

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Interruptor Inteligente controlado via rede

João Pedro Doimo Torrezan



São Carlos – SP

Interruptor Inteligente controlado via rede

João Pedro Doimo Torrezan

***Orientador:* Prof. Dr. Júlio Cezar Estrella**

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Engenheiro de Computação.

Área de Concentração: Sistemas Distribuídos

USP – São Carlos

Outubro de 2021

Torrezan, João Pedro Doimo
Interruptor Inteligente controlado via rede / João
Pedro Doimo Torrezan. - São Carlos - SP, 2021.
48 p.; 29,7 cm.

Orientador: Júlio Cezar Estrella.
Monografia (Graduação) - Instituto de Ciências
Matemáticas e de Computação (ICMC/USP), São Carlos -
SP, 2021.

1. IoT. 2. MQTT. 3. JSON. 4. Iluminação.
5. Microcontrolador. I. Estrella, Júlio Cezar. II.
Instituto de Ciências Matemáticas e de Computação
(ICMC/USP). III. Título.

Dedico este trabalho aos meus pais Karina e Fernando e, em especial, ao meu irmão Bruno, que me apoiaram em todas as decisões e escolhas tomadas, sem eles não estaria aqui.

AGRADECIMENTOS

Agradeço acima de tudo à minha família pelo apoio e incentivo não só durante a graduação, bem como em todas as etapas do meu desenvolvimento e, em especial, aos meus pais que batalharam muito para a concretização disso. Sou grato também a todos os meus amigos que trilharam esse caminho ao meu lado proporcionando momentos de muita alegria e felicidade, vale acrescentar ainda que sem estes não haveria êxito nessa jornada. Agradeço também aos professores pela sabedoria a mim transmitida e, especialmente, ao professor Júlio César Estrella pela disponibilidade na orientação do trabalho e disponibilização de recursos para sua realização.

“O sucesso é ir de fracasso em fracasso sem perder o entusiasmo.”
(Winston Churchill)

RESUMO

TORREZAN, J. D. . **Interruptor Inteligente controlado via rede**. 2021. 48 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Instituições e residências tendem a trocar seus dispositivos de iluminação por dispositivos *smart*, para que seja possível a gestão e a operação por meio da rede de dados. Segundo (W. ALEXIS Z.; M., 2019), em 2023 a indústria de automação estará presente em 53,9% das residências estadunidenses. Este trabalho apresenta uma alternativa barata, eficiente e durável a essa substituição de dispositivos. O dispositivo desenvolvido conta com um microcontrolador que possui acesso à rede Wi-fi e permite ao usuário o controle do sistema de iluminação presente através da *Internet*, sem a necessidade da substituição dos aparelhos.

Palavras-chave: IoT, MQTT, JSON, Iluminação, Microcontrolador.

ABSTRACT

TORREZAN, J. D. . **Interruptor Inteligente controlado via rede**. 2021. 48 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Institutions and households tend to exchange their lighting devices for smart ones, so it can be managed and operated through the network. According to (W. ALEXIS Z.; M., 2019), in 2023 the penetration rate of smart home tech in the United States will be around of 53.9%. This work presents a cheap, efficient, and durable alternative to this replacement. The device developed has a microcontroller that has access to the Wi-Fi network and allows the user to control the lighting system via Internet, without the need to replace the devices itself.

Key-words: IoT, MQTT, JSON, Lighting, Microcontrollers.

LISTA DE ILUSTRAÇÕES

Figura 1 – ESP8266	22
Figura 2 – ESP01	23
Figura 3 – Módulo relé.	24
Figura 4 – Módulo relé com ESP01	24
Figura 5 – Microcontrolador ESP32.	25
Figura 6 – Diagrama da comunicação para o HTTP	25
Figura 7 – Diagrama da comunicação usando <i>MQTT</i>	26
Figura 8 – Interface Node-RED	27
Figura 9 – Foto do dispositivo elaborado.	34
Figura 10 – Foto do ESP01 com as ligações elétricas feitas.	34
Figura 11 – Diagrama do funcionamento do sistema.	37
Figura 12 – Esquemático do dispositivo que envolve o ESP32.	38
Figura 13 – Gráfico de comparação entre o estado enviado e recebido como resposta. . .	41
Figura 14 – Gráfico do tempo de reinicialização do dispositivo.	42
Figura 15 – Telas de conexão a novas redes.	43

LISTA DE ABREVIATURAS E SIGLAS

<i>IoT</i>	<i>Internet of Things</i>
BIM	Building Information Modeling
HTTP	Hypertext Transfer Protocol
IAU	Instituto de Arquitetura e Urbanismo
IBM	<i>International Business Machines Corporation</i>
ICMC	Instituto de Ciências Matemáticas e de Computação
ICMC	Instituto de Ciências Matemáticas e de Computação
JSON	JavaScript Object Notation
LaSDPC	..	Laboratório de Sistemas Distribuídos e Programação Concorrente
LDR	<i>Light Dependent Sensor</i>
MQTT	...	Message Queuing Telemetry Transport
NTP	<i>Network Time Protocol</i>
OASIS	...	Organization for the Advancement of Structured Information Standards
PoE	<i>Power over Ethernet</i>
QoS	Quality of Service
RISC	<i>Reduced Instruction Set Computer</i>

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Contextualização	19
1.2	Motivação	19
1.3	Objetivos	20
1.4	Organização	20
2	MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS	21
2.1	Considerações Iniciais	21
2.2	Internet das Coisas	21
2.2.1	ESP8266	21
2.2.2	ESP01	22
2.2.3	Módulo Relé	23
2.2.4	ESP32	23
2.3	Protocolos de Comunicação	24
2.3.1	Hypertext Transfer Protocol - HTTP	24
2.3.2	Message Queuing Telemetry Transport - MQTT	26
2.4	Node-RED	27
2.5	Cluster - Andromeda	28
2.6	Considerações Finais	28
3	TRABALHOS RELACIONADOS	29
3.1	Considerações Iniciais	29
3.2	Trabalhos Relacionados	29
3.3	Considerações Finais	31
4	DESENVOLVIMENTO	33
4.1	Considerações Iniciais	33
4.2	O Projeto	33
4.3	Atividades Realizadas	35
4.3.1	Implementação	35
4.3.2	Broker MQTT	36
4.3.3	Montagem do dispositivo físico	37
4.3.4	Testes executados	37
4.4	Considerações Finais	39

5	RESULTADOS	41
5.1	Considerações Iniciais	41
5.2	Primeiro Experimento	41
5.3	Segundo Experimento	42
5.4	Terceiro Experimento	43
5.5	Publicações	44
5.6	Considerações Finais	44
6	CONCLUSÃO	45
6.1	Contribuições	45
6.2	Trabalhos Futuros	45
	REFERÊNCIAS	47

INTRODUÇÃO

1.1 Contextualização

A popularização da Internet das Coisas (do inglês, *Internet of Things* (*IoT*)) está transformando a interação entre humanos e máquinas bem como entre as máquinas. Com a expansão das redes de comunicação, a tendência é uma mudança completa dessas interações e por consequência, a abertura de um grande leque de oportunidades de modernizações. A monografia em questão traz uma dessas possibilidades dentre as inúmeras que estão por vir.

Todavia, atrelada à inovação, há a barreira do custo e quando o tema tratado é *IoT* há duas possíveis opções: a substituição completa de aparelhos não dotados de conexão com a rede, por aparelhos que possuam essa característica; ou a adição dessa característica ao dispositivo antigo. Inegavelmente, a segunda opção apresenta, na vasta maioria dos casos, um custo muito inferior frente à outra saída, já que os dispositivos necessários para a mudança proposta apresentam um preço relativamente baixo.

Com isso, surge a ideia do trabalho, que consiste na implementação de um dispositivo com conexão a *Internet* para realizar o controle da iluminação de ambientes, contando ainda com um baixo preço e boa eficiência.

1.2 Motivação

A motivação do trabalho foi a criação de um grupo de pesquisa, por parte do professor Júlio Cezar Estrella, cujo objetivo era automatizar completamente um dos laboratórios do Instituto de Ciências Matemáticas e de Computação (ICMC). Esse objetivo foi definido como inicial já que, após conseguir atingi-lo, os projetos envolvidos poderiam ser escalados e assim abranger áreas maiores, como por exemplo toda a gestão de recursos de infraestrutura do ICMC, ou até de todo campus.

Um dos módulos do grupo de pesquisa era o responsável pelo controle da iluminação do ambiente, daí então surgiu a motivação para esse trabalho.

1.3 Objetivos

O objetivo do trabalho é o desenvolvimento de um dispositivo de baixo custo que possa controlar a iluminação de ambientes, fazendo uma gestão eficiente de energia e trazendo a possibilidade do acompanhamento da qualidade da iluminação, visto que o dispositivo conta com um sensor de luminância, permitindo possíveis manutenções logo ao surgimento de defeitos na lâmpada. O projeto deve ainda permitir fácil integração com sistemas de gerência externos, para que assim possa haver uma adesão simplificada com sistemas de gestão de ambientes.

1.4 Organização

Essa monografia está estruturada da seguinte forma: no [Capítulo 2](#), apresentam-se os métodos, técnicas e tecnologias utilizadas no desenvolvimento deste trabalho. Em seguida, no [Capítulo 3](#), há a exposição de trabalhos relacionados à essa monografia, contendo uma análise da metodologia utilizada. Já no [Capítulo 4](#), apresenta-se a descrição da implementação e dos testes para a avaliação de desempenho. No [Capítulo 5](#), são discutidos os resultados obtidos nestes testes, e por fim, no [Capítulo 6](#), tem-se a conclusão desta monografia, bem como os trabalhos futuros e a relação com o curso de graduação, além de um comentário sobre o curso.

MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS

2.1 Considerações Iniciais

Este capítulo tem o intuito de enumerar e explicar todas as tecnologias e dispositivos utilizados na elaboração e desenvolvimento do trabalho. Inicialmente, será apresentada uma visão geral do projeto e seus principais conceitos. Tanto os protocolos de comunicação, como os dispositivos de *hardware* utilizados serão apresentados. Os protocolos utilizados foram *HTTP* e *MQTT*, já os dispositivos de *hardware* foram o ESP8266 e o ESP32, todos fabricados e distribuídos pela empresa Espressif Systems¹.

2.2 Internet das Coisas

A Internet das Coisas, comumente chamada de *IoT*, sigla do termo em inglês para *Internet of Things* descreve o conjunto de dispositivos físicos dotados de sensores, recursos de processamento, acesso à rede, dentre outras tecnologias. Esses dispositivos coletam e processam dados do mundo real, posteriormente enviando-os a outros sistemas ou dispositivos através, na maioria das vezes, da *Internet* (GILLIS, 2021). Esse é um conceito em voga nos dias atuais graças à constante melhoria e barateamento dos dispositivos de *hardware*, além da grande dispersão das redes e da necessidade constante da geração de dados para futura extração de informação destes.

Uma outra definição de *IoT* é dada em (GUBBI *et al.*, 2013), em que o conceito é dito como a mistura perfeita de sensores e atuadores com o ambiente ao nosso redor, permitindo assim que informações ambientais sejam compartilhadas entre plataformas, possibilitando, com isso, a gestão e o controle mais inteligentes dos recursos. No geral, dispositivos *IoT* possuem pequenas dimensões e baixo consumo de energia, para que a sua implementação em um sistema real possa realmente ser uma mistura perfeita, como citado pelo autor.

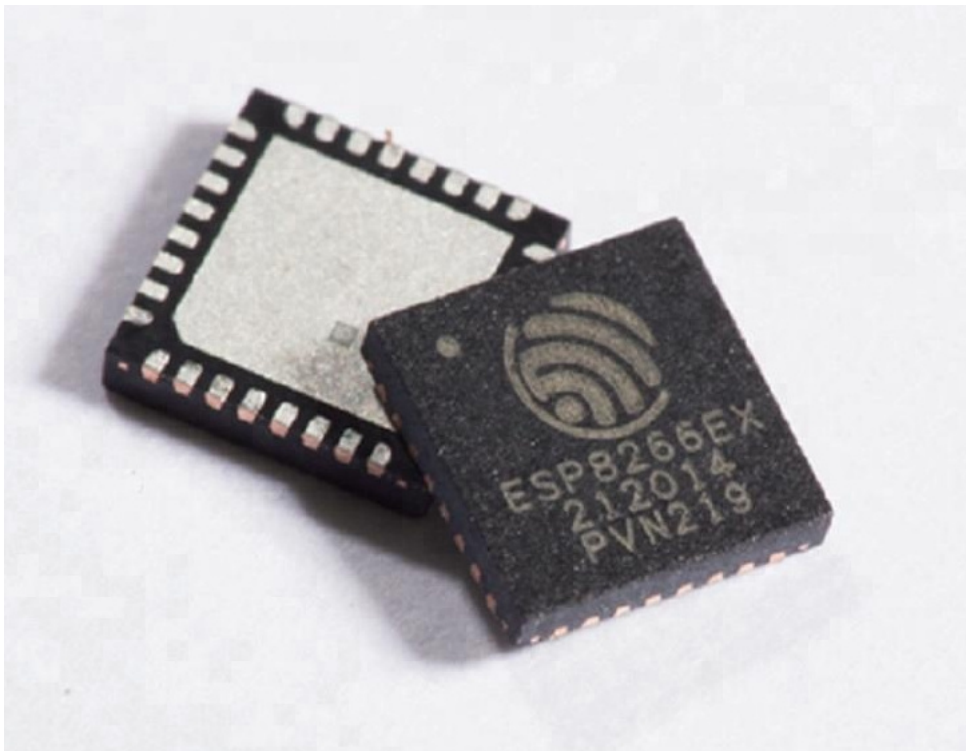
2.2.1 ESP8266

ESP8266 é um microcontrolador de baixo custo com interface de rede incluída. Desenvolvido e produzido pela multinacional chinesa Espressif Systems (ESP8266, 2021). Esse

¹ [Espressif Systems](#)

microcontrolador é muito popular dentre os entusiastas de *IoT*, dado o seu baixo custo, dimensões e possibilidades de uso. Além disso, é um microcontrolador com grande possibilidade de uso nas indústrias, já que possui a robustez para operar em um grande intervalo de temperaturas, contando ainda com um processador *Reduced Instruction Set Computer* (RISC) de 32 bits, o qual pode chegar a 160MHz de frequência de *clock* com uma tensão de operação de 3.3 volts, além de haver possibilidade de implementação da pilha TCP/IP. A [Figura 1](#) ilustra o *System on chip* descrito.

Figura 1 – ESP8266

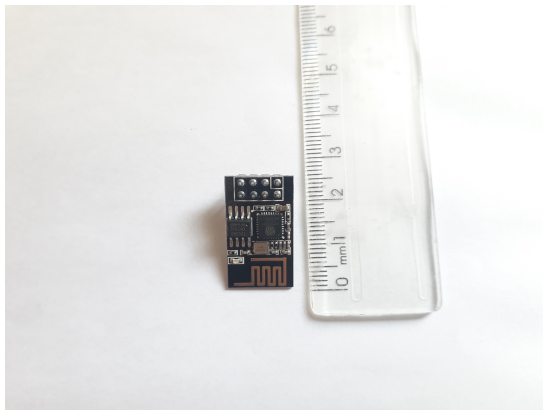


Fonte: ([ESP8266](#), 2021)

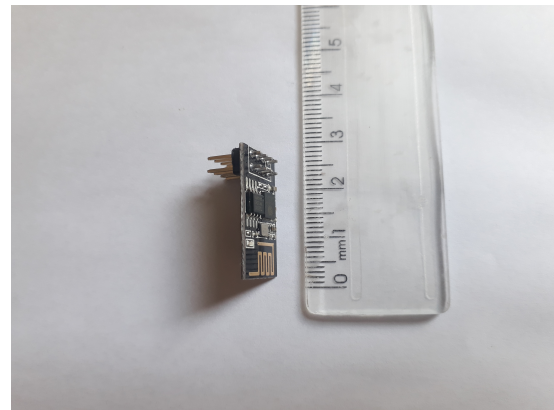
2.2.2 ESP01

Existem diversas montagens possíveis de placas que incluem o *chip* citado anteriormente, como por exemplo a NodeMCU, uma das mais conhecidas e utilizadas. Todavia, a placa utilizada para o projeto do Interruptor Inteligente foi a ESP01, visto que dentre todas as placas disponíveis era a mais compacta, permitindo assim ser alojada dentro de um interruptor já existente, sem necessidade de alterações na estrutura deste. A [Figura 2](#) ilustra a placa utilizada, nela podem ser observadas as dimensões, que valem 25mm de comprimento e 15mm de largura, aproximadamente.

Figura 2 – ESP01



(a) Vista superior da placa ESP01



(b) Vista lateral da placa ESP01

Fonte: Elaborado pelo Autor

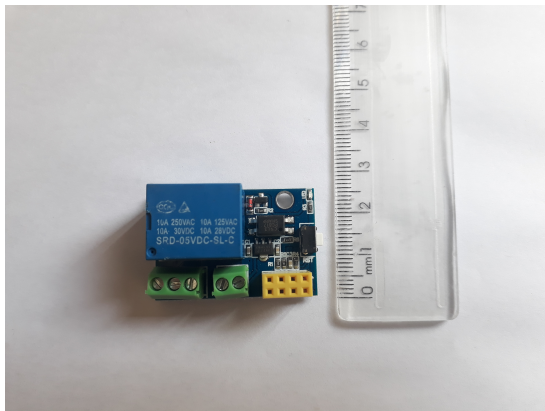
2.2.3 Módulo Relé

Um relé é um interruptor eletromecânico, composto por uma bobina, que ao ser percorrida por corrente elétrica, gera um campo magnético, atraindo assim a alavanca metálica que é responsável pela mudança de estado elétrico dos contatos. Essa ferramenta é utilizada no projeto com o intuito de interromper a passagem de corrente na rede elétrica que alimenta a lâmpada a ser controlada. O relé está integrado em um módulo que é acoplado ao microcontrolador ESP01 descrito acima. Nesse módulo existe, além dos pontos de contato do relé, dois pontos de contato de entrada de tensão, que proveem energia tanto ao relé, quanto ao ESP01. Esses contatos devem ser alimentados com uma diferença de potencial de, no máximo, 15 volts, pois um dos componentes da placa de circuito do módulo relé é um regulador de tensão AMS1117-3v3, o qual tem a função de manter fixa a tensão em 3.3 volts, mantendo a operabilidade do relé e também do microcontrolador. Na [Figura 3](#) encontram-se duas imagens do módulo relé, já na [Figura 4](#) há o módulo relé integrado ao microcontrolador. Esse conjunto possui 37mm de comprimento, 25mm de largura e 17mm de altura.

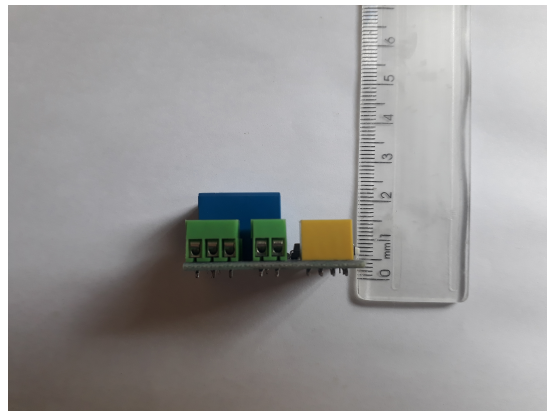
2.2.4 ESP32

ESP32 é um microcontrolador dotado de uma interface de rede, assim como o ESP8266 descrito anteriormente, entretanto, esse microcontrolador conta com alguns recursos a mais, como por exemplo, a possibilidade do uso da rede *Bluetooth*. Conta com a mesma robustez, baixo consumo de energia e baixo custo de seu antecessor ESP8266, sendo fabricado pela mesma empresa multinacional, Espressif System ([ESP32, 2021](#)). A [Figura 5](#) mostra a placa montada com esse microcontrolador, a qual foi utilizada para o desenvolvimento de um dos protótipos.

Figura 3 – Módulo relé.



(a) Vista superior do módulo relé.



(b) Vista lateral do módulo relé.

Fonte: Elaborado pelo Autor.

Figura 4 – Módulo relé com ESP01



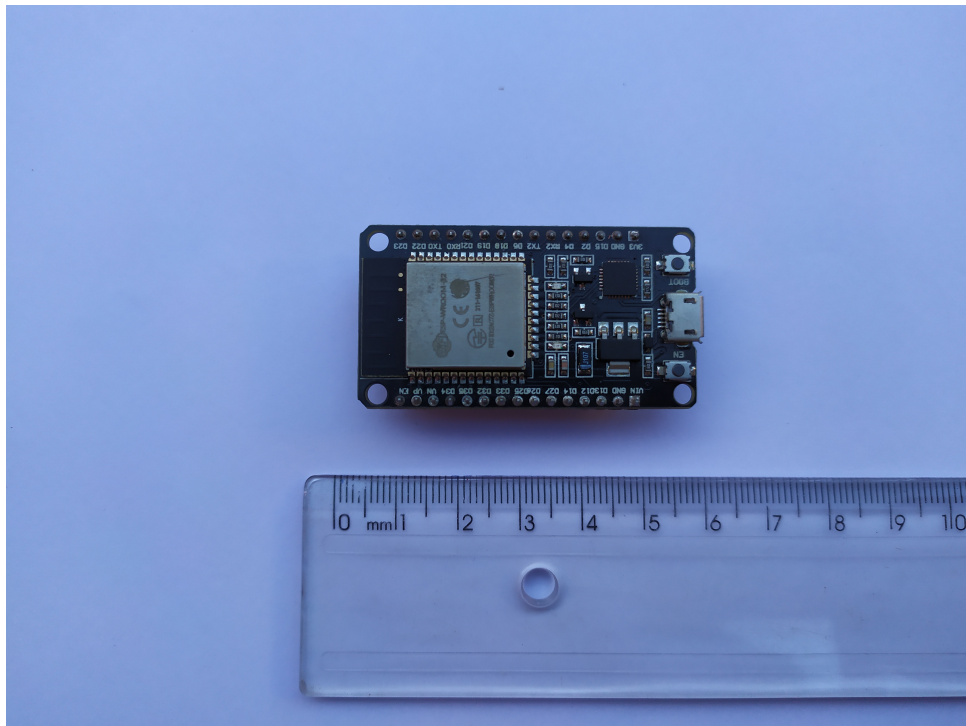
2.3 Protocolos de Comunicação

Nesta seção, serão descritos os protocolos de comunicação utilizados na implementação do trabalho.

2.3.1 *Hypertext Transfer Protocol - HTTP*

O protocolo *Hypertext Transfer Protocol (HTTP)* é implementado na camada de aplicação com a intenção de transmitir hipermídia. Ele vem sendo usado desde 1990 e utiliza um modelo

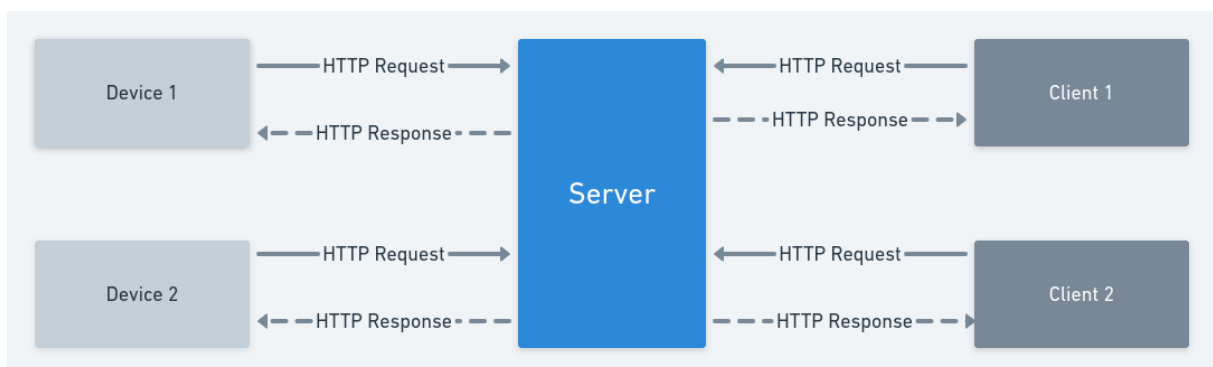
Figura 5 – Microcontrolador ESP32.



Fonte: Elaborado pelo Autor.

cliente-servidor, em que o primeiro faz uma requisição e aguarda enquanto o servidor encaminha a resposta. A [Figura 6](#) representa o processo de comunicação entre as partes. *HTTP* é um protocolo amplamente usado na comunicação web, principalmente, entre *browsers* e servidores.

Figura 6 – Diagrama da comunicação para o HTTP



Fonte: Elaborado pelo Autor

A quantidade de *headers* dos pacotes desse protocolo pode variar e atingir um grande número, visto que, por padrão, não há limite nem no tamanho de cada um dos *headers* nem em sua quantidade. Porém, no caso de servidores como o Apache 2.3 há o limite de, no máximo, 100 campos, cada um podendo conter até 8190 *bytes*. Cada um deles é composto por um dupla chave-valor em texto plano, ou seja, fazendo o uso de *strings*.

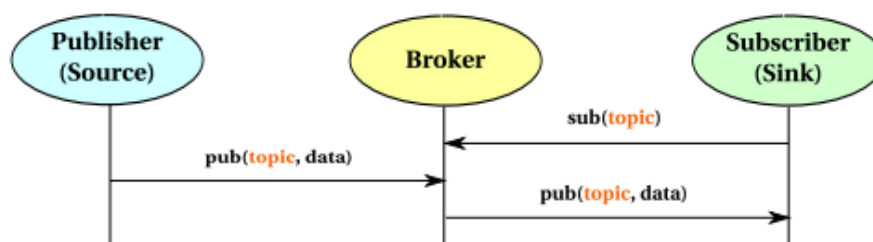
2.3.2 Message Queuing Telemetry Transport - MQTT

Message Queuing Telemetry Transport (MQTT) é um protocolo padronizado pela *Organization for the Advancement of Structured Information Standards (OASIS)*, desenvolvido visando ser extremamente leve, usando a metodologia *publish-subscribe*, economizando em recursos e na banda de *Internet* usada. Sua primeira versão foi publicada em 1999 pela *International Business Machines Corporation (IBM)*.

A sua arquitetura possui duas entidades, o *Broker* e os clientes. O primeiro é um servidor responsável por receber e distribuir as mensagens aos devidos destinatários, além de se manter em contato com os clientes para que seja possível notar uma eventual desconexão. Já o cliente é responsável por se conectar ao *Broker* e publicar e/ou se inscrever em tópicos para recepção de mensagens. O protocolo foi desenvolvido de maneira que a implementação por parte do cliente fosse algo simples, favorecendo desse modo dispositivos com limites de processamento e de armazenamento, como os presentes nos dispositivos *IoT* (HUNKELER; TRUONG; STANFORD-CLARK, 2008). Toda a complexidade do protocolo se dá no *Broker*, que deve ser implementado em um servidor com maior capacidade de processamento e armazenamento.

A Figura 7 ilustra o funcionamento da troca de mensagens entre os clientes usando o protocolo. Como pode ser visto, um cliente publica alguma mensagem em um tópico específico e todos os outros clientes que estavam previamente inscritos nesse tópico recebem a mensagem enviada através do *Broker*.

Figura 7 – Diagrama da comunicação usando MQTT



Fonte: Hunkeler, Truong e Stanford-Clark (2008).

Dentro da temática do envio das mensagens, o protocolo conta com um importante recurso, o *Quality of Service (QoS)*, que varia em três níveis: QoS 0 o qual indica que a mensagem será enviada aos devidos *subscribers* sem uma posterior confirmação; QoS 1 indica que a mensagem será enviada até receber uma confirmação dos clientes que deveriam tê-la recebido, podendo haver mais de uma recepção da mesma mensagem; QoS 2 indica que a mensagem será recebida pelos clientes e não somente recebida, como recebida uma única vez, sendo portanto a mais custosa em termos de tempo. Cabe ao desenvolvedor do sistema escolher qual *Quality of Service* será utilizado para cada uma das mensagens enviadas.

Na monografia em questão, o *Broker MQTT* utilizado foi o Aedes, que foi suportado em nuvem, para a solução ser mais escalável, entretanto, o *Broker* poderia estar em um dispositivo local de *hardware*, como uma Raspberry Pi.

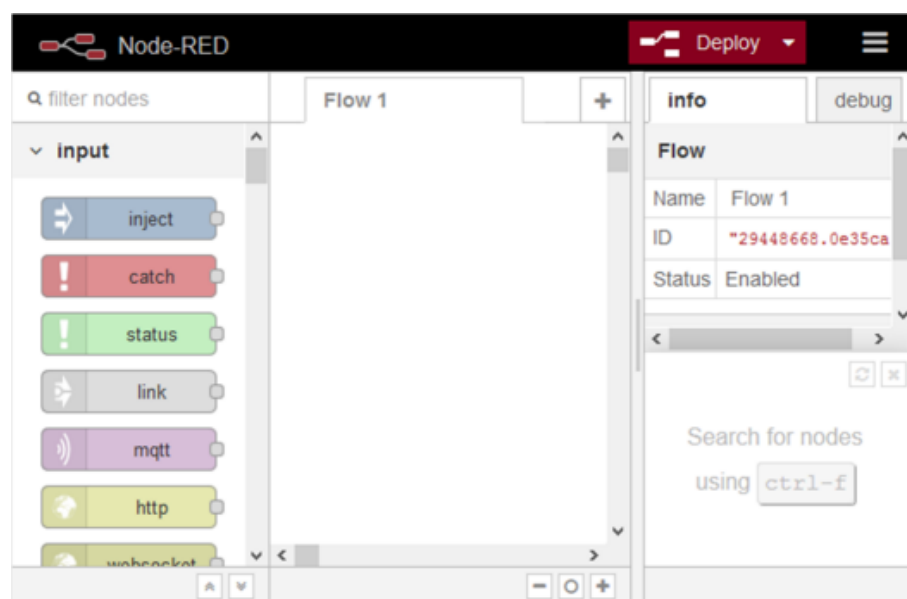
2.4 Node-RED

Node-RED é um *framework* de desenvolvimento *Open-Source* construída com o Node.js (NODE-RED, 2021). Sua interface é exibida em *browser*, como pode ser visto na Figura 8. Esse *framework* possui ferramentas muito intuitivas e as ações são nele executadas usando a metodologia *drag and drop*, o que torna o processo muito mais rápido. Além disso, todo o desenvolvimento é baseado no fluxo de informações.

Essa ferramenta possui nós que realizam diferentes funções, como por exemplo o nó de *debug*, que ilustra em tela as informações que passam por ele. Nós podem ser criados para que executem funções específicas, definidas pelo usuário, utilizando a linguagem de programação *JavaScript*. A ferramenta permite também o consumo de APIs, a integração com serviços *web* e a leitura/escrita do estado de um pino de saída de um microcontrolador, dentre muitas outras funções. (LEKIĆ; GARDAŠEVIĆ, 2018)

A ferramenta foi disponibilizada em nuvem e teve como *host* o Cluster Andromeda, situado no laboratório Laboratório de Sistemas Distribuídos e Programação Concorrente (LaSDPC).

Figura 8 – Interface Node-RED



Fonte: Lekić e Gardašević (2018).

2.5 *Cluster* - Andromeda

Disponibilizado no laboratório LaSDPC ¹ foi responsável por hospedar todos os serviços que se encontravam em nuvem, com isso, tornando a solução posteriormente mais escalável e ainda contribuindo com o desenvolvimento. O *cluster* está situado no endereço andromeda.lasdpc.icmc.usp.br e algumas portas foram alocadas para que os serviços lá disponibilizados pudessem ser acessados, como por exemplo o Node-RED que se encontra na porta 1880 e o *Broker* Aedes, localizado na porta 1883.

O Cluster se encontra fisicamente nos laboratórios 1-008 e 1-006 do bloco 1 do Instituto de Ciências Matemáticas e de Computação (ICMC).

2.6 Considerações Finais

Esse capítulo teve a função de introduzir vários dos conceitos utilizados no desenvolvimento dessa monografia. Com o esclarecimento deles, podemos seguir para o [Capítulo 4](#).

¹ [LaSDPC](#)

TRABALHOS RELACIONADOS

3.1 Considerações Iniciais

Serão citados, nessa seção, alguns artigos relacionados ao tema central deste trabalho. Tratando-se de projetos relacionados à rede e à *IoT* é possível observar uma vasta gama de estudos e outros projetos com intuitos parecidos, que seriam a geração de dados, o controle mais eficiente de dispositivos, a economia de energia e também a melhor gestão de problemas que surgem ao se escalar o número de dispositivos a serem controlados.

3.2 Trabalhos Relacionados

Em (LOBACHEV; CRETU, 2016), encontramos uma rede de sensores utilizados em construções para geração de dados de diferentes tipos, como: umidade, temperatura, luminosidade, dados vindos de acelerômetros, dentre outros. O sistema proposto conta com a possibilidade de se cadastrar vários dispositivos contendo um ou mais sensores, visto que, segundo a arquitetura proposta, um único dispositivo pode ser entendido como um *cluster* de sensores.

Este trabalho apresenta soluções para alguns dos problemas encontrados no desenvolvimento de dispositivos *IoT*, como por exemplo o fornecimento de energia aos dispositivos e também a dificuldade do acesso à rede. O trabalho em questão elucida a desvantagem da utilização de redes sem fio para grandes construções, visto que suas estruturas e a grande quantidade de dispositivos eletrônicos presentes podem interferir na transmissão do sinal pelo ar, causando então, instabilidade ao sistema de sensores proposto. A solução para esse problema foi o uso do *Power over Ethernet* (PoE) em que há um cabeamento único que garante acesso à rede e à energia para a alimentação dos dispositivos. Além disso, outro aspecto muito importante citado nesse trabalho é a necessidade de uma hierarquia sobre os dispositivos. A hierarquia abrange aspectos como restrições de acesso ao dispositivo, nível de controle e localização de erros, aumentando a facilidade de manutenção, assim como melhorando a usabilidade do sistema como um todo.

O sistema desenvolvido no trabalho conta ainda com servidores de coleta, armazenamento e distribuição de dados alocados em nuvem, o que o torna algo muito mais escalável, que é fator crucial na implementação de projetos *IoT*.

Em (RAMPRASAD *et al.*, 2018) há uma proposta de sistema para coleta, tratamento,

encaminhamento e armazenamento de dados gerados por "*Smart Buildings*". A solução proposta envolve a criação de um *framework* que possa gerir dados de diversos tipos e aspectos, tendo em vista que os *softwares* existentes possuem regras fixas e rígidas no tratamento dos dados, impedindo assim o uso de sensores e dispositivos genéricos. Essa solução é dividida em duas partes, o pré-processamento na borda da aplicação e o armazenamento e processamento final em nuvem. O pré-processamento dos dados diretamente na borda do sistema traz uma série de vantagens, visto que, por se tratar de uma fase do sistema em que a quantidade de dados a ser processada é pequena, a complexidade desse processamento diminui e assim um possível alerta de anormalidade é mais facilmente tratado. Além disso, a borda do sistema deve apresentar uma baixa latência e um alto *throughput* já que esses dispositivos são responsáveis pela geração de dados e por um pequeno processamento desses. A borda do sistema envolve apenas os sensores e algum controlador, este podendo ser uma *Raspberry Pi*.

Quando analisado o lado da nuvem, da solução proposta, sabe-se que o poder de processamento e de armazenamento aumenta, isso possibilita a análise dos dados feita na gerência dos recursos dos "*Smart Buildings*". Possuindo os dados de todos os sensores, é possível criar alertas e rotinas para grupos de sensores, dividindo-os, por exemplo, entre andares, salas ou até blocos de um possível prédio. Possibilitando assim uma gestão mais inteligente e, por consequência, um consumo de energia otimizado.

Em (LI; LOGENTHIRAN; WOO, 2016) apresenta-se um aplicativo para controle dos dispositivos *IoT* de um residência, o qual cadastra dispositivos inteligentes e se comunica com eles através do *Bluetooth*. Este controle provê uma economia de energia das residências, além de aumentar o conforto do morador, adicionando a possibilidade do controle de muitos aparelhos elétricos de sua casa por intermédio do celular.

Em (DOUKAS; ANTONELLI, 2013), tem-se a proposta de um *framework* chamado COMPOSE, o qual conta com a possibilidade do desenvolvimento de aplicações que entendam e utilizem informações do contexto do usuário, como por exemplo a sua localização, a temperatura e a umidade dessa região. O *Framework* pode utilizar essas informações com três finalidades: apresentar informações sobre o contexto ou então propor ações baseadas no contexto do usuário; realizar um serviço previamente estabelecido pelo usuário de acordo com a variação do seu contexto; distinguir dados gerados de acordo com o contexto em que estes foram obtidos, agregando mais informação aos dados.

Esse *framework* interage com os dispositivos *IoT* em uma relação bidirecional, através de HTTP usando uma metodologia REST. Com isso, os dispositivos podem receber dados do servidor, para que ações sejam executadas pelos atuadores, assim como enviar dados utilizando do mesmo protocolo.

Em (ALSUHLI; KHATTAB, 2019) apresenta-se uma rede de sensores e atuadores responsáveis por coletar dados de temperatura, umidade, ventilação e iluminação, permitindo fazer a gestão dos atuadores a fim de manter esses dados dentro de um intervalo aceitável de

forma eficiente energeticamente. Para isso, usam de um pré-processamento anterior ao envio dos dados à nuvem, esse pré-processamento é chamado de *Fog*, nele os dados são agrupados e convertidos a uma forma compreensível e tratável.

Os dispositivos de borda coletam esses dados e os enviam para o processamento na *Fog* através do protocolo MQTT, já que é um protocolo de comunicação muito leve e rápido, necessitando, entretanto, de um *broker*, o qual era servido nas mesmas máquinas que disponibilizavam o serviço da *Fog*, a qual realiza também o *feedback* aos dispositivos de borda para que esses mantenham os dados em níveis desejados por meio dos atuadores disponíveis.

A computação e o pré-processamento dos dados foram feitos utilizando um *software* chamado Node-RED. Já a nuvem conta com uma API em PHP para armazenamento e consulta dos dados já processados pela *Fog*, nos bancos de dados.

3.3 Considerações Finais

Este capítulo conta com trabalhos cujos temas estão relacionados com a monografia desenvolvida. Nota-se que o tema Internet das Coisas é amplamente abordado e estudado, pois existem muitas possibilidades de mercado para essas inovações, sendo no desenvolvimento de novos produtos ou na melhoria dos já existentes. Esta monografia tem como intuito apresentar um sistema contando com baixo custo, que permita a gestão de sistemas de iluminação e que viabilize uma integração e comunicação simples com *softwares* externos.

DESENVOLVIMENTO

4.1 Considerações Iniciais

Este capítulo tem por função explicar o projeto e os processos envolvidos ao seu desenvolvimento, além de dissertar sobre os testes pelos quais os dispositivos desenvolvidos passaram.

4.2 O Projeto

O objetivo desse trabalho é o desenvolvimento de uma ferramenta barata e eficiente para o controle de iluminação de ambientes através da *Internet*. Visando tais objetivos foi elaborada uma solução envolvendo os microcontroladores ESP8266-01 e o NODE-MCU ESP32, os quais possuem preços baixos e ótima eficiência, contando nativamente com interface de rede, boa capacidade de memória e de processamento, além de um consumo muito baixo de energia, não superando 300mA de corrente operando em uma tensão de 3.3 volts.

O projeto desenvolvido envolve duas possíveis soluções que operam de maneiras semelhantes, conforme será explicado a seguir. Inicialmente, os microcontroladores acessam a rede, conectando-se através da antena Wi-Fi presente em seu chip; após conseguirem o acesso a *Internet*, tornam-se clientes do *Broker MQTT* hospedado no *Cluster Andromeda*; em seguida, esses microcontroladores se inscrevem em tópicos que correspondem a sua localização, sendo assim possível o controle destes através da publicação de estruturas JavaScript Object Notation (JSON) específicas em seus tópicos no *Broker MQTT* em que estão conectados; após uma possível mensagem chegar ao microcontrolador, através do protocolo *MQTT*, este publicará uma mensagem em um tópico específico, chamado "response", confirmando a ação executada e no caso da solução que envolve o ESP32 esta mensagem contará com o nível de luminosidade que a lâmpada ligada apresenta.

A [Figura 9](#) ilustra o protótipo desenvolvido usando o microcontrolador ESP32 e o sensor de luminância. O dispositivo é posicionado entre o bocal e a lâmpada, sendo alimentado pelo bocal de 127V e alimentando a lâmpada com um bocal que compõe o dispositivo. No seu interior há um circuito, cujo esquemático que será demonstrado na [subseção 4.3.1](#).

A [Figura 10](#), por sua vez, apresenta a solução que envolve o microcontrolador ESP8266-

Figura 9 – Foto do dispositivo elaborado.



Fonte:Elaborado pelo autor.

01 o qual, mesmo em conjunto com o módulo relé, conta com pequenas dimensões, para que assim possa ser alocado no interior de um interruptor comum, fazendo o controle da rede elétrica com o relé e aproveitando do botão do interruptor para o controle manual, opção que não é possível no dispositivo que envolve o ESP32. Todavia, este dispositivo necessita de uma fonte de alimentação externa cuja tensão pode variar de 3.3 a 15 volts. Em utilização o ESP01 é inserido dentro do interruptor que é fechado e não apresenta qualquer diferença estética.

Figura 10 – Foto do ESP01 com as ligações elétricas feitas.



Fonte:Elaborado pelo autor.

Inicialmente o trabalho envolvia apenas uma das duas soluções, aquela que utiliza o microcontrolador ESP01 embutido no interruptor, entretanto houve a necessidade da anexação de um sensor para realizar a medição da luminosidade emitida pela lâmpada controlada. Essa necessidade surgiu do desenvolvimento de um artigo por parte da doutoranda Beatriz Campos Fialho, do Instituto de Arquitetura e Urbanismo (IAU). Seu artigo aborda a integração dos sistemas *IoT* com a técnica de desenvolvimento BIM (do inglês Building Information Modeling (BIM)). Com o acréscimo desse sensor, o protótipo passou a fornecer dados e fazer parte de um modelo 3D, desenvolvido no *software* Revit, baseado no ambiente em que estava inserido. Esse modelo 3D foi programado de modo a emitir alertas para manutenção ao passo do surgimento de anormalidades no dado fornecido pelo microcontrolador.

Em seu trabalho (FIALHO B. C.; CODINHOTO, 2021), a Beatriz Campos Fialho busca a modernização e melhoria na qualidade da infraestrutura fornecida por universidades, já que estas contam com um grande número de dispositivos de iluminação, cuja manutenção ocorre, na maioria dos casos, no momento em que os itens já não apresentam mais funcionamento, prejudicando as atividades que os envolvam. Com a solução desenvolvida surge a possibilidade de uma manutenção preventiva e sugerida de maneira automatizada, já que, atualmente, essa manutenção depende do apontamento do defeito por parte dos usuários ou funcionários responsáveis.

O artigo em questão foi publicado na revista *Buildings* que é tradicional no mercado imobiliário corporativo brasileiro, realizando publicações a mais de 12 anos no setor.

4.3 Atividades Realizadas

Esta seção apresenta o desenvolvimento do código da aplicação e do ambiente em nuvem necessários para que os dispositivos funcionem. Há também a apresentação da metodologia utilizada para a avaliação da eficiência dos dispositivos. O desenvolvimento do dispositivo foi feito usando a metodologia ágil, visto que haviam reuniões semanais, nas quais eram definidas metas a serem realizadas até a próxima interação e nessas reuniões eram apresentados os resultados obtidos até então.

4.3.1 Implementação

O código presente na memória dos microcontroladores, localizado no seguinte repositório do [GitLab](#), foi desenvolvido com o uso da linguagem de programação C++, fazendo uso também de bibliotecas prontas. O código inicialmente checa se há alguma rede *Wi-fi* cujo nome e senha estão em sua memória; se houver ele se conecta, caso contrário o microcontrolador se torna um *Access Point* fornecendo uma rede *Wi-fi* para conexão, nessa rede há um endereço IP específico, determinado pelo programador e que por padrão é 192.168.4.1. Nesse endereço há uma página *web* que lista as redes *Wi-fi* no alcance e permite a inserção e cadastro de senhas, após esse

cadastro o micro é reiniciado. As funcionalidades anteriormente descritas são implementadas pela biblioteca *WifiManager*¹.

Após conectado à rede e com acesso à *Internet* o microcontrolador atualiza o seu relógio interno, utilizando o servidor brasileiro de *Network Time Protocol* (NTP) e, em seguida, se conecta ao *Broker MQTT* Aedes, hospedado no *Cluster - Andromeda* e faz a inscrição nos tópicos que correspondem a sua localização. Todas as funções relacionadas ao *MQTT* são implementadas pela biblioteca *PubSubClient*². A partir desse momento, os dispositivos já estão funcionando aguardando apenas que uma mensagem seja publicada no tópico que o representa. Para a manipulação dos objetos *JSON* dentro do código é utilizada a biblioteca *ArduinoJSON*³.

O conteúdo da mensagem é enviado via *JSON* para que seja melhor manipulada por interfaces que possam ser desenvolvidas. Ademais, a arquitetura de chave-valor do *JSON* torna fácil e organizada a manipulação de seu conteúdo. Os campos de chave dos *JSON* foram convertidos para números, para que assim houvesse economia de tamanho da mensagem, economizando armazenamento, custo de processamento e, consequentemente, energia no microprocessador.

Após o recebimento da mensagem através do *Broker MQTT*, o microcontrolador publica uma resposta de confirmação da ação executada, contendo o ID da mensagem, para que uma possível aplicação atualize o estado da lâmpada em questão. A versão que possui o ESP32 conta com um sensor de luminância, que ao acender a lâmpada realiza uma medição e envia em um dos campos do *JSON* de resposta o seu valor. A [Figura 11](#) ilustra o funcionamento do sistema, assim como a estrutura de tópicos.

Os tópicos podem ser divididos em três partes, a primeira corresponde ao ID do ambiente em que o dispositivo se encontra; a segunda descreve qual tipo de dispositivo será controlado, no caso desta monografia, esse campo sempre será preenchido com "light"; já a terceira parte diz respeito ao ID do dispositivo. Esses campos são divididos usando o caractere "/". Como pode ser visto no diagrama representado na [Figura 11](#), os dispositivos exemplificados possuem os tópicos: "1/light/1" e "1/light/2", em que o primeiro número 1 presente simboliza o ID do ambiente, já os números 1 e 2 representam o ID de cada um dos dispositivos. O segundo campo foi feito pensando em projetos futuros, em que mais tipos de dispositivos possam vir a ser controlados via rede usando o mesmo protocolo.

4.3.2 *Broker MQTT*

O *Broker MQTT* foi implementado usando a plataforma Node-RED, o *broker* selecionado foi o Aedes⁴. O Node-RED por sua vez foi executado em uma das máquinas virtuais do *Cluster Andromeda* que se encontra no laboratório 1008 do ICMC.

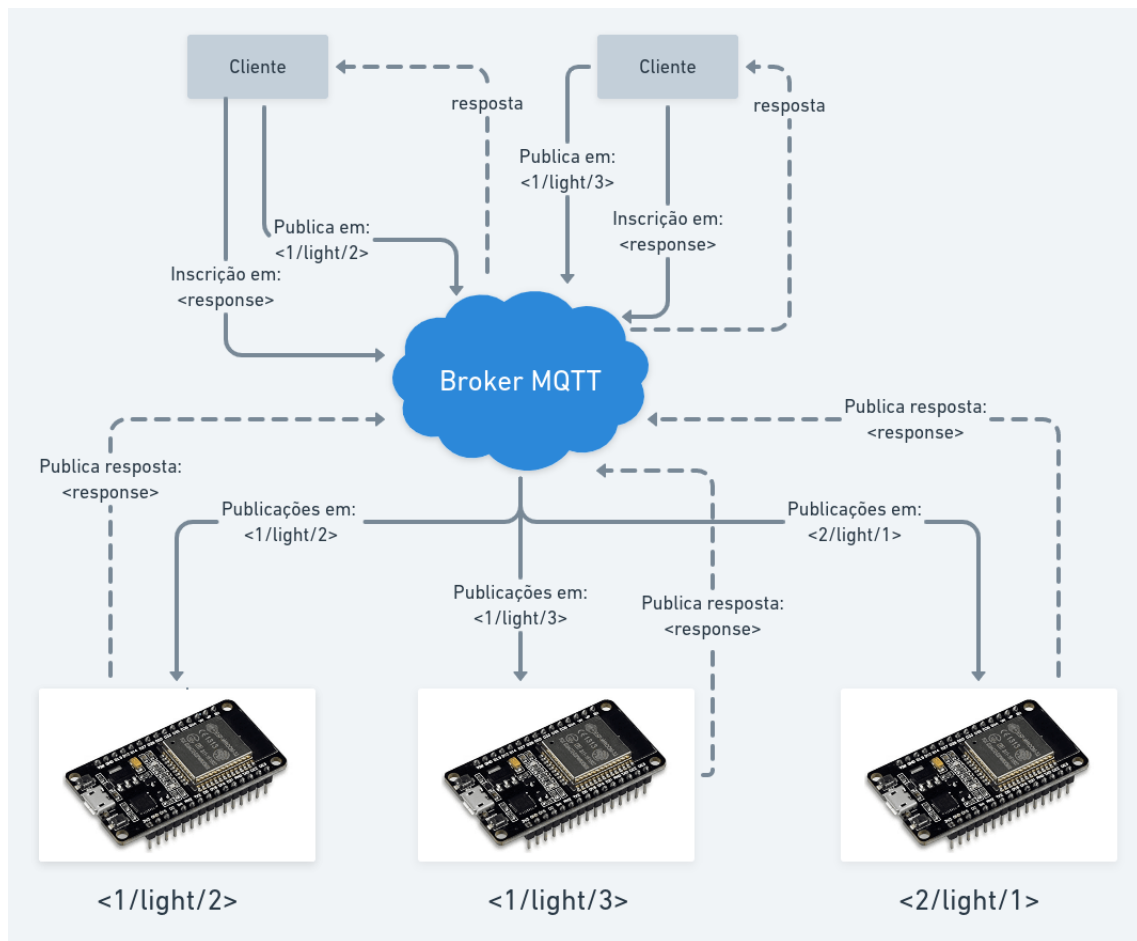
¹ [WiFiManager](#)

² [PubSubClient](#)

³ [ArduinoJSON](#)

⁴ [Aedes Broker MQTT](#)

Figura 11 – Diagrama do funcionamento do sistema.



Fonte:Elaborado pelo autor.

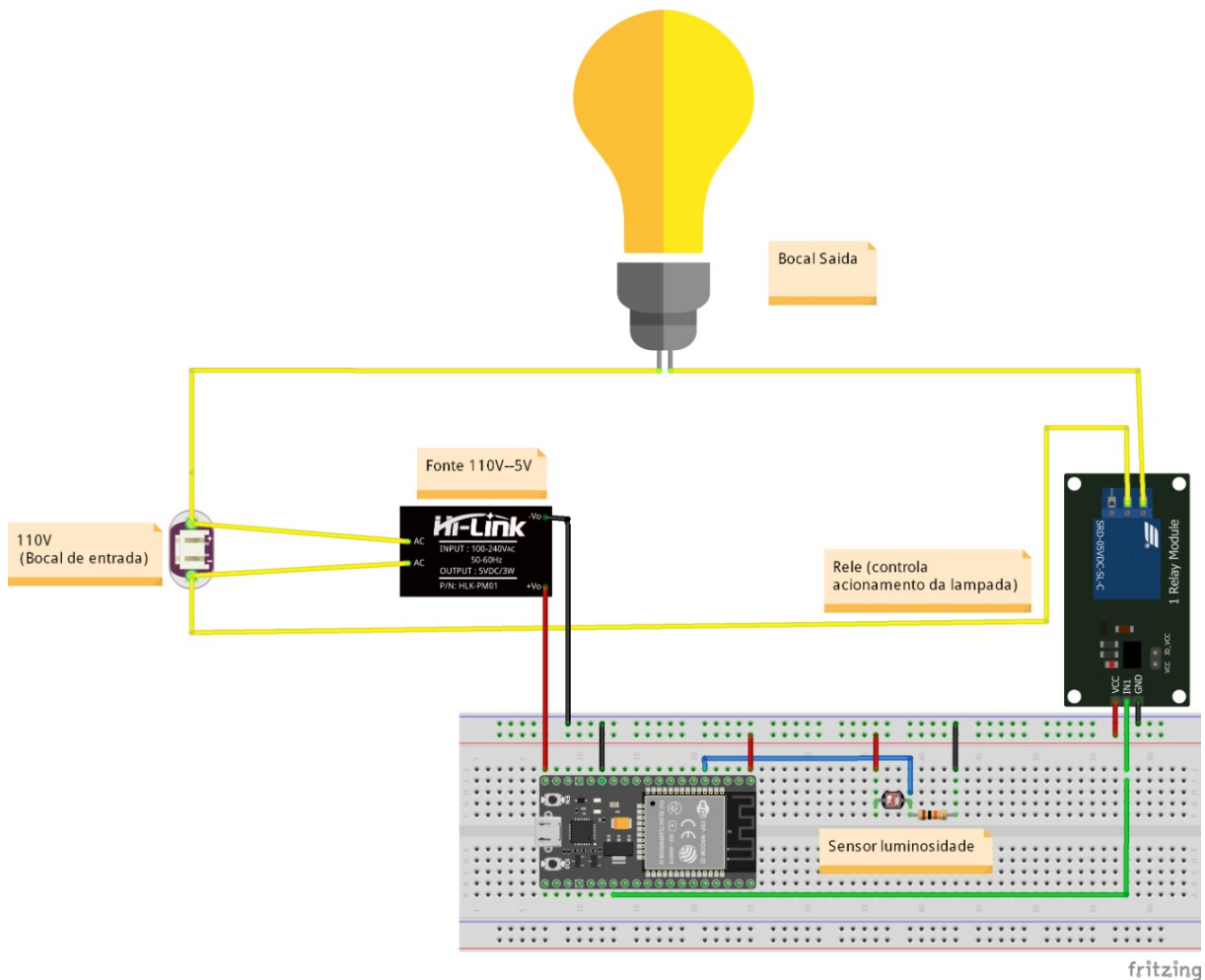
4.3.3 Montagem do dispositivo físico

Como pode ser visto na [Figura 9](#), o dispositivo conta com uma estrutura plástica. Em seu interior estão presentes o microcontrolador, uma fonte de alimentação de 5 Volts, um relé e na parte externa à carcaça plástica há um sensor envolvido por uma tira termo retrátil. Esse sensor é um *Light Dependent Sensor* (LDR), que nada mais é do que uma resistência cujo valor varia conforme a quantidade de luz incidente. O esquemático representado pela figura [Figura 12](#) ilustra as ligações elétricas do sistema. Essas ligações estão presentes no interior da estrutura e para que o dispositivo comece o seu funcionamento, basta conectar a lâmpada ao seu bocal e então ligá-lo no bocal da rede elétrica.

4.3.4 Testes executados

Os testes realizados no dispositivo são divididos em três, e eles envolvem dois tipos de testes diferentes, o primeiro e terceiro são testes funcionais, ou seja, buscam explorar e atestar o funcionamento de recursos do software, já o segundo teste é um teste de performance, avaliando o tempo necessário para o dispositivo estar pronto para uso. As definições dos testes foram

Figura 12 – Esquemático do dispositivo que envolve o ESP32.



Fonte:Elaborado pelo autor.

retiradas de (CATARINA, 2021).

Os testes foram feitos para avaliar o funcionamento do dispositivo e como o sistema reage frente às adversidades cotidianas. O primeiro teste envolve o envio de mensagens espaçadas em 2 segundos por 5 minutos, resultando em 150 mensagens. A mensagem enviada será comparada com a resposta dada pelo dispositivo no tópico "response", além de ser observada visualmente. Esse teste será realizado com auxílio da plataforma Node-RED, a qual enviará as mensagens variando aleatoriamente o campo responsável pelo estado da lâmpada no *JSON* enviado, assim como varia o campo referente ao ID da mensagem. Conforme for enviando as mensagens, o Node-RED receberá as mensagens de confirmação, as quais contam com o ID da mensagem original e também com o estado que a lâmpada foi setada, além de outras informações. Com isso, é possível comparar a mensagem enviada com a recebida e, caso haja disparidade dentre elas, uma mensagem será exibida na tela.

O segundo teste busca simular uma queda de energia no ambiente, em que o dispositivo

para de ser alimentado. Em seguida, a energia será reestabelecida e o dispositivo deverá estar pronto para uso, medindo também o tempo necessário para isso. Esse teste também será realizado com o auxílio do Node-RED, o qual enviará uma mensagem ao microcontrolador forçando o seu *reboot*, no momento em que essa mensagem for enviada um contador será iniciado e quando o dispositivo publicar no *broker* a mensagem que indica sua reconexão, o contador será registrado. Com esse registro é possível traçar um gráfico para uma posterior análise.

Por fim, será realizado um terceiro teste em que nome e senha da rede em que o dispositivo se encontra conectado serão alterados, a fim de ser testado e avaliado o serviço descrito na [subseção 4.3.1](#) de busca por novas redes.

O primeiro e segundo testes foram feitos com o auxílio do Node-RED, como já foi citado, esses testes podem ser executados em qualquer instância da plataforma, bastando apenas realizar a importação do arquivo que contém os nós usados. Esse arquivo encontra-se no seguinte [link](#).

4.4 Considerações Finais

Este capítulo descreve os detalhes do projeto além de sua implementação, do esquemático com as ligações do microcontrolador e os testes a serem executados, em seguida. No [Capítulo 5](#) serão exibidos os resultados obtidos por meio dos testes.

RESULTADOS

5.1 Considerações Iniciais

Este capítulo tem o intuito de apresentar os resultados obtidos nos testes descritos na [subseção 4.3.4](#).

5.2 Primeiro Experimento

Este experimento diz respeito à funcionalidade do dispositivo. Os resultados obtidos foram plotados em um gráfico de linha, em que uma das linhas representa o estado enviado na mensagem, representada na cor azul, e a outra, o estado para o qual a lâmpada foi setada, estado este obtido através da resposta enviada pelo microcontrolador, em cor rosa. Sendo o eixo Y a representação dos valores de estado da lâmpada e o eixo X o horário em que os comandos foram recebidos.

Figura 13 – Gráfico de comparação entre o estado enviado e recebido como resposta.



Fonte:Elaborado pelo autor.

O gráfico exibido na [Figura 13](#) possui todos os seus pontos em apenas 2 valores do eixo Y, 0 que representa a lâmpada desligada e 1 que representa a lâmpada ligada. Como pode ser

visto na legenda, os valores enviados na mensagem estão representados pela linha de cor azul, já a linha dos estados recebidos na resposta do microcontrolador possui a cor rosa. Analisando pode ser notado que ambas as linhas estão sobrepostas nos 150 pontos do gráfico, isso significa que o microcontrolador cumpriu, com exatidão, os comandos enviados.

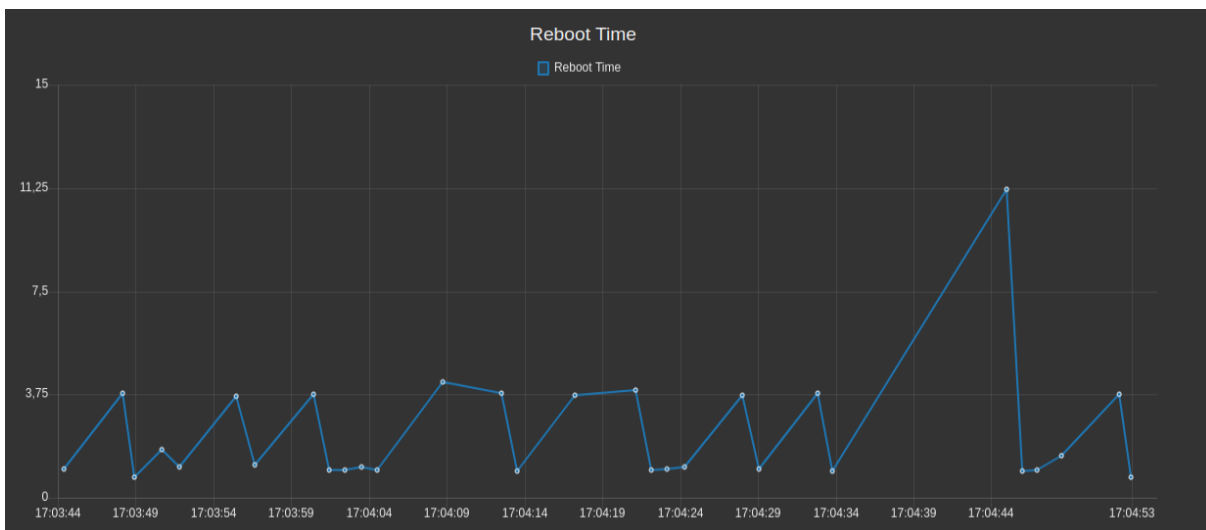
Pode ser acrescentado ainda que em todos os casos em que o comando enviado foi para que a lâmpada fosse ligada, a resposta recebida contava com o valor máximo do sensor de luminosidade, constatando, portanto, um bom funcionamento da lâmpada utilizada. Esse valor poderia ser usado em um sistema de manutenção de ambientes, visto que, quando a medição do sensor não atingisse o valor máximo na resposta, um alerta para manutenção seria emitido, pois, provavelmente, a lâmpada do dispositivo estaria com problemas de funcionamento.

5.3 Segundo Experimento

O segundo experimento busca analisar o tempo necessário para reconexão do dispositivo ao *Broker*, frente a uma queda na alimentação. A variável analisada é o tempo de reconexão ao *Broker*, pois, após realizar essa conexão, já está operando normalmente.

Foi montado um gráfico contendo 30 pontos, os quais representam o tempo necessário para a reconexão, o processo dessa montagem foi descrito na [subseção 4.3.4](#).

Figura 14 – Gráfico do tempo de reinicialização do dispositivo.



Fonte:Elaborado pelo autor.

Como pode ser visto na [Figura 14](#), em todos os casos o tempo de reconexão é muito baixo, não ultrapassando os 11.25 segundos, o que comprova a eficiência do dispositivo frente à possíveis queda de energia. Vale ainda ressaltar que, em quase todos os casos, o tempo de reconexão e disponibilidade para o uso não ultrapassou os 4 segundos.

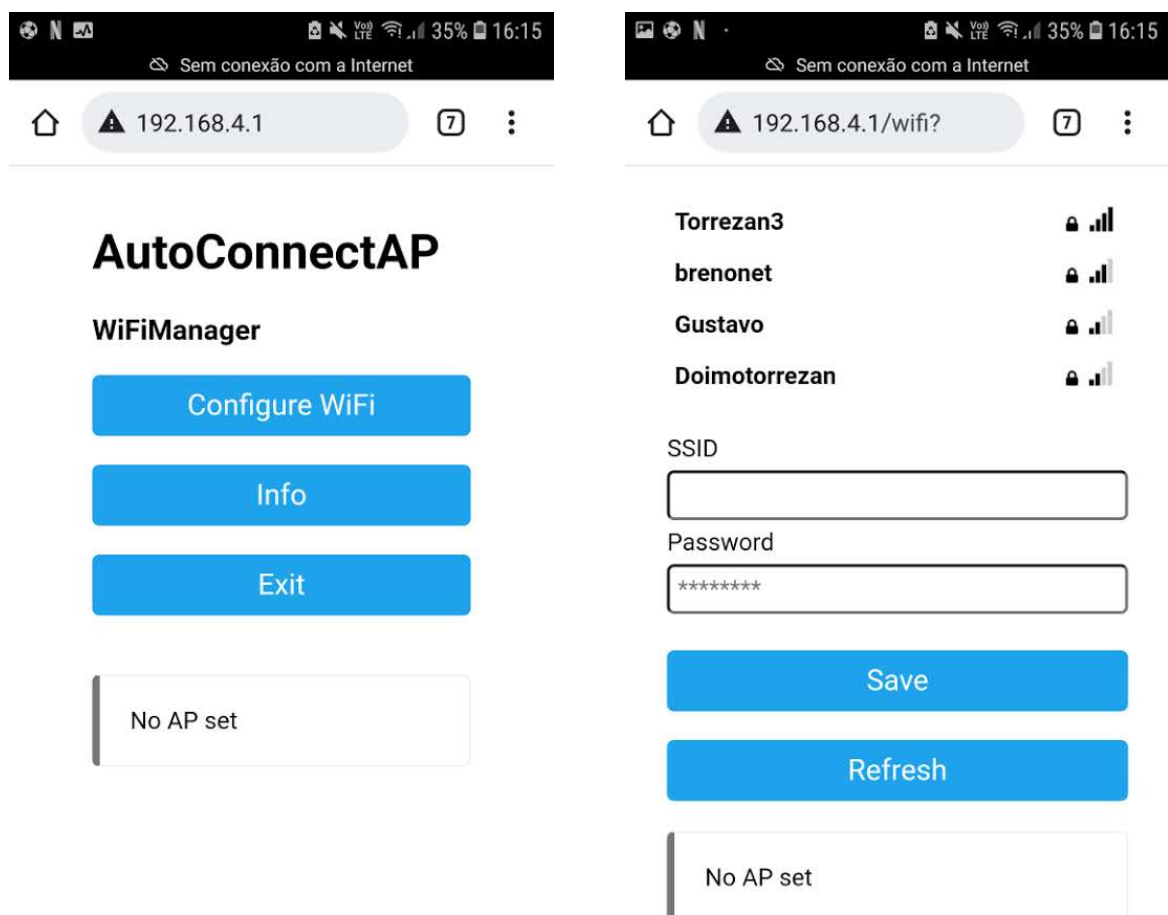
5.4 Terceiro Experimento

O terceiro experimento foi feito no intuito de testar a capacidade de busca por novas redes e de reconexão quando não houver redes conhecidas no alcance do microcontrolador.

Como explicado na [subseção 4.3.1](#) o microcontrolador, ao não encontrar redes Wi-Fi conhecidas no seu alcance, torna-se um *Acces Point* e cria uma página *web* que exibe todas as redes em seu alcance, possibilitando então ao usuário cadastrar novas redes em sua memória.

A [Figura 15](#) demonstra a tela da página *web* exibida ao se conectar à rede fornecida pelo microcontrolador. O usuário pode selecionar uma rede e inserir a senha, com isso, o microcontrolador salva nome e senha em sua memória e reinicia. Após inicializado novamente, ele se conectará à nova rede cadastrada.

Figura 15 – Telas de conexão a novas redes.



(a) Tela inicial da página web fornecida pelo *Acces Point* do microcontrolador

(b) Tela com as redes no alcance do microcontrolador.

Fonte: Elaborado pelo autor

5.5 Publicações

Este trabalho foi utilizado em um estudo de uma tese de doutorado da aluna Beatriz Campos Fialho do IAU - São Carlos, intitulado: **Smart Universities's Estates: BIM and IoT supporting Reactive Maintenance**. Esta interação resultou na submissão de um artigo para uma revista internacional, como descrito a seguir:

FIALHO, B. C.; CODINHOTO, R.; FABRICIO, M, M.; RIBEIRO, C. M. N.; TORREZAN, J. P. D.; BUENO, J. M. S.; ESTRELLA, J. C. . Development of a BIM and IoT-based Smart Lighting Maintenance System prototype for universities FM sector. In: Buildings, 2021 - MDPI - ISSN: 2075-5309; **(Submetido)**.

5.6 Considerações Finais

Neste capítulo observam-se os resultados dos testes executados e em todos, o dispositivo obteve boa resposta, mostrando-se eficiente em relação às adversidades cotidianas, além de confiável conforme os comandos enviados.

CONCLUSÃO

6.1 Contribuições

Como foi demonstrado nos testes, além da utilização no dia a dia, o sistema de controle de iluminação é plenamente capaz de realizar a função proposta, sendo uma solução de baixo custo, com grande eficiência e também de baixo consumo energético. Ainda assim, é observável que o dispositivo proposto é apenas um dos componentes de um possível sistema de gerenciamento de ambientes e demanda também, de uma interface para que possa haver o controle.

6.2 Trabalhos Futuros

Dentre os trabalhos futuros, aponta-se:

- Adição de sensoramento detector de presença no ambiente, com o intuito de permitir a gestão conforme mais esse parâmetro.
- Construção de um sistema robusto para gerência e manutenção de ambientes, tendo em vista a escalabilidade do projeto.
- Adição de sensoramento na passagem de corrente elétrica no dispositivo, para que assim possa ser feito o cálculo do consumo real de energia gasto pela iluminação do ambiente.
- Confecção de um modelo 3D para impressão, a fim de tornar o dispositivo algo mais profissional e robusto.
- Acréscimo de um sensor e de um emissor infravermelho no projeto que envolve o microcontrolador ESP32, com o objetivo de controlar a iluminação através de um botão físico.

REFERÊNCIAS

ALSUHLI, G.; KHATTAB, A. A fog-based iot platform for smart buildings. In: **2019 International Conference on Innovative Trends in Computer Engineering (ITCE)**. [S.l.: s.n.], 2019. p. 174–179. Citado na página 30.

CATARINA, R. **Conheça os 10 principais tipos de teste de software**. 2021. Disponível em: <<https://targettrust.com.br/blog/10-principais-tipos-de-teste-de-software/>>. Acesso em: 13/12/2021. Citado na página 38.

DOUKAS, C.; ANTONELLI, F. Compose: Building smart amp; context-aware mobile applications utilizing iot technologies. In: **Global Information Infrastructure Symposium - GIIS 2013**. [S.l.: s.n.], 2013. p. 1–6. ISSN 2150-329X. Citado na página 30.

ESP32. 2021. Disponível em: <<https://www.espressif.com/en/products/socs/esp32>>. Acesso em: 10/12/2021. Citado na página 23.

ESP8266. 2021. Disponível em: <<https://www.espressif.com/en/products/socs/esp8266>>. Acesso em: 10/12/2021. Citado 2 vezes nas páginas 21 e 22.

FIALHO B. C.; CODINHOTO, R. F. M. M. R. C. M. N. T. J. P. D. B. J. M. S. E. J. C. Development of a bim and iot-based smart lighting maintenance system prototype for universities fm sector. **Buildings**, 2021. ISSN 2075-5309. Citado na página 35.

GILLIS, A. S. What is internet of things (iot)? In: . [s.n.], 2021. Disponível em: <<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>>. Acesso em: 13/12/2021. Citado na página 21.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, v. 29, n. 7, p. 1645 – 1660, 2013. ISSN 0167-739X. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167739X13000241>>. Citado na página 21.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. Mqtt-s — a publish/subscribe protocol for wireless sensor networks. In: **2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)**. [S.l.: s.n.], 2008. p. 791–798. Citado na página 26.

LEKIĆ, M.; GARDAŠEVIĆ, G. Iot sensor integration to node-red platform. In: **2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)**. [S.l.: s.n.], 2018. p. 1–5. Citado na página 27.

LI, C.; LOGENTHIRAN, T.; WOO, W. L. Development of mobile application for smart home energy management: ishome. In: **2016 IEEE 6th International Conference on Power Systems (ICPS)**. [S.l.: s.n.], 2016. p. 1–6. Citado na página 30.

LOBACHEV, I.; CRETU, E. Smart sensor network for smart buildings. p. 1–7, Oct 2016. Citado na página 29.

NODE-RED. 2021. Disponível em: <<https://nodered.org/>>. Acesso em: 10/12/2021. Citado na página 27.

RAMPRASAD, B.; MCARTHUR, J.; FOKAEFS, M.; BARNA, C.; DAMM, M.; LITOIU, M. Leveraging existing sensor networks as iot devices for smart buildings. p. 452–457, Feb 2018. Citado na página 29.

W. ALEXIS Z., G. L. P. H. S. L. M. F. M.; M., C. **US Connected Home Market**. 2019. Disponível em: <<https://askwonder.com/research/us-connected-home-market-4qifpd55k>>. Acesso em: 27/12/2021. Citado 2 vezes nas páginas 9 e 11.