplicabilidade se amplia com dispositivos conectados via *Wi-Fi*, com a possibilidade de gerenciamento remoto por smartphones, tablets ou computadores. No contexto acadêmico, a gestão eficiente de laboratórios é essencial para a otimização de recursos e o aprimoramento da aprendizagem e da pesquisa. Diante desse cenário, este trabalho propõe a integração de desenvolvimento *web* e automação para modernizar o gerenciamento das salas e laboratórios de Informática da Engenharia da Computação da Universidade Federal do Maranhão (UFMA). Para isso, é desenvolvido um sistema baseado na linguagem PHP, utilizando o *framework* Laravel, e integrado ao módulo ESP8266. A comunicação entre a aplicação e os dispositivos ocorre por meio do servidor Mosquitto MQTT, que atua como intermediário entre as tecnologias. Ademais, para a coleta de resultados, foram realizados os cadastros e as configurações necessárias para o horário de aula de um professor, localizada no laboratório InovTec (UFMA), além de outras automações para os períodos de utilização do laboratório. O processo de coleta de dados e o funcionamento do sistema ocorreram ao longo de três semanas, entre os dias 12 e 31 de janeiro de 2025, durante as quais um conjunto de dispositivos com dois tipos de acionamento: relés ou infravermelho foram controlados via automações cadastradas no sistema *web*. A solução, além de buscar a automatização do controle dos dispositivos a um custo acessível, visa, também, permitir a coleta de dados relevantes, para uma gestão mais inteligente e um melhor consumo energético, conforme as tendências da transformação digital e da Internet das Coisas (IoT).

Palavras-chave: Automação, Internet das Coisas (IoT), Controle Inteligente, Gerenciamento de Laboratórios.

Abstract

Automated control is highly relevant in the fields of engineering and science, as well as in the residential domain, where automation integrates technological systems to provide control, comfort, and efficiency, covering security, communication, and energy consumption. Its applicability expands with devices connected via Wi-Fi, enabling remote management through smartphones, tablets, or computers. In the academic context, the efficient management of laboratories is essential for optimizing resources and enhancing both learning and research. In this context, this work proposes the integration of web development and automation to modernize the management of classrooms and computer laboratories in the Computer Engineering program at the Federal University of Maranhão (UFMA). To achieve this, a system was developed using PHP, based on the Laravel framework, and integrated with the ESP8266 module. Communication between the application and the devices occurs through the Mosquitto MQTT server, which acts as an intermediary between the technologies. Furthermore, to collect results, the necessary registrations and configurations were carried out for a professor's class schedule, located in the InovTec laboratory (UFMA), along with other automations for the laboratory's usage periods. The data collection process and system operation took place over three weeks, from January 12 to January 31, 2025, during which a set of devices with two types of activation—relays or infrared—was controlled via automations registered in the web system. In addition to aiming for cost-effective automation of device control, the solution also seeks to enable the collection of relevant data for smarter management and better energy consumption, aligning with digital transformation trends and the Internet of Things (IoT).

Keywords: Automation, Internet of Things (IoT), Intelligent Control, Laboratory Management.

Lista de ilustrações

Figura 1 – Fluxo Comandos SQL.	16
Figura 2 – Comparativo entre os <i>frameworks</i> Laravel, CodeIgniter e Zend.	18
Figura 3 – Módulo ESP8266 (NodeMcu v3)	24
Figura 4 – Instalação do Tasmotizer	29
Figura 5 – Primeiro acesso Tasmota	30
Figura 6 – Configurações gerais Tasmota	30
Figura 7 – Configuração dos pinos ESP8266 no Tasmota	31
Figura 8 – Mapas do Site - Administrador e Professor.	33
Figura 9 – Diagrama ER do banco de dados – Parte (a).	35
Figura 10 – Diagrama ER do banco de dados – Parte (b).	36
Figura 11 – Montagem e conexões do <i>Hardware</i>	39
Figura 12 – Esquemático elétrico <i>Hardware</i>	39
Figura 13 – Comunicação entre as tecnologias	41
Figura 14 – Acesso à infraestrutura (Ambiente de produção)	42
Figura 15 – Laboratório Inovtec	42
Figura 16 – <i>Hardware</i> Físico no laboratório	43
Figura 17 – Tela de login	44
Figura 18 – <i>Sidebar</i>	45
Figura 19 – Tela Inicial -(<i>Dashboard</i>).	46
Figura 20 – Cadastro, edição e exclusão de registros.	47
Figura 21 – Cadastro Grupo de Dispositivos.	47
Figura 22 – Cadastro de Dispositivo IR.	48
Figura 23 – Cadastro de Automação.	49
Figura 24 – Exibição da Automação.	49
Figura 25 – <i>Dashboard</i> do Professor.	50
Figura 26 – Automação do Professor.	50
Figura 27 – Grupo de Dispositivos.	51
Figura 28 – Automação 4: Registros de acionamento	52
Figura 29 – Console da aplicação Tasmota em inscrição (MQTT)	52
Figura 30 – Automação 4: Acionamento <i>Hardware</i>	53

Lista de tabelas

Tabela 1 – Parâmetros para conexão (CONNECT)	19
Tabela 2 – Parâmetros da Mensagem (CONNACK)	20
Tabela 3 – Parâmetros da Mensagem (SUBSCRIBE)	20
Tabela 4 – Parâmetros da Mensagem PUBLISH	20
Tabela 5 – Descrição das Telas do Sistema (Administrador) e suas Funcionalidades	34
Tabela 6 – Criação do <i>Hardware</i>	38
Tabela 7 – Custo do projeto	40
Tabela 8 – Registros de acionamento da Automação 1 no período de observação .	51

Lista de abreviaturas e siglas

ANSI	Instituto Nacional Americano de Padrões
CCET	Centro de Ciências Exatas e Tecnologia
CGI	Common Gateway Interface
DCS	Sistemas de Supervisão e Controle Distribuído
EDL	Eclipse Distribution License
EPL	Eclipse Public License
FNR	Requisitos Não Funcionais
FR	Requisitos Funcionais
GPIO	General Purpose Input/Output
HTML	HyperText Markup Language
IoT	Internet das Coisas
JSON	JavaScript Object Notation
OLE	OLE for Process Control
OOP	Programação Orientada a Objeto
OPC	Open Platform Communications
PHP	Hypertext Preprocessor
PID	Proporcional, Integral, Derivativo
PLCs	Controladores Lógicos Programáveis
QoS	Qualidade de Serviço
RISC	Reduced Instruction Set Computer
SQL	Structured Query Language
UFMA	Universidade Federal do Maranhão
XML	Extensible Markup Language

Sumário

1	INTRODUÇÃO	12
1.1	Justificativa	13
1.2	Objetivos	13
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	A linguagem PHP	15
2.2	Linguagem SQL	15
2.3	Framework Laravel	17
2.4	Protocolo MQTT	18
2.4.1	Software Mosquitto	20
2.5	Automação	21
2.6	Dispositivos IOT	22
2.7	Eficiência energética	22
2.8	Microcontroladores	23
2.9	Módulo ESP8266	24
2.10	Sonoff	25
2.11	Firmware Tasmota	25
3	METODOLOGIA	26
3.1	Levantamento dos requisitos	26
3.1.1	Requisitos funcionais (FR)	26
3.1.2	Requisitos não funcionais (FNR)	27
3.2	Criação do servidor MQTT	28
3.2.1	Configuração para o Broker Mosquitto	28
3.3	Instalação do Tasmota no NodeMCU (ESP8266)	29
3.3.1	Configuração Tasmota no NodeMCU	29
3.4	Desenvolvimento do sistema <i>web</i>	31
3.4.1	Banco de dados do projeto	35
3.5	Montagem do <i>hardware</i>	37
3.6	Custo do projeto	40
3.7	Integração, comunicação e ambiente de produção	40
3.7.1	Ambiente de produção	41
4	RESULTADOS	44

4.1	Páginas do sistema <i>web</i> em produção	44
4.1.1	Área do administrador	45
4.1.2	Área do Professor	50
4.2	Rotinas das automações	51
5	CONCLUSÃO	54
	REFERÊNCIAS	55

1 Introdução

No século XVIII, iniciou-se uma era pioneira no controle automatizado com o desenvolvimento do regulador centrífugo por James Watt, projetado para gerenciar a velocidade de máquinas a vapor (FRANÇA, 2018). Na contemporaneidade, a relevância do controle automatizado é inquestionável em diversas áreas da engenharia e ciência, pois esse assume um papel crucial em sistemas de veículos espaciais, na robótica, nos avançados processos de manufatura, assim como em operações industriais que exigem a precisa regulação de variáveis como temperatura, pressão, umidade, viscosidade e vazão (OGATA; COIMBRA; TANNURI, 2010).

Ademais, a automação, para o uso residencial, é caracterizada como um conjunto de serviços oferecidos por sistemas tecnológicos integrados, destinados a atender às necessidades essenciais de uma residência ou ambiente, com o objetivo primordial de proporcionar controle e conforto aos usuários. Essas demandas abrangem setores como segurança, comunicação, eficiência energética, conforto luminotécnico e controle de temperaturas (MURATORI; BÓ, 2017).

Sua aplicabilidade em diversas áreas simultâneas torna-se efetiva ao ser integrada a dispositivos que se comunicam por meio de smartphones, tablets ou computadores, utilizando conexões como *Wi-Fi* ou outros meios de comunicação. Essa versatilidade abrange setores diversos, capacitando a criação de uma residência inteligente e eficientemente controlada (LOPES, 2022).

Além disso, no contexto do controle de laboratórios acadêmicos, como apresentado por Marino Tedesco (TEDESCO et al., 2009), a eficiência no gerenciamento desses espaços é fundamental para garantir um ambiente adequado tanto para o controle do uso dos recursos da universidade quanto para a aprendizagem e a pesquisa. Um exemplo disso é o desperdício de energia causado por lâmpadas e aparelhos de ar-condicionado que permanecem ligados por horas sem necessidade. Além do impacto financeiro e ambiental, a gestão manual desses dispositivos exige a dedicação de funcionários, tornando-se uma tarefa exaustiva devido à abundância de salas e laboratórios.

Diante desse cenário, surge a necessidade de aprimorar o gerenciamento das salas e laboratório de Informática da Engenharia da Computação da Universidade Federal do Maranhão (UFMA). A proposta aspira integrar tecnologia e automação, visando otimizar não apenas o controle dos dispositivos, mas também coletar dados relevantes que possibilitem uma gestão mais inteligente e estratégica. Essa iniciativa está consoante às atuais tendências de transformação digital e Internet das Coisas (IoT), com o intuito de

modernizar os ambientes acadêmicos e promover o desenvolvimento tecnológico (ALVES et al., 2022).

1.1 Justificativa

O presente projeto visa desenvolver uma aplicação integrada com o módulo ESP8266 para gerenciamento, automação, e controle de usuários para as salas e laboratórios de Informática. É justificado não só pela necessidade de otimizar recursos como eletricidade e dependência ao acesso constante para controle de iluminação e climatização dos laboratórios, como também para gerenciamento partindo dos dados sobre o uso das instalações.

Além disso, a aplicação se alinha com as demandas contemporâneas de transformação digital e Internet das Coisas (IoT), promovendo a integração de dispositivos e aprimorando a conectividade no ambiente acadêmico. Assim, a realização deste projeto não apenas atende às necessidades específicas das salas e laboratórios, mas também representa uma contribuição valiosa para a modernização e inovação dos ambientes acadêmicos, preparando-os para os desafios tecnológicos do futuro.

1.2 Objetivos

1.2.1 Objetivo geral

O projeto visa automatizar e gerenciar completamente o acesso às salas e laboratórios da Engenharia da Computação, baseando-se no uso de microcontroladores e IoT. Inicialmente, o projeto-piloto será implementado no espaço InovTec, localizado no Centro de Ciências Exatas e Tecnologia (CCET) do campus São Luís da Universidade Federal do Maranhão (UFMA).

1.2.2 Objetivos específicos

Destaca-se como objetivos específicos deste trabalho:

- Realizar o levantamento dos requisitos funcionais e não funcionais para atender as necessidades do laboratório.
- Integrar um módulo ESP8266 ou ESP32 com uma aplicação *web* para gerenciar de forma eficiente as salas e laboratórios da Engenharia da Computação.
- Coletar e analisar dados do laboratório para fornecer informações relevantes, aprimorando o gerenciamento nas tomadas de decisões baseadas em dados.

-
- Integrar a aplicação ao laboratório conforme os princípios da Internet das Coisas (IoT), possibilitando uma conectividade eficaz e comunicação entre dispositivos para aprimorar a operação geral do laboratório.
 - Projetar a aplicação para gerenciar dispositivos, usuários e rotinas de automação.
 - Projetar um *hardware* eficiente, implementável e com custo aceitável.

2 Fundamentação Teórica

2.1 A linguagem PHP

O *Personal Home Page* (PHP), uma linguagem de roteiros projetada para a geração dinâmica de dados na *Web*, passou por um processo evolutivo significativo até atingir sua forma atualmente reconhecida. Surgiu em 1994 idealizada de Rasmus Lerdof, desenvolvido em linguagem C com base em conjuntos binários. Inicialmente concebido para monitorar currículos pessoais *online*, o PHP evoluiu consideravelmente por meio de implementações e integração com bancos de dados (CANTO, 2011).

O PHP é compatível com a maioria dos sistemas operacionais, como Linux, várias variantes do Unix (HP-UX, Solaris, OpenBSD), Microsoft Windows, macOS, RISC OS, entre outros. Além disso, oferece suporte a diversos servidores *web*, incluindo Apache, IIS, *lighttpd* e *nginx*, opera tanto como um módulo quanto como um processo no servidor, em resposta às solicitações do navegador, utilizando a *common gateway interface* (CGI, do inglês) (PHP.NET, 2025).

Ademais, é possível optar por programação estruturada, programação orientada a objeto (OOP, do inglês) ou uma combinação de ambas. Destaca-se, ainda, que ele não se restringe à geração de *HyperText Markup Language* (HTML), pois é capaz de criar tipos de arquivos complexos, como imagens e PDFs, criptografar dados, enviar *e-mails* e gerar arquivos de textos como JavaScript, *Object Notation* (JSON) ou *Extensible Markup Language* (XML) (PHP.NET, 2025).

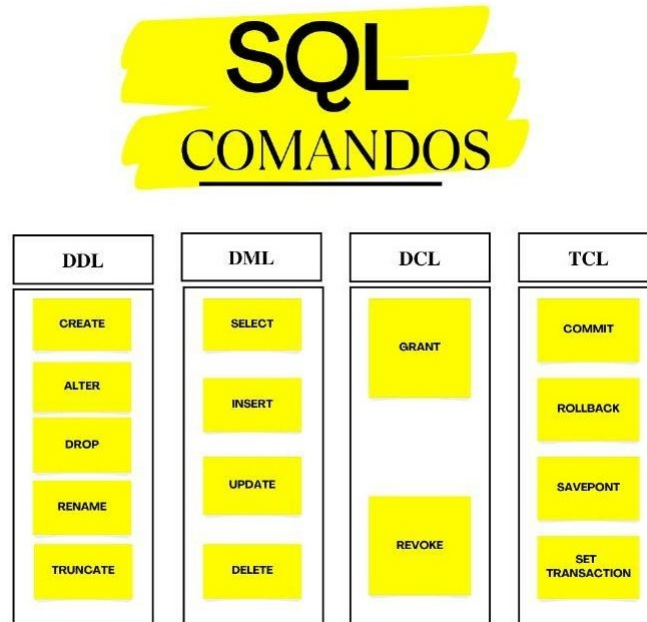
2.2 Linguagem SQL

Partindo da necessidade de armazenamento de informações, intensificaram-se os estudos para o desenvolvimento de uma linguagem que se adaptasse ao modelo relacional. O *Structured Query Language* (SQL) foi criado com esse propósito, sendo que o modelo relacional foi introduzido pelo matemático Edgar F. Codd, membro do laboratório de pesquisa da IBM, no ano de 1970.

Atualmente o SQL é a linguagem que se padronizou ao decorrer dos anos pelo Instituto Nacional Americano de Padrões (ANSI, do inglês) a maioria dos bancos de dados seguem criteriosamente esta padronização (SOUZA, 2015). Soma-se ainda, que o SQL é uma linguagem especificamente concebida para permitir que as pessoas criem, adicionem, recuperem e manipulem informações em um banco de dados (ELMASRI et al.,

2005). A linguagem SQL pode ser dividida em subconjuntos bem definidos, na Figura 1 é apresentado os subconjuntos de comandos de possível execução:

Figura 1 – Fluxo Comandos SQL.



Fonte: Autor.

Onde, tem-se:

- DDL (*Data Definition Language*) — Esses comandos são usados para definir o esquema da base de dados tendo ações, lidam com a criação do esquema do banco de dados e suas modificações posteriores, como alteração na propriedade da coluna da tabela ou exclusão de dados de uma tabela.
- DML (*Data Manipulation Language*) — É uma das partes mais utilizadas, visto que contém as instruções para fazer a manipulação de dados armazenados, bem como a exclusão, atualização ou inserção de novos dados.
- DCL (*Data Control Language*) — Parte responsável por fazer a configuração do controle de usuários da base de dados, tendo comandos para alterar permissões de acesso de usuários ou podendo removê-los.
- TCL (*Transaction Control Language*) — São usados para ajudar o usuário a gerenciar as transações que ocorrem em um banco de dados. Sendo transação de uma alteração proveniente de comandos SQL, assim o desenvolvedor pode usar esses comandos, caso tenha sucesso a alteração é feita permanente e caso haja falha é possível reverter algumas das alterações que já foram realizadas.

- DQL (Data Query Language) — Usada para buscar os dados do banco de dados usando apenas um comando, junto a ele pode ser atribuído uma condição descrita pela cláusula “*where*”.

Um banco de dados é comumente gerenciado por um Sistema de Gerenciamento de Banco de Dados (SGBD). Os tipos mais comuns de bancos são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes, podendo ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados (Oracle, 2025).

2.3 Framework Laravel

Entre os anos de 2004 e 2006, com o lançamento das versões 5.0 e 5.1 do PHP, surgiram os primeiros *frameworks* PHP, focados na orientação a objetos e na reutilização de código. O CodeIgniter, inspirado no Rails do Ruby, foi um dos mais populares na época, mas enfrentou problemas de manutenção quando a empresa responsável descontinuou o suporte ao projeto. Isso criou uma oportunidade para Taylor B. Otwell criar o Laravel em 2011 (CARDOSO, 2023).

Posteriormente, em 2013, com a versão 4, o Laravel foi reescrito com componentes do Symfony, adotando a arquitetura MVC (Model-View-Controller) que é a divisão dos sistemas em três camadas: *Model*, *View* e *Controller*.

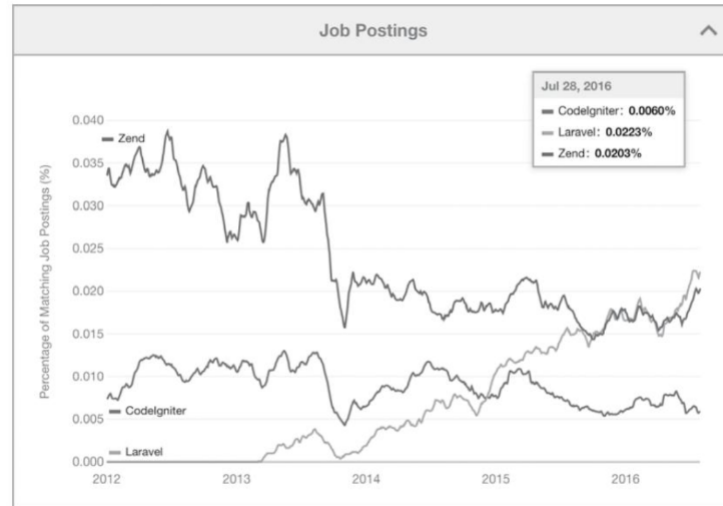
Atualmente, o Laravel é um *framework web* PHP com uma sintaxe expressiva e elegante, projetado para oferecer uma estrutura sólida para a criação de aplicações e APIs. Além de facilitar o desenvolvimento com recursos como injeção de dependência, abstração de banco de dados, filas, trabalhos agendados e testes, ele se destaca pela facilidade de aprendizado, sendo ideal não somente para iniciantes, como também para desenvolvedores experientes, que buscam construir aplicações profissionais e robustas (Laravel, 2023).

Ademais, é importante salientar que o Laravel é um *framework* progressivo, projetado para evoluir com o desenvolvedor. Ele oferece uma vasta gama de recursos, incluindo um amplo acervo de literatura, documentação detalhada, guias e tutoriais em vídeo, que facilitam o aprendizado de novos desenvolvedores. Além disso, o Laravel é altamente escalável, preparado para lidar com grandes cargas de trabalho, suportando sistemas de *cache* rápidos como Redis e permitindo escalabilidade horizontal (Laravel, 2023).

O Laravel conta com uma comunidade altamente ativa e uma excelente documentação, fatores que contribuíram significativamente para sua popularização entre as equipes de desenvolvimento (GABARDO, 2017). A Figura 2 apresenta um comparativo entre os *frameworks* PHP, onde se observa o crescimento da demanda por profissionais

especializados em Laravel em comparação aos *frameworks* CodeIgniter e Zend, entre os anos de 2012 a 2017.

Figura 2 – Comparativo entre os *frameworks* Laravel, CodeIgniter e Zend.



Fonte: (GABARDO, 2017).

Esse crescimento destaca a popularidade em ascensão do Laravel, que atualmente é o *framework* PHP mais utilizado do mundo (CARDOSO, 2023).

2.4 Protocolo MQTT

Desenvolvido pela IBM no final da década de 1990, o *Message Queuing Telemetry Transport* (MQTT) foi criado para conectar sensores em oleodutos a satélites. Como protocolo de mensagens, ele facilita a comunicação assíncrona entre as partes, permitindo que o remetente e o destinatário não precisem estar sincronizados no tempo ou no espaço. Isso torna o MQTT ideal para redes não confiáveis e altamente escaláveis.

É importante salientar que um recurso essencial desse protocolo é seu modelo de publicação e assinatura, que desvincula o publicador e o consumidor de dados, oferecendo flexibilidade e eficiência na troca de informações (IBM Developer, 2025).

O paradigma de publicação/assinatura é um modelo de comunicação em que os participantes assumem dois papéis distintos: publicadores e assinantes. Publicadores enviam mensagens para um tópico centralizado, enquanto assinantes registram interesse em tópicos específicos. Quando uma mensagem é publicada em um tópico, todos os assinantes desse tópico a recebem automaticamente.

Ademais, diferentemente da abordagem tradicional de solicitação-resposta, em que dispositivos IoT transferem dados para um servidor ou dispositivo final após uma

solicitação, o modelo de publicação/assinatura permite uma comunicação mais eficiente e distribuída (SOFIA; MENDES, 2019).

O protocolo MQTT é amplamente utilizado na Internet das Coisas (IoT) por sua simplicidade e leveza, sendo ideal para redes com baixa largura de banda, altas latências ou conexões instáveis. Esse protocolo não só oferece confiabilidade, mas também proporciona garantias de entrega, reduzindo os requisitos de *hardware* e o consumo de largura de banda da rede (LIU et al., 2020).

Em consonância a isso, temos que as bibliotecas e ferramentas MQTT oferecem métodos simples para manipular mensagens diretamente, preenchendo automaticamente alguns campos obrigatórios, como os identificadores (IDs) de mensagem e de cliente. No início da comunicação, o cliente é responsável por se conectar ao *broker* enviando uma mensagem “CONNECT”, solicitando o estabelecimento de uma conexão entre o cliente e o *broker*. A Tabela 1 apresenta os parâmetros e suas descrições enviados na mensagem “CONNECT” (IBM Developer, 2025).

Tabela 1 – Parâmetros para conexão (CONNECT)

Parâmetro	Descrição
cleanSession	Especifica se a conexão é persistente. Uma sessão persistente armazena todas as assinaturas e mensagens potencialmente perdidas dependendo do Nível de qualidade de serviço (QoS, do inglês) no <i>broker</i> .
username	Credenciais de autenticação e autorização para o <i>broker</i> . (Usuário).
password	Senha de autenticação e autorização para o <i>broker</i> .
lastWillTopic	Caso a conexão seja interrompida inesperadamente, o <i>broker</i> publicará automaticamente uma mensagem de “último desejo” no tópico.
lastWillQos	Nível de QoS para a mensagem de “último desejo”. Esta mensagem serve como um aviso para outros clientes sobre a desconexão do cliente original.
lastWillMessage	Conteúdo da mensagem de “último desejo”.
keepAlive	Intervalo de tempo em que o cliente envia um ping ao <i>broker</i> para manter a conexão ativa.

Fonte: Adaptado de (IBM Developer, 2025).

Ao conectar-se, o cliente recebe uma mensagem de retorno, a “CONNACK”, cujos parâmetros de retorno podem ser consultados na Tabela 2.

Tabela 2 – Parâmetros da Mensagem (CONNACK)

Parâmetro	Descrição
sessionPresent	Indica se já existe uma sessão persistente, incluindo tópicos assinados e entrega de mensagens perdidas.
returnCode	Status da conexão: 0 para sucesso, com outros valores especificando a causa de falha.

Fonte: Adaptado de ([IBM Developer, 2025](#)).

Após estabelecer a conexão, o cliente pode enviar uma ou mais mensagens “*SUBSCRIBE*” ao *broker*, especificando os tópicos dos quais deseja receber mensagens. Cada mensagem “*SUBSCRIBE*” pode incluir os parâmetros listados na Tabela 3.

Tabela 3 – Parâmetros da Mensagem (SUBSCRIBE)

Parâmetro	Descrição
QoS	Nível de qualidade de serviço para a entrega das mensagens.
topic	Tópico para assinar. Os tópicos podem ter múltiplos níveis, ou seja, “dw/demo” ou “ibm/cloud/mqtt”.

Fonte: Adaptado de ([IBM Developer, 2025](#)).

Posteriormente, no processo de envio de uma mensagem “*PUBLISH*” ao servidor, cada mensagem inclui um tópico e uma carga útil (*payload*) de dados, conforme descrito na Tabela 4. Posteriormente, o servidor encaminha a mensagem para todos os clientes inscritos neste tópico.

Tabela 4 – Parâmetros da Mensagem PUBLISH

Parâmetro	Descrição
Nome do Tópico	O tópico sob o qual a mensagem será publicada.
QoS	O nível de QoS para a entrega da mensagem. Ele especifica o grau de confiabilidade e a prioridade com que as mensagens são transmitidas e entregues em uma rede.
Retain Flag	Indica se o <i>broker</i> deve reter a mensagem como a última mensagem conhecida para este tópico.
Payload	Os dados contidos na mensagem, que podem ser uma string de texto ou um blob de dados binários.

Fonte: Adaptado de ([IBM Developer, 2025](#)).

2.4.1 Software Mosquitto

O Eclipse Mosquitto é um *broker* de mensagens de código aberto, licenciado sob *Eclipse Public License* (EPL) e *Eclipse Distribution License* (EDL), que implementa as

versões 5.0, 3.1.1 e 3.1 do protocolo MQTT. Projetado para ser leve, o Mosquitto pode ser utilizado em uma ampla variedade de dispositivos, desde computadores de placa única de baixo consumo até servidores robustos. Além disso, o projeto fornece uma biblioteca em linguagem C para a implementação de clientes MQTT, facilitando a interação com o protocolo ([Eclipse Foundation, 2024](#)).

O Mosquitto faz parte da Eclipse Foundation, e seu desenvolvimento é liderado pela organização Cedalo. O projeto é altamente portátil e compatível com diversas plataformas, disponibilizado para *download* em versões de código-fonte e binárias. Adicionalmente, a Cedalo oferece suporte profissional para instâncias hospedadas ou locais, além de treinamento gratuito por meio da plataforma MQTT Academy, que abrange desde conceitos básicos até tópicos avançados ([Eclipse Foundation, 2024](#)).

2.5 Automação

A domótica, originada da junção das palavras latinas “domus” (casa) e “robótica”, refere-se à automatização do ambiente residencial. Além do termo domótica, correlatos como automação residencial e casas inteligentes também descrevem o processo de automação domiciliar de maneira mais abrangente. A ideia de controlar dispositivos de forma inteligente remonta a 1894, quando Nikola Tesla demonstrou os primeiros experimentos de comunicação sem fio. Em 1898 ele apresentou um barco controlado por rádio, precursor do primeiro controle remoto ([JUNIOR; FARINELLI, 2018](#)).

Ademais, a automação residencial abrange a centralização e/ou controle remoto, bem como a autonomia no controle. Esses termos refletem avanços tecnológicos em eletrônica e telecomunicações, proporcionando acessibilidade técnica e financeira ([JUNIOR; FARINELLI, 2018](#)). Atualmente, é difícil conceber o conceito de casa inteligente sem a presença de dispositivos móveis como smartphones ou tablets, que servem como interfaces de usuário, possibilitando o acesso remoto aos controles via internet.

A automação, em um contexto técnico, refere-se à aplicação de sistemas de controle para realizar operações específicas com o mínimo de intervenção humana. Nos ambientes industriais, a automação é frequentemente implementada por meio de controladores lógicos programáveis (PLCs), computadores industriais e sistemas de supervisão e controle distribuído (DCS).

Destaca-se ainda que, o ciclo de automação inclui a coleta de dados por meio de sensores, o processamento dessas informações utilizando algoritmos lógicos e a geração de comandos para atuadores que executam as ações desejadas. A automação industrial emprega conceitos como malhas de controle, por exemplo: Proporcional, Integral, Derivativo (PID), para regular variáveis do processo, e sistemas de comunicação industrial, como o

Protocolo *Open Platform Communications* (OPC), para facilitar a troca de dados entre dispositivos e sistemas heterogêneos (COSTA; LISBOA; SANTOS, 2003).

2.6 Dispositivos IOT

Um sistema de IoT refere-se à comunicação entre meios físicos integrados ao ambiente, possibilitando que dispositivos gerem dados sobre a interação entre si e com o meio externo. Tais informações resultam de uma capacidade crescente de processamento de sistemas embarcados.

Partindo disso, temos que dispositivos inteligentes representam a computação ubíqua, termo utilizado para descrever a presença imperceptível de dispositivos conectados. Onde, em um futuro próximo, as máquinas estarão incorporadas aos objetos, invisíveis aos nossos olhos, e seus serviços estarão disponíveis 24 horas por dia, auxiliando o homem em suas tarefas diárias (Luiz Adolfo, 2025). Os computadores estarão interligados em rede, capazes de detectar a presença de usuários e outros dispositivos, interagindo automaticamente e configurando um contexto inteligente para uma utilização mais eficiente.

Dessa forma, a tecnologia de comunicação IoT se aplica a diversas áreas, como ressalta o autor Aguiar (AGUIAR, 2018), configurando sistemas, casas ou cidades inteligentes, tornando processos mais eficientes, por meio de controle logístico e otimizando recursos. (CONHAGO, 2019).

2.7 Eficiência energética

De acordo com Jannuzzi (JANNUZZI, 2005), uma mudança radical no atual sistema energético global é essencial para perseguir efetivamente a sustentabilidade e integrá-la nas decisões políticas e escolhas tecnológicas futuras. Os países em desenvolvimento enfrentam desafios adicionais e significativos, especialmente no que diz respeito ao contínuo desenvolvimento econômico e à necessidade de aumentar o acesso e os níveis de serviços energéticos para suas populações.

Visto isso, dado que a energia é um recurso imprescindível para a realização das atividades na sociedade moderna, seja para a utilização de aparelhos, lâmpadas, motores elétricos ou fábricas, torna-se evidente que um contexto de ambiente inteligente é crucial no que se refere à eficiência energética. Nesse sentido, mediante um maior controle, é possível reduzir o consumo mantendo a mesma demanda, contribuindo assim para evitar o desperdício e para redução dos custos associados. A automação, portanto, emerge como uma estratégia fundamental.

Partindo das demandas contemporâneas, a escolha de um dispositivo embarcado requer que o projetista leve em consideração a autonomia do sistema e a sua necessidade

de energia. Quanto maior a autonomia e menor a demanda energética, mais competitivo será o produto (CONHAGO, 2019).

2.8 Microcontroladores

Os microcontroladores, essencialmente circuitos integrados empregados em sistemas digitais, não apenas possuem recursos internos que capacitam a execução de programas, mas também possibilitam a leitura e escrita digital por meio de valores de tensão em suas portas. Esta característica favorece uma comunicação eficiente com uma variedade de periféricos externos, permitindo a expansão de suas funcionalidades e, por conseguinte, oferecendo inúmeras aplicações (FIOZERA, 2023).

É importante salientar que os microcontroladores, em particular da família AVR, com os modelos TinyAVR, MegaAVR, XmegaAVR e Atmel At94k, foram originalmente projetados por Alf-Egil Bogen e Vegard Wollan, estudantes da Universidade Norueguesa de Tecnologia e Ciência. Possuindo uma arquitetura *Reduced Instruction Set Computer* (RISC) e memória de programação Flash, caracterizando-se como dispositivos de baixo custo, alto desempenho e consumo de energia reduzido. Posteriormente, em colaboração de produção com a empresa Atmel, resultou na introdução do primeiro microcontrolador AVR de 8 *bits* (SILVA; ARAÚJO; CAVALCANTE, 2019).

Além disso, é indubitável que eles desempenham um papel crucial em sistemas embarcados e abrigam uma variedade de componentes essenciais para executar tarefas específicas, como a Unidade Central de Processamento (CPU) executa as instruções do programa armazenado na Memória de Programa (Flash), enquanto a Memória de Dados (RAM) armazena dados temporários (KERSCHBAUMER, 2025).

Temos, também, a Memória Não Volátil (EEPROM ou Flash). Que desempenha um papel imprescindível ao garantir a retenção de dados persistentes. Além disso, destacam-se os Periféricos de Entrada/Saída (I/O), que desempenham um papel facilitador na comunicação com o ambiente externo. Adicionalmente, componentes como Timer/Contador, Conversores Analógico-Digitais (ADC) e Clock/Oscilador assumem uma função primordial ao contribuir para a execução de tarefas essenciais, como temporização, medição e comunicação (KERSCHBAUMER, 2025).

Outro microcontrolador importante é o Tensilica Xtensa LX106, com 32 bits e baseado na arquitetura RISC, desenvolvido pela Cadence Design Systems. Este processador é amplamente utilizado em aplicações de Internet das Coisas (IoT), devido ao seu equilíbrio entre desempenho e eficiência energética. Por exemplo, o ESP8266, um módulo Wi-Fi popular, integra o Xtensa LX106, oferecendo conectividade sem fio a dispositivos embarcados. Conforme a especificação do ESP32, o processador principal é o Tensilica

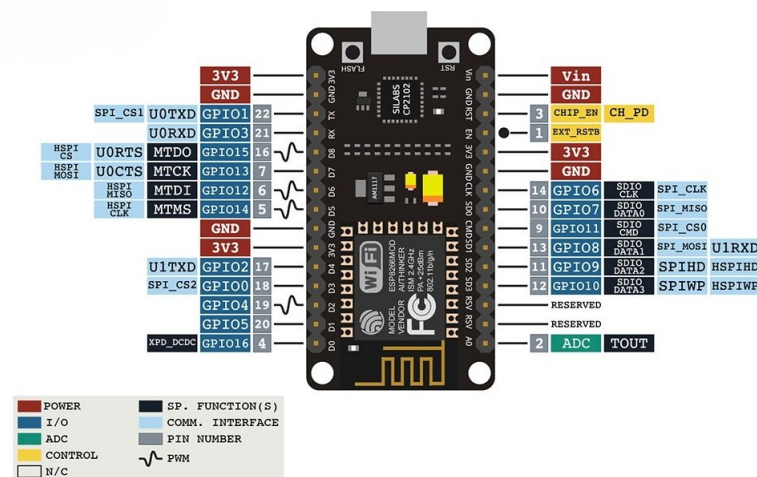
Xtensa de 32 bits LX6, com núcleos duplos e frequência de *clock* de até 240 MHz (SYSTEMS, 2025).

2.9 Módulo ESP8266

O módulo ESP8266, desenvolvido pela Espressif Systems e como enfatizado por Rodrigues (RODRIGUES, 2017), é fundamental na implementação prática da IoT. Notável por sua integração com o sistema de comunicação sem fio, o ESP8266 é frequentemente utilizado como um módulo *Wi-Fi* em dispositivos IoT. Sua característica distinta de oferecer uma solução compacta e com bom desempenho de processamento a um custo acessível o torna uma escolha popular para conectar dispositivos à Internet e possibilitar a troca de dados em ambientes inteligentes (SILVA; SALES, 2017).

O ESP8266 apresenta 17 interfaces de Entradas/Saídas de Uso Geral, conhecidas por *General Purpose Input/Output* (GPIO). Este dispositivo engloba uma variedade de componentes, incluindo interfaces periféricas digitais padrão, interruptores de antena, amplificador de potência, amplificador de recepção de baixo ruído, RF balun, filtros e módulos de gerenciamento de energia. A Figura 3 apresenta um resumo das conexões presentes no módulo.

Figura 3 – Módulo ESP8266 (NodeMcu v3)



Fonte: Adaptado de (AGUIAR, 2018)

Nesse contexto, o ESP8266 contribui para a criação de soluções inteligentes em diversos campos como automações residenciais, controle de sistemas e acesso à nuvem, fornecendo a conectividade necessária para tornar efetivas as aplicações propostas pela IoT.

2.10 Sonoff

A Sonoff é uma marca reconhecida no mercado de automação residencial, oferecendo soluções inovadoras e acessíveis para transformar dispositivos comuns em dispositivos inteligentes. Desenvolvida pela empresa ITEAD, a linha de produtos Sonoff permite que aparelhos elétricos sejam controlados remotamente via *Wi-Fi*, utilizando aplicativos móveis como o *eWeLink*. Com uma ampla gama de produtos, incluindo interruptores inteligentes, módulos de relé, sensores e *hubs* de automação, ela se destaca pela compatibilidade com protocolos como *MQTT* e *HTTP API* (SONOFF, 2025).

2.11 Firmware Tasmota

O Tasmota é um *firmware* de código aberto desenvolvido para dispositivos baseados nos *chipsets* ESP8266, ESP32, ESP32-S e ESP32-C3 da Espressif, criado e mantido por Theo Arends. O projeto iniciou-se como Sonoff-MQTT-OTA, em janeiro de 2016, visando fornecer aos dispositivos ITEAD Sonoff baseados em ESP8266 funcionalidades extras e maior controle dos usuários.

Outrossim, o Tasmota evoluiu de uma solução simples para modificar o Sonoff Basic, um dos primeiros dispositivos de automação residencial baratos e acessíveis do mercado, para um ecossistema completo, capaz de suportar uma vasta gama de dispositivos baseados no ESP8266. Atualmente, é amplamente adotado na automação residencial, destacando-se por sua flexibilidade em controlar dispositivos de maneira independente da nuvem. (Tasmota, 2025).

Suas principais características incluem a comunicação via protocolo MQTT, gerenciamento de conexões *Wi-Fi*, controle de saídas, leitura de entradas, e suporte a diversas bibliotecas integradas, como gerenciamento de emissores e receptores infravermelho (*IRremote*). Além disso, oferece uma interface gráfica acessível via navegador, além de permitir atualização de *firmware* através de *Over the Air* (OTA) (MasterWalker, 2025).

3 Metodologia

Para o desenvolvimento do projeto, a metodologia divide-se nas seguintes etapas: levantamento de requisitos para a aplicação (seção 3.1), criação do servidor MQTT (seção 3.2), instalação e configuração do Tasmota no NodeMCU (seção 3.3), desenvolvimento do sistema *web* (seção 3.4) e montagem do *hardware* (seção 3.5). Além disso, a seção 3.6 apresenta a análise de custos do projeto, enquanto a seção 3.7 aborda a integração, comunicação e ambiente de produção.

Os requisitos funcionais e não funcionais levantados são:

3.1 Levantamento dos requisitos

3.1.1 Requisitos funcionais (FR)

1. Controle de Acesso:

- O sistema deve gerenciar o acesso aos dispositivos das salas e laboratórios.
- Deve ser capaz de autenticar usuários autorizados para acesso à aplicação.
- Deve permitir a criação e gestão de perfis de usuários com dois níveis diferentes de acesso: Professores e Administradores.
- O perfil de administrador deve ser responsável por alimentar o sistema, como, por exemplo, realizar os cadastros de professores, salas, períodos, grupos de dispositivos, além de ter privilégios totais para edição e exclusão.
- O usuário com perfil de professor deve possuir acesso para alterar seus dados e controlar as rotinas de automação previamente atribuídas pelo administrador.

2. Integração com Módulo ESP8266:

- O sistema deve integrar-se eficientemente ao módulo ESP8266 (módulo disponível para o projeto).
- Deve possibilitar a comunicação bidirecional entre a aplicação *web* e o *hardware* dedicado através do *broker* MQTT seguro e responsivo.

3. Coleta e Análise de Dados:

- A aplicação tem como finalidade a coleta de dados no ambiente laboratorial, abrangendo o uso de dispositivos como ar-condicionado, tomadas, fechaduras elétricas de portas e sistemas de acesso.

- Deve oferecer recursos analíticos que forneçam percepções úteis, com gráficos e atalhos para facilitar a utilização da aplicação.
- O sistema deve registrar os *logs* de execução das rotinas e dos acionamentos manuais.

4. Gerenciamento de Dispositivos e Rotinas:

- O sistema deve permitir o gerenciamento eficiente de dispositivos no laboratório.
- Deve possibilitar a programação e automação de rotinas, como controle de iluminação e climatização.
- As rotinas devem seguir uma hierarquia, começando pelo período cadastrado, passando pelo dia da semana e pelos horários de início e fim definidos.
- Deverá ser possível ativar e desativar uma rotina a qualquer momento, controlando, assim, todos os grupos de dispositivos vinculados.

3.1.2 Requisitos não funcionais (FNR)

1. Eficiência do *Hardware*:

- O *hardware* dedicado deve ser projetado para operar de maneira eficiente, garantindo baixo consumo de energia e desempenho adequado.

2. Segurança:

- A aplicação *web* e o módulo ESP8266 devem adotar medidas robustas de segurança para prevenir acessos não autorizados e assegurar a integridade e confidencialidade dos dados. Para isso, devem ser implementados mecanismos como autenticação baseada em tokens seguros (*JWT* ou *OAuth 2.0*), proteção contra ataques como *SQL Injection*, *Cross-Site Scripting* (XSS) e *Cross-Site Request Forgery* (CSRF)

3. Conectividade IoT:

- A integração com os princípios da Internet das Coisas (IoT) deve assegurar uma conectividade eficaz entre dispositivos, promovendo uma comunicação fluida e confiável.

4. Usabilidade:

- A interface da aplicação *web* deve ser intuitiva e de fácil uso para auxiliar a interação dos usuários.

5. Custo Aceitável:

- O projeto do *hardware* deve ser economicamente viável, garantindo um custo aceitável para implementação e manutenção.

6. Disponibilidade:

- A aplicação deve apresentar alta disponibilidade, assegurando que o controle de acesso e a automação não sejam interrompidos significativamente.

3.2 Criação do servidor MQTT

Para a configuração do servidor MQTT, foi utilizado o *software* Mosquitto, apresentado na subseção 2.4.1. O processo de instalação foi inicialmente realizado em ambiente local (desenvolvimento, testes e validações). Sendo posteriormente replicado no servidor de produção descrito na seção 3.7.1, seguindo os passos a seguir:

3.2.1 Configuração para o Broker Mosquitto

Primeiramente, é necessária a instalação do *software* Mosquitto. Em seguida, configura-se uma senha para o tráfego de mensagens utilizando o comando “mosquitto_passwd”, no qual é possível definir a senha desejada. Os comandos são:

```
1 sudo apt-get install mosquitto
2 sudo apt-get install mosquitto-clients
3 sudo mosquitto_passwd -c /etc/mosquitto/passwd {senha_desejada}
```

Posteriormente, deve-se criar um arquivo de configuração para referenciar a senha criada. Para isso, é utilizado o comando:

```
1 sudo nano /etc/mosquitto/conf.d/default.conf
```

No arquivo de configuração aberto, são adicionadas duas linhas para garantir a segurança do servidor MQTT. A primeira desativa o acesso anônimo, assegurando que apenas clientes autenticados possam se conectar, publicar ou subscrever mensagens. A segunda define o caminho do arquivo onde estão armazenadas as credenciais de usuários e senhas:

```
1 allow_anonymous false
2 password_file /etc/mosquitto/passwd
3 listener 1883
4 bind_address 0.0.0.0
```

Essas configurações têm como objetivo de proteger o servidor e restringir o acesso exclusivamente a usuários autorizados.

Além disso, as linhas três e quatro são necessárias para que o servidor escute conexões na porta padrão 1883 em todas as interfaces de rede disponíveis. Para aplicar as alterações, reinicie o Mosquitto com o comando:

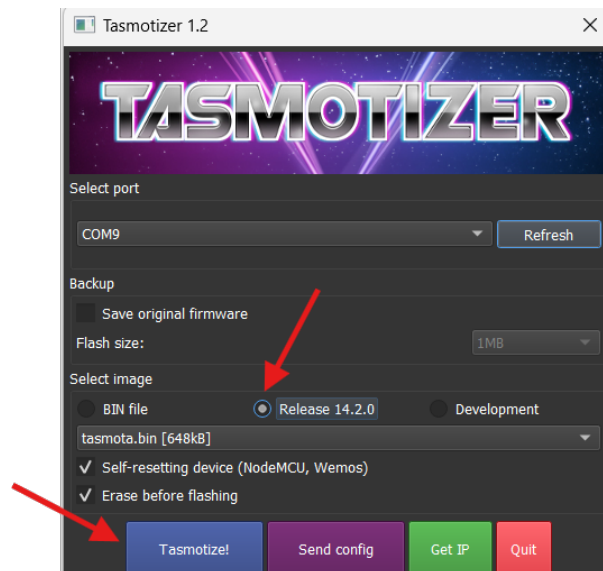
```
1 sudo systemctl restart mosquitto
```

Com essas configurações, é possível estabelecer uma comunicação segura e bem organizada utilizando o *software* Mosquitto para o uso do protocolo MQTT.

3.3 Instalação do Tasmota no NodeMCU (ESP8266)

No portal oficial do Tasmota, encontra-se a ferramenta de flash (Tasmotizer), disponível para diversos sistemas operacionais. Após descarregar a aplicação, deve-se seguir a documentação fornecida, que explica o processo em detalhes. Com o ESP8266 conectado via USB, o usuário deve abrir o programa e selecionar as configurações conforme mostrado na Figura 4. Primeiramente, na opção 'Release', em 'Select Image', a versão mais recente disponível será utilizada, com download realizado diretamente pelo Tasmotizer. Ao clicar no botão 'Tasmotize!', o *firmware* é instalado no microcontrolador.

Figura 4 – Instalação do Tasmotizer



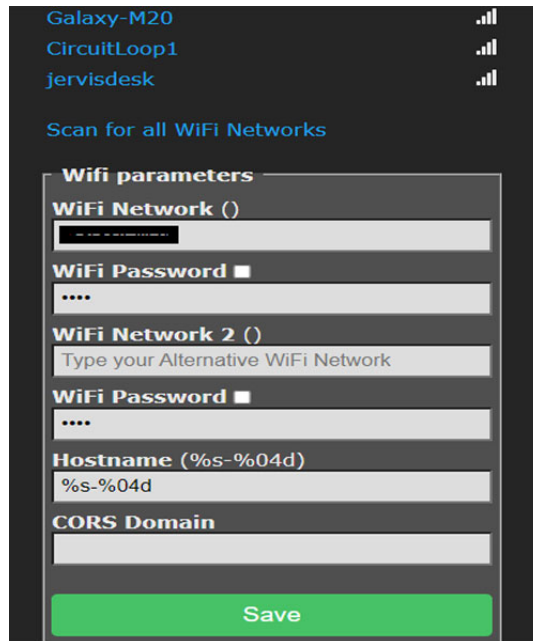
Fonte: (GABARDO, 2017).

3.3.1 Configuração Tasmota no NodeMCU

Após concluir a instalação do *firmware* no NodeMCU, o dispositivo reinicia, criando uma rede *Wi-Fi* temporária com o nome “tasmota-XXXX”. Para primeira configuração, é necessário conectar-se a essa rede *Wi-Fi*. Subsequentemente, a interface *web* pode ser acessada pelo endereço IP padrão “192.168.4.1”, conforme a Figura 5. Após adicionar a

rede *Wi-Fi* local para acesso à rede doméstica será necessário localizar o IP atribuído ao dispositivo. Esse IP é encontrado na interface do roteador ou utilizando um *scanner* de rede.

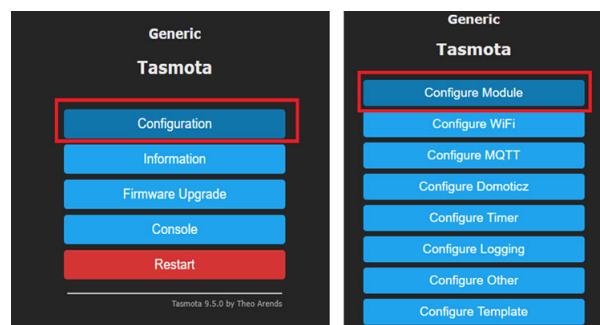
Figura 5 – Primeiro acesso Tasmota



Fonte: Autor.

Posteriormente, é necessário realizar as configurações fundamentais para o funcionamento adequado do dispositivo. Entre elas, incluem-se os dados de acesso ao servidor MQTT, que estão detalhados na seção 3.2, e a configuração do módulo utilizado. Para este projeto, a configuração dos pinos da ESP8266 é apresentada nas Figuras 6 e 7.

Figura 6 – Configurações gerais Tasmota



Fonte: Autor.

Figura 7 – Configuração dos pinos ESP8266 no Tasmota

The screenshot shows the 'Module parameters' section of the Tasmota configuration interface. The 'Module type' is set to 'Generic (0)'. Below this, a list of GPIO pins is shown with their configured functions and pin numbers. The pins are: D3 GPIO0 (Relay, 4), TX GPIO1 (None), D4 GPIO2 (Relay, 5), RX GPIO3 (IRsend, 1), D2 GPIO4 (Relay, 3), D1 GPIO5 (Relay, 2), FL GPIO9 (None), FL GPIO10 (None), D6 GPIO12 (Relay, 6), D7 GPIO13 (Relay, 7), D5 GPIO14 (IRrecv), D8 GPIO15 (Relay, 8), D0 GPIO16 (Relay, 1), and A0 GPIO17 (None). A green 'Save' button is at the bottom.

Pin	GPIO	Function	Pin Number
D3	GPIO0	Relay	4
TX	GPIO1	None	
D4	GPIO2	Relay	5
RX	GPIO3	IRsend	1
D2	GPIO4	Relay	3
D1	GPIO5	Relay	2
FL	GPIO9	None	
FL	GPIO10	None	
D6	GPIO12	Relay	6
D7	GPIO13	Relay	7
D5	GPIO14	IRrecv	
D8	GPIO15	Relay	8
D0	GPIO16	Relay	1
A0	GPIO17	None	

Fonte: Autor.

Ademais, é importante selecionar o tipo de módulo utilizado. No caso das placas ESP8266, recomenda-se optar pela classe genérica, que já fornece a listagem dos pinos conforme o padrão da maioria dos módulos disponíveis. Na configuração adotada, foram utilizados 8 pinos para o controle das relés, o pino RX para o “IRsend”, responsável pelo envio de sinais infravermelhos, e o pino D5 para o recebimento dos sinais.

O Tasmota, ao lidar com dispositivos infravermelhos, os classifica em duas categorias: IRSend e IRHVAC. Na categoria IRSend, o protocolo, o número de *bits* e o código de dados são utilizados para envio e recepção do sinal. Na categoria IRHVAC (Heating, Ventilation, and Air Conditioning, ou, em tradução literal, Aquecimento, Ventilação e Ar-Condicionado), é possível acionar todas as funções de um controle remoto a partir de uma única cópia de um botão. Parâmetros como a temperatura, a função *FanSpeed*, Turbo e Econômico podem ser enviados aos dispositivos.

3.4 Desenvolvimento do sistema web

Primeiramente, é necessário configurar um ambiente de desenvolvimento PHP. Para isso, utiliza-se a ferramenta XAMPP, que integra Apache, MySQL e PHP em um ambiente pré-configurado. Essa abordagem elimina a necessidade de instalar e configurar individualmente cada componente. Com o XAMPP, configura-se facilmente um servidor local, um banco de dados MySQL e a linguagem de *script* PHP, otimizando o desenvolvimento e os testes de aplicações.

Considerando o levantamento de requisitos, em consonância com a documentação do *framework* Laravel, o desenvolvimento da aplicação faz uso da funcionalidade ‘*make*’ do Laravel por sua praticidade na criação e integração do ‘*model*’, ‘*view*’ e ‘*controller*’.

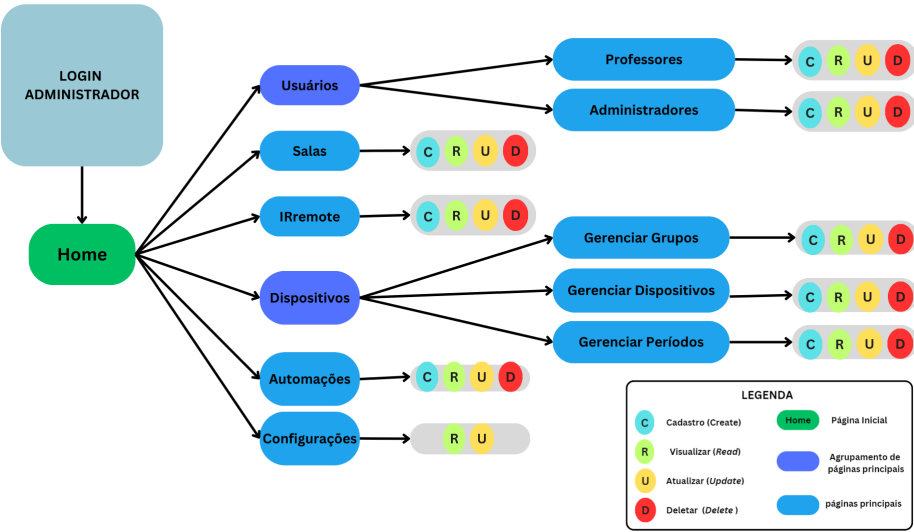
Essa abordagem é especialmente eficaz quando combinada com a ferramenta ‘*migrate*’, que automatiza a criação das colunas necessárias no banco de dados, impulsionando o processo de desenvolvimento.

Outra ferramenta importante utilizada é o “*schedule:work*”, do próprio do Laravel, pois permite agendar e automatizar a execução de comandos e *scripts* sem a necessidade de configurar as tarefas de agendamento do sistema operacional (*cron jobs*, do inglês). Isso é possível por meio do sistema de “*Task Scheduling*” do Laravel, que gerencia a execução das tarefas de forma integralizada.

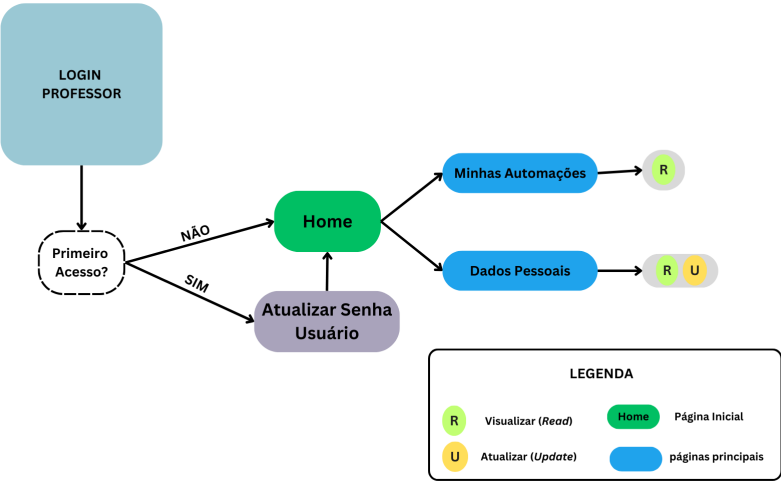
Partindo disso, e considerando que a aplicação *web* deve atender a dois tipos de usuários, administradores e professores, cada um com privilégios específicos, as Figuras 8a e 8b apresentam o mapa do site para cada grupo, ilustrando as páginas disponíveis e os multi-níveis de acesso.

É importante destacar que, para o grupo de professores, no primeiro acesso após o cadastro realizado pelo administrador, a senha padrão é o CPF do usuário. Nesse momento, o sistema redirecionará obrigatoriamente o professor para a atualização de sua senha de preferência. Vale ressaltar que seus privilégios, como apresentado na Figura 8a, são limitados ao acesso às automações associadas ao seu nome pelo administrador, além da possibilidade de atualizar alguns campos do seu cadastro, como *e-mail*, telefone e endereço.

Figura 8 – Mapas do Site - Administrador e Professor.



(a) Mapa do Site - Administrador



(b) Mapa do Site - Professor

Fonte: Autor.

Para uma melhor compreensão do Mapa do Site (Administrador), ilustrado na Figura 8b, a Tabela 5 apresenta a descrição das telas (páginas) do sistema e suas funcionalidades.

Tabela 5 – Descrição das Telas do Sistema (Administrador) e suas Funcionalidades

Tela	Descrição	Funcionalidades
Home	Página inicial do sistema, que apresenta um resumo das funcionalidades, gráficos e atalhos.	Exibe o cronograma de automações, dispositivos cadastrados e um controle manual de ativação/desativação.
Usuários	Gerenciamento de usuários: professores e administradores.	Adicionar, editar e excluir usuários; definir permissões.
Salas	Página para visualizar e gerenciar salas registradas.	Criar, editar e excluir salas.
Dispositivos	Configuração e controle dos dispositivos conectados ao sistema.	Listar dispositivos, visualizar status, configurar conexões e administra períodos.
Gerenciar Grupos	Administração de grupos de dispositivos para facilitar a organização.	Criar e editar grupos de dispositivos; associar dispositivos.
Gerenciar Dispositivos	Registro individual dos dispositivos cadastrados.	Unidade de dispositivos para controle de inventário.
Gerenciar Períodos	Definir períodos letivos de funcionamento das automações, exemplo: “2024.2”.	Cadastrar, visualizar, editar e excluir caso não vinculada a nenhuma automação.
IRemote	Interface para gerenciar o funcionamento dos dispositivos infravermelhos.	Cadastro, visualização e exclusão dos sinais infravermelhos.
Automações	Configuração de automações e regras de acionamento.	Permite criar rotinas automáticas e definir regras com base no período, dia da semana, horário de início e término, vinculando usuários (professores), grupos de dispositivos e salas.
Sistema	Configuração geral para funcionamento correto das integrações do sistema.	Ajustar preferências e a definições dos parâmetros globais, incluindo as conexões do servidor MQTT (Host, Usuário, Senha e Porta) e as características do módulo ESP8266 (Número Máximo de Grupos e Tópico Base).

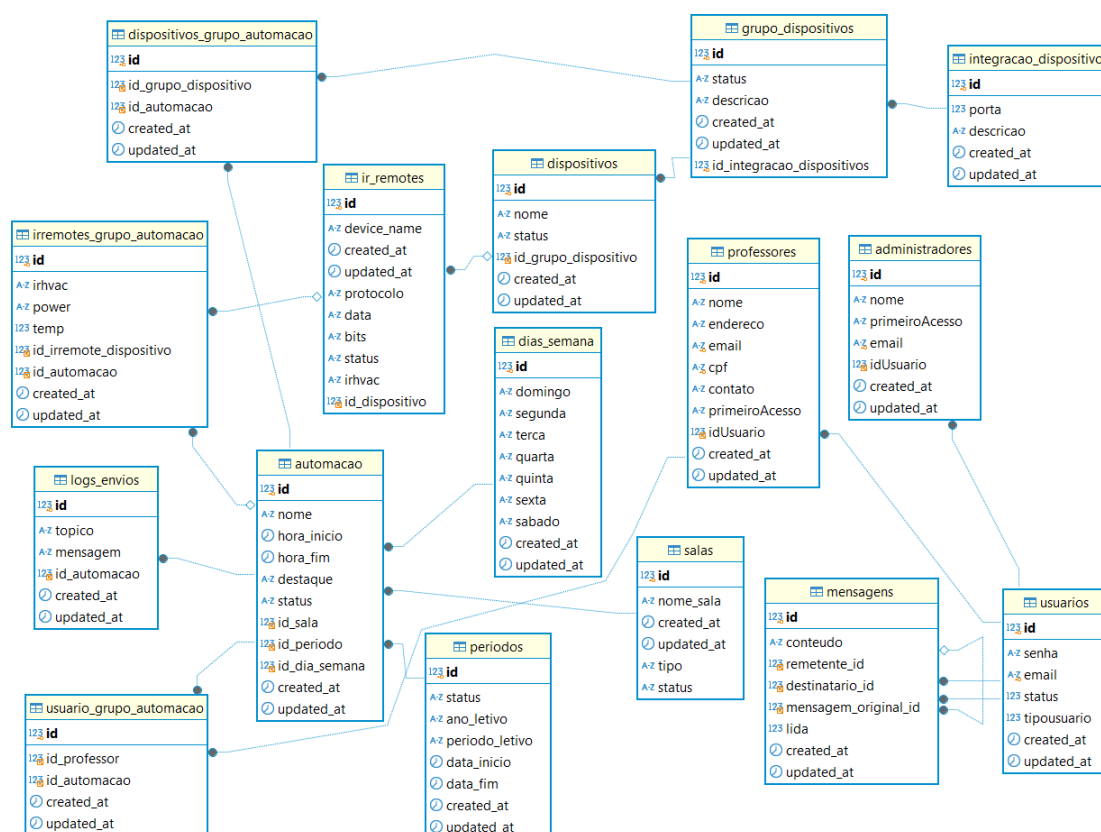
Fonte: Autor.

Para compreender o funcionamento da aplicação *web*, também é necessário analisar a construção do banco de dados, conforme a exposto na subseção 3.4.1.

3.4.1 Banco de dados do projeto

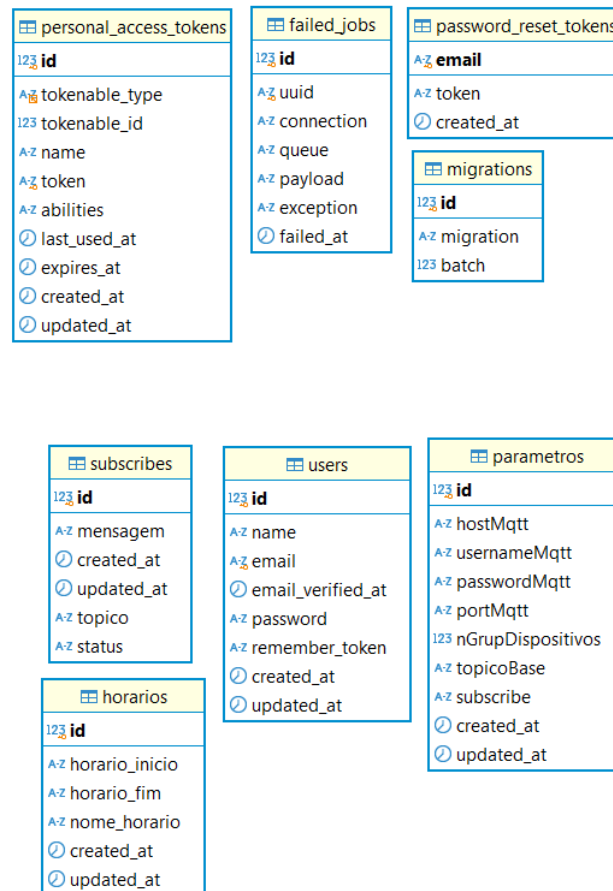
O diagrama ER do banco de dados, detalha a estrutura e os relacionamentos entre as entidades, atributos e tabelas, visando oferecer uma visão clara da organização e interconexão dos dados, encontra-se nas Figuras 9 e 10.

Figura 9 – Diagrama ER do banco de dados – Parte (a).



Fonte: Autor.

Figura 10 – Diagrama ER do banco de dados – Parte (b).



Fonte: Autor.

Analisando o diagrama ER do banco de dados, faz-se necessário fornecer comentários relevantes sobre algumas das principais tabelas, como:

A integração de dispositivos com o ambiente Tasmota é gerenciada pelas tabelas “parametros” e “integracao_dispositivos”. A tabela “parametros” armazena informações sobre o módulo ESP8266, como o número de saídas (pinos) utilizados, além das configurações do sistema, incluindo dados do *broker* MQTT, como host, porta e tópicos de comunicação. Já a tabela *integracao_dispositivos* registra a porta de comunicação dos módulos.

Para otimizar a administração desses dispositivos, a tabela “grupo_dispositivos” permite organizá-los em grupos, estabelecendo uma relação com a tabela *integracao_dispositivos* por meio de uma chave estrangeira (*id_integracao_dispositivos*). Além disso, a tabela “subscribes” mantém um registro das mensagens recebidas dos tópicos cadastrados na interface de configurações, garantindo um monitoramento eficiente da comunicação no sistema.

Já a automação de eventos no sistema é gerenciada pela tabela “*automacao*”, que se relaciona diretamente com as tabelas “*salas*” (*id_sala*), “*periodos*” (*id_periodo*), “*dispositivo_grupo_automacao*” e “*dias_semana*” (*id_dia_semana*). Essa estrutura não apenas possibilita o agendamento de ações em locais e períodos específicos, mas também permite a configuração de regras automatizadas para o acionamento dos grupos de dispositivos com base em horários predefinidos e dias da semana.

Além disso, os registros das execuções dessas automações são armazenados na tabela “*logs_envios*”, que mantém informações sobre as mensagens enviadas, sua associação com a automação correspondente aos sucessos nos acionamentos ou possíveis falhas na comunicação, permitindo a rastreabilidade e análise do funcionamento do sistema.

A gestão de usuários é realizada por meio da tabela “*usuarios*”, a qual armazena dados como *email*, *senha* e *status* do usuário. A estrutura prevê hierarquias, sendo que a tabela administradores referencia a tabela “*usuarios*”, identificando aqueles com privilégios administrativos. Similarmente, a tabela professores estabelece a relação entre docentes e usuários registrados no sistema.

Os períodos acadêmicos são administrados pela tabela “*periodos*”, que armazena informações sobre o semestre letivo, datas de início e fim, e *status* do período. Essa tabela é essencial para a vinculação com a automação de dispositivos e horários previamente cadastrados.

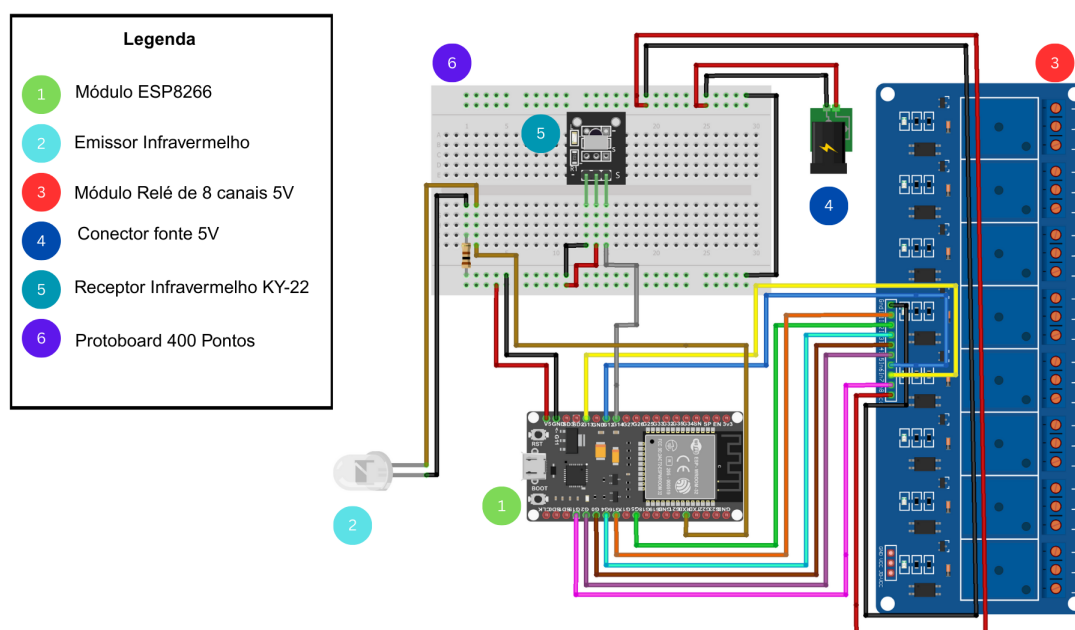
3.5 Montagem do *hardware*

Para atender aos requisitos e objetivos do projeto da aplicação, a Tabela 6 apresenta a lista de componentes utilizados na construção do *hardware*.

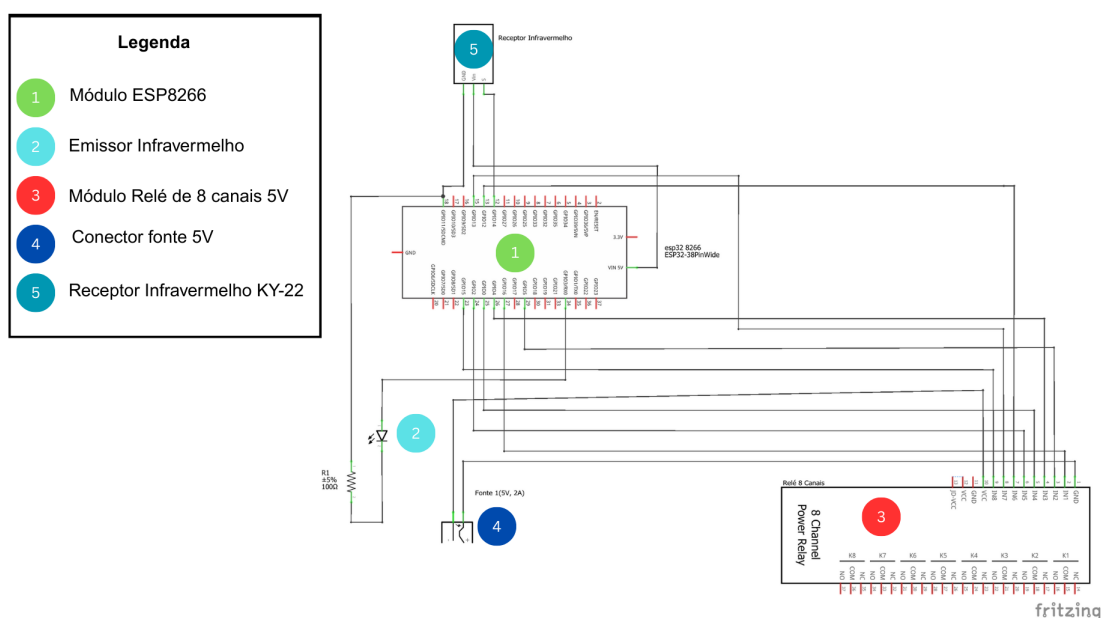
Tabela 6 – Criação do *Hardware*

Quantidade	Componente	Descrição
1x	Microcontrolador ESP8266	Gerenciar o funcionamento de todos os componentes.
1x	Fonte 5V 2A	Fornecer a tensão em todo o circuito que deve utilizar 5V.
1x	Módulo Relé de 8 canais 5V	Interruptor controlado eletricamente, oferece a capacidade de abrir ou fechar circuitos elétricos, destacando-se por sua habilidade em acionar equipamentos que operam com tensões de até 220V.
3x	Lâmpadas 220V	Gerenciado pelos Canais 1, 2 e 3 do Módulo Relé.
1x	Protoboard 400 Pontos	Alocar as ligações entre os componentes de tensão até 5V.
1x	Fonte 5V 1A	Fornecer a alimentação exclusiva para a placa do microcontrolador.
2x	Emissor Infravermelho	Responsável pelo envio de sinais infravermelhos para dispositivos remotos.
1x	Receptor Infravermelho KY-22	Responsável por receber sinais infravermelhos, permitindo o clone dos sinais dos controles remotos originais.
1x	Conjunto jumper macho-fêmea	Responsável por realizar a conexão entre pinos ou terminais de dispositivos eletrônicos.

Com base nos pinos definidos na interface do Tasmota, conforme descrito na subseção 3.3.1, foram desenvolvidas as conexões entre os componentes. A Figura 11 ilustra essa montagem, enquanto a Figura 12 apresenta o esquemático elétrico.

Figura 11 – Montagem e conexões do *Hardware*

Fonte: Autor.

Figura 12 – Esquemático elétrico *Hardware*

Fonte: Autor.

É importante salientar, que o microcontrolador, utilizando o *firmware* Tasmota em conjunto com a integração entre os *softwares* descritos na sessão 3.7, gerencia o acionamento dos grupos de dispositivos vinculados a cada canal do módulo relé. Além disso, ele recebe

sinais infravermelhos por meio do sensor KY-22 e, através do emissor infravermelho, é capaz de transmitir comandos.

Dessa forma, por meio de ciclos de rotina ou ajustes de parâmetros na aplicação *web*, é possível ativar ou desativar os dispositivos, não apenas via relés, mas também por meio de controle infravermelho.

3.6 Custo do projeto

Para a criação do sistema tem-se, na Tabela 7, os preços médios dos componentes essenciais levantados no segundo semestre de 2024. Este levantamento visa fornecer uma estimativa precisa dos custos envolvidos na criação do projeto, permitindo não somente uma análise detalhada, mas também uma gestão eficaz dos recursos financeiros destinados à sua execução.

Tabela 7 – Custo do projeto

Quantidade	Componente	Valor Unitário	Valor Total
1x	Módulo ESP8266	40,00	40,00
2x	Fonte 5v	15,00	30,00
1x	Módulo Relé de 8 canais 5V	40,00	40,00
1x	Protoboard 400 Pontos	20,00	20,00
2x	Led emissor infravermelho	1,50	3,00
1x	Módulo receptor infravermelho KY-22	05,00	05,00
1x	Fios e conectores	2,00	2,00
Total:			R\$ 140,00

Fonte: Autor.

Para cada item, a Tabela 7 apresenta a quantidade necessária, o componente específico, o valor unitário e o valor total correspondente à quantidade especificada. O valor total do projeto, considerando todos os componentes listados, é de R\$ 140,00, o que representa um custo significativamente baixo, especialmente quando comparado aos módulos semelhantes disponíveis no mercado, cujos preços ultrapassam, frequentemente, os R\$ 900,00.

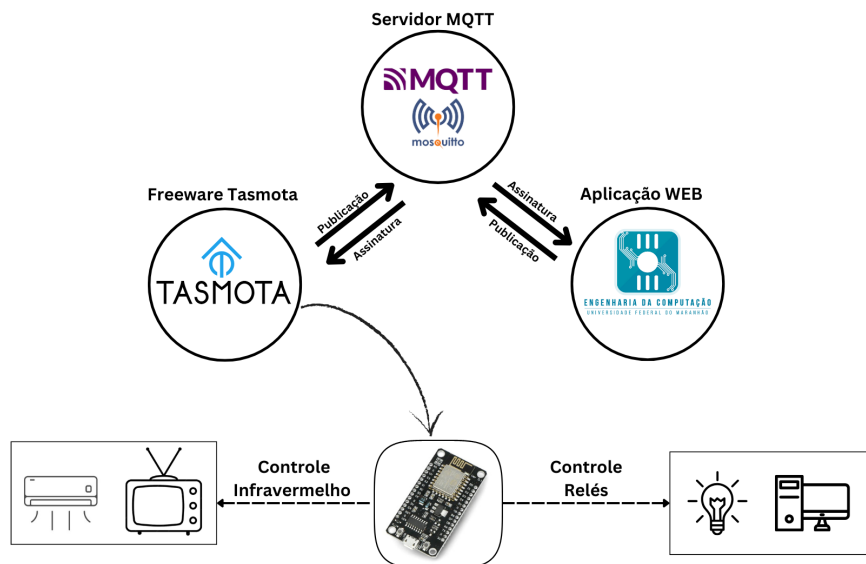
3.7 Integração, comunicação e ambiente de produção

A integração entre a aplicação *web* e o controle dos dispositivos ocorre por meio do servidor mosquitto MQTT, que atua como uma ponte entre as tecnologias. Utilizando funções executadas pelo mecanismo “*schedule:work*”, o sistema gerencia a inscrição nos

tópicos do servidor MQTT e a execução automatizada das ações nos horários programados, por meio da publicação de mensagens.

A Figura 13 expõe o fluxo de comunicação entre as tecnologias empregadas no sistema.

Figura 13 – Comunicação entre as tecnologias



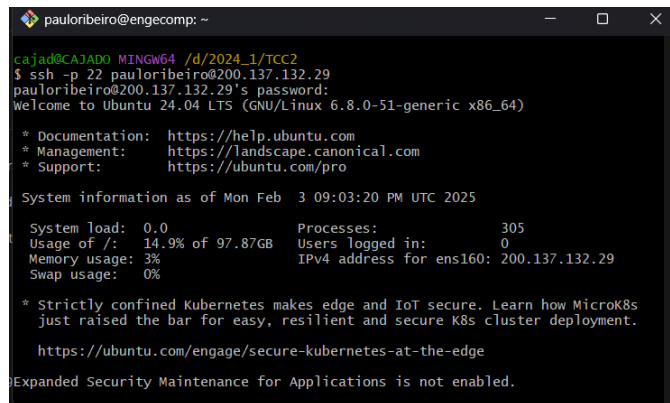
Fonte: Autor.

Como exemplo, considere uma automação cadastrada em um período letivo válido, com configurações para execução às segundas, quartas e sextas-feiras, iniciando às sete horas da manhã para ligar o ar-condicionado de um laboratório. O fluxo de acionamento ocorrerá por meio da publicação de um comando no projeto *web*, no tópico “base/cmd/IRHVAC”, com a mensagem {"Vendor": "LG", "Power": "On", "Temp": 18} no dia e horário programados. Nesse momento, o *firmware*, por estar inscrito nesse tópico, por definição, recebe a mensagem, a decodifica e, por meio do módulo ESP32, executa o envio do comando via emissor infravermelho.

3.7.1 Ambiente de produção

É relevante ressaltar que, para validar os testes e as conclusões deste trabalho, após a finalização do processo de desenvolvimento local, a pesquisa foi transferida para uma infraestrutura fornecida pelo Setor de Tecnologia da Informação (STI) da Universidade Federal do Maranhão, com acesso realizado por meio de chaves SSH. A Figura 14 apresenta um exemplo de conexão a esse servidor.

Figura 14 – Acesso à infraestrutura (Ambiente de produção)



```
pauloribeiro@engecomp: ~  
cajad@CAJADO MINGW64 /d/2024_1/TCC2  
$ ssh -p 22 pauloribeiro@200.137.132.29  
pauloribeiro@200.137.132.29's password:  
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-51-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of Mon Feb  3 09:03:20 PM UTC 2025  
  
System load:  0.0          Processes:            305  
Usage of /:   14.9% of 97.87GB  Users logged in:     0  
Memory usage: 3%          IPv4 address for ens160: 200.137.132.29  
Swap usage:   0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
  just raised the bar for easy, resilient and secure K8s cluster deployment.  
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
Expanded Security Maintenance for Applications is not enabled.
```

Fonte: Autor.

Ademais, para o processo de validação no ambiente de produção, a sala/laboratório utilizada, mencionada no levantamento de requisitos para a instalação e acompanhamento, foi o Espaço INOVTEC, conforme mostrado na Figura 15.

Figura 15 – Laboratório Inovtec

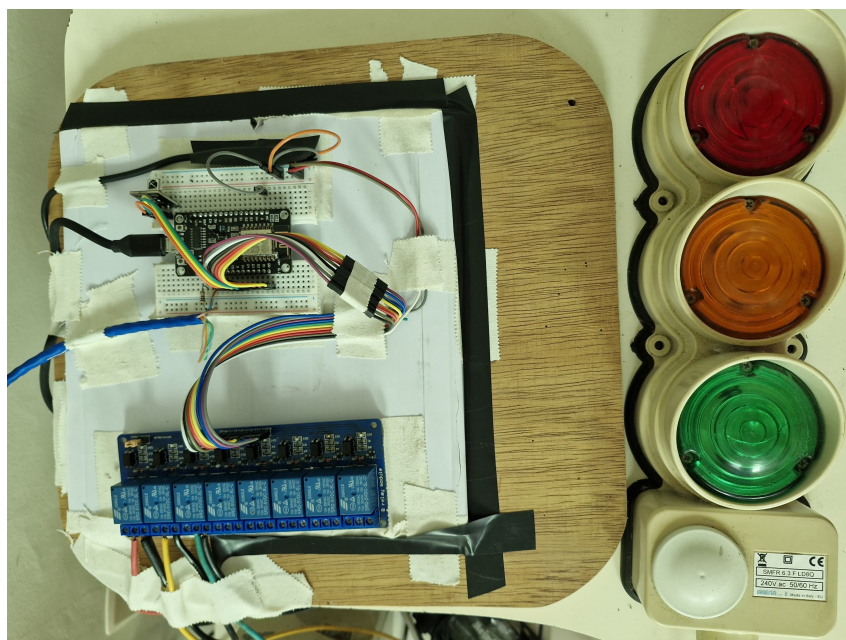


Fonte: Autor.

Este laboratório conta com dois dispositivos de controle remoto infravermelho (IR), um ar-condicionado da categoria “IRHVAC” e um projetor da categoria “IRSend”. Além disso, para o controle via relés e devido a limitações de acesso ao quadro geral, optou-se por utilizar três lâmpadas de 220V (vermelha, amarela e verde), nas quais cada uma representará um grupo de dispositivos conectados às saídas das relés. Dessa forma, um total de cinco dispositivos são controlados: dois via infravermelho e três relés de acionamento.

A Figura 16 apresenta-se o *hardware* instalado para controle do laboratório.

Figura 16 – *Hardware* Físico no laboratório



Fonte: Autor.

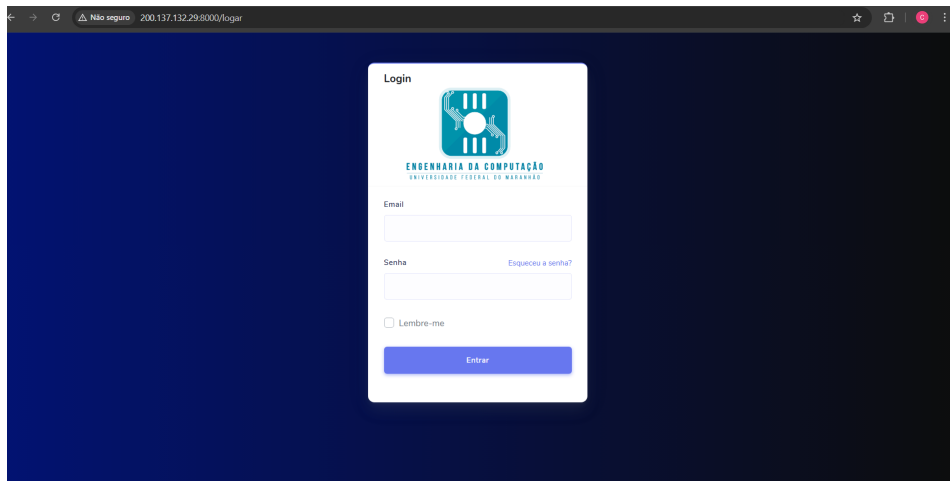
4 Resultados

Para o processo de coleta de resultados foram realizados os cadastros e as configurações necessárias no sistema em produção para o horário de aula de um dos professores do InovTec, além de outras automações para os horários de utilização do laboratório. A coleta de dados e o funcionamento do sistema ocorreram ao longo de três semanas, entre os dias 12 e 31 de janeiro de 2025. Os resultados estão divididos em duas seções: a aplicação desenvolvida, descrita na seção 4.1, e o funcionamento completo do projeto, detalhado na seção 4.2.

4.1 Páginas do sistema *web* em produção

Para iniciar o acesso à aplicação é necessário realizar o processo de login, conforme ilustrado na Figura 17. Ao autenticar-se, o usuário recebe um *token* de permissão, cuja validade é determinada pelo tempo de sessão definido pelo próprio *framework* Laravel.

Figura 17 – Tela de login



Fonte: Autor.

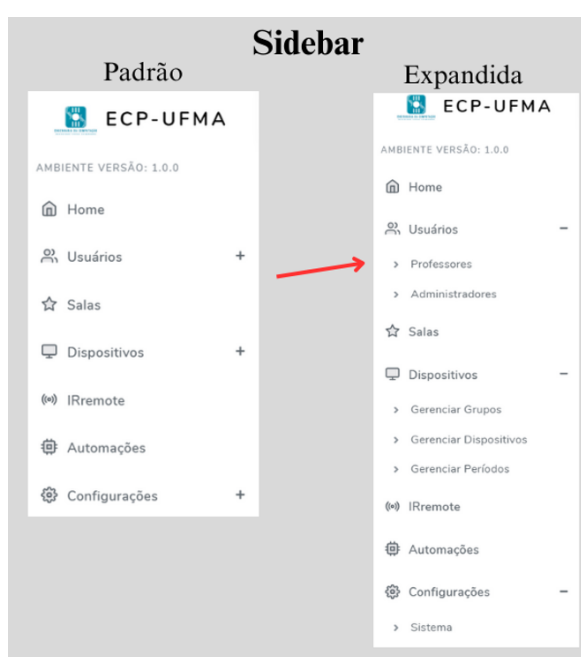
Após o processo de login, o usuário é redirecionado para a tela inicial (*dashboard*, do inglês) correspondente ao seu grupo de usuários. Para administradores, as páginas de navegação estão na Subseção 4.1.1, enquanto para professores, as informações estão na Subseção 4.1.2.

4.1.1 Área do administrador

Para a área do administrador, apresentam-se os resultados de algumas das páginas finais extraídas da aplicação em produção.

Inicialmente, visando proporcionar uma navegação rápida, a *sidebar* possui a capacidade de disponibilizar acesso a todas as páginas e funções do sistema. Algumas telas são agrupadas em submenus com base em sua similaridade, permitindo que a *sidebar* seja expandida ou recolhida para otimizar a experiência do usuário. Essa organização é ilustrada na Figura 18.

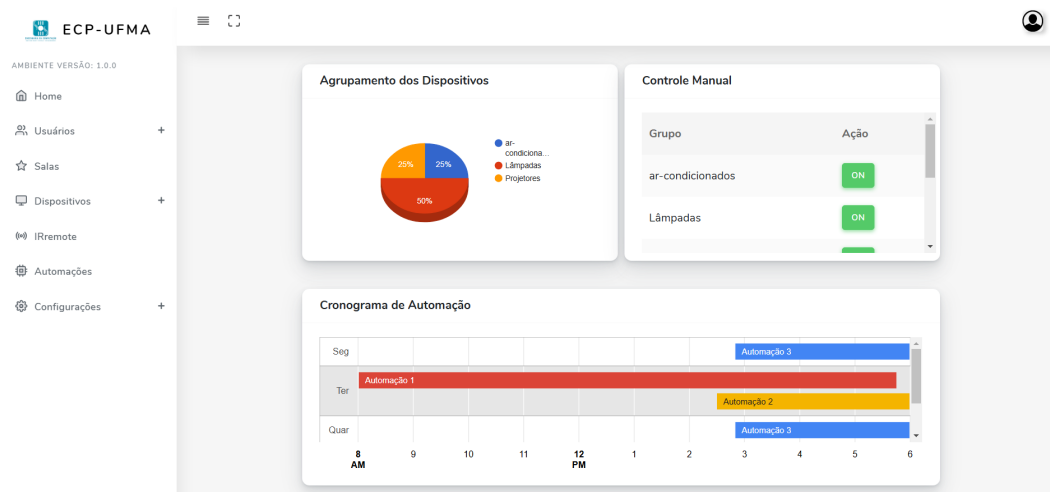
Figura 18 – *Sidebar*.



Fonte: Autor.

Outra característica relevante na área do administrador é que, na tela inicial (Figura 19), são apresentadas diversas informações relevantes para auxiliar o usuário da aplicação. Entre elas, destacam-se um gráfico que exhibe a distribuição do número de dispositivos dentro dos grupos, como lâmpadas, ares-condicionados e computadores, além de atalhos para acionamento rápido desses grupos, sendo possível ligar e desligar diretamente pelo *dashboard*.

Figura 19 – Tela Inicial -(Dashboard).



Fonte: Autor.

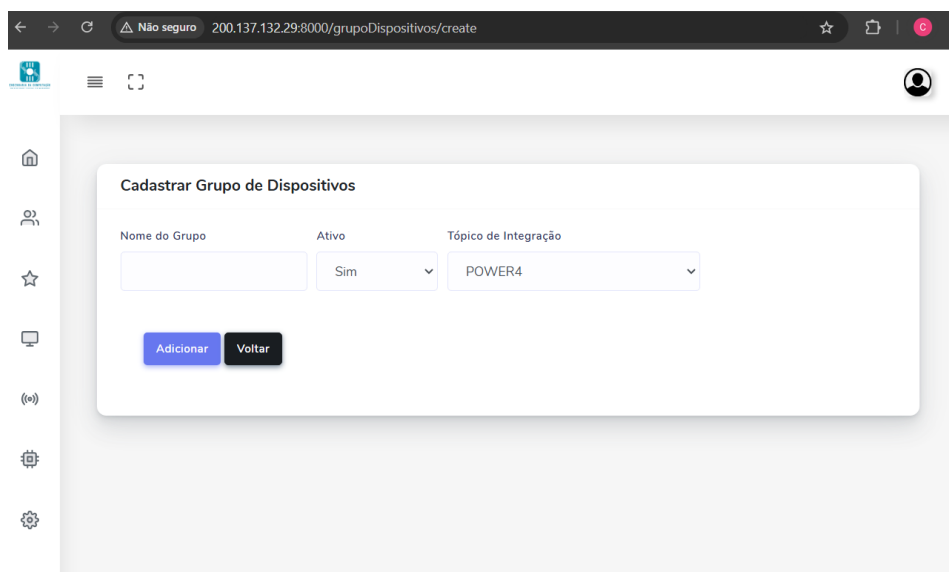
Outrossim, a Figura 19 apresenta, também, uma lista de cronogramas de automações ativas, com o dia da semana, horário inicial e final do ciclo, em paralelo com o seu tempo de duração. O comportamento e os resultados das automações, mostrados no gráfico de cronograma de automação, são detalhados na seção 4.2.

Cabe destacar que o fluxo de cadastro dos componentes do sistema é extremamente rápido e acessível, mesmo sendo rico em funcionalidades. É possível visualizar os registros com opções de pesquisa e ordenação, cadastrar novos com o mínimo de informações fornecidas pelos usuários, além de contar com mascaramentos e dicas nos campos de entrada. Entretanto, a edição e exclusão de registros são permitidas, desde que não possuam vínculos ativos, o que proporciona uma configuração completa do sistema em poucos minutos. As Figuras 20, 21, 22 e 23 ilustram essas características.

Figura 20 – Cadastro, edição e exclusão de registros.

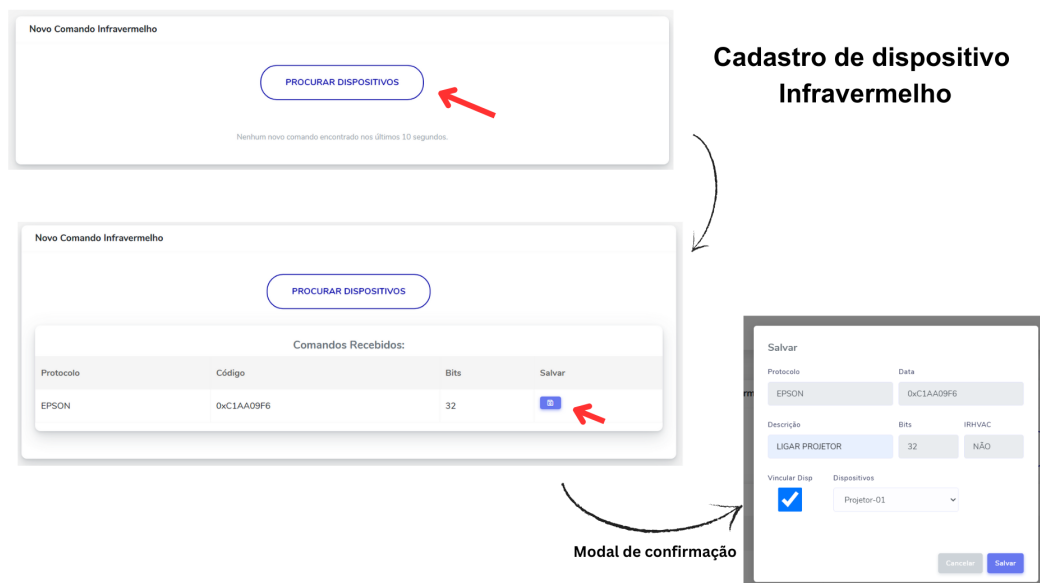
Fonte: Autor.

Figura 21 – Cadastro Grupo de Dispositivos.



Fonte: Autor.

Figura 22 – Cadastro de Dispositivo IR.



Fonte: Autor.

É notório que o cadastro de dispositivos infravermelhos, Figura 22, apresenta um comportamento distinto do cadastro de grupos de dispositivos acionados via relés, pois estes recebem um tópico de integração correspondente a um canal da relé. No caso dos dispositivos infravermelhos, ao clicar em “Novo Dispositivo”, é necessário pressionar a tecla do botão desejado, o que redireciona para uma tabela com os sinais detectados.

Após selecionar o comando desejado, basta fornecer uma breve descrição, pois os demais campos são preenchidos automaticamente devido à decodificação do sinal infravermelho (IR). Ressalta-se que o sensor KY-22 possui um alcance superior a cinco metros, garantindo a detecção dos sinais independentemente da distância em relação ao *hardware* instalado no laboratório.

A página principal do sistema, responsável pelo cadastro das automações, destaca-se pela sua usabilidade. Sua interface é majoritariamente composta por campos de seleção dos itens previamente cadastrados, o que simplifica o processo de configuração das automações.

Além disso, a adição e remoção dinâmica de itens, realizada por meio de cartões interativos (*chips*, do inglês), permite ao usuário gerenciar elementos de forma modular. Esse modelo não apenas facilita a organização e visualização dos campos, mas também possibilita uma personalização eficiente, adaptando o sistema às necessidades específicas de cada automação.

Figura 23 – Cadastro de Automação.

The screenshot shows the 'Cadastrar Automação' form in the ECP-UFMA system. The form is titled 'Cadastrar Automação' and includes the following fields and options:

- Nome da Automação:** Automação 3
- Destaque:** Sim
- Status:** Ativo
- Hora Inicial:** 07:15
- Hora Final:** 13:00
- Autorização de Usuários:** professor 1
- Sala:** Auditório
- Períodos:** 2025.1
- Comando IR:** LIGAR AR
- Grupos de Dispositivos:** ar-condicionados, Lâmpadas, Projetor
- Comandos IR Remote:** LIGAR AR
- Dias da Semana:** DOM, SEG, TER, QUA, QUI, SEX, SÁB

Buttons at the bottom: Adicionar, Voltar.

Fonte: Autor.

Na Figura 24, após o cadastro, é possível visualizar todas as informações relacionadas à automação, incluindo usuários autorizados, dias e horários de execução, além dos grupos de dispositivos vinculados. No caso de dispositivos infravermelhos de climatização, é possível modificar parâmetros como temperatura e número de acionamentos por ciclo, permitindo uma maior personalização. Essa flexibilidade facilita a adaptação das funções a diferentes cenários de automação, como salas de aula e laboratórios de diferentes características.

Figura 24 – Exibição da Automação.

The screenshot shows the 'Automação 1' details page in the ECP-UFMA system. The page displays the following information:

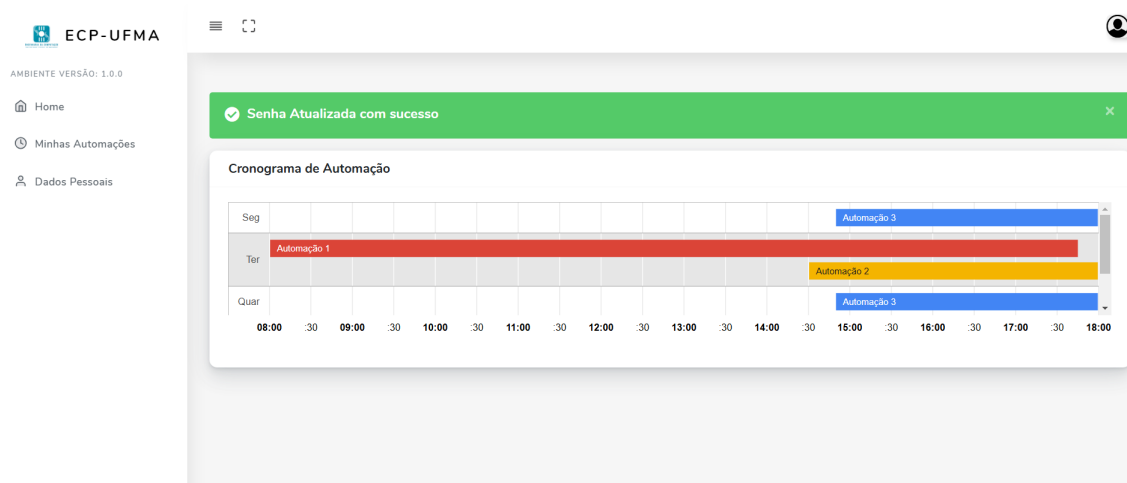
- Sala:** Auditório
- Período:** 2024.2
- Hora Inicial:** 08:00
- Hora Final:** 17:45
- Destaque:** Sim
- Status:** Ativo
- Dias da Semana:** DOM, SEG, TER, QUA, QUI, SEX, SÁB
- Controle Manual:** ON
- Dispositivos Vinculados:** Ar-01, Tópico: POWER1, Grupo: ar-condicionados
- Usuários Autorizados:** Professor 1, Professor 2
- Comandos Infravermelho:** CONTROLE AR-01, Protocolo: LG, Acionamento: Início e Fim, IRHVAC: SIM, Ajustes: 17

Fonte: Autor.

4.1.2 Área do Professor

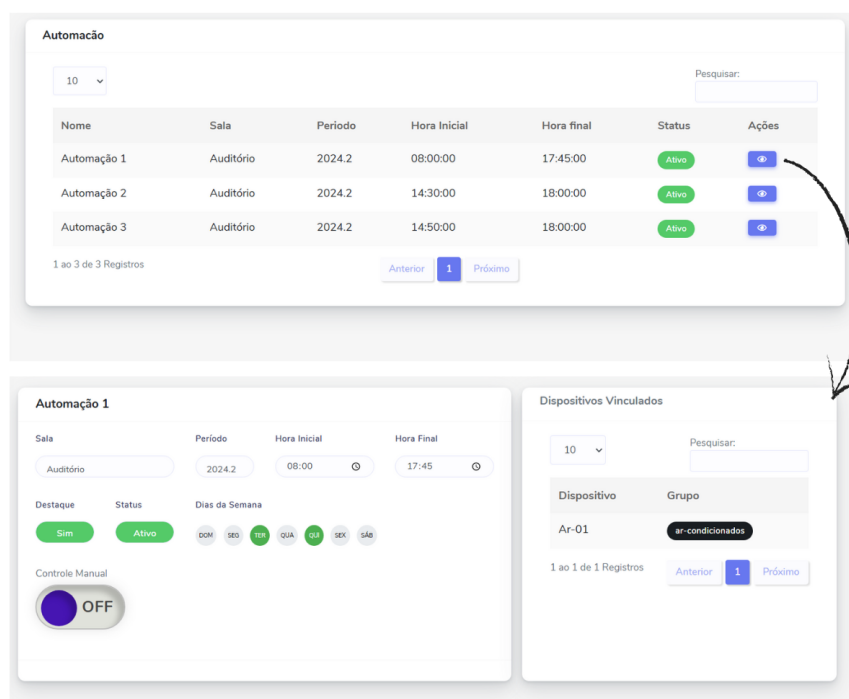
Para o grupo de usuários professores, as Figuras 25 e 26 apresentam alguns dos principais resultados exibidos nas telas do sistema. Essas interfaces reúnem informações para a gestão e controle das automações com permissões concedidas, permitindo a visualização do cronograma de automações programadas, horários agendados e atalhos para acionamento rápido dos dispositivos.

Figura 25 – *Dashboard* do Professor.



Fonte: Autor.

Figura 26 – Automação do Professor.



Fonte: Autor.

4.2 Rotinas das automações

Utilizando como referência as informações dos grupos de dispositivos cadastrados, Figura 27) em consonâncias com a “Automação 01”, apresentadas nas Figuras 19 e 24, é possível verificar, por meio do acompanhamento em laboratório e da consulta à tabela “logs_envios” do banco de dados, as datas e horários dos acionamentos da automação.

Figura 27 – Grupo de Dispositivos.

#	Grupo de Dispositivos	Data do Cadastro	Ultima Alteração	Tópico	Status	Ações
1	ar-condicionados	13/01/2025	13/01/2025	POWER1	Ativo	
2	Lâmpadas	13/01/2025	13/01/2025	POWER2	Ativo	
3	Projetores	13/01/2025	13/01/2025	POWER3	Ativo	
4	Liberado	17/01/2025	17/01/2025	POWER6	Ativo	
6	Atenção	17/01/2025	17/01/2025	POWER7	Ativo	
8	Aguarde	17/01/2025	17/01/2025	POWER8	Ativo	

Fonte: Autor.

A Tabela 8 apresenta esses registros:

Tabela 8 – Registros de acionamento da Automação 1 no período de observação

Tópico	Mensagem	ID Automação	Data e Hora
POWER1	ON	7	2025-01-14 08:00:00
POWER1	OFF	7	2025-01-14 17:45:00
POWER1	ON	7	2025-01-16 08:00:00
POWER1	OFF	7	2025-01-16 17:45:00
POWER1	ON	7	2025-01-21 08:00:00
POWER1	OFF	7	2025-01-21 17:45:00
POWER1	ON	7	2025-01-23 08:00:00
POWER1	OFF	7	2025-01-23 17:45:00
POWER1	ON	7	2025-01-28 08:00:00
POWER1	OFF	7	2025-01-28 17:45:00
POWER1	ON	7	2025-01-30 08:00:00
POWER1	OFF	7	2025-01-30 17:45:00

Fonte: Autor.

Como a automação está programada para ocorrer às terças e quintas, durante o período de análise definido no início da seção 4, foram registrados acionamentos em seis dias distintos, sempre das 8h às 17h45. Considerando tanto as ativações quanto as desativações, o dispositivo “ar-condicionado”, vinculado ao tópico POWER1, foi acionado um total de 12 vezes. Isso confirma que a automação funcionou conforme esperado, acionando o dispositivo em todas as ocasiões previstas no período de observação.

Do mesmo modo, é criada uma automação rápida com um ciclo de dois minutos entre o início e o fim, para permitir o acionamento de diversos dispositivos de diferentes tipos. Na Figura 28, estão registrados os eventos de acionamento, enquanto a Figura 29 exibe o console da aplicação Tasmota recebendo os comandos via MQTT por inscrição e iniciando o processo de acionamento.

Figura 28 – Automação 4: Registros de acionamento

Az topico	Az mensagem	id_automacao	created_at	updated_at
IRSend	{"Protocol":"EPSON","Bits":"32","Data":"0xC1AA09F6"}	22	2025-02-05 17:32:00	2025-02-05 17:32:00
IRHVAC	{"Vendor":"LG","Power":"Off","Temp":null}	22	2025-02-05 17:32:00	2025-02-05 17:32:00
POWER8	OFF	22	2025-02-05 17:32:00	2025-02-05 17:32:00
POWER7	OFF	22	2025-02-05 17:32:00	2025-02-05 17:32:00
POWER6	OFF	22	2025-02-05 17:32:00	2025-02-05 17:32:00
IRSend	{"Protocol":"EPSON","Bits":"32","Data":"0xC1AA09F6"}	22	2025-02-05 17:30:00	2025-02-05 17:30:00
IRHVAC	{"Vendor":"LG","Power":"On","Temp":18}	22	2025-02-05 17:30:00	2025-02-05 17:30:00
POWER8	ON	22	2025-02-05 17:30:00	2025-02-05 17:30:00
POWER7	ON	22	2025-02-05 17:30:00	2025-02-05 17:30:00
POWER6	ON	22	2025-02-05 17:30:00	2025-02-05 17:30:00

Fonte: Autor.

Figura 29 – Console da aplicação Tasmota em inscrição (MQTT)

```
03:17:44.512 MQTT: stat/teste/RESULT = {"POWER6":"OFF"}
03:17:44.517 MQTT: stat/teste/POWER6 = OFF
03:17:44.561 MQTT: stat/teste/RESULT = {"POWER7":"OFF"}
03:17:44.565 MQTT: stat/teste/POWER7 = OFF
03:17:44.612 MQTT: stat/teste/RESULT = {"POWER8":"OFF"}
03:17:44.617 MQTT: stat/teste/POWER8 = OFF
03:17:44.780 MQTT: stat/teste/RESULT = {"IRHVAC":{"Vendor":"LG","Model":-1,"Command":"Control","Mode":"Off","Power":"Off","Celsius":"On","Temp":18,"FanSpeed":"Auto","SwingV":"Off","SwingH":"Off","Quiet":1}}
03:17:44.789 IRIS: protocol 75
03:17:44.123 MQTT: stat/teste/RESULT = {"IRSend":"Done"}
03:19:44.651 MQTT: stat/teste/RESULT = {"POWER6":"ON"}
03:19:44.654 MQTT: stat/teste/POWER6 = ON
03:19:44.707 MQTT: stat/teste/RESULT = {"POWER7":"ON"}
03:19:44.706 MQTT: stat/teste/POWER7 = ON
03:19:44.750 MQTT: stat/teste/RESULT = {"POWER8":"ON"}
03:19:44.754 MQTT: stat/teste/POWER8 = ON
03:19:44.918 MQTT: stat/teste/RESULT = {"IRHVAC":{"Vendor":"LG","Model":-1,"Command":"Control","Mode":"Off","Power":"Off","Celsius":"On","Temp":0,"FanSpeed":"Auto","SwingV":"Off","SwingH":"Off","Quiet":1}}
03:19:44.928 IRIS: protocol 75
03:19:45.259 MQTT: stat/teste/RESULT = {"IRSend":"Done"}
```

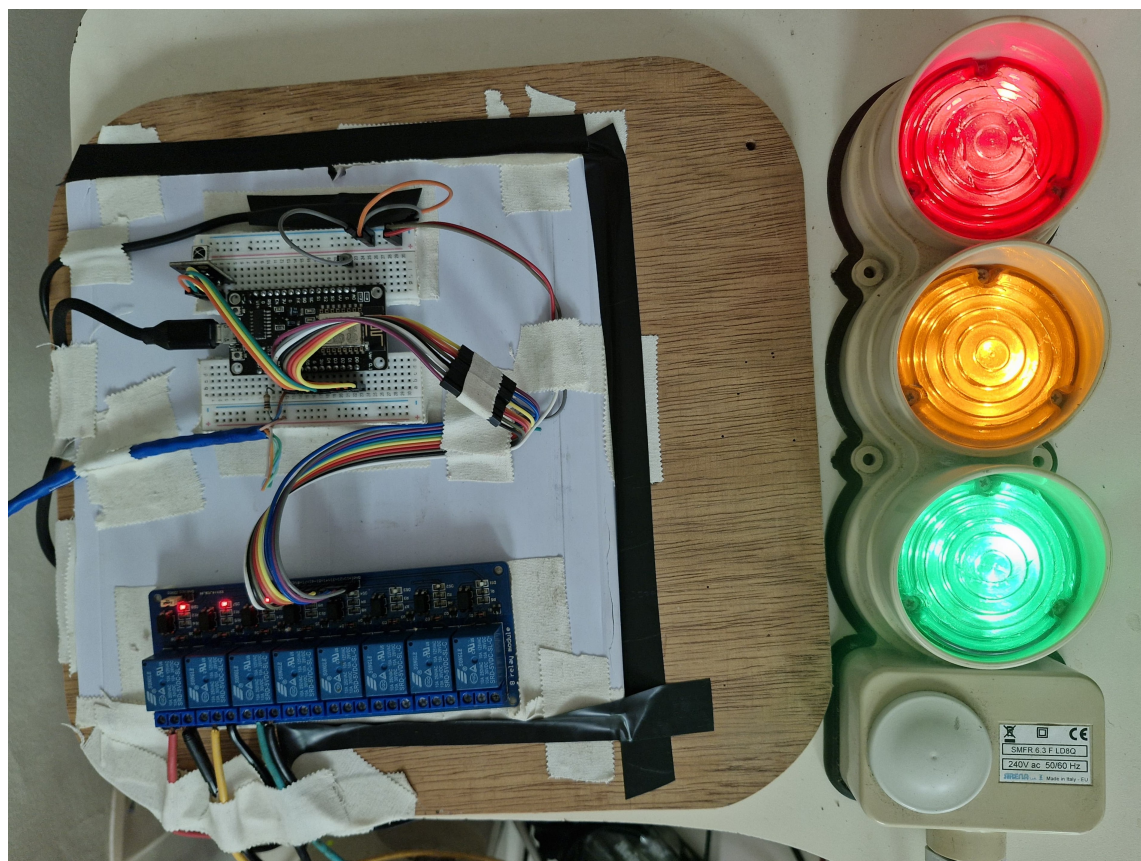
Fonte: Autor.

Observa-se que, após o recebimento do comando, os dispositivos acionados via relés, identificados como POWER6 a POWER8, retornam a mensagem “OK”, indicando o sucesso da operação no *firmware* Tasmota. No caso dos dispositivos infravermelhos, os comandos enviados por IRsend retornam “DONE” (“Feito”, em tradução literal), enquanto os dispositivos do tipo IRHVAC transmitem um sinal codificado próprio.

Como resultado, todos os dispositivos são acionados conforme esperado. A Figura 30 ilustra o acionamento de três dispositivos, representados pelas lâmpadas verde, amarela e

vermelha. Informações adicionais sobre o experimento, incluindo um vídeo demonstrativo¹, estão disponíveis no rodapé.

Figura 30 – Automação 4: Acionamento *Hardware*.



Fonte: Autor.

¹ <https://drive.google.com/file/d/1iMchl-NfsCKohPFJic_B9Y8Icau4FrUS/view?usp=drive_link>

5 Conclusão

O projeto ofereceu uma solução integrada e eficaz para otimizar a gestão. Desenvolveu-se uma aplicação *web* de fácil acesso e configuração simples, mas eficiente, voltada para automação e controle de acesso às instalações. Além disso, garantiu a eficiência energética e a redução da dependência de acesso presencial constante, por meio de ciclos de rotinas alinhados ao aprimoramento no controle da iluminação, climatização e funcionamento dos dispositivos.

Ademais, com base nos resultados obtidos durante o período de observação entre 12 e 31 de janeiro de 2025, a automação do sistema atendeu às expectativas definidas nas etapas de planejamento e execução. Soma-se a isso, uma significativa personalização das automações, onde, por meio de uma interface modular, foi possível adaptar as funções às necessidades específicas do ambiente laboratorial.

Outra característica relevante é que a comunicação e integração via MQTT para o controle simultâneo dos dispositivos infravermelhos e relés demonstraram ser muito satisfatórias, com acionamentos sem falhas, respondendo adequadamente aos comandos de ativação e desativação nas datas e horários programados pela *interface* do usuário administrador.

Por fim, é importante destacar o cumprimento dos objetivos, como o design implementável, levando em consideração a viabilidade do *hardware* a um custo acessível. Ao integrar tecnologias como automação, dispositivos IoT e a aplicação *web*, não apenas se moderniza o laboratório, mas também estabelece-se um modelo inovador para ambientes semelhantes, posicionando a solução como uma alternativa inteligente e abrangente para a gestão de salas e laboratórios. Nesse contexto, planeja-se testar a solução em várias salas e laboratórios da Engenharia da Computação.

Referências

AGUIAR, M. F. de. *Desenvolvimento de um sistema de controle de periféricos via Web para ambiente residencial, utilizando Internet das Coisas (IoT) e o protocolo Message Queuing Telemetry Transport (MQTT)*. 63 p. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) — Universidade Estadual do Amazonas, Manaus, AM, Brasil, 2018. Citado 2 vezes nas páginas 22 e 24.

ALVES, R. C.; CAMPELLO, L. H. P. de; PEREIRA, L. M.; SIMÕES, W. L. A indústria 4.0 no contexto dos estudantes de engenharia de produção: uma análise da produção acadêmica em eventos científicos. *ANAIS DA MOSTRA DE INICIAÇÃO CIENTÍFICA DO CESUCA-ISSN 2317-5915*, n. 16, p. 291–303, 2022. Citado na página 13.

CANTO, F. H. *Vulnerabilidades da linguagem PHP*. Trabalho de Conclusão de Curso (Instituto de Informática, Curso de Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil, 2011. Citado na página 15.

CARDOSO, R. *Laravel: Conheça o framework PHP mais utilizado hoje em dia!* 2023. Código aberto, 5 minutos. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/laravel-framework-php/>>. Citado 2 vezes nas páginas 17 e 18.

CONHAGO, R. T. *Desenvolvimento de um sistema de controle de acesso e acionamento da iluminação e ar-condicionado das salas de aula, utilizando Internet das Coisas (IoT)*. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) — Universidade do Estado do Amazonas, Manaus, AM, Brasil, 2019. Citado 2 vezes nas páginas 22 e 23.

COSTA, I. M.; LISBOA, S. N. D.; SANTOS, T. P. Automação industrial. *DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO-UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE*. Natal, RN, Brasil: Comitê Brasileiro de Barragens, 2003. Citado na página 22.

Eclipse Foundation. *Eclipse Mosquitto: An Open Source MQTT Broker*. 2024. Acessado em: 2 fev. 2025. Disponível em: <<https://mosquitto.org/>>. Citado na página 21.

ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G. et al. *Sistemas de banco de dados*. [S.l.]: Pearson Addison Wesley São Paulo, 2005. Citado na página 16.

FIOZERA. *Microcontroladores*. 2023. Site. Disponível em: <<https://fiozera.com.br/microcontroladores-914a59cbf7de>>. Citado na página 23.

FRANÇA, F. F. *Controlador PID para controle de temperatura de uma carga resistiva AC*. Trabalho de Conclusão de Curso (Instituto de Informática, Curso de Engenharia Elétrica), Guaratinguetá, SP, Brasil, 2018. Citado na página 12.

GABARDO, A. C. *Laravel para ninjas*. [S.l.]: Novatec Editora, 2017. Citado 3 vezes nas páginas 17, 18 e 29.

IBM Developer. *Getting to know MQTT*. 2025. Artigo sobre MQTT e IoT. Disponível em: <<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>>. Citado 3 vezes nas páginas 18, 19 e 20.

JANNUZZI, G. M. Power sector reforms in brazil and its impacts on energy efficiency and research and development activities. *Energy Policy*, v. 3, p. 1753–1762, 2005. Citado na página 22.

JUNIOR, S. L. S.; FARINELLI, F. A. *DOMÓTICA-Automação Residencial e Casas Inteligentes com Arduino e ESP8266*. [S.l.]: Saraiva Educação SA, 2018. Citado na página 21.

KERSCHBAUMER, R. *Microcontroladores*. Santa Catarina, Brasil: [s.n.], 2025. Apostila. Disponível em: <<https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2018/02/Apostila-Microcontroladores.pdf>>. Citado na página 23.

Laravel. *Meet Laravel*. 2023. Documentação oficial do Laravel. Disponível em: <<https://laravel.com/docs/11.x#meet-laravel>>. Citado na página 17.

LIU, X.; ZHANG, T.; HU, N.; ZHANG, P.; ZHANG, Y. The method of internet of things access and network communication based on mqtt. *Computer Communications*, Elsevier, v. 153, p. 169–176, 2020. Citado na página 19.

LOPES, J. F. Aplicação mobile react native para automação residencial utilizando esp8266. 2022. Citado na página 12.

Luiz Adolfo. *Internet das coisas: "Os benefícios são inúmeros"*. Entrevista especial com Luiz Adolfo. 2025. Entrevista publicada no site do IHU Unisinos. Disponível em: <<https://www.ihu.unisinos.br/entrevistas/28619-internet-das-coisas-%60os-beneficios-sao-inumeros-entrevista-especial-com-luiz-adolfo>>. Citado na página 22.

MasterWalker. *Sonoff DIY Mode API Protocol na prática*. 2025. Acesso em: 19 jan. 2025. Disponível em: <<https://blogmasterwalkershops.com.br/embarcados/esp8266/sonoff-diy-mode-api-protocol-na-pratica#:~:text=O%20Tasmota%20%C3%A9%20um%20firmware,protocolo%20MQTT%2C%20HTTP%2C%20serial%20ou>>. Citado na página 25.

MURATORI, J. R.; BÓ, P. H. D. Automação residencial: histórico, definições e conceitos. 2011. In: *Automação residencial*. CUIABÁ: Educere, 2017. p. 1–25. Citado na página 12.

OGATA, K. S.; COIMBRA, H.; TANNURI, E. A. *Engenharia de controle moderno*. [S.l.]: Pearson Prentice Hall, 2010. Citado na página 12.

Oracle. *O que é um banco de dados?* 2025. Site da Oracle. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Citado na página 17.

PHP.NET. *O que é o PHP e o que ele pode fazer?* 2025. Acessado em: 14 jan. 2025. Disponível em: <https://www.php.net/manual/pt_BR/introduction.php>. Citado na página 15.

RODRIGUES, R. O. *Uso do microcontrolador ESP8266 para automação residencial*. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) — Universidade Federal do Rio de Janeiro, Escola Politécnica, Rio de Janeiro, RJ, Brasil, 2017. Citado na página 24.

SILVA, R. O. d.; ARAÚJO, W. M.; CAVALCANTE, M. M. Visão geral sobre microcontroladores e prototipagem com arduino. *Tecnologias em Projeção*, v. 10, n. 1, p. 36–46, 2019. Citado na página 23.

- SILVA, R. T. G.; SALES, A. K. M. *Desenvolvimento de um Sistema de Domótica sem Fio de Baixo Custo Utilizando o Microcontrolador ESP8266*. Trabalho de Conclusão de Curso (Engenharia Mecânica), Guarapuava, PR, Brasil, 2017. Citado na página 24.
- SOFIA, R. C.; MENDES, P. M. An overview on push-based communication models for information-centric networking. *Future Internet*, v. 11, n. 3, p. 74, 2019. Disponível em: <<https://doi.org/10.3390/fi11030074>>. Citado na página 19.
- SONOFF. *Sobre*. 2025. Acesso em: 3 mar. 2025. Disponível em: <<https://www.sonoffbrasil.com.br/post/sobre>>. Citado na página 25.
- SOUZA, O. *Edgar Frank Codd e o Banco de Dados Relacional: uma contribuição para a História da Computação*. Dissertação (Dissertação (Mestrado em História da Ciência)) — Pontifícia Universidade Católica de São Paulo, 2015. Acesso aberto. Idioma: Português. Disponível em: <[lattes](#)>. Citado na página 15.
- SYSTEMS, C. D. *Tensilica Controllers and Extensible Processors*. 2025. Acesso em: 3 mar. 2025. Disponível em: <https://www.cadence.com/en_US/home/tools/silicon-solutions/compute-ip/tensilica-xtensa-controllers-and-extensible-processors.html>. Citado na página 24.
- Tasmota. *Tasmota Documentation*. 2025. Acesso em: 19 jan. 2025. Disponível em: <<https://tasmota.github.io/docs/About/>>. Citado na página 25.
- TEDESCO, M. J.; SANTOS, A. D. dos; GIANELLO, C.; SILVA, F. C. da; CUNHA, L. M. S.; HIGA, R. H.; MASSRUHÁ, S. Infra-estrutura laboratorial, gerenciamento de laboratório e automação da informação. 2009. Citado na página 12.