

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
BACHARELADO EM ENGENHARIA ELETRÔNICA

HORION SILVA DREHER

**AUTOMAÇÃO RESIDENCIAL UTILIZANDO O PROTOCOLO MQTT
CONTROLADO POR APP ANDROID/IOS**

Campo Mourão

2019

HORION SILVA DREHER

**AUTOMAÇÃO RESIDENCIAL UTILIZANDO O PROTOCOLO MQTT
CONTROLADO POR APP ANDROID/IOS**

Trabalho de Conclusão de curso, apresentado à disciplina de Trabalho de Conclusão de Curso, do curso Superior de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica - DAELN - da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Engenheiro em Eletrônica.

Orientador: Eduardo Giometti Bertogna

Campo Mourão

2019



TERMO DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO INTITULADO
AUTOMAÇÃO RESIDENCIAL UTILIZANDO O PROTOCOLO MQTT CONTROLADO POR APP ANDROID/IOS

DO(A) DISCENTE

HORION SILVA DREHER

Trabalho de Conclusão de Curso apresentado no dia 26 de novembro de 2019 ao Curso Superior de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, Campus Campo Mourão. O(A) discente foi arguido(a) pela Comissão Examinadora composta pelos professores abaixo assinados. Após deliberação, a comissão considerou o trabalho aprovado (aprovado, aprovado com alterações ou reprovado).

Prof. Marcio Rodrigues da Cunha
Avaliador(a) 1
UTFPR

Prof. Leandro Castilho Brolin
Avaliador(a) 2
UTFPR

Prof. Eduardo Giometti Bertogna
Orientador(a)
UTFPR

RESUMO

Há um crescimento constante no mercado econômico de dispositivos conectados à internet, trazendo grandes vantagens no dia a dia. Assim, muitas pessoas já tem acesso à automação residencial, mas este acesso ainda pode ser muito caro e os projetos de automação residencial podem ser muito complexos. O objetivo deste trabalho é desenvolver dispositivos para automação residencial de baixo custo e instalação fácil. Com isso, mais pessoas terão acesso à dispositivos inteligentes. Os resultados obtidos foram satisfatórios, com grande parte dos dispositivos apresentando o funcionamento correto, se comunicando com eficiência com o servidor e o aplicativo desenvolvido.

Palavras-chave: Automação Residencial. IoT. Dispositivos Inteligentes.

ABSTRACT

There is a constant economic market growth of internet connected devices, bringing great advantages on daily basis. Therefore, many people already have access to home automation, but this access still can be expensive and the projects of home automation can be very complex. The objective of this assignment is to develop low cost and easy setup home automation devices. Thereby, more people will have access to smart devices. The obtained results were satisfactory, with the correct working of most devices, from which, showed efficiency in the communication between the server and the developed application.

Keywords: Home Automation. IoT. Smart Devices

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Objetivo Geral	8
1.2	Objetivos Específicos	8
1.3	Justificativa	9
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Automação Residencial	11
2.1.1	Aplicações	11
2.2	Instalações Prediais	12
2.2.1	Redes de distribuição e ligações	12
2.2.2	Potência	12
2.3	Redes de Computadores	13
2.3.1	Redes Locais - Local Area Networks (LAN)	14
2.3.2	Protocolos	14
2.3.3	Camadas de protocolo e o modelo TCP/IP	14
2.3.4	<i>Message Queue Telemetry Transport</i> (MQTT)	15
2.4	Microcontroladores	16
2.4.1	ESP8266	17
2.5	Banco de dados	17
3	METODOLOGIA	20
3.1	Projeto do <i>Hardware</i>	20
3.1.1	Microcontrolador	20
3.1.2	Interruptor	21
3.1.3	Tomada	23
3.1.4	Tranca e Termostato	26
3.1.5	Fonte Chaveada	28
3.2	Projeto dos <i>Softwares</i>	30
3.2.1	Servidor	30
3.2.1.1	Banco de Dados	31
3.2.1.2	API	32
3.2.2	<i>Firmware</i>	32
3.2.3	Aplicativo	33
4	RESULTADOS	35
4.1	API e Banco de Dados	35

4.2	<i>Hardware</i>	38
4.2.1	Fonte Chaveada	38
4.2.2	Microcontroladores	38
4.2.3	Interruptor	40
4.2.4	Tomada	42
4.2.5	Tranca	42
5	CONCLUSÃO	44
	REFERÊNCIAS	46

1 INTRODUÇÃO

Nos primeiros anos do século XX, Henry Ford introduziu as linhas de produção nas indústrias, que foram pensadas para trazer eficiência e alta produtividade no trabalho, baseando a produção em movimentos básicos, divididos em várias etapas. Estes movimentos poderiam ser reproduzidos por máquinas e robôs por serem repetitivos e também por aumentar a precisão e consistência da linha de produção (RODIC, 2009).

Em 1901, esta automação também foi levada para as residências com o primeiro aspirador de pó, facilitando a limpeza. Nas próximas décadas, começaram a surgir máquinas de lavar roupa, secadores de roupa, ferros de passar, refrigeradores, e muitos outros eletrodomésticos que são comuns em nossa vida atualmente (HENDRICKS, 2014).

Já na década de 1990, vários fabricantes estavam conectando dispositivos à internet, com servidores web adicionados à produtos embarcados na indústria. Nesta época, surgiam as aplicações como sistemas operacionais e linguagens de programação de mais alto nível, o que aumentavam as possibilidades. Assim, foi possível que consumidores pudessem ter acesso à dispositivos domésticos conectados à rede, como termostatos, medidores de energia e sistemas de irrigação. Com isto, estes dispositivos simples passaram a se tornar inteligentes, facilitando a forma com que as pessoas realizam as tarefas diárias e otimizando o seu tempo (CHASE, 2013).

Hoje em dia, ter casas inteligentes está se tornando mais comum, atingindo um número maior de pessoas, não apenas as mais ricas. Estas casas podem ser monitoradas e controladas por qualquer *smartphone*, enviando mensagens ou comandos de fora de casa para acionar a iluminação ou controlar a temperatura. A televisão não serve apenas para assistir canais de TV, agora pode ser usada para acessar a internet, assim como um celular ou computador. (MACDONALD, 2007)

É assim que surge o termo *Internet of Things* (Internet das Coisas - IoT). Segundo uma pesquisa realizada por HP (2017), 67% das pessoas definem a Internet das Coisas como objetos do dia-a-dia que estão conectados com a internet. A pesquisa também diz que a Internet das Coisas está sendo cada vez mais usadas em casas e indústrias: 62% das indústrias já adotam dispositivos inteligentes e 83% dizem que obtiveram um aumento significativo na eficiência dos negócios.

Para o seu funcionamento, estes sistemas de casas inteligentes utilizam diversos meios e protocolos de comunicação. Pode ser citado o MQTT (*Message Queue Telemetry Transport*), um protocolo de mensagens leve, muito utilizado para sensores de baixa potência (YUAN, 2017). Também existem os padrões de comunicação ZigBee e Z-Wave, que são de curta distância e baixa potência que precisam de uma central de comando para funcionar, chamada de *smart home hub* e dificilmente são controlados pelo *smartphone*, descartando a necessidade de utilizarem o

Wi-Fi (ROUSE, 2018).

Os sistemas mais novos estão utilizando Inteligência Artificial e *Machine Learning* e estão crescendo muito em dispositivos para casas inteligentes, permitindo que o sistema se adapte de acordo com a rotina das pessoas e com o espaço em que está incluído. Amazon Echo e o Google Home são exemplos de sistemas que utilizam Inteligência Artificial (ROUSE, 2018).

Previsões mostram que até 2020, 50 bilhões de dispositivos serão conectados e, em um futuro próximo, as redes serão conectadas por trilhões de nós. Mas estes números somente serão atingidos simplificando-se o meio através do qual estes dispositivos são conectados e suas comunicações (CHASE, 2013).

1.1 Objetivo Geral

Desenvolver um sistema para automação residencial de baixo custo com fácil instalação e manuseio (*Plug and Play*), utilizando o protocolo MQTT para comunicação e uma interação homem-máquina por aplicativos em *smartphones* com os sistemas operacionais Android ou iOS.

1.2 Objetivos Específicos

Aqui são descritos todas os objetivos necessários para o funcionamento completo do sistema, divididos em:

- Confeccionar placas de circuito impresso para:
 - Fonte chaveada
 - Interruptor inteligente
 - Tomada inteligente
 - Termostato inteligente
 - Tranca inteligente
- Desenvolver um servidor utilizando o *Raspberry Pi* contendo:
 - Protocolo HTTP (*Hypertext Transfer Protocol*)
 - Banco de Dados *MongoDB*
 - Broker MQTT
- Desenvolver Firmware dedicado para dispositivos utilizando o microcontrolador ESP8266X
- Desenvolver Aplicativo para sistemas Android e iOS

1.3 Justificativa

Segundo a AURESIDE (2013) (Associação Brasileira de Automação Residencial e Predial), o Brasil possuía, em 2013, um potencial para que 1,8 milhões de residências se tornem inteligentes, com uma expectativa de valor de mercado de US\$ 59,68 bilhões em 2020. Porém, somente 300 mil residências possuem um sistema de automação residencial atualmente (INFOMONEY, 2018).

O maior problema em sistemas de automação é o custo para instalações em residências. Normalmente é preciso refazer a instalação elétrica da casa para adaptá-la. Logo, o custo e complexidade para implementação se tornam altos. Instalações básicas, envolvendo luzes, ar condicionado, cortinas, entre outros, podem custar cerca de R\$ 15 mil. Além disso, caso ocorra algum problema no equipamento, o usuário precisa esperar o atendimento técnico da empresa (COSTA, 2017).

Isto pode ser contornado utilizando dispositivos que se comunicam por redes sem fio, e também fazendo com que estes tenham uma instalação fácil, a qual o próprio usuário pode realizar. Este processo ainda pode ter custo alto dependendo da complexidade do projeto (NEOCONTROL, 2017).

Outro grande problema em automação residencial é segurança na rede. Vários dispositivos mostram-se vulneráveis à *hackers*. Dependendo do protocolo de redes usado, um invasor pode facilmente acessar todo o sistema por meio de suas vulnerabilidades. Após isso, os dispositivos podem ser controlados remotamente por invasores ou o usuário pode ter seus dados roubados (CONSTANTIN, 2013).

A pesquisa realizada por HP (2017) também apresenta os problemas na segurança: 84% dos entrevistados já experienciaram um ataque aos dispositivos. As invasões mais comuns foram causadas por *malware*, *spyware* ou falha humana. Isto acontece porque muitos desenvolvedores deixam portas de rede abertas para utilizar todos os benefícios do sistema.

Sabendo disso, adotar algumas medidas de segurança podem proteger o usuário. Exigir que o usuário utilize senha para o acesso do sistema e criptografar os dados transmitidos na rede são práticas que podem melhorar a segurança (CONSTANTIN, 2013).

Outra situação que HP (2017) apresenta é que 98% das organizações que adotaram IoT conseguem analisar os dados dos dispositivos, mas a maioria não conseguem criar um valor útil destes dados. Muitos não estão extraindo e analisando os dados para gerar melhores decisões para o negócio.

Com isso, é visto que o custo alto, instalações complexas e falta de segurança digital são problemas na automação residencial atualmente. Este trabalho pode trazer instalações mais simples, custos mais baixos, devido a maior simplicidade no projeto, e um aumento na segurança, focando em criptografia e maior controle na autenticação de usuários.

Desta forma o trabalho é justificável para o mercado, introduzindo dispositivos com um diferencial significativo em relação aos que já existem no mercado com o baixo custo proposto, que não apresentam grande qualidade ou boa segurança digital.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos de automação residencial e a forma como funciona, descrevendo fundamentos sobre instalações prediais e potência em corrente alternada, redes de computadores e seus protocolos, microcontroladores e banco de dados. Também será descrito o funcionamento básico do protocolo MQTT e conceitos sobre o microcontrolador ESP8266 e as plataformas Android e iOS.

2.1 Automação Residencial

O termo automação residencial pode ser definido como "um processo que, usando diferentes soluções e equipamentos, possibilita ao usuário usufruir o máximo de qualidade de vida na sua habitação" (BÓ; MURATORI, 2014, p. 15), e surgiu no Brasil a partir do *Home Automation*, após fabricantes americanos começarem a usar este sistema. Também pode ser chamado de Domótica (BÓ; MURATORI, 2014).

Os primeiros sistemas de automação residencial eram basicamente controles de serviços prediais como aquecimento, ventilação e ar-condicionados, com o objetivo de diminuir os gastos com energia e melhorar a qualidade de vida (SAUTER et al., 2011). Nos anos de 1960, já eram construídas algumas casas automatizadas por entusiastas, nos Estados Unidos, quando o custo deste sistema era muito caro e inviável. Em 1984, a Associação Americana de Construtores de Casas usam o termo "Casa Inteligente" pela primeira vez (HARPER, 2003).

2.1.1 Aplicações

De forma geral, a base de um projeto de automação residencial é formada por sensores e atuadores que enviam seus dados para controladores específicos, podendo ser usados juntamente com painéis para inserção Interface Humano-Máquina (IHM) na casa ou prédios. Entretanto, em sistemas tradicionais como este, haviam problemas com interoperabilidade, ou seja, camadas do sistema não se comunicavam de forma transparente com outras. Com a chegada da tecnologia Ethernet e protocolos como TCP/IP houve uma maior integração dessas camadas e, também, soluções e conceitos de alto nível (SAUTER et al., 2011).

Hoje em dia, com o crescimento de redes sem fio e dispositivos inteligentes, um projeto de automação residencial se tornou mais fácil e viável. *Smartphones* podem ser usados para controlar tomadas, luzes, temperatura e dispositivos de segurança, todos conectados em redes locais ou na internet. Em um exemplo de aplicação para automação residencial, um micro-servidor embarcado, dispositivos de controle e um smartphone, podem ser usados para controle e monitoração remoto. O micro-servidor recebe dados via internet e controla dispositivos físicos, interfaceados via atuadores e sensores (ASADULLAH; RAZA, 2016).

2.2 Instalações Prediais

A Norma Brasileira (NBR) 5410 foi elaborada no Comitê Brasileiro de Eletricidade (ABNT/CB-03), pela Comissão de Estudo de Instalações Elétricas de Baixa Tensão, e parte da Associação Brasileira de Normas Técnicas (ABNT). Esta NBR “estabelece as condições a que devem satisfazer as instalações elétricas de baixa tensão, a fim de garantir a segurança de pessoas e animais, o funcionamento adequado da instalação e a conservação dos bens” (ABNT, 2004, p. 1). Ela se aplica a tensões nominais menores ou iguais à 1000 V (Volts) em corrente alternada, com frequências menores que 400 Hz, ou 1500 V em corrente contínua. Sendo assim, a NBR 5410 se aplica diretamente às residências e comércio (ABNT, 2004).

2.2.1 Redes de distribuição e ligações

No Brasil, as tensões distribuídas nas cidades são de 127/220 V e 220 /380 V, para um sistema trifásico, e 220 V/127 V com fio neutro, para um sistema bifásico, todas em corrente alternada, com uma frequência de 60 Hz (Hertz), e que assim, fazem parte da NBR 5410. Estas chegam no comércio, residências ou indústrias por meio de transformadores abaixadores, que abaixam a tensão da linha de distribuição (entre 13,8 kV e 34,5 kV) para uma tensão que pode ter uma ligação monofásica (2 fios condutores), bifásica (3 fios condutores) ou trifásica (3 ou 4 fios condutores), dependendo do consumo de potência do consumidor (CREDER, 2016):

Tabela 1 – Ligações de acordo com a carga do consumidor.

Carga	Ligações
4 kW	Monofásica (2 condutores)
Entre 4 e 8 kW	Bifásica (3 condutores)
Maior que 8 kW	Trifásica (3 ou 4 condutores)

Fonte: Creder (2016).

2.2.2 Potência

Segundo Boylestad (2012), a potência média ou real é dada por:

$$P_m = V_{rms} I_{rms} \cos\theta \quad (2.1)$$

Da qual:

P_m - Potência Média

V_{rms} - Tensão eficaz

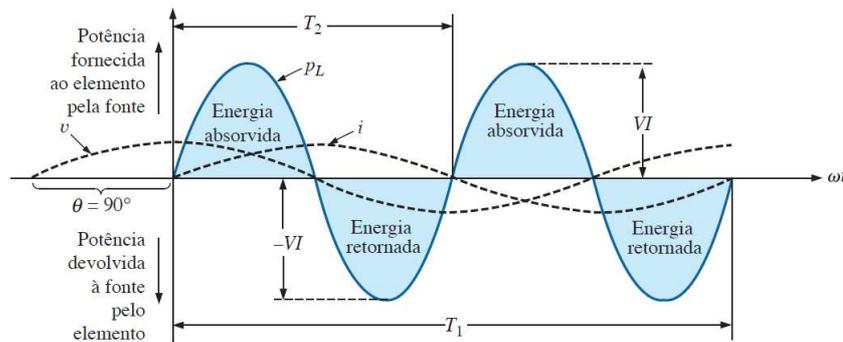
I_{rms} - Corrente eficaz

$\cos\theta_m$ - Fator de Potência

O fator de potência influencia diretamente na potência real. Uma carga totalmente resistiva possui $\theta = 0^\circ$ e é chamada de potência ativa. Uma carga totalmente indutiva ou capacitiva

possuem $\theta = 90^\circ$ e $\theta = -90^\circ$, respectivamente, e são chamadas de potências reativas. Esta diferença em θ leva à um adiantamento ou atraso na corrente. Ou seja, cargas totalmente indutivas ou capacitivas levam à uma potência média dissipada nula, o que pode ser visto na curva da Figura 1 (BOYLESTAD, 2012).

Figura 1 – Curva de Potência para uma carga indutiva



Fonte: Boylestad (2012).

Analisando a curva, é possível observar que, mesmo com a potência média sendo nula, ainda há potência sendo dissipada em uma parcela do tempo. Devido à isto, um Wattímetro pode não mostrar a potência real em um circuito indutivo ou capacitivo. Assim, vários equipamentos tem a potência especificada em Volt-Ampéres (VA), sendo esta, a potência aparente (BOYLESTAD, 2012).

Há regras que devem ser seguidas para instalações de pontos de iluminação e tomadas. Pontos de luz necessitam de uma potência mínima de 100 VA para dependências com área inferior ou igual à $6 m^2$ e adicionando 60 VA para cada aumento de $4 m^2$. Quanto às tomadas, é preciso de no mínimo 600 VA para até três pontos de tomada para áreas de serviço, banheiros, cozinhas, copas, lavanderias e locais relacionados, e 100 VA por pontos extras. Para os demais cômodos, no mínimo 100 VA por ponto de tomada (CAVALIN; CERVELIN, 2013).

2.3 Redes de Computadores

Uma rede de computadores pode ser definida por "um conjunto de computadores autônomos interconectados por uma única tecnologia" (TANEMBAUM, 2011, p. 2, **tradução nossa**). Segundo Tanenbaum (2011), esta conexão pode ser realizada ponto-a-ponto ou por transmissão.

Conexões ponto-a-ponto conectam duas máquinas. Para uma mensagem, chamada de pacote, ir de uma máquina para outra, pode ser preciso passar por máquinas entre elas. Numa rede por transmissão, o meio de comunicação é compartilhado por todas as máquinas, ou seja, todas as máquinas recebem o pacote transmitido. Assim, se um pacote é direcionado para apenas

uma máquina, as outras o ignoram. Uma rede sem fio, é um exemplo de rede por transmissão (TANEMBAUM, 2011).

2.3.1 Redes Locais - Local Area Networks (LAN)

Redes locais são redes privadas usadas para conectar computadores pessoais e outros dispositivos em pequenos locais, como uma casa ou um prédio. LANs com fio, normalmente conectadas com fios de cobre, possuem uma velocidade de comunicação de 100 Megabits por segundo (Mbps) até 1 Gigabit por segundo (Gbps), atrasos na escala de microssegundos e cometem poucos erros. O padrão utilizado nestas conexões é o IEEE 802.3, também conhecido como Ethernet. Os computadores se comunicam por meio de um dispositivo chamado *switch*, com uma comunicação ponto-a-ponto. Já as LANs sem fio, são conectadas via *Access Point* (AP) ou roteador sem fio, cuja conexão com os computadores é feita a partir de antenas instaladas neles. O padrão utilizado nestas redes é chamado de IEEE 802.11 (TANEMBAUM, 2011).

2.3.2 Protocolos

Em redes de computadores, protocolos são regras que determinam qual dado é transmitido, como é transmitido e quando é transmitido. Os termos usados para isto são chamados de sintaxe, semântica, e *timing*, e são explicados neste subcapítulo, segundo Forouzan (2010):

Sintaxe é a estrutura de um dado. Os dispositivos que recebem um dado precisam entender como ler um dado, para tal, precisam saber a quantidade e ordem dos bits na transmissão. Mesmo que dispositivos armazenem dados de maneira diferente, a estrutura do dado precisa ser a mesma quando estes se comunicam.

A maneira com que um dado é interpretado é chamada de semântica. Mesmo que o dado tenha sido recebido de forma correta, é necessário interpretá-lo. Assim, regras são necessárias para que um dado não seja usado de forma incorreta e fazer com que isso traga problemas para um sistema.

E o *timing* se refere à quando um dado deve ser enviado e a velocidade que pode ser usada na transmissão. Se isto for usado de maneira incorreta, como dois dispositivos usando velocidades diferentes, o dado pode ser perdido.

2.3.3 Camadas de protocolo e o modelo TCP/IP

Para organização de projetos de rede, utilizam-se protocolos em camadas. Estas camadas executam suas ações utilizando serviços fornecidos pela camada diretamente abaixo à ela. Esta organização oferece a vantagem de atualizar componentes do sistema individualmente (KUROSE; ROSS, 2010).

Em 1974, Cerf e Kahn desenvolveram o modelo TCP/IP (*Transmission Control Protocol/Internet Protocol*), com o objetivo de conectar várias redes sem grandes problemas

(TANEMBAUM, 2011). Segundo Forouzan (2010), o modelo TCP/IP possui quatro camadas de *software* em cima de uma camada de *hardware* e cada uma possui uma funcionalidade específica, mas não são independentes. Suas camadas são divididas em aplicação, transporte, rede, enlace e física, detalhadas na Tabela 2.

Tabela 2 – Modelo TCP/IP

Camada	Função
Aplicação	Aplicações de rede e seus protocolos (Como HTTP e FTP)
Transporte	Transmite dados usando os protocolos TCP e UDP
Rede	Endereçamento de pacotes usando o IP
Enlace	Roteamento de pacotes (Como Ethernet e Wi-Fi)
Física	Movimentação de bits individuais

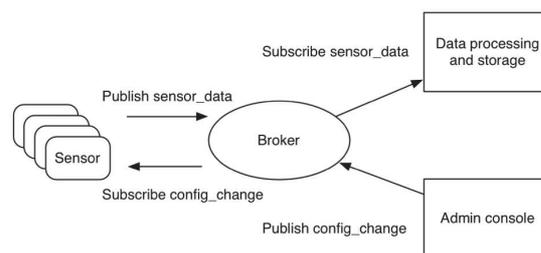
Fonte: Adaptado de Forouzan (2010).

2.3.4 Message Queue Telemetry Transport (MQTT)

Utilizando o modelo TCP/IP como base, a IBM criou um protocolo chamado MQTT, com o objetivo de conectar sensores em *pipelines* de petróleo a satélites. O MQTT envia e recebe mensagens de forma assíncrona, ou seja, não é necessária a verificação de conexão entre o remetente e o destinatário. Hoje em dia, este protocolo é utilizado em aplicações de Internet das Coisas e é descrito neste subcapítulo de acordo com Yuan (2017).

O protocolo MQTT funciona com um modelo de publicação e assinatura, organizado em um *broker* e seus clientes. O *broker* é um intermediário que interpreta as mensagens e as envia para seus clientes, e o cliente é qualquer dispositivo que consegue receber ou enviar mensagens interagindo com o *broker*. Esta comunicação é ilustrada na Figura 2.

Figura 2 – Modelo de inscrição e publicação



Fonte: Yuan (2017).

Ao se conectar com um *broker*, um cliente pode se inscrever em um tópico, assim, todas as mensagens direcionadas para este tópico, são recebidas pelo cliente. E o cliente pode enviar mensagens para um tópico, que faz com que todos os outros clientes inscritos recebam estas mensagens.

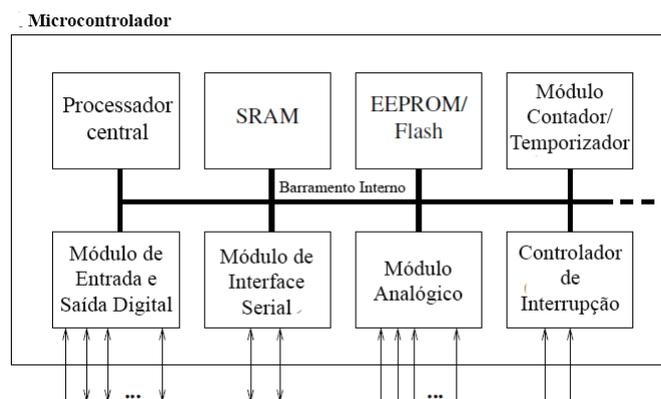
2.4 Microcontroladores

O microcontrolador é um dispositivo que possui vários componentes que funcionam em conjunto, em um só *chip*. Ele consiste, necessariamente de um microprocessador, responsável pelos cálculos aritméticos e comandos lógicos, memória, responsável pelo armazenamento do código e registros, e outros componentes que permitem que o microcontrolador funcione sem auxílio de componentes externos (TOCCI; WIDNER; MOSS, 2011; GRIDLING; WEISS, 2007).

Um dos primeiros microcontroladores surgiu em 1974 e se chamava TMS 1000, que além de possuir um microprocessador, incluía uma memória de acesso aleatório (memória RAM), memória somente de leitura (memória ROM), e pinos de entrada e saída. Os primeiros a serem usados em larga escala foram o Intel 8048 e Intel 8051, usados em teclados, e a série 68HCxx da Motorola (GRIDLING; WEISS, 2007).

A estrutura básica do microcontrolador pode ser vista no diagrama de blocos da Figura 3.

Figura 3 – Arquitetura básica de um microcontrolador



Fonte: Adaptado de Gridling e Weiss (2007).

Os itens a seguir são descritos de acordo com Hennessy e Patterson (2007), Gridling e Weiss (2007):

Unidade de processamento: se refere ao microprocessador, e contém uma unidade lógica aritmética e os registradores, um tipo de memória de altíssima velocidade.

SRAM: é uma RAM estática, sua principal característica é manter os dados sem necessidade de atualização constante, entretanto, perde seus dados ao deixar de ser alimentada.

EEPROM/Flash: do inglês, *Electrically Erasable and Programmable Read-Only Memory*, é uma memória não-volátil, ou seja, que mantém os dados armazenados mesmo sem alimentação. Os dados podem ser acessados ou apagados por meio de endereços de memória, um meio de identificar os dados salvos e suas posições. Em contrapartida, a memória flash, também não-volátil, só pode ter seus dados apagados em blocos, ao invés de apagados por endereço de

memória.

Temporizador/Contador: estes são usados pelo microcontrolador para marcar o tempo, e podem ser usados pelo programador para executar eventos cronometrados ou medir intervalos.

Pinos e entrada/saída digitais: são usados para controlar dispositivos externos. Podem ser programados de acordo com a função desejada e, normalmente, possuem funções especiais.

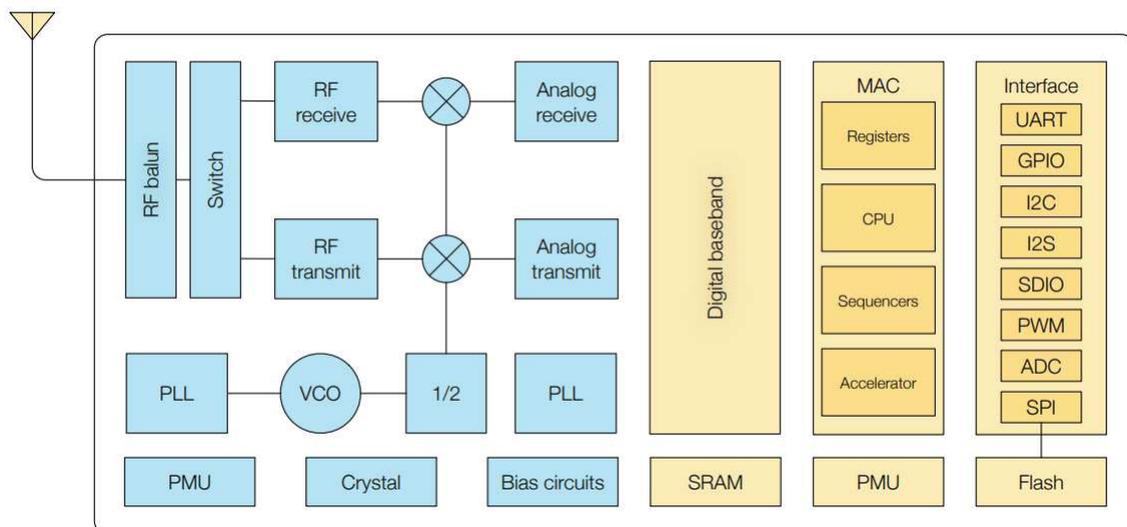
Interface Serial: se refere a comunicação serial, usada para a programação do dispositivo e interação com outros microcontroladores e periféricos. Sua comunicação exige um protocolo, cujos dispositivos devem ter em comum.

Pinos analógicos: sua função é de receber dados analógicos, que poderão ser interpretados pelo microcontrolador, via conversores digitais/analógicos.

2.4.1 ESP8266

Segundo Kolban (2016), o ESP8266 é um SoC (*System-on-a-Chip*) e foi desenvolvido pela *Espressif Systems*, uma empresa chinesa. Seu diferencial está em ter um módulo interno para comunicação Wi-Fi e também pode executar programas armazenados em sua memória. Este microcontrolador pode ser programado em diversas linguagens de programação, incluindo C, Javascript e Lua. A sua arquitetura é representada pelo diagrama de blocos da Figura 4.

Figura 4 – Arquitetura do ESP8266



Fonte: Espressif Systems (2018b).

2.5 Banco de dados

Um banco de dados pode ser definido por "um acervo de dados relacionados" (ELMASRI; NAVATHE, 2011, p. 4, **tradução nossa**). Ele é desenvolvido para armazenar grandes quantidades

de dados, que são acessados via software, de forma rápida e eficiente, e é usado de forma ampla, como, por exemplo, instituições bancárias, onde o banco de dados é usado para armazenar informações de clientes, transações bancárias, compras em crédito, e vendas e compras de ações (SILBERSCHATZ; KORTH; SUDARSHAN, 2011).

O sistema de banco de dados necessita de abstração de dados, ou seja, estes dados precisam estar organizados e armazenados de acordo com seus detalhes. A estrutura do banco de dados é organizada em tipos de dados, relações e limitações (ELMASRI; NAVATHE, 2011), o que pode ser visualizado na Figura 5.

Figura 5 – Estrutura de um banco de dados

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

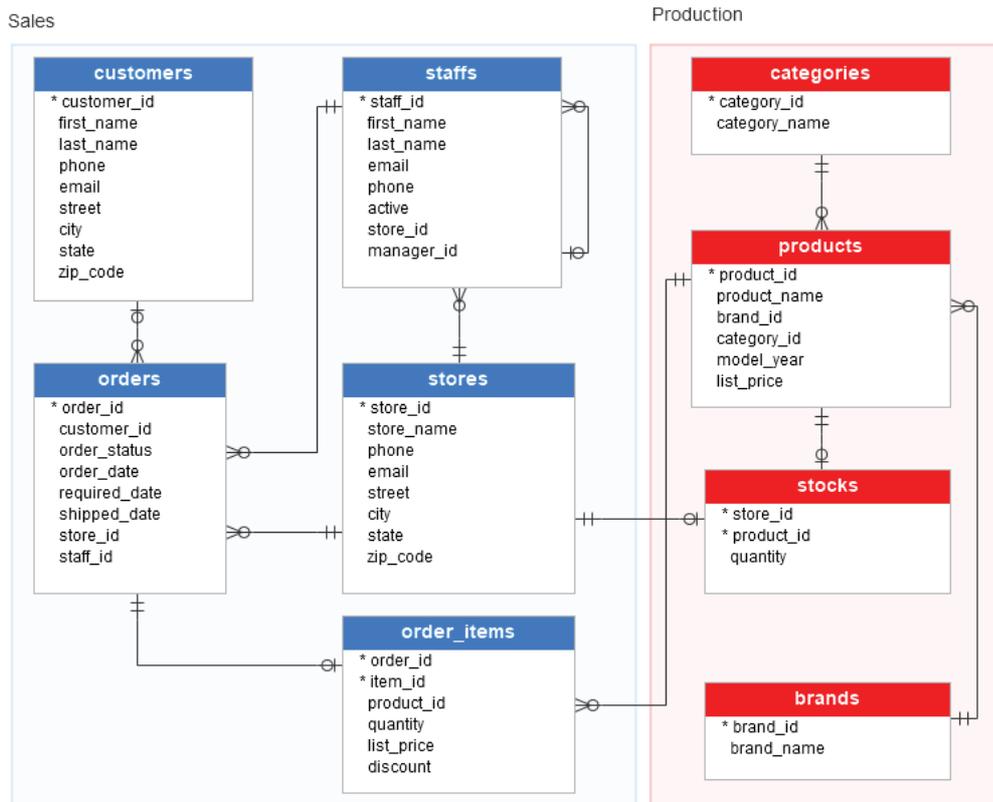
Student_number	Section_identifier	Grade
----------------	--------------------	-------

Fonte: Elmasri e Navathe (2011).

Os bancos de dados podem ser divididos em SQL (*Structured Query Language*) e NoSQL (*Not Only SQL*). SQL é uma linguagem de pesquisa relacional, ou seja, utiliza relações para armazenar dados, sendo que estas relações são chamadas de tabelas. Esta linguagem necessita de uma estrutura pré-definida para que os dados sejam armazenados, e como regra, estes dados precisam seguir esta mesma estrutura (SHIFF; ROWE, 2018).

Outra característica do banco de dados em SQL é sua escalabilidade. O SQL é escalado verticalmente, ou seja para aumentar o tamanho, é preciso adicionar componentes a um mesmo servidor, como discos rígidos, memórias voláteis e processadores. Logo, não é possível dividir o banco de dados em vários servidores (SHIFF; ROWE, 2018).

Figura 6 – Estrutura de um banco de dados em tabelas



Fonte: SQL Server Tutorial (2018).

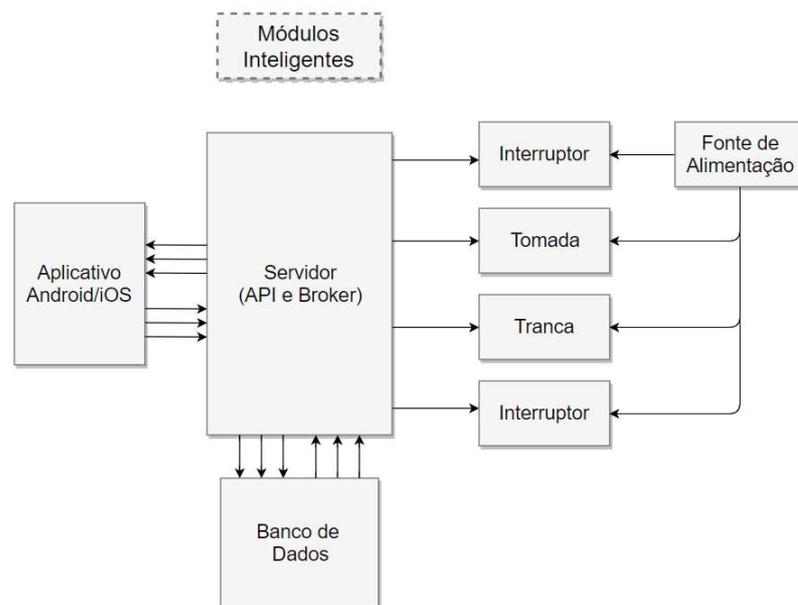
Já os banco de dados NoSQL, também chamados de Orientados a Documentos, armazenam seus dados em documentos autocontidos e autodescritivos, ou seja, o próprio documento define sua estrutura. Logo, diferente do SQL, não utiliza modelos relacionais, assim, não é preciso criar relações entre outros documentos, sendo possível separar os dados (MEDEITOS, 2014).

Isto torna o banco de dados mais flexível e dinâmico, aumentando a facilidade com que se consulta os dados. Sendo assim, é um banco de dados indicado para grande quantidade de dados, como sinais sensores. Também torna sua escalabilidade horizontal, ou seja, é possível aumentar o número de servidores que utilizam o mesmo banco de dados (MEDEITOS, 2014).

3 METODOLOGIA

O desenvolvimento do sistema foi dividido em etapas, que foram divididas, principalmente, entre o desenvolvimento dos *softwares*, projeto do *hardware* e testes. Em relação aos *softwares*, foram desenvolvidos um banco de dados, um servidor HTTP, um aplicativo nativo para os sistemas operacionais Android e iOS e um *firmware* para o microcontrolador. O *hardware* será composto por um Raspberry Pi B3+, que também servirá para a interação central dos *softwares* e dispositivos modulares, e estes também foram projetados, sendo eles: um interruptor, uma tomada, um controlador infravermelho e uma tranca, além de uma fonte chaveada para cada. Todo o sistema é visualizado na Figura 7.

Figura 7 – Diagrama de blocos do sistema



Fonte: Autoria Própria

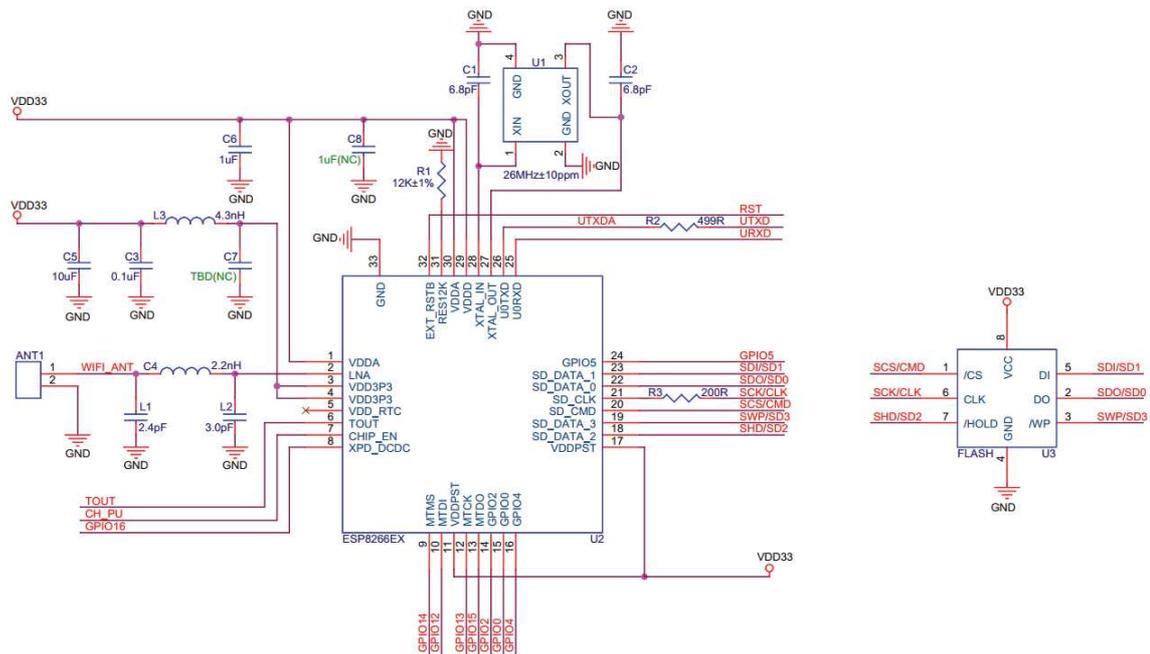
3.1 Projeto do *Hardware*

Esta etapa descreve o projeto do *hardware*, assim como as referências usadas para tal, a determinação dos componentes, simulações e montagem das placas de circuito impresso.

3.1.1 Microcontrolador

O projeto do circuito para o microcontrolador ESP8266 serviu de base para todos os módulos. Foi usado como base o guia para o projeto publicado pela Espressif, que apresenta todas as conexões necessárias para o funcionamento do sistema, descritos na Figura 8.

Figura 8 – Esquemático base para ESP8266



Fonte: Espressif Systems (2018a).

As especificações do projeto são:

- Frequência: 24 MHz
- Memória Flash: 4 MB
- Wi-Fi: 802.11 b/g/n/e/i, 2.4 GHz
- Consumo: cerca de 170 mA (Com Wi-Fi acionado)

3.1.2 Interruptor

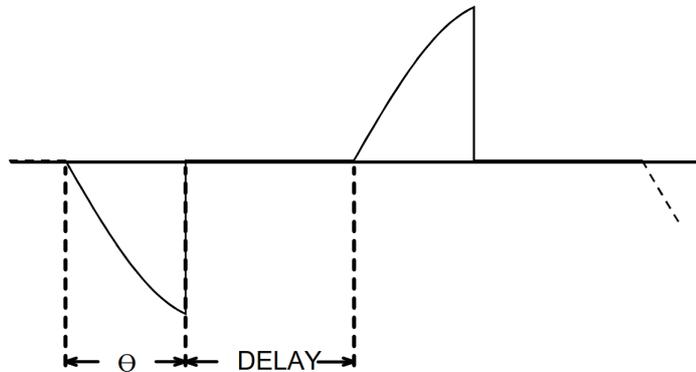
O interruptor, assim como todos os circuitos embarcados deste trabalho, é controlado pelo microcontrolador ESP8266EX. Os projetos destes circuitos foram baseados nas informações contidas no *datasheet* do mesmo, sendo que o diferencial entre os módulos é a etapa de saída de cada um.

Para a etapa de saída do interruptor foram considerados o acionamento de potência e a dimerização do mesmo, de forma que seja possível controlar lâmpadas incandescentes e lâmpadas LED (*light-emitting diode*) dimerizáveis. O restante das funções serão determinadas pelo *smartphone*.

Para o acionamento e dimerização foi projetado um circuito chamado *trailing edge dimmer* (Dimmer em Borda Traseira, em tradução literal). Este é baseado em uma detecção de zero na

senóide, gerando um atraso a partir deste ponto, e cortando a onda após o atraso, gerando a seguinte forma de onda na carga da Figura 9.

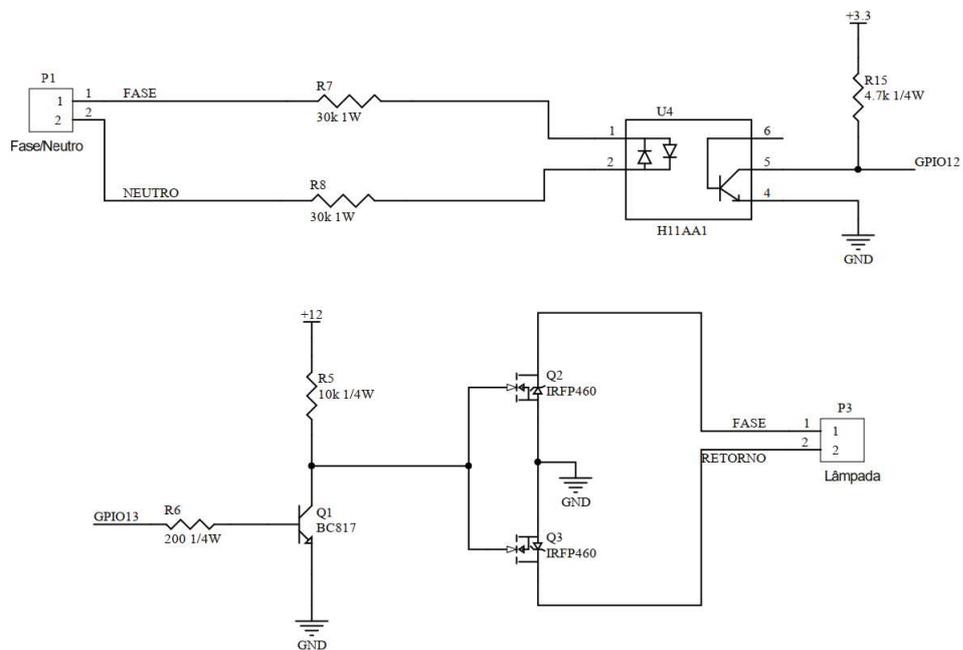
Figura 9 – *Trailing edge dimmer*



Fonte: TI (2011)

Logo, esta borda de descida gera menos ruído em comparação ao *dimmer* com TRIAC, além de possuir uma eficiência maior, pois detecta a tensão em zero com precisão. Sendo assim, o circuito a ser utilizado foi projetado da maneira ilustrada na Figura 10.

Figura 10 – Acionamento da lâmpada



Fonte: Autoria Própria

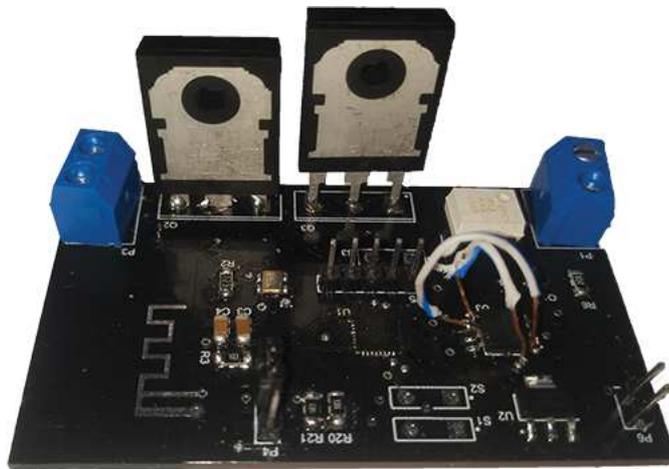
Seu princípio de funcionamento se baseia na detecção de zero por meio do optoacoplador U4, que possui dois LEDs em paralelo e inversamente posicionados, acoplados a um transistor, e

envia um sinal ao microcontrolador pelo pino GPIO12. Este sinal aciona uma interrupção no pino GPIO12, gerando um atraso interno. Após o tempo de atraso, o microcontrolador aciona o pino GPIO13, que desativa os MOSFETS, cortando a condutividade para a lâmpada, e gerando a forma de onda desejada.

Como adicional, para o controle manual do acionamento do interruptor, foram adicionados botões, que possuem prioridade em relação aos dados recebidos pelo MQTT.

A placa do circuito impresso deste módulo é visto na Figura 11.

Figura 11 – Placa de circuito impresso do interruptor

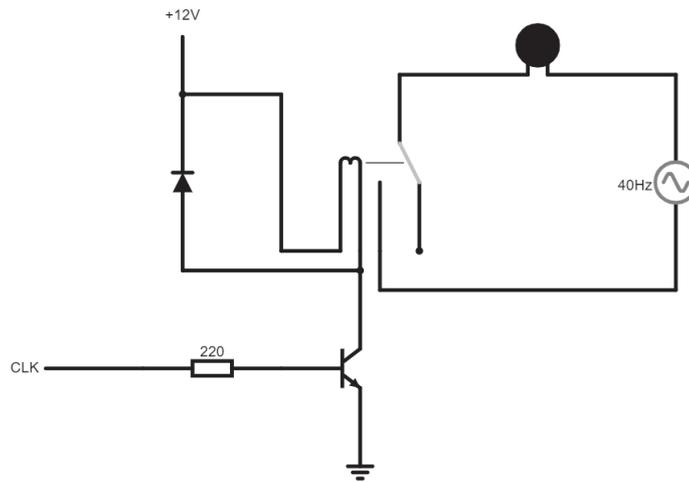


Fonte: Autoria Própria

3.1.3 Tomada

Para o projeto da tomada foram levados em consideração o seu acionamento, e a medição da potência utilizada pela carga conectada à tomada. Assim, foi projetado um circuito de acionamento por relé, assim como, foi utilizado um circuito integrado sensor de corrente por Efeito Hall, para medição da potência. O circuito base utilizado é visualizado na Figura 12.

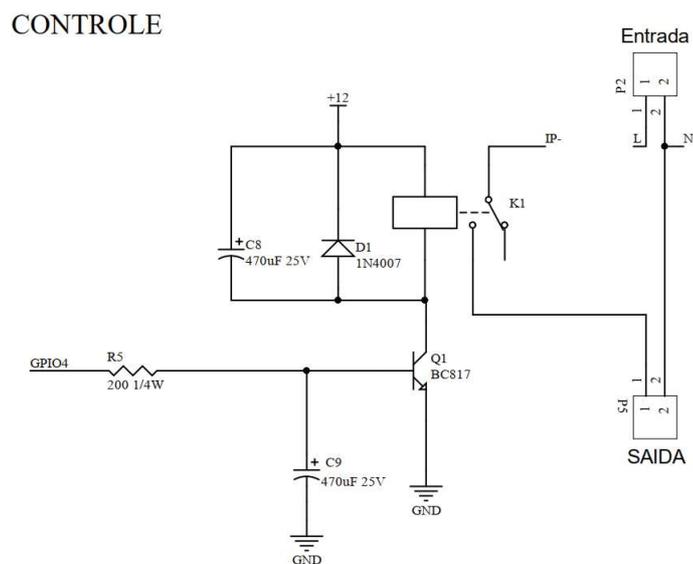
Figura 12 – Acionamento de relé



Fonte: Autoria Própria

No circuito final foram adicionados um capacitor em paralelo com a bobina, e outro em paralelo com a base-emissor. O propósito do primeiro é de manter a corrente estável na bobina, para evitar acionamentos indesejados no caso de variações na tensão. Já o segundo atrasa o acionamento na base do transistor, garantindo que a tomada só seja acionada se o sinal na base se manter. O circuito é apresentado na Figura 13.

Figura 13 – Acionamento da Tomada

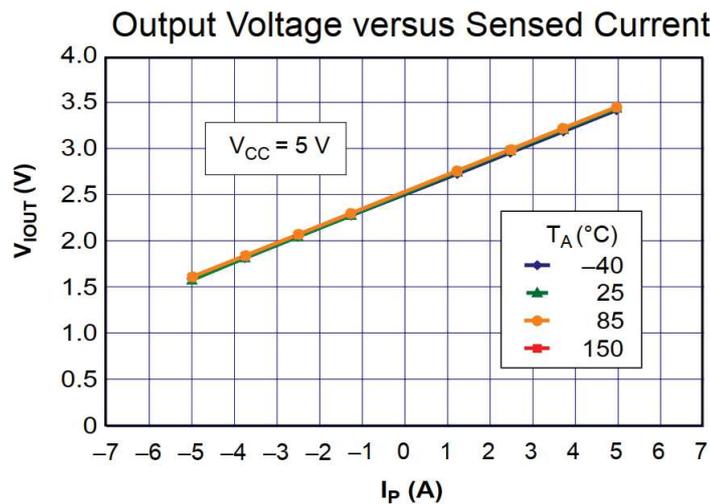


Fonte: Autoria Própria

Para o sensor de corrente foi utilizado o CI (Circuito Integrado) ACS712. Este CI converte

a corrente em tensão por meio do Efeito Hall. Devido a indisponibilidade do vendedor, foi adquirido a versão de 5 A. Este tem capacidade de medir corrente alternada ± 5 A, apresentando uma resolução de 185 mV/A (ALLEGRO, 2006). O datasheet apresenta o gráfico da tensão de saída, como visto na Figura 14, indicando que há um *offset* que precisou ser considerado no *firmware* mais tarde.

Figura 14 – Tensão de saída em relação a corrente na entrada



Fonte: Allegro (2006)

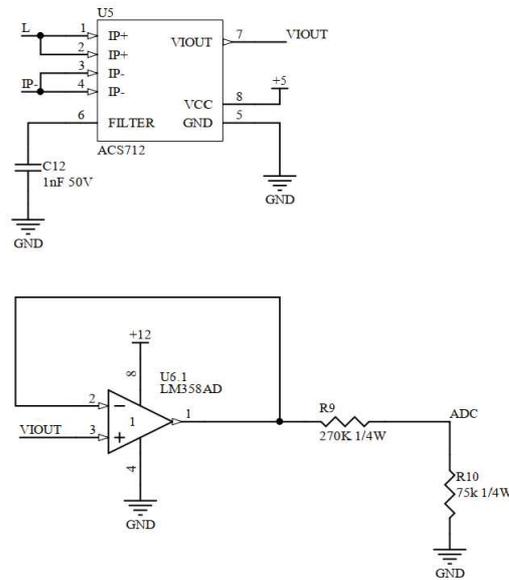
Segundo Espressif Systems (2018b), O ESP8266EX possui um conversor analógico digital com tensão máxima de 1 V. Para utilizá-lo foi preciso dimensionar um divisor de tensão para a saída do ACS712. O divisor foi definido utilizando:

$$R_{10} = V_{out} \frac{R_9}{V_{in} - V_{out}} \quad (3.1)$$

Tal que R_9 e R_{10} são os resistores usados para a divisão, V_{in} é a tensão de saída do ACS712, e V_{out} é a tensão que será lida pelo microcontrolador. R_9 foi fixado em $270 \text{ k}\Omega$, V_{in} foi definido para 4.5 V (1 V de tolerância) e V_{out} foi definido para 1 V, tendo como resultado um $R_{10} = 77 \text{ k}\Omega$. Utilizando um valor comercial, $R_{10} = 75 \text{ k}\Omega$.

Mas devido ao efeito de carga, causando atenuação na tensão de saída, foi adicionado um amplificador operacional, para melhor funcionamento da leitura da tensão analógica, tendo como projeto final:

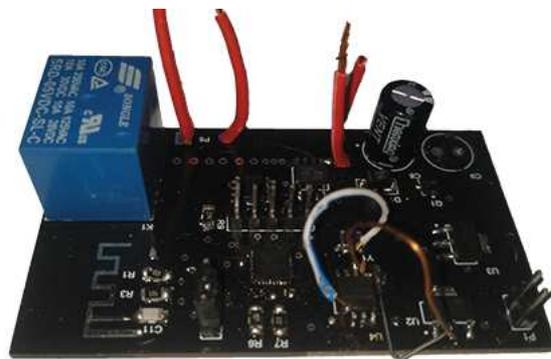
Figura 15 – Leitura de corrente



Fonte: Autoria Própria

A placa resultante da tomada inteligente é apresentada na Figura 16.

Figura 16 – Placa de circuito impresso da tomada



Fonte: Autoria Própria

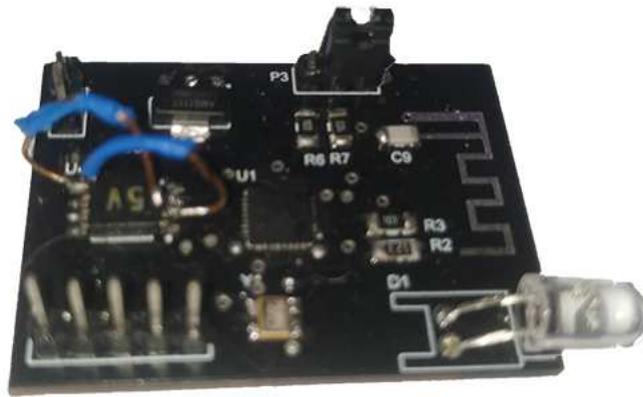
3.1.4 Tranca e Termostato

Estes circuitos possuem um esquemático mais simplificado em relação aos anteriores. O primeiro foi pensado para acionar trancas elétricas, que normalmente possuem acionamento com 12 V. Logo, o seu acionamento é via relé, de forma similar à tomada. Seu diferencial é devido a comunicação do microcontrolador com o módulo de identificação RFID (*Radio-Frequency IDentification*), que permite acesso à chaves (*tags*) que estão cadastradas no sistema. Desta forma, assim que a chave é aproximada do leitor, a tranca é aberta.

Já o termostato utiliza comunicação infravermelho. O microcontrolador envia sinais codificados para o eletrodoméstico que está sendo controlado. Devido ao acesso remoto, o ar-condicionado pode ser acionado de qualquer posição utilizando o *smartphone*, desde que o módulo esteja próximo ao mesmo.

As placas de circuito impresso resultantes para o termostato e a tranca são mostradas na Figura 17 e Figura 18, respectivamente.

Figura 17 – Placa de circuito impresso do termostato



Fonte: Autoria Própria

Figura 18 – Placa de circuito impresso da tranca



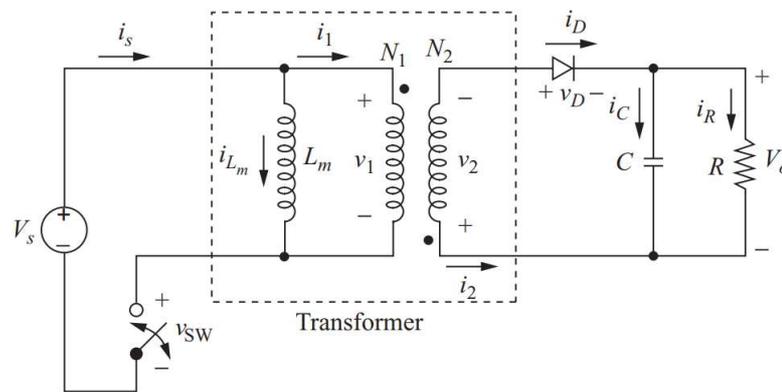
Fonte: Autoria Própria

3.1.5 Fonte Chaveada

Para a alimentação do circuito foi projetado um conversor CA/CC (Corrente Alternada/Corrente Contínua), para que os circuitos embarcados possam ser alimentados com a tensão da rede elétrica residencial. Para o seu projeto foi levado em consideração: a tensão necessária para o funcionamento do sistema embarcado e sua etapa de potência, o consumo de corrente do microcontrolador ao utilizar o *Wi-Fi* e também as informações contidas no *datasheet* do controlador de chaveamento utilizado no circuito.

Assim, foi definido a utilização de um conversor CA/CC *Flyback*, devido a sua ampla utilização, tamanho físico reduzido e a isolamento entre a entrada e a saída. As especificações determinadas foram $V_{out} = 12\text{ V}$ e $I_{out} = 1\text{ A}$.

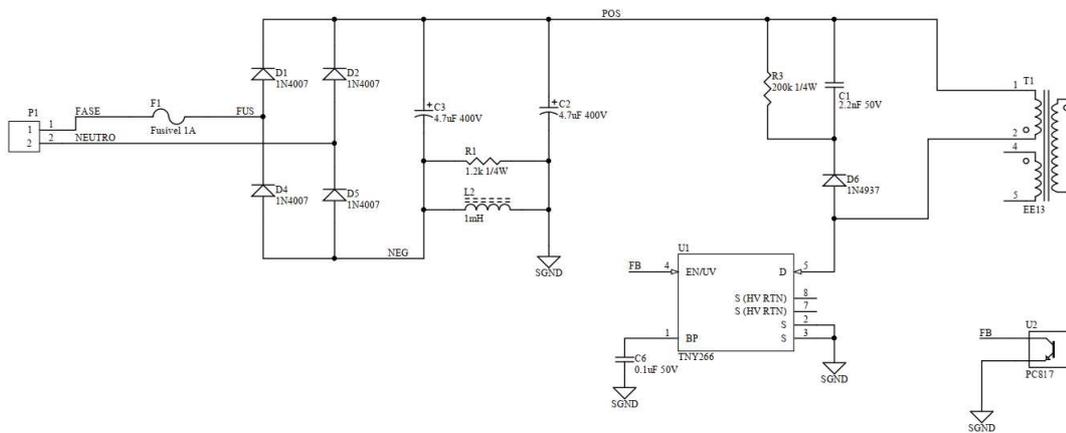
Figura 19 – Conversor CA/CC *Flyback* base



Fonte: Hart (2010)

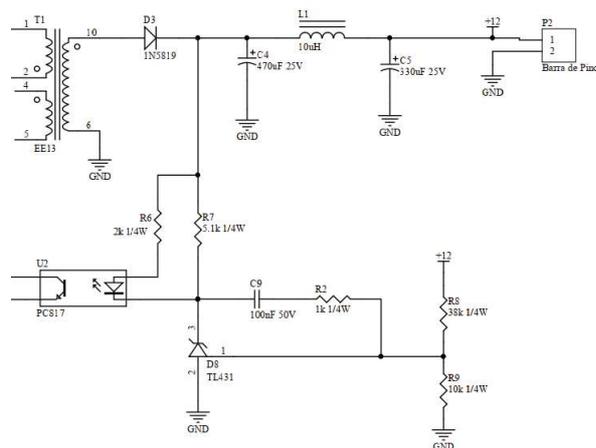
Visto que é necessário um chaveamento em alta frequência para o funcionamento da fonte chaveada, foi escolhido o TNY266 para o projeto, considerando indicações apontadas pelo fabricante em seu *datasheet*. O circuito final para a fonte chaveada pode ser visto nas figuras 20 e 21.

Figura 20 – Primário da fonte chaveada



Fonte: Autoria Própria

Figura 21 – Secundário da fonte chaveada



Fonte: Autoria Própria

No lado primário é possível identificar um filtro π formado pelos componentes C2, C3 e L2, um *snubber* RCD (Resistor-Capacitor-Diodo), formado por C1, R3 e D6 e o controlador TNY266. Segundo PI (2002) o filtro π é utilizado pela maior eficiência na redução de ruídos vindos da retificação. O circuito *snubber* limita a corrente de fuga do transformador, devido ao pico de tensão quando é desligado.

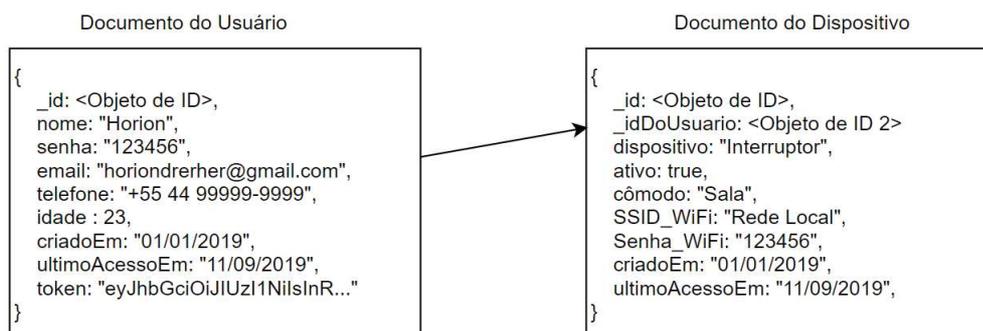
Entre os dois lados do transformador existe um circuito de realimentação definido pelo optoacoplador U2. Este aciona um pino de *feedback* do controlador, determinando a velocidade de chaveamento, e assim, a tensão de saída no secundário.

O sistema operacional escolhido para o Raspberry Pi foi o Ubuntu Server, por apresentar ferramentas dedicadas para uso de servidor, e também *softwares* atualizados, tal como o banco de dados utilizado, MongoDB. Para base do funcionamento do sistema, foi desenvolvido uma API (*Application Programming Interface*) que utiliza o protocolo para comunicação entre o *smartphone* e o banco de dados, assim como um *Broker* MQTT, para comunicação entre os *smartphone* e os módulos. O funcionamento de cada parte será descrito a seguir.

3.2.1.1 Banco de Dados

Foi implementado o banco de dados orientado a documentos, MongoDB, devido à sua maior velocidade em relação a um banco de dados SQL. Sabendo que o aplicativo necessita armazenar informações básicas para o usuário e seus módulos, assim como as informações pertinentes a este, o banco de dados construído apresentou a estrutura a seguir:

Figura 23 – Estrutura do banco de dados



Fonte: Autoria Própria

O documento do usuário contém informações dadas por ele e informações criadas pelo sistema. As informações dadas pelo usuário envolvem o nome, senha, email, telefone, idade, dispositivo conectado, localização e nome e senha do acesso *Wi-Fi* da residência. Estas informações podem ser atualizadas como o usuário quiser. Por outro lado, os itens, Objeto de ID (*Identity Document*), data de criação, data de último acesso e o *token* são dados pelo sistema. O objeto de ID é uma identificação única que cada item do banco de dados recebe e o *token* determina quais dispositivos estão conectados com estas identificações.

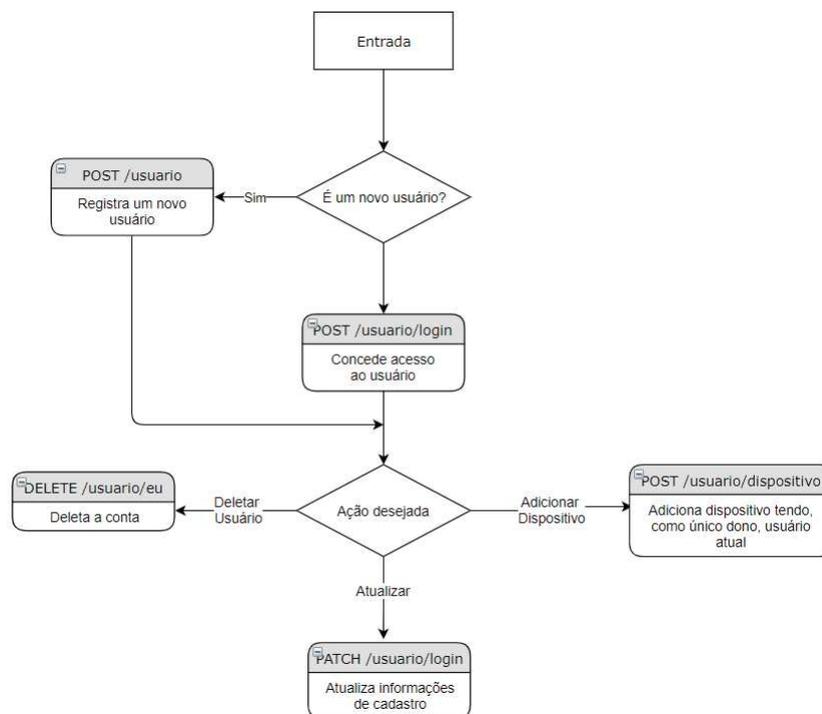
O *token*, em especial, é o que determina se o usuário pode acessar as informações do banco de dados. Ele é dado ao usuário no momento em que faz *login* e é removido ao ser desconectado. Assim, se a mesma conta estiver conectada em vários *smartphones*, vários *tokens* serão armazenados no sistema.

3.2.1.2 API

O aplicativo não apresenta a possibilidade de se comunicar diretamente com o banco de dados. Assim, foi necessária a criação de uma API, um programa com o protocolo HTTP que permite a comunicação entre os dois, criando uma ponte no sistema. A API foi escrita em *JavaScript* e foi armazenada no *Raspberry Pi*, onde fica em execução a todo momento.

Por utilizar o protocolo HTTP, a API apresenta quatro métodos básicos: *GET* (obter dados), *POST* (Criar dados), *PATCH* (atualizar dados) e *DELETE* (remover dados). Assim, a API funciona interpretando as requisições solicitadas pelo usuário no *smartphone*, e a partir destes métodos, decide quais operações serão realizadas aos dados contidos nas requisições. O fluxograma mostra, de forma mais clara, como a API implementada para o sistema funciona:

Figura 24 – Fluxograma de dados da API



Fonte: Autoria Própria

3.2.2 Firmware

Utilizando o microcontrolador ESP8266EX é possível acessar o Wi-Fi, e assim, utilizar protocolos de rede para enviar dados para outros dispositivos. O ESP8266EX ainda possui compatibilidade com a linguagem LUA, o que permite criar um *firmware* de alto nível, sendo mais fácil de gerir o sistema, garantindo maior estabilidade do mesmo.

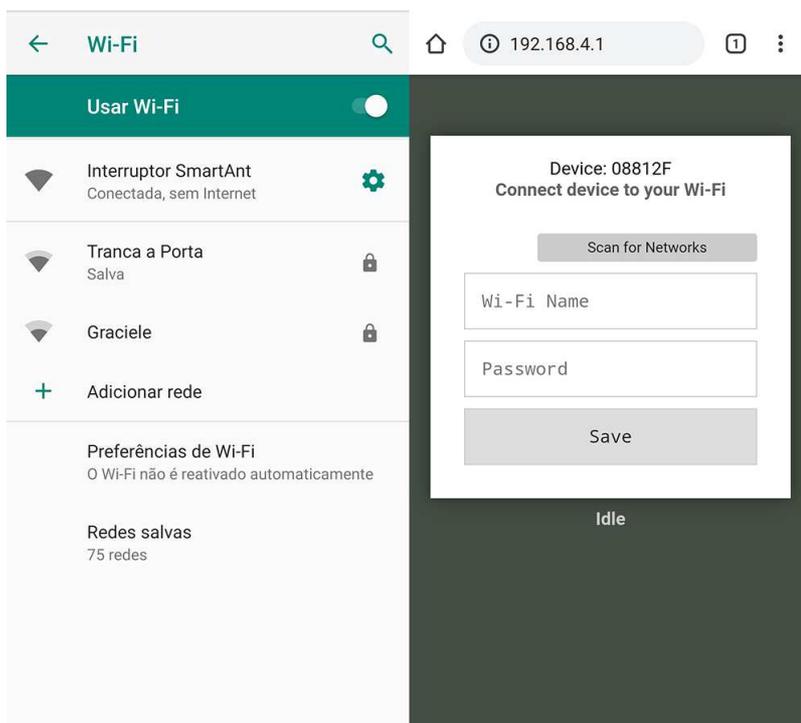
Usando estes recursos, foi implementado um *firmware* nos módulos capaz de enviar mensagens com o protocolo MQTT para o servidor, via Wi-Fi, que armazena o *broker*,

centralizando as mensagens enviadas. Também possui funções dedicadas para cada tipo de módulo:

- **Interruptor:** permite o acionamento, dimerização e temporização das lâmpadas.
- **Tomada:** permite o acionamento, temporização e leitura da potência das cargas conectadas à tomada.
- **Tranca:** permite o acionamento de trancas elétricas a partir da leitura de *tags* RFID autorizadas no sistema
- **Termostato:** permite o controle de ar-condicionados por meio de códigos infra-vermelho

Também foi adicionada uma pequena interface para o usuário conseguir adicionar o dispositivo à rede, inserindo o nome e senha da mesma. Isto faz com que o módulo seja ativado, inicializando a configuração:

Figura 25 – Configuração inicial do módulo



Fonte: Autoria Própria

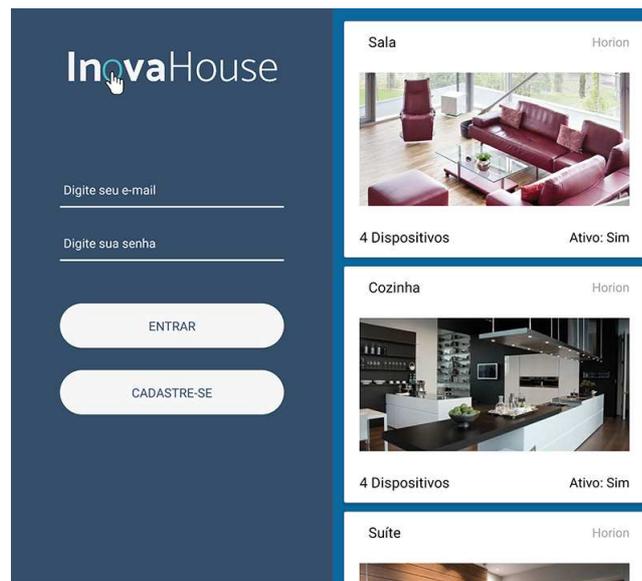
3.2.3 Aplicativo

Por fim, com todos os componentes que controlam o sistema criado, a interface de usuário foi desenvolvida para aplicativos Android/iOS. Para isto, um *framework JavaScript*, chamado

React Native, foi utilizado. *React Native* renderiza um programa escrito em *JavaScript* para uma aplicação nativa, facilitando o desenvolvimento entre plataformas.

Então, a partir do sistema implementado, foram criadas algumas telas. O aplicativo envia requisições à API, de acordo com os comandos inseridos, por meio do protocolo HTTP e que permitem que o usuário interaja com o sistema, criando contas, realizando *login*, acessando e controlando os dispositivos adicionados ao sistema:

Figura 26 – Exemplos de telas criadas para interface do usuário

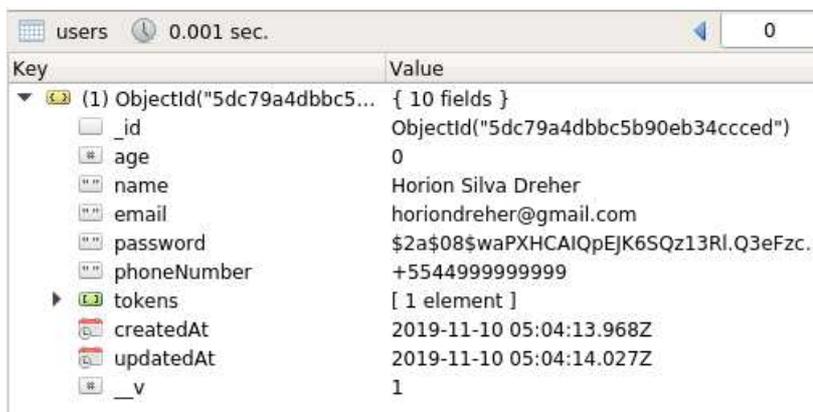


Fonte: Autoria Própria

superior da Figura 27) foi de 109 *ms* e retornou o *status* 201, que para o protocolo HTTP quer dizer "criado com sucesso". A requisição também enviou uma resposta (localizada na posição inferior da Figura 27) indicando o Objeto *ID*, horário de criação e horário de atualização, assim como foi configurado. Também retornou o *token* de acesso para o *Postman*.

Sendo assim, para verificação dos dados, utilizou-se o *Robo3T* conectado ao mesmo banco de dados utilizado pela API. O resultado da inserção dos dados é apresentado na Figura 28.

Figura 28 – Banco de dados após criação do usuário

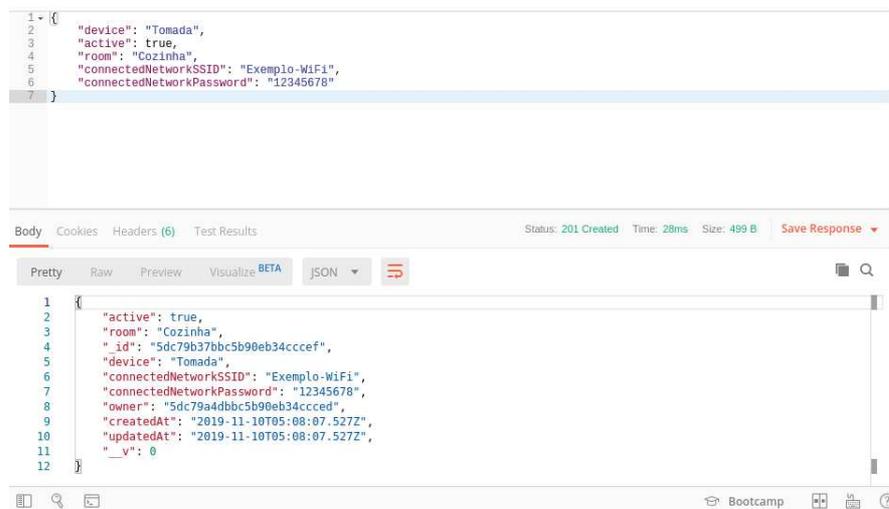


Key	Value
(1) ObjectId("5dc79a4dbbc5...")	{ 10 fields }
_id	ObjectId("5dc79a4dbbc5b90eb34ccced")
age	0
name	Horion Silva Dreher
email	horiondreher@gmail.com
password	\$2a\$08\$waPXHCAIQpEJK6SQz13RI.Q3eFzc...
phoneNumber	+5544999999999
tokens	[1 element]
createdAt	2019-11-10 05:04:13.968Z
updatedAt	2019-11-10 05:04:14.027Z
__v	1

Fonte: Autoria Própria

Vale observar que o usuário possui o mesmo Objeto de *ID*, horário de criação e atualização da requisição enviada, validando os dados. Também é visto que a senha no banco de dados não é a mesma. Isto acontece porque a API configurada codificou a senha utilizando criptografia *hash*, muito utilizada em segurança da informação. Apenas a API realiza a descriptografia.

Outro teste realizado foi a inserção de um dispositivo tendo o mesmo usuário como proprietário. O dispositivo simulado se chama "Tomada", está localizado na "Cozinha" e a rede em que está conectado se chama "Exemplo-WiFi", visualizado na Figura 29.

Figura 29 – Inserção do dispositivo realizada pelo *Postman*

Fonte: Autoria Própria

Como resposta, foi indicado o sucesso da criação. O resultado, mostrado na Figura 30, pôde ser visto no banco de dados.

Figura 30 – Dispositivo no banco de dados

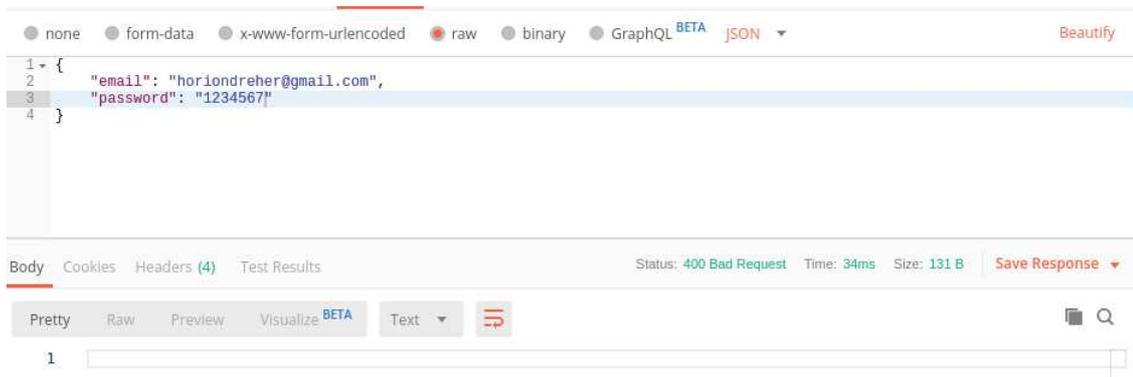
Key	Value
(1) ObjectId("5dc79b37bbc5...")	{ 10 fields }
_id	ObjectId("5dc79b37bbc5b90eb34c...")
active	true
room	Cozinha
device	Tomada
connectedNetworkSSID	Exemplo-WiFi
connectedNetworkPassw...	12345678
owner	ObjectId("5dc79a4dbbc5b90eb34cc...")
createdAt	2019-11-10 05:08:07.527Z
updatedAt	2019-11-10 05:08:07.527Z
_v	0

Fonte: Autoria Própria

O sucesso na operação é indicado pela comparação entre os dados enviados e os dados armazenados, assim como o Objeto *ID* do criador do dispositivo, indicado como "*owner*".

O último teste realizado na API foi a tentativa de *login* (GET) com uma senha incorreta ("1234567"). A resposta dada pela API foi o *status* 400 que quer dizer "requisição inválida", o que pode ser visto na Figura 31.

Figura 31 – Login incorreto na API



Fonte: Autoria Própria

Estes testes indicam o funcionamento correto da API criada para o sistema.

4.2 Hardware

Nesta etapa, foram realizados utilizando a tensão da rede e tensão da fonte de bancada fornecidas pela universidade. Para a visualização dos resultados foram utilizados um osciloscópio de dois canais e um multímetro.

4.2.1 Fonte Chaveada

Para iniciar os testes da fonte, foi, simplesmente, aplicada a tensão de rede nos seus terminais de entrada, e verificada a tensão nos terminais de saída.

Visto que nenhuma tensão estava sendo direcionada à saída, foram verificados os terminais de entrada do transformador, indicando a tensão retificada da rede esperada (179,60 V em corrente contínua).

Sendo assim, foi determinado o mal funcionamento do transformador utilizado na fonte chaveada, tal que não houve possibilidade de troca devido a não disponibilidade do mesmo e a grande dificuldade de adquirir ou construir outro com as propriedades desejadas.

4.2.2 Microcontroladores

O primeiro item testado após a confecção das placas de circuito impresso foi o funcionamento dos microcontroladores, sua memória *flash* externa e sua comunicação com o Wi-Fi.

Devido a falta de um driver USB nos módulos, utilizou-se um módulo de comunicação USB FTDI, que se comunica por meio de interface serial. Foi a partir dele que os dados foram visualizados para verificar o funcionamento correto do sistema embarcado.

Assim, após a confecção dos quatro módulos, todos os microcontroladores funcionaram apresentando a mesma resposta esperada. O primeiro passo do teste foi o carregamento do *firmware* base para o uso da linguagem Lua. A Figura 32 apresenta os dados do microcontrolador após uma conexão bem sucedida, que descrevem suas características como frequência de processamento o tamanho da memória de arquivo. Logo após mostra o tamanho do programa escrito em sua memória e uma mensagem de ação completada.

Figura 32 – Resposta do microcontrolador

```
[horiondreher@localhost ModuloTomada]$ sudo esptool.py --port /dev/ttyUSB0 --baud 115200 write_flash -fm dio
0x000000 ~/Documents/TCC/Binaries_NodeMCU/nodemcu_inova.bin
[sudo] password for horiondreher:
esptool.py v2.7
Serial port /dev/ttyUSB0
Connecting.....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 18:fe:34:f8:72:c2
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0240
Compressed 520192 bytes to 340123...
Wrote 520192 bytes (340123 compressed) at 0x00000000 in 30.2 seconds (effective 138.0 kbit/s)...
Hash of data verified.

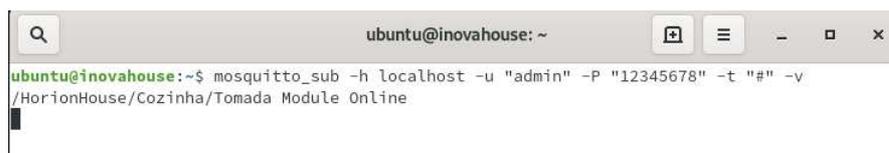
Leaving...
Hard resetting via RTS pin...
[horiondreher@localhost ModuloTomada]$
```

Fonte: Autoria Própria

É visto que o circuito do microcontrolador confeccionado apresentou o funcionamento correto, sendo que este identificou todas as especificações projetadas para o mesmo.

Logo após, foi carregado o *firmware* do sistema, esperando a conexão correta com o Wi-Fi e envio da primeira mensagem ao *broker* MQTT. Após o carregamento, foi possível a conexão correta com a rede sem fio, e momentos depois a mensagem foi recebida no *broker*, o que é visto na Figura 33.

Figura 33 – Módulo tomada conectado ao MQTT



```
ubuntu@inovahouse: ~
ubuntu@inovahouse:~$ mosquitto_sub -h localhost -u "admin" -P "12345678" -t "#" -v
/HorionHouse/Cozinha/Tomada Module Online
```

Fonte: Autoria Própria

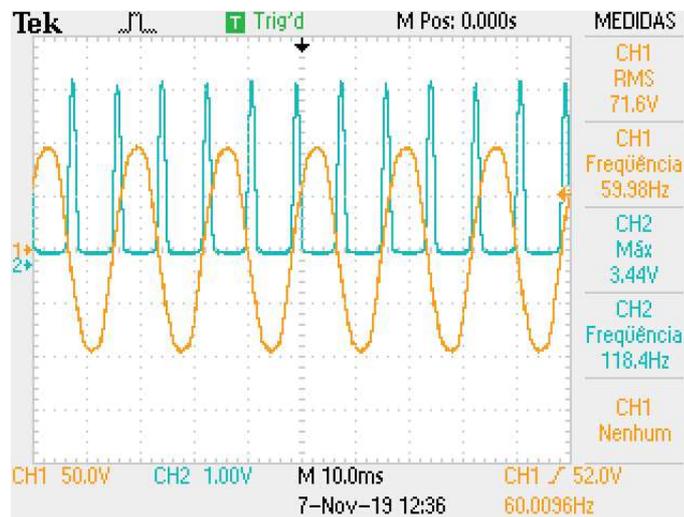
4.2.3 Interruptor

O teste do módulo do interruptor envolveu a inserção da tensão da rede no circuito e instalação de uma lâmpada incandescente na saída, para o teste de dimerização e acionamento via MQTT.

Devido ao correto funcionamento dos microcontroladores, as mensagens via protocolo MQTT foram recebidas normalmente, sendo possível o acionamento pelo protocolo.

O primeiro teste realizado foi o funcionamento da detecção de zero, que é utilizada para a função de dimerização. Utilizando o osciloscópio, o canal 1 (amarelo) foi conectado a tensão da rede, e o canal 2 (azul) foi conectado à saída do optoacoplador. O sinal é visualizado na Figura 34.

Figura 34 – Detecção de zero no circuito

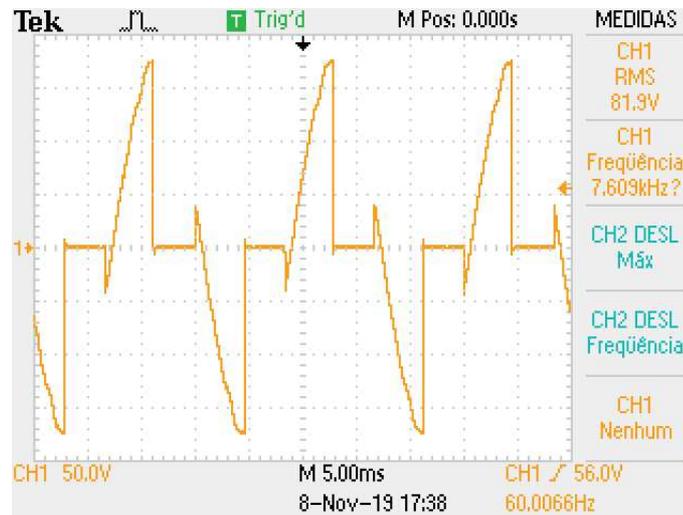


Fonte: Autoria Própria

É visto que o circuito apresenta o funcionamento correto, com o pico de tensão de 3 V assim que a senóide da rede atinge 0 V.

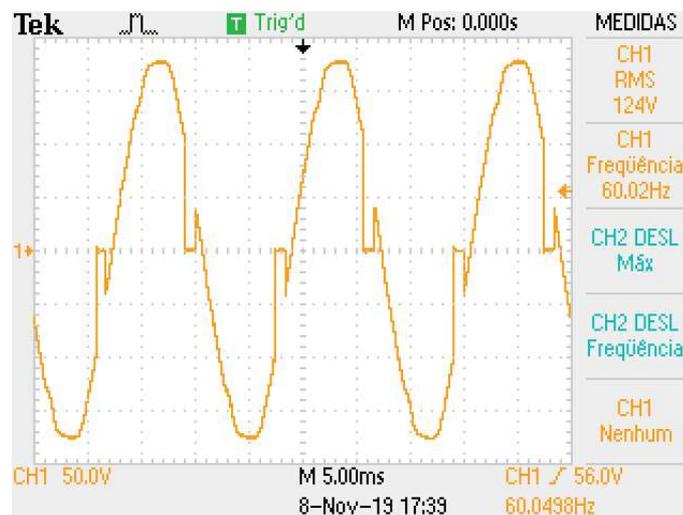
Logo depois a lâmpada foi conectada a saída do circuito e ao osciloscópio, e via MQTT foram definidas duas intensidades de iluminação: 50% e 100%, visualizados nas Figura 35 e Figura 36, respectivamente.

Figura 35 – 50% de intensidade da lâmpada



Fonte: Autoria Própria

Figura 36 – 100% de intensidade da lâmpada



Fonte: Autoria Própria

Foi observado picos de tensão perto do ponto onde a senóide atinge 0 V. Isto pode acontecer porque a detecção de zero pode estar apresentando uma borda de subida antes da senoíde atingir 0 V, o que interfere no funcionamento do *firmware*.

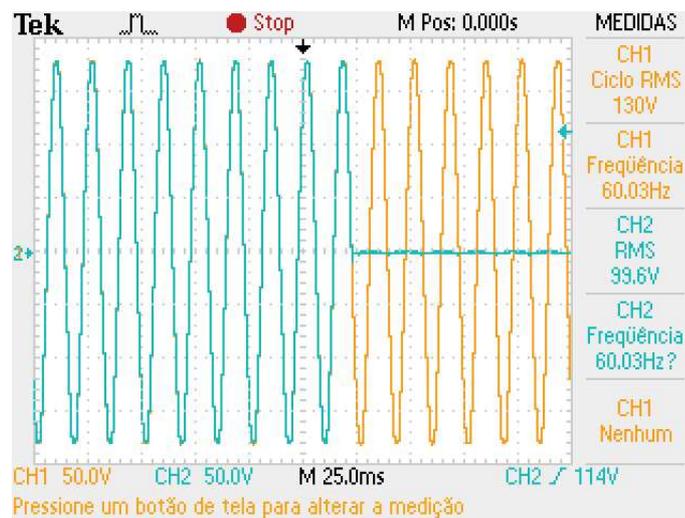
Assim, foi determinado que o módulo interruptor apresenta funcionamento correto, mas com algumas interferências.

4.2.4 Tomada

O teste do módulo tomada se deu pelo acionamento da tomada via MQTT, visualização do funcionamento no osciloscópio e da corrente utilizada pelo circuito.

Ao inserir a tensão de rede no circuito, foi realizado o teste de acionamento que pôde ser visto no osciloscópio, sendo o canal 1 (amarelo), a tensão na entrada, e o canal 2 (azul), a tensão na saída, o que é visto na Figura 37.

Figura 37 – Acionamento da tomada



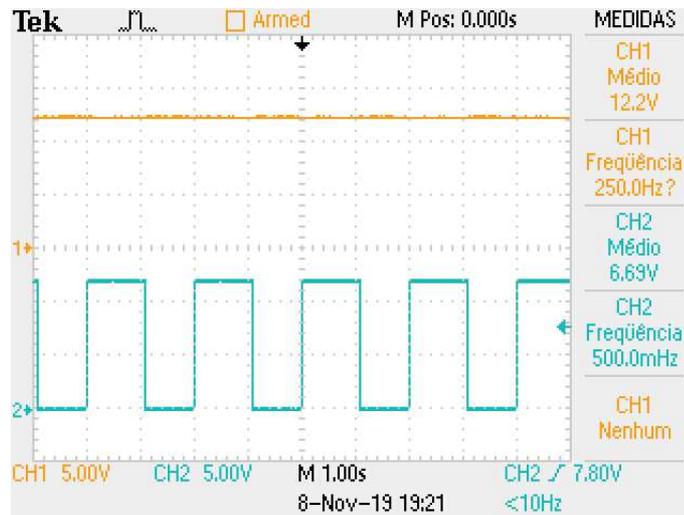
Fonte: Autoria Própria

Assim, o acionamento apresentou o funcionamento correto. Em contrapartida, o sensor de corrente não realizou as leituras corretamente, sempre apresentando a corrente máxima (5 A), mesmo sem alguma carga conectada à saída. Isto pode ter ocorrido devido ao circuito integrado estar danificado.

4.2.5 Tranca

Para a tranca testes foram realizados verificando o acionamento de 12 V, que foi controlado com um temporizador de 1 segundo, apenas para obter resultados visuais, que foram observados no osciloscópio, tal qual apresentou o funcionamento correto, mostrado na Figura 38.

Figura 38 – Acionamento da tranca



Fonte: Autoria Própria

Devido a problemas com a comunicação entre o módulo tranca e o leitor RFID, que utiliza comunicação SPI (*Serial Peripheral Interface*) não foi possível realizar a autenticação pela *tag*.

Entretando, para verificar o bom funcionamento do sistema, foram enviados dados para o servidor, simulando a codificação RFID, o que mostrou o funcionamento esperado da autenticação no sistema.

5 CONCLUSÃO

O desenvolvimento do presente trabalho apresentou a execução de um sistema de dispositivos inteligentes para residências, envolvendo a criação de um sistema de *softwares* que funcionam com o objetivo de permitir um controle mais fácil e acessível à casa.

Para tal, foi implementado um servidor em um computador acessível e o projeto de módulos inteligentes de baixo custo que permitem que várias funções sejam adicionadas devido a sua flexibilidade. Também foi desenvolvida uma interface para o usuário conseguir utilizar o sistema da maneira simplificada.

O estudo e desenvolvimento deste tipo de sistema inteligente está cada vez sendo de maior importância para o mercado: devido ao grande crescimento da tecnologia de Internet das Coisas, a procura de empresas e desenvolvedores, que possuem este tipo de conhecimento e produtos na área, está aumentando.

E assim, um sistema desta área apenas apresentando um funcionamento simples, como o ligar e desligar, não é mais suficiente. Este sistema precisa possuir grande qualidade de funcionamento, com comunicações rápidas e estáveis por trás dos dispositivos. E o mais importante: segurança digital, como foi discutida na introdução.

O maior foco deste trabalho foi a segurança digital no desenvolvimento da API. Com a utilização de *tokens*, *hash codes*, autenticação por usuário e senha e utilização de sistema operacional dedicado a servidores, houve uma utilização confiável do sistema.

Além destas medidas de segurança também houve a tentativa de implementar o protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) que utiliza criptografia ao enviar os dados pela rede, impedindo que estes dados sejam interceptados. Mas para isso é preciso obter um certificado SSL (*Secure Sockets Layer*), que apenas é concedido para servidores com domínios DNS (*Domain Name System*). Portanto, por falta de recursos e conhecimento nesta área, o protocolo HTTPS não foi implementado.

Entretanto, o desenvolvimento dos *softwares* apresentou o funcionamento esperado, atingindo o objetivo quando o servidor permitiu a ponte de comunicação entre o aplicativo e o *broker* e banco de dados, armazenando as informações corretamente.

De outra forma, as placas de circuito impresso não apresentaram todos os resultados esperados, como o não funcionamento do sensor de corrente no módulo tomada e o problema de comunicação do sensor RFID. Da mesma forma, a fonte chaveada que tinha o objetivo de alimentar os módulos com a tensão da rede, não mostrou o funcionamento.

Mas, com exceção destes problemas, todos os outros itens dos módulos apresentaram o funcionamento correto, com os *firmwares* conectando sem problemas à rede sem fio e

respondendo da forma desejada aos comandos enviados através *broker* como foram mostrados no capítulo de resultados. Logo, o objetivo da confecção das placas de circuito impresso foi atingido quase que completamente.

No futuro, espera-se que o sistema apresente maior segurança digital, com o uso do protocolo HTTPS, assim como a construção de placas de circuito impresso mais elaboradas e com erros corrigidos, para garantir a boa estabilidade dos módulos.

REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5410**: Instalações elétricas em baixa tensão. Rio de Janeiro, 2004. 217 p.
- ALLEGRO MICROSYSTEMS. **ACS712**: Datasheet. Worcester, 2006. 14 p. Disponível em: <<https://www.allegromicro.com/-/media/Files/Datasheets/ACS71240-Data-Sheet.ashx>>. Acesso em: 2 ago. 2019.
- ASADULLAH, M.; RAZA, A. An overview of home automation systems. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND ARTIFICIAL INTELLIGENCE (ICRAI), 2., 2016, Los Angeles. **Anais[...]**. Los Angeles: IEEE, 2016. p. 27–31.
- ASSOCIAÇÃO BRASILEIRA DE AUTOMAÇÃO RESIDENCIAL E PREDIAL. **Automação Residencial: demanda na Construção Civil**. 2013. Disponível em: <<http://www.aureside.org.br/noticias/automacao-residencial--demanda-na-construcao-civil>>. Acesso em: 19 abr. 2019.
- BÓ, P. H. D.; MURATORI, J. R. **Automação Residencial**: Conceitos e aplicações. 2. ed. Belo Horizonte: Educere, 2014.
- BOYLESTAD, R. L. **Introdução à Análise de Circuitos**. 12. ed. São Paulo: Pearson Prentice Hall, 2012.
- CAVALIN, G.; CERVELIN, S. **Instalações Elétricas Prediais**: Conforme norma nbr 5410:2004. 21. ed. São Paulo: Érica, 2013.
- CHASE, J. The evolution of the internet of things. **Texas Instruments**, v. 1, 2013.
- CONSTANTIN, L. **Sistemas de automação residencial contêm falhas, dizem especialistas**. 2013. Disponível em: <<https://itmidia.com/sistemas-de-automacao-residencial-contem-falhas-dizem-especialistas/>>. Acesso em: 19 abr. 2019.
- COSTA, A. **Automação residencial: tudo o que você precisa saber a respeito**. 2017. Disponível em: <<https://www.tuacasa.com.br/automacao-residencial/>>. Acesso em: 19 abr. 2019.
- CREDER, H. **Instalações Elétricas**. 16. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 2016.
- ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 6. ed. Boston: Addison-Wesley, 2011.
- ESPRESSIF SYSTEMS. **ESP8266**: Hardware design guidelines. Shangai, 2018. 29 p. Disponível em: <<https://www.espressif.com/en/support/download/documents>>. Acesso em: 6 jun. 2019.
- ESPRESSIF SYSTEMS. **ESP8266EX**: Datasheet. Shangai, 2018. 29 p. Disponível em: <<https://www.espressif.com/en/products/hardware/esp8266ex/overview>>. Acesso em: 2 abr. 2019.

- FOROUZAN, B. A. **TCP/IP Protocol Suite**. 4. ed. New York: McGraw-Hill, 2010.
- GRIDLING, G.; WEISS, B. **Introduction to Microcontrollers**. 1. ed. Viena: Vienna University of Technology, 2007.
- HARPER, R. **Inside the Smart Home**. 1. ed. London: Springer Verlag, 2003.
- HART, D. W. **Automation and Control**. 1. ed. New York: McGraw Hill, 2010.
- HENDRICKS, D. **The history of home automation**. 2014. Disponível em: <<https://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm>>. Acesso em: 13 maio 2019.
- HENNESSY, J. L.; PATTERSON, D. A. **Computer Architecture: A quantitative approach**. 4. ed. San Francisco: Elsevier, 2007.
- HP - HEWLETT-PACKARD. **The Internet of Things: Today and tomorrow**. Palo Alto, 2017. 20 p.
- INFOMONEY. **Mercado de automação residencial segue com boas perspectivas no país**. 2018. Disponível em: <<https://www.infomoney.com.br/negocios/noticias-corporativas/noticia/7807535/mercado-de-automacao-residencial-segue-com-boas-perspectivas-no-pais>>. Acesso em: 19 abr. 2019.
- KOLBAN, N. **Kolban's book on the ESP8266**. 1. ed. Texas: [s.n.], 2016.
- KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet: Uma abordagem top-down**. 5. ed. São Paulo: Pearson Education, 2010.
- MACDONALD, D. **The Smart Home Is Here Today**. 2007. Disponível em: <<https://ezinearticles.com/?The-Smart-Home-Is-Here-Today&id=481641>>. Acesso em: 30 out. 2019.
- MEDEITOS, H. **Introdução ao MongoDB**. 2014. Disponível em: <<https://www.devmedia.com.br/introducao-ao-mongodb/30792>>. Acesso em: 31 out. 2019.
- NEOCONTROL. **Por que a automação sem fio é o ideal para se tornar um integrador?** 2017. Disponível em: <<https://www.neocontrol.com.br/news/automacao-sem-fio/>>. Acesso em: 19 abr. 2019.
- POWER INTEGRATIONS. **TNY263-268**: Datasheet. San Rose, 2002. 22 p. Disponível em: <https://www.power.com/sites/default/files/product-docs/tny263_268.pdf>. Acesso em: 9 ago. 2019.
- RODIC, A. **Automation and Control: Theory and practice**. 1. ed. Vukova: Intech, 2009.
- ROUSE, M. **Smart home or building: home automation or domotics**. 2018. Disponível em: <<https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building>>. Acesso em: 29 maio 2019.
- SAUTER, T.; SOUCEK, S.; KASTNER, W.; DIETRICH, D. The evolution of factory and building automation. **IEEE Industrial Electronics Magazine**, v. 5, n. 3, p. 35–48, 2011. ISSN 1941-0115. Disponível em: <<https://ieeexplore-ieee-org.ez48.periodicos.capes.gov.br/document/6042563>>. Acesso em: 1 abr. 2019.

SHIFF, L.; ROWE, W. **What is the Difference between NoSQL and SQL?:** Nosql vs sql explained. 2018. Disponível em: <<https://www.bmc.com/blogs/sql-vs-nosql/>>. Acesso em: 31 out. 2019.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database Systems Concepts**. 6. ed. New York: McGraw-Hill, 2011.

SQL SERVER TUTORIAL. **SQL Server Sample Database**. 2018. Disponível em: <<http://www.sqlservertutorial.net/sql-server-sample-database/>>. Acesso em: 31 out. 2019.

TANEMBAUM, A. S. **Computer Networks**. 5. ed. Boston: Pearson Prentice Hall, 2011.

TEXAS INSTRUMENTS. **TL431 / TL432 Precision Programmable Reference**: Datasheet. Longmont, 2004. 83 p. Disponível em: <<https://www.ti.com/lit/ds/symlink/tl431.pdf>>. Acesso em: 9 ago. 2019.

TEXAS INSTRUMENTS. **LM3445/48 Phase Dimming Work Book**. Longmont, 2011. 65 p. Disponível em: <https://e2echina.ti.com/cfs-file/__key/communityserver-discussions-components-files/24/5707.Phase-Dimming-WB-_2800_LM3445_2600_LM3448_2900_.pdf>. Acesso em: 31 out. 2019.

TOCCI, R. J.; WIDNER, N. S.; MOSS, G. L. **Sistemas Digitais: Princípios e aplicações**. 11. ed. São Paulo: Pearson Prentice Hall, 2011.

YUAN, M. **Getting to know MQTT**. 2017. Disponível em: <<https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 3 abr. 2019.