

Victor Ferreira Martins

**AUTOMAÇÃO RESIDENCIAL USANDO PROTOCOLO MQTT, NODE-
RED E MOSQUITTO BROKER COM ESP32 E ESP8266**

UBERLÂNDIA

2019

Victor Ferreira Martins

AUTOMAÇÃO RESIDENCIAL USANDO PROTOCOLO MQTT, NODE-RED E MOSQUITTO BROKER COM ESP32 E ESP8266

Trabalho de Conclusão de Curso proposto ao Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia – UFU – Campus Santa Mônica, como requisito para obtenção do Título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia - UFU

Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Renato Santos Carrijo

UBERLÂNDIA

2019

Martins, Victor Ferreira

Automação residencial usando protocolo MQTT, Node-RED e Mosquitto Broker com ESP32 e ESP8266/ **Victor Ferreira Martins**. – **UBERLÂNDIA, 2019**- 53 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Renato Santos Carrijo

Trabalho de Conclusão de Curso - Universidade Federal de Uberlândia – UFU Faculdade de Engenharia Elétrica . **2019**.

Inclui bibliografia.

1. Automação residencial. 2. MQTT. 3. Internet das Coisas. I. Renato Santos Carrijo. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

AGRADECIMENTOS

Primeiramente, a Deus pela vida e pelo cuidado contínuo, em todos os momentos.

A meus pais e meu irmão pelo apoio em tudo que eu tive que enfrentar, além do restante da minha família, como meus tios e primos que tanto me ajudaram.

A todos os colegas que tanto me ajudaram durante o curso e tornaram os dias mais agradáveis. E também a todos os professores pela dedicação e paciência.

Por fim, agradeço também ao meu orientador pelo apoio e pelos vários conselhos que ajudaram no desenvolvimento deste trabalho.

RESUMO

A Internet das Coisas (IoT) procura analisar, controlar e integrar ambientes, sendo a automação residencial a aplicação da IoT na área residencial, com o propósito de trazer conforto, segurança e economia de energia para as residências. O objetivo deste trabalho é apresentar uma solução de automação residencial integrada com ferramentas open source que facilitem sua implementação.

Foi usado o protocolo de publicação/assinatura MQTT, com o Mosquitto Broker. Tal protocolo é capaz de permitir que os clientes da rede comuniquem entre si através da mediação do broker. Como clientes, foram utilizados os microcontroladores ESP32 e ESP8266. Foi utilizado também como cliente a ferramenta Node-RED, de forma que se comunicasse com o broker e criasse uma interface para o usuário.

Foi então montado o sistema, prevendo sua aplicação a dois cômodos: uma cozinha e um escritório. O sistema é capaz de monitorar dados de temperatura, umidade e estados de alarmes de segurança, além de permitir o acionamento desses alarmes e de saídas representadas por leds. Por fim, os resultados são mostrados na interface com o usuário, com telas específicas para cada cômodo.

Palavras-chave: Automação Residencial, Protocolo MQTT, ESP32, ESP8266, Internet das Coisas, Node-RED, Mosquitto, Broker.

ABSTRACT

The Internet of Things (IoT) aims to analyze, control and integrate environments, with home automation being the application of IoT in the residential area, with the purpose of bringing comfort, safety and energy saving to the houses. The objective of this work is to present an integrated home automation solution with open source tools that may simplify its implementation.

The publish/subscribe MQTT protocol was used with Mosquitto Broker. This protocol is capable of allowing clients in a network to communicate between themselves with the brokers mediation. As clients, the microcontrollers ESP32 and ESP8266 were used. The Node-RED tool was also used as a client, so that it may communicate with the broker and create a User Interface.

A system was then created, thinking of an application to two rooms: a kitchen and an office. The system is capable of monitoring temperature, humidity and the state of safety alarms, besides also allowing the actuation of these alarms and of outputs represented by leds. Finally, the results are shown in the user interface, with specific screens for each room.

Keyword: Home Automation, MQTT Protocol, ESP32, ESP8266, Internet of Things, Node-RED, Mosquitto, Broker.

LISTA DE FIGURAS

Figura 1 – Arquitetura do MQTT.....	16
Figura 2 – Exemplo de comunicação MQTT	17
Figura 3 – Exemplo de hierarquia por tópicos	18
Figura 4 – Layout do Node-RED	21
Figura 5 – Pinout da placa ESP-WROOM-32.....	24
Figura 6 – Pinout da placa Nodemcu ESP-12 com chip ESP8266.....	25
Figura 7 – Sensores DHT11 (esquerda) e DHT22 (direita)	26
Figura 8 – Sensor de movimento PIR	27
Figura 9 – Sensor de gás MQ-2	28
Figura 10 – Buzzer piezoelétrico	29
Figura 11 – Arduino IDE	29
Figura 12 – Arquitetura do projeto.....	30
Figura 13 – Configuração da Etapa 1.....	31
Figura 14 – Fluxo do Node-RED para Etapa 1	33
Figura 15 – Configuração da Etapa 2.....	34
Figura 16 – Fluxo do Node-RED para Etapa 2.....	36
Figura 17 – Configuração da Etapa 3.....	37
Figura 18 – Fluxo do Node-RED para Etapa 3.....	39
Figura 19 – Configuração da Alternativa	41
Figura 20 – Tela Inicial (Home) da UI do Node-RED	43
Figura 21 – Barra de opções da UI do Node-RED	44
Figura 22 – UI do Escritório.....	45
Figura 23 – Estados do alarme de movimento	46
Figura 24 – UI da Cozinha	47
Figura 25 – Estados do alarme de fumaça.....	48

LISTA DE TABELAS

Tabela 1 – Características dos níveis QoS do MQTT	19
Tabela 2 – Nós do Node-RED	22
Tabela 3 – Comparação entre recursos de microcontroladores	26
Tabela 4 – Comparação entre sensores DHT11 e DHT22	27
Tabela 5 – Inscrições e Publicações realizadas na Etapa 1	32
Tabela 6 – Inscrições e Publicações realizadas na Etapa 2	35
Tabela 7 – Inscrições e Publicações realizadas na Etapa 3	38
Tabela 8 – Inscrições e Publicações realizadas na Alternativa	42

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Conversor Analógico-Digital</i>
AMQP	<i>Advanced Message Queueing Protocol</i>
DAC	<i>Conversor Digital-Analógico</i>
GB	<i>Gigabytes</i>
GPIO	<i>General Purpose Input Output</i>
HTTP	<i>Hypertext Transfer Protocol</i>
Hz	<i>Hertz - Unidade de medida de frequência</i>
IoT	<i>Internet of Things, Internet das Coisas</i>
MQTT	<i>Message Queueing Telemetry Transfer</i>
QoS	<i>Quality of Service</i>
UI	<i>User Interface, Interface com o Usuário</i>
V2I	<i>Vehicle-to-Infrastructure</i>
V2V	<i>Vehicle-to-Vehicle</i>
XMPP	<i>Extensible Messaging and Presence Protocol</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa.....	12
1.2	Objetivos	13
2	REFERENCIAL TEÓRICO.....	14
2.1	Estado da Arte	14
2.2	Níveis de Automação Residencial	15
2.3	MQTT (Message Queuing Telemetry Transport).....	16
2.3.1	MQTT e outros protocolos.....	19
2.4	Node-RED.....	20
2.5	Mosquitto Broker.....	23
2.6	ESP32	23
2.7	ESP8266	24
2.8	Sensores de temperatura e umidade DHT11 e DHT22	26
2.9	Sensor de movimento PIR	27
2.10	Sensor de gás MQ-2	28
2.11	Buzzer piezoelétrico SFM27	28
2.12	Arduino IDE.....	29
3	METODOLOGIA	30
3.1	Etapa 1: Acionamento de lâmpadas e tomadas	31
3.2	Etapa 2: Controle de Temperatura de dois cômodos	34
3.3	Etapa 3: Adição do Sistema de Segurança	36
3.4	Alternativa: Separação dos Buzzers	40
4	RESULTADOS E DISCUSSÕES	43
4.1	Escritório.....	44
4.2	Cozinha	47
4.3	Alarmes	48
5	CONCLUSÃO	49
	REFERÊNCIAS.....	51

1 INTRODUÇÃO

Internet das coisas, do inglês Internet of Things, segundo Gubbi et al. (2013), é o conjunto de tecnologias que permite a análise, controle e integração dos mais variados ambientes, tanto fechados como abertos, através do trabalho conjunto de atuadores e sensores interligados em um grande sistema.

De acordo com Torres, Rocha e De Souza (2016), a Internet das coisas pode ser usada em aplicações de diversas áreas, como saúde, cidades inteligentes, escritórios, comércio e outros. Na área agrícola por exemplo, o estudo de estufas agrícolas controladas pode envolver técnicas avançadas de automação, onde é possível combater as condições adversas ao plantio de uma planta, como clima, umidade e condições do solo. Segundo Gao, Gao Xianwen e Chen (2014), países como Holanda e Israel têm enfrentado a escassez de recursos naturais com o uso de estufas controladas de alto nível.

Na área automotiva, a Internet das coisas pode atuar na comunicação entre veículos, utilizando tecnologias como Vehicle-to-Vehicle (V2V) e Vehicle-to-Infrastructure (V2I) permitindo que veículos comuniquem entre si e com os demais objetos utilizados na infraestrutura de estradas, conforme Dey et al. (2016).

Por fim, há também uma outra área da Internet das coisas conhecida como automação residencial, que é a sua aplicação na área residencial. Segundo Muratori e Dal Bó (2013), a definição de automação residencial surgiu no mercado brasileiro a partir do termo *Home Automation* utilizado nos Estados Unidos. Outro termo que pode ser utilizado é domótica, mais utilizado na Europa e que representa uma junção da palavra *domus* (do latim “casa”) e da palavra *robótica*.

Entre os principais benefícios trazidos pela automação residencial, estão o conforto, a segurança e a economia de energia. O conforto pode ser alcançado programando-se atividades do cotidiano, como apagar luzes e irrigar as plantas, para que funcionem automaticamente. Na área de segurança, é possível criar mais meios de informar ao morador sobre problemas e situações de risco (Muratori e Dal Bó, 2013).

Na área de economia, de acordo com Akram, Franco e Ismail (2016), mesmo pequenas atitudes, como aparelhos em modo de descanso ou iluminação acesa sem necessidade, podem se transformar em grandes desperdícios quando realizadas em maior escala. A automação residencial ajuda a combater tais desperdícios pela

construção de um sistema inteligente que seja capaz de administrar as atividades envolvendo energia em uma residência, de acordo com as necessidades do usuário.

Por fim, uma das escolhas que deve ser feita em cada projeto é qual protocolo de comunicação utilizar. Um dos protocolos que tem recebido destaque nos últimos anos para a área de Internet das Coisas é o protocolo MQTT, lançado pela IBM em 1999 e que possui um modelo de publicação/assinatura, do inglês *publish/subscribe* (Soni e Makwana, 2017).

1.1 Justificativa

Como justificativa para esse trabalho está o interesse crescente em tecnologias para automação residencial com o objetivo de oferecer cada vez mais qualidade de vida à população. De acordo com Silva e Gambarato (2016), atualmente já há expectativas de estilo de vida sendo criadas por uma população cada vez mais acostumada a viver com inovações tecnológicas, e essas expectativas estão se transformando em demandas para o ambiente residencial, onde se espera cada vez mais encontrar ambientes que possam ser facilmente gerenciáveis com conforto e segurança.

Segundo Muratori e Dal Bó (2013), nos últimos tempos novas situações tem se tornado comum em nossa sociedade que tornam o ambiente familiar ainda mais importante e presente no dia-a-dia das pessoas: trabalho em casa, estudo à distância, compras on-line, acesso a entretenimento e redes sociais são alguns exemplos de aplicações que tem aumentado o tempo que as pessoas passam em casa.

Além da pesquisa sobre automação residencial, o trabalho também visa estudar o protocolo de comunicação MQTT, ou Message Queuing Telemetry Transport, e sua aplicação na automação residencial. Segundo Roger A. et al (2017), o MQTT pode ser implementado em dispositivos de baixa potência, como microcontroladores, o tornando adequado para a área de Internet das Coisas e, portanto, para automação residencial. Soni e Makwana (2017) afirmam o mesmo, dizendo que o MQTT tem sido muito usado em Internet das Coisas por ter sido feito para funcionar como um protocolo leve para dispositivos de baixa potência.

Outra justificativa é a pesquisa dos microcontroladores ESP32 e ESP8266, que possuem preços acessíveis por unidade, porém com mais funcionalidades e potencial do que outros microcontroladores já populares no mercado. Segundo Maier, Sharp e

Vagapov (2017), uma das vantagens do ESP32 sobre outros microcontroladores no design de Internet das Coisas é o fato dele ser *open source*, permitindo que seja programado em diversas linguagens de programação.

1.2 Objetivo

O objetivo geral deste trabalho é propor uma solução de automação residencial que seja capaz de integrar três diferentes sistemas em uma casa: monitoramento de temperatura, alarmes de segurança e acionamento de iluminação e tomadas. O objetivo dessa integração é ajudar a alcançar três dos grandes objetivos da automação residencial: conforto, segurança e economia de energia. Os objetivos específicos são a utilização do protocolo MQTT para comunicação através do Mosquitto Broker, assim como o uso da ferramenta de programação visual Node-RED e dos microcontroladores ESP32 e ESP8266.

Para o sistema de temperatura, o objetivo específico é conseguir observar a temperatura e umidade do sistema e atuar sobre ela, de acordo com o desejo do usuário. Para o sistema de segurança, o objetivo é avisar o usuário na ocorrência de situações de risco. Por fim, o sistema de iluminação tem o propósito de permitir o controle de iluminação e tomadas, visando a economia de energia.

2 REFERENCIAL TEÓRICO

Nessa seção foram colocadas as referências teóricas para realização deste trabalho, desde trabalhos atuais relacionados com este, teoria de automação residencial e até explicações sobre o funcionamento do protocolo MQTT, dos microcontroladores ESP32 e ESP8266, da ferramenta de programação Node-RED, do Mosquitto Broker e dos sensores e atuadores utilizados no projeto.

2.1 Estado da Arte

Segundo Souza (2016), a automação residencial é uma área ainda carente de soluções práticas e acessíveis no mercado. Conforme indicou Muratori (2014), estudos da Associação Brasileira de Automação Residencial apontam que na época mais de 1,8 milhões de residências apresentavam potencial para a automatização, sendo que até 2013, apenas 300 mil delas tinham sua demanda atendida.

Em 2016, Souza propôs um projeto de automação residencial que já utilizava uma variedade de placas com microcontroladores: arduino uno, Arduino mega e esp8266. Porém, diferente da proposta deste trabalho, a comunicação era realizada através de radiofrequência, com a ajuda de módulos XBee.

Krishna e Lavanya (2017) propuseram um sistema de automação residencial controlado por comandos de voz, com a ajuda do programa BitVoicer. O objetivo do projeto era permitir não só aumentar o conforto e praticidade de todos, mas auxiliar na qualidade de vida de pessoas com deficiências ou idosos que não tem energia suficiente para se locomover e ligar/desligar aparelhos eletrônicos. Em 2014, Arul já havia proposto um sistema de automação com comandos de voz utilizando protocolo Zigbee. Seu objetivo era atender o mesmo grupo de pessoas, notando a necessidade de se investir no cuidado da faixa idosa da população, devido à tendência desse grupo de pessoas aumentar cada vez mais mundialmente, graças ao crescimento da média de expectativa de vida.

Alguns trabalhos similares a este já foram realizados. Upadhyay, Borole e Dileepan (2016) montaram o projeto de um sistema de automação residencial que utiliza o protocolo MQTT. Nele, foi utilizado uma Raspberry Pi como broker do sistema, tendo sensores conectados pelo protocolo Zigbee e fiação. O protocolo MQTT era

então utilizado para acessar os dados e intermediar a comunicação entre o usuário e os sensores, sendo a atuação do usuário feita através de uma webpage.

Quando se trata da utilização do ESP32 em Internet das Coisas, vários trabalhos foram feitos nos últimos anos. Allafi e Iqbal (2017) construíram um servidor web capaz de monitorar e coletar dados de um pequeno sistema de monitoramento de energia fotovoltaica. Já Biswas e Iqbal (2018) fizeram um sistema de bombeamento de água para países em desenvolvimento controlado pelo ESP32, onde o microcontrolador recebe dados de umidade, temperatura, umidade do solo e nível de água dos sensores e então o ESP32 aciona ou desliga uma bomba de água conforme os valores lidos.

O ESP32 também pode ser utilizado na área da saúde, como mostrado por Škraba et al. (2019), em um projeto em que é realizado o monitoramento da frequência cardíaca de um grupo de pessoas utilizando essa placa. Inicialmente o trabalho havia sido feito com uma das versões anteriores, o ESP8266, porém, foi escolhido o ESP32 devido à sua capacidade de comunicação bluetooth, e por conter múltiplas entradas analógicas, ao invés de apenas uma.

Na área de segurança, Rai e Rehman (2019) criaram um sistema de vigilância baseado no ESP32. Nele, com a ajuda de dispositivos auxiliares, o sistema é capaz de receber vídeo continuamente e transmiti-lo através de conexão Wi-Fi.

2.2 Níveis de Automação Residencial

Muratori e Dal Bó (2013) explicam que os diferentes sistemas de automação residencial podem ser divididos em três níveis, de acordo com suas particularidades: sistemas autônomos, integrados ou complexos.

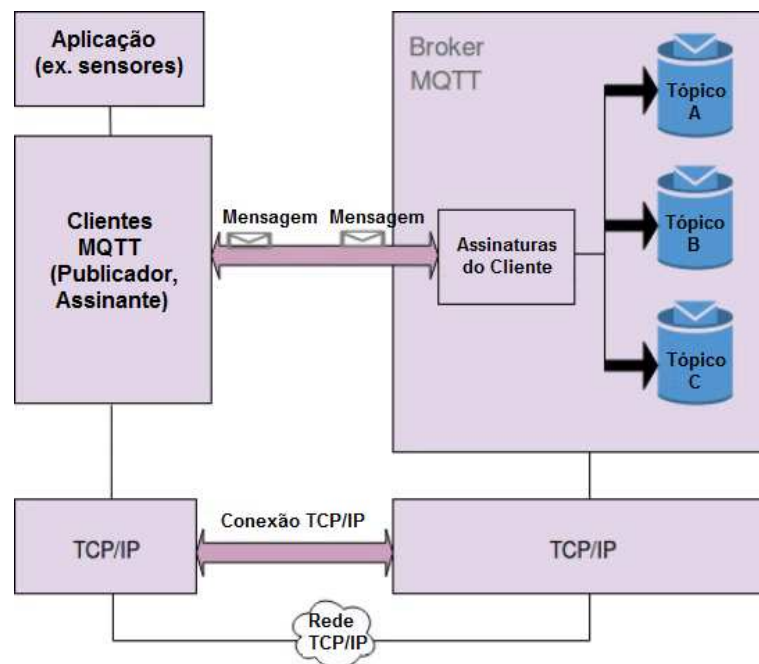
Sistemas autônomos são aqueles que operam sozinhos, sem interação com outros sistemas residenciais, ou sem dispositivos de controle centralizado, como portões automatizados, por exemplo. Sistemas integrados são aqueles na qual existe troca de informações entre os sistemas, de forma que sistemas autônomos podem agora se comunicar com outros sistemas da casa. Por exemplo, um alarme instalado em um sistema de segurança na sala de estar poderia ser acionado por causa de um vazamento de gás detectado por um sistema independente na cozinha. Por último, sistemas complexos são aqueles que utilizam tecnologias ainda em estágio de pesquisa e desenvolvimento, precisando ainda serem aceitos pelo público.

2.3 MQTT (Message Queuing Telemetry Transport)

Message Queuing Telemetry Transport, ou apenas MQTT, é um protocolo de comunicação que utiliza o modelo de mensagens de publicação/assinatura (do inglês *publish/subscribe*) por tópicos, feito para aplicações com dispositivos restritos, que são aqueles com recursos como CPU, memória e potência limitados. Foi projetado para transportar informações de forma leve e flexível. Ele é ideal para aplicações de Internet das Coisas devido à sua flexibilidade para se adaptar a diversos tipos de situações, além de sua leveza permitir que seja implementado mesmo em redes com largura de banda limitada (Yuan, 2017).

O princípio de funcionamento do modelo de publicação/assinatura é que os dispositivos se comunicam por meio de assinaturas. Quando um dispositivo deseja receber informações ele deve se tornar um assinante (*subscriber*). Para enviar informações ele deve publicá-las se tornando um publicador (*publisher*). O modelo prevê dois membros na rede: o *broker* e os clientes, sendo que o broker é o dispositivo responsável por intermediar a comunicação, recebendo as publicações e enviando aos seus devidos assinantes. Já o cliente é qualquer dispositivo que seja capaz de realizar essa interação com o broker, enviando e recebendo mensagens, como um microcontrolador, por exemplo (Yuan, 2017).

Figura 1 - Arquitetura do MQTT

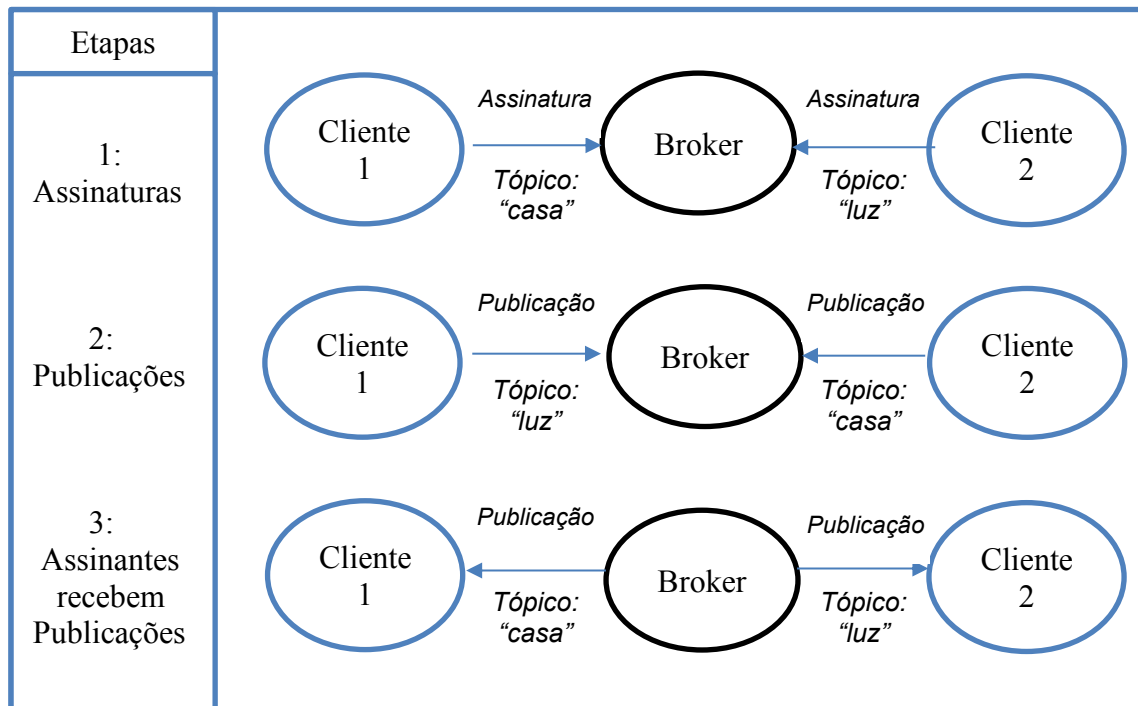


Fonte: Adaptado de (Soni e Makwana, 2017)

A arquitetura do protocolo é mostrada na figura 1. Os clientes informam de quais tópicos desejam se tornar assinantes e passam a receber as publicações realizadas nesses tópicos. A figura também mostra que o protocolo MQTT é baseado no protocolo TCP/IP, sendo o MQTT o protocolo da camada de aplicação, responsável apenas por cuidar do formato da mensagem. Portanto, para que as mensagens sejam enviadas, é preciso que seja feita uma conexão TCP/IP.

Segundo Hunkeler, Truong e Stanford-Clark (2008), há três tipos principais de modelo de publicação/assinatura: por tópico, por tipo e por conteúdo. O MQTT utiliza um modelo por tópico, onde os publicadores enviam mensagens em tópicos, enquanto os assinantes informam ao broker o tópico que desejam assinar. A partir de então toda mensagem que for publicada em um tópico será enviada para o broker que as direcionará para todos os assinantes daquele tópico e apenas para eles, conforme a figura 2, que mostra que a mensagem publicada no tópico “luz” pelo cliente 1 foi redirecionada pelo broker apenas para o cliente 2, por ele ser assinante desse tópico, da mesma forma que a mensagem publicada no tópico “casa” pelo cliente 2 foi direcionada apenas para o assinante, cliente 1.

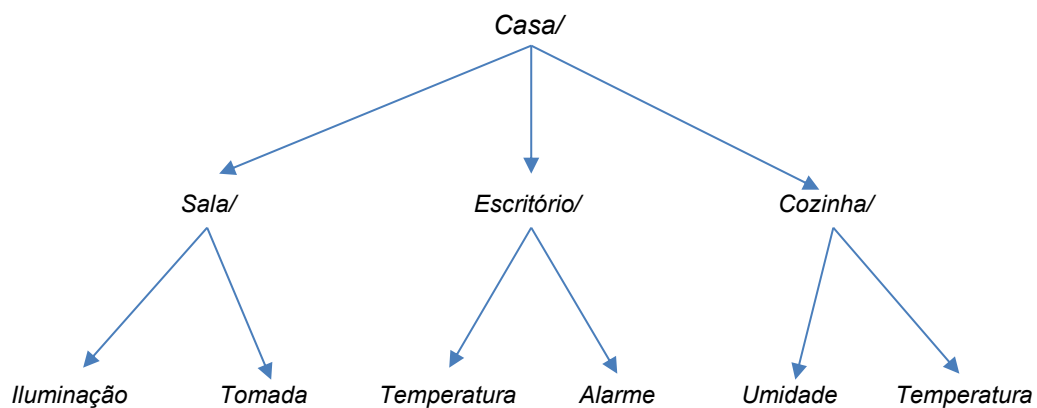
Figura 2 – Exemplo de Comunicação MQTT



Em um modelo de publicação/assinatura, um determinado sensor ou atuador pode exercer tanto a função de publicador como de assinante, como evidencia a figura 2, onde tanto o cliente 1 como o cliente 2 assinam tópicos e realizam publicações. Além disso, os assinantes podem ser alterados sem que um determinado publicador saiba, já que a comunicação é intermediada pelo broker. (Soni e Makwana, 2017).

A comunicação no MQTT funciona através de níveis hierárquicos de tópicos, separados através de barras invertidas “/”, conforme figura 3. Por exemplo, caso um dispositivo queria publicar uma mensagem indicando a temperatura atual do escritório lida por um sensor, ele poderia enviar uma mensagem para o tópico “Casa/Escritório/Temperatura”. Dois caracteres especiais, “+” e “#” podem ser utilizados para representar um grupo de dados. Por exemplo, o tópico hierárquico de “Casa/+/Temperatura” poderia ser utilizado para ler os dados de temperatura de todos os ambientes da casa. No caso da figura 3, ele se tornaria assinante tanto da temperatura do escritório, como da cozinha. Já o tópico “Casa/#” poderia ser usado para receber as informações de todos os níveis hierárquicos abaixo do nível “Casa” (Hunkeler, Truong e Stanford-Clark 2008).

Figura 3 – Exemplo de hierarquia por tópicos



Com relação à segurança dos dados, o MQTT provê três níveis de *Quality of Service (QoS)*, sendo este responsável por gerenciar a qualidade do tráfego de dados na rede. A escolha entre níveis depende da necessidade de cada aplicação. Como colocado na tabela 1, o nível 0 é o mais simples: o serviço não fornece garantias sobre a entrega da mensagem e não há retransmissão ou confirmação de recebimento. O

nível 1 já fornece um pouco mais de segurança: as mensagens são transmitidas até serem confirmadas pelos receptores, o que significa que as mensagens certamente chegarão, porém com o risco de chegarem múltiplas vezes. O nível 2 vai além do nível 1 e, além de garantir o recebimento das mensagens, garante que elas serão entregues apenas uma vez ao destinatário (Hunkeler, Truong e Stanford-Clark, 2008 e Soni e Makwana, 2017).

Tabela 1 - Características de Níveis de Segurança QoS do MQTT

Quality of Service (QoS)	Retransmissão	Confirmação de recebimento	Risco de múltiplas entregas
N0	Não	Não	Não
N1	Sim	Sim	Sim
N2	Sim	Sim	Não

2.3.1 MQTT e outros protocolos

As vantagens do MQTT para aplicação em IoT com pequenos dispositivos se mostram quando ele é comparado a outros protocolos. A Internet das Coisas lida com objetos restritos quanto a suas funções, atuando em ambientes com muitos dispositivos, sendo estes muitas vezes com alta latência e baixa confiabilidade. Essas características tornam alguns protocolos de comunicação mais adequados que outros.

De acordo com Serozhenko (2017), o protocolo HTTP (*Hypertext Transfer Protocol*) é o protocolo mais utilizado por desenvolvedores em várias áreas, porém seu uso não é recomendado em IoT, por várias razões. Ele é um protocolo pesado, com informações suficientes para ser compreendido por uma pessoa, porém isso não é necessário em sistemas com recursos restritos como IoT, onde é preferível um protocolo leve como o MQTT. Em termos de velocidade e segurança, o MQTT também é superior, apresentando taxa de transferência de dados muito mais rápida, além de possuir garantias de entrega dos dados que o HTTP não possui.

Segundo Soni e Makwana (2017), é preferível utilizar protocolos do tipo push em IoT, ao invés de protocolos *pull*. Em sistemas corporativos, um protocolo push muito utilizado é o AQMP (Advanced Message Queuing Protocol), por apresentar confiabilidade e interoperabilidade. Porém, conforme Yuan (2017), ele foi projetado

para atuar em sistemas ricos em recursos e não em sistemas com restrições de recursos, como em IoT. O mesmo acontece com o protocolo XMPP (Extensible Messaging and Presence Protocol), que requer muito mais recursos do que se espera em uma rede de IoT.

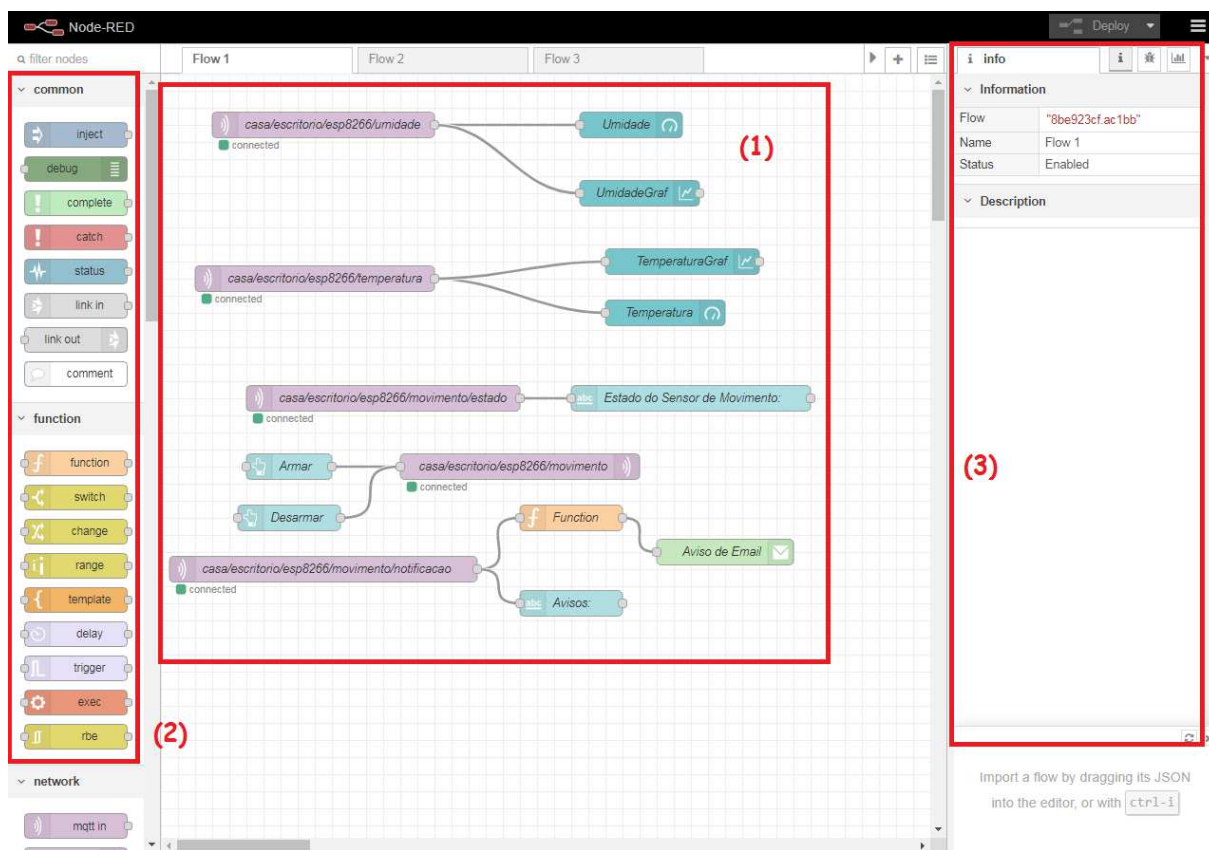
2.4 Node-RED

Node-RED é um software de código aberto que simplifica o processo de programação em aplicações de Internet das Coisas, através de uma programação visual, onde o usuário pode conectar blocos de código, chamados de nós, do inglês *nodes*, para que realizem uma determinada atividade. Quando os nós se juntam para realizar uma atividade, eles formam um fluxo, do inglês *flow*. A tabela 2 mostra os nós que foram usados neste trabalho, com uma breve descrição da função de cada.

Em comunicação MQTT, o Node-Red pode ser executado no dispositivo que for atuar como broker (como um computador, ou uma Raspberry Pi) de forma a mediar a comunicação deste com os demais dispositivos da rede.

O Node-RED também é capaz de criar uma interface gráfica para mediar a comunicação entre o usuário e o sistema, armazenar e coletar dados de bancos de dados, criar eventos acionados por tempo, coletar dados da web e muito mais.

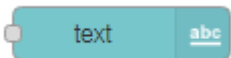
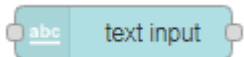
Figura 4 – Layout do Node-RED



O espaço onde os fluxos de nós são construídos é indicado por (1) na figura 4. A barra lateral esquerda indicada por (2) apresenta uma lista de nós que podem ser utilizados. É possível instalar mais nós além daqueles já presentes com o Node-Red padrão. Para esse trabalho, o único nó instalado a mais foi o nó de email, mostrado na tabela 2. Por último, a barra lateral (3) apresenta três abas: uma aba de informações sobre a seleção atual do usuário, outra de debug para auxiliar no processo de depuração de erros, e por último uma barra de configuração do dashboard.

Tabela 2 – Nós do Node-RED

Nó	Função
	Adiciona um botão à UI que, ao ser acionado, envia uma mensagem em sua saída.
	Plota o gráfico dos valores de entrada.
	Mostra algumas propriedades de uma mensagem na barra lateral de debug.
	Envia um email com os dados da entrada.
	Permite criar um bloco de função JavaScript que pode usar os dados de entrada para produzir uma saída.
	Adiciona um medidor visual à UI com os valores de entrada.
	Se conecta à um broker MQTT e se torna assinante do tópico especificado.
	Se conecta à um broker MQTT e realiza uma publicação no tópico especificado.
	Adiciona uma chave Liga/Desliga à UI.

	Exibe um texto não-editável na UI.
	Adiciona um campo de entrada de texto à UI.

2.5 Mosquitto Broker

O Mosquitto Broker é um broker de código aberto capaz de implementar o protocolo MQTT em uma máquina, sendo compatível com várias plataformas como Windows, Linux e mesmo Raspberry Pi. Mosquitto foi feito para ser usado em situações que precisem de comunicação leve, especialmente com dispositivos restritos, como muitas vezes é o caso em IoT (Light et al, 2017).

Torres, Rocha e De Souza (2016) realizaram um trabalho de comparação entre brokers de comunicação MQTT. A implementação do Mosquitto em linguagem C apresentou a menor carga de processamento e consumo de memória, confirmando sua característica de ser um broker leve. Porém, não obteve o melhor comportamento em termos de vazão de mensagens.

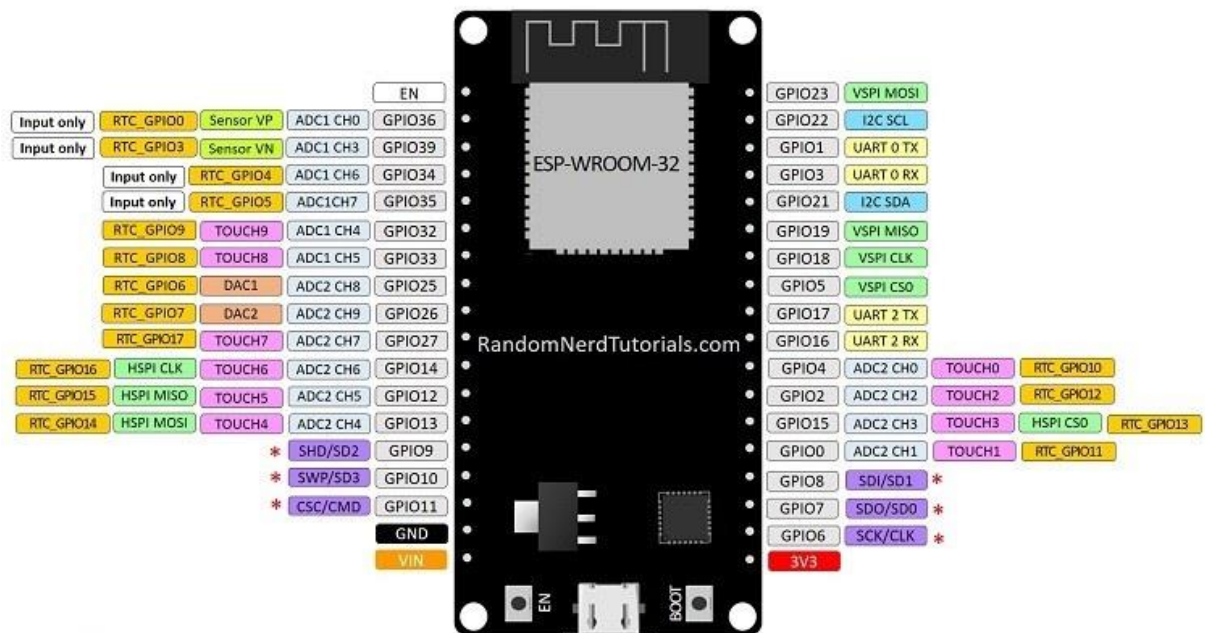
2.6 ESP32

O nome ESP32 se refere a um chip com função de microcontrolador produzido pela Espressif para trabalhar com dispositivos móveis e aplicações de Internet das Coisas. O ESP32 é capaz de funcionar em ambientes com temperatura na faixa de -40°C até 125°C. O módulo contém Wi-Fi e Bluetooth integrados, podendo funcionar como um sistema *standalone* ou como escravo de outro dispositivo. Conta com um processador de 32 bits (Espressif, 2019).

Como mostrado na tabela 3, apresenta vantagens em suas especificações quando comparado a outras placas com microcontroladores do mercado, tem como diferencial a presença de Bluetooth, muito mais GPIOs e ADCs, além da presença de 2 DACs e frequência de clock de 160 MHz.

O ESP-WROOM-32 é uma placa que contém o módulo ESP32, conforme ilustra a figura 5, onde é possível notar suas várias funcionalidades distribuídas em sua versão com 36 pinos. Possui pinos para interface GPIO, UART, I2C, sensores de toque, sensor de Efeito Hall, conversores AD e DA, e outros.

Figura 5 - Pinout da placa ESP-WROOM-32



Fonte: Adaptado de (RandomNerdTutorials, 2019)

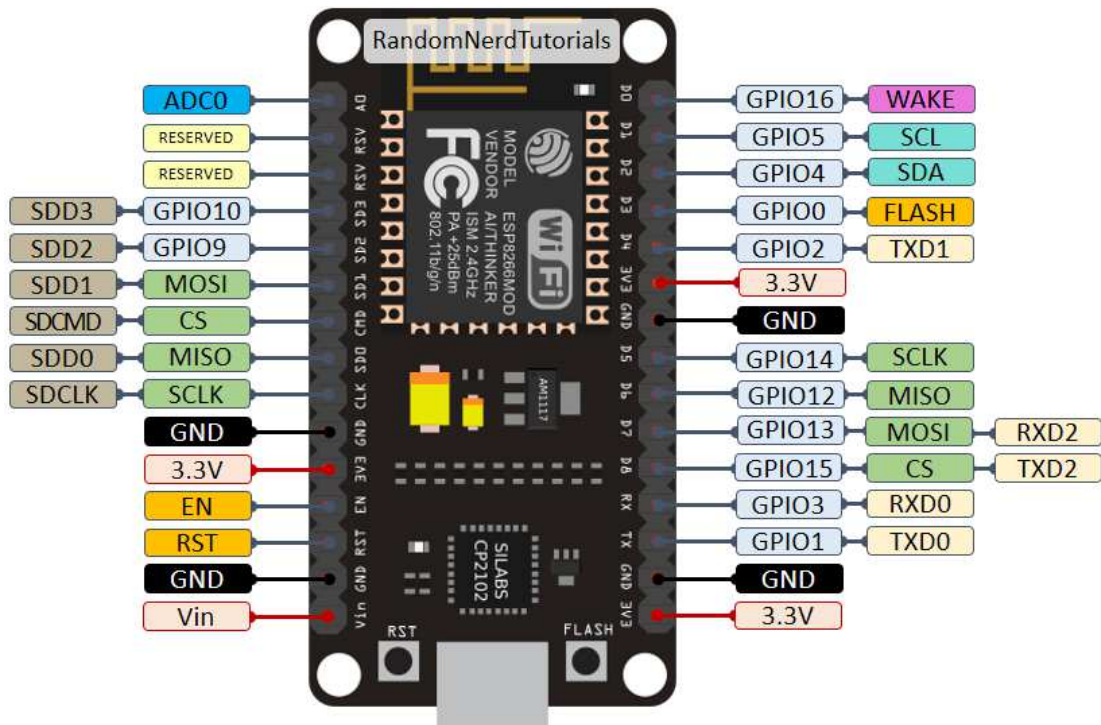
2.7 ESP8266

O nome ESP8266 se refere a um chip desenvolvido para Expressif para funcionar também como microcontrolador. Trata-se de uma solução altamente integrada para atender a demanda da área de Internet das Coisas por dispositivos compactos, eficientes e que ainda assim sejam confiáveis em termos de performance (Espressif, 2019). Uma das placas que contém esse chip é o Nodemcu ESP-12, cujo esquema pinout está representado na figura 6.

Uma de suas principais características é o poder de realizar conexões Wi-Fi, não apenas por si próprio, mas também servindo de adaptador para que outras placas realizem esse tipo de conexão. Apresenta um processador de 32 bits, interface

periférica para conexão a outros dispositivos por meio de GPIOs, UART, SPI, I2C, I2S e outros, conforme figura. A tabela 3 mostra que, apesar de ser inferior ao ESP32, ainda é muito superior em suas especificações a outra placa popular disponível no mercado, o Arduino Uno. De Oliveira (2017) menciona em seu livro que a ESP8266 tem bom potencial na área de IoT por causa de sua relação entre preço e recursos.

Figura 6 - Pinout da placa Nodemcu ESP8266



Fonte: Adaptado de (RandomNerdTutorials, 2019)

Tabela 3 - Comparação entre recursos dos microcontroladores

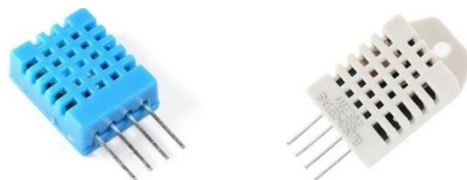
	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Fonte: Adaptado de (Fernando K, 2017)

2.8 Sensores de temperatura e umidade DHT11 e DHT22

Os sensores de umidade e temperatura DHT11 e DHT22, mostrados na figura 7, são sensores básicos e de baixo custo. São formados por um sensor de umidade capacitivo e um termistor, além de um chip que permite conversão do sinal analógico de temperatura ou umidade para um sinal digital, que pode então ser enviado a um microcontrolador. As diferenças entre os dois sensores são mostradas na tabela 4, sendo o DHT11 mais barato, porém menos preciso que o DHT22 (Adafruit, 2019).

Figura 7 - Sensores DHT11 (Esquerda) e DHT22 (Direita)



Fonte: Adaptado de (Filipe Flop, 2019c e 2019g)

Tabela 4 - Comparação entre sensores DHT11 e DHT22

	DHT11	DHT22
Custo	Muito baixo	Baixo
Energização	3 a 5V	3 a 5V
Corrente Máxima	2,5 mA	2,5 mA
Faixa de Umidade Ideal	20 a 80%	0 a 100%
Faixa de Temperatura Ideal	0 a 50 °C	-40 a 80°C
Tamanho	15,5 x 12 x 5,5 (mm)	15,1 x 25 x 7,7 (mm)
Taxa de Amostramento	1 Hz	0,5 Hz

Fonte: Autoria própria com dados de (Adafruit, 2019)

2.9 Sensor de movimento PIR

O sensor de movimento PIR, mostrado na figura 8, permite detectar movimentos, podendo ser usado em aplicações de segurança de automação residencial para verificar se há presença de pessoas em determinado cômodo. Pequeno, barato e fácil de usar, este sensor é feito de um sensor piroelétrico, que detecta radiação infravermelha, que é emitida por tudo, tanto objetos como pessoas, dependendo do calor emitido.

O sensor é dividido em duas metades, e se uma delas medir uma radiação diferente da outra, a saída é acionada. A sensibilidade da distância alcançada pelo sensor e o atraso no intervalo entre cada detecção podem ser alterados fisicamente no sensor, conforme figura, sendo capaz de detectar movimento em até 6 metros por 70° de faixa. Junto com o sensor está um chip BISS0001, responsável por receber o sinal analógico do sensor e transformá-lo em um sinal digital, funcionando, portanto, como uma entrada digital (Adafruit, 2018).

Figura 8 - Sensor de movimento PIR



Fonte: Adaptado de (Filipe Flop, 2019e)

2.10 Sensor de gás MQ-2

O sensor de gás MQ-2 mostrado na figura 9 é um sensor que reage a gases combustíveis e fumaça, apresentando sensibilidade particularmente alta a GLP, Propano e Hidrogênio. É formado por um micro tubo cerâmico, uma camada de dióxido de estanho sensível a gás, um eletrodo de medição e uma mola de aquecimento feita de uma liga de Níquel e Cromo. Possui tanto um pino de saída analógica, como um pino de saída digital que apenas indica se há ou não a presença de gás ou fumaça. Seu pequeno tamanho, baixo custo e baixa tensão de entrada o tornam adequado para uso em projetos de automação residencial. Configurado para detectar gases combustíveis entre 300 e 1000ppm de concentração (Pololu, 2019).

Figura 9 - Sensor de gás MQ-2



Fonte: Adaptado de (Filipe Flop, 2019d)

2.11 Buzzer piezoelétrico SFM27

O buzzer piezoelétrico SFM27 mostrado na figura 10 é um buzzer que opera entre tensões de 3 a 24 volts, reproduzindo um som maior ou igual a 95 dB operando em 12VDC/10cm. Projetado para operar em temperaturas na faixa de -20°C a 60°C. Tais características somadas à seu pequeno tamanho e baixo custo o tornam adequados à utilização em projetos de domótica (Open Circuit, 2019).

Figura 10 - Buzzer Piezoelétrico

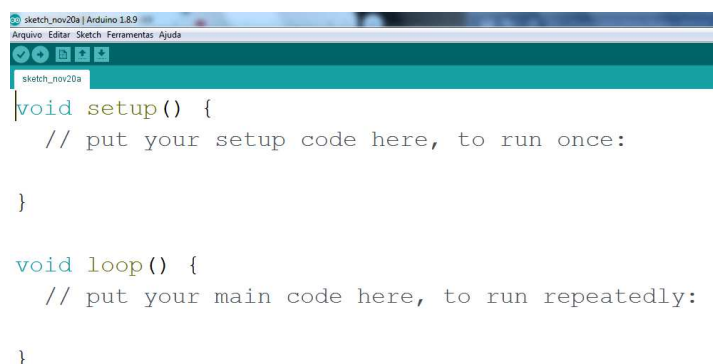


Fonte: Adaptado de (Filipe Flop, 2019f)

2.12 Arduino IDE

Arduino IDE é uma ferramenta que pode ser utilizada para compilação de programas a serem enviados para microcontroladores. Apesar do nome, ela tem compatibilidade com vários tipos de placas com microcontroladores, incluindo placas com ESP8266 e ESP32, desde que seja feito o download das bibliotecas que dão suporte à placa respectiva. O código a ser compilado deve encaixar em duas funções pré-definidas pela Arduino IDE, chamadas setup e loop, mostradas na figura 11. Enquanto a função setup é executada apenas uma vez, servindo para configurações iniciais, a função loop será executada repetidas vezes.

Figura 11 – Arduino IDE



```
sketch_nov20a | Arduino 1.8.9
Arquivo Editar Sketch Ferramentas Ajuda
sketch_nov20a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

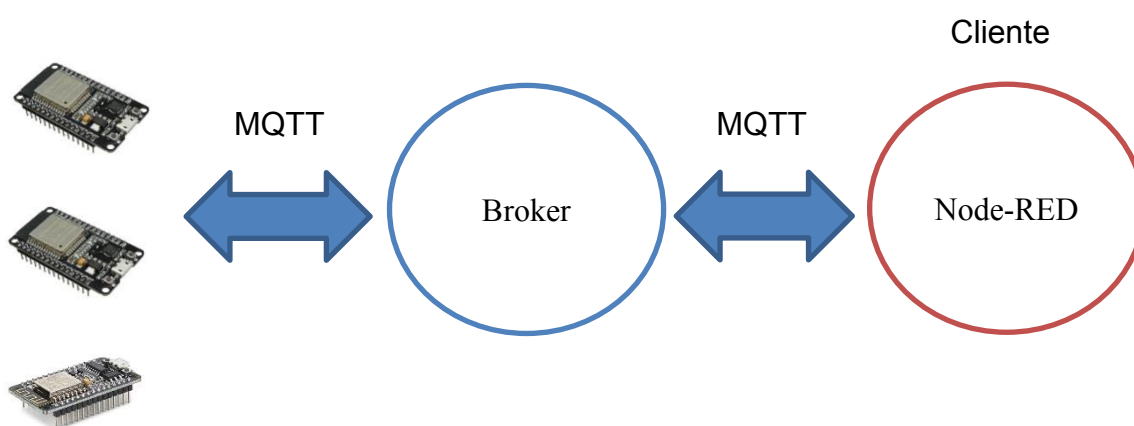
3 METODOLOGIA

Foi realizado o projeto de uma solução de automação residencial utilizando comunicação por protocolo MQTT com o uso das ferramentas Node-Red e Mosquitto Broker, além dos microcontroladores ESP32 e ESP8266.

A arquitetura do projeto é apresentada na figura 12. A automação residencial em si tratou-se da implementação de sistemas de monitoramento de temperatura, acionamento de lâmpadas e de segurança contra situações de risco. Esses sistemas foram distribuídos por três cômodos: uma cozinha, um escritório e uma sala de estar. Eles são controlados por microcontroladores que se comunicam com sensores e atuadores. Os sistemas funcionam como clientes, assim como o Node-RED, enquanto o Mosquitto Broker irá intermediar a comunicação entre esses clientes, de forma que toda mensagem deve passar por ele, seguindo o modelo de publicação/assinatura do protocolo MQTT.

Foi escolhido o nível QoS 2 para todas as mensagens transmitidas por protocolo MQTT, pois espera-se que serviços de automação residencial muitas vezes trabalhem com redes sem muita confiabilidade e este sistema não foi projetado para lidar com mensagens repetidas. Portanto, o nível QoS 2 ajuda a garantir que as mensagens serão entregues corretamente e apenas uma vez, ajudando a garantir o bom funcionamento de todo o sistema.

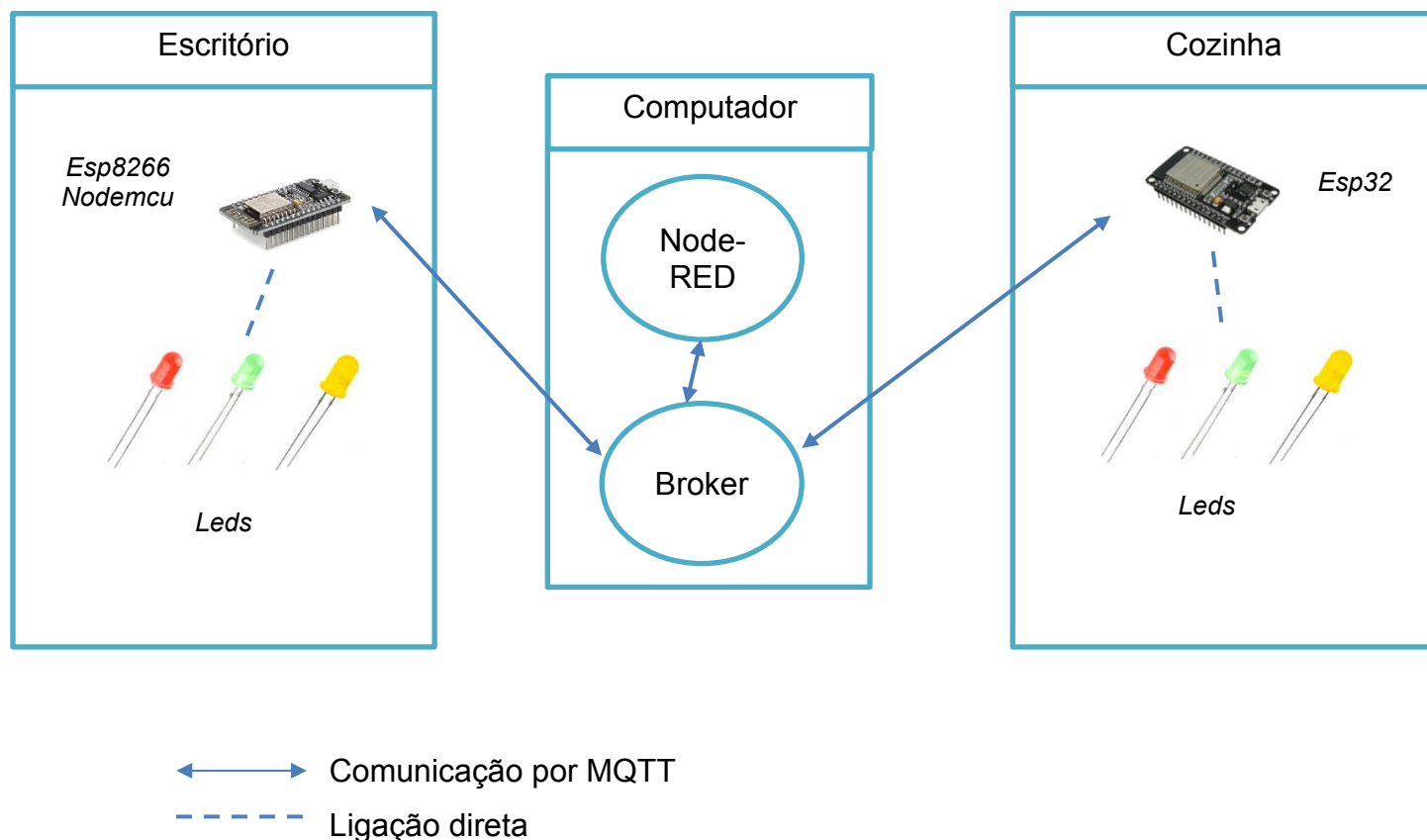
Figura 12 – Arquitetura do projeto



Fonte: Imagens de dispositivos adaptadas de (FILIPE FLOP, 2019)

3.1 Etapa 1: Acionamento de lâmpadas e tomadas

Figura 13 – Configuração da Etapa 1



Fonte: Imagens de dispositivos adaptadas de (FILIPE FLOP, 2019)

Para simular o acionamento de lâmpadas ou mesmo tomadas, foi montado a configuração da figura 13, onde foram utilizados dois leds em cada cômodo. Na cozinha foi utilizado um microcontrolador ESP32 em uma placa ESP-WROOM-32, enquanto no escritório foi usado uma microcontrolador ESP8266 em uma placa Nodemcu ESP8266.

No ESP32 foi carregado um programa através da Arduino IDE, no qual o dispositivo se torna assinante dos tópicos mostrados na tabela 5. Como resultado, ao receber mensagens de um desses tópicos, a placa irá escrever um valor digital alto ou baixo no led respectivo, de acordo com a mensagem presente no sinal recebido. Como a ligação entre a placa e os leds é direta, não há necessidade de realizar publicações.

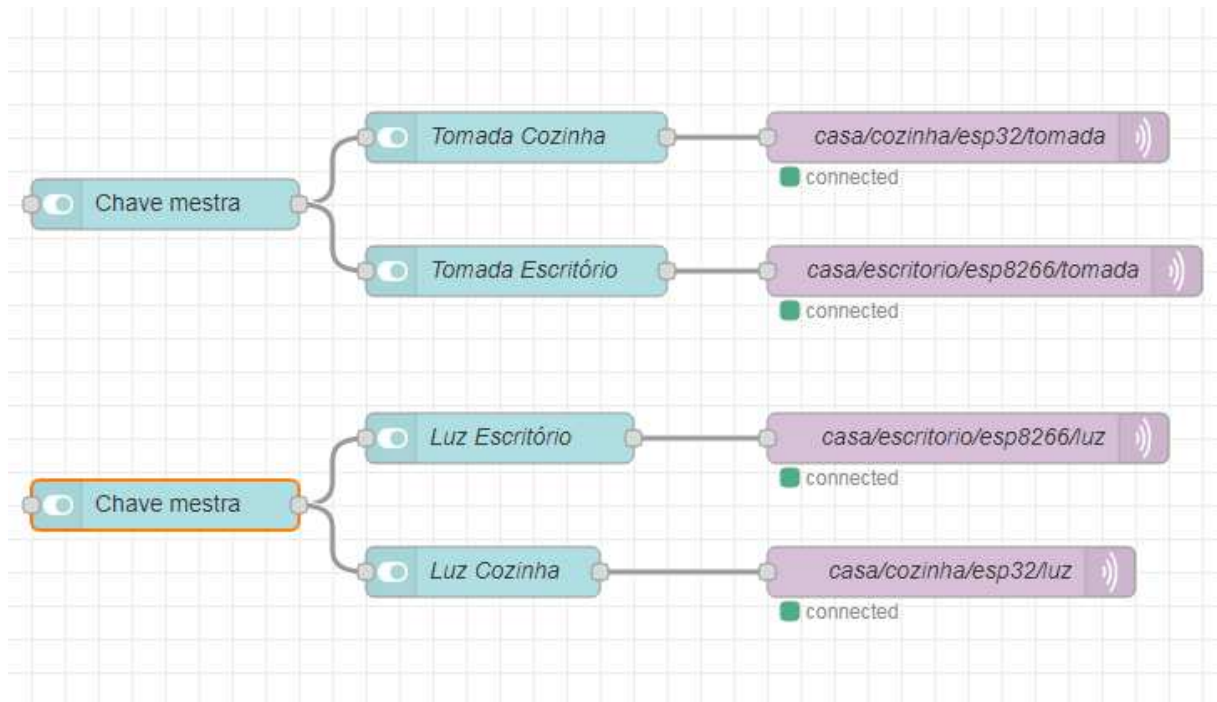
Tabela 5 – Inscrições e Publicações realizadas na Etapa 1

Dispositivo	Inscrições	Publicações
ESP8266 (Escritório)	casa/escritório/esp8266/ tomada	Nenhuma
	casa/escritório/esp8266/luz	
ESP32 (Cozinha)	casa/cozinha/esp32/tomada	Nenhuma
	casa/cozinha/esp32/luz	

O programa carregado no ESP32 também foi responsável por realizar a conexão Wi-Fi do microcontrolador com o Wi-Fi da casa. O próximo passo foi configurar o *broker* e a *User Interface (UI)*. Para tal, foram utilizadas duas ferramentas: o Mosquitto Broker e o Node-RED, em um sistema operacional Windows 7, rodando em um computador de mesa com memória RAM de 2 GB, e um processador Intel Core Duo.

Com o Node-RED e o Mosquitto Broker rodando no sistema através dos comandos “node-red” e “mosquitto -v” realizados no prompt de comando, o Node-RED se conecta automaticamente ao Broker e a UI do Node-RED pode agora ser acessada por um browser através do endereço local do computador. Para acesso à essa UI, foi instalado no Node-RED o *node-red-dashboard*, que será responsável por servir de interface, onde será possível visualizar em um dashboard os dados do sistema e atuar sobre ele, de acordo com a configuração de fluxos que tiver sido realizada e carregada previamente no Node-RED.

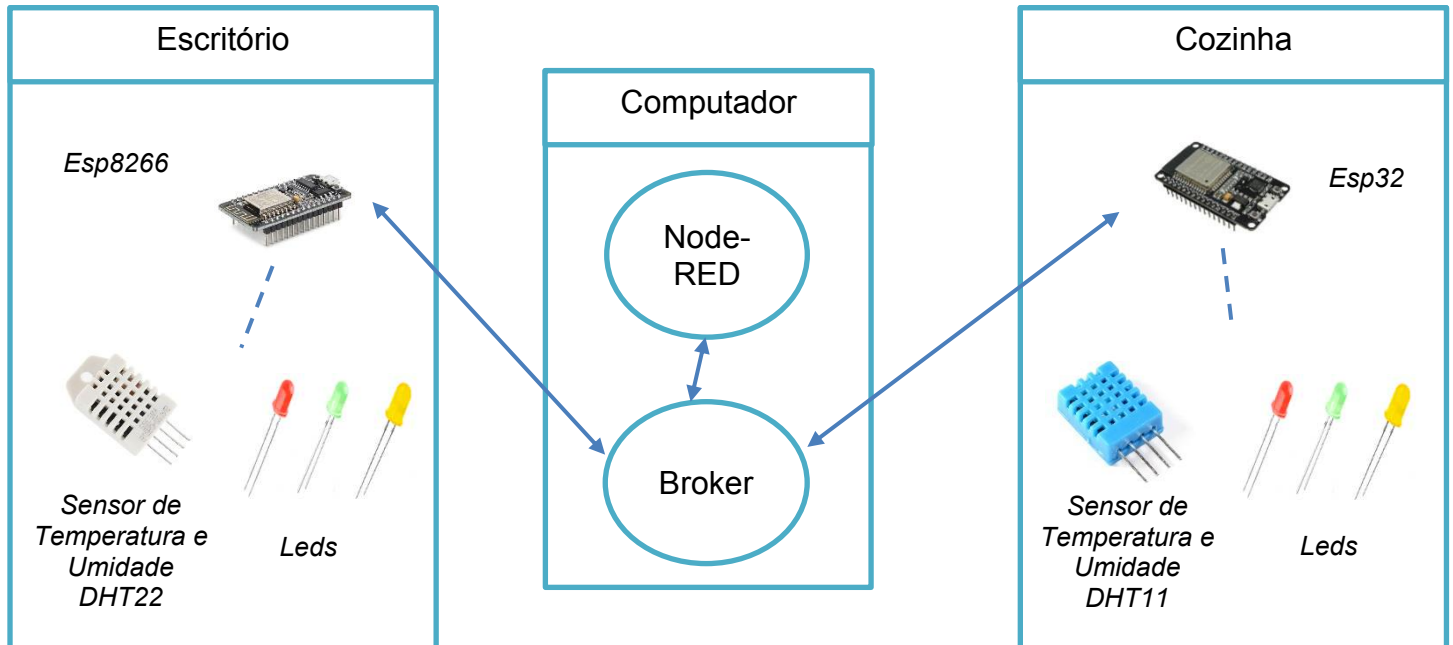
Figura 14 – Fluxo do Node-RED para a Etapa 1





Em seguida foi construído o fluxo da figura 14 no Node-Red, de forma que foram conectados quatro nós de botões a quatro nós de saída MQTT, sendo esses botões configurados para enviar um sinal “on” enquanto estiverem pressionados e um sinal “off” caso contrário, sendo este sinal enviado para o tópico MQTT ao qual o botão está conectado. O objetivo é que cada botão acenda um led diferente, sendo que dois botões representam, uma tomada e uma lâmpada na cozinha e outros dois representam os mesmos itens em um escritório. Finalmente será conectado um nó de botão chamado de “Chave Mestra” aos botões de tomadas e outro aos de luz, com a função de permitir que o usuário acione conjuntos de botões ao mesmo tempo. Os botões do escritório foram configurados para aparecer na página da UI que representa o escritório, enquanto os da cozinha foram configurados para aparecer na página da cozinha. Já as chaves mestras foram colocadas na página inicial da UI.

3.2 Etapa 2: Controle de Temperatura de dois cômodos

Figura 15 – Configuração da Etapa 2



 Comunicação por MQTT
 Ligação direta

Fonte: Imagens de dispositivos adaptadas de (FILIPE FLOP, 2019)

Será agora adicionado à primeira etapa o controle de temperatura dos dois cômodos, conforme figura 15. Para o escritório, foi utilizado um sensor DHT22 e para a cozinha um sensor DHT11. Em ambos os casos, os leds da primeira etapa permaneceram conectados.

Com o circuito montado, foi feito um programa na Arduino IDE com a intenção de estabelecer a comunicação MQTT em cada dispositivo, estabelecendo uma conexão Wi-Fi para cada dispositivo e especificando a quais tópicos cada um irá assinar, e em quais tópicos ele irá publicar informações.

Como o objetivo é ler as temperaturas do escritório e da cozinha, será preciso apenas que os microcontroladores, ESP8266 e ESP32, publiquem os valores de temperatura e umidade em tópicos que possam ser lidos pelo Node-RED, para depois

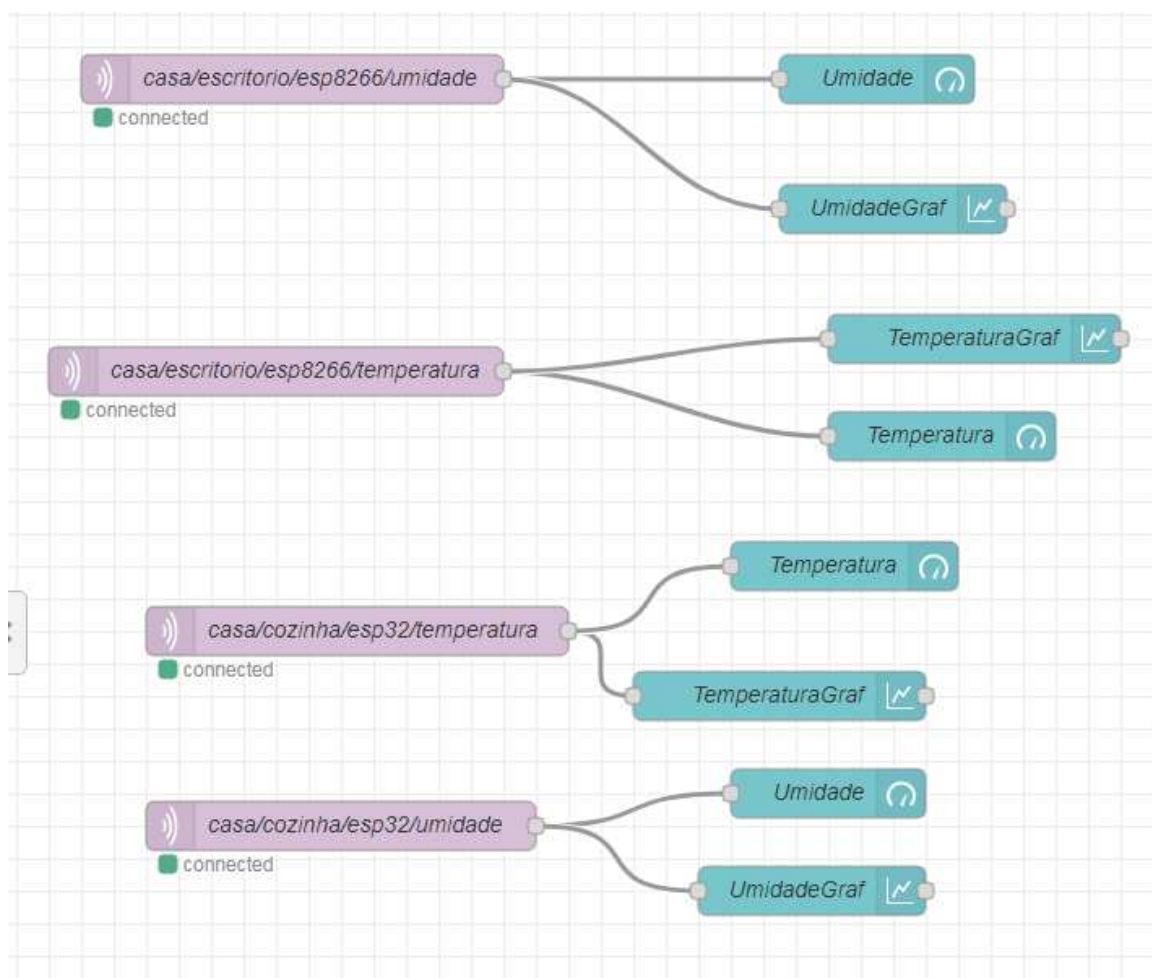
serem exibidos na UI. Em outras palavras, os novos dispositivos funcionarão agora também como publicadores. Com base nisso, a tabela 6 mostra a relação entre os microcontroladores e tópicos, com as publicações de temperatura e umidade do escritório e da cozinha.

Tabela 6 – Inscrições e Publicações realizadas na Etapa 2

Cliente	Inscrições	Publicações
ESP8266 (Escritório)	casa/escritório/esp8266/ tomada	casa/escritório/esp8266/ temperatura
	casa/escritório/esp8266/ luz	casa/escritório/esp8266/ umidade
ESP32 (Cozinha)	casa/cozinha/esp32/ tomada	casa/cozinha/esp32/ temperatura
	casa/cozinha/esp32/luz	casa/cozinha/esp32/umidade

A figura 16 mostra o fluxo montado para exibição da temperatura e da umidade na UI. Quatro nós de entrada MQTT, que funcionam como assinantes de tópicos, receberão publicações de temperatura e umidade correspondentes a cada cômodo. Os microcontroladores foram programados para publicar os valores de temperatura e umidade nesses tópicos. Os nós são ligados a outros dois nós cada, para que os valores recebidos sejam exibidos na UI do Node-RED através tanto de indicadores como de gráficos.

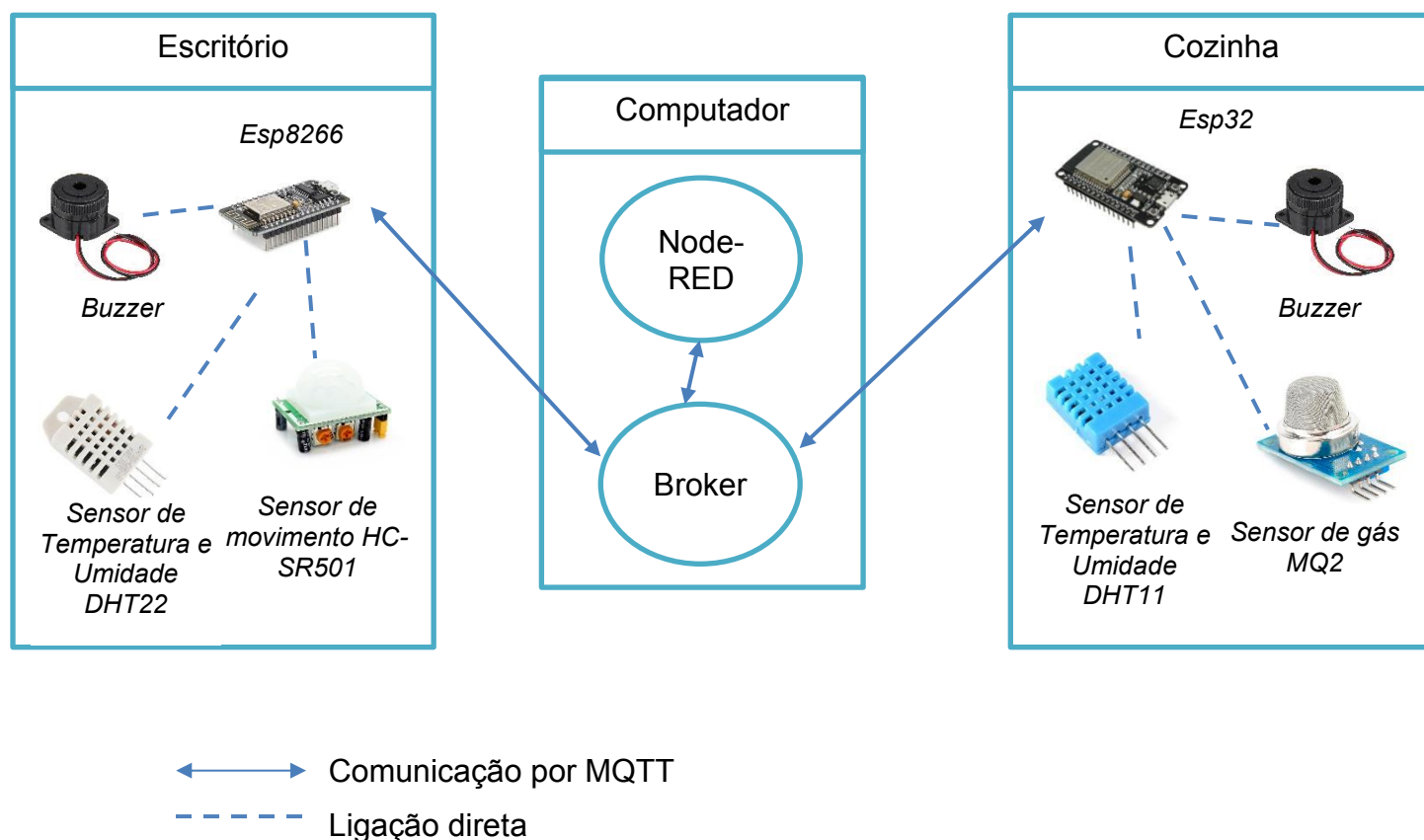
Figura 16 – Fluxo do Node-RED para a Etapa 2



3.3 Etapa 3: Adição do Sistema de Segurança

Como melhorar a segurança de moradores é um dos objetivos da automação residencial, nessa etapa serão inseridos alguns dispositivos de segurança, conforme figura 17. Na cozinha, será colocado um sensor de gás MQ2, enquanto no escritório será colocado um sensor de presença HC-SR501, além de um buzzer em cada um dos cômodos para funcionar como a sirene do alarme. Caso o alarme seja acionado pelo usuário, o buzzer irá tocar por um curto período de tempo, e o usuário também receberá uma notificação por email.

Figura 17 – Configuração da Etapa 3



Fonte: Imagens de dispositivos adaptadas de (FILIPE FLOP, 2019)

Com relação aos tópicos, conforme mostra a tabela 7, o ESP8266 do escritório agora também está inscrita no tópico “casa/escritório/esp8266/movimento”. A razão é mostrada na figura 18, onde o fluxo do Node-RED mostra que há dois botões (“Armar” e “Desarmar”) que estão ligados à uma saída MQTT, exatamente no tópico em que o microcontrolador está inscrito. Isso acontece para que ele possa em seu algoritmo desativar o alarme e as notificações caso o sistema seja desarmado, e ativá-los no caso contrário. O mesmo acontece com o ESP32 da cozinha, porém com o tópico “casa/escritório/esp32/fumaca”, com o intuito de armar ou desarmar o alarme de fumaça.

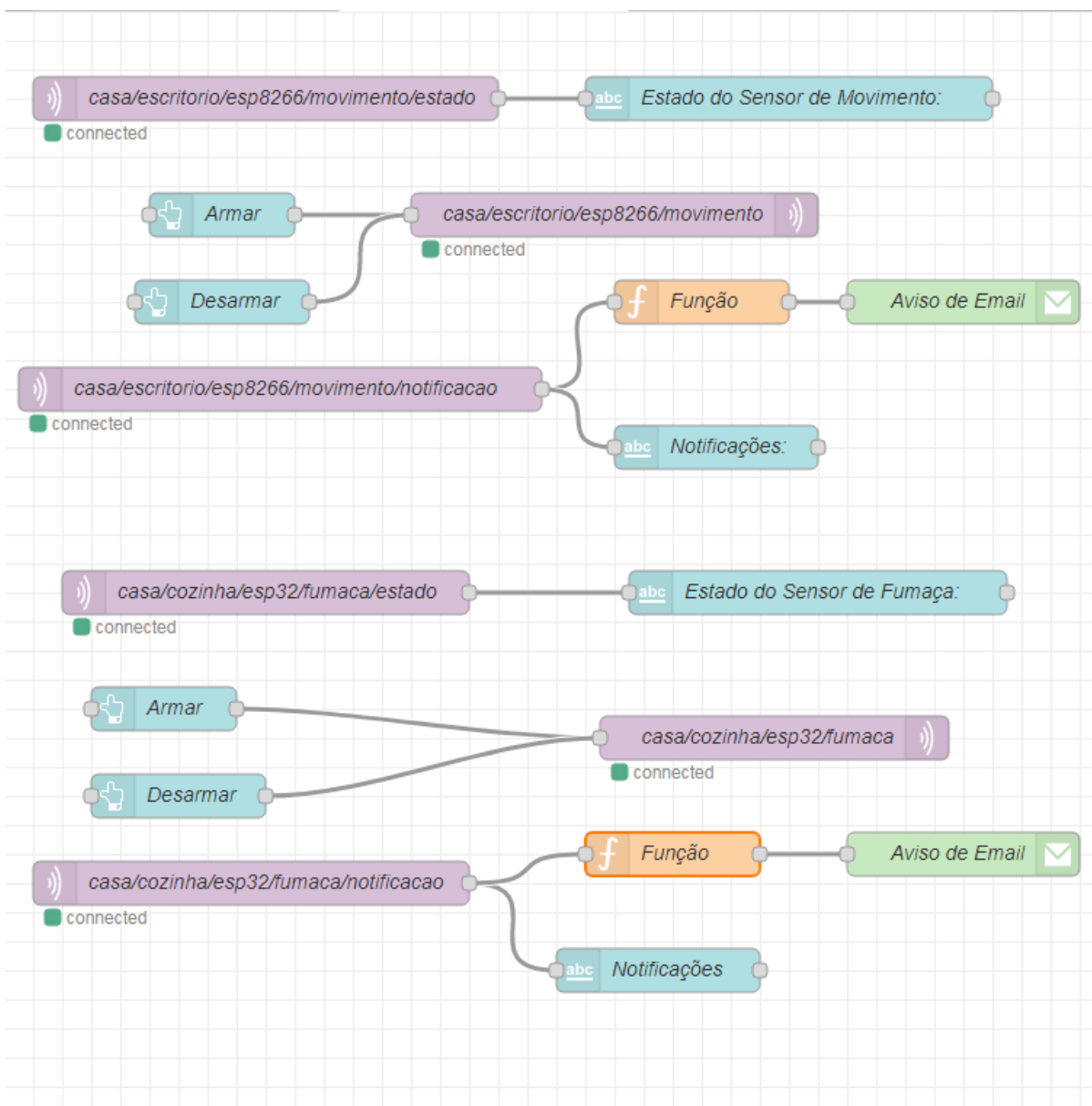
O ESP8266 também realiza publicações em dois novos tópicos: “casa/escritório/esp8266/movimento/estado” e também em “casa/escritório/esp8266/movimento/notificação”. O primeiro irá escrever uma mensagem na UI indicando o estado atual do alarme: “Armado” ou “Desarmado”, enquanto o segundo irá notificar sobre a presença ou não de movimento no escritório.

O mesmo acontece com o ESP32 da Cozinha que agora irá publicar nos tópicos: “casa/cozinha/esp32/fumaça/estado” e “casa/cozinha/esp32/fumaça/notificação”, sendo que a única diferença é que o estado e a notificação se referem agora ao alarme de fumaça da cozinha.

Tabela 7 - Inscrições e Publicações realizadas na Etapa 3

Dispositivo	Inscrição	Publicação
ESP8266 (Escritório)	casa/escritório/esp8266/ movimento	casa/escritório/esp8266/temperatura
		casa/escritório/esp8266/umidade
	casa/escritório/esp8266/ tomada	casa/escritório/esp8266/movimento/ estado
	casa/escritório/esp8266/ luz	casa/escritório/esp8266/movimento/ notificação
ESP32 (Cozinha)	casa/esp32/cozinha/fumaça	casa/cozinha/esp32/temperatura
		casa/cozinha/esp32/umidade
	casa/cozinha/esp32/ tomada	casa/cozinha/esp32/fumaça/estado
	casa/cozinha/esp32/ luz	casa/cozinha/esp32/fumaça/notificação

Figura 18 - Fluxo do Node-RED para a Etapa 3



Já no Node-RED, os fluxos de nós utilizados são aqueles indicados na figura x. Na metade superior da imagem estão os fluxos para o alarme de movimento do escritório, enquanto na metade de baixo estão aqueles utilizados para o alarme de fumaça da cozinha.

Como ambos são parecidos, sendo a única diferença os tópicos utilizados, a explicação do funcionamento é a mesma para ambos. Primeiro é formado um fluxo entre um nó de entrada MQTT de estado do alarme com um nó de saída de texto. Isso significa que o Node-RED está inscrito no tópico indicado pelo nó e irá receber as

informações que forem publicadas nele. Portanto, quando o microcontrolador publicar a mensagem indicando o estado do alarme, o Node-RED irá informar na UI.

Foram colocados também nós com botões para armar e desarmar os alarmes, que publicarão uma mensagem que poderá ser lida pelos dispositivos que estão inscritos nos tópicos respectivos. Por fim foi feito um fluxo para cada alarme que serve para enviar uma notificação ao usuário tanto pela UI do Node-RED, como por e-mail. O nó de função conectado ao nó de email irá verificar se a mensagem recebida indica que foi detectado movimento no escritório (ou fumaça no caso da cozinha) e, caso positivo, irá enviar um email ao usuário informando que o fato ocorreu.

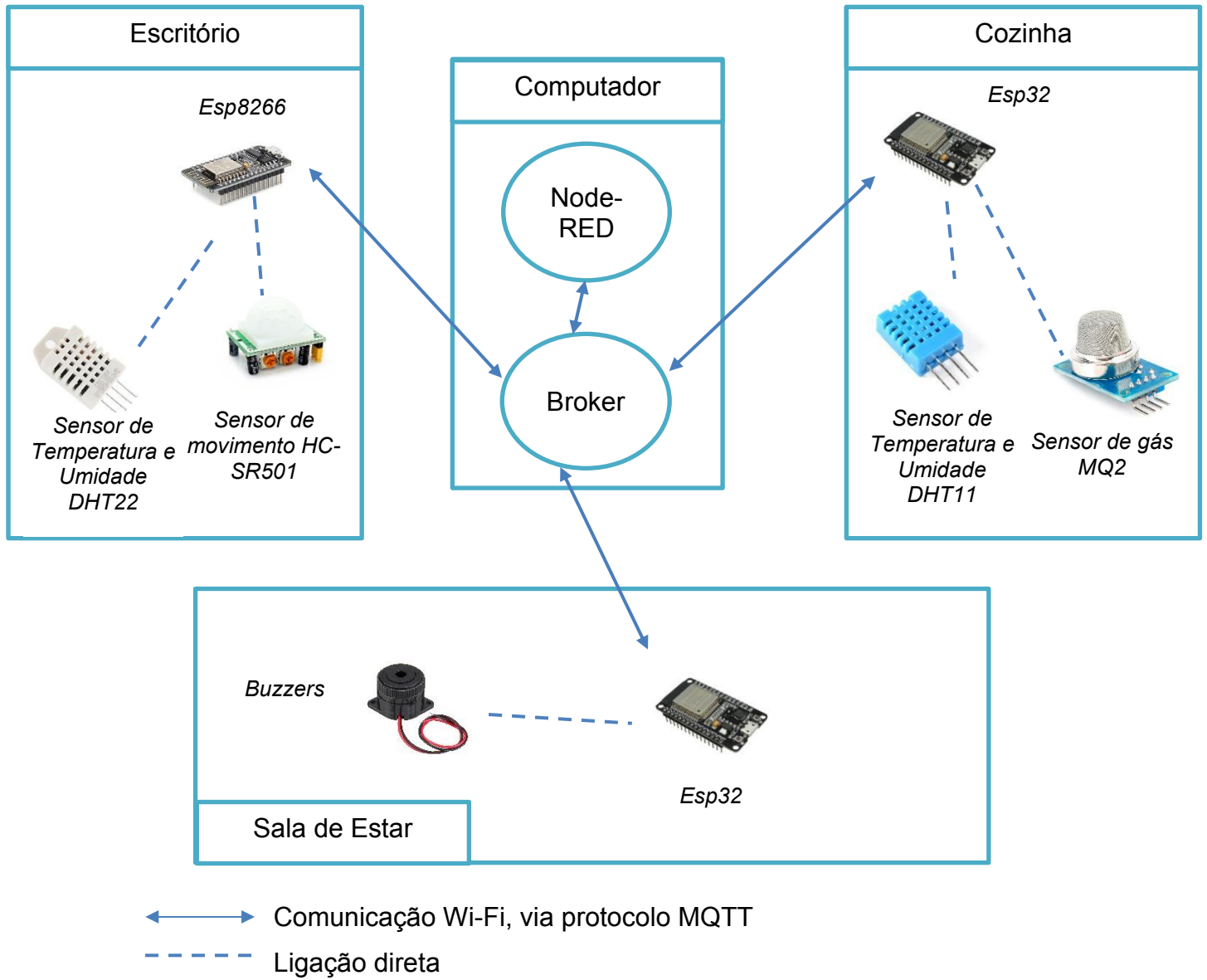
3.4 Alternativa: Separação dos Buzzers

O objetivo dessa etapa alternativa é separar os buzzers de alarme, tanto na cozinha como no escritório, ligando-os a uma nova placa ESP-WROOM-32 com microcontrolador ESP32 em uma sala de estar, conforme figura x. Dessa forma, foi possível testar a comunicação entre microcontroladores, verificando se os microcontroladores do escritório e da cozinha são capazes de se comunicar com os da sala de estar, já que agora os buzzers de alarme serão acionados por esta última.

Foram retirados, portanto, os buzzers dos circuitos da cozinha e do escritório, sendo eles agora adicionados ao circuito da sala de estar, formando o circuito da figura 19, diferente daquele formado anteriormente.

Não foram feitas alterações aos fluxos de Node-RED formados nas etapas anteriores. Nos microcontroladores, foi removido os trechos de código que realizavam o acionamento dos buzzers, tanto no ESP8266 do escritório, como no ESP32 da cozinha. O acionamento dos buzzers ficou agora sob a responsabilidade do ESP32 da sala de estar, que agora está inscrita nos tópicos “casa/escritório/esp8266/notificação” e “casa/cozinha/esp32/notificação”, conforme tabela 8, de forma que os buzzers serão acionados caso a placa da sala receba uma mensagem nesse tópico com a informação de que os alarmes devem ser disparados.

Figura 19 – Configuração da Alternativa



Fonte: Imagens de dispositivos adaptadas de (FILIPE FLOP, 2019)

Tabela 8 - Inscrições e Publicações realizadas na Etapa 4

Dispositivo	Inscrição	Publicação
ESP8266 (Escritório)	casa/escritório/esp8266/ movimento	casa/ escritório/esp8266/temperatura
		casa/escritório/esp8266/umidade
	casa/escritório/esp8266/ tomada	casa/escritório/esp8266/movimento/ estado
	casa/escritório/esp8266/ luz	casa/escritório/esp8266/movimento/ notificação
ESP32 (Cozinha)	casa/esp32/cozinha/fumaça	casa/cozinha/esp32/temperatura
	casa/cozinha/esp32/ tomada	casa/cozinha/esp32/umidade
	casa/cozinha/esp32/ luz	casa/cozinha/esp32/fumaça/estado
		casa/cozinha/esp32/fumaça/notificação
ESP32 (Sala de Estar)	casa/escritório/esp8266/ notificação	Nenhuma
	casa/cozinha/esp32/ notificação	

4 RESULTADOS E DISCUSSÕES

Nessa seção serão mostrados os resultados obtidos na UI fornecida pelo Node-RED. A UI pode ser acessada através de qualquer computador ou dispositivo móvel conectado à mesma rede Wi-Fi que o computador no qual o node-RED está rodando, utilizando o endereço IP do mesmo.

Ao acessar a UI, o usuário é recebido com uma mensagem de boas-vindas (figura 20), e com a opção de acionar duas chaves chamadas de “chaves mestras”, sendo um para iluminação e outro para tomadas. Tais chaves foram configurados conforme figura 14 para ligarem ou desligarem todos os leds relacionados à iluminação e às tomadas de acordo com o botão respectivo.

A partir dessa página o usuário pode agora acessar telas individuais para cada cômodo da casa, ao clicar no ícone de menu no canto superior esquerdo, abrindo a aba mostrada na figura 21.

Figura 20 – Tela Inicial (Home) da UI do Node-RED

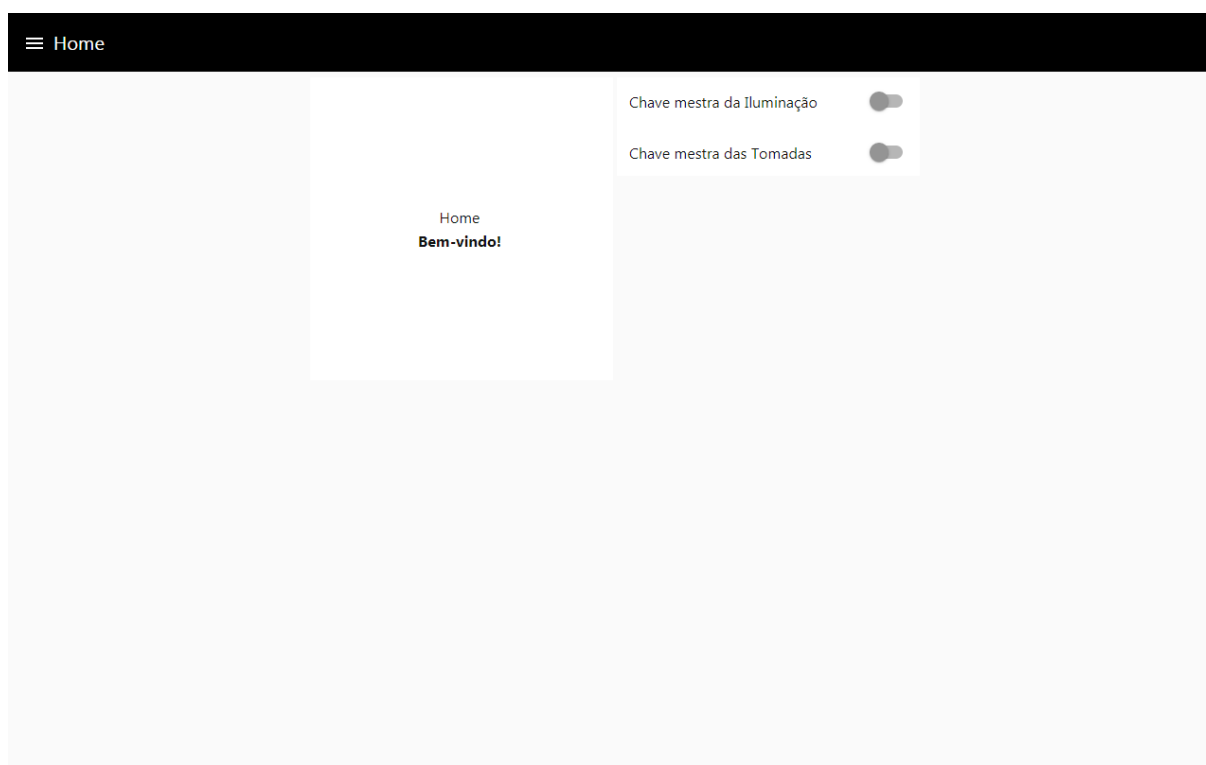
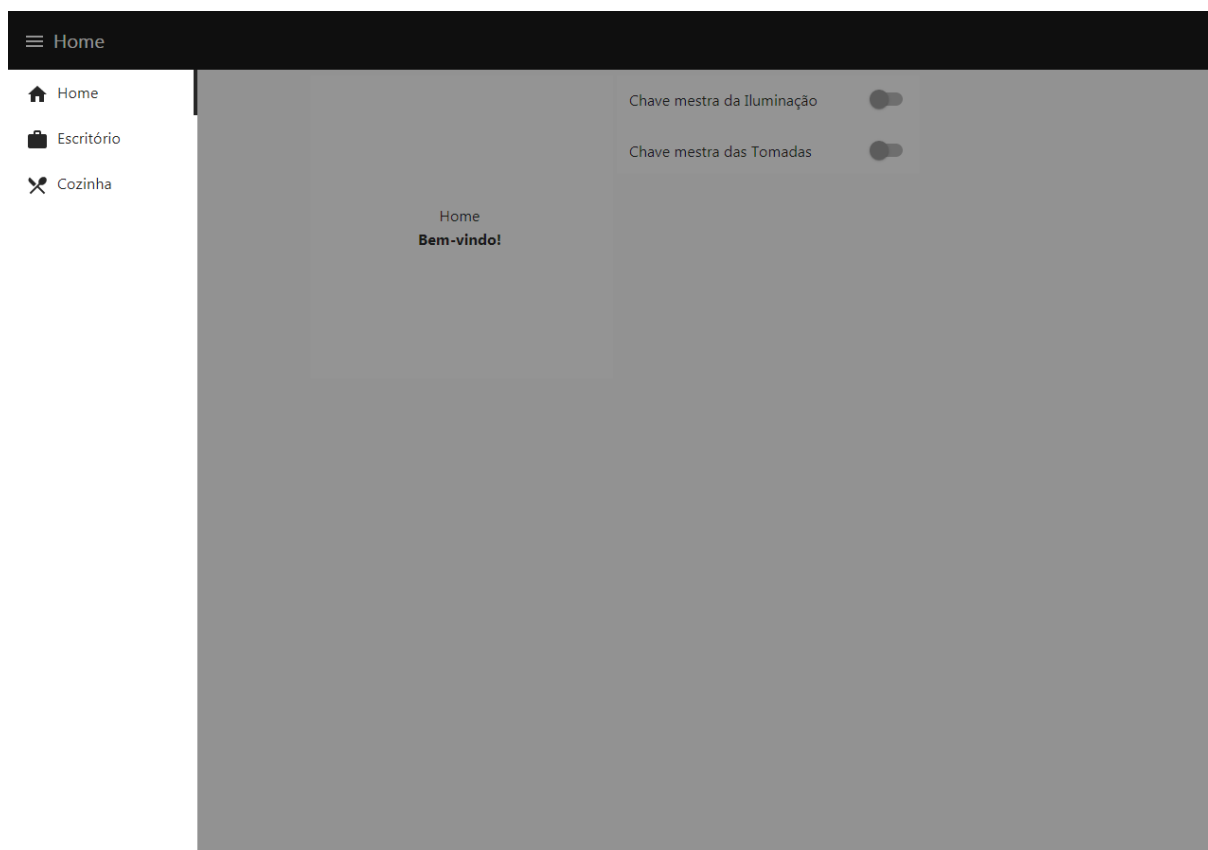


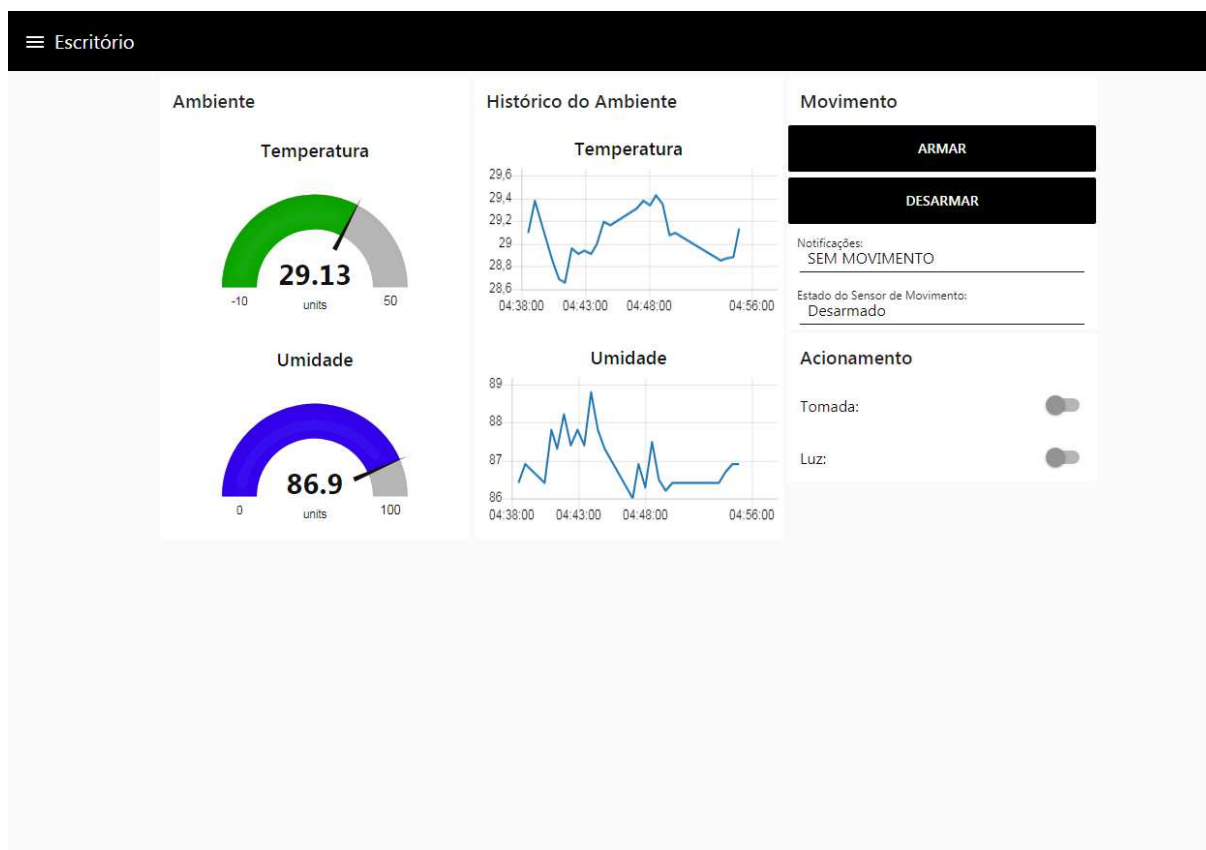
Figura 21 – Barra de opções da UI do Node-RED



4.1 Escritório

Ao clicar no item escritório, o usuário será levado à página mostrada na figura 22. Nela estão mostradas as informações de temperatura e umidade coletadas pelo sensor DHT22, conforme o que foi construído na Etapa 2 da metodologia. As informações são mostradas tanto através de indicadores como através de gráficos que indicam o histórico das informações, podendo indicá-las por até no máximo um intervalo de uma hora, sendo que esse intervalo pode ser configurado no Node-RED.

Figura 22 – UI do Escritório



Os indicadores de Temperatura e Umidade apresentam cores para ajudar a informar o usuário da situação em que seu cômodo se encontra. No caso da temperatura, o indicador apresentará cor azul em temperaturas no intervalo de -10°C a 15°C , cor verde para a faixa de 15°C até 30°C e, finalmente, cor vermelha entre 30°C e 50°C . Em outras palavras, a cor azul foi escolhida para representar temperaturas muito baixas, a cor vermelha para temperaturas muito altas, e a cor verde para temperaturas agradáveis. Para o indicador de umidade, foi escolhida a cor amarela para indicar níveis de umidade entre 0% e 30%, a cor verde para a faixa de 30% a 70% e a cor azul para níveis entre 70% e 100%.

A página de Escritório também conta com informações a respeito do alarme de movimento. É possível ao usuário armar ou desarmar o alarme e verificar o estado atual do sensor e, caso esteja armado, conferir se há movimento ou não. A figura 23 mostra os estados possíveis do sensor e das notificações. Quando o usuário pressiona o botão de “ARMAR” o estado altera para “Armado”. A partir de então, se um movimento for detectado, a notificação mostrará a mensagem: “MOVIMENTO DETECTADO”.

Por fim, a página conta com uma área de “Acionamento”. Nela é possível ligar e desligar os dois leds ligados ao microcontrolador do escritório que representam uma tomada e a lâmpada do próprio cômodo. Os botões também mudarão de estado caso a chave mestra da figura 20 que lhes corresponde seja acionada. Em uma aplicação real, tal lógica poderia ser utilizada para o acionamento de tomadas e lâmpadas controladas por rádio ao invés de leds, com o uso de receptores e transmissores de rádio frequência.

Figura 23 – Estados do alarme de movimento

The figure displays three screenshots of a mobile application interface for a motion alarm system, arranged in a grid. Each screenshot shows a title bar, two large buttons, and two text labels with values.

Estado	Notificações	Estado do Sensor de Movimento
Desarmado	SEM MOVIMENTO	Desarmado
Armado	SEM MOVIMENTO	Armado
Movimento Detectado	MOVIMENTO DETECTADO	Armado

4.2 Cozinha

Ao clicar no item cozinha, é exibida ao usuário uma tela praticamente idêntica à mostrada no item anterior, porém relacionada ao sistema montado para a cozinha mostrada na figura 24. Os medidores e gráficos de umidade e temperatura agora revelam as medidas coletadas pelo sensor DHT11 conectado ao ESP32 na cozinha. Há novamente uma seção de acionamento, para permitir que o usuário acenda os leds que simulam tomadas e iluminação. Por fim, o estado do alarme de fumaça é exibido, com dois botões que permitem que o usuário arme ou desarme o sistema. Os estados possíveis são exibidos na figura 25.

Figura 24 – UI da Cozinha

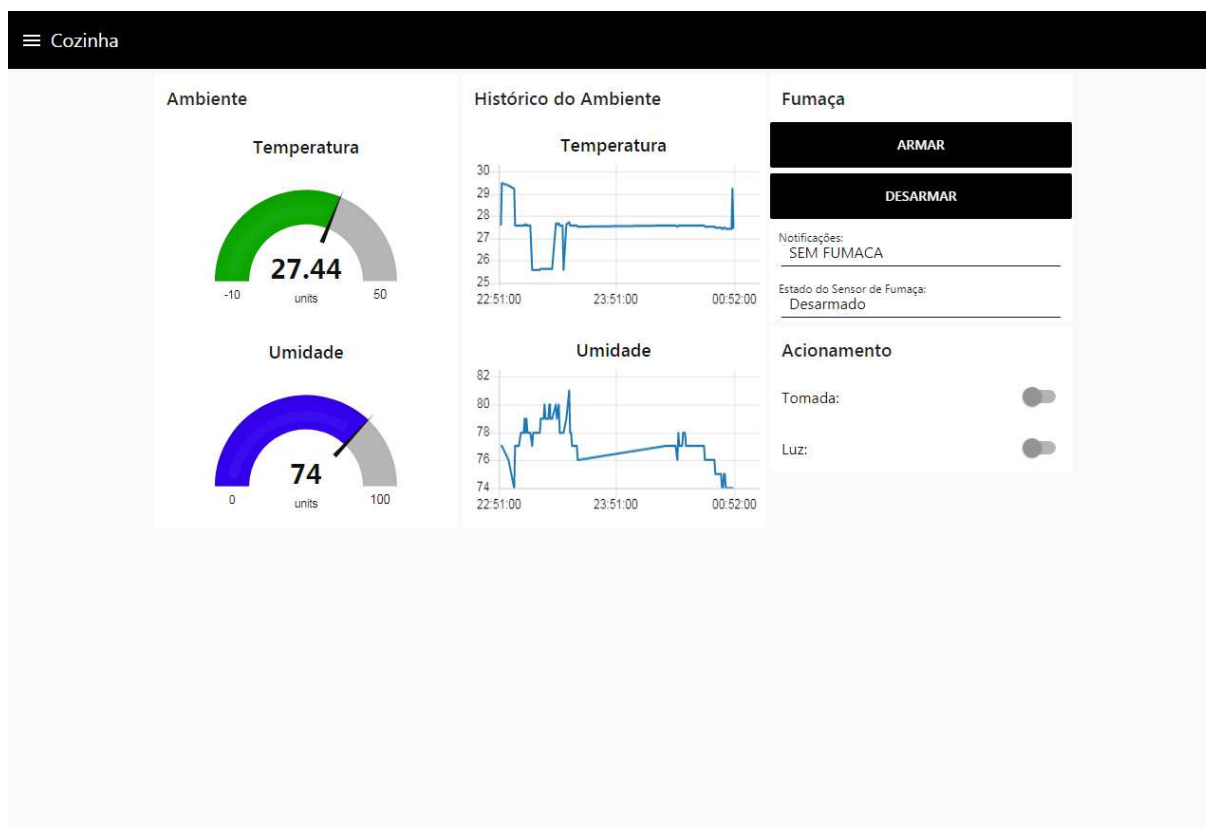


Figura 25 – Estados do alarme de fumaça

Fumaça	Fumaça
ARMAR	ARMAR
DESARMAR	DESARMAR
Notificações: SEM FUMACA	Notificações: SEM FUMACA
Estado do Sensor de Fumaça: Desarmado	Estado do Sensor de Fumaça: Armado

Fumaça
ARMAR
DESARMAR
Notificações: FUMACA DETECTADA
Estado do Sensor de Fumaça: Armado

4.3 Alarmes

O último passo da metodologia se referia à troca de posição dos buzzers, tirando-os da cozinha e do escritório e os colocando para serem acionados pelo ESP32 da sala de estar. O resultado foi que o sistema continuou funcionando corretamente, mesmo com os buzzers separados dos sistemas a qual eles se relacionam. Tanto o buzzer acionado pela detecção de fumaça na cozinha, quanto o buzzer acionado pelo sensor de presença no escritório agora são acionados a partir do ESP32 da Sala de Estar.

5 CONCLUSÃO

Através desse trabalho foi possível criar uma solução de automação residencial usando protocolo MQTT, através da proposta que envolveu a utilização da ferramenta Node-RED e do software Mosquitto Broker, além dos microcontroladores ESP32 e ESP8266 e de vários tipos de sensores e atuadores.

Considerando as características do protocolo MQTT, principalmente sua flexibilidade e leveza, foi possível notá-las na prática neste trabalho. A flexibilidade foi notada pelo fato dele ser capaz de se comunicar com dois tipos diferentes de microcontroladores e, além disso, permitir facilmente a troca de dispositivos, como mostrado na etapa 4 da metodologia, quando o buzzer foi trocado de lugar e ainda assim o sistema continuou funcionando como anteriormente, apenas com a troca de dispositivos assinantes do tópico de notificação. Além disso, foi possível notar que mesmo com dois dispositivos que não comunicam entre si, o broker é capaz de intermediar a comunicação, permitindo que o ESP32 da sala atuasse com base em informações provenientes do ESP32 da cozinha e do ESP8266 do escritório.

O trabalho também permitiu considerar o estado atual da área de automação residencial, com a proposta de uma solução que levou isso em conta. A solução permite ao usuário monitorar temperatura, acompanhar o estado de alarmes de segurança e atuar sobre eles e ainda acionar saídas com a simples mudança do estado de uma chave.

A supervisão da temperatura auxilia no conforto, pois permite que o usuário tenha uma ideia melhor das condições ambientes de sua casa, de forma que possa se preparar para enfrentar situações de muito calor ou de muito frio. Em um futuro trabalho, poderia ser adicionado um sistema de atuação na temperatura e umidade que permitisse alterá-las através da UI. Uma forma de fazê-lo seria com ar condicionados, ventiladores e umidificadores acionados por tomadas controladas de forma que o usuário pudesse acioná-las pela UI.

Enquanto isso, os alarmes de gás e de presença auxiliam na área de segurança. O alarme de gás na cozinha pode detectar o vazamento de gás de cozinha ou a presença de fogo pela fumaça. Já o alarme de presença no escritório pode detectar invasores não desejados e ajudar o usuário a tomar atitudes a respeito.

A etapa 1 da metodologia mostrou o acionamento de Leds pelo Node-RED que representavam tomadas e lâmpadas. Em um futuro trabalho poderiam ser utilizadas

lâmpadas e tomadas controladas remotamente, de forma que fosse possível contribuir com a eficiência energética da casa, ajudando a não só lembrar de apagar luzes, mas também a fazê-lo de forma prática e simples, mesmo para pessoas com dificuldades de locomoção.

A alternativa para a Etapa 3 mostrada no item 3.4 permitiu mostrar a flexibilidade do sistema proposto, pois mostrou que é possível trocar dispositivos de lugar dentro do sistema com facilidade, desde que a comunicação MQTT continue sendo obedecida, por causa do broker. No caso tudo o que foi necessário foi tornar o microcontrolador da sala assinante do tópico de notificação de alarme, de forma que ele pudesse atuar quando essa notificação indicasse a necessidade. Mesmo os sistemas se comunicando entre si, o broker é capaz de intermediar tal ligação.

Um trabalho futuro também poderia melhorar esse sistema estabelecendo conexões com redes não locais. Para isso poderia, por exemplo, ser usado um túnel que fosse capaz de tornar o endereço local do Node-RED em um endereço que pode ser acessado de qualquer lugar com conexão à internet.

Concluindo, este trabalho foi capaz de estudar a aplicação de ferramentas que tem o intuito de tornar o uso e a aplicação da tecnologia mais acessíveis, procurando auxiliar a encontrar um padrão que torne a tecnologia cada vez mais atrativa para os moradores e para a sociedade, com o objetivo de fornecer conforto, segurança e economia de energia de forma prática.

REFERÊNCIAS

ADAFRUIT. *DHT11, DHT22 and AM2302 Sensors*. 2019. Disponível em: <<https://www.mouser.com/datasheet/2/737/dht-932870.pdf>>. Acesso em: 28 Nov. 2019.

ADAFRUIT. *PIR Motion Sensor*. 2018. Disponível em: <<https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>>. Acesso em: 28 nov. 2019.

AKRAM, Omar Khasro; FRANCO, Daniel José; ISMAIL, Sumarni. Smart buildings—A new environment (Theoretical approach). *International Journal of Engineering Technology, Management and Applied Sciences*, v. 4, n. 4, p. 1-5, 2016.

ALLAFI, Ibrahim; IQBAL, Tariq. Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring. In: *2017 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, 2017. p. 1-5.

ARUL, S. Benjamin. Wireless home automation system using zigbee. *International Journal of Scientific & Engineering Research*, v. 5, n. 12, p. 133-138, 2014.

BISWAS, Shatadru Bipasha; IQBAL, M. Tariq. Solar water pumping system control using a low cost ESP32 microcontroller. In: *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*. IEEE, 2018. p. 1-5.

DE OLIVEIRA, Sérgio. *Internet das coisas com ESP8266, Arduino e Raspberry Pi*. Novatec Editora, 2017.

DEY, Kakan Chandra et al. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network—Performance evaluation. *Transportation Research Part C: Emerging Technologies*, v. 68, p. 168-184, 2016.

ESPRESSIF. *ESP32-WROOM32 Datasheet*. 2019. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Acesso em: 28 nov. 2019.

ESPRESSIF. *ESP8266EX Datasheet*. 2019. Disponível em: <https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>. Acesso em: 28 nov. 2019.

FILIPE FLOP. *Módulo WiFi ESP32 Bluetooth*. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/>>. Acesso em: 2 dez. 2019a. *Módulo WiFi ESP8266 NodeMcu ESP-12*. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/>>. Acesso em: 2 dez. 2019b. *Sensor de Umidade e Temperatura DHT11*. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-dht11/>>. Acesso em: 2 dez. 2019c. *Sensor de Gás MQ-2 Inflamável e Fumaça*. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-gas-mq-2-inflamavel-e-fumaca/>>.

Acesso em: 2 dez. 2019d. *Sensor de Movimento Presença PIR*. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-movimento-presenca-pir/>>. Acesso em: 2 dez. 2019e. *Módulo Buzzer Piezoelétrico*. Disponível em: <<https://www.filipeflop.com/produto/modulo-buzzer-piezoelétrico/>>. Acesso em: 2 dez. 2019f. *Sensor de Umidade e Temperatura AM2302 DHT22*. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/>>. Acesso em: 2 dez. 2019g. *Sensor de Umidade e Temperatura AM2302 DHT22*. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-umidade-e-temperatura-am2302-dht22/>>. Acesso em: 2 dez. 2019h. *LED Difuso Verde*. Disponível em: <<https://www.filipeflop.com/produto/led-difuso-5mm-verde-x10-unidades/>>. Acesso em: 2 dez. 2019i. *LED Difuso Vermelho*. Disponível em: <<https://www.filipeflop.com/produto/led-difuso-5mm-vermelho-x10-unidades/>>. Acesso em: 2 dez. 2019j. *LED Difuso Amarelo*. Disponível em: <<https://www.filipeflop.com/produto/led-difuso-5mm-amarelo-x10-unidades/>>. Acesso em: 2 dez. 2019k.

GAO, Qian; GAO, Xianwen; CHEN, Xiuliang. Analysis of current situation and existing problems in greenhouse environment control. In: *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*. IEEE, 2014. p. 4516-4520.

GUBBI, Jayavardhana et al. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, v. 29, n. 7, p. 1645-1660, 2013.

HUNKELER, Urs; TRUONG, Hong Linh; STANFORD-CLARK, Andy. 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)*. IEEE, 2008. p. 791-798.

KRISHNA, Ishan; LAVANYA, K. Intelligent home automation system using BitVoicer. In: *2017 11th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2017. p. 14-20.

KOYANAGI, Fernando. Introdução ao ESP32. 2017. Disponível em: <<https://www.fernandok.com/2017/11/introducao-ao-esp32.html>>. Acesso em: 28 nov. 2019.

LIGHT, Roger A. et al. Mosquitto: server and client implementation of the MQTT protocol. *J. Open Source Software*, v. 2, n. 13, p. 265, 2017.

MAIER, Alexander; SHARP, Andrew; VAGAPOV, Yuriy. Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. In: *2017 Internet Technologies and Applications (ITA)*. IEEE, 2017. p. 143-148.

MURATORI, J. R. Os desafios do mercado de Automação Residencial. 2013. Disponível em: <http://www.aecweb.com.br/cont/a/os-desafios-do-mercado-daautomacao-residencial_8192>. Acesso em: 16 out. 2019.

MURATORI, José Roberto; DAL BÓ, Paulo Henrique. Automação Residencial: Conceitos e Aplicações. *Belo Horizonte: Educere*, 2013.

NAIK, Nitin. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In: *2017 IEEE international systems engineering symposium (ISSE)*. IEEE, 2017. p. 1-7.

OPEN CIRCUIT. SFM27 Datasheet. [S.l.]. Disponível em: <<https://opencircuit.shop/resources/file/7379f0193d4549c771f0d61bd31d06b316cd4706466/SFM27-Datasheet.pdf>>. Acesso em: 28 nov. 2019.

POLOLU. MQ-2 Datasheet. [S.l.]. Disponível em: <<https://www.pololu.com/file/0J309/MQ2.pdf>>. Acesso em: 28 nov. 2019.

RAI, Pertab; REHMAN, Murk. ESP32 Based Smart Surveillance System. In: *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2019. p. 1-3.

SILVA, Mauricio César; GAMBARATO, Vivian Toledo Santos. Domótica e tecnologias utilizadas na automação residencial. *Tekhne e Logos*, v. 7, n. 2, p. 56-67, 2016.

SEROZHENKO, Marina. MQTT vs HTTP: which one is the best for IoT? 2017. Disponível em: < <https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105>>. Acesso em: 1 nov. 2019.

ŠKRABA, Andrej et al. Prototype of Group Heart Rate Monitoring with ESP32. In: *2019 8th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2019. p. 1-4.

SONI, Dipa; MAKWANA, Ashwin. A survey on mqtt: a protocol of internet of things (iot). In: *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*. 2017.

SOUZA, Marcelo Varela de. *Domótica de baixo custo usando princípios de IoT*. 2016. Dissertação de Mestrado. Brasil.

TORRES, Andrei BB; ROCHA, Atslands R.; DE SOUZA, José Neuman. Análise de desempenho de brokers mqtt em sistema de baixo custo. In: *Anais do XXXVI congresso da sociedade brasileira de computação. Sociedade Brasileira de Computação*. 2016.

UPADHYAY, Yuvraj; BOROLE, Amol; DILEEPAN, D. MQTT based secured home automation system. In: *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016. p. 1-4.

YUAN, Michael. Conhecendo o mqtt. *IBM*, 2017. Disponível em: <<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>>. Acesso em: 1 nov. 2019.